



HAL
open science

Contributions au développement d'un solveur volumes finis sur grille cartésienne localement raffinée en vue d'application à l'hydrodynamique navale

Louis Vittoz

► **To cite this version:**

Louis Vittoz. Contributions au développement d'un solveur volumes finis sur grille cartésienne localement raffinée en vue d'application à l'hydrodynamique navale. Mécanique des fluides [physics.class-ph]. École centrale de Nantes, 2018. Français. NNT : 2018ECDN0020 . tel-01938995

HAL Id: tel-01938995

<https://theses.hal.science/tel-01938995>

Submitted on 29 Nov 2018

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.

THESE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 602
Sciences pour l'Ingénieur
Spécialité : Mécanique des milieux fluides

Par

Louis VITTOZ

Contributions au développement d'un solveur volumes finis sur grille cartésienne localement raffinée en vue d'application à l'hydrodynamique navale

Thèse présentée et soutenue à l'Ecole Centrale de Nantes, le 10/09/2018
Unité de recherche : UMR 6598

Rapporteurs avant soutenance :

Raphaël LOUBERE, Directeur de Recherche CNRS,
Université de Bordeaux

Stéphane POPINET, Directeur de Recherche CNRS,
Sorbonne Université

Composition du Jury :

Président

Mazen SAAD, Professeur des universités, Ecole Centrale de Nantes

Examineurs

Lisl WEYNANS, Maître de conférences, Université de Bordeaux

Invité

Jean-Luc ESTIVALEZES, Professeur associé (HDR), ONERA

Directeur de thèse

David LE TOUZE, Professeur des universités, Ecole Centrale de Nantes

Co-encadrants de thèse

Guillaume OGER, Ingénieur de recherche, Ecole Centrale de Nantes

Zhe LI, Maître de conférences, Ecole Centrale de Nantes

Remerciements

Je souhaiterais en tout premier lieu exprimer ma profonde gratitude à Messieurs Raphaël Loubère et Stéphane Popinet, rapporteurs de cette thèse, pour leur lecture attentive de ce manuscrit et leurs commentaires éclairés. Je suis tout aussi reconnaissant à Madame Lisl Weynans et Messieurs Jean-Luc Estivalezes et Mazen Saad d'avoir accepté de faire partie de mon jury.

D'autre part je voudrais remercier les instigateurs de cette thèse, Pierre-Michel Guilcher, Erwan Jacquin et David Le Touzé, d'avoir convaincu de se lancer dans cette aventure, le profane que j'étais et qui ignorait tout de la richesse de la recherche académique.

Ma reconnaissance va aussi à l'entreprise Nextflow Software pour le financement de cette thèse et pour l'expérience pointue de ses collaborateurs dont j'ai pu bénéficier.

Je sais gré à Guillaume, Zhe, Matthieu et David pour leur encadrement exceptionnel sans lequel cette thèse n'aurait pas abouti aussi favorablement. Plus particulièrement, un grand merci à toi Guillaume, pour ta disponibilité, ton soutien sans faille et tes accueils toujours chaleureux et enthousiastes dans ton bureau. Je ne pense pas me tromper en affirmant que tous les problèmes techniques soulevés au cours cette thèse ont trouvé solution lors d'échanges constructifs sur les passerelles du bâtiment D.

Je voudrais dire un grand merci à Élodie, Sonia et Anne sans qui les démarches administratives ne paraîtraient pas aussi simples.

Par ailleurs je souhaiterais remercier les doctorants du labo, les courageux qui débutent comme les impatientes qui attendent la fin, mais aussi tous les autres membres du LHEEA qui participent à l'ambiance conviviale du labo : Vincent L & A, Marion, Laetitia, PY, Marco, mes co-bureaux Boris et Young Myung, Catherine, Julien, Imad, Paul-Emile, Nicolas, Zhaobin, Wenjin, Gaël, Chunmei, Alban, Paul, Sijo, Zhenrong sans oublier les anciens : Hélène, Marie C & R, Camille, Francesc, Coirentin, Laurent, Charles, Maïté, Baptiste, Benjamin, Yoann, Patrick.

Je suis reconnaissant à mes proches qui m'ont toujours soutenu tout au long de cette thèse, Jo, mes parents et mon frère.

Pour finir je remercie de façon informelle et dans le désordre les personnes ou toutes ces petites choses de la vie qui m'ont accompagné pendant ces trois années de thèse et ont contribué à les rendre plus belles : la coinche au TU, les débats au RU, le bâtiment E fraîchement rénové, Tanguy Pastureau, l'émission Par Jupiter, Noëlle du RU, l'anesthésiste de la clinique de la main, B. Pavard, K. Mbappé, M. Fourcade et les biathlètes français, PAB, R. Lavillenie, la team Tupperware, la SNCF, la série Stranger Things, Burger Quiz, Hubert Bonisseur de La Bath, les Nuits de l'Erdre, Fakear, Deezer, la chocolate house, le Berlin, le Vieux Quimper, la Fraiseriaie, l'usine Cointreau d'Angers, les reblochons Agnellet, les tisanes les 2 marmottes, les pâtisseries 5 étoiles cuisinées par PY, la croziflette, sans oublier certains lieux géniaux, Belle-Île-en-mer, Saint-Malo, Batz-sur-mer, l'Inde, l'île de Mull, la Moselle, Vilnius, Vienne, et pour clore cette liste non exhaustive : la Haute-Savoie qui tient une grande place dans ma vie.

"À une époque où tout est de plus en plus planifié, programmé, organisé,
pouvoir se perdre sera bientôt un délice et un luxe exceptionnels."
La montagne est mon domaine, Gaston Rébuffat.

Table des matières

Introduction générale	17
1 Présentation de la méthode WCCH	23
1.1 Modèle mathématique	24
1.1.1 Équations de Navier-Stokes en formulation compressible	24
1.1.2 Équations de Navier-Stokes en formulation incompressible	24
1.1.3 Stratégies de résolution des écoulements à bas nombre de Mach : approche faiblement-compressible ou purement incompressible	25
1.1.4 Conditions aux limites incompressibles	25
1.2 Description du solveur volumes finis de WCCH	27
1.2.1 Formulation volumes finis	27
1.2.2 Calculs des flux	28
1.2.3 Schéma d'avance en temps	32
1.2.4 Méthode de frontière immergée	33
1.2.5 Maillage et structure de données	35
1.2.6 Parallélisation	40
1.2.7 Exemple d'un écoulement autour d'un cylindre simulé avec WCCH	41
1.3 Problématiques actuelles du code WCCH	41
1.3.1 Limitations de l'approche faiblement-compressible pour la simulation à bas nombre de Mach	41
1.3.2 Précision et robustesse de la méthode de frontière immergée	44
1.3.3 Immersion de géométries complexes	45
1.4 Conclusion	45
2 Développement d'une formulation incompressible dans le code WCCH	47
2.1 Revue des schémas incompressibles	47
2.1.1 Rappel des équations en formulation incompressible	47
2.1.2 Méthodes à correction de pression	48
2.1.3 Méthodes de compressibilité artificielle	52
2.1.4 Bilan des méthodes incompressibles	53
2.2 Méthode de projection de Chorin et ses variantes	54
2.2.1 Méthode originelle	54
2.2.2 Déclinaisons	55
2.2.3 Schéma de projection finalement retenu	60
2.3 Traitements numériques	61
2.3.1 Calcul de la vitesse prédite	61
2.3.2 Résolution de l'équation de Poisson	62
2.3.3 Discrétisation temporelle	71

2.3.4	Conditions aux limites	73
2.4	Adaptation de la méthode de frontière immergée pour la formulation incompressible	77
2.5	Conclusion	81
3	Extension de la formulation pour des grilles raffinées localement	83
3.1	Présentation de la difficulté	83
3.1.1	Maillage raffiné localement et interpolation d'ordre élevé	83
3.1.2	Maillage raffiné localement et calcul de flux sur maillage MAC	84
3.2	État de l'art	84
3.2.1	Stratégies pour la résolution de l'équation de Poisson	85
3.2.2	Méthodes pour le calcul des flux	89
3.2.3	Bilan et choix d'une méthode	92
3.3	Discrétisation aux interfaces de raffinement	92
3.3.1	Discrétisation et implémentation des opérateurs divergence et gradient	92
3.3.2	Discrétisation des termes convectifs et diffusifs	94
3.4	Conclusion	102
4	Validations numériques et comparaisons incompressible/faiblement-compressible	105
4.1	Validation sur grille cartésienne uniforme	105
4.1.1	Précision et diffusion numérique	106
4.1.2	Solveur visqueux	108
4.2	Validation sur grille raffinée localement	113
4.2.1	Comparaison guard-cell/stretching	113
4.2.2	Écoulement laminaire autour d'un cylindre	121
4.3	Comparaison des formulations incompressible et faiblement-compressible	125
4.3.1	Diffusion numérique et précision	125
4.3.2	Écoulement de Taylor-Couette	128
4.3.3	Écoulement autour d'un profil NACA	132
4.3.4	Discussion sur les éléments de calcul haute performance	139
4.4	Conclusion	142
5	Analyse de stabilité et amélioration de la méthode de frontière immergée	145
5.1	Introduction	145
5.2	Revue des méthodes de frontière immergée	146
5.2.1	Continuous forcing	146
5.2.2	Discrete forcing	150
5.2.3	Calcul des efforts	160
5.2.4	Frontière immergée et corps mobiles	161
5.3	Position de la méthode initialement implémentée dans WCCH	162
5.4	Consistance, stabilité et convergence de la méthode	164
5.4.1	Simplification du cas d'étude	164
5.4.2	Position du problème	164
5.4.3	Précision de la prise en compte des conditions aux limites	165
5.4.4	Validations et expériences numériques	169
5.5	Amélioration et montée en ordre de la méthode de frontière immergée	171
5.5.1	Réduction de la sensibilité au pas de temps	171
5.5.2	Adaptation de la méthode pour les schémas Runge-Kutta	173
5.5.3	Vers une montée en ordre de la méthode	175
5.5.4	Validations numériques de la montée en ordre	179

5.6	Conclusions	187
6	Implémentation d'une méthode d'immersion de géométries	189
6.1	État de l'art	190
6.1.1	Problématique	190
6.1.2	Algorithmes de calcul de distance signée	190
6.1.3	Structure de partitionnement de la géométrie	196
6.1.4	Conclusion et choix d'une méthode	199
6.2	Implémentation de la librairie <i>libOL</i> dans WCCH	200
6.2.1	Description de la librairie <i>libOL</i>	200
6.2.2	Utilisation de <i>libOL</i> et calcul de distance signée dans WCCH	201
6.2.3	Prise en compte des corps mobiles	205
6.3	Applications à un écoulement d'hydrodynamique navale	206
6.3.1	Introduction	206
6.3.2	Paramétrisation du calcul	207
6.3.3	Résultats	209
6.4	Conclusion	215
	Conclusion et perspectives	217
A	Schéma JST et schémas similaires	235
B	Implémentation de PETSc dans le code WCCH	241
C	Impact d'une accélération ou de l'optimisation des algorithmes de WCCH	245
D	Implémentation d'un schéma PIMPLE incompressible dans WCCH	249
E	Implémentation d'un critère de raffinement basé sur la matrice hessienne de pression	259

Table des figures

1	Exemple d'optimisation de bulbe d'étrave pour la réduction de la consommation de carburant. <i>Crédits images Tecnicas-HydrOcean.</i>	18
2	Exemple de maillage complexe généré avec HEXPRESS TM . <i>Source : http://www.parsmakina.com/en/product/hexpress. Crédits IHC Dredger International et ECN/CNRS.</i>	18
3	Caractéristiques du solveur développé	19
4	Exemples de grilles cartésiennes multi-niveaux sur lesquelles s'appuie WCCH	20
5	Exemples de simulations réalisées à l'aide du code WCCH	20
1.1	Différences entre les spectres d'un écoulement réel à bas Mach pour l'approche compressible, faiblement-compressible et incompressible. ■ : partie purement incompressible, ■ : partie acoustique.	26
1.2	Illustration du problème de Riemann à l'interface entre deux cellules	29

1.3	Polynômes de reconstruction WENO 5	31
1.4	Reconstructions WENO en 2D par directions alternées.	32
1.5	Approche à maillage adapté ou "body-fitted" (à gauche) et immersion d'un cercle dans une grille cartésienne par frontière immergée (à droite)	34
1.6	Fonction de régularisation à la traversée de la frontière	35
1.7	Maillage cartésien structuré uniforme (à gauche) ou déstructuré, éléments triangles (à droite).	36
1.8	Centre de face e et intersection e'	36
1.9	Stratégie de raffinement de maillage cartésien autour d'un cercle	37
1.10	Raffinement par patches ou en octree	38
1.11	Raffinement par octree et numérotation avec le Z de Morton	39
1.12	Contrainte sur les niveaux de raffinement entre blocs voisins	40
1.13	Blocs LOCAL - INNER - OUTER	41
1.14	Validation de WCCH pour un écoulement à $Re = 100$ autour d'un cylindre	42
1.15	Ecoulement quasi 1D	43
2.1	Algorithme SIMPLE pour un calcul stationnaire ou un pas de temps d'un calcul instationnaire	51
2.2	Procédure de type Godunov pour le schéma BCG	57
2.3	Grille colocalisée et valeurs aux faces	58
2.4	Illustration du découplage pair-impair sur grille colocalisée	59
2.5	Maillage MAC avec variables entrelacées et cellules duales (délimitées par des tirets)	61
2.6	Emplacement des reconstructions WENO \times et calculs de flux \rightarrow (sur maillage MAC)	62
2.7	Principe des méthodes multigrilles	69
2.8	Extension du domaine de calcul aux frontières à l'aide de cellules fantômes pour les vitesses	74
2.9	Condition aux limites périodique dans la direction horizontale	74
2.10	Condition de glissement	75
2.11	Condition entrante et flux entrant	76
2.12	Condition sortante et reconstructions WENO disponibles \times et manquantes $*$	77
2.13	Gradient de la fonction d'amplification de l'accélération solide à la traversée de la frontière	79
2.14	Définition des cellules fluides \blacksquare , des cellules solides \blacksquare et des cellules de la zone de transition \blacksquare	79
3.1	Non conservation des flux \rightarrow et non coïncidence des volumes de contrôle	84
3.2	Définition des vitesses aux interfaces de raffinement	85
3.3	Calcul des gradients de pression à une interface de raffinement.	88
3.4	Structure des volumes de contrôle. De gauche à droite, maillage initial, maillage dual, maillage diamant. Extrait de Hubert et al. [Hubert et Tesson, 2017]	89
3.5	Diagramme de Voronoï à une interface de raffinement. Extrait de Guittet et al. [Guittet et al., 2015]	90
3.6	Opérations de restriction et de prolongation	91
3.7	Correction du flux grossier pour la conservation de la quantité de mouvement pour variables colocalisées	91
3.8	Étirement du volume de contrôle de la vitesse de la cellule grossière	91
3.9	Discrétisation des opérateurs divergence et gradient	93
3.10	Emplacement des variables des grilles fines \blacktriangle et grossières \blacktriangle pour les opérations de restriction-prolongation	94
3.11	Étirement du volume de contrôle associé à la vitesse d'une interface de raffinement	95
3.12	Reconstruction WENO et calcul des flux dans la direction normale aux interfaces de raffinement.	97
3.13	Mise en évidence du stencil avec recouvrement des volumes de contrôle aux interfaces de raffinement.	98

3.14	Calcul du flux tangentiel à l'interface de raffinement	99
3.15	Reconstruction WENO et calcul des flux dans la direction tangentielle aux interfaces de raffinement.	100
3.16	Interpolations tangentielles dans une direction supplémentaire en 3D pour les vitesses u symbolisées par $\mathbf{-}\blacktriangleright$	102
3.17	Interpolations tangentielles dans une direction supplémentaire en 3D pour les vitesses v symbolisées par $\mathbf{-}\blacktriangleright$	102
4.1	Champs initiaux du cas de convection de tourbillon	106
4.2	Comparaison des profils de pression (à gauche) et de vitesse (à droite) au bout de 40 passages	107
4.3	Comparaison des champs de vitesse horizontale pour les différents codes au bout de 40 passages	108
4.4	Comparaison des profils de vitesse V suivant x au 40 ^{ème} passage pour un maillage 60×60 .	109
4.5	Écoulement de Poiseuille	109
4.6	Écoulement de Poiseuille	110
4.7	Cavité entraînée	111
4.8	Iso-vorticité de la cavité entraînée à $Re = 1000$ pour un maillage 128×128	111
4.9	Profils de vitesse horizontale et verticale à $Re = 1000$	112
4.10	Lignes de courant de la cavité entraînée à $Re = 1000$	113
4.11	Maillages raffinés localement (imposés pour la conférence FVCA)	114
4.12	Pression et vitesse horizontale pour le cas test de Bercovier-Engelman	117
4.13	Pression et norme du vecteur vitesse pour l'écoulement vortex	118
4.14	Tourbillons instationnaires. Pression et vitesse initiales pour l'écoulement instationnaire . . .	119
4.15	Conditions de calcul et maillage ($\frac{D}{\Delta x} = 17$)	121
4.16	Pression et vitesse autour du cylindre à $Re = 40$ pour $\frac{D}{\Delta x} = 68$	122
4.17	Coefficient de pression sur le cylindre (gauche) et coefficient de traînée (droite)	123
4.18	Critères géométriques de la zone de recirculation	123
4.19	Écoulement autour du cylindre à $Re = 100$ en formulation incompressible pour le maillage $\frac{D}{\Delta x} = 68$ à $\frac{tu_\infty}{D} = 125$	124
4.20	Tourbillons de Taylor-Green. Comparaison de la pression à $\frac{tU_0}{L} = 3$ pour l'approche faiblement-compressible (gauche) et incompressible (milieu). Taux de dissipation de l'énergie cinétique en fonction du temps (droite)	126
4.21	Convection de vortex. Champ de pression à $t = 0$ (gauche) et $t = 10\frac{L}{U_0}$ avec 80×80 cellules en incompressible (milieu) et faiblement-compressible (droite).	127
4.22	Convection de vortex. Convergence en pression (gauche) et vitesse (droite).	128
4.23	Écoulement de Taylor-Couette	129
4.24	Écoulement de Taylor-Couette. Comparaison des champs de vitesse et de pression à $t = 230s$ en formulation faiblement-compressible (gauche) et incompressible (droite)	130
4.25	Écoulement de Taylor-Couette. Comparaison des profils de vitesse (gauche) et de pression (droite) à $t = 230s$	131
4.26	Écoulement de Taylor-Couette. Comparaison du temps CPU en fonction de l'erreur (gauche) et du gain incompressible/faiblement-compressible (droite)	132
4.27	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Domaine de calcul avec dimensions adimensionnalisées par la corde du profil (gauche) et maillage au voisinage du solide avec $\frac{c}{\Delta x} = 75$ (droite)	133
4.28	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Coefficient de traînée en fonction du temps.	134

4.29	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de vorticité à $\frac{tU_0}{D} = 100$. Comparaison incompressible (gauche), faiblement-compressible (droite), DVH (bas)	135
4.30	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de pression à $\frac{tU_0}{D} = 100$ pour $\frac{corde}{\Delta x} = 185$, incompressible (gauche) and faiblement-compressible (droite).	136
4.31	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Temps CPU moyen d'un pas de temps par cellule (gauche) et pourcentage du temps CPU consacré à la résolution de l'équation de Poisson en incompressible (droite).	138
4.32	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Convergence spatiale sur la traînée (gauche) et sur la portance (droite)	138
4.33	Étude de scalabilité. Incompressible (gauche) et faiblement-compressible (à droite)	140
4.34	Temps CPU d'un pas de temps par cellule et par processeur. Incompressible (gauche) et faiblement-compressible (à droite)	140
4.35	Accélération du solveur faiblement-compressible de WCCH par portage du code sur GPU.	141
5.1	Procédures d'interpolation possibles tirées de [Fadlun et al., 2000] : (a) pas d'interpolation, (b) pondération par la fraction de volume, (c) interpolation	152
5.2	Erreur de convergence sur la vitesse en fonction du nombre de points de calcul [Fadlun et al., 2000] : (●) pas d'interpolation, (■) pondération par la fraction de volume, (●) interpolation	152
5.3	Identification de l'intersection d'une cellule d'un maillage cartésien avec une frontière solide.	153
5.4	Comparaison des schémas d'interpolation des vitesses au bord d'un solide par la méthode de Fadlun (a) et de Balaras (b) (extrait de [Balaras, 2004])	154
5.5	Représentation du champ de vitesse initial (a), du champ de vitesse imposé après forçage (b) et du champ de vitesse après avance en temps (c) (extrait de [Mohd-Yusof, 1997]). Le point (■) représente le point où la vitesse est inversée, (□) est le point fluide et (●) le point frontière sur lequel la vitesse est imposée.	156
5.6	(a) Représentation du domaine de calcul avec les points de calcul fluide (×) et les cellules fantômes (Δ). (b) Représentation des points d'appui utilisés pour l'extrapolation (extrait de [Tseng et Ferziger, 2003])	157
5.7	Deux solutions pour traiter les points du fluide proches de la frontière solide (extrait de [Tseng et Ferziger, 2003])	158
5.8	Exemple de redécoupage géométrique de cellules intersectées par la frontière (extrait de [Ye et al., 1999])	159
5.9	Méthode LS-STAG (extrait de [Chen et Botella, 2010])	159
5.10	Calcul des efforts appliqués à un corps par application de la conservation de la quantité de mouvement	160
5.11	Discrétisation du domaine pour l'analyse de la méthode de frontière immergée.	165
5.12	Comparaison de la sensibilité à la position de la frontière avec et sans régularisation du Heaviside	170
5.13	Courbes de convergence de la méthode de frontière immergée pour différentes configurations et différentes valeurs de Pe	172
5.14	Nouvelle fonction Heaviside pour différentes valeurs du ratio $\frac{\Delta t_0}{\Delta t}$ (gauche) et effet lisseur pour un pas de temps oscillant (droite)	173
5.15	Position de la frontière par rapport au domaine discrétisé	176
5.16	Comparaison des méthodes de frontière immergée, ordre 1 (à gauche) ordre 2 (à droite)	177
5.17	Reconstruction des points aux frontières (●) à partir des points de calcul (●)	177
5.18	■ : 1 ^{ère} couche de cellules fluide, ■ : 1 ^{ère} couche de cellules solide	178
5.19	Valeurs des coefficients de pondération pour la direction horizontale (gauche) et verticale (droite).	179

5.20	Lissage de l'imposition du terme au source au voisinage du corps pour une méthode d'ordre 2	180
5.21	Écoulement autour d'un cylindre à $Re = 40$: comparaison des champs de pressions pour les formulations incompressible (gauche) et faiblement-compressible (droite) pour les différentes méthodes de frontière immergée à $\frac{D}{\Delta x} = 17$.	181
5.22	Écoulement autour d'un cylindre à $Re = 40$: comparaison des profils de pression (haut) et de vitesse horizontale (bas) sur l'axe horizontal médian en incompressible (gauche) et faiblement-compressible (droite), $\frac{D}{\Delta x} = 17$.	182
5.23	Écoulement autour d'un cylindre à $Re = 40$: incrément de pression adimensionnalisé en régime stationnaire pour la méthode d'ordre 1 lissé (gauche) et la méthode d'ordre 2 (droite)	183
5.24	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de vorticité à $\frac{tU_0}{c} = 100$. Comparaison des formulations incompressible (gauche) et faiblement-compressible (droite), $\frac{c}{\Delta x} = 37$, DVH (bas)	185
5.25	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de vorticité à $\frac{tU_0}{c} = 100$. Comparaison des formulations incompressible (gauche) et faiblement-compressible (droite), $\frac{c}{\Delta x} = 150$, DVH (bas)	186
5.26	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de pression à $\frac{tU_0}{c} = 100$. Comparaison des formulations incompressible (gauche) et faiblement-compressible (droite), $\frac{c}{\Delta x} = 150$	187
6.1	Illustration du principe de la fast marching method : points <i>Far away</i> (◐), points <i>Narrow band</i> (◑), points <i>Frozen points</i> (◒)	191
6.2	Rectangles et cônes définis par extrusion de la géométrie en 2D, extrait de [Mauch, 2000]	192
6.3	(a) Polyèdres des faces triangulaires, (b) polyèdres des arêtes, (c) polyèdre d'une arête, (d) polyèdres des sommets, extrait de [Mauch, 2000]	192
6.4	Principe de la rasterisation (ou scan conversion en anglais), extrait de [Mauch, 2000]	193
6.5	Configurations possibles de la projection d'un point sur la surface d'un triangle pour calculer le signe	194
6.6	Angle d'incidence de chaque sommet dans la méthode <i>Angle Weighted Pseudo-normal</i> , extrait de [Bærentzen et Aanaes, 2005]	195
6.7	Exemple de triangles qui s'intersectent (ou <i>self-intersected triangles</i> en anglais), extrait de [Yamakawa et Shimada, 2009]	195
6.8	Principe du calcul de l'indice <i>generalized winding number</i> en 2D et 3D, extrait de [Jacobson et al., 2013]	196
6.9	Test intérieur/extérieur par comptage des intersections de rayon	197
6.10	Partionnement par volume englobant avec des AABB	197
6.11	Partionnement par grille cartésienne	198
6.12	Partionnement par octree	198
6.13	Partionnement par kd-tree	199
6.14	Octree généré à partir du maillage surfacique d'une voiture	201
6.15	Maillage surfacique d'une sphère	202
6.16	Temps CPU de la construction de l'octree	203
6.17	Temps CPU du calcul de distance par cellule par triangle en fonction du nombre de cellules (à gauche) et en fonction du nombre de triangles (à droite)	204
6.18	Distance signée autour d'une voiture pour différents maillages.	205
6.19	Changement de repère pour le calcul de distance à des corps mobiles	206
6.20	Maillage surfacique de l'hélice KCS	207
6.21	Plan de coupe du maillage volumique	208
6.22	Iso-surface de la fonction distance pour une valeur de 0.	209

6.23	Domaine de calcul et conditions aux limites	210
6.24	Isocontour 1000 du critère Q et lignes de courant (en haut à gauche), champ de vitesse horizontale dans des plans de coupe en aval de l'hélice (en haut à droite), champ de vitesse dans un plan horizontal (milieu), champ de vitesse transversale (en bas).	211
6.25	Vues du maillage Star-CCM+. Maillage surfacique (à gauche) et maillage volumique avec couche limite (à droite)	212
6.26	Vues de $y+$ pour le maillage WCCH	212
6.27	Champ de pression sur l'intrados et l'extrados des pales. Résultats WCCH (haut, hélice complète) résultats Star-CCM+ (bas, une seule pale)	213
6.28	Courbe de performance de l'hélice issue des expériences et point de fonctionnement simulé .	214
A.1	Comparaison des champs de pression à $t=3s$ et le taux de décroissance d'énergie cinétique pour un maillage 128×128	238
A.2	Profils de vitesse verticale et de pression au bout de 10 passages dans la boîte périodique de 128×128 cellules.	239
B.1	Définition d'une matrice parallèle en sous-matrices locales ■ et sous-matrices extra-diagonales ■	243
C.1	Gain CPU incompressible/faiblement-compressible en fonction de l'accélération de l'algorithme faiblement-compressible à maillage constant.	247
C.2	Gain CPU incompressible/faiblement-compressible en fonction de l'accélération potentielle de l'algorithme faiblement-compressible à précision constante.	248
D.1	Algorithme PIMPLE d'un pas de temps	250
D.2	Convergence des résidus pour un maillage 192×192	251
D.3	Champ de pression après un passage du tourbillon pour un maillage 192×192	252
D.4	Comparaison des convergences en vitesse et en pression de différents schémas, PIMPLE, projection de Chorin, faiblement-compressible, etc.	253
D.5	Convergence des résidus du schéma PIMPLE (tirets : sans <i>deferred correction</i> , trait plein : avec <i>deferred correction</i>)	254
D.6	Comparaison des champs de vitesse de la cavité à $Re = 1000$ pour un maillage 64×64 . . .	255
D.7	Comparaison des champs de pression de la cavité à $Re = 1000$ pour un maillage 64×64 . .	256
D.8	Comparaison des profils verticaux en $x = L/2$ de la vitesse horizontale de la cavité à $Re = 1000$ pour un maillage 64×64	257
E.1	Champ de pression servant à la validation du calcul des dérivées spatiales	260
E.2	Emplacement des inconnues à une interface de raffinement de maillage.	261
E.3	Convergence du calcul des dérivées de pression pour la méthode 1	262
E.4	Convergence du calcul des dérivées de pression pour la méthode 2	263
E.5	Convergence du calcul des dérivées de pression pour la méthode 3	263
E.6	Convergence du calcul de dérivée seconde $\partial^2 p / \partial x \partial y$	264
E.7	Convergence du calcul de dérivée seconde $\partial^2 p / \partial x \partial y$	264
E.8	Convergence du critère sur un champ de pression de l'écoulement de Taylor-Green	265
E.9	Maillage adapté selon le hessien de pression pour les tourbillons de Taylor-Green	266
E.10	Champ de pression et profil de vitesse pour l'écoulement de cavité entraînée à $Re = 1000$.	267

Liste des tableaux

4.1	Caractéristiques des codes du benchmark de la convection de tourbillon	107
4.2	Localisation et intensité du tourbillon central principal (ψ_{min}) et du tourbillon secondaire (ψ_{max}) dans le coin inférieur droit.	113
4.3	Convergence en maillage FVCA : 2D Bercovier–Engelman – guard-cells	116
4.4	Convergence en maillage FVCA : 2D Bercovier–Engelman – stretching	116
4.5	Convergence en maillage FVCA : Vortex 2D - guard-cells	118
4.6	Convergence en maillage FVCA : Vortex 2D - stretching	119
4.7	Convergence en maillage FVCA : Tourbillons instationnaires - $\nu = 10^{-1}$ - guard-cells . . .	120
4.8	Convergence en maillage FVCA : Tourbillons instationnaires - $\nu = 10^{-1}$ - stretching	120
4.9	Convergence en maillage FVCA : Tourbillons instationnaires - $\nu = 10^{-2}$ - guard-cells . . .	120
4.10	Convergence en maillage FVCA : Tourbillons instationnaires - $\nu = 10^{-2}$ - stretching	121
4.11	Comparaison des caractéristiques de l'écoulement autour du cylindre à $Re = 40$ par rapport à des données expérimentales et numériques	123
4.12	Coefficient de traînée et de portance et nombre de Strouhal pour l'écoulement autour d'un cylindre à $Re = 100$	125
4.13	Tourbillons de Taylor-Green. Viscosités numériques en $m^2.s^{-1}$	126
4.14	Écoulement de Taylor-Couette. Convergence spatiale en maillage pour la formulation incompressible	131
4.15	Écoulement de Taylor-Couette. Convergence spatiale en maillage pour la formulation faiblement-compressible	131
4.16	Écoulement 2D autour d'un profil NACA à $Re = 1400$. Paramètres et temps de simulations .	136
5.1	Écoulement autour d'un cylindre à $Re = 40$: comparaison des coefficients de traînée pour les différents ordres de frontière immergée	183
6.1	Caractéristiques des octrees générés	203

Introduction générale

Motivations

La mécanique des fluides numérique (MFN ou CFD pour Computational Fluid Dynamics en anglais) a grandi avec l'essor des moyens informatiques au cours de la seconde moitié du XX^{ème} siècle. Elle s'intéresse à la résolution numérique des équations aux dérivées partielles qui régissent les fluides. Les premières approches modernes de résolution des équations d'Euler, qui par là donnèrent naissance à la CFD, remontent à la Seconde Guerre mondiale. Elles sont attribuées à John Von Neumann pour ses travaux de recherche sur les effets aérodynamiques des explosions et l'estimation des ondes de chocs générées par les armes nucléaires. L'avènement des premiers calculateurs à la fin des années 1950 et la définition de nouveaux outils mathématiques en ont fait une nouvelle discipline devenue aujourd'hui une approche incontournable dans la recherche et l'industrie. De nos jours, les codes CFD se sont démocratisés et des outils commerciaux, comme STAR-CCM+ et Fluent, ou open source comme OpenFOAM, sont quotidiennement utilisés par les bureaux d'études. Leurs champs d'applications sont vastes : partout où il est question d'un fluide, dans les domaines de l'énergie, de la santé, des procédés, etc. Ces outils ont révolutionné la conception des produits manufacturés (avions, voitures, composants électroniques, bâtiments, etc.) et continuent de repousser sans cesse les frontières de la simulation numérique. Le domaine des transports, notamment les industries aéronautique, automobile et navale, profite largement de l'engouement pour les codes CFD. D'une démarche de conception principalement empirique et construite sur des modèles simplifiés, ces industries sont passées à une approche combinée mêlant théorie, expérimentation et simulation numérique. Loin d'entraîner la disparition de l'expérimentation au réel, la mécanique des fluides numérique s'est intégrée aux approches de conception historiques pour accroître le rythme de développement des produits, garantir leur fiabilité, optimiser leur coût, assurer la sécurité des usagers mais aussi augmenter le niveau confort.

En hydrodynamique navale, les premiers codes de calcul qui ont émergé dans les années 1970 étaient basés sur une hypothèse de fluide parfait irrotationnel avec une résolution par la méthode des singularités. Ces premiers solveurs permettaient d'estimer la résistance de vague d'un navire avançant à vitesse constante. Au cours des années 1980 et 1990, les méthodes de résolution des équations de Navier-Stokes de type volumes finis ont connu un élan tout particulier du fait de leur robustesse et leur champ d'application étendu par rapport aux codes potentiels, avec une résolution reposant sur des hypothèses moins restrictives sur le fluide. Il devint alors possible de prendre en compte les phénomènes de turbulence (méthodes RANS, Reynolds Average Navier-Stokes) et l'interface complexe de la surface libre (méthode Volume of Fluid ou Level-Set). Les codes volumes finis ont bénéficié de dizaines d'années de recherche, de développement et de validation. Ce sont aujourd'hui des codes aboutis et éprouvés pour traiter une majorité des problèmes rencontrés dans l'industrie navale et offshore comme l'optimisation d'une carène de navire pour réduire sa consommation en carburant (Fig. 1), le comportement des structures dans la houle, les impacts de vague, etc.

Le processus de mise en place d'une simulation suit toujours le même ordre. Une représentation numérique de l'objet autour duquel ou dans lequel l'écoulement est étudié est générée à partir d'un logiciel de CAO (Conception Assistée par Ordinateur). Un logiciel de maillage permet ensuite de créer une grille

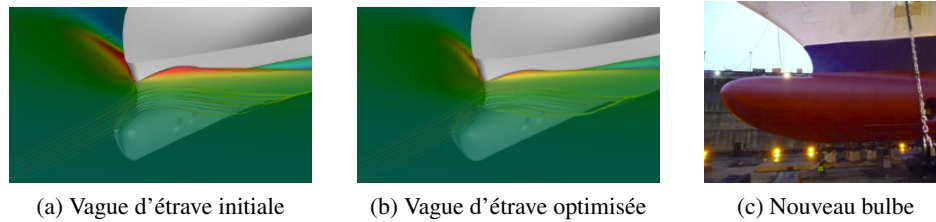


FIGURE 1 – Exemple d’optimisation de bulbe d’étrave pour la réduction de la consommation de carburant. *Crédits images Tecnitas-HydrOcean.*

de calcul autour de la géométrie et sur laquelle les équations sont discrétisées en vue de leur résolution. L’outil de maillage peut être intégré dans une plateforme logicielle complète comme STAR-CCM+ ou être un logiciel dédié comme les maillages HEXPRESSTM ou ANSYS ICEM CFD. La génération du maillage est exigeante et sa qualité est primordiale pour la précision des résultats. Même si la plupart des maillages sont maintenant bien automatisés et pourvus de fonctions d’optimisation pour générer le maillage de meilleure qualité possible, cette tâche reste fastidieuse et demande plusieurs jours (voire semaines) de temps ingénieur avant d’aboutir à un maillage optimal (Fig. 2).

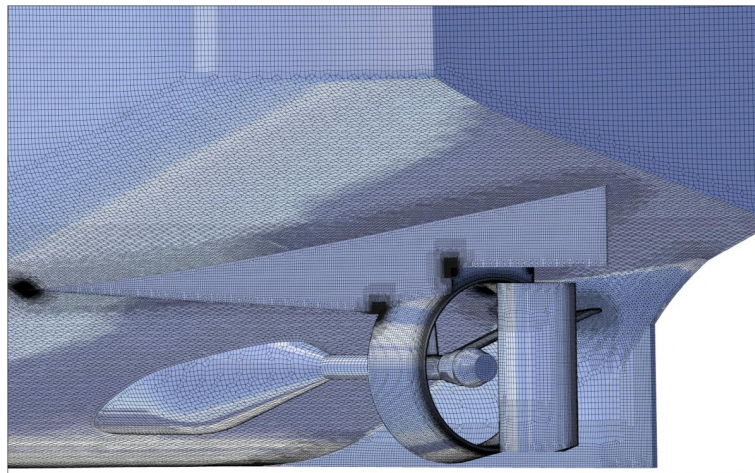


FIGURE 2 – Exemple de maillage complexe généré avec HEXPRESSTM. *Source : <http://www.parsmakina.com/en/product/hexpres>. Crédits IHC Dredger International et ECN/CNRS.*

En outre, si l’outil CFD ne dispose pas de fonctionnalités de raffinement et de déraffinement automatique pour modifier dynamiquement le maillage au cours de la simulation, il est alors essentiel de connaître a priori l’écoulement afin de raffiner le maillage dans les zones d’intérêt de l’écoulement qui nécessiteront une précision accrue comme par exemple les couches limites, les zones de décollement d’écoulements, etc.

Une fois le maillage généré, le calcul peut être lancé en renseignant des paramètres physiques de l’écoulement et des paramètres numériques liés au solveur, avant d’être post-traité pour en extraire les résultats. Puisque les problèmes résolus comportent plusieurs millions d’inconnues, les codes sont souvent développés pour s’exécuter sur des ressources de calcul parallèles afin de réduire les temps de restitution des résultats. Néanmoins, les méthodes numériques classiquement utilisées en hydrodynamique dans les codes standards ont une scalabilité relativement faible. Autrement dit elles ne sont pas toujours adaptées pour une exécution sur un grand nombre de nœuds de calcul (> 1000) simultanément et de façon efficace. Augmenter les ressources de calcul d’un certain facteur n’entraîne pas systématiquement une diminution du temps de restitution par ce facteur. Ce point est une difficulté de taille pour traiter des problèmes de complexité

croissante sur les infrastructures massives de calcul parallèle qui se démocratisent. Enfin les schémas numériques implémentés dans les outils CFD actuels sont en général d'ordre 2. Cela signifie qu'un maillage avec un pas d'espace deux fois plus petit que celui du maillage initial verra l'erreur de sa solution divisée par quatre (2^2). Une convergence en maillage permet de s'assurer qu'à partir d'un certain niveau de raffinement croissant, les variations de résultat d'un maillage à l'autre ne sont plus significatives. Si l'ordre 2 de convergence des codes actuels peut être satisfaisant pour la majorité des applications industrielles navales, il peut se révéler insuffisant pour résoudre des écoulements complexes, hautement turbulents, instationnaires et multi-échelles.

Pour compléter le spectre de résolution des outils CFD actuels, il est pertinent de développer un solveur s'appuyant à la fois sur des schémas de discrétisation spatio-temporels d'ordre élevé et sur une prise en compte aisée des corps ne nécessitant pas (ou très peu) d'action humaine de maillage. Par ailleurs, ce solveur doit pouvoir également profiter pleinement des nouvelles architectures de calcul parallèle afin de restituer des résultats avec un coût CPU acceptable (Fig. 3).

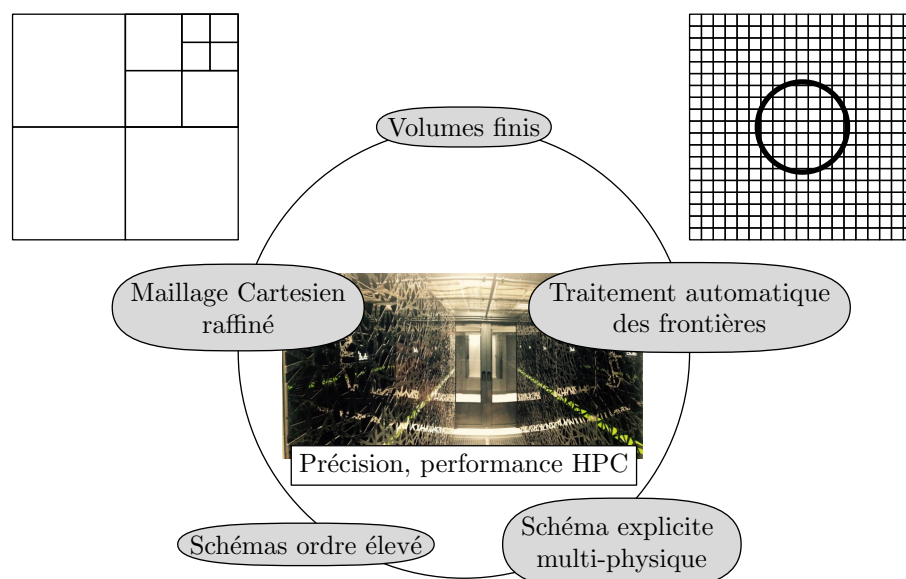


FIGURE 3 – Caractéristiques du solveur développé

Contexte

En 2008, le Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement Atmosphérique (LHEEA) de l'École Centrale de Nantes a initié le développement d'un tel solveur se différenciant des approches CFD conventionnelles de l'hydrodynamique navale et répondant aux exigences listées précédemment. L'idée était alors de profiter de l'expertise acquise par le laboratoire dans le développement de codes d'ordre 2 à surface libre et de la méthode SPH (Smoothed Particles Hydrodynamics) [Monaghan, 1992, Le Touzé et al., 2008] pour proposer un code volumes finis original fondé sur une approche faiblement-compressible [Leroy, 2012]. Il a été donné au solveur le nom de la méthode : WCCH pour Weakly-Compressible Cartesian Hydrodynamics. En se basant sur une approche faiblement-compressible, les équations sont intégrées en temps de façon explicite. Le maillage est une grille cartésienne multi-niveaux (Fig. 4). Ces deux éléments simplifient le recours à des schémas numériques d'ordre élevé et permettent une implémentation scalable de la méthode.

HydrOcean, société d'ingénierie spécialisée dans la simulation numérique en hydrodynamique pour le domaine maritime, a accompagné le développement de cette méthode dès 2011 avec une première thèse

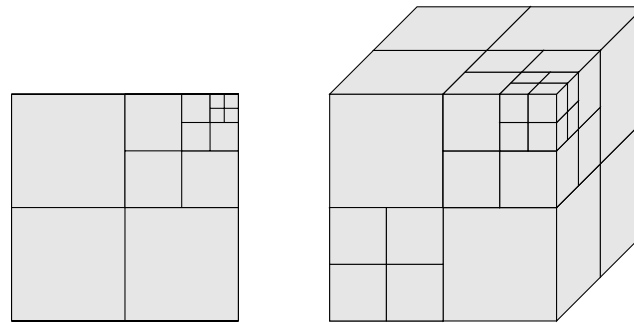
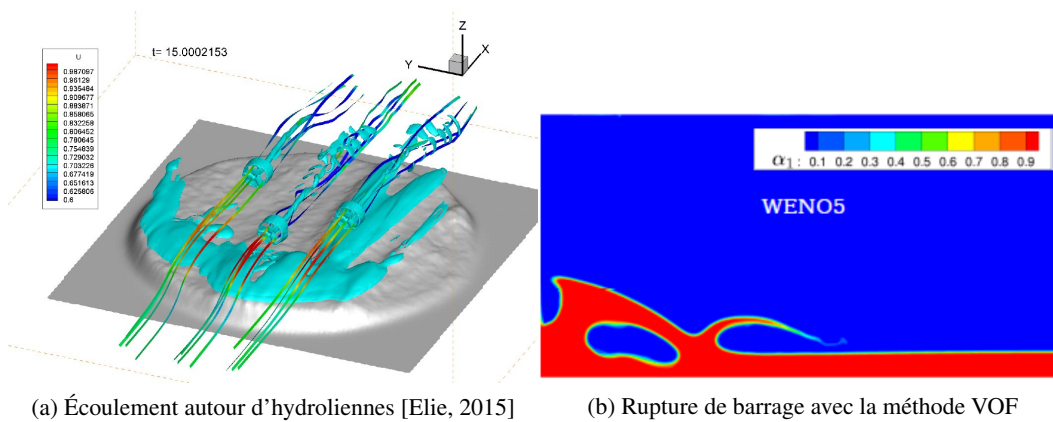


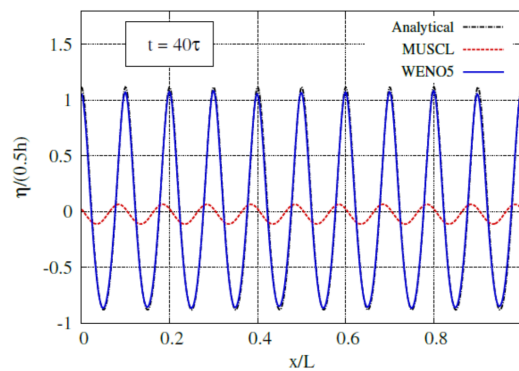
FIGURE 4 – Exemples de grilles cartésiennes multi-niveaux sur lesquelles s’appuie WCCH

CIFRE. Ce travail de recherche a permis d’aboutir à un solveur d’ordre élevé capable de calculer avec précision l’écoulement autour de corps simples en régime laminaire instationnaire à l’aide d’un raffinement statique du maillage [Bigay, 2015]. Deux autres acteurs ont contribué au développement du solveur : Alstom et l’Institut de Recherche Technologique Jules Verne. Les travaux qu’ils ont financé au laboratoire ont notamment permis d’appliquer la méthode aux sillages d’hydroliennes [Elie, 2015] ou encore d’introduire la prise en compte de la surface libre via une méthode de type Level-Set [Bardin, 2015] ou VOF. Dans le premier cas, l’objectif visé est la simulation de fermes d’hydroliennes en vue de leur optimisation. Les premiers résultats sont encourageants et confirment le potentiel de la méthode (Fig. 5).



(a) Écoulement autour d’hydroliennes [Elie, 2015]

(b) Rupture de barrage avec la méthode VOF



(c) Propagation de houle périodique par la méthode VOF, comparaison WENO5 et MUSCL

FIGURE 5 – Exemples de simulations réalisées à l’aide du code WCCH

Aujourd'hui le développement se poursuit avec cette thèse CIFRE conduite par le LHEEA et l'entreprise Nextflow Software, spin-off indépendante d'HydrOcean après le rachat de celle-ci par le groupe Bureau Veritas. L'activité de Nextflow Software se concentre sur l'édition de logiciels de simulation de mécanique des fluides comme le code *SPH-flow*.

En l'état actuel, même si le solveur dispose de caractéristiques intéressantes, de nombreux défis scientifiques restent à relever pour qu'il atteigne l'applicabilité industrielle. À cet effet, ce travail de thèse s'est concentré principalement sur la réduction du temps de restitution des calculs, vue sous différents angles.

En parallèle de ce travail, de nombreux développements ont été conduits sur le code par d'autres acteurs : méthode VOF pour renforcer la robustesse de la prise en compte de la surface libre, modèle de turbulence de type Spalart-Allmaras, méthode SWENSE pour la propagation de houle, raffinement dynamique sur la base de critères physiques, portage du code sur GPU, etc.

Objectifs

La présente thèse vise à développer, améliorer et réduire les temps de restitution de la méthode WCCH afin de répondre aux problématiques industrielles énoncées au début de cette introduction. Notamment les écoulements envisagés à long terme pour la méthode sont turbulents et fortement instationnaires. Les simulations de ces écoulements s'étalent sur de longues durées physiques et exigent une bonne précision au voisinage du corps où les instationnarités d'écoulement sont générées. Néanmoins l'un des points pénalisants pour atteindre les exigences fixées, et inhérent à l'approche faiblement-compressible explicite, est la limitation du pas de temps par la vitesse du son, même en utilisant une vitesse du son fictive permettant de se placer à la limite de l'hypothèse de faible compressibilité. De plus, pour des écoulements à bas nombre de Mach (comme ceux qui sont visés) une approche incompressible est classiquement privilégiée par les méthodes conventionnelles. Cette approche permet de s'affranchir de la contrainte de la vitesse du son. Il est alors apparu nécessaire d'implémenter un schéma incompressible afin de quantifier précisément les atouts et les désavantages de l'approche faiblement-compressible. L'idée n'est pas d'abandonner l'approche faiblement-compressible au profit d'une formulation incompressible mais d'enrichir le code pour justifier ou infirmer l'intérêt du faiblement-compressible sur certains aspects numériques (diffusion des schémas, compatibilité avec l'immersion de corps solides dans la grille cartésienne, capacité à résoudre des écoulements très instationnaires, etc.) et HPC (parallélisation aisée, scalabilité sur des architectures massivement parallèles impliquant un nombre de processeurs supérieur à 1000) afin de mettre en lumière de façon quantitative les avantages relatifs de chaque formulation.

Dans ce manuscrit, une première partie est consacrée à une présentation partielle du code WCCH afin de présenter l'outil dans lequel les méthodes numériques ont été implémentées mais aussi pour mettre en avant les limitations de l'approche faiblement-compressible.

Vient ensuite la présentation de la formulation incompressible qui a été choisie. L'extension à des maillages raffinés localement est également décrite.

La formulation incompressible proposée est ensuite validée numériquement sur des cas tests académiques et comparée à l'approche originelle sur la base de critères numériques et d'efficacité computationnelle.

Ensuite, la méthode de frontière immergée originelle est analysée en détail pour suggérer des améliorations et renforcer la robustesse de cette méthode. Une méthode d'ordre plus élevé est également esquissée. Les difficultés mathématiques inhérentes à cette problématique sont discutées.

La dernière partie de ce manuscrit traite de l'immersion efficace et robuste de géométries complexes dans le code. La technique mise en place est évaluée sur la simulation numérique directe de l'écoulement autour d'un propulseur de navire en rotation en eau libre. Elle laisse entrevoir les applications d'hydrodynamique navale qui pourront être envisagées avec le solveur WCCH.

Enfin il est à noter que cette thèse n'aborde pas tous les aspects de la méthode WCCH. En particulier, les cas étudiés sont limités à des écoulements monophasiques et laminaires.

Chapitre 1

Présentation de la méthode WCCH

Nous présentons dans ce chapitre les équations physiques employées pour modéliser les écoulements hydrodynamiques ainsi que leurs hypothèses associées. La résolution numérique de ces équations dans le code WCCH est ensuite décrite en incluant une brève introduction à la méthode de frontière immergée utilisée. Enfin les limitations de l'approche faiblement-compressible sont discutées.

Ce chapitre vise, d'une part, à fixer l'environnement numérique et algorithmique dans lequel les travaux de thèse ont été conduits et d'autre part à introduire les motivations des différents axes de recherche pour l'amélioration de la méthode WCCH : formulation incompressible, montée en ordre de la méthode de frontière immergée, immersion rapide de géométries complexes.

1.1	Modèle mathématique	24
1.1.1	Équations de Navier-Stokes en formulation compressible	24
1.1.2	Équations de Navier-Stokes en formulation incompressible	24
1.1.3	Stratégies de résolution des écoulements à bas nombre de Mach : approche faiblement-compressible ou purement incompressible	25
1.1.4	Conditions aux limites incompressibles	25
1.2	Description du solveur volumes finis de WCCH	27
1.2.1	Formulation volumes finis	27
1.2.2	Calculs des flux	28
1.2.3	Schéma d'avance en temps	32
1.2.4	Méthode de frontière immergée	33
1.2.5	Maillage et structure de données	35
1.2.6	Parallélisation	40
1.2.7	Exemple d'un écoulement autour d'un cylindre simulé avec WCCH	41
1.3	Problématiques actuelles du code WCCH	41
1.3.1	Limitations de l'approche faiblement-compressible pour la simulation à bas nombre de Mach	41
1.3.2	Précision et robustesse de la méthode de frontière immergée	44
1.3.3	Immersion de géométries complexes	45
1.4	Conclusion	45

Remarque : pour alléger la notation dans les équations, les vecteurs sont en gras et ∇ désigne l'opérateur gradient alors que $\nabla \cdot$ correspond à l'opérateur divergence.

1.1 Modèle mathématique

1.1.1 Équations de Navier-Stokes en formulation compressible

Les équations fondamentales de la mécanique des fluides sont issues des lois de conservations de la masse, de la quantité de mouvement et de l'énergie. Dans le cas d'un écoulement compressible isentropique, i.e. adiabatique réversible, les équations complètes s'écrivent sous forme locale de la façon suivante :

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \\ \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \bar{\bar{\tau}} + \rho \mathbf{f} \\ \frac{\partial \rho e}{\partial t} + \nabla \cdot [(\rho e + p)\mathbf{u}] = \nabla \cdot (\bar{\bar{\tau}} \cdot \mathbf{u}) \end{cases} \quad (1.1)$$

Dans ces équations \otimes désigne le produit tensoriel, ρ la masse volumique, $\mathbf{u} = (u, v, w)^t$ le vecteur des vitesses eulériennes, p la pression, \mathbf{f} un terme source comme la gravité, e l'énergie totale par unité de masse et $\bar{\bar{\tau}}$ le tenseur des contraintes visqueuses. Ce dernier prend la forme suivante pour les fluides newtoniens $\bar{\bar{\tau}} = 2\mu[\frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^t}{2} - \frac{1}{3}(\nabla \cdot \mathbf{u})\bar{\bar{I}}]$, avec μ le coefficient de viscosité dynamique et $\bar{\bar{I}}$ le tenseur identité. Le système d'équations (1.1) correspond alors aux équations de Navier-Stokes. On remarque par ailleurs qu'il n'est pas fermé avec 6 inconnues ρ , \mathbf{u} , e et p pour seulement 5 équations (en 3D). Pour fermer ce système une équation d'état reliant les variables d'état p et ρ ainsi que e lui est adjointe. Dans le cadre des problèmes physiques de l'hydrodynamique navale, l'équation d'état barotropique de Tait peut être employée :

$$p = \frac{\rho_0 c_0^2}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \quad (1.2)$$

dans laquelle γ , c_0 , ρ_0 représentent respectivement le coefficient adiabatique, la masse volumique nominale du fluide et la vitesse du son nominale. Cette loi d'état est adaptée aux écoulements dont les variations de pression dépendent uniquement des variations de densité. En conséquence les isobares sont parallèles aux isothermes. Elle convient pour la majorité des applications rencontrées en hydrodynamique navale. L'avantage de cette loi d'état est qu'elle permet de découpler l'équation portant sur l'énergie du reste du système. Celle-ci n'est donc pas résolue dans ce cas. γ permet d'ajuster la raideur de l'équation et donc la compressibilité du fluide. Ses valeurs couramment prises sont de 1.4 pour l'air et 7 pour l'eau.

On note qu'avec cette loi d'état il est possible d'avancer en temps le système (1.1) de façon totalement explicite, ce qui ne sera plus le cas en incompressible, comme cela va être montré dans la suite.

1.1.2 Équations de Navier-Stokes en formulation incompressible

Dans le cas où les variations de densité sont négligeables dans un écoulement monophasique, les équations précédentes (1.1) deviennent :

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{\nabla p}{\rho} + \frac{1}{\rho} \nabla \cdot \bar{\bar{\tau}} + \mathbf{f} \end{cases} \quad (1.3)$$

Le principal changement avec (1.1) est que la densité n'apparaît plus dans l'équation de conservation de la masse. De plus (1.3) n'a pas besoin d'être fermé par une équation d'état.

1.1.3 Stratégies de résolution des écoulements à bas nombre de Mach : approche faiblement-compressible ou purement incompressible

Tout d'abord il convient de bien distinguer les qualificatifs *compressible/incompressible* appliqués à un fluide de ceux qui définissent un écoulement. Tout fluide réel est compressible et sa compressibilité est une propriété intrinsèque qui donne la variation de volume occupé par une masse donnée suivant la pression extérieure. Certains fluides sont plus compressibles (gaz) que d'autres (liquides) mais selon la nature des écoulements étudiés, cette propriété se manifeste plus ou moins et l'hypothèse d'incompressibilité peut être posée ou non. Pour des écoulements adiabatiques, cette distinction repose sur le nombre de Mach Ma défini comme le rapport de la vitesse maximale de l'écoulement et de la vitesse du son $Ma = \frac{u_{max}}{c}$. Un écoulement pourra être qualifié d'incompressible si le nombre de Mach est suffisamment petit. $Ma < 0.2$ est un critère d'incompressibilité communément admis.

En hydrodynamique, les écoulements rencontrés sont très souvent considérés comme incompressibles. Les variations de densité du fluide sont alors négligeables (hors écoulement multiphasique). La majorité des codes CFD d'hydrodynamique résolvent donc le système d'équations (1.3) plutôt que (1.1), notamment parce qu'il présente certains avantages pour la résolution numérique des équations de Navier-Stokes. Toutefois ces équations ne restent qu'une approximation de la réalité qui ne convient que dans certaines situations. En faisant l'hypothèse que les variations de densité sont négligeables, on considère que les ondes acoustiques se propagent à vitesse infinie (Fig. 1.1). La conséquence directe est que l'information acoustique se propage instantanément. Ceci n'a bien entendu pas de sens physique puisque l'information se déplace toujours à une vitesse finie en mécanique Newtonienne. Néanmoins elle offre une bonne représentation de la réalité dans le cadre des écoulements ciblés.

Une autre approche tout aussi recevable est d'utiliser les équations de la mécanique des fluides compressibles (1.1) pour résoudre un écoulement quasi-incompressible. Ce choix est consistant dans la mesure où un développement asymptotique des équations de Navier-Stokes compressibles permet de montrer que la solution compressible de (1.1) est la somme de la solution incompressible de (1.3), d'une contribution acoustique en $O(Ma)$ et de termes additionnels en $O(Ma^2)$ (cf. cours de G. Capdeville [Capdeville, 2006]). Le système des équations incompressibles (1.3) est donc une limite singulière des équations compressibles (1.1) lorsque $Ma \rightarrow 0$.

L'approche faiblement-compressible retenue dans WCCH consiste à fixer arbitrairement la vitesse du son de sorte que $Ma \leq 0.1$. Ce choix est fondé car les perturbations acoustiques resteront infimes et n'impacteront pas la partie incompressible de l'écoulement (Fig. 1.1). Le développement des équations en série de perturbations acoustiques montre en effet que $p = p_\infty + O(Ma^2)$ [Capdeville, 2006], autrement dit les fluctuations de pression (ou de densité) sont proportionnelles au nombre de Mach au carré.

L'approche requiert donc l'estimation a priori des valeurs maximales de vitesse dans l'écoulement afin de pouvoir déterminer la valeur de vitesse du son permettant de respecter les plages de valeur de Ma pour lesquelles l'influence de la partie compressible des équations reste négligeable, i.e. $Ma < 0.1$ partout et tout le temps (Fig. 1.1). Il faut aussi vérifier a posteriori que les variations de densité, témoins des variations de pression, sont négligeables ($< 1\%$).

En définitive, l'hypothèse d'incompressibilité et celle de faible compressibilité sont deux représentations recevables d'un écoulement réel à bas nombre de Mach. WCCH repose sur la deuxième hypothèse. Si à l'échelle continue, aucune des deux hypothèses n'est plus admissible que l'autre, à l'échelle discrète, elles ont des conséquences directes sur le choix des schémas numériques qui peuvent être employés.

1.1.4 Conditions aux limites incompressibles

Les équations de Navier-Stokes ne peuvent être résolues dans un domaine continu infini, l'étude de l'écoulement est donc restreinte à un domaine borné. Aux frontières du domaine, des contraintes spécifiques

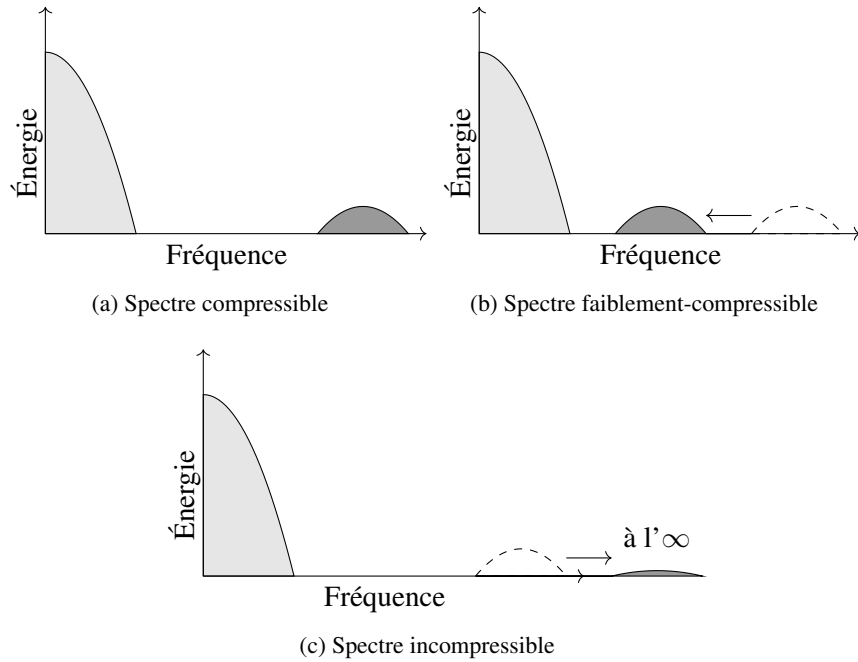


FIGURE 1.1 – Différences entre les spectres d’un écoulement réel à bas Mach pour l’approche compressible, faiblement-compressible et incompressible. ■ : partie purement incompressible, ■ : partie acoustique.

sont nécessaires afin de fermer le système d’équations. Le traitement des conditions aux limites est un point sensible dans la mesure où leur mauvaise prise en compte peut affecter l’ensemble de l’écoulement. En pratique, pour les écoulements externes, on veille à toujours placer les conditions aux limites loin des zones d’intérêt pour limiter leur influence.

- La condition la plus couramment rencontrée au niveau des frontières solides est la condition d’adhérence qui se traduit par :

$$\mathbf{u} = \mathbf{u}_{paroi} \quad (1.4)$$

Aucune information sur la pression n’est disponible même si d’un point de vue numérique il peut être nécessaire d’avoir recours à une condition aux limites sur cette grandeur. Pour des écoulements à grand nombre de Reynolds, il est possible de se placer dans le cadre de la théorie de Prandtl des couches limites minces cisailées qui conduit à

$$\nabla p \cdot \mathbf{n}_{paroi} = 0 \quad (1.5)$$

où \mathbf{n}_{paroi} désigne le vecteur normal à la paroi.

- En cas de non-adhérence à la paroi, on parle de condition de glissement :

$$\mathbf{u} \cdot \mathbf{n}_{paroi} = 0 \quad (1.6)$$

Cette condition traduit uniquement la non-pénétration du fluide à travers la paroi.

- Une condition entrante revient à imposer un profil de vitesse sur la frontière du domaine :

$$\mathbf{u} \cdot \mathbf{n} = u_{entrant} \quad (1.7)$$

- Pour une condition sortante on impose classiquement la pression :

$$p = p_{sortie} \text{ et } \mathbf{n} \cdot \nabla(\mathbf{u} \cdot \mathbf{n}) = 0 \quad (1.8)$$

1.2 Description du solveur volumes finis de WCCH

Dans cette section, nous présentons la méthode WCCH dans ses grandes lignes. Les schémas décrits sont représentatifs des fonctionnalités présentes dans le code au début de ce travail de thèse.

1.2.1 Formulation volumes finis

Le système d'équations (1.1) peut s'écrire de façon plus compacte en faisant apparaître clairement les flux de matière et de quantité de mouvement :

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \Psi_c(\mathbf{W}) = \nabla \cdot \Psi_v(\mathbf{W}) + \mathbf{S} \quad (1.9)$$

avec le vecteur des variables conservatives $\mathbf{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \end{pmatrix}$,

la partie hyperbolique convective $\nabla \cdot \Psi_c(\mathbf{W}) = \underbrace{\begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \end{pmatrix}}_{\mathbf{F}_c(\mathbf{W})_{,x},x} + \underbrace{\begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \end{pmatrix}}_{\mathbf{G}_c(\mathbf{W})_{,y},y} + \underbrace{\begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \end{pmatrix}}_{\mathbf{H}_c(\mathbf{W})_{,z},z},$

la partie elliptique visqueuse $\nabla \cdot \Psi_v(\mathbf{W}) = \underbrace{\begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \end{pmatrix}}_{\mathbf{F}_v(\mathbf{W})_{,x},x} + \underbrace{\begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \end{pmatrix}}_{\mathbf{G}_v(\mathbf{W})_{,y},y} + \underbrace{\begin{pmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \end{pmatrix}}_{\mathbf{H}_v(\mathbf{W})_{,z},z}$

et le terme source $\mathbf{S} = \begin{pmatrix} 0 \\ \rho \mathbf{f} \end{pmatrix}$.

\mathbf{F} , \mathbf{G} et \mathbf{H} désignent le vecteur des flux physiques dans les directions x , y et z .

La méthode des volumes finis est utilisée pour discrétiser (1.9). L'ensemble du domaine de définition Ω est partitionné en un ensemble de volumes de contrôle, encore appelés cellules ou mailles, et notés $C_{i,j,k}$. Comme une topologie cartésienne est adoptée dans le code, ces volumes de contrôle sont des parallélépipèdes rectangles de centre (x_i, y_j, z_k) : $C_{i,j,k} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \times [z_{k-\frac{1}{2}}, z_{k+\frac{1}{2}}]$. Les équations sont discrétisées séparément en espace et en temps selon la méthode des lignes. En les intégrant sur $C_{i,j,k}$ et en appliquant le théorème de Green-Ostrogradski et de transport de Reynolds, on obtient l'équation suivante :

$$\frac{d}{dt} \iiint_{C_{i,j,k}} \mathbf{W} dV - \iint_{\partial C_{i,j,k}} \mathbf{W} (\mathbf{U}_{\text{maillage}} \cdot \mathbf{n}) dS + \iint_{\partial C_{i,j,k}} [\Psi_c(\mathbf{W}) - \Psi_v(\mathbf{W})] \cdot \mathbf{n} dS = \iiint_{C_{i,j,k}} \mathbf{S} dV \quad (1.10)$$

En considérant dès à présent que les volumes de contrôle seront fixes et indéformables, il vient $\mathbf{U}_{\text{maillage}} =$

0. Sous forme semi-discrète les équations s'écrivent finalement :

$$\begin{aligned}
 \frac{d\bar{W}_{i,j,k}}{dt} = & \frac{1}{\Delta x} (\mathbf{F}_{i-\frac{1}{2},j,k} - \mathbf{F}_{i+\frac{1}{2},j,k}) + \\
 & \frac{1}{\Delta y} (\mathbf{G}_{i,j-\frac{1}{2},k} - \mathbf{G}_{i,j+\frac{1}{2},k}) + \\
 & \frac{1}{\Delta z} (\mathbf{H}_{i,j,k-\frac{1}{2}} - \mathbf{H}_{i,j,k+\frac{1}{2}}) + \\
 & \frac{1}{\Delta x \Delta y \Delta z} \mathbf{S}_{i,j,k}
 \end{aligned} \tag{1.11}$$

$\bar{W}_{i,j,k} = \frac{1}{\Delta x \Delta y \Delta z} \iiint_{C_{i,j,k}} \mathbf{W}(x, y, z) dx dy dz$ représente la valeur moyenne de \mathbf{W} sur la cellule. Dans la suite nous abandonnons la notation $\bar{\cdot}$ et $\mathbf{W}_{i,j,k}$ désignera la valeur moyenne de la cellule bien qu'elle soit distincte de la valeur au centre. $\mathbf{F} = -\mathbf{F}_c + \mathbf{F}_v$ réunit les flux convectifs et visqueux pour les faces de $C_{i,j,k}$ normales à la direction x , $\mathbf{G} = -\mathbf{G}_c + \mathbf{G}_v$ pour la direction y , etc. De ce fait la méthode des volumes finis revient à lier les flux physiques à travers les faces des volumes de contrôle, aux valeurs moyennes des grandeurs conservatives sur ces mêmes volumes de contrôle. D'un point de vue pratique, les flux physiques sont sommés après leur évaluation sur chacune des faces des cellules puis (1.11) est intégrée en temps comme une équation différentielle ordinaire (EDO).

1.2.2 Calculs des flux

L'intégrale représentant le flux est évaluée numériquement au centre de la face selon la méthode du point médian (méthode de quadrature à un point de Gauss). L'erreur commise est alors d'ordre 2 mais offre un bon compromis coût/précision.

Résolution de la partie hyperbolique

En enlevant la contribution visqueuse des équations de Navier-Stokes compressible (1.1), nous obtenons les équations d'Euler qui constituent un système d'équations aux dérivées partielles (EDP) de nature hyperbolique. Autrement dit la matrice jacobienne \mathbf{A} des flux définie par :

$$\mathbf{A} = \frac{\partial \Psi_c \cdot \mathbf{n}}{\partial \mathbf{W}} \tag{1.12}$$

est diagonalisable avec des valeurs propres réelles. Les valeurs propres de la jacobienne \mathbf{A} valent :

$$\lambda_1 = \mathbf{u} \cdot \mathbf{n} - c, \lambda_2 = \mathbf{u} \cdot \mathbf{n}, \lambda_3 = \mathbf{u} \cdot \mathbf{n} + c \tag{1.13}$$

où c est la vitesse du son. Les équations hyperboliques décrivent des phénomènes de propagation (de nature acoustique dans le cas des équations d'Euler) et sont récurrentes en mécanique. De l'avènement de la CFD au milieu du XX^{ème} siècle jusqu'à nos jours, les schémas numériques de discrétisation des équations d'Euler ont été abondamment étudiés.

La propriété essentielle que doivent respecter ces schémas est la convergence de la solution numérique vers la solution exacte des équations d'Euler. L'étude mathématique de la convergence est difficilement accessible, surtout pour des équations non-linéaires. Cependant le théorème d'équivalence de Lax stipule que la stabilité et la consistance d'un schéma sont nécessaires et suffisants pour assurer sa convergence [Lax et Richtmyer, 1956]. Bien que restreint aux équations linéaires, ce principe n'en reste pas moins essentiel pour les schémas numériques d'équations non-linéaires comme celles de Navier-Stokes.

Un schéma est dit consistant si l'équation discrète qui résulte de l'application de ce schéma tend vers l'équation aux dérivées partielles exacte lorsque les pas de discrétisation en espace et en temps tendent vers

0 indépendamment. Un schéma itératif est dit stable si des perturbations de la solution numérique ne sont pas amplifiées au cours des itérations.

En pratique la stabilité est obtenue en s'assurant que les schémas de calcul des flux introduisent suffisamment de diffusion numérique. L'ouvrage de Hirsch [Hirsch, 2007] constitue une bonne introduction au lien entre stabilité et diffusion numérique des schémas. Dans la résolution des équations compressibles, cette dissipation numérique est primordiale au voisinage des chocs pour prévenir l'apparition d'oscillations non voulues. A l'inverse elle doit être minimum dans les régions régulières de l'écoulement. Différentes techniques de stabilisation sont utilisées : les méthodes à viscosité artificielle¹, où la dissipation est explicitement introduite ; les méthodes de décentrement, qui peuvent être basées sur la résolution d'un problème de Riemann² ; les méthodes flux-splitting ; etc..

La méthode de viscosité artificielle constitue la façon la plus évidente d'ajouter de la diffusion numérique. Le schéma JST proposé conjointement par Jameson, Schmidt et Turkel en 1981 [Jameson et al., 1981] est une illustration de ce type de schéma pour la résolution d'écoulements compressibles avec discontinuités en volumes finis. Les méthodes décentrées présentent quant à elles l'avantage d'être plus robustes et compatibles avec des schémas d'ordre élevé.

Le problème de Riemann est un problème numérique qui consiste à déterminer la solution d'une équation différentielle hyperbolique étant donné deux états (à gauche et à droite) constants par morceaux. Physiquement, cette description mathématique traduit l'expérience du tube à choc qui consiste à retirer une membrane séparant un gaz soumis à des conditions de température et de pression différentes de chaque côté. Dans le cas des volumes finis, ce problème apparaît naturellement quand il s'agit de calculer la valeur d'une grandeur (vitesse ou densité) à l'interface des deux cellules en vue de calculer le flux numérique convectif à la face considérée. La figure 1.2 illustre le problème de Riemann entre la cellule i et la cellule $i + 1$. Sa résolution consiste à déterminer le système d'ondes (chocs, détentes ou discontinuités de contact) qui se développe à partir de cette discontinuité. Les solveurs de Riemann permettent de résoudre un tel problème de façon exacte (comme le solveur de Godunov) ou de résoudre un problème approché (comme le solveur de Roe).

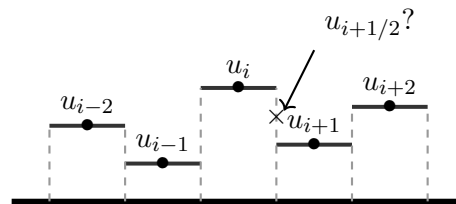


FIGURE 1.2 – Illustration du problème de Riemann à l'interface entre deux cellules

Dans le code WCCH, le décentrement n'est pas basé sur un solveur de Riemann mais sur une méthode à décomposition de flux proposée par [Ghidaglia et al., 1996]. Nous présentons ici la méthode en dimension 2.

Le flux numérique ϕ_c à une interface de normale \mathbf{n} qui sépare deux états à gauche \mathbf{W}_L et à droite \mathbf{W}_R peut s'écrire en une partie centrée et une partie décentrée :

$$\phi_c = \frac{\Psi_c(\mathbf{W}_L) + \Psi_c(\mathbf{W}_R)}{2} \cdot \mathbf{n} - \mathbf{S}(\mathbf{W}_L, \mathbf{W}_R) \frac{\Psi_c(\mathbf{W}_R) - \Psi_c(\mathbf{W}_L)}{2} \cdot \mathbf{n} \quad (1.14)$$

1. Le concept de viscosité artificielle est attribué à Richtmyer et Von Neumann [VonNeumann et Richtmyer, 1950] bien que la déclassification de certains documents du laboratoire de Los Alamos a montré que Richtmyer en avait déjà décrit le principe en 1948, seul.

2. La première résolution numérique d'un problème de Riemann a été présentée par Godunov à la fin des années 1950 [Godunov, 1959].

Pour la méthode à flux caractéristiques, la matrice S est égale au signe de la matrice jacobienne $S = \text{sign}(\mathbf{A})$.

Pour le flux F_c suivant x et le flux G_c suivant y , les matrices jacobiennes s'écrivent respectivement :

$$\mathbf{A}_{F_c} = \begin{pmatrix} 0 & 1 & 0 \\ c^2 - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \text{ et } \mathbf{A}_{G_c} = \begin{pmatrix} 0 & 0 & 1 \\ -uv & v & u \\ c^2 - v^2 & 0 & 2v \end{pmatrix} \quad (1.15)$$

$S = \text{sign}(\mathbf{A})$ signifie que S possède les mêmes vecteurs propres que la jacobienne mais ses valeurs propres sont égales au signe des valeurs propres de \mathbf{A} . S représente la viscosité numérique ajoutée pour stabiliser le système. Elle est évaluée à l'interface avec comme état moyen $\mathbf{W} = \frac{\mathbf{W}_L + \mathbf{W}_R}{2}$. Par rapport aux solveurs de Riemann comme celui de Godunov, la décomposition de flux présente un gain de temps et une robustesse non négligeable tout en donnant des résultats comparables.

Par ailleurs, les états droits et gauches définissant le problème de Riemann peuvent être les valeurs moyennes des cellules (conduisant à un schéma d'ordre un) ou des reconstructions d'ordre plus élevé à l'interface pour réduire la diffusion numérique. Van Leer a ainsi proposé en 1979 le schéma MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) où l'approximation constante par morceaux est remplacée par une approximation linéaire par morceaux. A la fin des années 1980, le développement de schémas ENO (Essentially Non-Oscillatory) par Harten, Engquist et Osher [Harten et al., 1987] a ouvert la porte à des procédures de reconstruction d'ordre supérieur à deux. D'une façon générale, la reconstruction des valeurs aux faces repose sur un ensemble de points d'appui (appelé *stencil* dans la littérature anglophone) de part et d'autre de la face. Plus l'ordre auquel on souhaite reconstruire les valeurs aux faces est élevé, plus le stencil sera grand. L'idée de base des schémas ENO est de choisir, suivant des critères bien précis, le stencil qui contient la solution discrète la plus régulière parmi un certain nombre de candidats. Les schémas WENO (Weighted Essentially Non-Oscillatory) introduits par Liu, Osher et Chan [Liu et al., 1994] ordonnent ces schémas ENO en une combinaison convexe de stencils pondérés par des indicateurs de régularité. Ceci permet d'augmenter l'ordre de reconstruction par rapport au schéma ENO. Dans les régions où l'écoulement est régulier, l'ordre de reconstruction est maximal. A l'inverse, en présence d'un choc, les indicateurs de régularité sélectionnent le stencil le plus adapté, faisant chuter localement l'ordre de reconstruction. De nombreuses déclinaisons des schémas ENO ou WENO ont depuis été proposées : CENO (Central ENO) [Ivan et Groth, 2007], CWENO (Central WENO) [Levy et al., 1999], CRWENO (Compact-Reconstruction WENO) [Ghosh et Baeder, 2012], schéma MOOD (Multi-dimensional Optimal Order Detection) [Clain et al., 2011], etc.

Dans WCCH, le choix s'est porté sur le schéma originel WENO d'ordre 5 de Shu [Jiang et Shu, 1996]. Cinq cellules d'appui sont donc nécessaires et trois stencils sont envisagés (Fig. 1.3) :

$$S_0 = (i, i + 1, i + 2), S_1 = (i - 1, i, i + 1) \text{ et } S_2 = (i - 2, i - 1, i) \quad (1.16)$$

Les indicateurs de régularité pour chacun des stencils sont :

$$\begin{aligned} IS_0 &= \frac{13}{12}(u_i - 2u_{i+1} + u_{i+2})^2 + \frac{1}{4}(3u_i - 4u_{i+1} + u_{i+2})^2 \\ IS_1 &= \frac{13}{12}(u_{i-1} - 2u_i + u_{i+1})^2 + \frac{1}{4}(u_{i-1} - u_{i+1})^2 \\ IS_2 &= \frac{13}{12}(u_{i-2} - 2u_{i-1} + u_i)^2 + \frac{1}{4}(u_{i-2} - 4u_{i-1} + 3u_i)^2 \end{aligned} \quad (1.17)$$

Les coefficients de pondération ω_r affectés à chaque polynôme sont déterminés à partir des indicateurs de régularité :

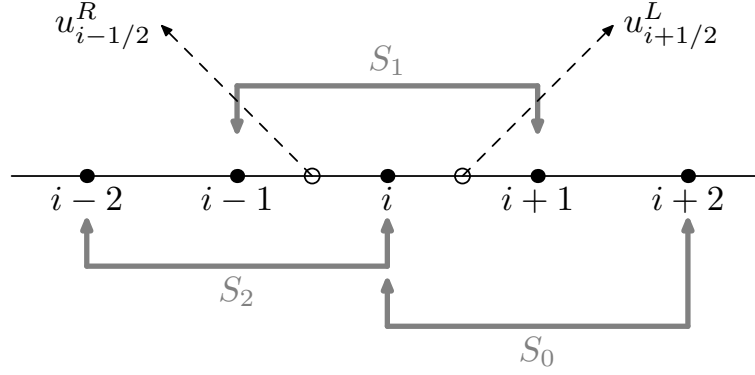


FIGURE 1.3 – Polynômes de reconstruction WENO 5

$$\alpha_r = \frac{d_r}{\epsilon + IS_r} \text{ et } \omega_r = \frac{\alpha_r}{\sum_{s=0}^2 \alpha_s}, r = 0, 1, 2 \quad (1.18)$$

où $\epsilon = 10^{-6}$ permet d'éviter les divisions par 0. $d_0 = \frac{3}{10}$, $d_1 = \frac{3}{5}$, $d_2 = \frac{1}{10}$ pour la reconstruction de $u_{i+1/2}^L$ et $d_0 = \frac{1}{10}$, $d_1 = \frac{3}{5}$, $d_2 = \frac{3}{10}$ pour la reconstruction de $u_{i-1/2}^R$ (Fig. 1.3). Ces poids conduisent au polynôme optimal sur cinq points (polynôme non biaisé).

Finalement les valeurs reconstruites aux interfaces gauche et droite sont :

$$\begin{aligned} u_{i+1/2}^L &= \omega_0 \left(\frac{1}{3}u_i + \frac{5}{6}u_{i+1} - \frac{1}{6}u_{i+2} \right) \\ &\quad + \omega_1 \left(-\frac{1}{6}u_{i-1} + \frac{5}{6}u_i + \frac{1}{3}u_{i+1} \right) \\ &\quad + \omega_2 \left(\frac{1}{3}u_{i-2} - \frac{7}{6}u_{i-1} + \frac{11}{6}u_i \right) \\ u_{i-1/2}^R &= \omega_0 \left(\frac{11}{6}u_i - \frac{7}{6}u_{i+1} + \frac{1}{3}u_{i+2} \right) \\ &\quad + \omega_1 \left(\frac{1}{3}u_{i-1} + \frac{5}{6}u_i - \frac{1}{6}u_{i+1} \right) \\ &\quad + \omega_2 \left(-\frac{1}{6}u_{i-2} + \frac{5}{6}u_{i-1} + \frac{1}{3}u_i \right) \end{aligned} \quad (1.19)$$

$u_{i+1/2}^L$ et $u_{i-1/2}^R$ définissent alors les états à gauche et à droite du problème de Riemann à l'interface.

Pour un problème multidimensionnel, la définition d'un schéma WENO avec des polynômes d'interpolation multi-variables est épineux. On préfère souvent traiter les opérateurs spatiaux dans chaque direction d'espace de façon indépendante suivant le principe des directions alternées. Un premier balayage de la grille cartésienne suivant x permet ainsi de reconstruire les variables conservatives sur les faces de normale x puis un autre balayage suivant y permet la reconstruction aux faces de normale y , etc (Fig. 1.4).

Discrétisation des termes visqueux

Le tenseur des contraintes visqueuses $\bar{\bar{\tau}} = 2\mu \left[\frac{\bar{\nabla} \mathbf{u} + (\bar{\nabla} \mathbf{u})^t}{2} - \frac{1}{3}(\nabla \cdot \mathbf{u}) \bar{\mathbf{I}} \right]$ est évalué sur chaque face des cellules du maillage. Les écoulements étudiés étant (quasi-) incompressibles, le terme de divergence de vitesse est négligeable dans cette expression d'où $\bar{\bar{\tau}} \approx \mu(\bar{\nabla} \mathbf{u} + (\bar{\nabla} \mathbf{u})^t)$.

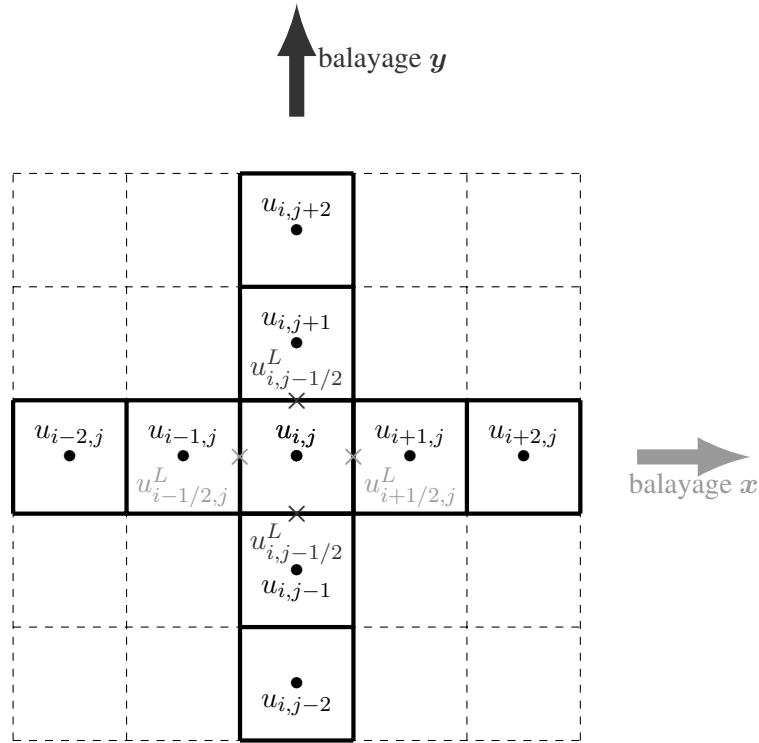


FIGURE 1.4 – Reconstructions WENO en 2D par directions alternées.

En 3D le tenseur $\overline{\overline{\nabla}} \mathbf{u}$ s'écrit :

$$\overline{\overline{\nabla}} \mathbf{u} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix} \quad (1.20)$$

En pratique les termes du tenseur n'ont pas besoin d'être tous calculés. En effet $\nabla \cdot \overline{\tau}$ se simplifie en un Laplacien de vitesse, $\Delta \mathbf{u}$, avec l'hypothèse $\nabla \cdot \mathbf{u} = 0$. Par exemple, pour une face de normale x , seuls les calculs des dérivées partielles des vitesses par rapport à x sont effectués. Un schéma centré d'ordre deux en espace est utilisé.

1.2.3 Schéma d'avance en temps

L'avance en temps intervient après le calcul des flux par l'intégration temporelle de l'équation différentielle ordinaire (1.11). La résolution explicite de (1.11) permet de faire évoluer $\mathbf{W}_{i,j,k}$ d'un instant t^n à l'instant suivant $t^n + \Delta t$ à partir de flux évalués à l'instant courant t^n . Cette résolution est très rapide mais instable si le pas de temps n'est pas suffisamment petit et ne respecte pas la condition de Courant-Friedrichs-Lewy (CFL).

Ainsi pour la partie hyperbolique des équations de Navier-Stokes, cette condition s'écrit :

$$\Delta t_c \leq CFL_c \min_{i \in I} \frac{\Delta x_i}{|u_i + c_i|} \quad (1.21)$$

avec Δx_i la taille d'une cellule i , u_i sa vitesse et c_i sa vitesse du son. Pour un schéma Euler d'ordre 1, il faut par exemple respecter la condition $CFL_c < 1$ qui traduit le fait que l'information ne peut pas se déplacer de plus d'une taille de cellule au cours d'un pas de temps.

Par ailleurs la résolution explicite de la partie elliptique de l'équation de quantité de mouvement nécessite elle aussi le respect d'une condition de stabilité additionnelle sur le pas de temps employé. Cette condition s'exprime sous la forme :

$$\Delta t_v \leq CFL_v \rho \frac{\min_{i \in I} \Delta x_i^2}{\mu} \quad (1.22)$$

CFL_v étant le coefficient CFL de diffusion. En théorie, sur une équation linéaire d'advection-diffusion, on peut montrer que ce nombre ne peut dépasser 0.5 pour un schéma Euler d'ordre 1 [Hirsch, 2007]. Finalement le pas de temps du schéma numérique est choisi pour respecter ces deux conditions :

$$\Delta t = \min(\Delta t_c, \Delta t_v) \quad (1.23)$$

Pour des applications à nombre de Reynolds élevé, le pas de temps est plutôt imposé par la partie hyperbolique en pratique.

Dans WCCH une méthode à pas multiples de Runge et Kutta est principalement employée. Pour une EDO de la forme $\mathbf{y}' = f(t, \mathbf{y})$, la méthode de Runge et Kutta consiste à évaluer la dérivée f en plusieurs points de l'intervalle $[t^n, t^{n+1}]$ pour augmenter la précision de façon itérative. Une méthode Runge-Kutta d'ordre s est donnée par :

$$\forall i = 1, \dots, s \begin{cases} t^i = t^n + c_i \Delta t \\ \mathbf{y}^i = \mathbf{y}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} f(t^j, \mathbf{y}^j) \end{cases} \quad (1.24)$$

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \Delta t \sum_{j=1}^s b_j f(t^j, \mathbf{y}^j)$$

Les coefficients a_{ij} , b_j et c_j sont résumés dans le tableau de Butcher :

c_1					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
\vdots	\vdots		\ddots		
c_s	a_{s1}	a_{s2}	\cdots	$a_{s,s-1}$	
	b_1	b_2	\cdots	b_{s-1}	b_s

Pour la méthode de Runge-Kutta d'ordre 4 utilisée dans WCCH :

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Pour ce schéma, nous pouvons fixer CFL_c jusqu'à des valeurs assez élevées : $CFL_c \approx 0.7$, tout en maintenant une bonne stabilité numérique.

1.2.4 Méthode de frontière immergée

Nous présentons succinctement dans ce paragraphe la méthode de frontière immergée initialement présente dans WCCH pour prendre en compte la présence d'un corps solide au sein de la grille cartésienne.

Le chapitre 5 sera ensuite consacré à son analyse en détail, et à la manière dont elle s'insère dans le large spectre des méthodes de frontière immergée.

Si une grille cartésienne simplifie la construction algorithmique d'un solveur d'ordre élevé, elle se révèle moins pratique d'usage lorsqu'il s'agit de traiter des écoulements autour d'obstacles. L'approche par maillage adapté (body-fitted) qui consiste à faire coïncider les nœuds du maillages avec la frontière du corps s'est largement imposée, au point d'être aujourd'hui un point commun à la majorité des codes commerciaux (Fig. 1.5). Avec cette approche, imposer des conditions aux limites est assez naturel. Néanmoins elle nécessite la construction de cellules de maillage qui s'adaptent parfaitement à la géométrie et respectant des critères de forme pour garantir la précision de la solution. La mise au point de maillages automatiques performants sur des géométries complexes reste une tâche ardue. Le terme de frontière immergée regroupe l'ensemble des méthodes qui permettent de traiter un solide dans une grille sans que ses frontières correspondent aux nœuds du maillage volumique (Fig. 1.5). Ces méthodes, bien qu'initiées au début des années 70, sont toujours un thème actif de recherche. L'un des avantages majeurs des méthodes de frontière immergée est en général la facilité avec laquelle le maillage peut être généré. Un autre atout est de pouvoir traiter de grands déplacements de solides ou des interactions multi-corps. En revanche, imposer des conditions aux limites adéquates n'est plus aussi immédiat qu'en maillage adapté. Notamment le lien entre le traitement des conditions aux limites et la précision (ou les propriétés de conservation) n'est pas trivial.

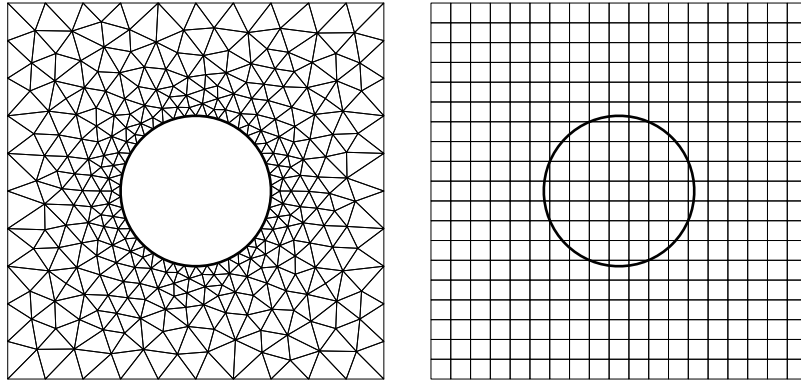


FIGURE 1.5 – Approche à maillage adapté ou "body-fitted" (à gauche) et immersion d'un cercle dans une grille cartésienne par frontière immergée (à droite)

Dans le code WCCH, l'approche adoptée consiste à décomposer la quantité de mouvement d'une cellule (i, j, k) à un instant t^n en une quantité de mouvement fluide et une quantité de mouvement solide [Bigay et al., 2017] :

$$\left(\frac{\partial(\rho\mathbf{u})_{i,j,k}}{\partial t} \right)^n = (1 - H_{i,j,k}) \left(\frac{\partial(\rho\mathbf{u})_{i,j,k}}{\partial t} \right)^n_{\text{fluide}} + H_{i,j,k} \rho_{i,j,k}^n \left(\frac{\partial\mathbf{u}_{i,j,k}}{\partial t} \right)^n_{\text{solide}} \quad (1.25)$$

$H_{i,j,k}$ est une fonction indicatrice qui explicite l'appartenance ou non d'une cellule au fluide :

$$H_{i,j,k} = \begin{cases} 0 & \text{si } (i, j, k) \text{ appartient au fluide} \\ 1 & \text{si } (i, j, k) \text{ appartient au corps} \end{cases} \quad (1.26)$$

Afin de ne pas imposer brutalement l'accélération dans le corps, H est régularisée à la traversée de la frontière pour progressivement atténuer le terme $\left(\frac{\partial(\rho\mathbf{u})_{i,j,k}}{\partial t} \right)^n_{\text{fluide}}$ et activer le forçage :

$$H_{i,j,k} \left(\frac{\phi}{\Delta x} \right) = \frac{1}{2} \left(1 + \tanh \left(-\alpha \frac{\phi_{i,j,k}}{\Delta x} \right) \right) \quad (1.27)$$

où $\phi_{i,j,k}$ désigne la distance signée de la cellule (i, j, k) par rapport à la frontière solide au sens de la ligne de niveau (ou level-set en anglais). ϕ est prise négative dans le corps et positive dans le fluide. Le coefficient α permet d'ajuster la largeur de la zone de transition. Plus ce nombre est proche de 0, plus la transition est diffuse et s'étale sur de nombreuses cellules. A l'inverse une valeur très élevée de α fait tendre H vers la fonction Heaviside stricte (1.26). Une valeur de $\alpha = 3$ donne en général un bon compromis entre le lissage des champs et la convergence en espace du calcul. La zone de transition s'étale alors sur $\sim 2.5\Delta x$.

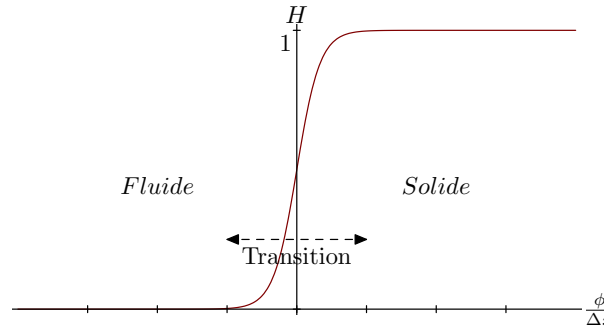


FIGURE 1.6 – Fonction de régularisation à la traversée de la frontière

Par ailleurs $\left(\frac{\partial(\rho\mathbf{u})_{i,j,k}}{\partial t}\right)_{\text{fluide}}^n$ est directement issu du calcul de la somme des flux et des termes sources notée $\mathbf{RHS}_{i,j,k}^n$:

$$\left(\frac{\partial(\rho\mathbf{u})_{i,j,k}}{\partial t}\right)_{\text{fluide}}^n = \mathbf{RHS}_{i,j,k}^n \quad (1.28)$$

En faisant l'hypothèse d'une cinématique de corps imposée, comme cela a été envisagé jusqu'à présent dans WCCH, l'accélération solide est connue et peut être approchée au cours d'une itération en temps par :

$$\left(\frac{\partial\mathbf{u}_{i,j,k}}{\partial t}\right)_{\text{solide}}^n \approx \frac{\mathbf{u}_{\text{solide}}^{n+1} - \mathbf{u}_{i,j,k}^n}{\Delta t} = \mathbf{S}_{i,j,k}^n \quad (1.29)$$

où $\mathbf{u}_{\text{solide}}^{n+1}$ est la vitesse du corps à l'instant t^{n+1} prise au centre de la cellule (i, j, k) . Ainsi l'équation (1.25) devient :

$$\left(\frac{\partial(\rho\mathbf{u})_{i,j,k}}{\partial t}\right)^n = (1 - H_{i,j,k})\mathbf{RHS}_{i,j,k}^n + H_{i,j,k}\rho_{i,j,k}^n\mathbf{S}_{i,j,k}^n \quad (1.30)$$

ou encore :

$$\left(\frac{\partial(\rho\mathbf{u})_{i,j,k}}{\partial t}\right)^n = \mathbf{RHS}_{i,j,k}^n + H_{i,j,k}\left(\rho_{i,j,k}^n\mathbf{S}_{i,j,k}^n - \mathbf{RHS}_{i,j,k}^n\right) \quad (1.31)$$

pour mettre en évidence le forçage sous forme d'un terme source additionnel.

Enfin les efforts appliqués par le fluide sur le solide sont calculés en sommant ces termes sources :

$$\mathbf{F} = \sum_{i,j,k} H_{i,j,k} \left(\rho_{i,j,k} \mathbf{S}_{i,j,k}^n - \mathbf{RHS}_{i,j,k}^n \right) \quad (1.32)$$

1.2.5 Maillage et structure de données

En CFD on distingue classiquement deux grandes classes de maillage : les maillages structurés et les maillages déstructurés. Les maillages purement cartésiens appartiennent à la première catégorie. Au sens

large, le maillage structuré est défini par un réseau de courbes, dans les trois directions de l'espace, qui s'intersectent. La définition du voisinage de chaque point de la grille structurée est aisé à partir d'indices de repérage de l'espace i, j, k . Déstructuré, le maillage est défini par une liste d'interconnexion entre des quelconques (Fig. 1.7).

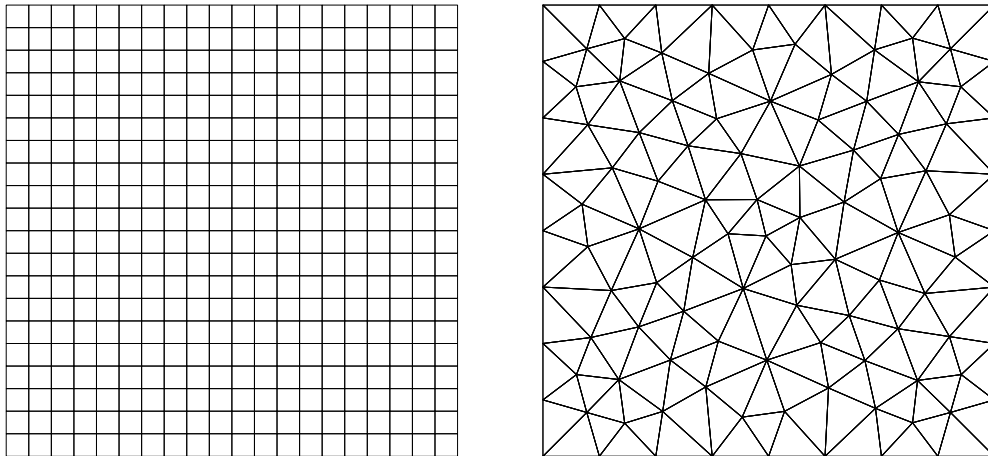


FIGURE 1.7 – Maillage cartésien structuré uniforme (à gauche) ou déstructuré, éléments triangles (à droite).

Historiquement les maillages cartésiens structurés ont été les premiers utilisés puisque l'application de schémas de différences finies peut être par exemple immédiate sur ces grilles. Le voisinage de chaque cellule est effectivement clairement défini de sorte que la vectorisation des codes de calcul s'en trouve grandement facilitée. De plus, dans le cadre des volumes finis, les maillages cartésiens structurés préservent naturellement l'ordre 2 de la méthode du point milieu utilisée pour évaluer les flux aux faces des cellules. En effet les centres de cellules où sont stockés les variables d'état sont parfaitement alignés avec les centres des faces où sont calculés les flux (Fig. 1.8).

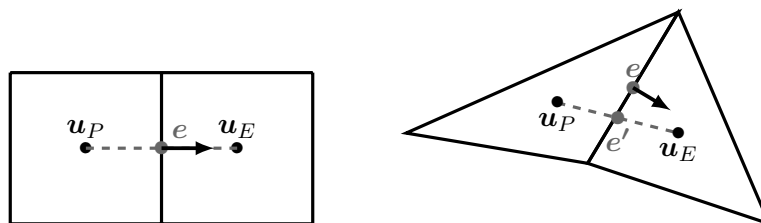


FIGURE 1.8 – Centre de face e et intersection e'

Sur les maillages déstructurés, des calculs supplémentaires sont nécessaires afin de reconstruire les variables et leurs gradients au centre des faces des cellules (Fig. 1.8). De plus une reconstruction de qualité requiert en général un angle minimal entre le vecteur normal de la face et la droite liant les centres de gravité des cellules contiguës. Les logiciels de maillage ont à charge de produire des maillages qui tendent vers une orthogonalité maximale pour minimiser l'erreur (cf. [Ferziger et Perić, 1996]). Sur un maillage purement cartésien, ces problèmes de métrique sont absents.

Les maillages cartésiens structurés présentent donc un certain nombre d'atouts. Néanmoins, le défaut principal de ces maillages est la difficulté à résoudre un écoulement hautement complexe multi-échelle avec un coût CPU acceptable. On peut distinguer alors trois approches possibles, dont les dénominations en anglais sont :

- *h-refinement*. La taille des cellules est adaptée en augmentant ou diminuant le nombre de points discrets.
- *p-refinement*. Cette approche courante pour la méthode des éléments finis consiste à augmenter le degré p du polynôme des fonctions nodales. Dans un contexte de volumes finis, cela revient à augmenter l'ordre de reconstruction des variables aux faces en élargissant le stencil d'interpolation.
- *r-refinement*. A partir d'un nombre fixe de points de discrétisation, ceux-ci sont réorganisés dans le domaine suivant des critères précis.

La méthode *p-refinement* paraissait peu adaptée dans le cadre du code WCCH puisqu'un schéma WENO 5 est déjà employé. Pour éviter d'élargir le stencil, la méthode Discontinuous-Galerkin (DG) constituerait une alternative qui tire profit des avantages des volumes finis et éléments finis en découplant les éléments et en permettant une montée en ordre locale. Ensuite pour des grilles cartésiennes, le *r-refinement* reviendrait à étirer ou resserrer les lignes qui définissent la grille (stretching en anglais). Mais les maillages produits ne sont pas optimaux et génèrent des zones de raffinement non voulues hors des régions d'intérêt (Fig. 1.9). Enfin, l'approche *h-refinement* selon un raffinement local est une stratégie séduisante car elle permet de concentrer le maillage de façon locale là où l'écoulement est sous-évalué, souvent à proximité des frontières solides. Le caractère cartésien est alors conservé localement (Fig. 1.9). Ainsi le coût CPU s'en trouve grandement amélioré avec peu de pertes de précision. De plus il est aisé de raffiner/déraffiner le maillage au cours de la simulation pour l'adapter à la physique de l'écoulement (AMR ou Adaptive Mesh Refinement). Pour les applications hydrodynamiques, un autre intérêt est de limiter les effets de bords comme la réflexion de vagues aux frontières du domaine en déraffinant exagérément le maillage pour diffuser ces ondes non désirées.

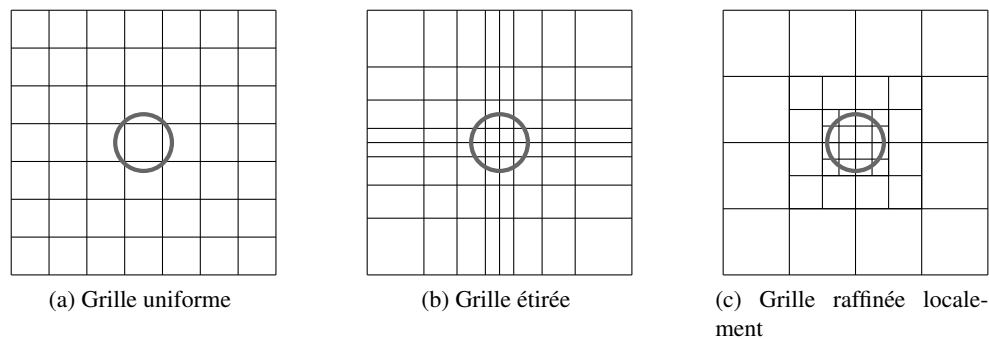


FIGURE 1.9 – Stratégie de raffinement de maillage cartésien autour d'un cercle

Dans WCCH, c'est donc finalement l'approche *h-refinement* qui a été adoptée.

Deux types de structure de données sont classiquement envisagées pour les maillages raffinés localement : le raffinement par patchs introduit par Berger et Olinger [Berger et Olinger, 1984] (ou patch-based en anglais) et le raffinement par quadtree/octree comme il a été employé par DeZeeuw, Powell et Coirier pour la résolution d'écoulements compressibles [DeZeeuw et Powell, 1993, Coirier, 1994].

Le raffinement par patchs est historiquement la première structure à avoir été proposée et consiste à raffiner des régions de l'espace à l'aide de patchs rectangulaires ou parallélépipédiques. Des grilles de plus en plus fines viennent donc remplacer des grilles originales moins raffinées. Les équations sont résolues sur les grilles les plus fines et la solution est interpolée sur les grilles plus grossières. La stratégie introduite initialement par Berger n'exigeait que peu de contraintes sur l'ordonnancement des grilles [Berger et Olinger, 1984]. Aucun alignement particulier des patchs n'était nécessaire. Plus tard, Berger et Colella ont montré

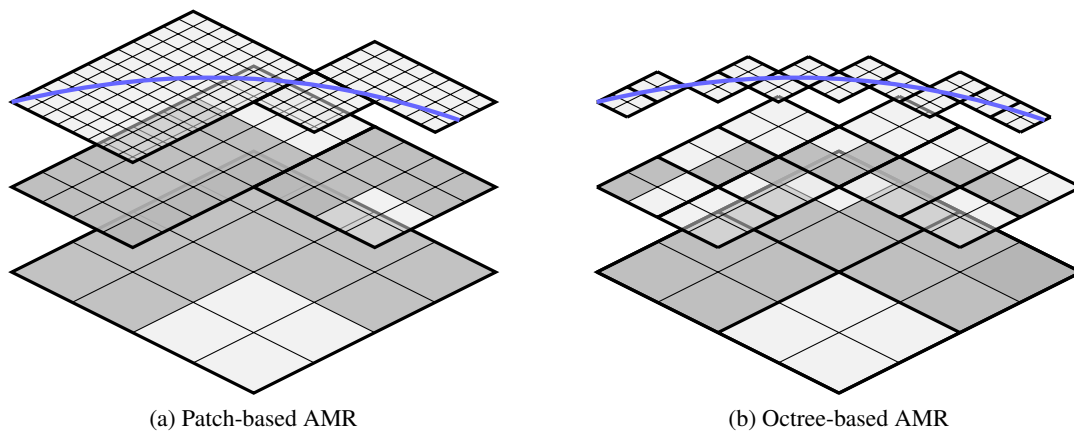


FIGURE 1.10 – Raffinement par patches ou en octree

que les schémas numériques étaient plus efficaces sur des patches alignés dont les coins coïncident entre eux avec un rapport de taille de maille n'excédant pas deux entre deux patches voisins [Berger et Colella, 1989]. La grille finale est formée par agglomération des patches (clustering en anglais) [Berger et Rigoutsos, 1991]. Le lien de parenté entre les grilles est conservé au moyen d'un graphe acyclique ou d'une liste chaînée.

L'extension d'un code développé sur une grille cartésienne structurée vers des grilles raffinées par des patches est assez facile en faisant appel à une librairie externe spécialisée qui gère les patches. L'uniformité des patches favorise la vectorisation du code et la parallélisation hybride OpenMP/MPI. L'équilibrage de la partition du maillage (load-balancing) est en revanche moins aisée. AMReX (anciennement BoxLib) [CCSE, 2017] est une librairie de raffinement par patches répandue. Un certain nombre de solveurs l'utilisent, CAESTRO [Almgren et al., 2010a], MAESTRO [Nonaka et al., 2010], Nyx [Almgren et al., 2013b], etc. SAMRAI [LLNL, 1997] et Chombo [Colella et Graves, 2000] sont également deux bibliothèques activement développées. La première présente l'avantage de s'interfacer aisément avec la librairie de résolution de systèmes linéaires PETSc [ANL, 1997] et de préconditionnement HYPRE [Falgout et Yang, 2002].

L'autre structure de données, l'octree (ou quadtree en 2D), est une structure hiérarchique qui subdivise l'espace de manière récursive. Chaque nœud de l'arbre peut se subdiviser en quatre ou huit enfants respectivement en 2D ou 3D.

Cette structure facilite la recherche du voisinage de chaque nœud et de nombreux algorithmes existent pour parcourir l'arbre de façon efficace. D'un point de vue algorithmique, les techniques de stockage de l'arbre sont variées, à base de pointeurs pour la méthode Fully Threaded Tree [Khokhlov, 1998] ou avec des courbes de remplissage spatial en Z de type Morton ou Lebesgue. En outre, il est possible de raffiner localement avec plus d'efficacité qu'avec la structure par patch (Fig. 1.10). De multiples bibliothèques ou codes parallèles de gestion d'octree/quadtree existent : Dendro [Sampath et al., 2008], Octor [Tu et al., 2005], PARAMESH [MacNeice et al., 2000], p4est [Burstedde et al., 2011] pour ne citer que les plus connus.

Pour une revue plus détaillée des structures à raffinement de grille adaptatif, le lecteur pourra se référer à l'article de Dubey et al. [Dubey et al., 2014].

Dans le code WCCH, c'est la structure en arbre octree qui a été adoptée.

Pour clarifier l'exposé de la structure employée, nous définissons quelques termes :

Cellule (cell) : un élément de discrétisation spatiale sur lequel les équations sont discrétisées.

Bloc (block) : élément rectangulaire ou parallélépipédique pouvant être raffiné par division en 4 ou 8 éléments enfants (en 2D ou 3D respectivement).

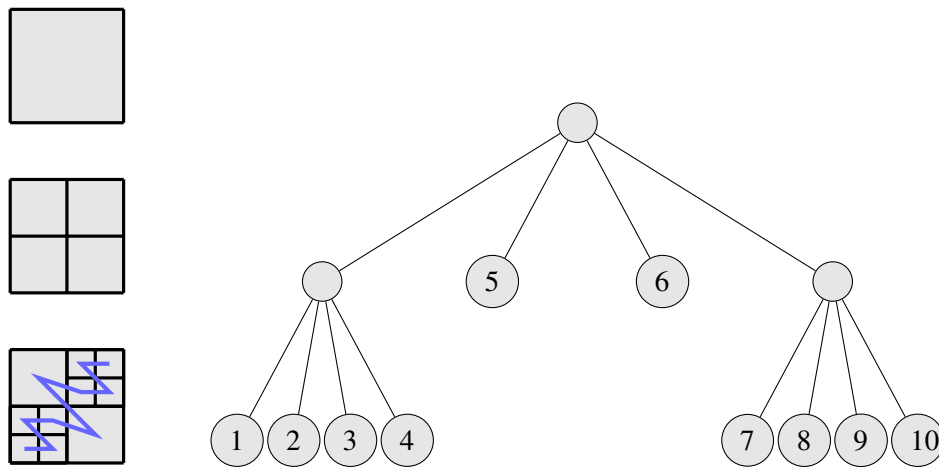


FIGURE 1.11 – Raffinement par octree et numérotation avec le Z de Morton

Arbre (quadtree/octree) : structure hiérarchique de données contenant le lien de filiation des blocs.

Bloc feuille (leaf block) : bloc en bout d'arbre contenant les cellules. Le nombre de cellules par bloc et par direction est identique pour tous les blocs. Par la suite le terme bloc désignera uniquement les blocs feuilles.

Bloc racine (root block) : il s'agit du premier bloc de raffinement dont la taille coïncide avec les dimensions du domaine.

Niveau de raffinement : nombre de divisions récursives appliquées pour partitionner l'espace. Si le bloc root a pour dimension $[\Delta x, \Delta y, \Delta z]$ alors le bloc raffiné au niveau l a pour dimension $[\frac{\Delta x}{2^l}, \frac{\Delta y}{2^l}, \frac{\Delta z}{2^l}]$. Le niveau du bloc root est $l = 1$.

Les blocs feuilles sont rangés en mémoire à l'aide d'un balayage en Z. Cette courbe passe par tous les centres des blocs voisins en décrivant un Z (Fig. 1.11). Chaque bloc est alors identifié par un code de Morton obtenu en entrelaçant la représentation binaire de ses coordonnées spatiales (x, y, z) .

Cette façon de ranger les éléments en mémoire permet de préserver au mieux la localité du voisinage pour chaque octant. Cela prévient des défauts de cache qui peuvent se produire lorsque les données sont trop dispersées dans la mémoire de l'ordinateur. En effet, les données contiguës sont chargées dans la mémoire cache d'accès rapide. Si une donnée ne s'y trouve pas au cours de l'exécution d'une portion de programme, celle-ci doit être chargée depuis la mémoire centrale avec un coût d'accès supérieur, ralentissant alors l'exécution.

Les blocs ne constituent pas les volumes de contrôle utilisés pour la résolution en volumes finis. Un nombre donné de cellules est attribué pour tous les blocs dans chaque direction $(n_{cells}(x), n_{cells}(y), n_{cells}(z))$. Ces nombres doivent être pairs avec un minimum de deux cellules par direction. Les cellules des blocs sont rangées en mémoire suivant un ordre lexicographique.

Chaque bloc possède un attribut (flag) de raffinement et de déraffinement. La valeur de cet attribut est basée sur des critères spécifiques. Ceux-ci peuvent-être purement géométriques : raffinement dans une région de l'espace ou en fonction d'une distance à un corps solide, ou encore sur la base de critères physiques : gradient de pression, vorticité de l'écoulement, etc. On parle dans les deux cas d'adaptation de maillage (AMR pour Adaptive Mesh Refinement). L'évaluation des critères se fait sur les cellules mais la décision de raffiner ou de déraffiner s'opère sur les blocs puisque seuls ces derniers ont la capacité de se subdiviser.

Deux fonctionnalités de raffinement sont présentes dans WCCH : l'adaptation statique et l'adaptation dynamique. Dans le premier cas, le maillage est généré uniquement en début de calcul alors que dans le second, l'adaptation de maillage est appelée régulièrement au cours de la simulation. Le nombre de cellules

évolue alors librement au cours du temps. Un niveau maximum de raffinement est cependant défini en début de calcul.

D'autre part une contrainte est imposée sur les blocs : il ne peut y avoir plus d'un niveau de raffinement d'écart entre deux blocs voisins, qu'ils soient voisins par face ou par arête commune (Fig. 1.12).

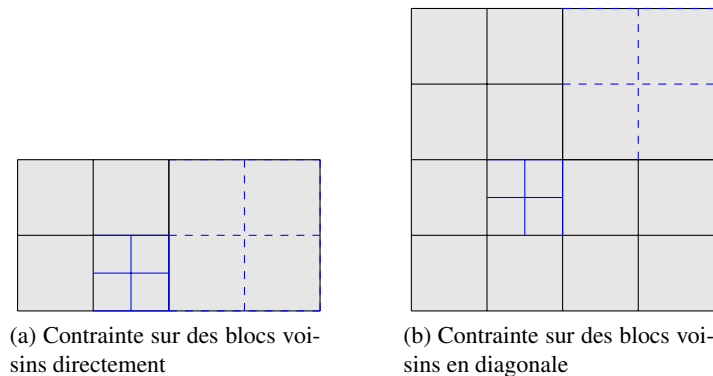


FIGURE 1.12 – Contrainte sur les niveaux de raffinement entre blocs voisins

Cette contrainte permet de faciliter les interpolations aux interfaces de raffinement puisqu'elle limite les configurations possibles. Elle permet aussi de limiter l'écart de résolution spatiale pouvant poser des problèmes numériques.

La gestion du maillage dans WCCH est détachée du programme principal de résolution des équations de Navier-Stokes. Une interface permet de faire le lien entre les blocs de l'arbre et les cellules de calculs. Cette distinction est utile notamment si une structure de données autre que l'octree est envisagée et laisse la possibilité d'intégrer une librairie tierce pour gérer le maillage à la place de notre propre implémentation.

Enfin, en ce qui concerne les schémas de discrétisation spatiale, ceux-ci sont adaptés aux interfaces de raffinement avec une procédure de pas variable.

1.2.6 Parallélisation

Afin d'accélérer les calculs, le code WCCH est parallélisé pour s'exécuter sur des machines à mémoire distribuée selon le standard MPI (Message Passing Interface). Chaque processeur se voit attribuer une portion de l'espace physique. La partition et la répartition équilibrée (load-balancing) du maillage sur chaque processeur sont instantanées puisqu'elles consistent à découper le maillage en suivant la numérotation de Morton des blocs. Elles ne nécessitent pas le recours à une librairie externe comme par exemple SCOTCH [Pellegrini et Roman, 1996] ou ParMETIS [Karypis et Kumar, 1998]. Les sous-domaines résultant de la partition ne se recouvrent pas. Lors de l'exécution du code, les cellules hébergées par un processeur ont besoin de valeurs de cellules hébergées par ses processeurs voisins (pour le calcul des flux par exemple). En pratique, le code étant explicite, seules les valeurs des cellules situées à la frontière des sous-domaines ont besoin d'être échangées par communications MPI. Chaque bloc est qualifié soit de *local*, *inner* ou *outer* (Fig. 1.13). Cette classification LIO renvoie à la façon dont sont perçus les blocs du point de vue d'un processeur (cf. [Oger et al., 2016]) :

- Les blocs LOCAL sont situés à l'intérieur du sous-domaine et n'ont pas besoin d'interagir avec les blocs des autres processeurs.
- Les blocs INNER sont à la frontière du sous-domaine et sont amenés à interagir avec les blocs OUTER (qui sont les blocs INNER des processeurs voisins). Leurs données sont envoyées aux processeurs voisins concernés.

— Les blocs OUTER sont hébergés par les autres processeurs (pour qui ils sont des blocs INNER).

Leurs données sont reçues par le processeur, en provenance des INNER des processeurs voisins.

Les communications se font donc entre blocs INNER et OUTER. Toute l'efficacité parallèle du code repose sur la stratégie de masquage des communications INNER/OUTER par des calculs sur les blocs locaux et via des communications non bloquantes.

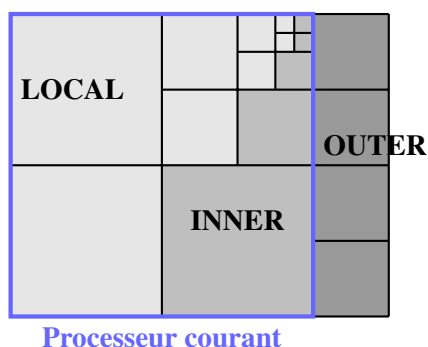


FIGURE 1.13 – Blocs LOCAL - INNER - OUTER

1.2.7 Exemple d'un écoulement autour d'un cylindre simulé avec WCCH

Pour illustrer la résolution d'un écoulement hydrodynamique par le code WCCH, une simulation d'écoulement 2D autour d'un cylindre à Reynolds 100 est présentée en figure 1.14. Elle est issue de la thèse de P. Bigay [Bigay, 2015], qui regroupe un certain nombre de cas de validation de la méthode. Pour ce nombre de Reynolds, l'écoulement est instationnaire et un détachement périodique alterné de tourbillons s'installe (allées de Von Kármán). Cette simulation est comparée à des résultats obtenus avec la méthode DVH (Diffused Vortex Hydrodynamics). Cette méthode particulière sans maillage développée par Colagrossi et Graziani [Rossi et al., 2015] est spécialement adaptée aux écoulements instationnaires autour de géométries complexes à Reynolds relativement modérés ce qui nous permet de considérer ces résultats comme de référence. Les résultats du code WCCH sont obtenus avec une discrétisation spatiale donnant 40 cellules dans le diamètre du cylindre (Fig. 1.14).

1.3 Problématiques actuelles du code WCCH

Après cette présentation partielle du code, il est possible d'exposer plus en détail les éléments de la méthode WCCH qui peuvent représenter un frein à son efficacité et à sa précision.

1.3.1 Limitations de l'approche faiblement-compressible pour la simulation à bas nombre de Mach

Limitation des solveurs compressibles à bas Mach

Les écoulements (quasi-)incompressibles constituent une limite singulière des écoulements compressibles. Sans modification particulière, la majorité des schémas développés en formulation compressible se révèle inefficace quand le nombre de Mach tend vers 0, voire instable pour des nombres de Mach très faibles. Il convient de se demander dans quelle mesure l'approche faiblement-compressible peut être affectée par ces problèmes et si les solutions apportées en formulation compressible à bas Mach pourraient s'appliquer dans notre contexte.

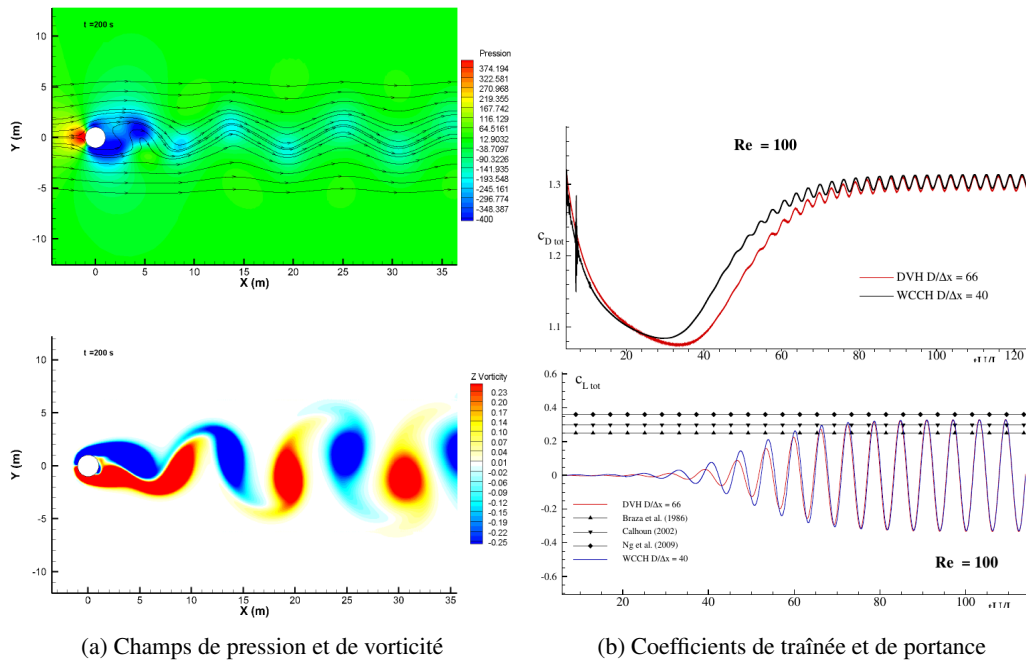


FIGURE 1.14 – Validation de WCCH pour un écoulement à $Re = 100$ autour d'un cylindre

Un développement asymptotique en nombre de Mach permet de montrer que les fluctuations de pression sont en $O(Ma^2)$ [Capdeville, 2006]. Mais Guillard et Viozat ont rigoureusement montré que le système d'équations discret obtenu avec une méthode de décomposition de flux comme celle utilisée dans WCCH conduit à des fluctuations numériques de pression en $O(Ma)$ [Guillard et Viozat, 1999]. Par ailleurs, la diffusion introduite par la partie décentrée des méthodes à décomposition de flux est proportionnelle à $O\left(\frac{1}{Ma}\right)$, entraînant une dissipation excessive de la solution discrète pour $Ma \rightarrow 0$.

En faiblement-compressible, bien que les choses soient un peu différentes puisque le nombre de Mach est fixé a priori autour de 0.1, il n'en reste pas moins qu'une diffusion numérique en $O\left(\frac{1}{Ma}\right)$ et des fluctuations de pression en $O(Ma)$ soulèvent des interrogations.

La limitation embarrassante des schémas compressibles se manifeste dans un certain nombre de simulations d'écoulements industriels. En aéronautique par exemple, les écoulements autour de profils d'ailes d'avions à des angles d'attaque élevés génèrent des écoulements locaux à bas nombre de Mach alors que des chocs et d'autres effets compressibles peuvent se produire par ailleurs dans d'autres régions de l'écoulement. Les transferts thermiques peuvent influencer également sur la compressibilité du fluide. Finalement les ingénieurs aéronautiques privilégient des outils reposant sur des schémas purement compressibles pour simuler l'ensemble des phénomènes physiques existants. Dans l'industrie nucléaire, les écoulements dans les circuits primaires et secondaires sont aussi soumis à des régimes variables. Ainsi depuis les années 1980, de nouveaux schémas dits *tout régime* ont été proposés dans la littérature. Malgré tout, la mise au point de tels schémas constitue toujours un défi et un thème actif de recherche.

Deux techniques principales ont émergé pour résoudre des écoulements à bas Mach : celle du préconditionnement et les méthodes asymptotiques. La première, introduite par Turkel en 1987 [Turkel, 1987], consiste à prémultiplier le système d'équations de Navier-Stokes par une matrice dont le rôle est de réduire les disparités des ordres de grandeur des valeurs propres. Choi et Merkle ont étendu la méthode à des écoulements instationnaires visqueux [Choi et Merkle, 1993]. Parce que le préconditionnement abaisse le nombre de Courant par rapport au schéma sans normalisation, une partie des termes des équations est souvent implicite, conduisant ainsi à la résolution d'un système linéaire. Pour le schéma à décomposition de flux de

Ghidaglia, une renormalisation de la matrice de diffusion a été proposée par Dauvergne et al. [Dauvergne et al., 2008]. Le principe des méthodes asymptotiques est quant à lui de développer asymptotiquement les équations discrètes en nombre de Mach afin de séparer la résolution des phénomènes rapides (acoustiques) et celle des phénomènes "lents" (advection-diffusion). Des schémas semi-implicites en temps sont alors élaborés.

Solutions envisagées

Malgré les solutions attractives apportées par ces schémas *tout régime*, notre objectif diffère quelque peu des préoccupations des industries citées ci-dessus puisque les écoulements à régime variable sont peu fréquents en hydrodynamique navale. Si on exclut certains cas particuliers comme les écoulements de bal-lottements dans les cuves de méthanières (sloshing) ou les explosions sous-marines (militaires ou pour la prospection pétrolière), les écoulements rencontrés sont en grande majorité incompressibles. Le choix du faiblement-compressible pour WCCH s'est fait avant tout pour le caractère explicite de la méthode, qui la rend particulièrement adaptée aux architectures de calcul modernes. Envisager une approche *tout régime* pour le code WCCH ne serait a priori pas appropriée pour nos applications. Dans la mesure où la majorité des solutions énoncées auparavant repose sur la résolution d'un système linéaire, passer à une approche purement incompressible paraît plus cohérent et adapté. Ainsi vient l'idée de développer un schéma incompressible dans le code. L'objectif est alors de positionner l'approche faiblement-compressible par rapport à une approche purement incompressible et de trancher sur les avantages de chacune des méthodes sur des critères de diffusion numérique, de performances HPC, etc.

Estimation théorique de la diffusion numérique faiblement-compressible/incompressible

Sans entrer dans les détails des schémas numériques, il est possible d'estimer théoriquement et sommairement le gain de la formulation incompressible par rapport à la formulation faiblement-compressible.

Nous considérons un écoulement 1D dans un tube à section variable (Fig. 1.15). Les grandeurs sont supposées constantes sur chaque section et ne varient que suivant x .

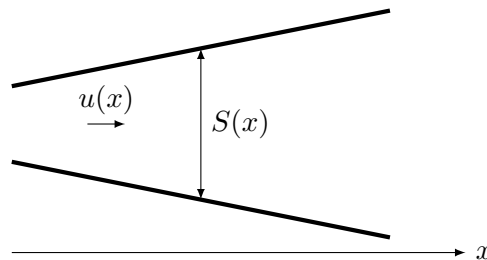


FIGURE 1.15 – Ecoulement quasi 1D

Pour un écoulement non visqueux, l'équation 1D qui modélise l'écoulement compressible dans la tuyère s'écrit :

$$\frac{\partial w}{\partial t} + \frac{\partial F(w)}{\partial x} = f(x) \text{ avec } w = (\rho S, \rho u S), F(w) = (\rho u S, (\rho u^2 + p)S) \text{ et } f = \left(0, p \frac{dS}{dx}\right) \quad (1.33)$$

Ces équations diffèrent des équations d'Euler 1D standards par la présence de la section S de la tuyère et du terme source f . Néanmoins la matrice Jacobienne du flux reste la même que celle de l'équation 1D avec section constante.

$$A = \frac{\partial F(w)}{\partial w} = \begin{pmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{pmatrix} \quad (1.34)$$

Pour simplifier l'exposé, nous faisons l'hypothèse forte qu'aucun schéma de reconstruction des variables aux faces n'est employé. Le flux numérique à l'interface calculé avec la méthode de Ghidaglia [Ghidaglia et al., 1996, Ghidaglia, 2001] s'écrit $\phi = \frac{F_L + F_R}{2} - \text{sign}(A) \frac{F_R - F_L}{2}$. L et R désignent les états gauche et droit de l'interface des cellules. $\text{sign}(A)$ est la matrice de diffusion numérique comme introduite Eq. (1.14).

$$\text{sign}(A) = \frac{\text{sign}(u)}{2} \begin{pmatrix} -Ma & \frac{Ma}{u} \\ u(\frac{1}{Ma} - Ma) & Ma \end{pmatrix} \quad (1.35)$$

La dissipation introduite dans les équations de quantité de mouvement par la partie décentrée, notée χ_{comp} , est d'environ :

$$\chi_{comp} = \frac{\text{sign}(u)}{2} u \frac{1}{Ma} \Delta(\rho u) + O(\Delta(\cdot) \cdot Ma) \quad (1.36)$$

Δ est la variation de l'état droit par rapport à l'état gauche. Par exemple $\Delta(\rho u) = (\rho u)_R - (\rho u)_L$. Avec l'hypothèse d'une densité constante, l'équation (1.33) devient :

$$\begin{cases} \frac{\partial u}{\partial x} = -\frac{u}{S} \frac{dS}{dx} \\ \frac{\partial \rho u}{\partial t} + \frac{\partial \frac{1}{2} \rho u^2 + p}{\partial x} = 0 \end{cases} \quad (1.37)$$

Le flux numérique de la partie convective est alors :

$$\phi = \frac{(\frac{1}{2} \rho u^2)_L + (\frac{1}{2} \rho u^2)_R}{2} - \text{sign}(u) \frac{(\frac{1}{2} \rho u^2)_R - (\frac{1}{2} \rho u^2)_L}{2} \quad (1.38)$$

par analogie avec le schéma de décomposition de flux. La diffusion numérique vaut alors

$$\chi_{inc} = \frac{\text{sign}(u)}{2} \Delta(\frac{1}{2} \rho u^2) \approx \frac{\text{sign}(u)}{2} u \Delta(\rho u) \quad (1.39)$$

Le rapport des diffusions numériques est donc

$$\frac{\chi_{inc}}{\chi_{comp}} \approx \frac{1}{Ma} \approx 10 \text{ avec l'approche faiblement-compressible } (Ma \approx 0.1) \quad (1.40)$$

La quantité de diffusion numérique introduite par le décentrement dans le calcul du terme convectif en formulation faiblement-compressible est donc environ 10 fois plus importante que pour une approche incompressible. Ce résultat intéressant, et en faveur, de la formulation incompressible est toutefois sujet à question dans la mesure où en pratique des schémas de reconstruction d'ordre élevé (MUSCL ou WENO) sont privilégiés pour limiter cette diffusion numérique. De plus la quantité de diffusion introduite est à mettre en regard avec la diffusion réelle liée à la physique de l'écoulement. Sans doute que pour des écoulements à Reynolds modéré, la différence de traitement entre incompressible et faiblement-compressible sera minime. Mais pour des nombres de Reynolds plus élevés pour lesquels la partie convective domine, il est possible que ce problème revête son importance.

1.3.2 Précision et robustesse de la méthode de frontière immergée

Jusqu'à présent la méthode WCCH et sa technique d'immersion de frontière n'ont été validées que sur des écoulements laminaires autour d'obstacles à la géométrie simple. Le niveau d'exigence sur le raffinement de maillage et la précision des schémas numériques au voisinage de la frontière va s'accroître à mesure que l'on souhaitera viser des écoulements à plus haut Reynolds. Or, même si cela reste à démontrer, la méthode telle qu'elle a été décrite dans ce chapitre laisse supposer que son ordre de convergence est limité à 1. La non prise en compte de la position exacte de la frontière dans le forçage représente sans doute une limitation de la précision de la méthode. Dans le chapitre qui sera consacré aux méthodes de frontière immergée nous tâcherons d'analyser en détail la méthode actuelle pour ensuite en proposer une version d'ordre supérieur.

1.3.3 Immersion de géométries complexes

Au début de ce travail de thèse, l'immersion de solide se limitait à des formes analytiques (cylindre, ellipse, sphère, etc.) ou des géométries simples réduisant considérablement les champs d'application de la méthode. L'algorithme original d'immersion fonctionnait tant que la géométrie ne présentait pas de points singuliers comme le bord de fuite d'une aile d'avion. Sa complexité algorithmique en $O(N_{cells}N_{tri})$, où N_{cells} désigne le nombre de volumes de contrôle du maillage fluide et N_{tri} le nombre de triangles du maillage surfacique de la géométrie, ne permettait pas non plus de l'appeler régulièrement au cours de la simulation. La recherche d'une technique robuste et compatible avec des temps de restitution abordable a donc constitué le dernier aspect de cette thèse.

1.4 Conclusion

Ce chapitre a permis de présenter le solveur volumes finis faiblement-compressible WCCH tel qu'existant au début de ce travail de doctorat. Cette description non exhaustive du solveur se limite aux fonctionnalités du code ou aux aspects de la méthode qui apparaissent comme essentiels pour la présentation des développements conduits lors de cette thèse. Il fixe ainsi le cadre numérique et algorithmique des contributions apportées pendant ce travail de recherche, et qui font l'objet des prochains chapitres.

Chapitre 2

Développement d'une formulation incompressible dans le code WCCH

Dans le chapitre précédent, nous avons présenté les caractéristiques générales de WCCH. Le présent chapitre est dédié à la présentation de la formulation incompressible adoptée. Une revue partielle des schémas incompressibles est exposée avant de détailler l'approche choisie et son implémentation algorithmique.

2.1	Revue des schémas incompressibles	47
2.1.1	Rappel des équations en formulation incompressible	47
2.1.2	Méthodes à correction de pression	48
2.1.3	Méthodes de compressibilité artificielle	52
2.1.4	Bilan des méthodes incompressibles	53
2.2	Méthode de projection de Chorin et ses variantes	54
2.2.1	Méthode originelle	54
2.2.2	Déclinaisons	55
2.2.3	Schéma de projection finalement retenu	60
2.3	Traitements numériques	61
2.3.1	Calcul de la vitesse prédite	61
2.3.2	Résolution de l'équation de Poisson	62
2.3.3	Discretisation temporelle	71
2.3.4	Conditions aux limites	73
2.4	Adaptation de la méthode de frontière immergée pour la formulation incompressible	77
2.5	Conclusion	81

Remarque : dans les chapitres qui suivent, il pourra être fait référence au code *WCCH incompressible* pour désigner le schéma incompressible que nous avons implémenté dans le code WCCH, bien que l'association de l'acronyme WCCH et du terme *incompressible* soit antinomique.

2.1 Revue des schémas incompressibles

2.1.1 Rappel des équations en formulation incompressible

Nous rappelons dans un premier temps les équations de Navier-Stokes incompressibles sous forme locale :

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} + \mathbf{f} \end{cases} \quad (2.1)$$

où $\nu = \frac{\mu}{\rho}$ désigne la viscosité cinématique du fluide en $m^2.s^{-1}$. Toute la difficulté de ces équations est l'absence de la pression dans l'équation de continuité. Elle n'apparaît plus que dans les équations de quantité de mouvement via son gradient. L'équation de continuité prend la forme d'une contrainte cinématique sur la vitesse et n'a plus la forme d'une équation aux dérivées partielles (EDP) classique qui fait évoluer dynamiquement une quantité. Une façon d'envisager ces équations est de considérer la pression comme une variable permettant d'imposer la contrainte de divergence nulle de l'équation de continuité. La pression possède alors toutes les caractéristiques d'un multiplicateur de Lagrange qui permet à chaque instant aux champs de vitesses (dont l'évolution temporelle est dictée par les équations de quantité de mouvement) de respecter $\nabla \cdot \mathbf{u} = 0$. En termes mathématiques, le système d'équations (2.1) doit être envisagé comme un système d'équations différentielles-algébriques (Differential-Algebraic Equations, cf. [Ascher et Petzold, 1998]). Cette représentation de la vitesse et de la pression est à la base des méthodes de correction de pression et conduit à une résolution découplée de la vitesse et de la pression. La vitesse est clairement issue de l'équation de quantité de mouvement. Ce qui laisse à l'équation de continuité le rôle de fixer la pression. En pratique une équation instantanée sur la variable algébrique de pression est construite en appliquant l'opérateur divergence $\nabla \cdot$ à l'équation de quantité de mouvement :

$$\nabla \cdot \frac{\nabla p}{\rho} = \nabla \cdot \left(-\frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nu \nabla^2 \mathbf{u} + \mathbf{f} \right) \quad (2.2)$$

(2.2) est une équation de Poisson sur la pression dont la résolution est implicite en inversant un système linéaire. La relative simplicité algorithmique des méthodes à correction de pression, leur robustesse et la facilité avec laquelle il est possible de prendre en compte d'autres effets physiques (thermique, turbulence, transfert de masse, etc.) ont conduit à leur adhésion massive dans tous les codes industriels. Dans la suite, nous nous concentrons donc en particulier sur ces approches, la description du large éventail des schémas incompressibles étant hors de portée et présentant peu d'intérêt dans le cadre de cette thèse. Néanmoins nous présenterons au cours d'un paragraphe l'approche de résolution par *compressibilité artificielle* puisqu'elle ressemble en certains points à l'approche faiblement-compressible.

2.1.2 Méthodes à correction de pression

Le principe de ces méthodes est de résoudre dans un premier temps, de façon explicite ou implicite, le champ de vitesse à l'aide des équations de quantité de mouvement. Ce champ ne respecte pas en général la contrainte de divergence nulle. Une étape de correction est alors nécessaire pour rendre la vitesse à divergence nulle, avec un gradient de pression, qui aura été déterminé préalablement par la résolution de l'équation de Poisson.

Méthode de Harlow et Welch

Historiquement, Harlow et Welch [Harlow et Welch, 1965] ont été des pionniers dans les approches à correction de pression avec la méthode MAC (Marker-and-Cell Method) pour des écoulements à surface libre. Les équations sont discrétisées sur un maillage entrelacé (staggered en anglais), c'est-à-dire que la pression est stockée au centre des cellules alors que les vitesses sont stockées aux faces. Par la suite, le nom de la méthode MAC sera utilisé dans la littérature pour désigner ce type d'emplacement des variables bien qu'il se réfère à une méthode pour le suivi de surface libre.

\mathbf{u} vérifie l'équation semi-discrète :

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) - \frac{\nabla p^{n+1}}{\rho} + \nu \nabla^2 \mathbf{u}^n \quad (2.3)$$

dans laquelle la pression p^{n+1} est inconnue. En prenant la divergence de cette équation, et en considérant $\nabla \cdot \mathbf{u}^{n+1} = 0$ et $\nabla \cdot \mathbf{u}^n = 0$, il vient l'équation de Poisson :

$$\nabla \cdot \frac{\nabla p^{n+1}}{\rho} = \nabla \cdot \left[-\nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) \right] \quad (2.4)$$

la divergence du terme visqueux étant en effet nulle. La pression ainsi obtenue après la résolution du système linéaire est introduite dans (2.3).

Cette démarche a ensuite été formalisée à la fin des années 1960 par Chorin [Chorin, 1968] et Témam [Temam, 1969] avec les méthodes de projection.

Méthode de projection de Chorin et Témam

Ces méthodes de projection partent du constat que les équations de Navier-Stokes incompressibles sont semblables à une décomposition de Helmholtz-Hodge. Le théorème de décomposition de Helmholtz-Hodge stipule que tout champ de vecteur \mathbf{u} peut se décomposer en un champ de vecteur à divergence nulle \mathbf{u}^d et un champ gradient d'une fonction scalaire $\nabla\phi$. On peut ainsi définir un opérateur de projection P tel que :

$$\mathbf{u}^d = P(\mathbf{u}) \text{ et } \nabla\phi = (I - P)(\mathbf{u}) \quad (2.5)$$

P est un projecteur, au sens mathématique du terme, projetant un champ de vitesse quelconque dans l'espace des champs de vitesse à divergence nulle. I est l'opérateur identité. Ainsi, dans un schéma de projection, l'application de cet opérateur P permet d'assurer la condition d'incompressibilité pour le champ de vitesse recherché. En pratique, pour la résolution des équations de Navier-Stokes incompressibles, cette projection s'inscrit dans l'approche "prédiction-corrrection". Dans un premier temps un champ de vitesse intermédiaire est calculé de façon explicite (à l'origine implicite dans la méthode proposée par Chorin), puis l'équation de Poisson est formulée en prenant la divergence de l'équation de quantité de mouvement. Enfin la vitesse intermédiaire est projetée afin de la rendre à divergence nulle. Dans la formulation de Chorin, cela revient à résoudre successivement les équations suivantes au sein d'un pas de temps :

1. Prédiction

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \nu \nabla^2 \mathbf{u}^n - \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \mathbf{f}^n \quad (2.6)$$

2. Équation de Poisson en pression

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla p^{n+1} \right) = \nabla \cdot \mathbf{u}^* \quad (2.7)$$

3. Correction

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla p^{n+1} \quad (2.8)$$

La pression étant déterminée par la résolution d'un système linéaire, un tel schéma d'avance en temps peut être qualifié de semi-implicite. Le pas de temps choisi doit aussi respecter une condition de stabilité de type CFL.

La méthode de projection est simple à mettre en œuvre. Mais dans sa version explicite elle était peu pratique d'utilisation à une époque où les ressources informatiques limitées réduisaient les applications industrielles à des écoulements stationnaires ou quasi-stationnaires. Des approches implicites ont donc été suggérées pour la simulation de ces écoulements.

Méthode SIMPLE de Patankar et Spalding

Le schéma SIMPLE, pour Semi-Implicit Method for Pressure Linked Equations, a été introduit par Spalding et Patankar [Patankar et Spalding, 1972] au début des années 1970. En dépit de son acronyme, il s'agit bien d'une méthode implicite. La discrétisation et la linéarisation de (2.1) conduit au système matriciel suivant :

$$\begin{pmatrix} E - Q & G \\ D & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \quad (2.9)$$

$E - Q$ désigne la matrice résultant de la discrétisation des équations de quantité de mouvement. E est sa portion diagonale alors que Q est sa partie extra-diagonale. f est le terme source. G est l'opérateur gradient et D l'opérateur divergence. A cause des non-linéarités, le système (2.9) est résolu de manière itérative en répétant :

1. Le système (2.9) est construit en partant d'une estimation u^k de la vitesse et une pression p^k (issues de la précédente itération ou de l'initialisation au début de l'algorithme)
2. Les équations de quantité de mouvement sont résolues :

$$u^* = (E - Q)^{-1}(f - Gp^k) \quad (2.10)$$

3. u^* ne satisfait pas l'équation de continuité. Notons u' et p' les incréments de vitesse et de pression tels que $u^{k+1} = u^* + u'$ et $p^{k+1} = p^k + p'$ avec u^{k+1} et p^{k+1} qui vérifieront $(E - Q)u^{k+1} = f - Gp^{k+1}$ et $Du^{k+1} = 0$. Il vient alors $(E - Q)u' = -Gp'$ qui peut s'écrire : $u' = E^{-1}Qu' - E^{-1}Gp'$. En prenant la divergence de cette équation, on forme une équation sur p' :

$$DE^{-1}Gp' = Du^* + DE^{-1}Qu' \quad (2.11)$$

Le dernier terme $DE^{-1}Qu'$ étant inconnu et normalement nul après convergence des itérations, il est négligé en première approche [Ferziger et Perić, 1996]. L'équation de Poisson se simplifie en :

$$DE^{-1}Gp' = Du^* \quad (2.12)$$

4. u^* et p^k sont enfin corrigées : $u^{k+1} = u^* - E^{-1}Gp'$ et $p^{k+1} = p^k + p'$. u^{k+1} est à divergence nulle mais ne satisfait pas l'équation de quantité de mouvement. Le processus est alors réitéré (Fig. 2.1).

Le critère d'arrêt des itérations non-linéaires pour stopper le calcul ou passer au pas de temps suivant est en général basé sur le résidu des équations discrètes.

On remarque que si les équations de quantité de mouvement sont résolues explicitement dans la première étape alors une itération de l'algorithme SIMPLE correspond à une étape de projection de Chorin.

Pour favoriser la convergence, la vitesse et la pression sont souvent relaxées. La nouvelle vitesse u^* issue de la résolution du système linéaire de l'équation de quantité de mouvement peut, d'après (2.10), se mettre sous la forme :

$$Eu^* = Qu^* + f - Gp^k \quad (2.13)$$

ou encore

$$u^* = u^k + \left[E^{-1}(Qu^* + f - Gp^k) - u^k \right] \quad (2.14)$$

Le terme entre crochets représente la variation de u^* . Pour favoriser la convergence, un coefficient α compris entre 0 et 1 est introduit.

$$u^* = u^k + \alpha \left[E^{-1}(Qu^* + f - Gp^k) - u^k \right] \quad (2.15)$$

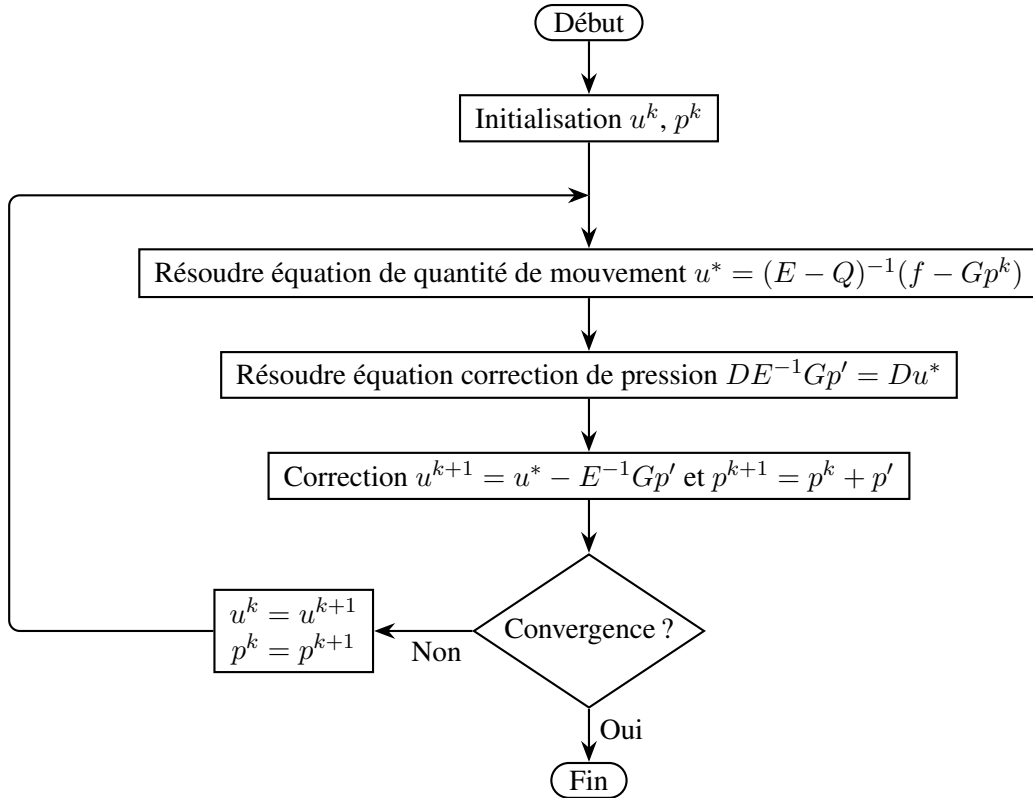


FIGURE 2.1 – Algorithme SIMPLE pour un calcul stationnaire ou un pas de temps d'un calcul instationnaire

soit si on revient à une forme semblable à (2.10) :

$$\left(\frac{E}{\alpha} - Q\right)u^* = f - Gp^k + \frac{1 - \alpha}{\alpha}E^{-1}u^k \quad (2.16)$$

Cette opération de relaxation présente l'avantage de préserver la forme initiale du système linéaire en n'affectant que la partie diagonale et le membre de droite. Pour la pression, la relaxation s'effectue après la résolution de la correction de pression : $p^{k+1} = p^k + \alpha_p p'$. La relaxation favorise la convergence vers un état stationnaire, en particulier sur les maillages de "mauvaise" qualité pour lesquels le calcul peut rapidement diverger.

Le détail et les subtilités de l'algorithme sont repris dans de nombreux ouvrages de CFD [Patankar, 1980, Ferziger et Perić, 1996, Versteeg et Malalasekera, 2007]. Indépendamment de la façon dont sont résolus les systèmes linéaires, l'algorithme SIMPLE est relativement lent à converger, notamment parce que le terme $E^{-1}Qu'$ est négligé dans l'équation de pression mais aussi parce que le terme convectif est linéarisé $\mathbf{u} \otimes \mathbf{u} \approx \mathbf{u}^k \otimes \mathbf{u}^{k+1}$. Pour accélérer la convergence, il est possible de renforcer la diagonale E avec un pseudo-temps comme c'est par exemple le cas dans le code ISIS-CFD. De nombreuses variantes du schéma SIMPLE ont aussi été proposées pour pallier ses défauts : SIMPLER [Patankar, 1980], SIMPLEC [Van Doormaal et Raithby, 1984], ou PISO (Pressure Implicit with Splitting of Operator) [Issa, 1986] que l'on décrit ci-après.

Méthode PISO de Issa, 1986

A l'origine, cet algorithme a été développé pour des écoulements instationnaires compressibles. Il s'adapte cependant avec succès pour les calculs itératifs des problèmes stationnaires. Il est semblable au

SIMPLE mais avec des corrections supplémentaires. Introduisons u'' et p'' les seconds incréments. Alors en manipulant les équations de quantité de mouvement, on aboutit à :

$$Eu'' - Qu' - Qu'' = -Gp'' \quad (2.17)$$

Ou encore

$$u'' = E^{-1}(Qu') - E^{-1}Gp'' + E^{-1}Qu'' \quad (2.18)$$

Le dernier terme $E^{-1}Qu''$ inconnu est négligé de la même façon que $E^{-1}Qu'$ était négligé pour l'algorithme SIMPLE. De plus, la vitesse corrigée doit être à divergence nulle $D(u^* + u' + u'') = 0$ et $D(u^* + u') = 0$ après la première correction. La seconde équation de Poisson se simplifie en :

$$DE^{-1}(Qu') = DE^{-1}Gp'' \quad (2.19)$$

On peut définir de cette façon des incréments successifs. En générale deux corrections suffisent pour relever une amélioration notable de la convergence par rapport au SIMPLE. Il s'agit en fait d'un procédé itératif pour résoudre (2.11) en traitant le dernier terme de façon purement explicite. On retrouve l'algorithme PISO dans de nombreux codes commerciaux comme STAR-CCM+.

En version instationnaire, u est résolue une seule fois et corrigée par deux résolutions d'équation de correction de pression à chaque pas de temps. Le recours aux techniques de relaxation n'est pas nécessaire. Cet algorithme n'étant pas totalement implicite, la taille des pas de temps demeure relativement limitée.

Méthode PIMPLE

L'acronyme PIMPLE est le résultat de la contraction de PISO et SIMPLE. C'est un algorithme par exemple disponible dans le code OpenFOAM. L'idée est de combiner l'approche PISO et SIMPLE pour autoriser de plus grands pas de temps qu'avec l'algorithme PISO pour la simulation d'écoulements instationnaires. Pour cela, à chaque pas de temps, l'écoulement est envisagé dans un état stationnaire vers lequel il doit converger (philosophie du SIMPLE). Le schéma PIMPLE est équivalent à un schéma PISO en itérant sur la résolution de u et p à chaque pas de temps. Des coefficients de relaxation sont souvent appliqués, sauf pour la dernière itération, afin de garantir un champ de vitesse à divergence nulle à l'itération suivante.

2.1.3 Méthodes de compressibilité artificielle

Les équations incompressibles se distinguent des équations compressibles par l'absence de la densité (ou pression) dans l'équation de continuité de sorte que les équations incompressibles ne peuvent être avancées en temps explicitement. Ces deux types de formulation sont mathématiquement bien différentes. Les équations compressibles sont hyperboliques. Elles ont des caractéristiques réelles sur lesquelles se propagent l'information à une vitesse finie. A l'inverse, les équations incompressibles ont un caractère mixte parabolique-elliptique (suivant la nature de l'écoulement, stationnaire ou instationnaire) avec une propagation à vitesse infinie de l'information de pression. L'idée de l'approche par compressibilité artificielle est d'introduire une dérivée partielle en temps de la pression $\partial p / \partial t$ dans l'équation de continuité. Les équations recouvrent ainsi leur caractère hyperbolique. Toutes les méthodes développées pour la formulation compressible (solveur de Riemann, intégration explicite en temps, etc.) sont alors directement transposables. Chorin a proposé une telle approche intégrant une dérivée en temps de la pression [Chorin, 1967] :

$$\frac{1}{\beta} \frac{\partial p}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.20)$$

où β est le paramètre de compressibilité artificielle. Il est homogène au carré d'une vitesse. Le terme "compressibilité artificielle" vient du fait que la pression p et la densité ρ vérifient une loi d'état fictive $p = \rho / \beta$.

Le terme en pression de l'équation (2.20) est un artifice pour résoudre Navier-Stokes mais n'a pas de justification physique. Ainsi l'historique généré par la résolution de cette équation ne correspond en rien à l'écoulement incompressible modélisé. Seul l'état stationnaire final a un sens physique. Le choix du paramètre β est critique. Plus sa valeur est grande, plus on s'approche de l'incompressible mais plus le système d'équations devient raide. Une pseudo-vitesse du son peut être définie par $c = \sqrt{\beta + u^2}$. Choi et Merkle ont confronté les approches de compressibilité artificielle et purement compressible sur des écoulements à bas Mach et mis en évidence un choix approprié de la pseudo-vitesse du son [Choi et Merkle, 1985]. En outre, les pas de temps demeurent souvent rédhitoires et imposent une résolution (semi-)implicite. L'avantage est qu'il est possible d'accélérer la convergence efficacement avec par exemple des méthodes multigrilles géométriques car seul l'état final compte. La méthode de compressibilité artificielle n'est qu'un cas particulier de la méthode de préconditionnement proposée par Turkel [Turkel, 1987] et qui généralise "l'hyperbolisation" des équations incompressibles.

Merkle et Athavale [Merkle, 1987] ont suggéré une version instationnaire en introduisant un pseudo-temps τ :

$$\begin{cases} \frac{1}{\beta} \frac{\partial p}{\partial \tau} + \nabla \cdot (\rho \mathbf{u}) = 0 \\ \frac{\partial \mathbf{u}}{\partial \tau} + \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} + \mathbf{f} \end{cases} \quad (2.21)$$

Ainsi, à chaque pas de temps, les équations sont résolues pour le pseudo-temps jusqu'à convergence :

$$\begin{cases} \frac{\partial p}{\partial \tau} \rightarrow 0 \\ \frac{\partial \mathbf{u}}{\partial \tau} \rightarrow 0 \end{cases} \quad (2.22)$$

Cette approche est bien moins répandue dans les codes industriels que les méthodes découplées implicites SIMPLE, PIMPLE, etc. En hydrodynamique navale, l'outil χ navis développé au CNR-INSEAN à Rome met en œuvre ce type d'approche de manière implicite avec une méthode multigrille [Di Mascio et al., 2007, Di Mascio et al., 2014]. En aérodynamique, le code open-source PyFR initialement dédié aux écoulements compressibles a récemment été étendu aux écoulements incompressibles grâce à une approche pseudo-compressible et une méthode p-multigrille [Loppi et al., 2018].

2.1.4 Bilan des méthodes incompressibles

Les méthodes implicites de type SIMPLE ou PIMPLE ont été largement adoptées dans les codes génériques multiphysiques commerciaux comme STAR-CCM+, Fluent, Flow3D, etc. mais aussi dans les codes dédiés aux applications navales : CFD Ship Iowa (IIHR, Etats-Unis), ISIS-CFD (LHEEA ECN-CNRS, France), ReFRESKO (MARIN, Pays-Bas). Elles répondent à la majeure partie des problèmes d'hydrodynamique navale. Le caractère implicite de ces méthodes rend leur mise en œuvre particulièrement stable pour de grandes valeurs du nombre CFL (> 1), y compris pour des schémas centrés qui sont par essence moins diffusif que les schémas décentrés. Cependant la partie convective reste souvent traitée par des schémas hybrides centrés/décentrés pour prévenir l'apparition d'instabilités et les difficultés de résolution de système linéaire liées à un mauvais conditionnement des matrices.

D'autre part ces méthodes présentent un certain nombre d'incompatibilités avec les caractéristiques de la méthode que nous souhaitons développer comme par exemple l'ordre élevé des schémas numériques.

En effet les schémas d'ordre élevés en espace s'appuient sur des stencils trop longs pour une utilisation pratique avec une approche implicite. La correction différée (deferred correction) est une ruse néanmoins commode pour monter en ordre les flux par exemple. Elle consiste à décomposer le flux en une partie

explicite et une partie implicite. Un schéma d'ordre bas est utilisé pour la partie implicite alors que la différence entre le schéma d'ordre élevé et le schéma d'ordre plus faible est évaluée explicitement :

$$F = F_{implicite}^L + (F_{explicite}^H - F_{explicite}^L) \quad (2.23)$$

Dans un processus itératif, les évaluations implicites et explicites d'ordre bas s'annulent à la convergence $F_{implicite}^L = F_{explicite}^L$ ne laissant finalement que le flux d'ordre élevé. En pratique, la convergence peut être lente et une relaxation nécessaire pour éviter l'apparition d'instabilités numériques amplifiées par le terme d'ordre élevé.

Par ailleurs la scalabilité de ces méthodes implicites reste limitée sur un très grand nombre de processeurs. Elles sont donc difficilement compatibles avec le cadre que nous nous sommes fixés autour de la méthode WCCH qui vise un déploiement efficace sur des architectures de calcul massivement parallèles.

Finalement l'approche qui répond le mieux à nos spécifications est le schéma initial de projection de Chorin dont la simplicité apparente en fait une méthode de choix. De surcroît, un nombre croissant de codes CFD dédiés aux applications marines adoptent les schémas de projection : ComFLOW [Kleefsman et al., 2005], NFA [Sussman et Dommermuth, 2014], REEF3D [Chella et al., 2015], etc. Ces codes démontrent que les méthodes de projection sont hautement compatibles avec des écoulements instationnaires multiphysiques turbulents. Force est de constater que cette méthode âgée d'un demi-siècle répond aux exigences actuelles des calculs haute performance.

Enfin, bien que peu d'études se soient intéressées à la comparaison entre des approches classiques de type SIMPLE et les schémas de projection, les rares travaux sur ce sujet tendent à montrer une meilleure efficacité des méthodes de projection sur des écoulements fortement instationnaires nécessitant de faibles pas de temps de simulation [Hines, 2008]. Naturellement, ce type de schéma est beaucoup moins compétitif que les méthodes implicites pour la simulation d'écoulements stationnaires ou à dynamique lente. Nous présentons toutefois en annexe les résultats d'une implémentation d'un schéma PIMPLE dans le code WCCH (cf. annexe D).

2.2 Méthode de projection de Chorin et ses variantes

Dans cette section nous introduisons les différentes méthodes de projection qui ont été proposées suite aux travaux de Chorin et Témam.

2.2.1 Méthode originelle

La méthode de projection proposée séparément par Chorin [Chorin, 1968] et Témam [Temam, 1969] découple le calcul de la vitesse et de la pression. Malgré son apparente simplicité, la méthode de projection originelle n'est pas exempte de certaines difficultés. Pour les mettre en évidence, écrivons le problème sous la forme semi-discrète :

$$\begin{cases} \frac{1}{\Delta t}(\mathbf{u}^* - \mathbf{u}^n) = \nu \nabla^2 \mathbf{u}^n - \nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) + \mathbf{f}^n \\ \frac{1}{\Delta t}(\mathbf{u}^{n+1} - \mathbf{u}^*) + \nabla p^{n+1} = 0 \\ \nabla \cdot \mathbf{u}^{n+1} = 0 \text{ sur } \Omega, \mathbf{u}^{n+1} = 0 \text{ sur } \partial\Omega \end{cases} \quad (2.24)$$

Ω est le volume fluide sur lequel le système (2.24) est résolu et $\partial\Omega$ désigne la frontière du domaine. Nous considérons ici que des conditions aux limites de Dirichlet sont appliquées sur la vitesse avec une condition de non-pénétration $\mathbf{u}^{n+1} \cdot \mathbf{n} = 0$. Il apparaît alors qu'une condition de Neumann $\nabla p^{n+1} \cdot \mathbf{n} = 0$ est tacitement imposée sur la pression puisque $\mathbf{u}^n \cdot \mathbf{n} = 0$ et $\mathbf{u}^* \cdot \mathbf{n} = 0$ (voir [Guermond et al., 2006]). Cette

condition artificielle et non désirée génère une couche limite numérique qui empêche le schéma de projection d'être pleinement d'ordre 1 pour la pression. Rannacher [Rannacher, 1992] a montré qu'elle n'est que d'ordre $1/2$ en norme L_∞ . Si pour des conditions aux limites périodiques, un tel problème ne se manifeste pas, il peut devenir de taille pour un bon nombre d'applications rencontrées. De nombreux mathématiciens et numériciens se sont penchés sur ce problème pour trouver des schémas d'ordre supérieur et montrer théoriquement leur ordre de convergence. Le choix des conditions aux limites à appliquer fait toujours l'objet de controverses et l'expérience numérique confrontée à la théorie a souvent montré de meilleurs résultats que ceux attendus. Dans la suite, nous exposons quelques unes des variantes du schéma de projection. L'article de Guermond [Guermond et al., 2006] constitue une référence solide et complète pour tout lecteur désireux d'avoir un aperçu plus large des différentes méthodes de projection dans un formalisme mathématique rigoureux.

2.2.2 Déclinaisons

Projection incrémentale Goda, 1979 et Van Kan, 1986

L'idée simple proposée par Goda [Goda, 1979] est d'ajouter le gradient de pression à l'instant t^n dans l'étape de prédiction pour augmenter la précision du schéma. La vitesse \mathbf{u}^* est alors corrigée par le gradient d'un incrément de pression ψ en lieu et place du gradient de pression présent le schéma originel.

$$\begin{cases} \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \nu \nabla^2 \mathbf{u}^n - \nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) - \frac{\nabla p^n}{\rho} + \mathbf{f}^n \\ \nabla \cdot \left(\frac{\Delta t}{\rho} \nabla \psi^{n+1} \right) = \nabla \cdot \mathbf{u}^* \\ \mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla \psi^{n+1} \\ p^{n+1} = p^n + \psi^{n+1} \end{cases} \quad (2.25)$$

Van Kan a ensuite popularisé cette démarche en proposant une projection incrémentale globalement d'ordre 2 avec un schéma de Crank-Nicolson [Van Kan, 1986]. Cependant le problème de couche limite numérique persiste puisque $\nabla p^{n+1} \cdot \mathbf{n} = \nabla p^n \cdot \mathbf{n} = \dots = \nabla p^0 \cdot \mathbf{n}$ sur $\partial\Omega$. Weinan et Liu [Weinan et Liu, 1995] ont montré que la pression est d'ordre 1 en norme L_∞ .

Timmermans, Minev et Van de Vosse, 1996

Timmermans, Minev et Van de Vosse [Timmermans et al., 1996] ont introduit une formulation qualifiée par Guermond de projection incrémentale rotationnelle [Guermond et al., 2006]. Une prise en compte implicite du terme visqueux est nécessaire et le calcul de la pression à l'instant t^{n+1} est légèrement différent du schéma de projection classique. Avec une dérivée temporelle de vitesse estimée par une formule d'ordre 2, ce schéma de projection s'écrit :

$$\begin{cases} \left(\frac{3}{2\Delta t} - \nu \nabla^2 \right) \mathbf{u}^* = \frac{2}{\Delta t} \mathbf{u}^n - \frac{1}{2\Delta t} \mathbf{u}^{n-1} - \nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) - \frac{\nabla p^n}{\rho} + \mathbf{f}^n \\ \nabla \cdot \left(\frac{\Delta t}{\rho} \nabla \psi^{n+1} \right) = \frac{3}{2} \nabla \cdot \mathbf{u}^* \\ \mathbf{u}^{n+1} = \mathbf{u}^* - \frac{2\Delta t}{3\rho} \nabla \psi^{n+1} \\ p^{n+1} = p^n + \psi^{n+1} - \mu \nabla \cdot \mathbf{u}^* \end{cases} \quad (2.26)$$

Le schéma numérique d'intégration en temps n'est pas important ici. En remarquant que $\nabla^2 \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times \nabla \mathbf{u}$, il est possible de montrer que $\nabla p^{n+1} \cdot \mathbf{n} = \frac{\rho}{\Delta t} (-\nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) + \mathbf{f}^n - \nu \nabla \times \nabla \mathbf{u}^{n+1}) \cdot \mathbf{n}$. Cette condition aux limites sur la pression est consistante.

Guermond a généralisé ce type de projection tout en montrant que le schéma populaire de Kim et Moin était équivalent [Kim et Moin, 1985].

Kim et Moin, 1985

Kim et Moin ont proposé en 1985 un schéma de projection d'ordre 2 particulièrement prisé et qui repose sur une discrétisation avec un schéma d'Adams-Bashforth (AB2) du terme convectif combiné à un schéma de Crank-Nicolson pour le terme visqueux.

$$\begin{cases} \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\frac{1}{2}(3\nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) - \nabla \cdot (\mathbf{u}^{n-1} \otimes \mathbf{u}^{n-1})) + \frac{\nu}{2}\nabla^2(\mathbf{u}^* + \mathbf{u}^n) + \mathbf{f}^n \\ \nabla \cdot \left(\frac{\Delta t}{\rho} \nabla \psi^{n+1} \right) = \nabla \cdot \mathbf{u}^* \\ \mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla \psi^{n+1} \\ p^{n+1} = \psi^{n+1} - \Delta t \frac{\nu}{2} \nabla^2 \psi^{n+1} \end{cases} \quad (2.27)$$

Le terme visqueux implicite est résolu efficacement sur grille cartésienne par une factorisation et une méthode ADI (Alternating Direction Implicit) pour appliquer l'algorithme de Thomas (TDMA) dans chaque direction.

Brown, Cortez et Minion, 2001

Brown, Cortez et Minion ont proposé une méthode de projection originale : la méthode de jauge qui implique de résoudre une formulation faible de l'équation de Poisson [Brown et al., 2001]. Cette approche semble répondre aux exigences de précision attendue.

Bell, Colella, Glaz, 1989

Outre le problème des conditions aux limites, la construction d'un schéma robuste pour la simulation d'écoulements à nombre de Reynolds élevé est un autre défi des méthodes de projection. A la fin des années 1980, Bell, Colella et Glaz ont introduit un schéma qui a gagné en popularité au fil des années [Bell et al., 1989]. Le schéma BCG, du nom de ses trois auteurs, permet de traiter le terme convectif à l'ordre 2 en temps et en espace selon une procédure similaire à celle de Godunov pour résoudre les équations d'Euler compressibles. La particularité de ce schéma est d'évaluer le terme convectif à l'instant $t^{n+1/2}$ à l'aide d'un développement en série de Taylor en temps et en espace des vitesses. Les dérivées partielles en temps du développement limité sont remplacées directement par des termes de l'équation de conservation de la quantité de mouvement.

Pour illustrer la procédure BCG, nous considérons le calcul du terme convectif à l'interface de deux cellules de normale x en dimension 2 (Fig. 2.2). Nous supposons que les inconnues de vitesses $\mathbf{u}_{i,j}$ sont localisées aux centres des cellules et que le flux convectif F est évalué sur la face :

$$F = \mathbf{u}_{i+1/2,j}^{n+1/2} u_{i+1/2,j}^{n+1/2} \quad (2.28)$$

où $u_{i+1/2,j}^{n+1/2}$ représente la vitesse advective suivant x et $\mathbf{u}_{i+1/2,j}^{n+1/2}$ est le vecteur vitesse sur la face.

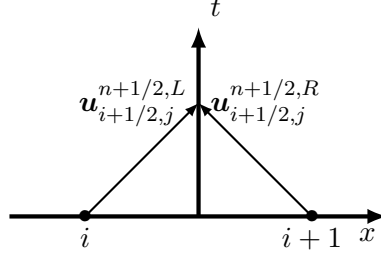


FIGURE 2.2 – Procédure de type Godunov pour le schéma BCG

Un développement en série de Taylor à l'ordre 2 en espace et en temps de la vitesse $\mathbf{u}_{i,j}$ conduit à :

$$\mathbf{u}_{i+1/2,j}^{n+1/2,L} = \mathbf{u}_{i,j}^n + \frac{\Delta x}{2} \frac{\partial \mathbf{u}^n}{\partial x} \Big|_{i,j} + \frac{\Delta t}{2} \frac{\partial \mathbf{u}^n}{\partial t} \Big|_{i,j} + O(\Delta t^2, \Delta x^2) \quad (2.29)$$

La dérivée en temps $\frac{\partial \mathbf{u}^n}{\partial t} \Big|_{i,j}$ peut être remplacée directement par des dérivées spatiales grâce à l'équation de conservation de la quantité de mouvement :

$$\frac{\partial \mathbf{u}^n}{\partial t} \Big|_{i,j} = -u_{i,j}^n \frac{\partial \mathbf{u}^n}{\partial x} \Big|_{i,j} - v_{i,j}^n \frac{\partial \mathbf{u}^n}{\partial y} \Big|_{i,j} - \frac{\nabla p^n}{\rho} \Big|_{i,j} \quad (2.30)$$

Ainsi :

$$\mathbf{u}_{i+1/2,j}^{n+1/2,L} = \mathbf{u}_{i,j}^n + \left[\frac{\Delta x}{2} - \frac{\Delta t}{2} u_{i,j}^n \right] \frac{\partial \mathbf{u}^n}{\partial x} \Big|_{i,j} - \frac{\Delta t}{2} v_{i,j}^n \frac{\partial \mathbf{u}^n}{\partial y} \Big|_{i,j} - \frac{\Delta t}{2} \frac{\nabla p^n}{\rho} \Big|_{i,j} + O(\Delta t^2, \Delta x^2) \quad (2.31)$$

Les dérivées normale $\frac{\partial \mathbf{u}^n}{\partial x} \Big|_{i,j}$ et tangentielle $\frac{\partial \mathbf{u}^n}{\partial y} \Big|_{i,j}$ peuvent être déterminées de plusieurs manières ; avec un schéma centré comme par exemple :

$$\frac{\partial \mathbf{u}^n}{\partial x} \Big|_{i,j} = \frac{\mathbf{u}_{i+1,j}^n - \mathbf{u}_{i-1,j}^n}{2\Delta x} \quad (2.32)$$

ou décentré "upwind" comme par exemple :

$$\frac{\partial \mathbf{u}^n}{\partial y} \Big|_{i,j} = \begin{cases} \frac{\mathbf{u}_{i,j}^n - \mathbf{u}_{i,j-1}^n}{\Delta y} & \text{si } v_{i,j} > 0 \\ \frac{\mathbf{u}_{i,j+1}^n - \mathbf{u}_{i,j}^n}{\Delta y} & \text{si } v_{i,j} < 0 \end{cases} \quad (2.33)$$

De la même façon, un état droit peut être reconstruit :

$$\mathbf{u}_{i+1/2,j}^{n+1/2,R} = \mathbf{u}_{i+1,j}^n - \left[\frac{\Delta x}{2} + \frac{\Delta t}{2} u_{i+1,j}^n \right] \frac{\partial \mathbf{u}^n}{\partial x} \Big|_{i+1,j} - \frac{\Delta t}{2} v_{i+1,j}^n \frac{\partial \mathbf{u}^n}{\partial y} \Big|_{i+1,j} - \frac{\Delta t}{2} \frac{\nabla p^n}{\rho} \Big|_{i+1,j} + O(\Delta t^2, \Delta x^2) \quad (2.34)$$

Un simple upwind permet ensuite de choisir entre l'état gauche ou droit :

$$\mathbf{u}_{i+1/2,j}^{n+1/2} = \begin{cases} \mathbf{u}_{i+1/2,j}^{n+1/2,L} & \text{si } u_{i,j} + u_{i+1,j} > 0 \\ \mathbf{u}_{i+1/2,j}^{n+1/2,R} & \text{si } u_{i,j} + u_{i+1,j} < 0 \\ \frac{1}{2}(\mathbf{u}_{i+1/2,j}^{n+1/2,L} + \mathbf{u}_{i+1/2,j}^{n+1/2,R}) & \text{sinon} \end{cases} \quad (2.35)$$

Bell, Colella et Glaz utilisent une méthode de type éléments finis pour leur schéma de projection avec les variables de pression aux centres des cellules et les vitesses aux nœuds. Malgré tout, cette approche génère un découplage local. Bell, Colella et Howell [Bell et al., 1991] ont étendu cette version pour des variables colocalisées, mais ont démontré la présence d'un découplage pair-impair. Au milieu des années 1990, Almgren et al. ont introduit le concept de projection approchée pour désigner des méthodes de projection robustes s'appliquant à des grilles colocalisées [Almgren et al., 1996].

Almgren, Bell et Szymczak, 1996

En formulation incompressible le problème de découplage pair-impair lors de la formation de l'équation de Poisson sur un maillage colocalisé est clairement identifié. Il apparaît quelle que soit la méthode de résolution (différences finies ou volumes finis) lorsqu'une équation de Poisson doit être résolue, et est directement lié à la disposition des variables. Afin d'illustrer ce problème, nous considérons les équations de Navier-Stokes discrétisées et appliquées à une cellule (i, j) (voir Fig. 2.3). Elles sont réécrites en 2D en considérant que $\Delta x = \Delta y$ et en faisant apparaître la vitesse intermédiaire \mathbf{u}^* :

$$\begin{cases} u_{i,j}^{n+1} = u_{i,j}^* - \frac{\Delta t}{\rho} \frac{p_e - p_o}{\Delta x} \\ v_{i,j}^{n+1} = v_{i,j}^* - \frac{\Delta t}{\rho} \frac{p_n - p_s}{\Delta x} \\ u_e^{n+1} - u_o^{n+1} + v_n^{n+1} - v_s^{n+1} = 0 \end{cases} \quad (2.36)$$

Pour construire l'équation de Poisson à partir de l'équation de conservation de la masse, une première approche peut consister à interpoler les grandeurs aux faces à partir des valeurs moyennes centrées : $p_e = \frac{1}{2}(p_{i+1,j} + p_{i,j})$, $u_e^{n+1} = \frac{1}{2}(u_{i+1,j}^{n+1} + u_{i,j}^{n+1})$.

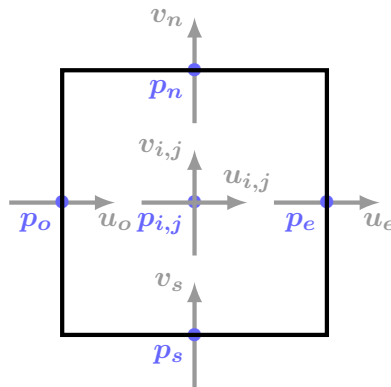


FIGURE 2.3 – Grille colocalisée et valeurs aux faces

L'équation de quantité mouvement s'écrit alors :

$$u_{i,j}^{n+1} = u_{i,j}^* - \frac{\Delta t}{\rho} \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x} \quad (2.37)$$

Avec cette expression, et en remplaçant $u_e^{n+1} = \frac{1}{2}(u_{i+1,j}^{n+1} + u_{i,j}^{n+1})$ (et en faisant de même pour u_o^{n+1} , v_n^{n+1} et v_s^{n+1}) dans l'équation de continuité $u_e^{n+1} - u_o^{n+1} + v_n^{n+1} - v_s^{n+1} = 0$, il vient :

$$p_{i+2,j} + p_{i-2,j} + p_{i,j+2} + p_{i,j-2} - 4p_{i,j} = \rho \frac{2\Delta x}{\Delta t} (u_{i+1,j}^* - u_{i,j}^* + v_{i+1,j}^* - v_{i,j}^*) \quad (2.38)$$

Ainsi la pression est résolue sur quatre grilles indépendantes en 2D. Ce découplage est donc une source d'oscillations et d'instabilités (voir Fig. (2.4)).

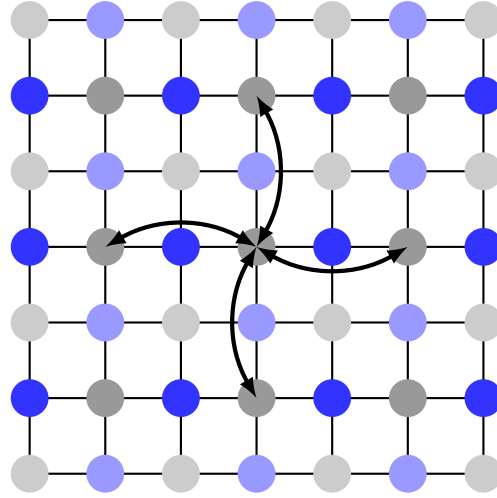


FIGURE 2.4 – Illustration du découplage pair-impair sur grille colocalisée

Une solution pour remédier à ce problème consiste à interpoler la vitesse intermédiaire sur les faces au lieu de la vitesse finale. De la sorte, c'est la vitesse aux faces qui est rendue à divergence nulle et non la vitesse localisée au centre de la cellule $u_e^{n+1} = u_e^* - \frac{\Delta t}{\rho} \frac{p_{i+1,j} - p_{i-1,j}}{\Delta x}$ avec $u_e^* = \frac{1}{2}(u_{i+1,j}^* + u_{i,j}^*)$. En injectant ces expressions dans l'équation de conservation de la masse, on obtient :

$$p_{i+1,j} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - 4p_{i,j} = \rho \frac{2\Delta x}{\Delta t} (u_{i+1,j}^* - u_{i,j}^* + v_{i+1,j}^* - v_{i,j}^*) \quad (2.39)$$

Ainsi le découplage pair-impair est supprimé, mais la condition de divergence nulle exacte pour les vitesses au centre n'est plus respectée. Almgren, Bell et Szymczak [Almgren et al., 1996] ont montré que relaxer cette condition en projetant les vitesses des faces et non les vitesses situées au centre ne se révélait pas si problématique que cela en pratique. Ils ont désigné ce type d'approche sous le terme de *projection approchée*. Ce terme renvoie au fait que l'opérateur P de projection de la vitesse intermédiaire n'est pas idempotent c'est-à-dire $P(P(\mathbf{u})) \neq P(\mathbf{u})$. Si la projection approchée permet d'éviter les découplages, elle peut néanmoins être source d'instabilités numériques, notamment sur des simulations longues. Une analyse mathématique poussée sur la projection approchée et la méthode BCG a été réalisée par Almgren [Almgren et al., 2000] et Rider [Rider, 1995]. Ce dernier propose notamment le recours à des filtres. Par ailleurs, comme Bell et al. [Bell et al., 1991], Almgren et al. [Almgren et al., 1996] préconisent de projeter la vitesse reconstruite $\mathbf{u}_{i+1/2,j}^{n+1/2}$ avant le calcul du terme convectif pour la rendre à divergence nulle et renforcer la robustesse du schéma. Ainsi, deux projections sont effectuées par pas de temps : une première exacte pour projeter la vitesse reconstruite aux faces (à l'instant $t^{n+1/2}$ avec la procédure de Godunov) et une seconde, approchée, pour projeter les vitesses des faces obtenues après l'avance en temps (à l'instant t^{n+1}).

Martin et Collela, 2000

De nombreuses variantes pour le schéma BCG ont été proposées. Notamment, Martin et Collela ont proposé une amélioration du calcul du terme advectif [Martin et Collela, 2000] sous la forme :

$$\mathbf{u}_{i+1/2,j}^{n+1/2,L} = \mathbf{u}_{i,j}^n + \min \left[\frac{1}{2} \left(1 - u_{i,j}^n \frac{\Delta t}{\Delta x} \right), \frac{1}{2} \right] \Delta x \frac{\partial \mathbf{u}^n}{\partial x} \Big|_{i,j} - \frac{\Delta t}{2} v_{i,j}^n \frac{\partial \mathbf{u}^n}{\partial y} \Big|_{i,j} \quad (2.40)$$

Popinet, 2003

Le code open-source Gerris constitue une bonne référence intégrant ce concept de projection approchée avec un schéma BCG dans un code AMR diphasique avec frontière immergée (cut-cells) [Popinet, 2003a].

Minion, 2004

Aujourd'hui les méthodes de projection approchée d'ordre 2 sur grille colocalisée ont ainsi atteint un certain niveau de maturité. Toutefois, une montée en ordre semble difficile car la méthode BCG est strictement d'ordre 2 en espace et en temps. Des avancées récentes ont été réalisées pour proposer des schémas d'ordre plus élevé et compatibles avec les contraintes numériques de la projection. Minion a introduit en 2004 une méthode SISDC (Semi-Implicit Spectral Deferred Correction) [Minion, 2004, Minion et al., 2003] basée sur les méthodes SDC (Spectral Deferred Correction). Une méthode d'ordre 4 en espace et en temps a été mise au point par Kadioglu [Kadioglu et al., 2008] et appliquée à des écoulements turbulents par Almgren et al. [Almgren et al., 2013a]. Si cette méthode présente l'avantage de pouvoir être généralisée à n'importe quel ordre temporel, elle nécessite de nombreuses projections intermédiaires des vitesses reconstruites aux faces (jusqu'à 9 résolutions d'équation de Poisson par pas de temps pour un schéma d'ordre 4 en temps). Enfin, les extensions sur des grilles raffinées localement n'en sont qu'à leur début [Zhang, 2014].

2.2.3 Schéma de projection finalement retenu

Le schéma de projection proposé initialement par Chorin et Témam a connu de nombreux développements au cours de ces cinq dernières décennies. Longtemps source d'incompréhension, le problème de couche limite numérique lié aux conditions aux limites est maintenant clairement identifié. La quête d'un schéma rigoureusement d'ordre 2 continue d'éveiller la curiosité de la communauté des mathématiciens. L'expérience numérique a souvent montré que les schémas pouvaient donner de meilleurs résultats que ceux attendus théoriquement. Les schémas de type BCG avec projection approchée comme celui de Almgren et al. sont maintenant courants et sont compatibles avec la majorité des exigences souhaitées dans nos travaux de recherche. Cette approche séduisante a été abondamment validée et a démontré sa capacité à traiter des écoulements diphasiques sur grille AMR avec des frontières immergées. De plus, la disposition colocalisée des variables dans WCCH minimise les modifications algorithmiques à apporter pour passer le solveur en formulation incompressible. Néanmoins, cette approche est difficilement compatible avec l'ordre élevé des schémas spatiaux qui est une caractéristique attendue du code.

Par conséquent un schéma de projection basique de Chorin sur un maillage MAC avec la pression située aux centres des cellules et les vitesses aux faces a été privilégié, laissant ainsi la possibilité de recourir au schéma WENO5 pour les reconstructions aux faces. Cette approche facilitera la comparaison entre les formulations incompressible et faiblement-compressible puisque les mêmes schémas numériques seront employés dans la mesure du possible.

Dans la suite nous présentons les schémas de discrétisation et leur implémentation dans le code pour une grille cartésienne uniforme.

2.3 Traitements numériques

La modification de l'emplacement des variables par rapport à l'approche faiblement-compressible impose de repenser les procédures de reconstruction et de calcul de gradient. Il n'y a plus une seule grille de maillage mais trois en 2D (quatre en 3D) : une pour chaque inconnue. Restant au centre, la pression est toujours associée au maillage physique de base. En revanche les vitesses se retrouvent associées à des maillageaux fictifs définis de sorte que les centres des faces du maillage physique coïncident avec le barycentre de chaque cellule duale (Fig. 2.5 où les cellules duales sont représentées par des tirets). L'algorithmique pour les maillages MAC est donc globalement plus complexe que pour celle pour les maillages de variables colocalisées. Enfin les traitements numériques d'un schéma de projection avec des reconstructions WENO5 sont bien explicités dans [Couderc, 2007, Zuzio, 2010].

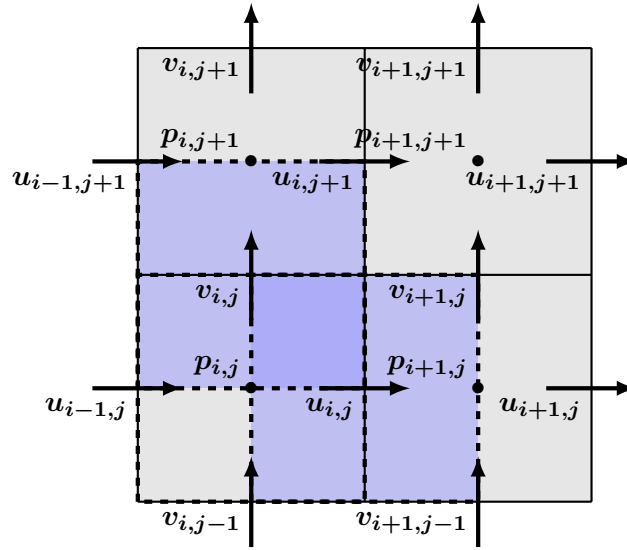


FIGURE 2.5 – Maillage MAC avec variables entrelacées et cellules duales (délimitées par des tirets)

2.3.1 Calcul de la vitesse prédite

Calcul du terme convectif

La procédure des reconstructions WENO en formulation incompressible ne changera pas par rapport à la procédure de balayage multidirectionnel employée en faiblement-compressible même si l'emplacement des variables est différent. Les maillages duaux sont certes décalés par rapport au maillage physique mais les positions relatives des inconnues de vitesse restent les mêmes.

En revanche le calcul des flux convectifs sera différent. Prenons l'exemple de l'inconnue de vitesse suivant x : u (voir Fig. 2.6). Le calcul du flux dans la direction x , noté F_{C_u} , entre les cellules duales de (i, j) et de $(i + 1, j)$ reste assez simple. Connaissant les reconstructions de u à gauche et à droite de l'interface, un décentrement basique est appliqué :

$$F_{C_u} = \begin{cases} (u_{i+1/2,j}^L)^2 & \text{si } u_{adv} > 0 \\ (u_{i+1/2,j}^R)^2 & \text{si } u_{adv} < 0 \\ \frac{1}{4}(u_{i+1/2,j}^L + u_{i+1/2,j}^R)^2 & \text{sinon} \end{cases} \quad u_{adv} = \frac{1}{2}(u_{i+1/2,j}^L + u_{i+1/2,j}^R) \quad (2.41)$$

On note que la vitesse advective u_{adv} peut-être calculée "au vol" au moment du calcul du flux entre les cellules duales de i, j et de $(i + 1, j)$.

Pour la direction y qui est tangentielle à la direction de la variable u , le calcul du flux entre les cellules duales de (i, j) et de $(i, j + 1)$ se fait de la façon suivante :

$$G_{c_u} = \begin{cases} v_{adv} u_{i,j+1/2}^L & \text{si } v_{adv} > 0 \\ v_{adv} u_{i,j+1/2}^R & \text{si } v_{adv} < 0 \\ \frac{1}{2} v_{adv} (u_{i,j+1/2}^L + u_{i,j+1/2}^R) & \text{sinon} \end{cases} \quad v_{adv} = \frac{1}{2} (v_{i+1/2,j}^L + v_{i+1/2,j}^R) \quad (2.42)$$

D'un point de vue algorithmique v_{adv} ne peut être calculée lors du calcul du flux puisqu'en pratique $v_{i+1/2,j}^R$ est stockée par la cellule duale associée à $v_{i+1,j}$ qui n'est pas impliquée dans le calcul du flux. Ainsi, les vitesses d'advection dans les directions tangentielles sont calculées et enregistrées lors d'un balayage des cellules juste après les reconstructions WENO et avant le calcul des flux convectifs. Cette étape représente un surcoût algorithmique et de stockage mémoire pour la formulation incompressible par rapport à la formulation faiblement-compressible.

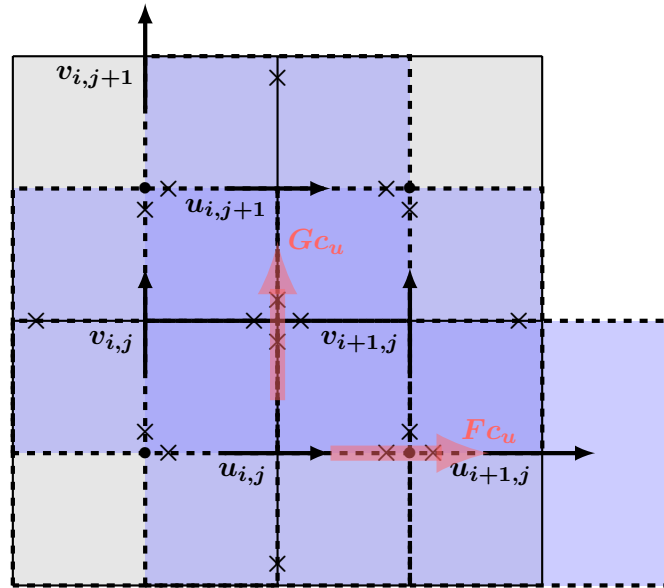


FIGURE 2.6 – Emplacement des reconstructions WENO \times et calculs de flux \rightarrow (sur maillage MAC)

Calcul du terme visqueux

Le calcul des termes visqueux est analogue à celui effectué pour la formulation faiblement-compressible. En reprenant la configuration ci-dessus, le flux visqueux entre les cellules duales de i, j et de $i + 1, j$ est :

$$Fv_u = \nu \frac{u_{i+1,j} - u_{i,j}}{\Delta x} \quad (2.43)$$

Entre les cellules duales de i, j et $i, j + 1$:

$$Gv_u = \nu \frac{u_{i,j+1} - u_{i,j}}{\Delta y} \quad (2.44)$$

2.3.2 Résolution de l'équation de Poisson

Définition des opérateurs gradient et divergence

Outre le calcul du terme convectif, la résolution de l'équation de Poisson mérite également une attention particulière. En effet la détermination de la pression dans l'écoulement passe par l'inversion d'un système

linéaire. De la rapidité de la résolution de l'équation de Poisson dépend l'efficacité globale du solveur. Rappelons d'abord l'équation :

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla p^{n+1} \right) = \nabla \cdot \mathbf{u}^* \quad (2.45)$$

Les opérateurs gradient ∇ et divergence $\nabla \cdot$ doivent être définis. Sur maillage MAC, la discrétisation à l'ordre 2 des opérateurs gradient ∇ et divergence $\nabla \cdot$ est immédiate. Prenons une cellule (i, j) en 2D avec les maillages duaux associés (Fig. 2.5).

L'opérateur divergence s'applique aux variables stockées aux faces (vitesse ou gradient de pression). Une formulation en différences finies à l'ordre 2 s'écrit :

$$\nabla \cdot \mathbf{u} \approx \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + \frac{v_{i,j} - v_{i,j-1}}{\Delta y} \quad (2.46)$$

En volumes finis, elle s'écrit plutôt :

$$\iint \nabla \cdot \mathbf{u} dx dy \approx u_{i,j} \Delta y - u_{i-1,j} \Delta y + v_{i,j} \Delta x - v_{i-1,j} \Delta x \quad (2.47)$$

Sur grille cartésienne uniforme, ces deux formulations sont strictement identiques et conservatives.

Les gradients sont évalués aux faces des cellules à partir de valeurs centrées comme la pression. Par exemple pour une face de vecteur normal x , une approximation centrée à l'ordre 2 donne :

$$\frac{\partial p}{\partial x} \approx \frac{p_{i+1,j} - p_{i,j}}{\Delta x} \quad (2.48)$$

Finalement, l'équation de Poisson est globalement discrétisée à l'ordre 2. En se limitant à cet ordre de convergence, un bon ratio précision/coût CPU de résolution est conservé. De rapides tests sur une équation de Poisson discrétisée à l'ordre 4 n'ont pas mis en évidence un gain spectaculaire sur la précision numérique.

Une fois la discrétisation des opérateurs établie, se pose la question de la façon de résoudre le système linéaire. Toute résolution directe, avec par exemple la méthode du pivot de Gauss, est à proscrire compte-tenu de la taille du système d'équations. La complexité algorithmique d'une telle méthode est en $O(N^3)$, où N désigne le nombre d'inconnues total (jusqu'à plusieurs millions dans nos applications). Autrement dit le nombre d'étapes de calcul avant d'atteindre le résultat est directement proportionnel à N^3 .

La matrice de l'équation de Poisson est une grande matrice creuse. Sur une grille cartésienne, chaque ligne de la matrice possède seulement 5 éléments non nuls en 2D et 7 en 3D (hors conditions aux limites). Des méthodes itératives permettent d'approcher la solution d'un tel système linéaire en un nombre d'opérations beaucoup plus acceptable que la résolution directe. Dans ce qui suit, nous introduisons quelques généralités sur les méthodes itératives pour éclairer les choix qui ont été opérés quant à la sélection d'un algorithme de résolution de l'équation de Poisson.

Principe de la résolution itérative

Considérons un système linéaire de la forme

$$Ax = b \quad (2.49)$$

Les méthodes itératives utilisent cette équation pour améliorer l'estimation x^n de la solution \bar{x} de (2.49) après n itérations. Définissons un schéma itératif en prenant une décomposition arbitraire de la matrice à inverser $A = P - Q$:

$$Px^{n+1} = Qx^n + b \quad (2.50)$$

A convergence $x^{n+1} = x^n = \bar{x}$ et l'équation (2.50) est équivalente au système linéaire (2.49). Bien qu'arbitraire, le choix de P est crucial, de sorte que le système (2.50) soit facilement résoluble. L'écart entre l'itéré x^n et la solution exacte du problème \bar{x} vaut :

$$\epsilon^n = x^n - \bar{x} \quad (2.51)$$

En prémultipliant par P l'équation ci-dessus, il vient :

$$\begin{aligned} P\epsilon^n &= Px^n - P\bar{x} \\ &= Qx^{n-1} + b - P\bar{x} \\ &= Qx^{n-1} - Q\bar{x} \\ &= Q\epsilon^{n-1} \end{aligned} \quad (2.52)$$

Récursivement on obtient :

$$\epsilon^n = (P^{-1}Q)^n \epsilon^0 = (I - P^{-1}A)^n \epsilon^0 = G^n \epsilon^0 \quad (2.53)$$

G est la matrice d'amplification de l'erreur. Ses valeurs propres λ_k et ses vecteurs propres v_k ont un rôle déterminant. Par définition des valeurs propres et des vecteurs propres :

$$Gv_k = \lambda_k v_k \text{ pour } k = 1, \dots, N \quad (2.54)$$

Nous supposons que les valeurs propres sont rangées par ordre croissant :

$$|\lambda_1| < |\lambda_2| < \dots < |\lambda_N| \quad (2.55)$$

Les vecteurs v_k forment une base de \mathbb{R}^N . L'erreur initiale ϵ^0 peut donc se décomposer dans cette base sous forme d'une combinaison linéaire :

$$\epsilon^0 = \sum_{k=1}^N \alpha_k v_k \quad (2.56)$$

α_k étant les coefficients de pondération. Ainsi

$$\epsilon^1 = \sum_{k=1}^N \alpha_k Gv_k = \sum_{k=1}^N \alpha_k \lambda_k v_k \quad (2.57)$$

Récursivement :

$$\epsilon^n = \sum_{k=1}^N \alpha_k (\lambda_k)^n v_k \quad (2.58)$$

La méthode itérative est convergente, i.e. $\epsilon^n \rightarrow 0$, si le rayon spectral de G noté $\rho(G)$ est strictement inférieur à 1 :

$$\rho(G) = \max_{1 \leq k \leq N} |\lambda_k| = |\lambda_N| < 1 \quad (2.59)$$

Après un nombre d'itérations suffisamment grand, l'erreur est dominée par λ_N puisque ce terme est le plus lent à converger :

$$\epsilon^n \sim \alpha_N (\lambda_N)^n v_N \quad (2.60)$$

Formulé autrement, les erreurs associées aux grandes longueurs d'onde (basses fréquences) sont les plus lentes à s'amortir et à être dissipées. Ainsi, plus le rayon spectral de G , $\rho(G)$, est proche de 1, plus la méthode convergera lentement. Le choix de P n'est donc pas anodin. Plus P est proche de la matrice A , plus le rayon spectral de G tend vers 0 améliorant ainsi l'efficacité de la méthode.

Pour illustrer cet aspect, nous présentons quelques méthodes itératives courantes.

Méthodes itératives basiques

Nous considérons l'équation de Poisson suivante :

$$\nabla^2 p = f \quad (2.61)$$

sur un domaine 1D de longueur L . Il est discrétisé en N points discrets avec un pas d'espace Δx . Des conditions homogènes de Dirichlet sont appliquées à p aux frontières du domaine. Avec des schémas centrés d'ordre 2 pour le calcul du gradient et de la divergence, le système discret s'écrit :

$$\begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{N-1} \\ p_N \end{bmatrix} = \Delta x^2 \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix} \quad (2.62)$$

Les valeurs propres de la matrice du Laplacien peuvent s'écrire :

$$\sigma_k = 2 \left(1 - \cos \left(\frac{k\pi}{N} \right) \right), k = 1, \dots, N \quad (2.63)$$

La méthode de Jacobi prend comme matrice P le choix le plus élémentaire : la diagonale de A . Q correspond alors à la partie extra-diagonale. En appliquant cette méthode à (2.61) :

$$p_i^{n+1} = -\frac{1}{2}(f_i \Delta x^2 - (p_{i+1}^n + p_{i-1}^n)) \quad (2.64)$$

Les valeurs propres de P étant égales à -2 , on en déduit les valeurs propres de $G = I - P^{-1}A$:

$$\lambda_k = 1 - \frac{\sigma_k}{2} = \cos \left(\frac{k\pi}{N} \right), k = 1, \dots, N \quad (2.65)$$

Le rayon spectral vaut donc :

$$\rho(G) = \cos \left(\frac{\pi}{N} \right) \approx 1 - \frac{\pi^2}{2N^2} \quad (2.66)$$

pour N suffisamment grand. On peut ainsi constater que plus on augmente le nombre de points de discrétisation N , plus le rayon spectral sera proche de 1 et plus la méthode de Jacobi va être lente à converger. Cela illustre le constat suivant : la résolution d'une équation de la forme (2.61) converge plus rapidement sur un maillage grossier que sur un maillage fin.

Une autre méthode itérative classique est celle de Gauss-Seidel. Dans ce cas P est égale à la partie triangulaire inférieure de A . Q est prise égale à la partie triangulaire supérieure stricte. Par rapport à la méthode de Jacobi, elle exploite le fait que, pour un volume de contrôle i , p_{i-1} a déjà été calculé à l'instant t^{n+1} . Elle utilise donc cette valeur plutôt que celle de t^n :

$$p_i^{n+1} = -\frac{1}{2}(f_i \Delta x^2 - \sum_{j=1}^{i-1} \delta_{ij} p_j^{n+1} - \sum_{j=i+1}^N \delta_{ij} p_j^n) \quad (2.67)$$

δ_{ij} est le symbole de Kronecker : $\delta_{ij} = 1$ si $i = j$, 0 sinon. Une meilleure convergence est attendue. Young [Young, 2014] a montré que le rayon spectral de la matrice d'amplification est :

$$\rho(G) = \cos \left(\frac{\pi}{N} \right)^2 \approx 1 - \frac{\pi^2}{N^2} \quad (2.68)$$

Cette méthode converge donc deux fois plus vite que la méthode de Jacobi. Mais, comme pour la méthode de Jacobi, sa complexité algorithmique est en $O(N^2)$ ce qui en fait une méthode peu efficace pour les problèmes de grande dimension.

Méthodes de sur-relaxation

Les méthodes sur-relaxation permettent d'accélérer la convergence et modifient les itérations de la manière suivante :

$$x^{n+1} = (1 - \omega)x^n + \omega P^{-1}(Qx^n + b) \quad (2.69)$$

ω est le paramètre de relaxation. Pour accélérer la convergence, il faut choisir ω de sorte que :

$$1 < \omega < 2 \quad (2.70)$$

Appliquée à la méthode de Gauss-Seidel, la sur-relaxation est souvent nommée SOR (Successive Overrelaxation method). Une valeur optimale de ω peut être calculée pour minimiser le rayon spectral de la matrice d'amplification :

$$\omega_{opt} = 2 - \frac{2\pi}{N} \quad (2.71)$$

Cette méthode est facile à mettre en œuvre et permet de réduire la complexité algorithmique de la méthode de Gauss-Seidel à $O(N^{3/2})$.

Méthodes de Krylov

Une grande partie des méthodes itératives utilisées de nos jours en calcul scientifique appartiennent à une autre grande classe de résolution itérative : les méthodes de Krylov. Krylov a défini un espace vectoriel d'ordre n et noté K_n et dont une base est construite par les vecteurs :

$$r^0, Ar^0, \dots, A^{n-1}r^0 \quad (2.72)$$

Avec r^0 le résidu initial $r^0 = b - Ax^0$, x^0 l'estimation initiale du vecteur x . Les méthodes de Krylov s'appuient sur ces espaces pour la résolution itérative de systèmes linéaires. Elles consistent à chercher les itérés x^n dans l'espace $x^0 + K_n$:

$$x^n = x^0 + V_n z^n \quad (2.73)$$

V_n est une matrice dont les colonnes sont des vecteurs qui définissent une base de K_n . Les colonnes de V_n et les n composantes de z^n sont construites de façon incrémentale suivant un procédé d'orthogonalisation de type Gram-Schmidt. Le choix du type d'orthogonalité du vecteur résidu r^n permet de définir la méthode :

$$(r^n)^T V_n = 0 \quad (2.74)$$

pour les méthodes OR (Orthogonal Residual) comme le gradient conjugué, ou encore

$$(r^n)^T A V_n = 0 \quad (2.75)$$

pour les méthodes de résidu minimal comme par exemple GMRES [Saad et Schultz, 1986]. La méthode du gradient conjugué (CG) se limite à des matrices symétriques définies positives. La méthode du gradient biconjugué (BiCG) est une généralisation pour les matrices non symétriques. Des versions plus stables et plus robustes ont été développées : CGS, CGSTAB, BiCGSTAB, etc.

Les méthodes de Krylov convergent en un nombre fini d'itérations au plus égal à la dimension N du problème. En pratique N peut être très grand mais ces méthodes fournissent une bonne approximation en beaucoup moins d'itérations. Néanmoins ces méthodes ont quelques inconvénients si elles sont utilisées

en l'état. Par exemple la méthode de gradient conjugué dépend du conditionnement de la matrice $\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$. Plus le conditionnement est élevé, plus la convergence est lente. Pour la méthode GMRES, le coût des itérations croît en $O(n^2)$, avec n le nombre d'itérations.

Ainsi les méthodes de Krylov sont souvent combinées à des méthodes de préconditionnement ou d'accélération multigrille.

Préconditionnement

Le principe d'une méthode de préconditionnement est de remplacer le système linéaire à résoudre

$$Ax = b \quad (2.76)$$

par

$$C^{-1}Ax = C^{-1}b \quad (2.77)$$

afin de réduire le nombre d'itérations de la méthode de résolution. Le choix de C est encore une fois arbitraire mais doit permettre de mieux conditionner le système. Idéalement $C = A$.

Un préconditionnement souvent employé est celui basé sur la factorisation de Cholesky. Cette dernière décompose toute matrice en un produit d'une matrice triangulaire inférieure et une matrice triangulaire supérieure.

$$A = LU \text{ ou } A = LL^T \text{ pour des matrices symétriques} \quad (2.78)$$

Une décomposition incomplète de Cholesky (Incomplete Cholesky) pour une matrice symétrique est donnée par $\tilde{L}\tilde{L}^T$ où \tilde{L} désigne une matrice creuse égale à L mais avec des 0 partout où A contient des 0. De la sorte, la factorisation n'est pas exacte, $\tilde{L}\tilde{L}^T \approx LL^T$. Cette décomposition fournit un bon préconditionneur pour la méthode de gradient conjugué. Une version asymétrique, appelée factorisation LU incomplète (ILU) existe également.

Une autre méthode de décomposition similaire et employée pour la résolution de systèmes linéaires en CFD est la méthode SIP (Strongly Implicit Procedure) de Stone [Stone, 1968].

Multigrille géométrique

Comme nous l'avons exposé précédemment, les méthodes de relaxation classiques (Jacobi, Gauss-Seidel,...) sont peu efficaces lors de la résolution de grands systèmes linéaires du fait de leur faible efficacité à amortir les basses fréquences. Ce problème s'accroît avec le nombre d'inconnues du problème. Les méthodes de Krylov combinées au principe du préconditionnement donnent de meilleurs résultats de convergence.

Les méthodes multigrilles sont d'une efficacité et d'une performance théorique encore supérieure. Leur principe repose sur l'analyse modale effectuée qualitativement précédemment. Elles consistent à effectuer quelques itérations de résolution avec une méthode classique (Jacobi ou Gauss-Seidel par exemple) sur un maillage donné afin d'éliminer les hautes fréquences. Puis les basses fréquences sont transférées sur des maillages de plus en plus grossiers et pour lesquelles elles se retrouveront dans le spectre des hautes fréquences. Progressivement ces basses fréquences (devenues hautes fréquences par rapport à la taille de maille) sont amorties. L'information est ensuite retransférée vers les grilles les plus fines.

La méthode de correction de grille grossière (Coarse Grid Correction) définit le processus de base de toute méthode multigrille linéaire. Définissons une grille fine qui a des cellules de taille Δh et une grille grossière avec des cellules de taille Δ_H . L'indice h fait référence au maillage fin par opposition à l'indice H qui désigne le maillage grossier. Pour des maillages cartésiens structurés, il est facile de définir un maillage grossier en fusionnant par exemple 2 (en 1D) ou 4 (en 2D) cellules du maillage fin. Les étapes d'une correction de grille grossière sont :

- Tout d'abord quelques itérations de relaxation sont effectuées sur la grille fine. La méthode itérative utilisée (Jacobi ou autre) est désignée sous le terme de lisseur (smoother). On parle également de relaxation dans le sens où le but est d'éliminer des hautes fréquences plus que d'atteindre une précision souhaitée. Un résidu R_h^n peut alors être calculé à partir de la formule :

$$R_h^n = Ax^n - b = (P_h - Q_h)x_h^n - b \quad (2.79)$$

De plus comme $P_h x_h^{n+1} = Q_h x_h^n + b$, le résidu vérifie le système linéaire suivant :

$$P_h \Delta x_h^n = -R_h^n \quad (2.80)$$

Avec $\Delta x_h^n = x_h^{n+1} - x_h^n$. C'est le résidu qui est transféré vers des grilles plus grossières.

- Des opérateurs d'interpolation I_h^H doivent ensuite permettre de faire passer l'information sur la grille grossière :

$$R_H^n = I_h^H R_h^n \quad (2.81)$$

R_H^n est le résidu interpolé sur la grille grossière. Cette opération d'interpolation de la grille fine à la grille grossière est appelée restriction. L'équation du résidu est résolue (par une méthode directe ou itérative) :

$$P_H \Delta x_H^n = -R_H^n \quad (2.82)$$

- La solution Δx_H^n est ensuite remontée sur la grille fine à l'aide d'une nouvelle interpolation :

$$\Delta x_h^n = I_H^h \Delta x_H^n \quad (2.83)$$

I_H^h est appelé opérateur de prolongation.

- L'estimation initiale de x sur la grille la plus fine peut alors être corrigée :

$$x_h^{n+1} = x_h^n + \Delta x_h^n \quad (2.84)$$

Un nouveau lissage est ensuite effectué.

Si on note G_h et G_H les matrices d'amplification du maillage fin et du maillage grossier, on peut montrer que la matrice d'amplification du processus complet s'écrit G_{MG} :

$$G_{MG} = G_h^{n_h} (I - I_H^h P_H^{-1} I_h^H P_h) G_H^{n_H} \quad (2.85)$$

n_H et n_h sont respectivement le nombre de relaxations effectuées sur les maillages grossier et fin. Cette expression montre que la correction de grille grossière s'apparente à un préconditionnement du système linéaire. Il ne s'agit là que d'un exemple d'illustration mais ces cycles CGC peuvent être répétés récursivement de la grille la plus fine à la grille la plus grossière et inversement. On parle alors de méthode multigrille (Fig. 2.7). La qualité et la performance de la méthode multigrille vont dépendre de l'efficacité du lisseur, du nombre de relaxations, etc.

A l'origine, les approches multigrilles ont été développées au milieu des années 1960 pour les équations elliptiques, comme l'équation de Poisson, puis systématisées à d'autres types d'équations, par Brandt notamment [Brandt, 1977]. Elles ont connu un intérêt grandissant dans les codes CFD. Pour plus de détails sur les méthodes multigrilles, on pourra se référer aux ouvrages de Hackbusch [Hackbusch, 2013], Wesseling [Wesseling, 1995] ou Briggs et al. [Briggs et al., 2000].

En plus d'accélérer la résolution du système linéaire, les méthodes multigrilles ont la propriété remarquable de converger à une vitesse indépendante de la taille du maillage. Ainsi la complexité algorithmique est de l'ordre de grandeur du nombre d'opérations effectuées par les relaxations, c'est-à-dire $O(N)$. Cette caractéristique en fait une méthode de choix pour la conception d'un solveur performant sur une architecture parallèle.

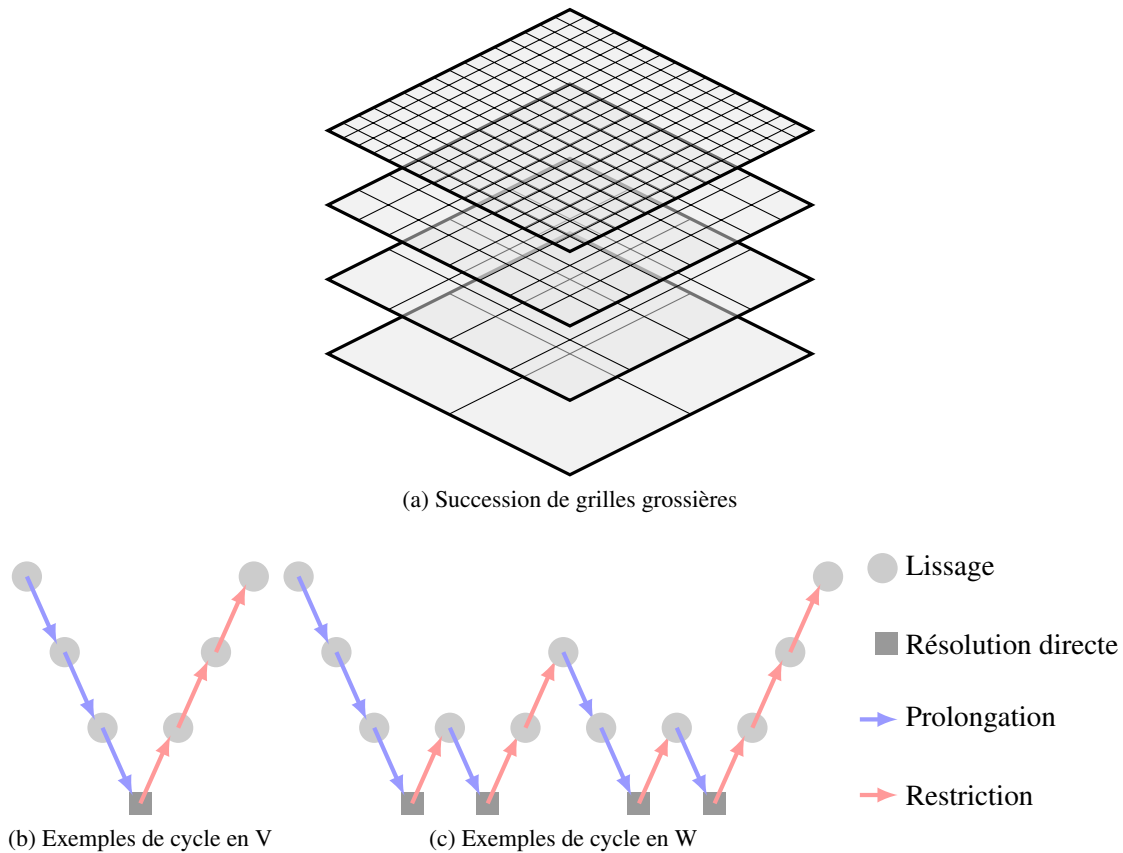


FIGURE 2.7 – Principe des méthodes multigrilles

De plus la méthode multigrille pour les problèmes elliptiques est compatible avec le raffinement local de maillage avec les méthodes MLAT (Multi Level Adaptive Technique) [McCormick et Thomas, 1986] et FAC (Fast Adaptive Composite grid solver) [Bai et Brandt, 1987].

Néanmoins l'implémentation d'une méthode multigrille parallèle efficace n'est pas aisée et le réglage de certains paramètres des cycles reste critique.

Multigrille algébrique

Les méthodes multigrilles géométriques se prêtent bien aux maillages cartésiens structurés pour lesquels il est aisé de définir des maillages de plus en plus grossiers par agglomération de cellules. Les méthodes multigrilles algébriques (Algebraic Multigrid ou AMG en anglais) sont nées de la volonté de généraliser ce type d'approche à des maillages déstructurés et des matrices quelconques [Ruge et Stüben, 1987]. Elles forment des agrégats pour définir des grilles "grossières" fictives, uniquement à partir de l'analyse de la matrice du système linéaire à résoudre. Les agrégats sont construits de manière adaptative pour respecter les critères de convergence requis par l'utilisateur. Un des grands avantages de ces méthodes est qu'elles s'utilisent en "boîte noire". Leur utilisation est combinée avec une résolution itérative classique comme le gradient conjugué ou GMRES. Le développement et l'amélioration des méthodes multigrilles algébriques pour atteindre et égaler les performances théoriques des méthodes multigrilles géométriques est un thème actif de recherche. L'enjeu est d'obtenir une complexité algorithmique proche de $O(N)$.

De nombreuses bibliothèques implémentent le multigrille algébrique. La bibliothèque HYPRE développée par le Lawrence Livermore National Laboratory est une bibliothèque qui intègre des préconditionneurs et des méthodes itératives pour la résolution de systèmes linéaires sur des infrastructures massivement parallèles [Falgot et

Yang, 2002]. BoomerAMG est une librairie de méthodes multigrilles algébriques rattachée à HYPRE et constitue une référence dans le domaine [Yang et al., 2002, Cleary et al., 2000].

Choix final d'une méthode de résolution de système linéaire

Le choix final d'une méthode de résolution a été orienté par deux contraintes. La première est la volonté d'avoir une méthode scalable et robuste pour être exécutée dans un environnement massivement parallèle. Le multigrille géométrique s'imposerait alors naturellement puisqu'il répond parfaitement à cette exigence. Toutefois, l'implémentation d'une méthode multigrille est apparue comme une direction un peu risquée pour une première approche de résolution de systèmes linéaires. L'intégration dans le code aurait été assurément laborieuse et chronophage alors que l'apport de cette thèse se situe plutôt dans la validation d'une méthode incompressible et sa comparaison avec l'approche faiblement-compressible.

D'autre part, l'implémentation d'un algorithme de gradient conjugué ou GMRES présenterait peu d'intérêt si elle n'était pas couplée à un puissant préconditionnement.

Notre choix s'est finalement porté sur l'utilisation de la librairie PETSc (Portable, Extensible Toolkit for Scientific Computation) [ANL, 1997]. Cette librairie développée au Argonne National Laboratory depuis près d'une vingtaine d'années fournit un ensemble de structures de données et de routines adaptées à la résolution d'équations aux dérivées partielles. PETSc bénéficie du soutien actif de la communauté scientifique qui en fait aujourd'hui un outil puissant et riche. La librairie présente l'avantage d'être massivement parallèle sur des architectures variées (CPU et GPU) et a démontré son efficacité [Balay et al., 1997]. PETSc fournit les structures de données de la matrice du système linéaire et la méthode de Krylov pour résoudre l'équation de Poisson. Son utilisation est couplée à la librairie HYPRE pour accélérer la résolution avec une méthode multigrille algébrique. L'association des deux librairies offre d'excellentes performances sur les architectures parallèles tout en minimisant notre effort d'intégration. Elles sont par ailleurs basées sur la norme MPI, comme le code WCCH. Quelques exemples d'intégration de PETSc dans WCCH sont présentés en annexe B.

L'utilisation de PETSc nous impose de construire explicitement les matrices des opérateurs gradient et divergence. Pour cela nous utilisons une propriété mathématique notable de ces deux opérateurs sur maillage MAC cartésien. Considérons l'espace de Hilbert $H_0^1(\Omega)$ qui contient les fonctions différentiables une fois et définies dans le domaine Ω . Par ailleurs leur dérivée première s'annule aux frontières du domaine. Le produit scalaire associé à cet espace est : $\langle \mathbf{u}, \mathbf{v} \rangle = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} dV$.

∇ est un endomorphisme autoadjoint de $H_0^1(\Omega)$ ce qui signifie :

$$\nabla^* = -(\nabla \cdot) \quad (2.86)$$

Pour établir cette relation, on choisit une fonction scalaire ϕ et un vecteur \mathbf{u} dans $H_0^1(\Omega)$. Par définition \mathbf{u} est nulle sur le bord $\partial\Omega$ du domaine Ω :

$$0 = \int_{\partial\Omega} (\phi \mathbf{u}) \cdot \mathbf{n} dS = \int_{\Omega} \nabla \cdot (\phi \mathbf{u}) dV = \int_{\Omega} (\phi \nabla \cdot \mathbf{u} + \nabla \phi \cdot \mathbf{u}) dV \quad (2.87)$$

soit

$$\int_{\Omega} \phi \nabla \cdot \mathbf{u} dV = - \int_{\Omega} \nabla \phi \cdot \mathbf{u} dV \quad (2.88)$$

ou, exprimé en termes de produit scalaire :

$$\langle \phi, \nabla \cdot \mathbf{u} \rangle = - \langle \nabla \phi, \mathbf{u} \rangle \quad (2.89)$$

ce qui est la définition même d'un endomorphisme autoadjoint pour l'opérateur ∇ . Le maillage MAC permet de naturellement conserver cette propriété du domaine continu au domaine discret. Ainsi en pratique, nous pouvons calculer la matrice de l'opérateur divergence $\nabla \cdot$, et la matrice gradient ∇ est obtenue naturellement

en prenant la transposée et l'opposée de la matrice de divergence, aux conditions aux limites près. Par ailleurs les matrices ne dépendent que de la métrique du maillage. Elles n'ont besoin d'être recalculées qu'après chaque changement de maillage (raffinement adaptatif par exemple).

Enfin concernant le choix d'une méthode itérative de résolution, la méthode du gradient conjugué donne d'excellents résultats sur l'équation de Poisson au point d'en faire une méthode de choix. La symétrie du système linéaire est pleinement exploitée. En hydrodynamique navale, le traitement de la surface libre dissymétrise l'équation de Poisson et un algorithme BiCGSTAB est souvent privilégié. Après avoir testé les nombreux solveurs qu'offre PETSc, il s'est avéré que la combinaison GMRES redémarré + BoomerAMG donnait les meilleures performances comme cela est suggéré dans [Yang et al., 2002]. La méthode GMRES redémarré ou GMRES(k) consiste à relancer la méthode au bout de k itérations car le coût des itérations croît en $O(n^2)$, où n est la $n^{\text{ième}}$ itération.

Pour conclure, la résolution de l'équation de Poisson par des libraires tierces convient mieux à notre approche de "prototypage" rapide de schémas numériques. Le but reste de faire une preuve de concept sur l'apport de la résolution incompressible par rapport au faiblement-compressible pour ensuite convaincre de la nécessité éventuelle de pousser les développements plus loin. L'adoption d'un schéma multigrille géométrique deviendra alors vraisemblablement incontournable.

2.3.3 Discrétisation temporelle

Dans les méthodes de projection, l'intégration temporelle est souvent réalisée à l'aide d'un schéma d'Adams-Bashforth comme cela a été proposé par Kim et Moin [Kim et Moin, 1985]. Cette méthode nécessite de stocker en mémoire la valeur du terme convectif à l'instant précédent. Par ailleurs, à $t = 0$, un autre schéma numérique doit être utilisé comme un schéma explicite d'Euler à l'ordre 1. Le et Moin ont suggéré une amélioration avec un schéma Runge-Kutta d'ordre 3 [Le et Moin, 1991] qui étend le domaine de stabilité temporelle (en théorie $CFL = \sqrt{3}$). Morinishi et al. [Morinishi et al., 1998] utilisent également ce schéma RK3 en combinaison avec une discrétisation spatiale d'ordre élevé. L'emploi du schéma Runge-Kutta d'ordre 4 reste limité [Pereira et al., 2001, Kampanis et Ekaterinaris, 2006]. En général chaque sous-étape du schéma Runge-Kutta est envisagée comme une étape de projection. Une pression est déterminée à chaque sous-étape pour rendre les vitesses intermédiaires à divergence nulle. Seulement, l'influence d'une telle démarche sur les ordres de convergence de la vitesse et de la pression n'est pas toujours clairement établie. Il est souvent supposé que la vitesse converge à l'ordre théorique du schéma et qu'elle n'est pas affectée par la pression. Sanderse et Koren [Sanderse et Koren, 2012] ont récemment apporté des précisions sur ces zones d'ombre qui entourent l'ordre de convergence de la vitesse et de la pression pour les schémas Runge-Kutta. Nous reprenons ici quelques éléments de leur exposé.

L'utilisation courante des schémas Runge-Kutta explicites d'ordre s consiste donc à réaliser une étape de projection à chaque sous-étape :

$$\left\{ \begin{array}{l} \tilde{\mathbf{u}}^i = \mathbf{u}^n + \Delta t \sum_{j=1}^i \tilde{a}_{ij} \mathbf{RHS}^j \\ \nabla \cdot \frac{\tilde{c}_i \Delta t}{\rho} \nabla \phi^i = \nabla \cdot \tilde{\mathbf{u}}^i \\ \mathbf{u}^i = \tilde{\mathbf{u}}^i - \frac{\tilde{c}_i \Delta t}{\rho} \nabla \phi^i \end{array} \right. \quad i = 1, \dots, s \quad (2.90)$$

Les coefficients \tilde{a}_{ij} et \tilde{c}_i sont une représentation compacte et fréquente des schémas Runge-Kutta explicites :

$$\tilde{A} = \begin{pmatrix} a_{21} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{s1} & \cdots & a_{s,s-1} & 0 \\ b_1 & \cdots & b_{s-1} & b_s \end{pmatrix}, \tilde{c} = \begin{pmatrix} c_2 \\ \vdots \\ c_s \\ 1 \end{pmatrix} \quad (2.91)$$

ϕ^i est une variable qui impose la divergence nulle pour chaque vitesse intermédiaire et dont on peut montrer qu'elle est une approximation à l'ordre 1 en temps de la pression [Sanderse et Koren, 2012] pour des conditions aux limites périodiques :

$$\phi^i = p^i + O(\Delta t) \quad (2.92)$$

En prenant, comme pression dans l'écoulement la pression issue de la dernière sous-étape Runge-Kutta, on a donc une approximation à l'ordre 1 mais aucun résultat de convergence ne peut être attendu. \mathbf{RHS}^j regroupe la somme des flux convectifs et visqueux.

Le schéma ci-dessus est équivalent à

$$\begin{cases} \mathbf{u}^i = \mathbf{u}^n + \Delta t \sum_{j=1}^i \tilde{a}_{ij} \mathbf{RHS}^j - \frac{\tilde{c}_i \Delta t}{\rho} \nabla \phi^j \\ \nabla \cdot \frac{\tilde{c}_i}{\rho} \nabla \phi^i = \nabla \cdot \sum_{j=1}^i \tilde{a}_{ij} \mathbf{RHS}^j \end{cases} \quad i = 1, \dots, s \quad (2.93)$$

en supposant $\nabla \cdot \mathbf{u}^n = 0$.

Au lieu de calculer une pression ψ à chaque sous-étape, Sanderse et Koren proposent d'utiliser une combinaison des pressions à l'aide d'un nouveau tableau de Butcher \tilde{A}^p :

$$\begin{cases} \mathbf{u}^i = \mathbf{u}^n + \Delta t \sum_{j=1}^i \tilde{a}_{ij} \mathbf{RHS}^j - \frac{\Delta t}{\rho} \sum_{j=1}^i \tilde{a}_{ij}^p \nabla \psi^j \\ \sum_{j=1}^i \tilde{a}_{ij}^p \nabla \cdot \frac{1}{\rho} \nabla \psi^j = \nabla \cdot \sum_{j=1}^i \tilde{a}_{ij} \mathbf{RHS}^j \end{cases} \quad i = 1, \dots, s \quad (2.94)$$

\tilde{A}^p peut être pris différent ou égal à \tilde{A} . En comparant les systèmes (2.93) et (2.94), il vient

$$\sum_{j=1}^i \tilde{a}_{ij}^p \psi^j = \tilde{c}_i \phi^i, \quad i = 1, \dots, s \quad (2.95)$$

Ou encore

$$\psi = (\tilde{A}^p)^{-1} \begin{pmatrix} c_2 & 0 & \cdots & 0 \\ 0 & c_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} \phi \quad (2.96)$$

Finalement il est donc possible de construire une pression à l'ordre s en temps au terme des sous-étapes mais seulement pour des conditions aux limites périodiques.

Dans le cadre du schéma Runge-Kutta 4 classique utilisé en formulation faiblement-compressible dans WCCH, l'évaluation de (2.96) avec $\tilde{A}^p = \tilde{A}$ donne :

$$\psi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & -2 & -2 & 6 \end{pmatrix} \phi \quad (2.97)$$

Une pression à l'ordre 4 en temps est donc donnée par :

$$p^{n+1} = \psi^4 = -\phi^1 - 2\phi^2 - 2\phi^3 + 6\phi^4 \quad (2.98)$$

Pour conclure, un tel schéma pour la pression n'a d'intérêt que pour le post-traitement de cette grandeur. En effet comme la pression n'intervient que pour imposer instantanément la contrainte de divergence nulle indépendamment du temps, ce schéma ne va pas améliorer la précision globale de la solution. Il présenterait un avantage si la pression était utilisée à chaque pas de temps pour, par exemple, intégrer un effort sur un corps solide. La méthode de frontière immergée que nous utilisons nous permet de nous affranchir de cela. De plus dans le cas où des conditions aux limites de Dirichlet sont imposées sur la vitesse, l'ordre du schéma pour la pression chute à l'ordre du schéma de projection de base. Ainsi, sauf mention explicite, c'est le schéma Runge-Kutta avec pression issue de la dernière sous-étape qui est utilisé dans la suite de ce manuscrit.

2.3.4 Conditions aux limites

La définition des conditions aux limites est l'un des derniers obstacles, et non des moindres, vers l'implémentation précise et robuste d'un schéma numérique. Le domaine d'étude d'un écoulement étant borné, il est nécessaire d'appliquer des conditions aux limites pour clore le système d'équations.

En formulation compressible, le traitement des conditions aux limites est dicté par la théorie des caractéristiques. Cette dernière impose le nombre de conditions physiques, i.e. les conditions issues de la définition physique du problème fluide. D'autre part, le schéma numérique intérieur du domaine peut requérir des informations supplémentaires non fournies par la physique. Des conditions numériques doivent alors être ajoutées pour que le problème reste bien posé. En formulation incompressible, le problème n'est plus de nature hyperbolique. Le caractère mixte parabolique/elliptique des équations de Navier-Stokes incompressibles rend cette distinction de conditions aux limites *physiques* et numériques caduque. Le choix, la justification et la mise en place des conditions aux limites en incompressible n'en reste pas moins subtile.

Nous avons déjà vu que l'application de conditions aux limites de type Dirichlet sur la vitesse faisait chuter l'ordre du schéma de projection originel. Les autres types de conditions aux limites et leur traitement sont très peu évoqués dans la littérature. Guermond [Guermond et al., 2006] apporte quelques éléments de réponse pour des frontières ouvertes.

Par ailleurs, pour les cellules situées à la frontière du domaine, se pose la question du choix des points d'interpolation, que ce soit pour les reconstructions à l'aide du schéma WENO ou le calcul des dérivées partielles de vitesse pour le terme diffusif. En effet les stencils étant tronqués, les schémas numériques doivent être adaptés localement pour prendre en compte la présence de la frontière. Une façon élégante et simple de procéder au traitement des frontières est d'étendre le domaine de calcul au-delà de la frontière à l'aide de cellules fantômes (ghost-cells en anglais). Des valeurs de pression et de vitesse sont attribuées à ces cellules de façon à respecter la condition numérique sur la frontière (Fig. 2.8). Pour un schéma WENO qui s'appuie sur 5 points d'interpolation, au moins 3 cellules fantômes sont nécessaires.

Dans le code WCCH, le traitement des frontières pour le schéma WENO ne recourt pas aux cellules fantômes. Les cellules situées à proximité de la frontière et pour lesquelles certains points d'interpolation ne sont pas disponibles voient leurs coefficients de pondération des points qui manquent annulés. Par exemple,

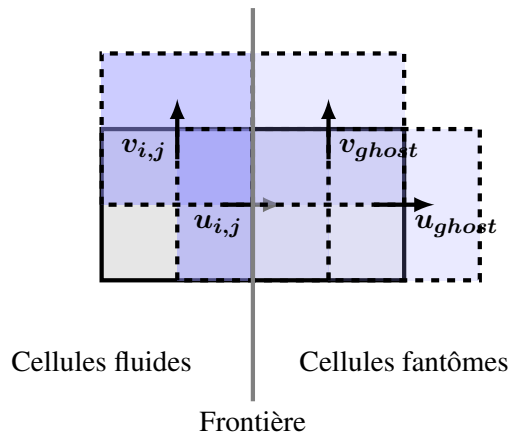


FIGURE 2.8 – Extension du domaine de calcul aux frontières à l’aide de cellules fantômes pour les vitesses

pour une cellule avec une frontière à sa droite, seul le stencil $S2$ (Fig 1.3) est considéré. $S1$ et $S0$ ne sont pas pris en compte en annulant leurs pondérations associées : $d_0 = 0$ et $d_1 = 0$. Ce traitement particulier diffère de celui des cellules fantômes mais reste néanmoins consistant et d’ordre équivalent. Pour un exemple de prise en compte par des conditions aux limites par les cellules fantômes, on pourra se reporter à la thèse de Couderc [Couderc, 2007]. Quant au calcul du terme diffusif, une procédure de type cellules fantômes est adoptée.

Condition périodique

Il s’agit de la plus simple des conditions aux limites qui consiste à répéter le domaine indéfiniment dans l’espace (Fig. 2.9). L’avantage est qu’elles sont exactes et permettent d’étudier les schémas numériques sans l’influence de conditions aux limites. Elles permettent aussi par exemple de simuler un écoulement autour d’un motif qui se répète à l’infini.

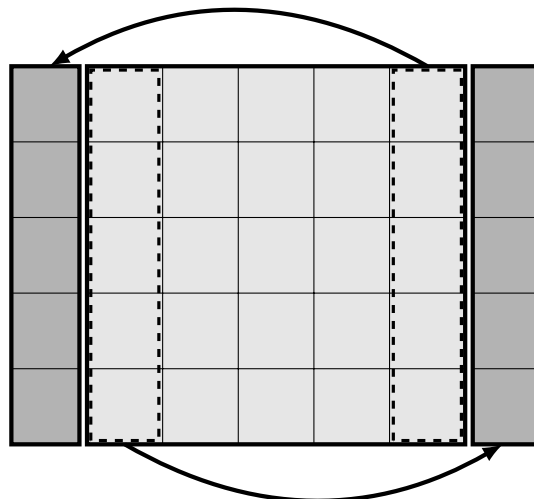


FIGURE 2.9 – Condition aux limites périodique dans la direction horizontale

Condition de glissement

La condition de glissement exprime que le fluide glisse sur la surface solide mais ne la pénètre pas. Si \mathbf{n} est le vecteur normal à la paroi :

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad (2.99)$$

Cette condition convient notamment pour les écoulements non visqueux (équations d'Euler). Nous choisissons d'imposer une condition aux limites seulement pour la composante tangentielle du vecteur vitesse puisque son volume de contrôle est le seul en contact avec la frontière (Fig. 2.10).

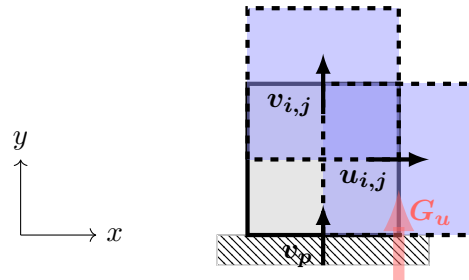


FIGURE 2.10 – Condition de glissement

Pour le calcul du terme convectif, la vitesse tangentielle à la frontière ($u_{i,j}$ sur Fig. 2.10) voit le flux entrant dans son volume de contrôle au niveau de la frontière s'annuler. En effet $uv = 0$ à la paroi à cause de la condition de non-pénétration $v_p = 0$. Pour le terme diffusif, la condition de glissement est construite à partir d'une condition de symétrie $\frac{\partial u}{\partial n} = 0$.

Ainsi les flux à la frontière entrant dans le volume de contrôle de la vitesse tangentielle sont annulés. Dans l'exemple illustré sur la figure 2.10 cela se traduit par $G_u = 0$.

D'un point de vue continu, la condition pour la vitesse \mathbf{u}^* est identique à celle sur \mathbf{u}^{n+1} puisque la pression respecte une condition de Neumann suivant la condition de symétrie $\frac{\partial p}{\partial n} = 0$. La condition de glissement n'est donc pas responsable d'une dégradation de l'ordre de la méthode de projection originelle.

Condition d'adhérence

Dans le cas des écoulements visqueux, la condition d'adhérence remplace la condition de glissement. La vitesse tangentielle du fluide à la paroi doit alors être aussi fixée. La paroi peut être immobile ou en mouvement à la vitesse u_p . En reprenant l'exemple de la figure 2.10, le flux convectif normal à la paroi ne contribue pas, comme pour la condition de glissement, mais le flux visqueux est non nul :

$$G_u = \nu \frac{\partial u}{\partial y} \approx \nu \frac{u_{i,j} - u_p}{\Delta y/2} \quad (2.100)$$

Le résultat est similaire à celui obtenu par des cellules fantômes.

Condition entrante

Une condition entrante consiste à imposer un profil de vitesse sur la frontière du domaine. Dans notre cas nous nous intéressons uniquement à des profils laminaires et stationnaires bien qu'une implémentation de condition entrante turbulente ait déjà été réalisée dans le code WCCH pour la simulation d'écoulements autour d'hydroliennes en formulation faiblement-compressible [Elie et al., 2017]. Prenons une paroi du domaine perpendiculaire à la direction x . Le profil de vitesse à la frontière est de la forme $\mathbf{u} \cdot \mathbf{e}_x = u(y)$ avec $\mathbf{u} \cdot \mathbf{e}_y = 0$, \mathbf{e}_x et \mathbf{e}_y étant les vecteurs unitaires des directions x et y . Pour un écoulement plan

unidirectionnel, le fluide est mis en mouvement par un gradient de pression si bien que la vitesse et la pression sont liées par la relation de perte de charge :

$$\frac{\partial p}{\partial x} = \mu \frac{d^2 u(y)}{dy^2} \quad (2.101)$$

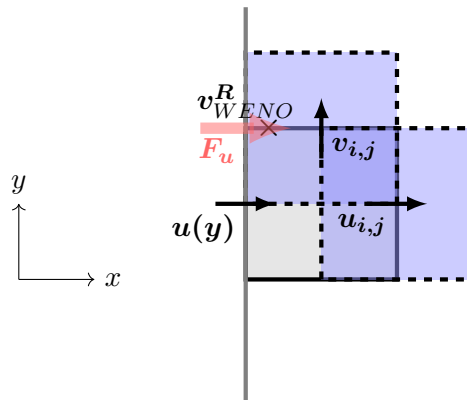


FIGURE 2.11 – Condition entrante et flux entrant

En considérant l'exemple décrit sur la figure 2.11, le calcul du flux F_u entrant dans le volume de contrôle de la vitesse tangentielle à la paroi ($v_{i,j}$ dans l'exemple) se résume au calcul du flux convectif puisque $\frac{\partial v}{\partial x} = 0$, annulant ainsi le flux diffusif. De plus on peut considérer que la reconstruction WENO à gauche de la frontière est égale à la reconstruction à droite (toujours lié au fait que $\frac{\partial v}{\partial x} = 0$). Ainsi le flux F_u est réduit à :

$$F_u = u(y)v_{WENO}^R \quad (2.102)$$

quel que soit le signe de $u(y)$.

D'autre part l'équation (2.101) est incompatible avec la condition de Neumann artificiellement créée sur la pression par le schéma de projection de Chorin lorsqu'une condition de Dirichlet est appliquée sur la vitesse (cf. section 2.2), sauf dans le cas où le gradient de pression est physiquement nul comme c'est le cas pour un profil de vitesse uniforme ou linéaire.

Dans les écoulements que nous étudierons par la suite, nous n'utiliserons que des profils de vitesse constants sur toute la frontière. Le problème de couche limite artificielle (cf. section 2.2) ne se manifestera donc pas. Enfin comme pour la condition de glissement, la condition sur \mathbf{u}^* est la même que pour \mathbf{u}^{n+1} puisqu'une condition de Neumann homogène est appliquée à la pression.

Condition de sortie

Cette condition est délicate à traiter puisque aucune information n'est disponible a priori en aval de l'écoulement. De nombreuses parades mathématiques ont été échafaudées pour contourner ce problème, avec des résultats souvent mitigés. Les méthodes de prise en compte de telles conditions restent souvent basées sur l'expérience numérique. Une condition aux limites de Dirichlet sur la pression fait souvent consensus comme $p^{n+1} = 0$ (en l'absence de gravité). Un principe général de base consiste aussi à placer cette condition suffisamment loin des zones d'intérêt pour limiter son influence. Pour la vitesse, nous emploierons la condition suivante :

$$\begin{aligned} \frac{\partial \mathbf{u}^*}{\partial n} &= 0 \\ \nabla \cdot \mathbf{u}^{n+1} &= 0 \end{aligned} \quad (2.103)$$

Les contributions visqueuses sur les flux sortants sont donc nulles. Les reconstructions WENO manquantes côté extérieur du domaine (symbole *, à droite sur la figure 2.12) sont prises égales aux reconstructions côté fluide (symbole ×, à gauche sur la figure 2.12).

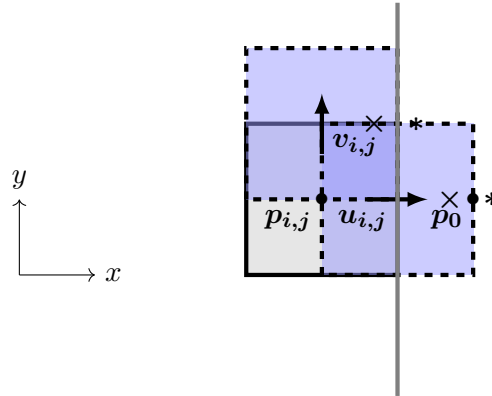


FIGURE 2.12 – Condition sortante et reconstructions WENO disponibles × et manquantes *

L'ordre de convergence de cette condition aux limites pour le schéma de projection de Chorin n'a pas été établi formellement dans la littérature. Néanmoins Guermond [Guermond et al., 2006] a montré que des conditions aux limites similaires convergent avec le bon ordre attendu pour la pression dans le cas d'un schéma de projection rotationnelle. Même si nous ne pouvons pas avancer un tel résultat pour notre schéma, les conditions choisies ne sont pas dénuées de sens et laissent espérer une convergence satisfaisante.

Traitement particulier de l'équation de Poisson avec des conditions de Neumann sur la pression ou des domaines périodiques

Dans le cas où uniquement des conditions de Dirichlet sur la vitesse sont appliquées (conditions de glissement ou d'adhérence), la condition de Neumann implicite qui en résulte sur la pression rend la matrice de discrétisation de l'équation de Poisson singulière. Il en est de même pour les domaines périodiques. Physiquement cela s'explique par le fait que la pression n'est alors définie qu'à une constante près. Une contrainte additionnelle sur le champ de pression est nécessaire pour résoudre ce problème. On peut par exemple fixer à chaque instant la valeur moyenne de la pression dans tout le domaine :

$$\iiint_{\Omega} p(\mathbf{x}, t) dV = 0 \quad (2.104)$$

Une alternative est de fixer la pression d'une cellule i du maillage à une pression de référence p_{ref} . Le système linéaire de l'équation de Poisson est alors modifié en remplaçant la ligne de la matrice du système linéaire correspondant à la cellule concernée par une ligne d'éléments nuls sauf pour le terme diagonal fixé à 1 :

$$i^{\text{ème}} \text{ ligne} \begin{pmatrix} \cdots & \cdots & & \cdots \\ \cdots & 0 & 1 & 0 & \cdots \\ & & \cdots & & \end{pmatrix} \begin{pmatrix} \vdots \\ p_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ p_{ref} \\ \vdots \end{pmatrix} \quad (2.105)$$

2.4 Adaptation de la méthode de frontière immergée pour la formulation incompressible

La méthode de frontière immergée adoptée dans WCCH pour prendre en compte des corps solides dans des maillages cartésiens consiste à décomposer l'accélération globale d'une cellule en une accélération fluide

et une accélération solide. En réécrivant l'équation (1.25) avec les dérivées temporelles de vitesse au lieu de celles des variables conservatives utilisées en faiblement-compressible, nous obtenons :

$$\left(\frac{\partial \mathbf{u}}{\partial t}\right)^n = (1 - H_{i,j,k}) \left(\frac{\partial \mathbf{u}}{\partial t}\right)_{\text{fluide}}^n + H \left(\frac{\partial \mathbf{u}}{\partial t}\right)_{\text{solide}}^n \quad (2.106)$$

Nous rappelons que H désigne ici une fonction de type Heaviside qui vaut 1 dans le corps et 0 ailleurs. En pratique H est régularisée au voisinage de la frontière solide pour lisser la transition entre l'accélération fluide et l'accélération solide (Fig. 1.6). Ainsi, au voisinage du corps il existe une mince région volumique (qui tend vers l'enveloppe surfacique du corps à mesure que le maillage est raffiné) dans laquelle coexistent les deux accélérations. En formulation faiblement-compressible, l'équation de continuité n'est pas modifiée à la traversée du corps et établit naturellement un champ de pression dans le corps. En formulation incompressible, le rôle particulier de la pression complique le choix mais aussi la justification de l'extension de la méthode de frontière immergée au schéma de projection. Notamment, comment imposer la contrainte de divergence nulle ? Sur le champ de vitesse fluide ? ou sur le champ fluide modifié par la prise en compte de la contrainte cinématique solide ?

Dans un premier temps nous discrétisons l'équation (2.106) à l'aide du schéma d'Euler d'ordre 1 pour faire apparaître une relation entre les champs de vitesses fluide et solide notés respectivement u_f et u_c :

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = (1 - H) \frac{\mathbf{u}_f^{n+1} - \mathbf{u}^n}{\Delta t} + H \frac{\mathbf{u}_c^{n+1} - \mathbf{u}^n}{\Delta t} \quad (2.107)$$

Cette formulation nécessite la connaissance a priori de \mathbf{u}_c^{n+1} mais permet de relier les différents champs de vitesse sous la même forme de combinaison convexe :

$$\mathbf{u}^{n+1} = (1 - H)\mathbf{u}_f^{n+1} + H\mathbf{u}_c^{n+1} \quad (2.108)$$

Dans le chapitre 5 nous reviendrons sur la différence et l'utilité pratique des deux formulations, en accélération (2.106) ou en vitesse (2.108).

Le champ de vitesse \mathbf{u}_c^{n+1} est celui d'un corps rigide indéformable. En théorie il est à divergence nulle. Supposons que l'on parvienne dans une première approche à construire un champ fluide \mathbf{u}_f^{n+1} à divergence nulle, alors la divergence du champ global \mathbf{u}^{n+1} donne :

$$\begin{aligned} \nabla \cdot \mathbf{u}^{n+1} &= \nabla \cdot [(1 - H)\mathbf{u}_f^{n+1}] + \nabla \cdot [H\mathbf{u}_c^{n+1}] \\ &= (1 - H)\nabla \cdot \mathbf{u}_f^{n+1} + \mathbf{u}_f^{n+1} \cdot \nabla(1 - H) + H\nabla \cdot \mathbf{u}_c^{n+1} + \mathbf{u}_c^{n+1} \cdot \nabla H \\ &= (\mathbf{u}_c^{n+1} - \mathbf{u}_f^{n+1}) \cdot \nabla H \end{aligned} \quad (2.109)$$

Considérons que H désigne la fonction Heaviside singulière, non régularisée au voisinage de l'interface fluide/solide. A la frontière solide, ∇H est alors un vecteur normal dont la norme est égale à la distribution de Dirac. ∇H est nul partout sauf à la surface du corps. Mais comme $\mathbf{u}_c^{n+1} = \mathbf{u}_f^{n+1}$ par entraînement, on a finalement $\nabla \cdot \mathbf{u}^{n+1} = 0$ partout.

A l'échelle discrète ∇H est nul partout sauf au voisinage de la frontière du solide comme le montre la figure 2.13 et donc $\nabla \cdot \mathbf{u}^{n+1} \neq 0$ dans cette région de l'écoulement.

Ainsi si nous calculons dans un premier temps une vitesse prédite \mathbf{u}^* partout dans le domaine de calcul, que celle-ci est projetée pour la rendre à divergence nulle et enfin pondérée avec la vitesse solide selon (2.108), alors la vitesse finale n'est plus à divergence nulle. Si nous cherchions malgré tout à rendre le champ final à divergence nulle, cela aurait des conséquences importantes sur l'équation de Poisson. Reprenons la décomposition de vitesse globale (2.108) en faisant apparaître la décomposition de Helmholtz-Hodge de la vitesse fluide :

$$\mathbf{u}^{n+1} = (1 - H)\left(\mathbf{u}_f^* - \frac{\nabla p^{n+1}}{\rho}\right) + H\mathbf{u}_c^{n+1} \quad (2.110)$$

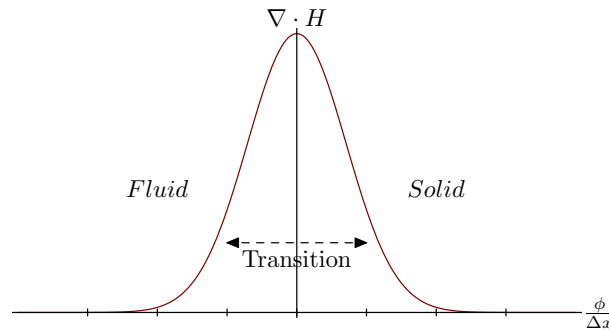


FIGURE 2.13 – Gradient de la fonction d’amplification de l’accélération solide à la traversée de la frontière

En appliquant l’opérateur divergence à cette expression, il vient :

$$\nabla \cdot \mathbf{u}^{n+1} = \nabla \cdot \left[(1 - H)(\mathbf{u}_f^* - \frac{\Delta t}{\rho} \nabla p^{n+1}) \right] + \nabla \cdot [H\mathbf{u}_c^{n+1}] \quad (2.111)$$

Avec le souhait que $\nabla \cdot \mathbf{u}^{n+1} = 0$, on forme l’équation de Poisson suivante :

$$\nabla \cdot [H\mathbf{u}_c^{n+1} + (1 - H)(\mathbf{u}_f^*)] = \nabla \cdot \left[(1 - H) \frac{\Delta t}{\rho} \nabla p^{n+1} \right] \quad (2.112)$$

La discrétisation de cette équation va conduire à une matrice de Laplacien singulière puisque $H = 1$ à l’intérieur du corps. Une solution pourrait être d’imposer une pression de référence pour tous les points intérieurs au corps pour lesquels H est presque égal à 1 ($1 - H < \epsilon$, ϵ très petit) et n’inclure dans la résolution de l’équation de Poisson que les points intérieurs au corps situés dans la zone de transition (Fig. 2.14), à proximité de la frontière solide (pour lesquels $1 - H > \epsilon$).

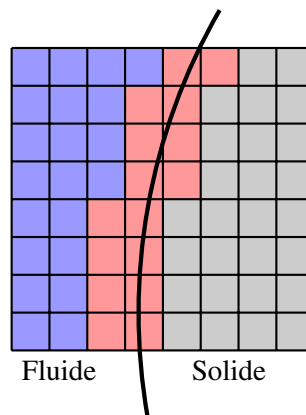


FIGURE 2.14 – Définition des cellules fluides ■, des cellules solides ■ et des cellules de la zone de transition ■

Cette méthode a l’avantage de ne modifier que légèrement le processus de construction du système linéaire. Toutefois si $1 - H$ est proche de ϵ (et est donc très petit), la valeur propre minimale de la matrice du Laplacien modifiée sera proche de 0. La matrice sera alors mal conditionnée avec toutes les difficultés de résolution du système linéaire par des méthodes itératives que cela peut entraîner. La bonne démarche serait finalement d’adapter la résolution de l’équation de Poisson à des domaines irréguliers comme dans [Johansen et Colella, 1998].

Une deuxième approche peut consister à inverser l'ordre d'application des contraintes en pondérant d'abord une vitesse fluide prédite \mathbf{u}^* (qui ne sera donc pas à divergence nulle) avec une vitesse solide puis en projetant la vitesse résultante. La vitesse globale est alors à divergence nulle mais la condition cinématique de vitesse solide n'est plus respectée. Cette approche est équivalente à ajouter un terme de pression pour l'accélération solide à l'intérieur du corps :

$$\begin{aligned}\nabla \cdot \mathbf{u}^{n+1} &= \nabla \cdot \left[(1-H)\left(\mathbf{u}_f^* - \frac{\Delta t}{\rho} \nabla p^{n+1}\right) \right] + \nabla \cdot \left[H\left(\mathbf{u}_c^{n+1} - \frac{\Delta t}{\rho} \nabla p^{n+1}\right) \right] \\ &= \nabla \cdot \left[(1-H)\mathbf{u}_f^* \right] + \nabla \cdot \left[H\mathbf{u}_c^{n+1} \right] - \nabla \cdot \frac{\Delta t}{\rho} \nabla p^{n+1}\end{aligned}\quad (2.113)$$

La pression p^{n+1} impose donc le champ de vitesse $(1-H)\mathbf{u}_f^* + H\mathbf{u}_c^{n+1}$ à divergence nulle avec l'équation de Poisson :

$$\nabla \cdot \left[(1-H)\mathbf{u}_f^* \right] + \nabla \cdot \left[H\mathbf{u}_c^{n+1} \right] = \nabla \cdot \frac{\Delta t}{\rho} \nabla p^{n+1}\quad (2.114)$$

A l'intérieur du corps où $H \equiv 1$, $\nabla^2 p^{n+1} = 0$. p^{n+1} est donc harmonique sur cette partie du domaine.

Pour résumer, deux approches ont été envisagées dans cette partie, une avec pondération par l'équation (2.108) après la résolution de l'équation de Poisson et l'autre avant. Elles ont un problème commun bien identifié dans la littérature qui a pour origine le fait que l'on cherche à imposer deux contraintes simultanément : un champ de vitesse fluide à divergence nulle et une cinématique de vitesse solide. En traitant les contraintes successivement, le respect de l'une entraîne le non-respect de l'autre. Une façon de bien traiter le problème serait de le résoudre de manière fortement couplée en résolvant simultanément les deux contraintes comme dans [Taira et Colonius, 2007] ou d'itérer sur les contraintes.

Sans aller jusque là, l'approche qui respecte la contrainte de vitesse à divergence nulle en l'appliquant en dernier est souvent retenue [Kim et al., 2001]. Elle est préférable dans la mesure où des problèmes de stabilité et de robustesse au voisinage de la frontière peuvent être redoutés si le champ de vitesse globale n'est pas à divergence nulle.

En résumé le schéma de projection de type Chorin adapté à la méthode de frontière immergée s'écrit :

1. Prédiction

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t(\nu \nabla^2 \mathbf{u}^n - \nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) + \mathbf{f}^n)\quad (2.115)$$

2. Ajout de la contrainte cinématique solide

$$\mathbf{u}^{**} = (1-H)\mathbf{u}^* + H\mathbf{u}_c^{n+1}\quad (2.116)$$

3. Equation de Poisson en pression

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla p^{n+1} \right) = \nabla \cdot \mathbf{u}^{**}\quad (2.117)$$

4. Correction

$$\mathbf{u}^{n+1} = \mathbf{u}^{**} - \frac{\Delta t}{\rho} \nabla p^{n+1}\quad (2.118)$$

En pratique cette formulation n'est pas exactement celle qui a été utilisée par la suite. Nous avons privilégié une approche incrémentale du schéma de projection sous la forme :

1. Prédiction

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t(\nu \nabla^2 \mathbf{u}^n - \nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) + \mathbf{f}^n - \frac{\nabla p^n}{\rho})\quad (2.119)$$

2. Ajout de la contrainte cinématique solide

$$\mathbf{u}^{**} = (1 - H)\mathbf{u}^* + H\mathbf{u}_c^{n+1} \quad (2.120)$$

3. Equation de Poisson en pression

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla \psi^{n+1} \right) = \nabla \cdot \mathbf{u}^{**} \quad (2.121)$$

4. Correction

$$\mathbf{u}^{n+1} = \mathbf{u}^{**} - \frac{\Delta t}{\rho} \nabla \psi^{n+1} \quad (2.122)$$

5. Incrémentation de la pression

$$p^{n+1} = p^n + \psi^{n+1} \quad (2.123)$$

En effet l'expérience numérique nous a montré que l'approche incrémentale donne de meilleurs résultats. Il a été constaté que la correction apportée à \mathbf{u}^{**} par le gradient de pression ∇p^{n+1} (et le fait de ne plus respecter la condition cinématique de vitesse) au voisinage de la frontière avait une incidence non négligeable à l'intérieur du solide. Le champ de vitesse final \mathbf{u}^{n+1} dans le corps est alors éloigné du champ solide attendu \mathbf{u}_c^{n+1} . En formulation incrémentale la correction apportée dans le corps par $\nabla \psi^{n+1}$ à \mathbf{u}^{**} est de moindre intensité. De plus, nous avons constaté que ψ^{n+1} tend vers 0 partout dans le domaine pour des écoulements stationnaires. Dans ces cas précis les contraintes de cinématique solide et de divergence nulle sont respectées simultanément. \mathbf{u}^{n+1} est rigoureusement égale à \mathbf{u}_c^{n+1} dans le corps.

Enfin pour conclure ce paragraphe sur la méthode de frontière immergée en formulation incompressible, on notera qu'en forçant progressivement la cinématique au voisinage et dans le corps, une forme de condition de Dirichlet est imposée sur la vitesse. Une couche limite artificielle peut donc se produire à la frontière solide. Cependant le fait de lisser la transition entre le fluide et le solide et donc d'imposer progressivement le champ de vitesse à la traversée de la frontière permet d'affirmer que son influence sera faible. Par ailleurs cet effet est censé décroître avec l'augmentation du nombre de Reynolds.

2.5 Conclusion

Dans ce chapitre nous avons sélectionné un algorithme de résolution des équations de Navier-Stokes incompressibles. Le schéma de projection de Chorin semble répondre le mieux à nos exigences et nous permet de ne pas renoncer aux caractéristiques de la formulation faiblement-compressible : efficacité algorithmique et schémas de discrétisation d'ordre élevé en espace et en temps.

La principale difficulté rencontrée lors de l'implémentation est le passage des maillages colocalisés de l'approche faiblement-compressible aux maillages MAC. En effet, le code WCCH a été pensé pour gérer des variables localisées au centre des volumes de contrôle. De nombreuses structures et routines ont dû être adaptées pour prendre en compte aussi des variables stockées aux faces.

Au vu du bref état de l'art dressé dans ce chapitre, nous sommes en mesure d'attendre pour le solveur incompressible des performances algorithmiques proches du solveur faiblement-compressible. L'un des éléments susceptibles d'affecter la scalabilité globale du code est la résolution de l'équation de Poisson, incontournable en incompressible. Mais il s'agit là d'un système linéaire à coefficient constant pour lequel de nombreuses méthodes numériques ont été mises au point. La méthode multigrille algébrique laisse envisager une bonne accélération de la résolution et une complexité algorithmique linéaire. Elle minimise aussi pour nous l'effort d'intégration.

Pour ce qui est de l'ordre de précision du schéma, le gain sur la diffusion numérique est assuré mais nous pouvons craindre un ordre de convergence plus faible en formulation incompressible par rapport à la formulation faiblement-compressible, notamment imputable au rôle particulier joué par la pression.

En outre, la méthode de frontière immergée initialement développée pour l'approche faiblement-compressible a été étendue au formalisme incompressible. La principale difficulté est l'impossibilité d'imposer à la fois la contrainte cinématique du solide rigide et la contrainte de divergence nulle du champ de vitesse fluide. La solution retenue est de préserver la divergence nulle du champ de vitesse global au détriment de la contrainte cinématique du champ solide. L'influence de ce choix sera étudiée plus en détail dans le chapitre consacré à l'analyse et l'amélioration de la méthode de frontière immergée.

Dans le chapitre suivant nous présentons l'extension de ce schéma incompressible sur des grilles cartésiennes raffinées localement.

Chapitre 3

Extension de la formulation pour des grilles raffinées localement

Ce chapitre est consacré à l'extension du solveur incompressible présenté dans le chapitre précédent pour des grilles raffinées localement. Toute la complexité de ce type de grille est de maintenir la consistance, la précision et la robustesse des opérateurs aux interfaces de raffinement.

Dans notre exposé qui suit, nous ne considérons que des maillages raffinés localement et n'évoluant pas au cours du temps (raffinement "statique").

3.1	Présentation de la difficulté	83
3.1.1	Maillage raffiné localement et interpolation d'ordre élevé	83
3.1.2	Maillage raffiné localement et calcul de flux sur maillage MAC	84
3.2	État de l'art	84
3.2.1	Stratégies pour la résolution de l'équation de Poisson	85
3.2.2	Méthodes pour le calcul des flux	89
3.2.3	Bilan et choix d'une méthode	92
3.3	Discretisation aux interfaces de raffinement	92
3.3.1	Discretisation et implémentation des opérateurs divergence et gradient	92
3.3.2	Discretisation des termes convectifs et diffusifs	94
3.4	Conclusion	102

3.1 Présentation de la difficulté

Dans cette partie nous identifions les aspects du schéma de projection qui doivent être adaptés pour appliquer le schéma incompressible sur une grille cartésienne raffinée localement.

3.1.1 Maillage raffiné localement et interpolation d'ordre élevé

La construction d'un schéma d'ordre élevé, comme par exemple le schéma des reconstructions WENO d'ordre 5, sur une grille raffinée localement n'est pas immédiate. La non-uniformité de la grille et le non-alignement des variables en 2D et 3D font que les stencils d'interpolation ne sont plus clairement définis entre deux blocs de niveau de raffinement différents. De plus, le maintien de l'ordre des opérateurs numériques à une interface de raffinement n'est pas toujours évident. Cette interrogation n'est pas nouvelle et se pose également en formulation faiblement-compressible avec l'utilisation de schémas à pas variable.

Pour un maillage non uniforme, il est possible de montrer que l'ordre de convergence d'un opérateur calculé en différences finies est conservé [Ferziger et Perić, 1996]. La grille doit cependant rester suffisamment régulière. Les variations de taille d'un point d'appui du maillage à l'autre ne doivent pas être démesurées. Pour une discrétisation en volumes finis, Herbin démontre dans son cours d'analyse des EDP [Herbin, 2011] que la conservativité et la consistance des flux assurent la convergence. En particulier sur un maillage non uniforme, si l'interface est au milieu des centres des volumes de contrôle, alors l'erreur de consistance est d'ordre 2 sur les flux et le schéma est globalement d'ordre 2.

En dépit de ces résultats, les analyses de schémas d'ordre élevé sur des maillages raffinés localement avec un rapport de taille de 1 pour 2 restent rares.

3.1.2 Maillage raffiné localement et calcul de flux sur maillage MAC

Une autre difficulté du passage d'une grille cartésienne uniforme à une grille raffinée localement est liée à l'emplacement entrelacé des variables. Le problème qui apparaît avec les variables entrelacées en formulation incompressible (absent en formulation faiblement-compressible) est la définition des volumes de contrôle aux interfaces de raffinement. En effet ceux-ci ne sont plus parfaitement jointifs si on les définit de sorte que les vitesses soient localisées à leur barycentre comme pour une grille cartésienne uniforme (Fig. 3.1). La vitesse à l'interface peut être portée par la cellule grossière ou les cellules fines et les volumes de contrôle des vitesses tangentielles ne sont plus alignés.

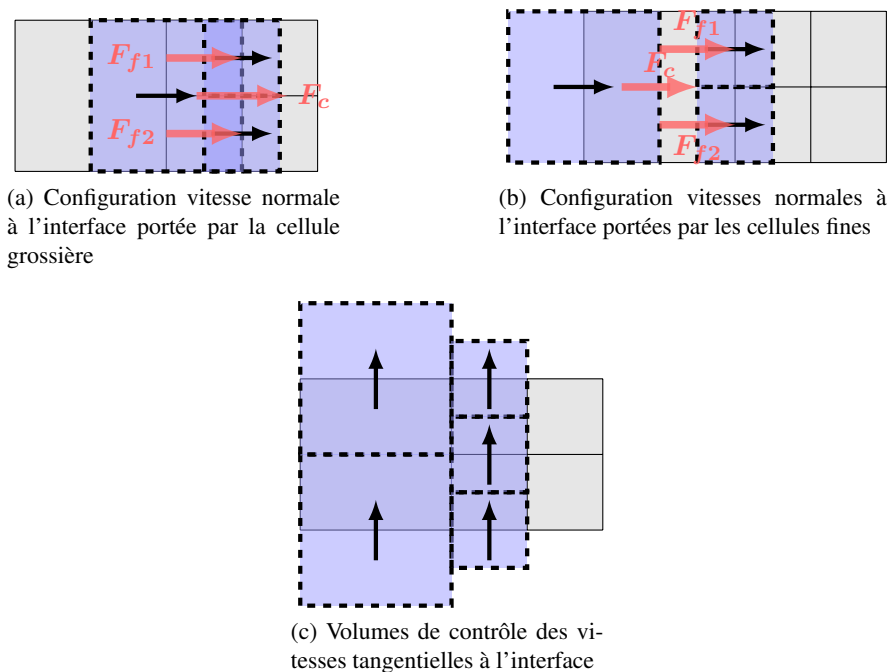


FIGURE 3.1 – Non conservation des flux \rightarrow et non coïncidence des volumes de contrôle

La conséquence immédiate est l'impossibilité d'assurer une conservation des flux de quantité de mouvement en prolongeant une telle définition des volumes de contrôle adaptée au maillage cartésien uniforme.

3.2 État de l'art

Nous passons ici en revue les différentes approches permettant de résoudre les deux étapes importantes du schéma de projection de Chorin : le calcul des flux pour la vitesse prédite u^* et la projection proprement

dite avec la résolution de l'équation de Poisson.

3.2.1 Stratégies pour la résolution de l'équation de Poisson

La résolution de l'équation de Poisson sur une grille raffinée localement est un problème bien répertorié dans la littérature et a fait l'objet de nombreuses études. Il reste assez indépendant du clivage "collocated"/"staggered" des variables car la pression est au centre des cellules dans les deux cas. L'opérateur Laplacien est donc identique pour les deux dispositions de variable.

Opérateur divergence $\nabla \cdot$

Si chaque cellule possède sur ses faces la composante de vitesse correspondant à la direction normale, alors la définition de l'opérateur divergence est en tout point identique à celle d'une grille cartésienne uniforme. Cependant aux interfaces de raffinement, trois inconnues de vitesse sont présentes : une associée à la cellule grossière et deux aux cellules fines (notées respectivement u_c et u_{f1}, u_{f2} , Fig. 3.2).

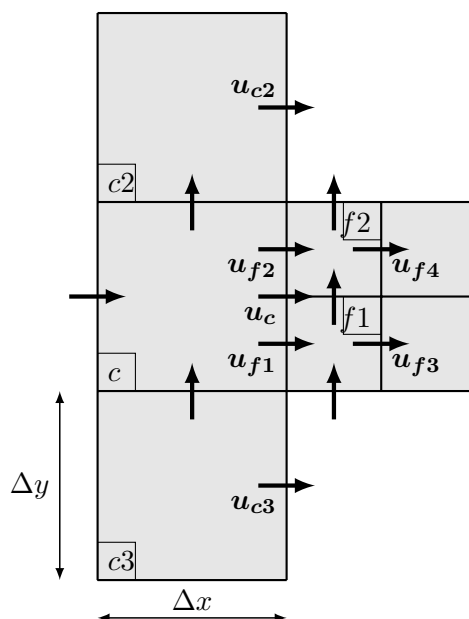


FIGURE 3.2 – Définition des vitesses aux interfaces de raffinement

Pour que le schéma conserve la masse, il faut que les vitesses aux interfaces respectent l'égalité des flux de masse, qui se traduit par :

$$u_c = \frac{1}{2}(u_{f1} + u_{f2}) \quad (3.1)$$

Les vitesses u_{f1}, u_{f2} et u_c peuvent être soit résolues soit obtenues par interpolation.

Un choix courant consiste à ne résoudre que les vitesses des cellules fines u_{f1}, u_{f2} et à déduire la vitesse de la cellule grossière u_c avec la relation de conservation de la masse (3.1).

Une deuxième approche consiste à résoudre uniquement la vitesse grossière u_c et à calculer les vitesses fines u_{f1}, u_{f2} de l'interface de raffinement. Différentes façons d'interpoler ces vitesses fines sont possibles. Nous détaillons ci-dessous quelques schémas d'interpolation afin de mettre en évidence les enjeux qui se cachent derrière. L'objectif est le calcul de la divergence de vitesse sur les cellules fines de l'interface de raffinement avec une précision acceptable.

Considérons la cellule $f2$ (Fig. 3.2). Le calcul de la divergence de vitesse sur cette cellule passe par l'évaluation au centre de la cellule de :

$$\frac{\partial u}{\partial x} \approx \frac{u_{f4} - u_{f2}}{\Delta x/2} \quad (3.2)$$

pour un schéma en différences finies. Les conclusions de cette étude resteront toutefois valables en volumes finis. Pour évaluer l'influence du choix de la méthode d'interpolation sur la précision du calcul, nous calculons un développement en séries de Taylor de (3.2) pour chaque possibilité :

- Si tout d'abord la vitesse est considérée comme constante le long de l'interface de raffinement, alors $u_{f2} = u_c$. Un développement en série de Taylor de u_c donne :

$$u_c = u_f - \frac{\Delta x}{4} \frac{\partial u}{\partial x} \Big|_f - \frac{\Delta y}{4} \frac{\partial u}{\partial y} \Big|_f + \frac{\Delta x}{4} \frac{\Delta y}{4} \frac{\partial^2 u}{\partial x \partial y} \Big|_f + \frac{\Delta x^2}{32} \frac{\partial^2 u}{\partial x^2} \Big|_f + \frac{\Delta y^2}{32} \frac{\partial^2 u}{\partial y^2} \Big|_f + O(\Delta x^3 + \Delta y^3) \quad (3.3)$$

où u_f est la vitesse au centre de la cellule $f2$. Par ailleurs :

$$u_{f4} = u_f + \frac{\Delta x}{4} \frac{\partial u}{\partial x} \Big|_f + \frac{\Delta x^2}{32} \frac{\partial^2 u}{\partial x^2} \Big|_f + O(\Delta x^3) \quad (3.4)$$

Le calcul de la divergence sur la cellule fine donne donc :

$$\frac{u_{f4} - u_{f2}}{\Delta x/2} = \frac{\partial u}{\partial x} \Big|_f + \frac{\Delta y}{2\Delta x} \frac{\partial u}{\partial y} \Big|_f + O\left(\Delta y + \frac{\Delta y^2}{\Delta x}\right) \quad (3.5)$$

Ce qui n'est pas consistant puisque $\Delta y = r\Delta x$ où r désigne le rapport d'aspect des cellules ($r = 1$ pour des cellules carrées). Cet exemple illustre l'idée, souvent avancée au sujet du raffinement de maillage, que les opérateurs discrets perdent inévitablement un ordre aux interfaces de raffinement (cf. annexe E pour l'illustration de cette perte d'ordre sur le calcul d'une matrice hessienne de pression). Pour remédier à cette inconsistance, l'erreur de reconstruction de u_{f2} doit être montée au moins à l'ordre 2.

- Un choix plus précis consiste à considérer une interpolation linéaire de u_{f2} à partir de u_{c2} et u_c :

$$u_{f2} = u_c + \frac{\Delta y}{4} \frac{u_{c2} - u_c}{\Delta y} = \frac{3}{4}u_c + \frac{1}{4}u_{c2} \quad (3.6)$$

De même pour u_{f1} on pourra envisager :

$$u_{f1} = u_c - \frac{\Delta y}{4} \frac{u_c - u_{c3}}{\Delta y} = \frac{3}{4}u_c + \frac{1}{4}u_{c3} \quad (3.7)$$

En outre un développement en série de Taylor permet d'exprimer u_{c2} en fonction des dérivées partielles au centre de la cellule $f2$:

$$u_{c2} = u_f - \frac{\Delta x}{4} \frac{\partial u}{\partial x} \Big|_f + \frac{3\Delta y}{4} \frac{\partial u}{\partial y} \Big|_f + O(\Delta x^2 + \Delta y^2) \quad (3.8)$$

D'où :

$$u_{f2} = \frac{3}{4}u_c + \frac{1}{4}u_{c2} = u_f - \frac{\Delta x}{4} \frac{\partial u}{\partial x} \Big|_f + O(\Delta x^2 + \Delta y^2) \quad (3.9)$$

Et le calcul de la divergence sur la cellule fine est alors consistant, d'ordre 1 :

$$\frac{u_{f4} - u_{f2}}{\Delta x/2} = \frac{\partial u}{\partial x} \Big|_f + O\left(\Delta x + \frac{\Delta y^2}{\Delta x}\right) \quad (3.10)$$

En revanche la masse n'est pas conservée puisque la relation (3.1) n'est pas respectée :

$$u_c = \frac{1}{2}(u_{f1} + u_{f2}) = \frac{3}{4}u_c + \frac{1}{8}(u_{c2} + u_{c3}) \neq u_c \quad (3.11)$$

La perte du caractère conservatif s'explique par la non-symétrie de l'interpolation. En effet la pente de l'interpolation linéaire est différente pour la cellule fine du haut $f2$ et la cellule fine du bas $f1$. Elle vaut $\frac{u_{c2}-u_c}{\Delta y}$ dans le premier cas et $\frac{u_{c3}-u_c}{\Delta y}$ dans le second.

- Pour rendre l'interpolation symétrique, il faut choisir une pente commune, comme par exemple $\frac{u_{c2}-u_{c3}}{2\Delta y}$ pour laquelle :

$$u_{f2} = u_c + \frac{\Delta y}{4} \frac{u_{c2} - u_{c3}}{2\Delta y} = u_c + \frac{1}{8}(u_{c2} - u_{c3}) \quad (3.12)$$

et

$$u_{f1} = u_c - \frac{1}{8}(u_{c2} - u_{c3}) \quad (3.13)$$

Le calcul de la divergence demeure à l'ordre 1 dans ce cas mais la conservation de la masse est garantie.

Il est possible de réaliser des reconstructions à des ordres supérieurs mais Minion a mis en évidence qu'une interpolation linéaire symétrique est le seul moyen de conserver la masse [Minion, 1996]. Il n'est pas possible d'avoir un opérateur compact qui soit à la fois d'ordre élevé et conservatif.

Enfin l'expérience numérique montre que malgré un ordre théorique de convergence de 1 au voisinage de l'interface de raffinement, le schéma numérique donne finalement de meilleurs résultats avec un ordre compris entre 1 et 2 [Losasso et al., 2004, Min et al., 2006, Min et Gibou, 2006].

Opérateur gradient ∇

Comme pour la grille uniforme, le schéma de projection exige que les gradients de pression soient évalués aux emplacements des variables de vitesses qui doivent être rendues à divergence nulle. Considérons la configuration de la figure 3.3.

Supposons d'abord, comme précédemment, que seules les vitesses fines de l'interface de raffinement sont résolues. Pour calculer le gradient sur la face de la vitesse u_{f2} , une approche classique est de reconstruire une pression p_g alignée avec p_{f2} et p_{f4} . Pour garantir un calcul de gradient à l'ordre 2, Popinet propose l'interpolation suivante [Popinet, 2003a] :

$$p_g = \frac{15}{16}p_c - \frac{3}{32}p_{c3} + \frac{5}{32}p_{c2} \quad (3.14)$$

Puis le gradient à l'ordre 2 est donné par :

$$\frac{\partial p}{\partial x} \Big|_{f2} \approx \frac{1}{\Delta x} \left(\frac{1}{3}p_{f4} + \frac{1}{5}p_{f2} - \frac{8}{15}p_g \right) \quad (3.15)$$

Cette formulation à l'avantage d'être rigoureusement à l'ordre 2. Néanmoins elle implique un stencil assez grand et doit être adaptée à toutes les configurations possibles comme par exemple des cellules fines à la place des cellules grossières $c2$ et $c3$. De plus le système linéaire à résoudre qui en découle n'est pas

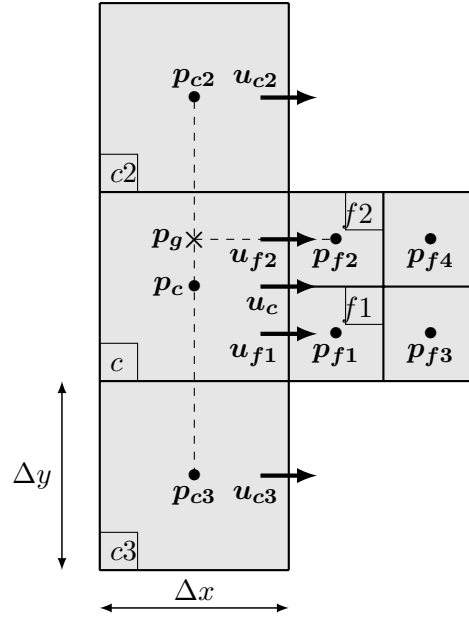


FIGURE 3.3 – Calcul des gradients de pression à une interface de raffinement.

symétrique. En effet le Laplacien de pression sur la cellule fine f_2 va dépendre de c_2 et f_4 alors que le Laplacien sur la cellule f_4 n'impliquera pas c_2 .

En notant que $p_g = p_c + O(\Delta x)$, Losasso et al. [Losasso et al., 2004] proposent une approche simple pour préserver la symétrie consistant à évaluer directement le gradient de pression par :

$$\left. \frac{\partial p}{\partial x} \right|_{f_2} \approx \frac{p_{f_2} - p_c}{3/4\Delta x} \quad (3.16)$$

Selon ses auteurs, cette méthode allège considérablement le stencil, ne pénalise pas trop la précision et favorise la résolution numérique du système linéaire en le rendant symétrique.

Par ailleurs, comme vu précédemment, il est aussi envisageable de ne résoudre que la vitesse grossière u_c à l'interface de raffinement. Le calcul du gradient de pression à l'emplacement de cette vitesse peut être effectué à l'ordre 2 par :

$$\left. \frac{\partial p}{\partial x} \right|_c \approx \frac{1}{\Delta x} \left[\frac{1}{4}(p_{f_1} + p_{f_2} + p_{f_3} + p_{f_4}) - p_c \right] \quad (3.17)$$

ou se limiter à l'ordre 1 par :

$$\left. \frac{\partial p}{\partial x} \right|_c \approx \frac{1}{\Delta x} \left[\frac{1}{2}(p_{f_1} + p_{f_2}) - p_c \right] \quad (3.18)$$

Pour calculer les gradients aux emplacements des cellules fines, la même démarche d'interpolation que celle présentée pour le calcul de l'opérateur divergence peut être employée.

Multigrille géométrique

Outre la définition des opérateurs, le préconditionnement de l'équation de Poisson doit également être adapté pour les grilles raffinées localement. Pour une approche multigrille géométrique, la hiérarchie des grilles successives doit être modifiée. Deux principales techniques sont référencées dans [Brown et Lowe,

2005]. La première méthode, appelée MLAT pour Multi Level Adaptive Technique [Bai et Brandt, 1987], consiste à définir la hiérarchie en regroupant les grilles de niveau identique. Elle convient bien pour le raffinement local par patch. Pour un maillage en quad/octree la hiérarchie de la méthode FAC (Fast Adaptive Composite grid) de McCormick est souvent préférée [McCormick, 1984]. Dans ce cas, les opérations de relaxation de la méthode multigrille sont conduites sur tout le domaine de calcul en définissant les grilles successives par restrictions jusqu'à la grille la plus grossière.

Méthode algébrique

Dans le cas où la résolution du système linéaire est accélérée par multigrille algébrique, aucune modification n'est requise puisque la méthode se base sur une analyse de graphe de la matrice du système linéaire.

3.2.2 Méthodes pour le calcul des flux

Dans la littérature, le traitement des flux de l'équation de quantité de mouvement avec des variables entrelacées sur des grilles raffinées localement reste moins représenté par rapport aux variables colocalisées. Néanmoins les propriétés de stabilité naturelle des schémas avec variables entrelacées encouragent leur extension sur des grilles raffinées localement. Dans notre revue des différentes techniques employées sur les maillages MAC avec raffinement local, nous avons retenu quatre méthodes potentiellement intéressantes pour le traitement des flux.

Méthode DDFV

La méthode DDFV pour Discrete Duality Finite Volume est une méthode de type volumes finis, introduite par Hermeline [Hermeline, 2000]. Elle a été développée pour des maillages non conformes (maillage avec des sommets au milieu des faces) et sans condition d'orthogonalité. L'idée est de construire un maillage dual à partir du maillage primaire en reliant les centres de gravité du maillage primaire. Les équations sont discrétisées sur les deux maillages. Le nombre d'inconnues est donc multiplié par 2. Un troisième maillage en diamant permet de lier les deux premiers maillages (Fig. 3.4).

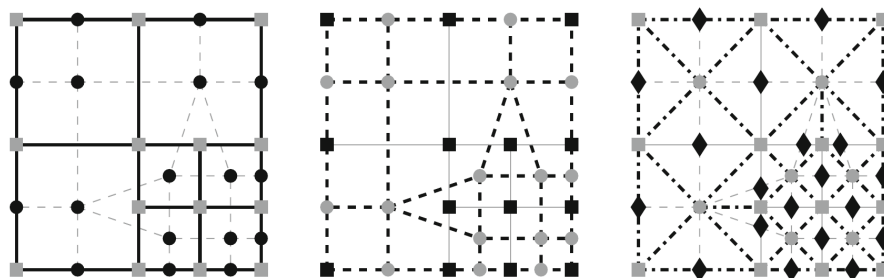


FIGURE 3.4 – Structure des volumes de contrôle. De gauche à droite, maillage initial, maillage dual, maillage diamant. Extrait de Hubert et al. [Hubert et Tesson, 2017]

La méthode DDFV a été étendue pour des schémas WENO d'ordre élevé [Hubert et Tesson, 2017]. Enfin même si les variables sont entrelacées, la méthode DDFV ne place pas exactement la pression et la vitesse aux mêmes points que le maillage MAC.

Méthode Voronoï

Une extension exacte de la méthode MAC sur des maillages raffinés localement a été proposée par Chénier et al. [Chénier et al., 2011, Chénier et al., 2015]. Les équations ne sont plus résolues directement sur

un maillage primaire mais sur un pavage de l'espace (triangulation de Delaunay ou diagramme de Voronoï). Guittet et al. [Guittet et al., 2015] ont introduit un schéma de projection de Chorin sur maillage octree de type MAC avec des volumes de contrôle définis par les cellules du diagramme de Voronoï associé.

Un diagramme de Voronoï est une décomposition de l'espace en cellules à partir d'un ensemble de points discrets. Chacun de ces points est lié à ses voisins les plus proches. Une cellule de Voronoï est définie par les plans bissecteurs aux segments qui joignent deux points les plus proches. Les cellules de Voronoï forment une tessellation de l'espace, et constituent une nouvelle partition de l'espace pour les volumes de contrôle des vitesses aux interfaces de raffinement (Fig. 3.5).

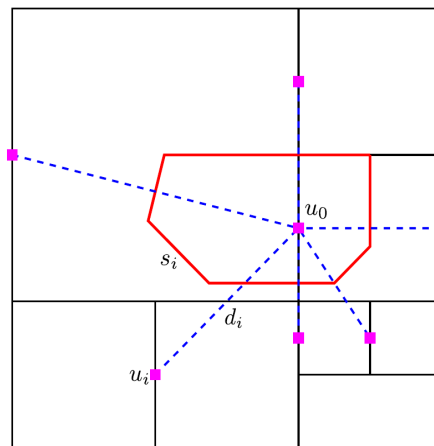


FIGURE 3.5 – Diagramme de Voronoï à une interface de raffinement. Extrait de Guittet et al. [Guittet et al., 2015]

Le diagramme de Voronoï a des propriétés attrayantes du fait de l'orthogonalité des faces des cellules par rapport aux segments qui lient les centres de gravité. Le calcul des flux aux faces est cependant limité à l'ordre 2. Dans [Guittet et al., 2015], la tessellation en diagramme de Voronoï est parallélisée à l'aide d'une librairie tierce, Voropp [Rycroft, 2009], sur tout le domaine de calcul et pas uniquement aux interfaces de raffinement.

Méthode des cellules de garde ou "guard-cells"

Une méthode courante pour la gestion des interfaces de raffinement sur maillage MAC est la méthode des cellules de garde (ou guard-cells en anglais) [Zuzio et Estivalezes, 2011, Liu, 2011]. Elle consiste à étendre le domaine de part et d'autre de l'interface de raffinement avec des cellules fictives de même niveau de raffinement. Les guard-cells ne sont pas directement résolues et avancées en temps. Leur rôle est d'étendre le domaine pour pouvoir appliquer les stencils d'interpolation des schémas numériques sans troncature aux interfaces de raffinement. Des valeurs de pression et de vitesse doivent être attribuées pour ces cellules à partir des cellules physiques. Des opérateurs d'interpolation permettent de transférer l'information depuis les cellules réelles vers les guard-cells. L'opération dite de "restriction" permet d'affecter des valeurs aux guard-cells à partir de cellules réelles situées à un niveau de raffinement supérieur. L'opération réciproque est appelée "prolongation" et permet de remplir des guard-cells à partir d'un maillage plus grossier (Fig. 3.6). Les restrictions et prolongations doivent être effectuées à chaque avance en temps.

Ainsi dès lors que les opérateurs de restriction et de prolongation ont été définis, une méthode initialement développée pour une grille cartésienne structurée peut tout à fait s'adapter à des maillages raffinés localement.

Pour conserver les flux de masse ou de quantité de mouvement il faut calculer dans un premier temps les flux des cellules fines puis celui des cellules grossières en moyennant celui des cellules fines (Fig. 3.7).

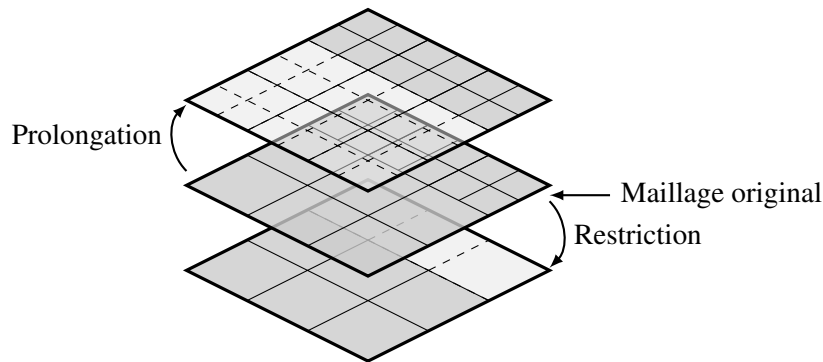


FIGURE 3.6 – Opérations de restriction et de prolongation

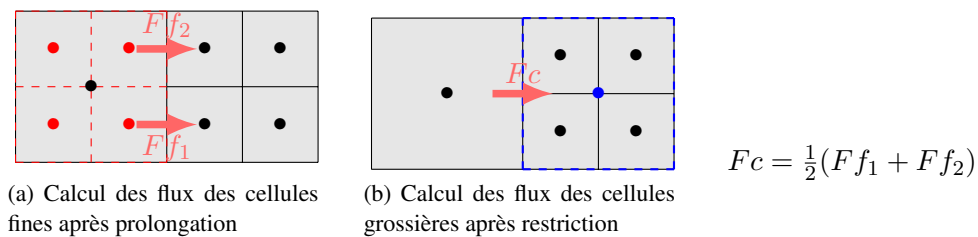


FIGURE 3.7 – Correction du flux grossier pour la conservation de la quantité de mouvement pour variables colocalisées

Dans la version faiblement-compressible, la conservation est garantie de cette façon.

Sur maillage MAC, la non-coïncidence des volumes de contrôle rend une telle approche inopérante. Une solution est d'inscrire la résolution des flux dans un contexte de résolution multigrille. La méthode FAC (Fast Adaptive Composite grid) de McCormick [McCormick, 1984] permet de hiérarchiser les grilles de niveaux de raffinement différents et d'assurer la conservation d'une grille à l'autre. La méthode FIC pour Flux Interface Correction proposée par Angot et al. [Angot et Laugier, 1994, Khadra et al., 1996] permet de satisfaire une relation de continuité des flux aux interfaces de raffinement. Un résidu est établi par la différence entre le flux de la cellule grossière et le flux des cellules fines à l'interface de raffinement. Une méthode multigrille permet de faire décroître ce résidu. La méthode FIC est comparée à la méthode FAC dans [Khadra et al., 1996].

Méthode de type stretching

Van der Plas [van der Plas, 2016] a proposé une méthode originale qui combine une technique d'étirement des volumes de contrôle (stretching) et le recours aux guard-cells. Le stretching est appliqué pour le volume de contrôle de la vitesse normale à l'interface (Fig. 3.8). Pour la direction tangentielle à l'interface, une procédure de type guard-cells est utilisée.

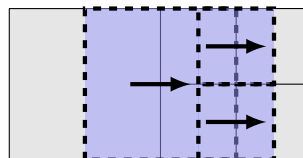


FIGURE 3.8 – Étirement du volume de contrôle de la vitesse de la cellule grossière

3.2.3 Bilan et choix d'une méthode

La scalabilité, la facilité d'implémentation et un bon compromis coût de calcul/précision sont les principaux critères de choix pour sélectionner une méthode.

Pour la résolution de l'équation de Poisson, notre choix se porte sur une vitesse aux interfaces de raffinement portée par les cellules grossières. Les gradients de pression et les vitesses seront interpolés sur les cellules fines de façon symétrique à partir de l'équation (3.13). En effet, cette technique permet de limiter le nombre d'éléments non nuls sur les lignes de la matrice du système linéaire tout en préservant la conservation de la masse (même si localement l'ordre chute à 1). Dans sa thèse [van der Plas, 2016], van der Plas a analysé les différentes configurations entre celle avec une vitesse portée par la cellule grossière ou celle avec une vitesse portée par les cellules fines sur différents cas tests dont des cas d'application marine (propagation de vagues). En se basant sur la précision et l'efficacité des deux techniques, il a finalement retenu la première qui est celle que l'on a choisie également. Par ailleurs nous maintenons la résolution du système linéaire par une méthode de type GMRES redémarré combinée avec une méthode de multigrille algébrique comme pour les grilles uniformes. La construction d'une méthode de multigrille géométrique scalable pour des maillages octree reste, encore aujourd'hui, une tâche technique ardue.

En ce qui concerne le calcul des flux, les deux premières méthodes exposées ci-dessus et reposant sur la méthode DDFV ou le diagramme de Voronoï ne sont pas très attrayantes dans notre contexte puisqu'elles nécessitent une opération coûteuse de remaillage dont on peut douter des performances parallèles. De plus, il faudrait redévelopper un solveur pour ces maillages non cartésiens.

La méthode basée sur les guard-cells est l'une des plus couramment employées. Notamment parce qu'elle s'interface bien avec une procédure multigrille pour la résolution des équations de quantité de mouvement ou l'équation de Poisson. Le principal inconvénient que l'on peut relever est la gestion mémoire des guard-cells qui peut devenir un peu lourde si le maillage devient trop hétérogène au cours de la simulation, avec un nombre important d'interfaces de raffinement (sachant que pour un schéma WENO5, au moins 3 cellules de type guard-cells sont par exemple nécessaires).

Enfin la dernière solution proposée par van der Plas est originale et semble répondre à nos exigences. Il resterait cependant à l'adapter pour calculer le terme convectif avec un schéma WENO 5, van der Plas ne la proposant que pour des schémas centrés d'ordre 2.

Finalement, il a été décidé d'évaluer, dans un premier temps la méthode de type guard-cells dont l'implémentation est assez rapide. De plus, les opérations de restriction et de prolongation sont indispensables pour envisager un raffinement dynamique du maillage. Ensuite une méthode originale proche de celle de [van der Plas, 2016], combinant approche de type stretching et guard-cells, a été développée pour être compatible avec les reconstructions WENO5. Cette méthode reste dans l'esprit de la méthode à pas adaptatif employée en formulation faiblement-compressible.

3.3 Discrétisation aux interfaces de raffinement

Nous détaillons dans cette partie la discrétisation de l'équation de Poisson et le calcul des flux convectifs et diffusifs.

3.3.1 Discrétisation et implémentation des opérateurs divergence et gradient

La discrétisation de l'équation de Poisson est commune pour les approches guard-cells et stretching. Prenons comme exemple l'interface de raffinement décrite par la figure 3.9.

Nous considérons d'abord la discrétisation de l'opérateur divergence. D'un point de vue algorithmique, nous disposons de deux listes numérotées : une liste des indices de cellule et une liste des indices de vitesse. Pour construire la matrice de divergence, un simple balayage des indices de cellule du maillage est effectué

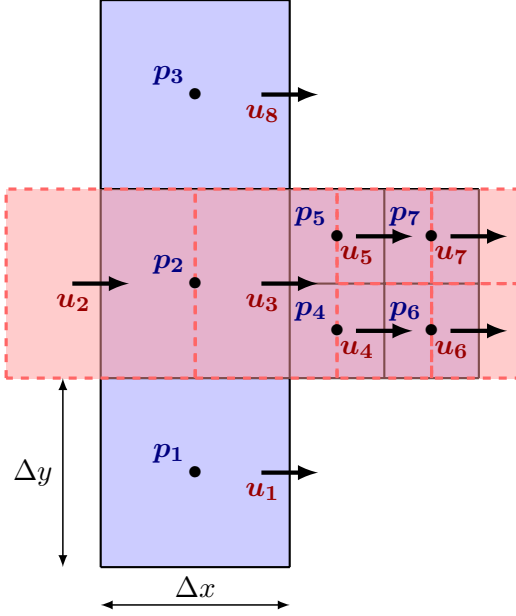


FIGURE 3.9 – Discrétisation des opérateurs divergence et gradient

(de couleur bleue sur la figure 3.9). Ainsi la matrice, notée M , est remplie ligne par ligne. Sous forme conservative elle s'écrit :

$$M = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{matrix} 2 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \left(\begin{array}{cccccccc} & -\Delta y & \Delta y & & & & & \\ & & -\Delta y/2 & \Delta y/2 & & & & \\ \frac{1}{8}\Delta y/2 & & -\Delta y/2 & & \Delta y/2 & & & -\frac{1}{8}\Delta y/2 \\ & & & -\Delta y/2 & & \Delta y/2 & & \\ & & & & -\Delta y/2 & & \Delta y/2 & \\ & & & & & & \Delta y/2 & \end{array} \right) \end{matrix} \quad (3.19)$$

Un remplissage colonne par colonne (c'est-à-dire en parcourant les indices de vitesses) serait sans doute plus optimal, en particulier au niveau des interfaces de raffinement. Cependant le code WCCH ne permet qu'un balayage des cellules puisqu'il a été conçu initialement pour des variables colocalisées. La mise en place d'une structure de données associant les indices de vitesse à chaque indice de cellule permettrait cependant de régler ce problème.

Les éléments de la matrice (3.19) en vert ne sont pas évalués lors du premier balayage. Ces termes permettent d'assurer la conservation de la masse à l'interface, et ne sont ajoutés que lors d'un deuxième balayage, après le calcul de la matrice de gradient que nous allons détailler ci-après. Ce balayage ne cible que les cellules situées aux interfaces de raffinement.

Pour la construction de la matrice du gradient G , nous utilisons à nouveau l'astuce mise en place sur les grilles cartésiennes et reposant sur la propriété d'anti-symétrie : $\nabla = -(\nabla \cdot)^T$. G est calculée directement à partir de :

$$G = \frac{-1}{Vol} M^T \quad (3.20)$$

où Vol désigne la matrice diagonale qui contient les volumes des volumes de contrôle des vitesses (en rouge sur la figure 3.9). Par exemple pour la cellule grossière à l'interface de raffinement, le volume vaut $3/4\Delta x\Delta y$. M^T désigne la matrice transposée de M et ne contient pas les termes correctifs pour assurer la conservation de la masse.

Par conséquent, pour l'exemple considéré sur la figure 3.9, G vaut :

$$G = \begin{matrix} & \begin{matrix} 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} \frac{1}{3/4\Delta x\Delta y} & & \\ & \frac{1}{\Delta x/2\Delta y/2} & \\ & & \frac{1}{\Delta x/2\Delta y/2} \end{pmatrix} \end{matrix} \times \begin{matrix} & \begin{matrix} 2 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} -\Delta y & \Delta y/2 & \Delta y/2 & & \\ & -\Delta y/2 & & \Delta y/2 & \\ & & -\Delta y/2 & & \Delta y/2 \end{pmatrix} \end{matrix} \quad (3.21)$$

3.3.2 Discrétisation des termes convectifs et diffusifs

Méthode guard-cells

Cette première méthode testée consiste à affecter des valeurs de vitesse aux cellules fictives pour étendre uniformément le maillage de part et d'autre de l'interface de raffinement. Les opérateurs de prolongation et de restriction sont définis uniquement pour les inconnues de vitesses situées aux faces des cellules (Fig. 3.10).

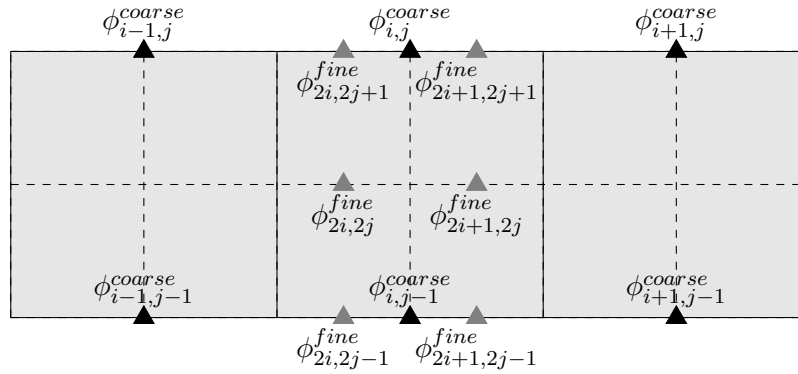


FIGURE 3.10 – Emplacement des variables des grilles fines ▲ et grossières ▲ pour les opérations de restriction-prolongation

La restriction transfère les variables d'une grille fine à une grille grossière. Une cellule grossière est issue de l'agglomération de 4 cellules plus fines (8 en 3D). D'après la figure 3.10, l'opération la plus simple pour remplir la variable grossière $\phi_{i,j}^{coarse}$ est d'effectuer une moyenne des variables fines qui partagent la même face :

$$\phi_{i,j}^{coarse} = \frac{1}{2}(\phi_{2i,2j+1}^{fine} + \phi_{2i+1,2j+1}^{fine}) \quad (3.22)$$

Les valeurs des cellules fines situées à l'intérieur de la cellule grossière $\phi_{2i,2j}^{fine}$ et $\phi_{2i+1,2j}^{fine}$ ne contribuent pas. On note que cette opération est d'ordre 2.

La prolongation est l'opération réciproque qui transfère des variables depuis les cellules grossières vers les cellules fines. L'approche la plus simple est d'injecter directement la valeur de la cellule grossière aux 4 cellules fines (8 en 3D).

$$\begin{cases} \phi_{2i+1,2j+1}^{fine} = \phi_{i,j}^{coarse} \\ \phi_{2i+1,2j}^{fine} = \phi_{i,j}^{coarse} \end{cases} \quad (3.23)$$

Mais cette méthode n'est pas très précise, or des reconstructions d'ordre plus élevé, (bi-linéaire ou bi-quadratique notamment) sont possibles, en s'appuyant sur un stencil de cellules grossières plus large. Balsara et Tóth [Balsara, 2001, Tóth et Roe, 2002] ont proposé des interpolations d'ordre élevé et qui préservent la divergence nulle des champs de vitesse interpolés.

Nous avons retenu un schéma d'interpolation linéaire symétrique. Les variables des cellules fines $\phi_{2i,2j+1}^{fine}$ et $\phi_{2i+1,2j+1}^{fine}$ qui partagent leur face avec celle de la cellule grossière $\phi_{i,j}^{coarse}$ sont d'abord interpolées de la même façon que pour le calcul de l'opérateur divergence (3.13). Les vitesses de cellules fines intérieures $\phi_{2i,2j}^{fine}$ et $\phi_{2i+1,2j}^{fine}$ sont ensuite calculées par une moyenne des vitesses fines précédemment fixées $\phi_{2i,2j+1}^{fine}$, $\phi_{2i+1,2j+1}^{fine}$, $\phi_{2i,2j-1}^{fine}$ et $\phi_{2i+1,2j-1}^{fine}$.

$$\begin{cases} \phi_{2i+1,2j+1}^{fine} = \phi_{i,j}^{coarse} + \frac{\phi_{i+1,j}^{coarse} - \phi_{i-1,j}^{coarse}}{8} \\ \phi_{2i+1,2j}^{fine} = \frac{\phi_{i,j}^{coarse} + \phi_{i,j-1}^{coarse}}{2} + \frac{\phi_{i+1,j}^{coarse} + \phi_{i+1,j-1}^{coarse} - \phi_{i-1,j}^{coarse} + \phi_{i-1,j-1}^{coarse}}{16} \end{cases} \quad (3.24)$$

Cette méthode préserve la conservation de la masse à la face de la cellule grossière :

$$\phi_{i,j}^{coarse} = \frac{1}{2}(\phi_{2i,2j+1}^{fine} + \phi_{2i+1,2j+1}^{fine}) \quad (3.25)$$

En pratique, l'opération de restriction est appliquée en premier lieu pour d'abord étendre le domaine grossier. Elle agit comme un filtre qui lisse les hautes fréquences. Cette méthode est donc supposée robuste. Puis la prolongation étend la grille de niveau supérieur.

Les opérations de restriction et de prolongation sont appelées à chaque pas de temps avant le calcul des flux. Comme 3 cellules de type guard-cells sont nécessaires pour étendre le stencil WENO5, au moins 6 cellules par bloc sont requises. Après restriction, ces dernières donneront 3 cellules guard-cells plus grossières. Cet aspect nous pousse à émettre une réserve sur l'utilisation des guard-cells dans notre algorithme dans la mesure où nous souhaiterions réaliser des simulations optimales avec un nombre de cellule minimal par bloc, c'est-à-dire 2. Ceci nous a invité à investiguer l'approche de type "stretching".

Méthode de type stretching

Nous distinguons par la suite le traitement de la vitesse normale à l'interface de raffinement de celui de la vitesse tangentielle à l'interface.

- Cas de la composante de vitesse normale à l'interface.

La méthode d'étirement permet de faire coïncider les volumes de contrôle de vitesses normales à l'interface de raffinement (Fig. 3.11) et présente un intérêt pour la conservation des flux de quantité de mouvement dans la direction normale.

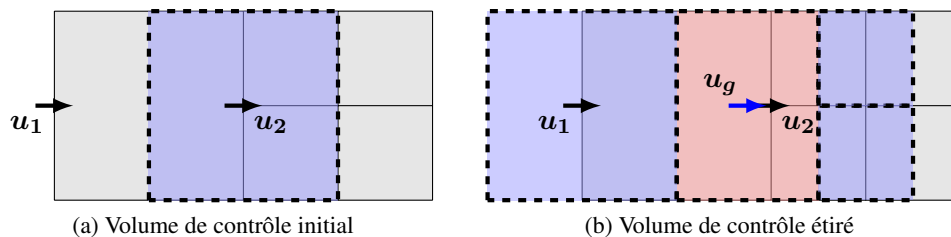


FIGURE 3.11 – Étirement du volume de contrôle associé à la vitesse d'une interface de raffinement

Dans notre approche, nous souhaitons faire coexister les deux volumes de contrôle. En effet, le volume de contrôle de base (en bleu, figure de gauche 3.11) et plus précisément sa vitesse associée u_2 conservent

leur intérêt pour servir de point d'appui au schéma d'interpolation (reconstructions WENO ou calcul des dérivées partielles de vitesse). Le nouveau volume de contrôle (en rouge, figure de droite 3.11) et sa vitesse associée u_g présente, lui, un intérêt pour l'évaluation des flux.

La méthode originale que nous proposons est de lier les volumes de contrôle des vitesses u_2 et u_g en limitant la chute d'ordre du schéma volumes finis.

Sur la figure 3.11 u_2 est la vitesse à l'interface de raffinement. Dans le cadre du schéma volumes finis, elle désigne la valeur moyenne du volume de contrôle de base. La valeur moyenne est en règle générale placée au centre du volume de contrôle car la valeur moyenne \bar{u}_2 et la valeur au centre u_2 sont assimilables avec un écart d'ordre 2 :

$$\bar{u}_2 = u_2 + \frac{\Delta x^2}{24} \frac{\partial^2 u}{\partial x^2} \Big|_2 + O(\Delta x^3) \quad (3.26)$$

Jusqu'à présent nous avons abandonné la notation \bar{u} au profit d'une simple écriture sous la forme u , pour désigner la valeur moyenne. Nous reprenons la convention $\bar{\cdot}$ par souci de clarté, le temps de l'exposé qui suit.

L'intégration, sur le nouveau volume de contrôle, d'un développement en série de Taylor de la vitesse au voisinage de l'interface donne :

$$\bar{u}_g = u_2 - \frac{\Delta x}{8} \frac{\partial u}{\partial x} \Big|_2 + \frac{3}{128} \Delta x^2 \frac{\partial^2 u}{\partial x^2} \Big|_2 + O(\Delta x^3) \quad (3.27)$$

On en déduit la relation suivante entre \bar{u}_2 et \bar{u}_g :

$$\bar{u}_g = \bar{u}_2 - \frac{\Delta x}{8} \frac{\partial u}{\partial x} \Big|_2 + \frac{13}{384} \Delta x^2 \frac{\partial^2 u}{\partial x^2} \Big|_2 + O(\Delta x^3) \quad (3.28)$$

Par conséquent, en approximant (grossièrement) la valeur du nouveau volume de contrôle (rouge) par la valeur du volume de contrôle de départ (bleu), $\bar{u}_g \approx \bar{u}_2$, le schéma chute à l'ordre 1 localement.

Pour renouer avec un ordre 2, il est nécessaire de recourir à une interpolation linéaire :

$$\bar{u}_g = \frac{7}{8} \bar{u}_2 + \frac{1}{8} \bar{u}_1 + O(\Delta x^2) \quad (3.29)$$

ou encore

$$\bar{u}_2 = \frac{8}{7} \bar{u}_g - \frac{1}{7} \bar{u}_1 + O(\Delta x^2) \quad (3.30)$$

Cette équation relie les volumes de contrôle de base et le volume de contrôle étiré. Elle s'applique sur les valeurs des vitesses moyennes mais également sur les variations de flux aux faces d'après l'équation semi-discrète (1.11).

En pratique, les reconstructions WENO5 et le calcul des dérivées partielles de vitesse sont donc effectués avec la vitesse u_2 comme point d'appui mais sont positionnées aux faces du nouveau volume de contrôle, c'est-à-dire là où sont calculés les flux. Puis les flux aux faces sont sommés sur les nouveaux volumes de contrôle et aux interfaces de raffinement, avant d'être "déplacés" sur les volumes de contrôle de base à l'aide de la relation (3.30). On pourra reprocher à l'équation (3.30) d'être une formule d'extrapolation plus que d'interpolation puisque \bar{u}_2 n'est pas localisée entre \bar{u}_g et \bar{u}_1 . Si des problèmes de stabilité apparaissaient il faudrait sans doute employer un autre stencil.

Cette approche est conservative et préserve l'ordre 2 du schéma.

Nous nous intéressons ensuite aux reconstructions WENO elles-mêmes. Prenons la situation dépeinte par la figure 3.12.

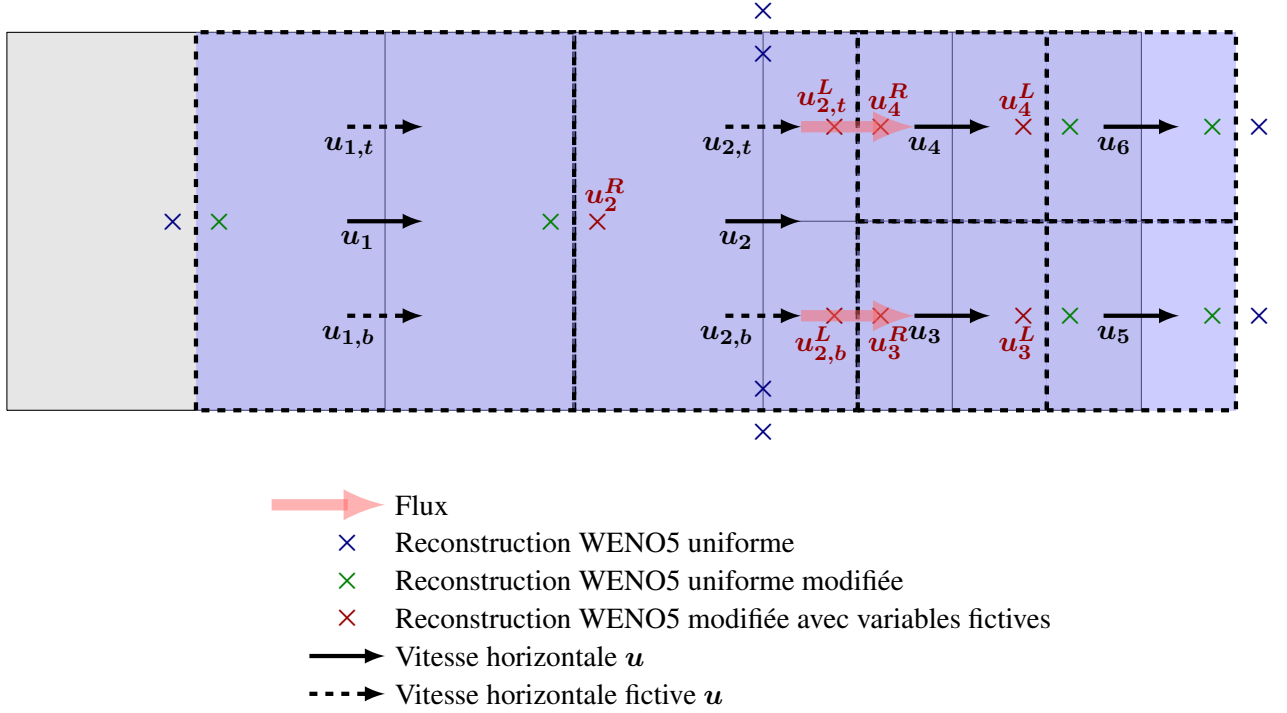


FIGURE 3.12 – Reconstruction WENO et calcul des flux dans la direction normale aux interfaces de raffinement.

Les emplacements pour lesquels les reconstructions WENO ne sont pas ambiguës et peuvent être réalisées comme sur un maillage cartésien uniforme sont représentés par une croix bleue. A mesure que l'on s'approche de l'interface de raffinement, la définition des stencils n'est plus immédiate. Par exemple pour le volume de contrôle de la vitesse u_1 , le stencil S_0 ($i, i + 1, i + 2$) n'est pas défini car il manque la vitesse associée $i + 2$ (Fig. 1.3). Nous pourrions définir un élément $i + 2$ fictif comme étant la moyenne de u_5 et u_6 (approche semblable aux guard-cells) mais nous avons privilégié une approche consistant à pénaliser le stencil S_0 en lui affectant une pondération nulle ($d_0 = 0$) comme pour le traitement des frontières du domaine. En effet, cette dernière est plus simple d'un point de vue algorithmique. Par conséquent, les valeurs reconstruites représentées par une croix verte ne s'appuient que sur 2 stencils au lieu de 3. L'ordre de l'erreur de reconstruction chute donc de 5 à 3. Pour les vitesses au plus près de l'interface u_2 , u_3 et u_4 , on ne peut appliquer la stratégie de pénalisation des stencils. Par exemple si nous faisons le choix de reconstruire des valeurs pour le volume de contrôle u_4 uniquement à partir de vitesses existantes, alors cela conduit à n'utiliser que le stencil S_0 . La valeur reconstruite à gauche u_4^L ne s'appuie alors que sur des valeurs situées en aval de sa position, ce qui n'est pas compatible avec un décentrement robuste. On attend en effet que u_4^L s'appuie sur des valeurs situées en amont. Il apparaît ainsi nécessaire de calculer des vitesses fictives pour disposer de points d'appui de part et d'autre des vitesses à reconstruire. Par exemple $u_{2,t}$ permet de définir un point d'appui en amont pour la reconstruction de u_4 . $u_{2,t}$ est obtenue selon le principe de l'interpolation linéaire symétrique (3.13). Puis u_4^L et u_4^R sont calculées à partir du stencil central S_1 seul :

$$\begin{cases} u_4^L = -\frac{1}{6}u_{2,t} + \frac{5}{6}u_4 + \frac{1}{3}u_6 \\ u_4^R = -\frac{1}{6}u_6 + \frac{5}{6}u_4 + \frac{1}{3}u_{2,t} \end{cases} \quad (3.31)$$

Sachant que $u_{2,t}$ est interpolée à l'ordre 2, il n'est pas nécessaire d'inclure plus de points d'appui (comme $u_{1,t}$ par exemple) pour monter l'ordre de reconstruction.

Pour les reconstructions (notées u_2^R , $u_{2,t}^L$ et $u_{2,b}^L$) aux faces du volume de contrôle étiré de la vitesse u_2 , nous mettons en place une procédure particulière. En effet nous rappelons que u_2 désigne la valeur moyenne du volume de contrôle non-étiré recouvrant partiellement les volumes de contrôle des cellules fines (cf. Fig. 3.1). La figure 3.13 illustre la problématique de la reconstruction de u_2^R .

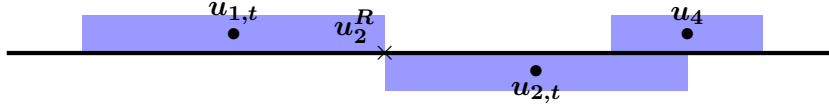


FIGURE 3.13 – Mise en évidence du stencil avec recouvrement des volumes de contrôle aux interfaces de raffinement.

Nous définissons donc un schéma d'interpolation s'appuyant sur des volumes de contrôle qui se recouvrent. La construction d'un tel schéma se fait en écrivant un développement en série de Taylor à l'ordre 2 au voisinage de l'emplacement du point de reconstruction. Cette expression est ensuite intégrée sur chaque volume de contrôle. Un système linéaire est résolu pour déterminer les pondérations de chaque vitesse afin d'annuler les erreurs jusqu'à l'ordre 2. Ces pondérations sont inscrites en dur dans le code.

Cela donne pour u_2^R :

$$u_2^R = -\frac{17}{66}u_1 + \frac{27}{22}u_2 - \frac{16}{33}\frac{u_4 + u_3}{2} \quad (3.32)$$

Pour $u_{2,t}^L$ et $u_{2,b}^L$ nous conservons la définition de ce stencil d'interpolation avec par exemple pour $u_{2,t}^L$:

$$u_{2,t}^L = -\frac{1}{12}u_{1,t} + \frac{3}{4}u_{2,t} + \frac{1}{3}u_4 \quad (3.33)$$

$u_{1,t}$ est encore une fois obtenu par interpolation linéaire symétrique.

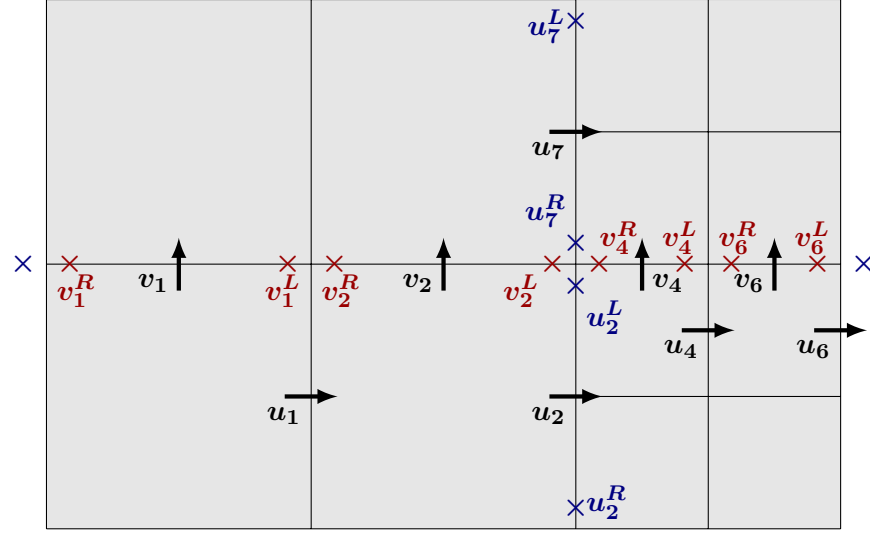
Le calcul des flux diffusifs dans la direction normale à l'interface de raffinement pose peu de difficultés puisque immédiatement :

$$\begin{cases} \left. \frac{\partial u}{\partial x} \right|_t \approx \frac{u_4 - u_{2,t}}{\Delta x/2} \\ \left. \frac{\partial u}{\partial x} \right|_b \approx \frac{u_3 - u_{2,b}}{\Delta x/2} \end{cases} \quad (3.34)$$

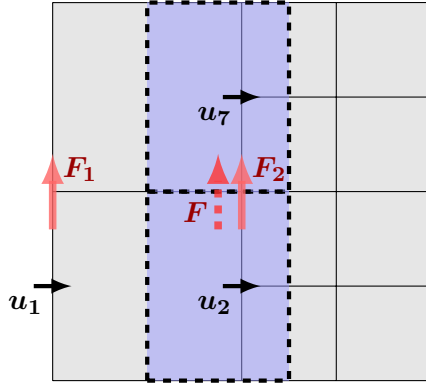
Nous avons présenté ci-dessus la façon de procéder pour le calcul des flux dans la direction normale de l'interface de raffinement. Les flux tangentiels sont plus faciles à évaluer grâce à l'alignement des vitesses u dans la direction tangentielle à l'interface et des vitesses v dans la direction normale (Fig. 3.14).

Pour le calcul du flux noté F_2 sur la figure 3.14, il faut dans un premier temps évaluer v_2^L et v_4^R pour le flux convectif. Un schéma WENO à pas variable, développé pour le schéma faiblement-compressible, est employé. Par exemple le stencil $S_0 (i, i + 1, i + 2)$ de la vitesse v_1 vaut :

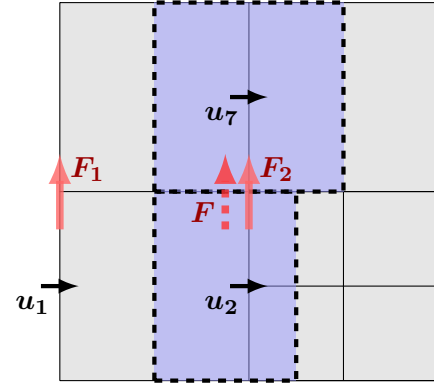
$$\begin{cases} v_1^L = \frac{3}{10}v_1 + \frac{29}{30}v_2 - \frac{4}{15}v_4 \\ v_1^R = \frac{19}{10}v_1 - \frac{43}{30}v_2 + \frac{8}{15}v_4 \end{cases} \quad (3.35)$$



(a) Reconstitutions WENO tangentielles



(b) Calcul du flux tangentiel



(c) Calcul du flux tangentiel dans un coin de raffinement

FIGURE 3.14 – Calcul du flux tangentiel à l'interface de raffinement

Pour v_2 et v_4 seul le stencil central S_1 est utilisé par commodité. S_2 et S_0 sont pénalisés en annulant les pondérations WENO associées $d_2 = 0$ et $d_0 = 0$. Pour v_2 , nous utilisons la formule d'interpolation suivante :

$$\begin{cases} v_2^L = -\frac{1}{10}v_1 + \frac{17}{30}v_2 + \frac{8}{15}v_4 \\ v_2^R = \frac{3}{10}v_1 + \frac{29}{30}v_2 - \frac{4}{15}v_4 \end{cases} \quad (3.36)$$

Pour v_4 :

$$\begin{cases} v_4^L = -\frac{1}{12}v_1 + \frac{17}{24}v_2 + \frac{3}{8}v_4 \\ v_4^R = \frac{1}{6}v_2 + \frac{13}{12}v_3 - \frac{1}{4}v_5 \end{cases} \quad (3.37)$$

Dans un deuxième temps, le calcul de la dérivée partielle de vitesse pour le flux diffusif est réalisé comme sur maillage structuré :

$$\frac{\partial u}{\partial y} \approx \frac{u_7 - u_2}{\Delta y} \quad (3.38)$$

Par ailleurs, les faces des volumes de contrôle ne coïncident pas nécessairement dans la direction tangentielle, et spécialement dans un coin de raffinement (Fig. 3.14). Dans le cas (b) de la figure 3.14, les surfaces coïncident mais le flux n'est pas évalué au centre de la face du volume de contrôle. En ne respectant pas la formule du point milieu, l'évaluation du flux n'est plus précise. Une interpolation linéaire remplace le flux au centre de la face :

$$F = \frac{7}{8}F_2 + \frac{1}{8}F_1 \quad (3.39)$$

Dans le cas (c) de la figure 3.14, il n'y a plus coïncidence des faces. A défaut d'une meilleure solution pour maintenir des intégrations correctes des flux, l'option choisie est malheureusement de ne plus conserver la quantité de mouvement, en prenant F pour flux sortant de u_2 et F_2 pour flux entrant dans u_7 .

- Cas de la composante de vitesse tangentielle à l'interface.

Dans la direction tangentielle, il n'est pas nécessaire de déformer des volumes de contrôle, puisque ceux-ci définissent déjà un découpage exact du domaine de calcul sans "recouvrement" ou "trou". La figure 3.14 illustre la configuration dans laquelle nous nous plaçons.

La détermination des flux (en gris sur la figure 3.15) dans la direction tangente à l'interface de raffinement se déroule comme sur grille cartésienne structurée, puisque les variables de vitesse v sont parfaitement alignées.

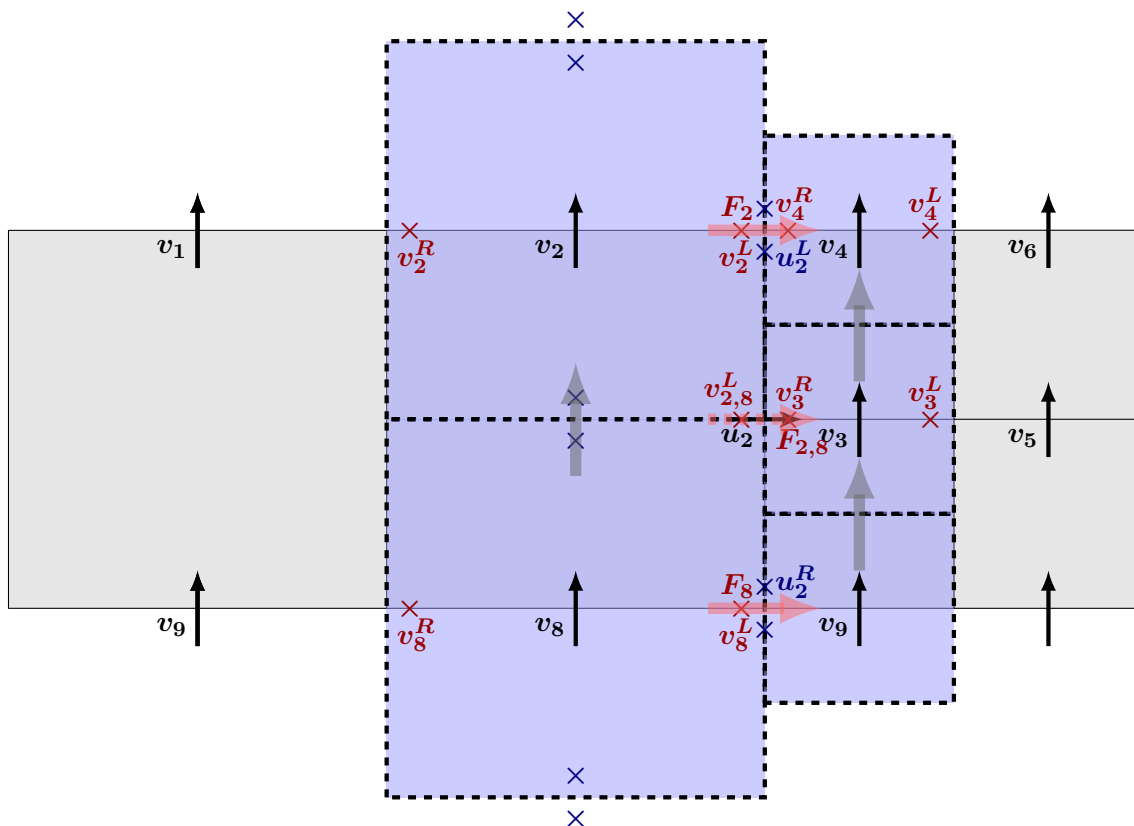


FIGURE 3.15 – Reconstruction WENO et calcul des flux dans la direction tangentielle aux interfaces de raffinement.

L'évaluation des flux normaux à l'interface de raffinement (en rouge sur la figure) est plus embarrassante. Pour le flux sortant du volume de contrôle de v_2 , toutes les informations sont disponibles pour le calcul des flux. v_2^L et v_4^R sont calculées suivant un schéma à pas spatial variable (équation (3.36) par exemple). La

dérivée partielle de vitesse, utile au flux visqueux, se calcule simplement à partir de :

$$\frac{\partial v}{\partial x} \approx \frac{(v_4 + v_6)/2 - v_2}{\Delta x} \quad (3.40)$$

Cependant les volumes de contrôle de v_2 et v_4 ne sont pas de taille identique. L'option retenue est de prendre le flux entrant dans v_4 comme égal au flux sortant de v_2 . Pour l'intégration, chaque flux est ensuite multiplié par la surface de la face du volume correspondant. Le rapport des surfaces entre la cellule grossière et la cellule fine est de $1/2$ en 2D et $1/4$ en 3D. Cette approche fait perdre la conservation de la quantité de mouvement de la vitesse v dans la direction normale à l'interface. De même nous calculons de cette façon le flux sortant de v_8 et celui entrant dans v_9 .

Pour le volume de contrôle de v_3 les choses sont différentes puisqu'il fait face à deux volumes de contrôle simultanément, v_2 et v_8 . La reconstruction WENO $v_{2,8}^L$ pose en premier lieu un problème. Nous utilisons la formule suivante qui s'appuie principalement sur des points en amont :

$$v_{2,8}^L = -\frac{1}{10} \frac{v_1 + v_9}{2} + \frac{17}{30} \frac{v_2 + v_8}{2} + \frac{8}{15} v_3 \quad (3.41)$$

v_3^R est reconstruite avec la formule d'interpolation suivante, identique à (3.37) :

$$v_3^R = \frac{1}{6} \frac{v_2 + v_8}{2} + \frac{13}{12} v_3 - \frac{1}{4} v_5 \quad (3.42)$$

Nous employons directement u_2 comme vitesse d'advection pour le flux convectif.

Pour le flux diffusif, la dérivée partielle de v est évaluée par :

$$\frac{\partial v}{\partial x} = \frac{(v_3 + v_5)/2 - (v_2 + v_8)/2}{\Delta x} \quad (3.43)$$

Le flux entrant dans v_3 , noté $F_{2,8}$ sur la figure 3.15, n'est rattaché à aucun flux sortant et enfreint aussi le principe de conservation.

Une approche conservative pourrait être mise en place. Elle consisterait à évaluer des flux sur des portions de face coïncidentes, (cf Fig. 3.15). Il faudrait alors interpoler le flux en ces points en utilisant une formule linéaire et symétrique comme par exemple :

$$F = F_{2,8} + \frac{F_2 - F_8}{\Delta y} \left(y - \frac{\Delta y}{2} \right) \quad (3.44)$$

- Extension en 3D (implémentée partiellement) :

Prétendre que l'extension en 3D de toute l'approche décrite ici est instantanée serait exagéré. Néanmoins elle ne demande finalement que peu d'adaptation. En reprenant les exemples qui ont été détaillés jusqu'à présent avec une interface de raffinement dans la direction de la composante u du vecteur vitesse, ajouter une dimension revient à ajouter une composante tangentielle w dont le traitement se fait d'une façon similaire au traitement de la composante v . Ainsi rien n'est nouveau, hormis des reconstructions tangentielles supplémentaires par rapport au 2D.

Par exemple pour le calcul des flux dans la direction normale à l'interface de u , il faut réaligner les points à l'aide d'une interpolation dans la direction tangentielle (Fig. 3.16).

Pour les composantes de vitesse tangentielles à l'interface de raffinement, une interpolation supplémentaire dans la direction tangentielle est également nécessaire pour le calcul des flux dans la direction normale par rapport au cas 2D (Fig. 3.17).

En définitive la méthode présentée ci-dessus est plus complexe d'un point de vue algorithmique par rapport à l'approche guard-cells. Nous avons pu constater qu'il serait possible de rendre la méthode totalement conservative. Néanmoins pour faciliter les interpolations et limiter le coût algorithmique nous avons

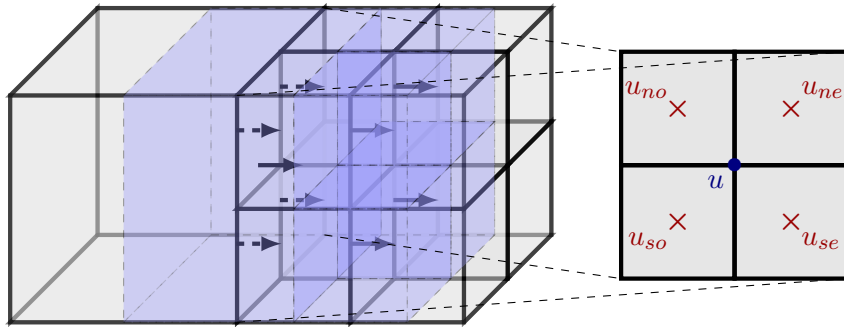


FIGURE 3.16 – Interpolations tangentielles dans une direction supplémentaire en 3D pour les vitesses u symbolisées par \dashrightarrow

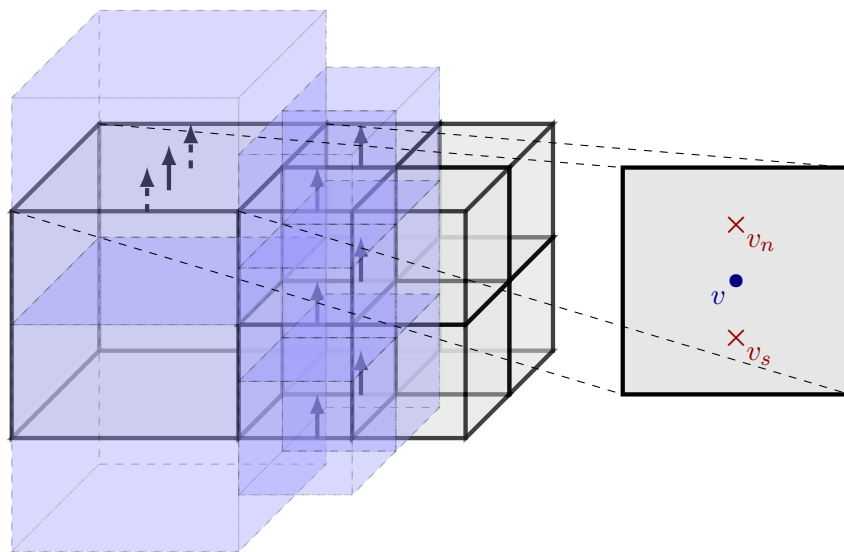


FIGURE 3.17 – Interpolations tangentielles dans une direction supplémentaire en 3D pour les vitesses v symbolisées par \dashrightarrow

fait certains choix simplificateurs qui ont des impacts sur la précision de la méthode. Par ailleurs nous ne pouvons plus garantir l'efficacité du schéma WENO pour la capture de choc au voisinage des interfaces de raffinement. En effet, pour ces zones du maillage, un seul stencil est reconstruit parmi les trois possibles, ne laissant qu'une option dans le choix du polynôme.

3.4 Conclusion

Le solveur incompressible présenté dans le chapitre 2 a été étendu sur grille cartésienne raffinée localement.

La résolution de l'équation de Poisson sur une telle grille ne pose pas de nouvelles difficultés, dans la mesure où ce problème est bien documenté dans la littérature. En effet, le Laplacien de pression demeure le même quelle que soit la disposition des variables, colocalisée ou entrelacée (maillage MAC). Nous avons choisi une discrétisation du gradient et de la divergence qui minimise la perte de précision aux interfaces de raffinement tout en limitant la taille du stencil. Par ailleurs, la symétrie du système linéaire est maintenue sur les grilles raffinées localement bien que nous n'exploitions pas cette propriété puisque nous résolvons l'équation de Poisson avec une méthode de type GMRES préconditionné. De plus, nous avons choisi de

conserver la résolution du système linéaire par une méthode multigrille algébrique. Cette première approche nous a permis de faire l'économie du développement (long et complexe) d'une méthode multigrille géométrique parallèle.

Par ailleurs, l'emplacement des vitesses aux faces du maillage physique complique l'évaluation des flux par rapport à une approche avec la vitesse au centre des cellules comme en formulation faiblement-compressible. L'approche la plus répandue pour faire face à ce type de problème est la technique des cellules de type guard-cells qui étendent le maillage de manière uniforme à l'interface de raffinement. Les opérateurs de restrictions et de prolongations transfèrent l'information entre deux grilles séparées d'un niveau de raffinement. Dès que ces opérations d'interpolation sont définies, le solveur de grille uniforme s'applique sans limitation. Nous avons donc retenu cette méthode comme première approche pour notre solveur incompressible. Néanmoins, peu satisfait du nombre important de cellules de type guards-cells générées pour traiter les stencils longs (WENO5), nous avons cherché une méthode plus efficace de ce point de vue. Nous proposons ainsi une méthode qui combine un étirement des volumes de contrôle dans la direction normale à l'interface de raffinement et une technique assimilable aux guards-cells dans la direction tangentielle. Cette méthode diffère de ce qui existe dans la littérature, de par la nature des interpolations réalisées ainsi que la volonté de maintenir un ordre de convergence proche de 2.

Chapitre 4

Validations numériques et comparaisons incompressible/faiblement-compressible

Dans ce chapitre nous validons l'approche incompressible présentée dans les chapitres précédents sur différents cas de complexité croissante et pour lesquels nous disposons de solutions analytiques ou de résultats de référence. Une première partie est consacrée à la validation des schémas numériques sur des grilles cartésiennes uniformes. Nous présentons ensuite une validation de l'extension du solveur sur les grilles raffinées localement avec une comparaison des deux méthodes retenues (la méthode "guard-cells" et la méthode de type "stretching"). Enfin, une section conséquente est vouée à la comparaison entre les versions faiblement-compressible et incompressible du code WCCH. Différents critères sont examinés sur des cas variés, principalement la diffusion numérique, l'ordre de convergence et la scalabilité. Enfin, les cas étudiés se limitent à des écoulements 2D, l'implémentation 3D de la formulation incompressible sur grille raffinée localement n'étant pas totalement validée.

4.1	Validation sur grille cartésienne uniforme	105
4.1.1	Précision et diffusion numérique	106
4.1.2	Solveur visqueux	108
4.2	Validation sur grille raffinée localement	113
4.2.1	Comparaison guard-cell/stretching	113
4.2.2	Écoulement laminaire autour d'un cylindre	121
4.3	Comparaison des formulations incompressible et faiblement-compressible	125
4.3.1	Diffusion numérique et précision	125
4.3.2	Écoulement de Taylor-Couette	128
4.3.3	Écoulement autour d'un profil NACA	132
4.3.4	Discussion sur les éléments de calcul haute performance	139
4.4	Conclusion	142

4.1 Validation sur grille cartésienne uniforme

Nous testons l'implémentation de la formulation incompressible sur différents écoulements visqueux/non-visqueux et stationnaires/instationnaires afin de valider l'ensemble des "briques" numériques pour des grilles cartésiennes uniformes : schéma WENO5, flux visqueux, équation de Poisson et schéma d'avance en temps.

4.1.1 Précision et diffusion numérique

Afin d'évaluer la diffusion numérique du schéma, nous étudions un écoulement 2D isovolume pour lequel une solution analytique est disponible : la convection d'un tourbillon dans un courant. Dans ce cas l'écoulement est périodique, ce qui permet de s'affranchir des contraintes liées aux conditions aux limites, et notamment du problème de couche limite artificielle créée par des conditions de Dirichlet sur la vitesse. Le benchmark de la convection de tourbillon tel qu'il a été proposé par le CERFACS [CERFACS, 2011] est un cas test relativement difficile dans la mesure où il met en défaut les schémas trop diffusifs. Un tourbillon de rayon R_c est convecté dans une boîte périodique à une vitesse U_0 de $35m/s$. L'écoulement est non-visqueux de sorte que le tourbillon doit se conserver parfaitement. Ce cas test est discriminant et permet de sélectionner les codes performants qui seront par exemple en mesure de modéliser finement des écoulements turbulents ou des sillages.

Paramétrisation du calcul

Le cas test est initialisé avec les valeurs de champ suivantes (Fig. 4.1) :

$$\begin{cases} u(x, y) = U_0 + \frac{\partial \Phi}{\partial y} \\ v(x, y) = -\frac{\partial \Phi}{\partial x} \\ p(x, y) = -\frac{\rho_0 \Phi^2}{2R_c^2} \end{cases} \quad (4.1)$$

$$\text{avec } \Phi(x, y) = \begin{cases} \Gamma e^{-\frac{(x-x_c)^2+(y-y_c)^2}{2R_c^2}} & \text{si } \sqrt{(x-x_c)^2+(y-y_c)^2} < 4R_c. \\ 0 & \text{sinon} \end{cases}$$

Les paramètres sont fixés à : $U_0 = 35m/s$, $L = 0.3112m$, $R_c = \frac{L}{20}$, $\rho_0 = 1.7170kg/m^3$ et $\Gamma = 0.0359157m^2 \cdot s^{-1}$, le fluide considéré étant de l'air. Le nombre CFL est fixé à 0.6.

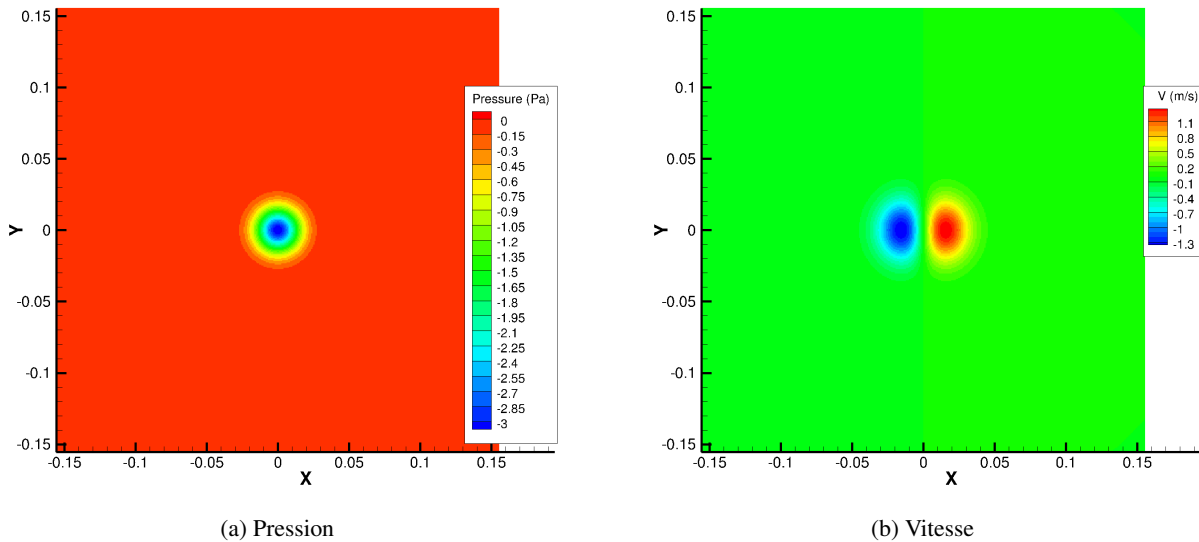


FIGURE 4.1 – Champs initiaux du cas de convection de tourbillon

Le tourbillon effectue 40 passages au terme desquels les profils de pression et de vitesse sont relevés. Plusieurs résolutions spatiales sont testées : 32, 64, 128 et 256 cellules par direction. Les résultats sont

présentés sur la figure 4.2. La pression affichée est celle de la dernière sous-étape du schéma Runge-Kutta. Il s'agit donc d'une pression d'ordre minimal.

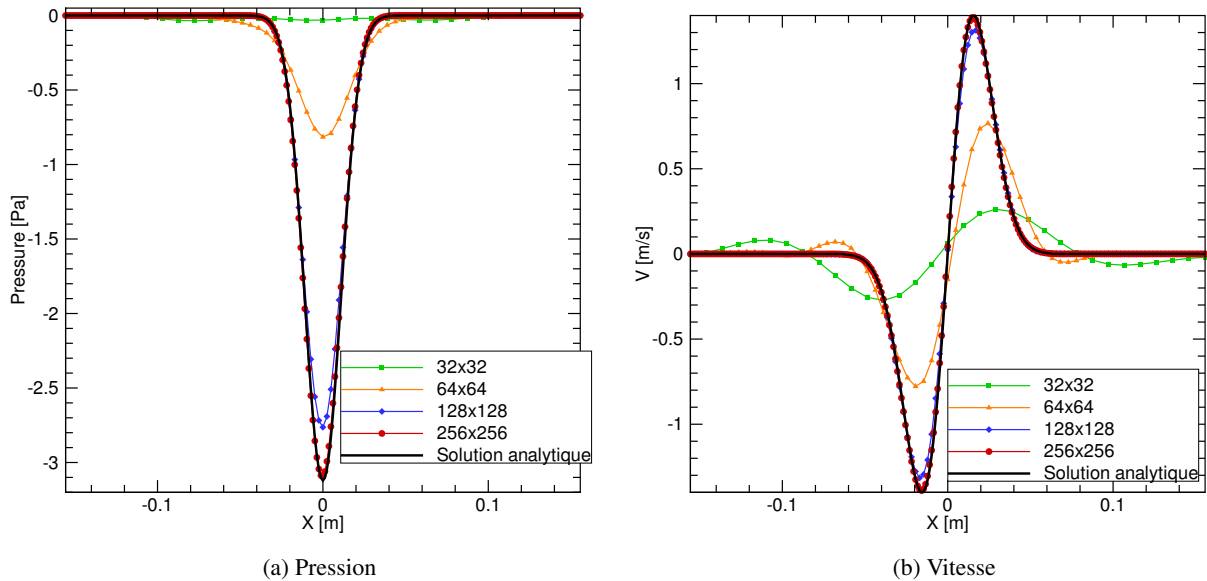


FIGURE 4.2 – Comparaison des profils de pression (à gauche) et de vitesse (à droite) au bout de 40 passages

A partir d'une résolution de 128×128 cellules, l'erreur en amplitude sur les extrema de vitesse et de pression devient inférieure à 10%. Elle est négligeable sur le maillage avec 256 cellules par direction. Ceci valide l'implémentation du schéma incompressible et en particulier le schéma WENO5.

Afin de positionner notre solveur incompressible, nous comparons nos résultats avec ceux issus d'autres codes, disponibles sur le site du CERFACS.

Comparaison avec les autres codes du benchmark

Les codes du benchmark sont elsA (ONERA), AVBP (CERFACS), OpenFOAM (Imperial College) et YALES2 (Coria). elsA est un code volumes finis d'ordre 6 en espace et en temps. AVBP est un code éléments finis d'ordre 3 en espace et en temps. OpenFOAM est un code volumes finis, il est ici utilisé dans sa version compressible avec un schéma centré d'ordre 4 à viscosité artificielle. L'intégration en temps est d'ordre 2 avec un schéma de type PISO. Enfin YALES2 est un code volumes finis d'ordre 4 en espace et en temps avec un schéma spatial centré à viscosité artificielle (Tab. 4.1).

Solveurs	Méthodes	Schéma spatial	Schéma temporel	CFL	Pas de temps [s]
elsA	Volumes finis, compressible	Compact ordre 6	Explicite RK ordre 6	0.7	5.0E-06
AVBP	Elements finis Taylor-Galerkin, incompressible	Ordre 3	Ordre 3	0.7	5.17E-06
OpenFOAM	Volumes finis, compressible	Centré ordre 4	Implicit ordre 2	0.3	7.5E-06
YALES2	Volumes finis, incompressible	Centré ordre 4	Explicite ordre 4	0.7	3.21E-05
WCCH (incompressible)	Volumes finis, incompressible	WENO 5	Explicite RK4	0.7	7.5E-05
WCCH (faiblement-compressible)	Volumes finis, faiblement-compressible	WENO 5	Explicite RK4	0.7	7.05E-06

TABLE 4.1 – Caractéristiques des codes du benchmark de la convection de tourbillon

Tous les codes du benchmark sont en formulation compressible, à l'exception de YALES2. Sur la figure 4.3, nous présentons les champs de vitesse des différents codes après 40 passages du tourbillon et pour

une résolution de 80×80 cellules. Nous jugeons pertinent d'ajouter les résultats obtenus en faiblement-compressible (avec une vitesse du son nominale $c_0 = 350m/s$ même si les deux codes seront plus abondamment comparés sur cet écoulement par la suite (en calculant notamment les ordres de convergence).

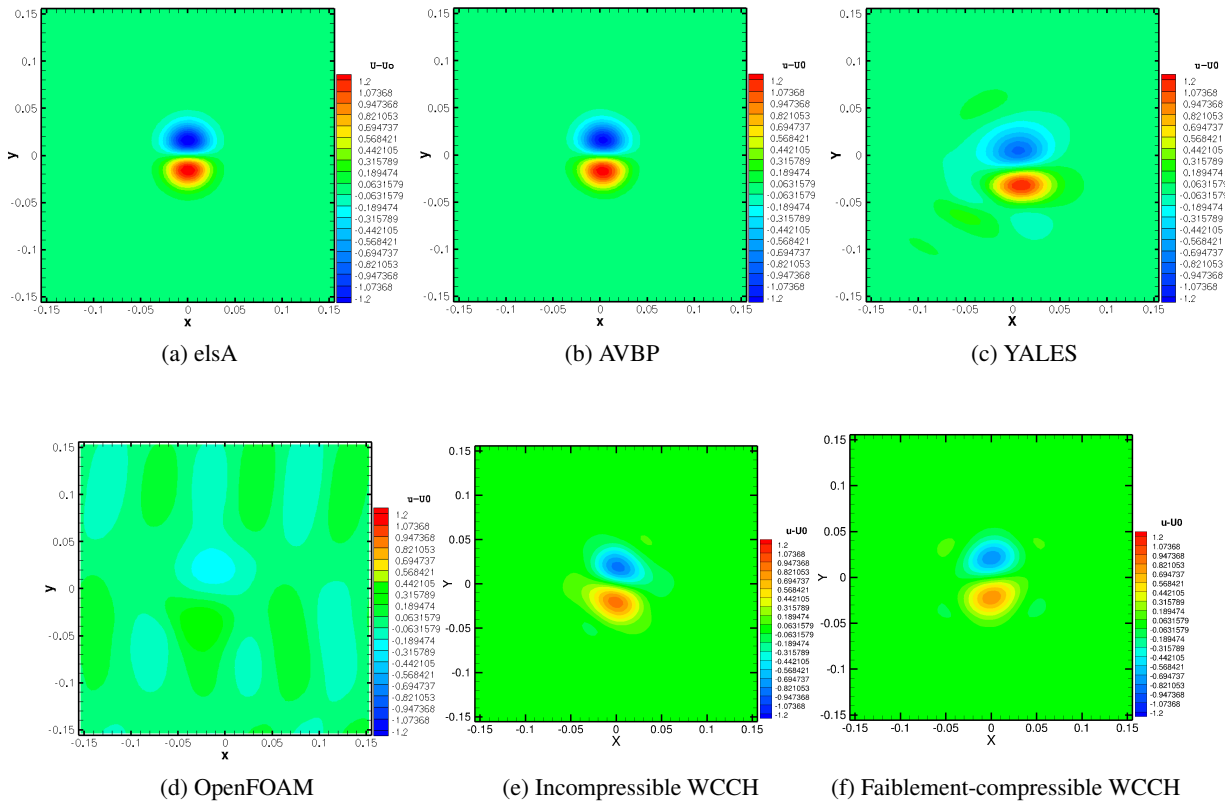


FIGURE 4.3 – Comparaison des champs de vitesse horizontale pour les différents codes au bout de 40 passages

elsA et AVBP donnent d'excellents résultats : le tourbillon est très peu altéré alors que les deux codes à viscosité artificielle (OpenFOAM et YALES2) atténuent le profil de vitesse (Fig. 4.4) et déforment le tourbillon (Fig. 4.3). Toute la complexité de ces solveurs à viscosité artificielle est d'ajuster le coefficient de diffusion à une juste valeur pour stabiliser au mieux le schéma centré sans pour autant diffuser complètement la solution (voir annexe A pour un exemple d'implémentation dans le code WCCH de schémas à viscosité artificielle).

Les versions faiblement-compressible et incompressible de WCCH donnent des résultats intermédiaires entre les solveurs à viscosité artificielle et les solveurs d'ordre élevé (elsA et AVBP). Le profil incompressible est cependant moins altéré que le faiblement-compressible. La version incompressible est environ 6 fois plus rapide.

4.1.2 Solveur visqueux

Écoulement de Poiseuille périodique bidimensionnel

Ce cas test classique représente l'écoulement laminaire stationnaire d'un fluide dans une conduite et pour lequel le profil de vitesse attendu est parabolique. Dans notre cas, l'écoulement est supposé bidimensionnel et une condition de périodicité est placée en entrée et en sortie du canal. Cela permet d'avoir un écoulement

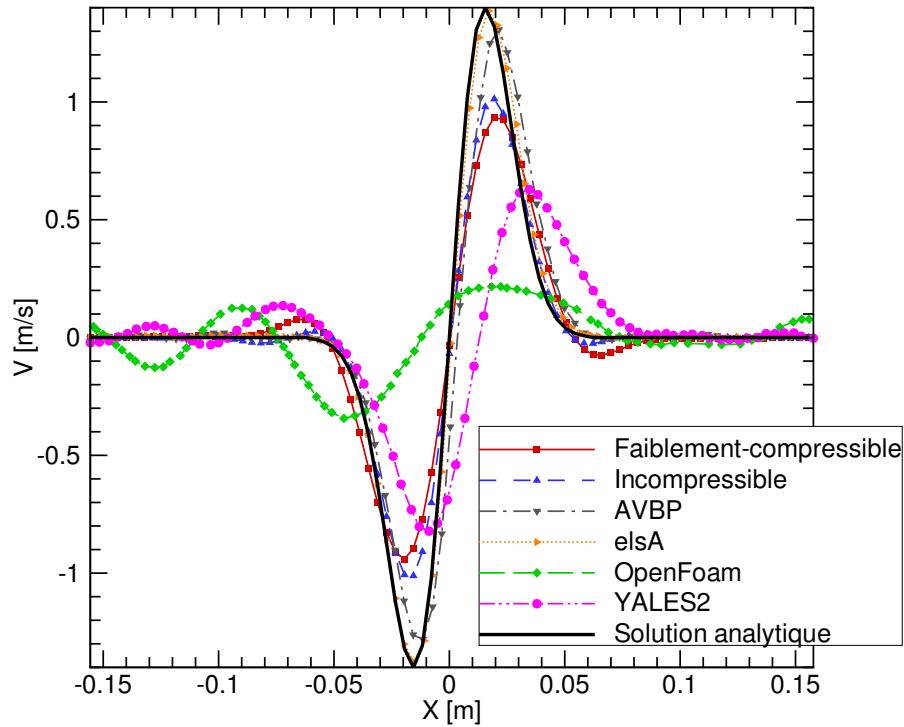


FIGURE 4.4 – Comparaison des profils de vitesse V suivant x au 40^{ème} passage pour un maillage 60×60

parfaitement établi, contrairement au cas où des conditions entrante et sortante sont définies. Le fluide est mis en mouvement sous l'action d'un gradient de pression extérieur.

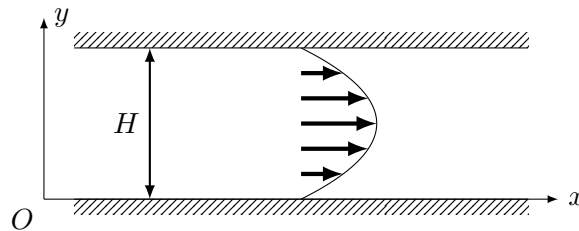


FIGURE 4.5 – Écoulement de Poiseuille

Soit L la longueur du canal et H sa hauteur. La solution analytique est (Fig. 4.5) :

$$u(y) = \frac{\nabla p}{2\mu} y(H - y) \quad (4.2)$$

Le gradient de pression est fixé suivant la vitesse maximale voulue dans l'écoulement :

$$\nabla p = \frac{\partial p}{\partial x} = \mu \frac{8u_{max}}{H^2} \quad (4.3)$$

Le calcul est effectué avec un maillage de 32×23 cellules avec $L = 1m$, $H = 1m$, $u_{max} = 1m/s$, $\rho = 1kg/m^3$ et $\mu = 1kg/(m.s)$ de sorte que le nombre de Reynolds $Re = \frac{\rho u_{max} H}{\mu} = 1$.

Le profil de vitesse obtenu est comparé avec le profil parabolique théorique (Fig. 4.6).

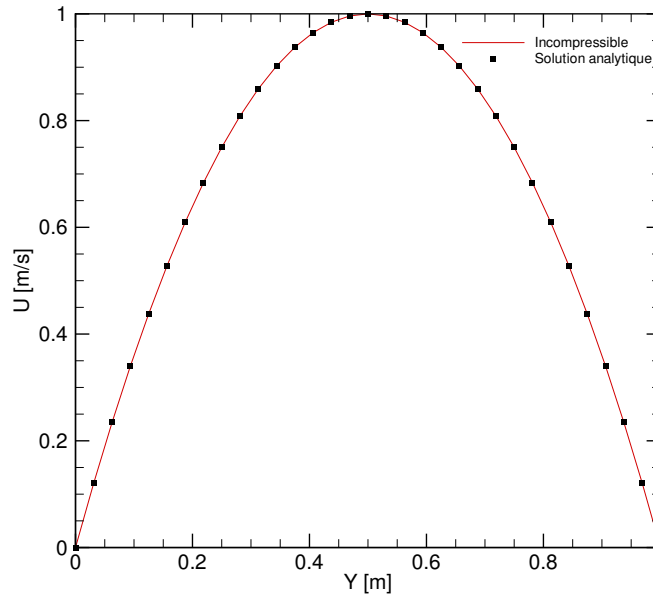


FIGURE 4.6 – Écoulement de Poiseuille

Le résultat est en accord parfait avec la solution analytique, et permet de valider les développements effectués sur le solveur visqueux.

Écoulement dans une cavité entraînée

Le cas test d'un fluide confiné dans une cavité et entraîné par la paroi supérieure à une vitesse constante imposée (Fig. 4.7) est un cas test classique de validation des codes CFD et est abondamment documenté dans la littérature [Ghia et al., 1982, Bruneau et Saad, 2006, Botella et Peyret, 1998]. Le calcul peut être conduit pour différents nombres de Reynolds, en faisant varier les caractéristiques du fluide. Pour des nombres de Reynolds modérés (jusque 8000 environ), l'écoulement est stationnaire et un tourbillon central se forme, accompagné de recirculations (simples, doubles voire triples) dans les coins [Bruneau et Saad, 2006]. Le nombre, la taille et l'emplacement des tourbillons dépendent du nombre de Reynolds. L'intérêt de ce cas est de démontrer la capacité du code à simuler des écoulements laminaires et à capturer des recirculations, difficiles à obtenir si le code n'est pas suffisamment précis. Les résultats obtenus seront comparés à ceux de Botella et Peyret [Botella et Peyret, 1998] qui utilisent une méthode spectrale de type Chebyshev. Un développement asymptotique des équations de Navier-Stokes dans les coins de la cavité leur permet aussi de capturer avec une grande précision les singularités de vitesse en ces points. Les résultats du schéma d'ordre 3 de Bruneau et Saad [Bruneau et Saad, 2006] pour ce cas constituent également une bonne référence.

Le nombre de Reynolds est ici fixé à 1000 ($\rho = 1000 \text{ kg/m}^3$, $\mu = 1 \text{ Pa}\cdot\text{s}$, $L = 1 \text{ m}$, $U = 1 \text{ m/s}$). Le fluide est initialement au repos, puis mis en mouvement par la paroi mobile. La simulation est conduite sur une durée physique suffisamment longue (120 secondes) pour atteindre l'état stationnaire. Le domaine est discrétisé dans un premier temps par 128×128 cellules.

La figure 4.8 compare le champ de vorticit  ($\omega = \nabla \wedge \mathbf{u}$) obtenu par notre sch ma avec celui issu de Botella et Peyret. Les discr tisations sont identiques (128×128 cellules) ainsi que l' chelle des isocontours. Dans notre cas, le champ de vorticit  est  valu  num riquement au centre des cellules   l'aide d'un sch ma centr  de diff rences finies d'ordre 2. Nous avons retourn  la figure de Botella et Peyret pour se placer dans une configuration o  la paroi sup rieure est anim e d'un mouvement de la gauche vers la droite.

Nous observons une excellente similitude entre les deux champs de vorticit , y compris aux coins de la cavit .

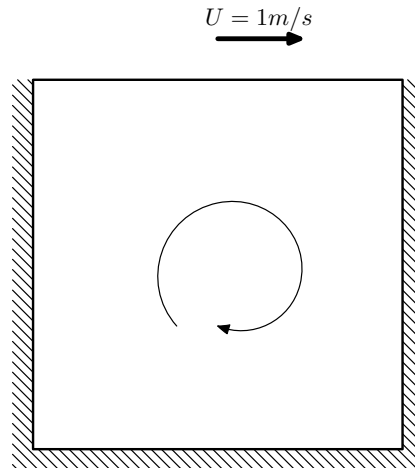
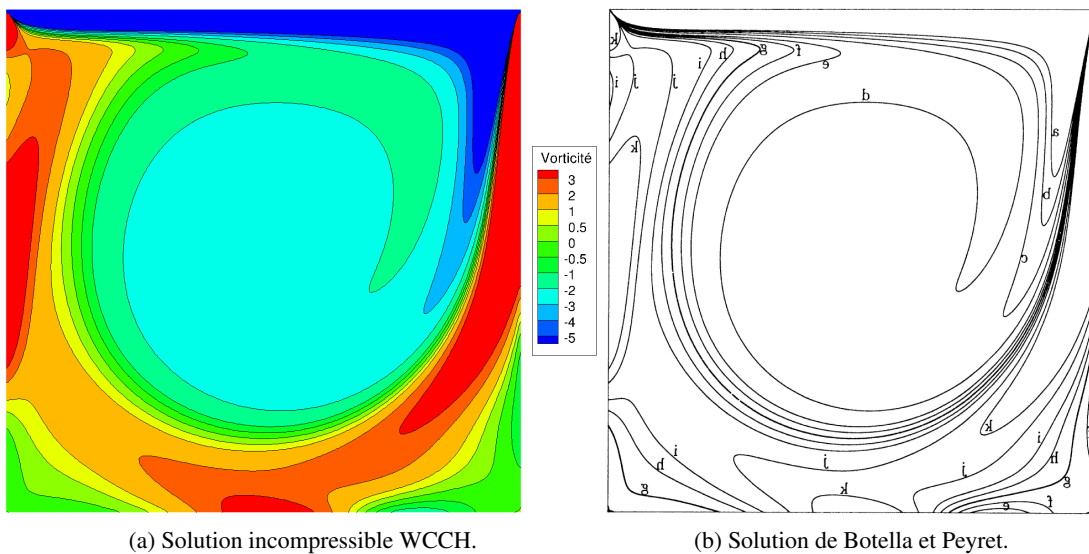


FIGURE 4.7 – Cavité entraînée

FIGURE 4.8 – Iso-vorticité de la cavité entraînée à $Re = 1000$ pour un maillage 128×128

Nous examinons ensuite les profils de vitesse horizontaux pris en $y = 0.5m$ et verticaux pris en $x = 0.5m$. Un premier maillage de 128×128 permet de constater que les profils de vitesse obtenus sont très proches des références (Fig. 4.9). Le profil de Botella et Peyret est issu d'une discrétisation 128×128 alors que celui de Bruneau et Saad vient d'un maillage beaucoup plus fin, de 1024×1024 . Les extrema de vitesse des profils de notre schéma restent néanmoins insuffisamment capturés, un maillage d'au moins 256×256 cellules est nécessaire pour être en accord avec les références. Un zoom autour des minima de vitesse est présenté figure 4.9.

Une autre grandeur souvent évaluée est la fonction courant ψ définie par :

$$\begin{cases} u = \frac{\partial \psi}{\partial y} \\ v = -\frac{\partial \psi}{\partial x} \end{cases} \quad (4.4)$$

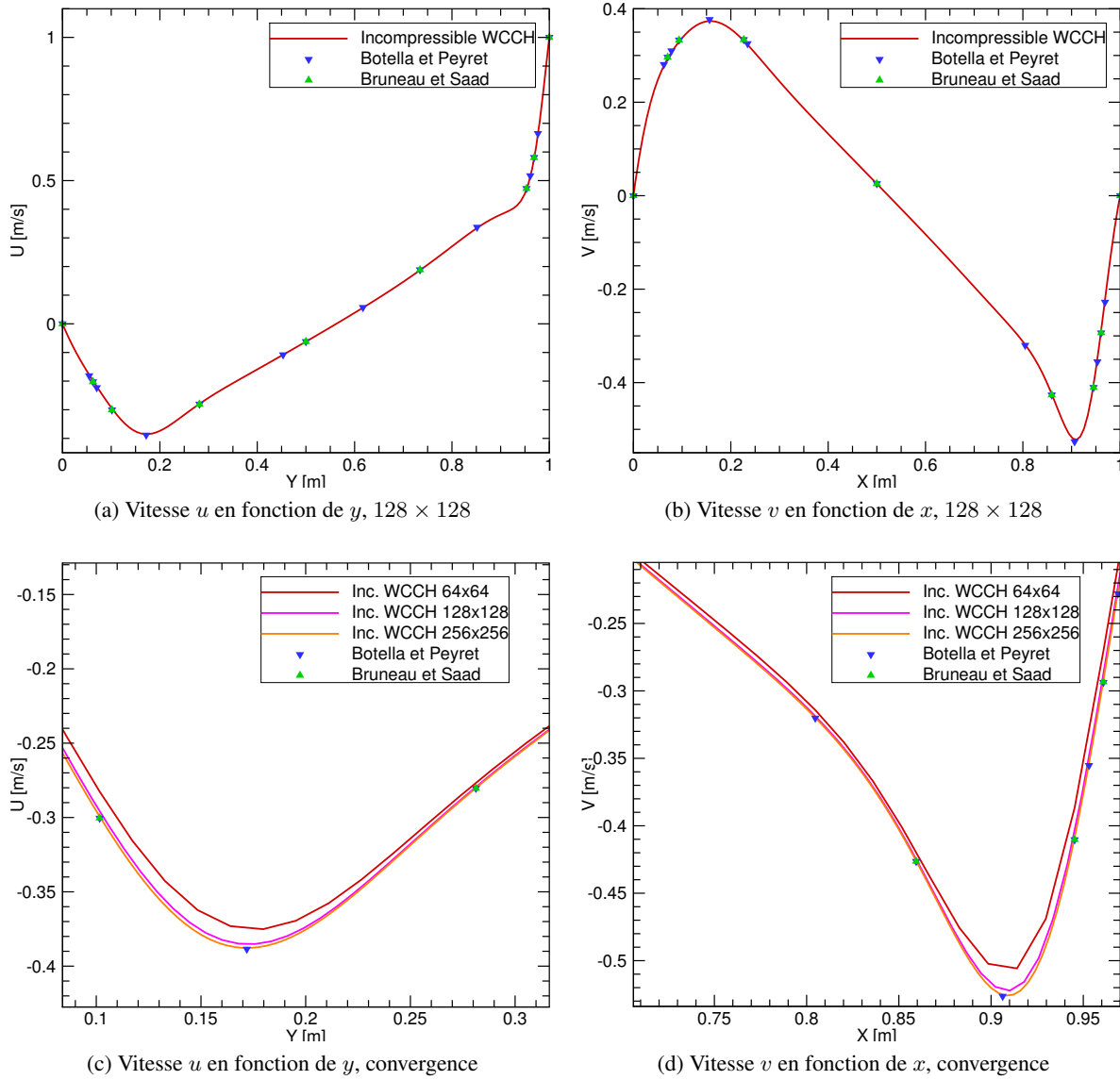


FIGURE 4.9 – Profils de vitesse horizontale et verticale à $Re = 1000$

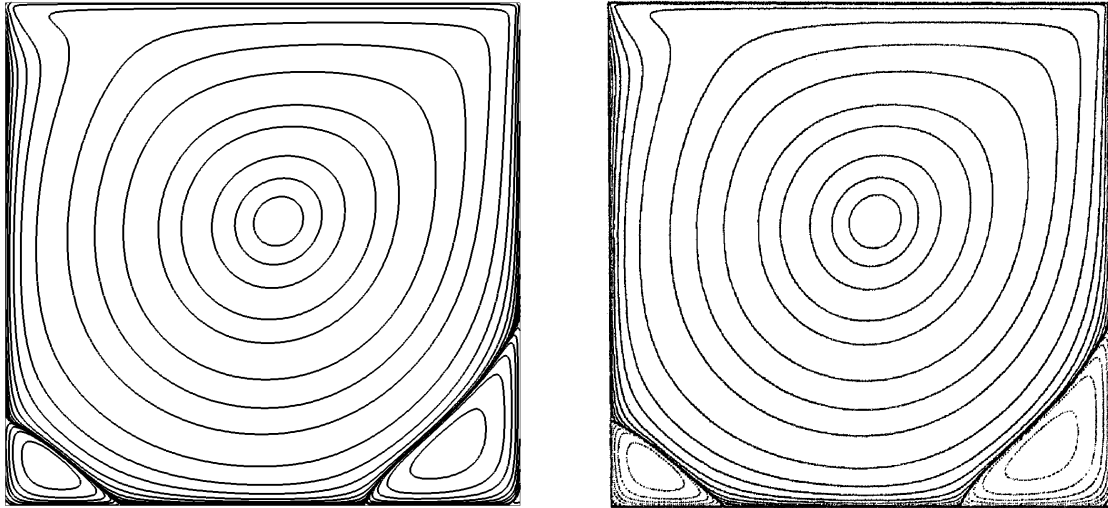
ψ est calculée numériquement en résolvant l'équation de Poisson :

$$\nabla^2 \psi = -\omega \quad (4.5)$$

avec ω la composante de la vorticit  dans la direction normale au plan de l' coulement. La r solution s'effectue comme pour l' quation de Poisson en pression, avec PETSc et HYPRE. Une condition de Dirichlet sur la fonction courant est appliqu e sur la paroi horizontale inf rieure $\psi = 0$.

La r solution de ψ est affich e sur la figure 4.10 et compar e aux r sultats de Bruneau et Saad issus d'un maillage 1024×1024 . Les isocontours de ψ d finissent les lignes de courant de l' coulement. Leur affichage permet de localiser les tourbillons. Plus pr cis ment, le centre des tourbillons correspond aux extrema de ψ .

Une convergence en maillage est r alis e sur la localisation des centres des recirculations. Les maillages vont de 4   256 cellules par direction. Les valeurs de ψ affich es dans le tableau 4.2 correspondent directement aux valeurs des cellules, aucune interpolation n'est r alis e pour estimer plus finement les coordonn es

(a) Solution incompressible WCCH, 128×128 (b) Solution de Bruneau et Saad, 1024×1024 FIGURE 4.10 – Lignes de courant de la cavité entraînée à $Re = 1000$

Maillage	x_{min}	y_{min}	ψ_{min}	x_{max}	y_{max}	ψ_{max}	$\epsilon_{x_{max}} (\%)$	$\epsilon_{y_{max}} (\%)$	$\epsilon_{\psi_{max}} (\%)$
4×4	0.625	0.625	-0.024	0.125	0.875	3.658e-03	85.53	682.65	111.45
8×8	0.562	0.562	-0.053	0.062	0.938	3.598e-03	92.82	739.00	107.98
16×16	0.594	0.594	-0.083	0.969	0.156	1.015e-03	12.15	39.53	41.33
32×32	0.547	0.578	-0.105	0.859	0.141	2.035e-03	0.58	26.12	17.63
64×64	0.539	0.570	-0.115	0.867	0.117	1.827e-03	0.35	4.65	5.61
128×128	0.527	0.566	-0.118	0.863	0.113	1.750e-03	0.12	1.07	1.16
256×256	0.529	0.564	-0.118	0.865	0.111	1.727e-03	0.12	0.72	0.17
Botella et Peyret	-	-	-	0.864	0.1118	1.730e-03			
Bruneau et Saad	0.531	0.565	-0.1189	0.8643	0.1123	1.7292e-03			

TABLE 4.2 – Localisation et intensité du tourbillon central principal (ψ_{min}) et du tourbillon secondaire (ψ_{max}) dans le coin inférieur droit.

et les valeurs des extrema. Ainsi pour le premier maillage de 4×4 , les coordonnées (x, y) sont précises à $0.25m$ près. Sur le maillage le plus fin, l'erreur n'est que de $4.10^{-3}m$ (inférieure à 1%), confirmant l'aptitude du code à capter finement des phénomènes complexes. Les écarts relatifs en pourcent par rapport à Botella et Peyret pour le tourbillon secondaire dans le coin inférieur gauche sont aussi présentés dans le tableau 4.2.

4.2 Validation sur grille raffinée localement

Dans cette partie, nous nous attachons à valider le raffinement statique local sur des grilles cartésiennes.

4.2.1 Comparaison guard-cell/stretching

Tous les cas tests présentés dans cette section sont extraits d'une session benchmark de la conférence FVCA 2017 (Finite Volume for Complex Applications) à laquelle nous avons participé pour éprouver la

formulation incompressible. Un certain nombre de cas tests devant se faire sur une grille raffinée localement, nous avons pu valider notre implémentation des guard-cells [Vittoz et al., 2017]. Nous reprenons ici quelques cas intéressants dans le but de comparer les deux méthodes guard-cells et stretching.

Introduction

L'objectif initial des benchmarks FVCA est de comparer les performances des solveurs incompressibles des participants sur des écoulements académiques. Les maillages étant imposés, il était demandé aux participants de fournir l'erreur et l'ordre de convergence des résultats numériques produits par leur solveur. Le domaine d'étude est un carré unité aux bords duquel des conditions de Dirichlet sur la vitesse sont spécifiées. Les maillages raffinés localement se composent de trois grilles de niveau de raffinement croissant dans le coin inférieur droit (Fig. 4.11).

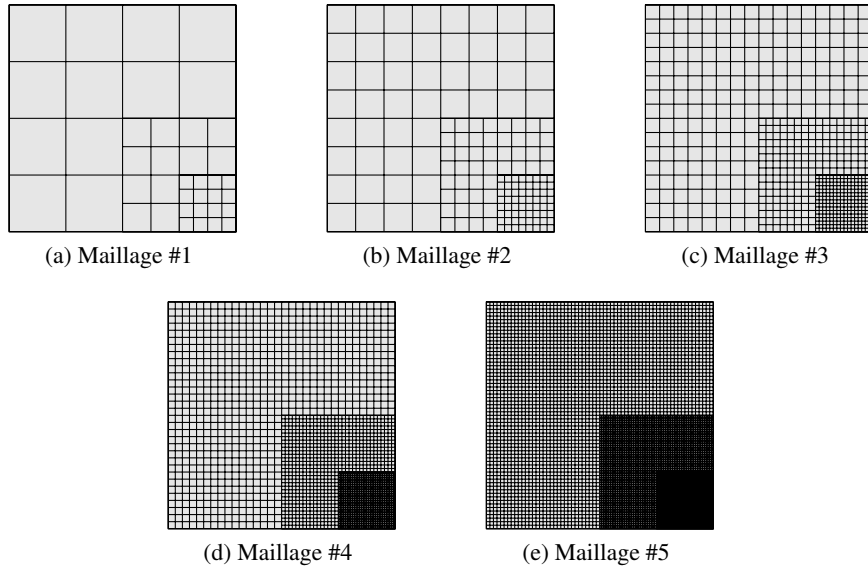


FIGURE 4.11 – Maillages raffinés localement (imposés pour la conférence FVCA)

Les maillages sont numérotés de 1 à 5. Le nombre de cellules par niveau de raffinement est doublé d'un maillage à l'autre et va de 4 à 64 cellules par direction.

Pour l'analyse de la précision, quatre erreurs de type L_2 sont calculées :

- Erreur sur le gradient de vitesse :

$$errgu = \left[\frac{\int_{\Omega} |\nabla(\mathbf{u} - \mathbf{u}_{ex})|^2}{\int_{\Omega} |\nabla \mathbf{u}_{ex}|^2} \right]^{1/2} \quad (4.6)$$

où Ω est le domaine étudié $\Omega = [0, 1] \times [0, 1]$. \mathbf{u}_{ex} est la solution analytique. Le gradient de vitesse est évalué par un schéma centré d'ordre 2. La norme matricielle utilisée est une norme de Frobenius :

$$\|A\|_2 = \sqrt{\sum_{i=1}^2 \sum_{j=1}^2 |a_{ij}|^2}.$$

- Erreur sur la vitesse :

$$erru = \left[\frac{\int_{\Omega} |\mathbf{u} - \mathbf{u}_{ex}|^2}{\int_{\Omega} |\mathbf{u}_{ex}|^2} \right]^{1/2} \quad (4.7)$$

- Erreur sur la pression :

$$errp = \left[\frac{\int_{\Omega} |p - p_{ex}|^2}{\int_{\Omega} |p_{ex}|^2} \right]^{1/2} \quad (4.8)$$

- Erreur sur la divergence de vitesse :

$$errdivu = \left[\int_{\Omega} |\nabla \cdot \mathbf{u}|^2 \right]^{1/2} \quad (4.9)$$

Pour un écoulement instationnaire, l'erreur est intégrée en temps sur toute la durée T de l'écoulement simulé. Par exemple pour le gradient de vitesse, elle est donnée par :

$$errgu = \left[\frac{\int_0^T \int_{\Omega} |\nabla(\mathbf{u} - \mathbf{u}_{ex})|^2}{\int_0^T \int_{\Omega} |\nabla \mathbf{u}_{ex}|^2} \right]^{1/2} \quad (4.10)$$

Toutes ces erreurs sont relatives et donc adimensionnelles à l'exception de celle portant sur $\nabla \cdot \mathbf{u}$ en stationnaire qui s'exprime en m^2/s .

Un ordre de convergence entre deux maillages i et $i + 1$ est ensuite calculé d'après :

$$ord = -2 \frac{\ln(err_i) - \ln(err_{i+1})}{\ln(n_i) - \ln(n_{i+1})} \quad (4.11)$$

n_i étant le nombre d'inconnues de la variable dont on calcule l'erreur pour le maillage i .

Écoulement 2D de Bercovier-Engelman

Ce premier cas test est un écoulement de Stokes stationnaire avec pour solution exacte :

$$\begin{aligned} \mathbf{u}_{ex} &= (u_1(x, y), -u_1(y, x))^T \text{ avec, } u_1(x, y) = -256x^2(x-1)^2y(y-1)(2y-1) \\ p_{ex} &= (x-1/2)(y-1/2) \end{aligned} \quad (4.12)$$

Un terme source non nul est ajouté :

$$\begin{cases} \mathbf{f} = (f_1(x, y) + (y-1/2), -f_1(y, x) + (x-1/2))^T \\ f_1(x, y) = 256 [x^2(x-1)^2(12y-6) + y(y-1)(2y-1)(12x^2-12x+2)] \end{cases} \quad (4.13)$$

La principale difficulté de ce cas test est la faible valeur du gradient du terme source en comparaison avec son rotationnel, compliquant ainsi la capture du champ de pression par une méthode numérique. Puisqu'il s'agit d'un écoulement de Stokes, la partie convective n'intervient pas et seuls la pression et le terme visqueux sont discrétisés. Les opérateurs de divergence et gradient et les flux visqueux sont discrétisés à l'ordre 2 partout sauf aux interfaces de raffinement où l'ordre est localement inférieur, mais nous pouvons toutefois attendre un ordre de convergence proche de 2 en norme L_2 .

La viscosité cinématique du fluide est fixée à $\nu = 1m^2/s$. L'écoulement est simulé sur un temps physique $T = 0.2s$ afin de s'assurer que le régime stationnaire est bien atteint.

On présente sur la figure 4.12 les champs de pression obtenus par les deux approches guard-cells et stretching. On remarque immédiatement une forme commune de découplage sur le maillage grossier #3. Ce problème est volontairement mis en évidence avec le logiciel de post-traitement avec plus de 17 niveaux de contour. Même si de tels résultats sont peu rassurants et font douter de la validité de l'implémentation des schémas numériques, une convergence est malgré tout observée sur des maillages plus fins. Pour le maillage #5, nous constatons que le phénomène de découplage est grandement atténué et se concentre principalement

aux interfaces de raffinement proches des frontières du domaine. Par ailleurs, le découplage de pression n'est apparent que pour les cellules fines, il est attribuable à la procédure d'interpolation des vitesses et gradients de pression des cellules fines de l'interface. L'erreur relative en pression est encore supérieure à 10% pour les deux approches sur le maillage le plus fin. A la vue des résultats, la discrétisation des opérateurs semble de mauvaise qualité aux interfaces de raffinement. Toutefois de telles conclusions sur la qualité de la résolution de l'équation de Poisson sont à nuancer et à mettre en regard avec le choix du maillage. En effet, celui-ci est peu avantageux pour ce cas test. Si le choix des grilles de raffinement était laissé libre, les interfaces de raffinement ne seraient pas placées là où le gradient de pression est le plus difficile à capturer. Enfin le découplage de pression n'a pas d'impact sur l'allure du champ de vitesse qui est conforme à ce qui est attendu (Fig. 4.12). Le champ est parfaitement régulier à la traversée des interfaces.

Nous synthétisons ensuite les calculs d'erreur et d'ordre de convergence des champs dans les tableaux 4.3 et 4.4. Une première remarque qui peut être formulée est la confirmation de la convergence des champs et notamment de la pression. Les erreurs sont sensiblement équivalentes entre les deux méthodes comme cela est attendu. En effet la résolution de l'équation de Poisson est strictement identique pour les deux. La seule différence est le calcul des flux visqueux dans la direction normale aux interfaces de raffinement. Les ordres de convergence sont également semblables et proches de 2 comme cela est attendu.

Maillage	errgu	ordgu	erru	ordu	errp	ordp	errdivu	orddivu
1	0.269	–	0.359	–	25.150	–	$6.603 \cdot 10^{-16}$	–
2	$8.103 \cdot 10^{-2}$	1.730	$8.836 \cdot 10^{-2}$	2.023	7.383	1.768	$9.364 \cdot 10^{-16}$	–
3	$2.284 \cdot 10^{-2}$	1.827	$2.176 \cdot 10^{-2}$	2.022	2.050	1.849	$1.635 \cdot 10^{-15}$	–
4	$6.116 \cdot 10^{-3}$	1.901	$5.413 \cdot 10^{-3}$	2.007	0.545	1.912	$3.039 \cdot 10^{-15}$	–
5	$1.602 \cdot 10^{-3}$	1.933	$1.354 \cdot 10^{-3}$	1.999	0.142	1.940	$6.064 \cdot 10^{-15}$	–

TABLE 4.3 – Convergence en maillage FVCA : 2D Bercovier–Engelman – guard-cells

Maillage	errgu	ordgu	erru	ordu	errp	ordp	errdivu	orddivu
1	0.254	–	0.286	–	19.940	–	$6.81 \cdot 10^{-16}$	–
2	$7.914 \cdot 10^{-2}$	1.681	$7.798 \cdot 10^{-2}$	1.874	6.025	1.727	$1.087 \cdot 10^{-15}$	–
3	$2.252 \cdot 10^{-2}$	1.813	$2.049 \cdot 10^{-2}$	1.928	1.703	1.823	$1.694 \cdot 10^{-15}$	–
4	$6.047 \cdot 10^{-3}$	1.897	$5.269 \cdot 10^{-3}$	1.959	0.459	1.892	$3.18 \cdot 10^{-15}$	–
5	$1.583 \cdot 10^{-3}$	1.934	$1.339 \cdot 10^{-3}$	1.976	0.121	1.925	$6.306 \cdot 10^{-15}$	–

TABLE 4.4 – Convergence en maillage FVCA : 2D Bercovier–Engelman – stretching

Nous constatons aussi que l'erreur sur la divergence de vitesse est très proche du zéro machine, ce qui valide la conservation de la masse. Elle n'est pas toujours constante et dépend du critère d'arrêt des itérations de résolution de l'équation de Poisson. Pour ce cas test il est fixé par une norme L_2 des résidus valant 10^{-8} . Dans les cas tests suivants, nous avons parfois été amenés à abaisser ce seuil pour faire décroître l'erreur sur la divergence de vitesse jusqu'à une valeur insignifiante.

Finalement, ce cas test permet de confirmer que la résolution de l'équation de Poisson et le calcul des gradients de vitesse pour le flux visqueux sont très proches de l'ordre 2, même si localement il est sans doute plus faible.

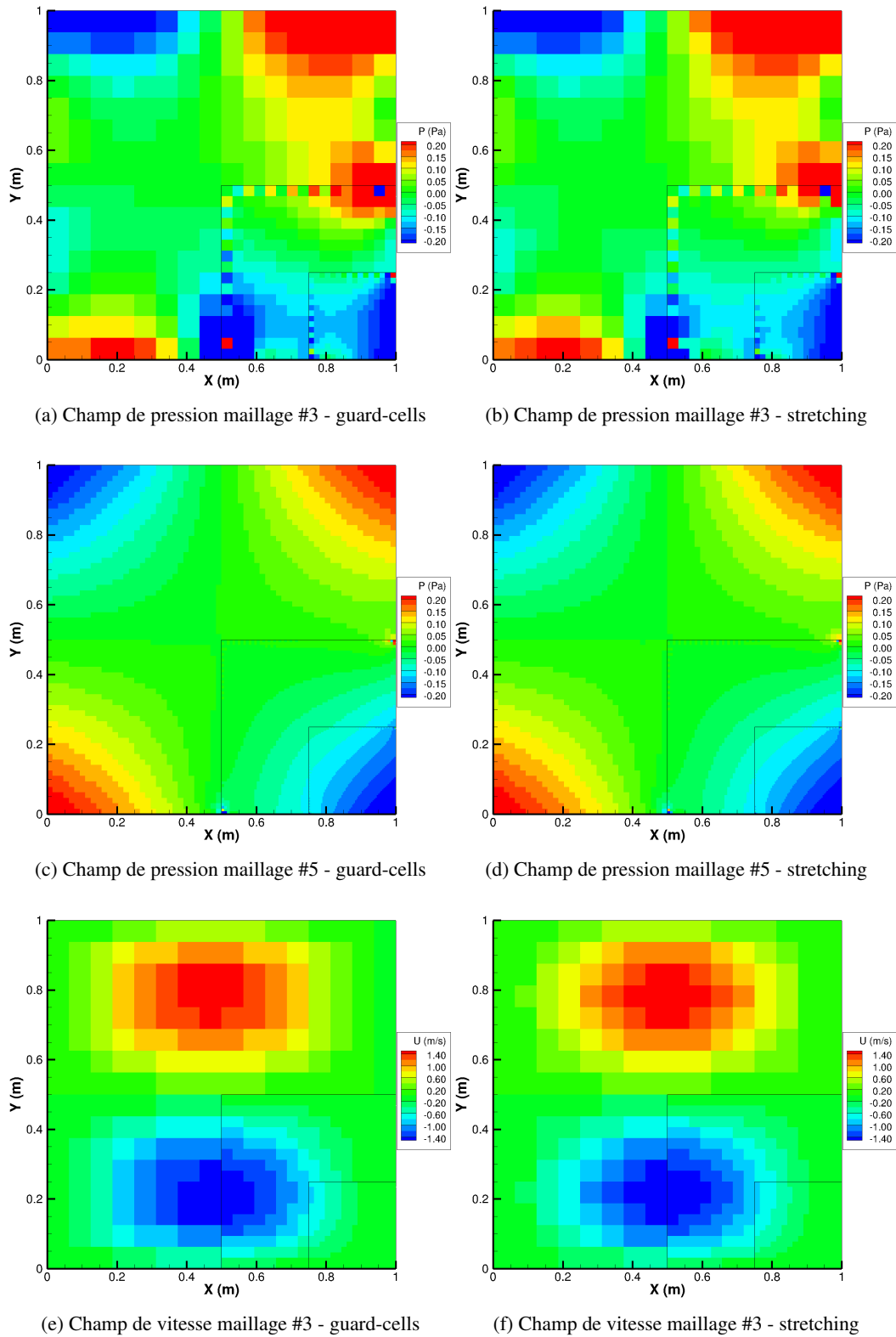


FIGURE 4.12 – Pression et vitesse horizontale pour le cas test de Bercovier-Engelman

Écoulement 2D stationnaire d'un vortex

Le cas test précédent ne faisait pas intervenir le terme convectif. Nous étudions à présent un vortex dont la vitesse s'exprime simplement par :

$$\begin{aligned} \mathbf{u}_{ex} &= (y, -x)^T \\ p_{ex} &= \frac{1}{2}(x^2 + y^2) - \frac{1}{3} \end{aligned} \tag{4.14}$$

La pression est donc quadratique et des conditions non homogènes de Dirichlet sont appliquées sur la vitesse aux frontières du domaine.

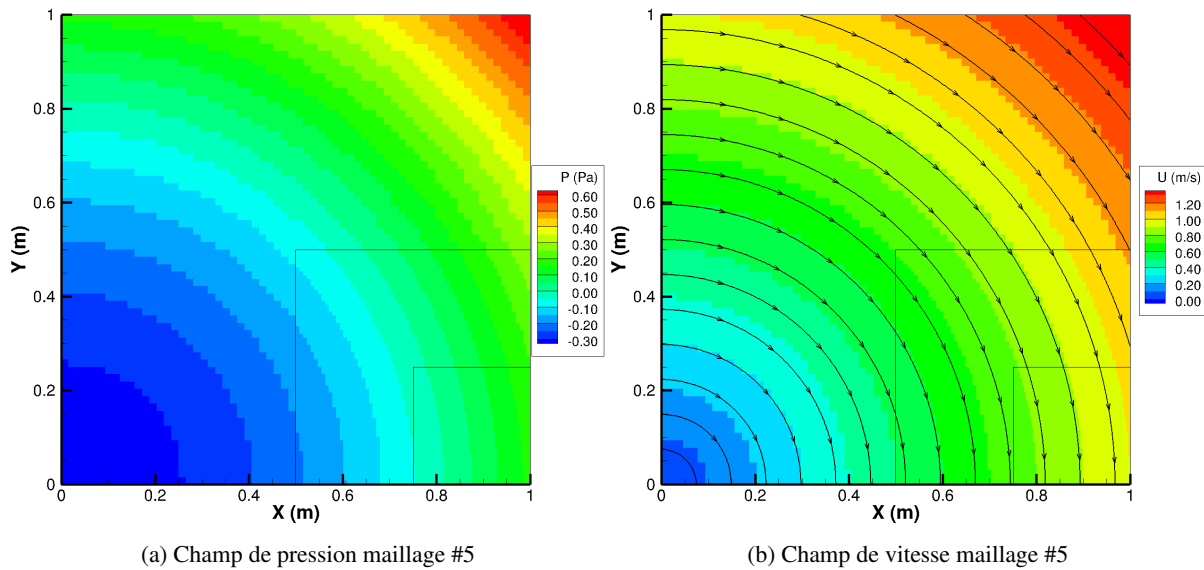


FIGURE 4.13 – Pression et norme du vecteur vitesse pour l'écoulement vortex

Puisque les champs de vitesse évoluent linéairement et que l'ensemble de nos schémas numériques ont un ordre de convergence strictement supérieur à 1, une erreur de discrétisation nulle pour les inconnues de vitesses est logiquement attendue.

La viscosité physique est fixée à $\nu = 10^{-1} m^2/s$ et le temps de simulation à $T = 3s$. Les tableaux 4.5 et 4.6 regroupent les erreurs et les ordres de convergence.

Maillage	errgu	ordgu	erru	ordu	errp	ordp	errdivu	orddivu
1	0.000	–	0.000	–	$2.225 \cdot 10^{-2}$	–	0.000	–
2	$4.907 \cdot 10^{-18}$	–	$2.304 \cdot 10^{-19}$	–	$5.434 \cdot 10^{-3}$	2.034	$6.939 \cdot 10^{-18}$	–
3	$2.273 \cdot 10^{-17}$	–	$8.635 \cdot 10^{-19}$	–	$1.35 \cdot 10^{-3}$	2.009	$1.604 \cdot 10^{-17}$	–
4	$1.234 \cdot 10^{-17}$	–	$7.131 \cdot 10^{-19}$	–	$3.371 \cdot 10^{-4}$	2.002	$1.138 \cdot 10^{-17}$	–
5	$7.142 \cdot 10^{-17}$	–	$1.419 \cdot 10^{-18}$	–	$8.425 \cdot 10^{-5}$	2.000	$7.152 \cdot 10^{-18}$	–

TABLE 4.5 – Convergence en maillage FVCA : Vortex 2D - guard-cells

Comme attendu, les erreurs sur la vitesse, son gradient et sa divergence sont nulles. La pression converge quant à elle à l'ordre 2.

Ce cas test a de rassurant qu'il confirme que les schémas numériques de discrétisation spatiale sont à un ordre strictement supérieur à 1, y compris pour les maillages avec uniquement 2 cellules par bloc, comme c'est le cas pour le maillage #1.

Maillage	errgu	ordgu	erru	ordu	errp	ordp	errdivu	orddivu
1	$3.621 \cdot 10^{-16}$	–	$7.758 \cdot 10^{-17}$	–	$2.046 \cdot 10^{-2}$	–	$1.837 \cdot 10^{-16}$	–
2	$7.023 \cdot 10^{-16}$	–	$7.854 \cdot 10^{-17}$	–	$4.994 \cdot 10^{-3}$	2.035	$5.317 \cdot 10^{-16}$	–
3	$1.013 \cdot 10^{-15}$	–	$3.748 \cdot 10^{-17}$	–	$1.241 \cdot 10^{-3}$	2.009	$8.516 \cdot 10^{-16}$	–
4	$1.547 \cdot 10^{-15}$	–	$2.266 \cdot 10^{-17}$	–	$3.099 \cdot 10^{-4}$	2.002	$9.087 \cdot 10^{-16}$	–
5	$3.223 \cdot 10^{-15}$	–	$2.47 \cdot 10^{-17}$	–	$7.744 \cdot 10^{-5}$	2.001	$2.49 \cdot 10^{-15}$	–

TABLE 4.6 – Convergence en maillage FVCA : Vortex 2D - stretching

Écoulement 2D instationnaire

Nous nous intéressons maintenant à un cas test d'une complexité supérieure par rapport aux écoulements précédents. Il s'agit d'un écoulement instationnaire de tourbillons dont la décroissance de l'amplitude est pilotée par la viscosité :

$$\begin{aligned}
 \psi &= e^{-5\nu\pi^2 t} \cos(\pi x) \cos(2\pi y) \\
 \mathbf{u}_{ex} &= (\partial_y \psi, -\partial_x \psi)^T \\
 p_{ex} &= -\frac{1}{4} e^{-10\nu\pi^2 t} \pi^2 (4 \cos(2\pi x) + \cos(4\pi y))
 \end{aligned} \tag{4.15}$$

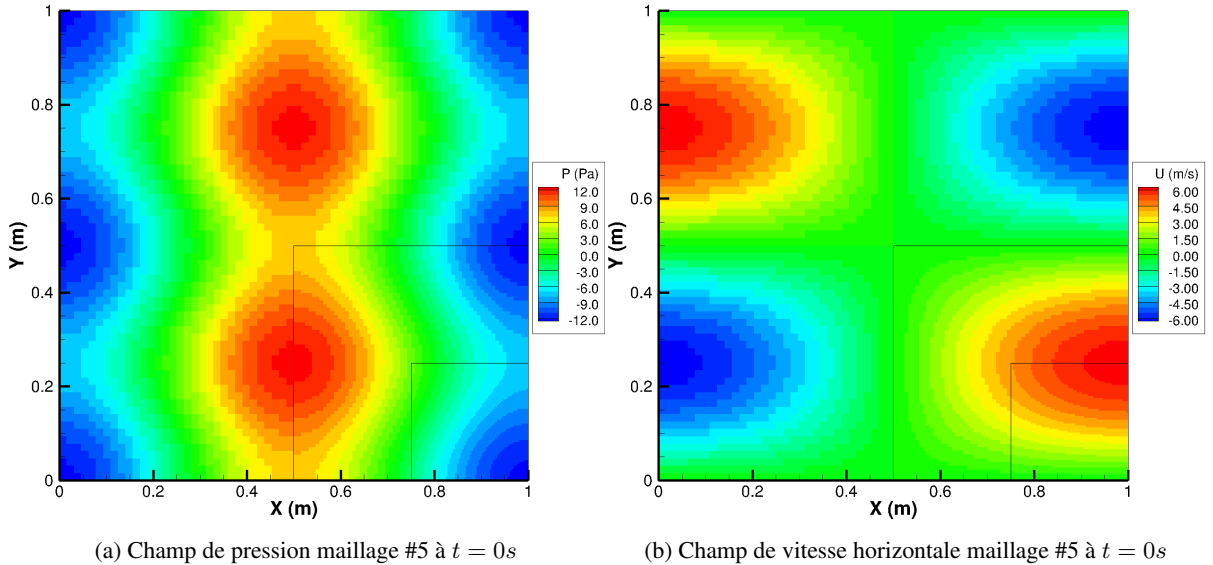


FIGURE 4.14 – Tourbillons instationnaires. Pression et vitesse initiales pour l'écoulement instationnaire

Des conditions non homogènes et instationnaires de Dirichlet sont appliquées sur la vitesse. Nous nous attendons à une chute de l'ordre de convergence de la pression à 1. Pour la vitesse, il est difficile de nous avancer sur un ordre de convergence théorique.

Une première simulation est conduite pendant un temps physique de $T = 1s$ avec une viscosité cinématique $\nu = 0.1m^2/s$. Les résultats sont présentés dans les tableaux 4.7 et 4.8. Pour les deux approches, les erreurs et les ordres de convergence sont semblables.

On prend ensuite une viscosité de $0.01m^2/s$ pour un temps simulé de $T = 10s$ ce qui fait que l'amplitude des tourbillons à la fin de la simulation est la même que pour la précédente viscosité. Les résultats obtenus (Tab. 4.9 et 4.10) sont bien différents. D'une part l'ordre de convergence est bien moins net pour les deux approches, d'autre part des écarts significatifs sont observables d'une approche à l'autre. A partir

CHAPITRE 4. VALIDATIONS NUMÉRIQUES ET COMPARAISONS
INCOMPRESSIBLE/FAIBLEMENT-COMPRESSIBLE

Maillage	errgu	ordgu	erru	ordu	errp	ordp	errdivu	orddivu
1	0.341	–	$8.53 \cdot 10^{-2}$	–	0.264	–	$6.216 \cdot 10^{-16}$	–
2	$9.724 \cdot 10^{-2}$	1.810	$1.955 \cdot 10^{-2}$	2.125	0.124	1.084	$1.057 \cdot 10^{-15}$	–
3	$2.55 \cdot 10^{-2}$	1.931	$4.834 \cdot 10^{-3}$	2.016	$6.11 \cdot 10^{-2}$	1.025	$2.107 \cdot 10^{-15}$	–
4	$6.528 \cdot 10^{-3}$	1.966	$1.215 \cdot 10^{-3}$	1.992	$3.037 \cdot 10^{-2}$	1.009	$4.083 \cdot 10^{-15}$	–
5	$1.69 \cdot 10^{-3}$	1.950	$3.238 \cdot 10^{-4}$	1.908	$1.516 \cdot 10^{-2}$	1.002	$8.107 \cdot 10^{-15}$	–

TABLE 4.7 – Convergence en maillage FVCA : Tourbillons instationnaires - $\nu = 10^{-1}$ - guard-cells

Maillage	errgu	ordgu	erru	ordu	errp	ordp	errdivu	orddivu
1	0.345	–	$8.654 \cdot 10^{-2}$	–	0.305	–	$6.296 \cdot 10^{-16}$	–
2	$9.793 \cdot 10^{-2}$	1.817	$2.551 \cdot 10^{-2}$	1.762	0.130	1.231	$1.107 \cdot 10^{-15}$	–
3	$2.571 \cdot 10^{-2}$	1.929	$6.644 \cdot 10^{-3}$	1.941	$6.221 \cdot 10^{-2}$	1.063	$2.219 \cdot 10^{-15}$	–
4	$6.55 \cdot 10^{-3}$	1.973	$1.647 \cdot 10^{-3}$	2.012	$3.052 \cdot 10^{-2}$	1.027	$4.133 \cdot 10^{-15}$	–
5	$1.672 \cdot 10^{-3}$	1.970	$4.157 \cdot 10^{-4}$	1.986	$1.518 \cdot 10^{-2}$	1.008	$8.068 \cdot 10^{-15}$	–

TABLE 4.8 – Convergence en maillage FVCA : Tourbillons instationnaires - $\nu = 10^{-1}$ - stretching

des maillages #3, la méthode guard-cells donne des erreurs 2 à 3 fois moins importantes que l'approche de type stretching. Cet exemple tend donc à montrer que l'approche guard-cells donne de meilleurs résultats à Reynolds élevé lorsque la partie convective devient prépondérante. L'ordre de convergence de la vitesse est également plus important (> 2.5).

Maillage	errgu	ordgu	erru	ordu	errp	ordp	errdivu	orddivu
1	0.943	–	1.110	–	1.009	–	$3.186 \cdot 10^{-15}$	–
2	0.195	2.275	0.173	2.686	0.125	3.014	$3.759 \cdot 10^{-15}$	–
3	$3.149 \cdot 10^{-2}$	2.628	$1.203 \cdot 10^{-2}$	3.842	$1.646 \cdot 10^{-2}$	2.924	$7.473 \cdot 10^{-15}$	–
4	$7.995 \cdot 10^{-3}$	1.978	$2.428 \cdot 10^{-3}$	2.309	$4.469 \cdot 10^{-3}$	1.881	$1.446 \cdot 10^{-14}$	–
5	$1.995 \cdot 10^{-3}$	2.003	$5.197 \cdot 10^{-4}$	2.224	$1.692 \cdot 10^{-3}$	1.401	$2.654 \cdot 10^{-14}$	–

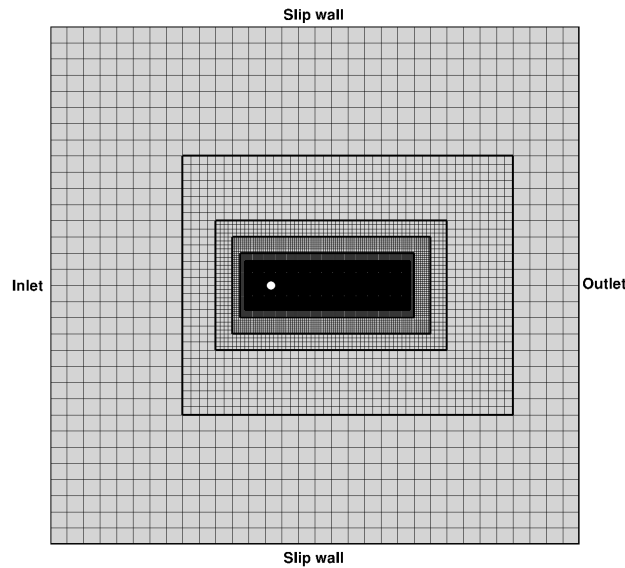
TABLE 4.9 – Convergence en maillage FVCA : Tourbillons instationnaires - $\nu = 10^{-2}$ - guard-cells

Maillage	errgu	ordgu	erru	ordu	errp	ordp	errdivu	orddivu
1	0.792	–	1.163	–	1.210	–	$3.841 \cdot 10^{-15}$	–
2	0.458	0.791	0.471	1.305	0.434	1.479	$6.515 \cdot 10^{-15}$	–
3	$6.174 \cdot 10^{-2}$	2.889	$3.786 \cdot 10^{-2}$	3.636	$3.868 \cdot 10^{-2}$	3.488	$1.254 \cdot 10^{-14}$	–
4	$1.497 \cdot 10^{-2}$	2.044	$1.005 \cdot 10^{-2}$	1.913	$1.126 \cdot 10^{-2}$	1.780	$2.208 \cdot 10^{-14}$	–
5	$3.664 \cdot 10^{-3}$	2.031	$2.511 \cdot 10^{-3}$	2.001	$3.137 \cdot 10^{-3}$	1.844	$3.118 \cdot 10^{-14}$	–

TABLE 4.10 – Convergence en maillage FVCA : Tourbillons instationnaires - $\nu = 10^{-2}$ - stretching

4.2.2 Écoulement laminaire autour d'un cylindre

Nous étudions à présent un écoulement plus complexe autour d'un cylindre pour valider l'adaptation de la méthode de frontière immergée à la formulation incompressible présentée dans la section 2.4. La simulation est réalisée avec un raffinement statique de maillage afin de concentrer les cellules au voisinage du corps et dans le sillage. En entrée, à gauche du domaine, une condition entrante fixe la vitesse horizontale à $u_\infty = 1m/s$. A droite, en sortie de domaine, une condition sortante est imposée. Les frontières supérieure et inférieure du domaine ont une condition de glissement. Le domaine de calcul est de forme carrée et de dimension $60D$ où D est le diamètre du cylindre, ici fixé à $1m$ (Fig 4.15). Les frontières du domaine sont placées suffisamment loin pour éviter tout effet de bord.

FIGURE 4.15 – Conditions de calcul et maillage ($\frac{D}{\Delta x} = 17$)

Le nombre de Reynolds de l'écoulement est défini par :

$$Re = \frac{\rho u_\infty D}{\mu} \quad (4.16)$$

Jusqu'à un nombre de Reynolds d'environ 50, l'écoulement autour d'un cylindre est stable et la zone de recirculation en aval de l'écoulement est parfaitement symétrique. Au-delà le sillage devient instationnaire et des détachements alternés de tourbillons se produisent (allée de Von Kármán).

Nous effectuons des simulations à deux nombres de Reynolds différents : 40 et 100. Dans chaque cas, trois maillages sont testés : un premier grossier, avec 17 cellules dans le diamètre puis $\frac{D}{\Delta x} = 34$ et $\frac{D}{\Delta x} = 68$

dans les zones les plus raffinées. Le nombre de blocs est identique pour les trois configurations mais le nombre de cellules par bloc est adapté : 4, 8 et 16 cellules.

Ce cas test est un souvent employé pour l'évaluation des solveurs visqueux, et de nombreuses données expérimentales ou numériques sont disponibles dans la littérature. Un moyen d'évaluer la précision du solveur est de calculer les coefficients de traînée et de portance :

$$C_d = \frac{F_x}{\frac{1}{2}\rho u_\infty^2 D} \text{ et } C_l = \frac{F_y}{\frac{1}{2}\rho u_\infty^2 D} \quad (4.17)$$

Écoulement à Reynolds 40

Pour ce nombre de Reynolds, l'écoulement attendu est stationnaire et parfaitement symétrique. La figure 4.16 donne les champs de pression (adimensionnalisée) et de vitesse horizontale pour le maillage $D/\Delta x = 68$.

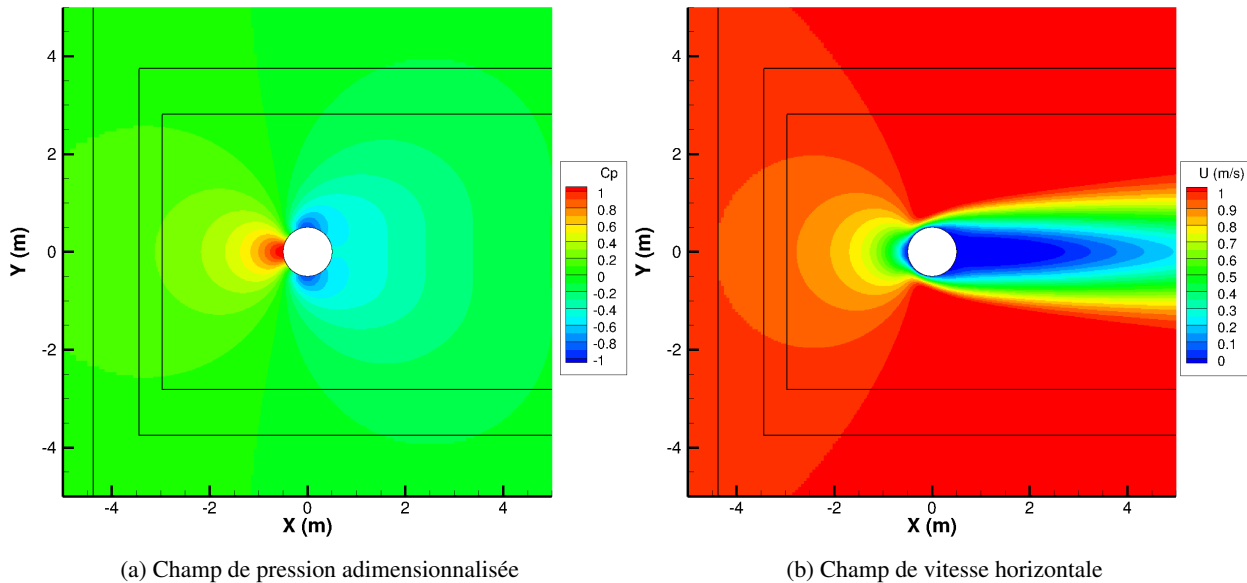


FIGURE 4.16 – Pression et vitesse autour du cylindre à $Re = 40$ pour $\frac{D}{\Delta x} = 68$

Pour regarder plus en détail le champ de pression, nous traçons le coefficient de pression $C_p = p/\frac{1}{2}\rho u_\infty^2$ sur le cylindre en fonction de l'angle (Fig. 4.17). L'angle 0 correspond au point d'arrêt de l'écoulement où la pression est maximale. Nous comparons nos résultats à plusieurs résultats numériques [Berthelsen et Faltinsen, 2008, Tseng et Ferziger, 2003, Dennis et Chang, 1970] ou expérimentaux de référence [Grove et al., 1964]. On remarque que la pression que l'on obtient est assez proche de celles des références jusqu'à un angle d'environ 60 degrés. Au delà de cet angle, l'écart se creuse jusqu'à être maximum pour le minimum de pression. A 180 degrés, le champ de pression de notre solution correspond à nouveau aux références. L'écart observé peut être attribué au logiciel de post-traitement (Tecplot) qui extrait le profil de pression le long la frontière du corps avec des interpolations dont la qualité n'est pas maîtrisée. Une autre explication est à mettre en lien avec le problème de couche limite artificielle produite par le schéma de projection. En effet l'écart est maximum dans la région où l'écoulement est le plus accéléré. Les gradients de vitesse y sont maximaux et le forçage par le terme source est donc plus "brutal".

La valeur de C_d sur le maillage le plus fin est de 1.523 et se situe dans l'intervalle des valeurs rencontrées dans la littérature entre 1.49 et 1.62. Par ailleurs, on relève que la convergence est très rapide si on prend

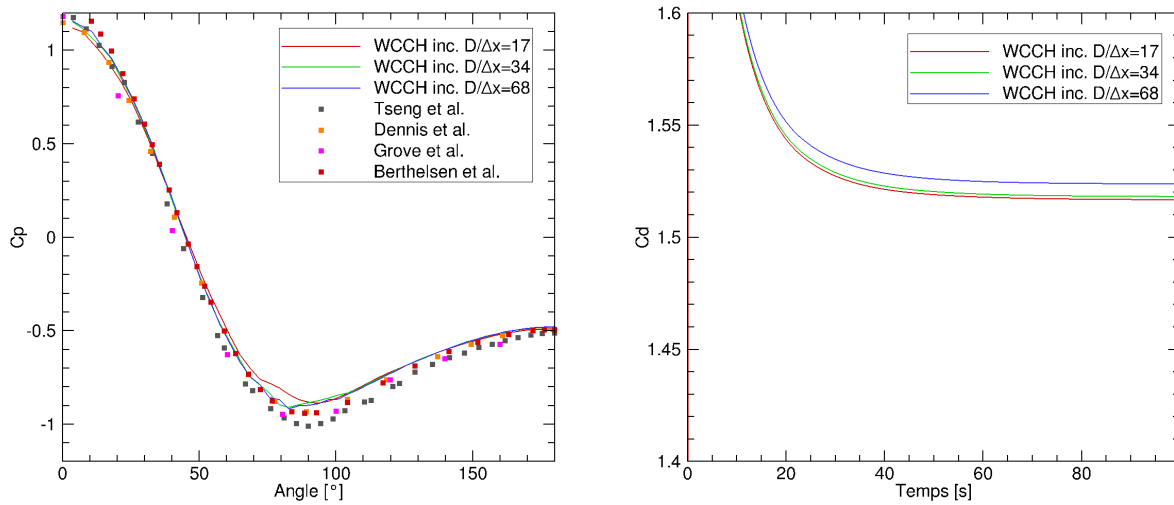


FIGURE 4.17 – Coefficient de pression sur le cylindre (gauche) et coefficient de traînée (droite)

comme référence le coefficient de traînée du maillage $\frac{D}{\Delta x} = 68$. Les deux autres maillages donnent en effet des écarts relatifs inférieur à 0.5% par rapport à ce maillage.

D'autres critères d'évaluation sont disponibles dans la littérature et concernent la zone de recirculation en aval du cylindre et l'angle de décollement θ (Fig. 4.18).

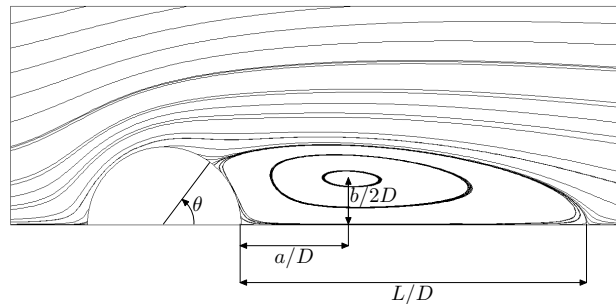


FIGURE 4.18 – Critères géométriques de la zone de recirculation

Les critères sont regroupés dans le tableau 4.11.

	C_d	$\frac{L}{D}$	$\frac{a}{D}$	$\frac{b}{D}$	θ
Coutanceau and Bouard (Exp) [Coutanceau et Bouard, 1977]	-	2.13	0.76	0.59	53.5
Tritton (Exp) [Tritton, 1959]	1.48	-	-	-	-
Linnick and Fasel [Linnick et Fasel, 2005]	1.54	2.28	0.72	0.60	53.6
WCCH inc. $\frac{D}{\Delta x} = 68$	1.523	2.22	0.70	0.61	52.9

 TABLE 4.11 – Comparaison des caractéristiques de l'écoulement autour du cylindre à $Re = 40$ par rapport à des données expérimentales et numériques

La zone de recirculation obtenue sur le maillage le plus fin est très proche des références.

Enfin nous donnons quelques éléments sur les temps de calcul associés à ces simulations. Pour le maillage le plus fin, le nombre total de cellules est d'environ 600000. Pour simuler 100 secondes d'écou-

lement physique, le calcul s'est exécuté en 22 heures sur 16 cœurs d'un nœud contenant deux processeurs Intel Xeon E5-2680 v3 Twelve-Core Haswell 2.5GHz. A noter que dans ces conditions de calcul, le pas de temps est fixé par la condition de stabilité sur le nombre CFL visqueux.

Écoulement à Reynolds 100

Pour un nombre de Reynolds égal à 100, l'écoulement est donc instationnaire et un détachement périodique alterné de tourbillons s'installe (allée de Von Kármán) (Fig. 4.19). On remarque que les champs (vitesse, pression et vorticité) sont parfaitement continus à la traversée des interfaces de raffinement.

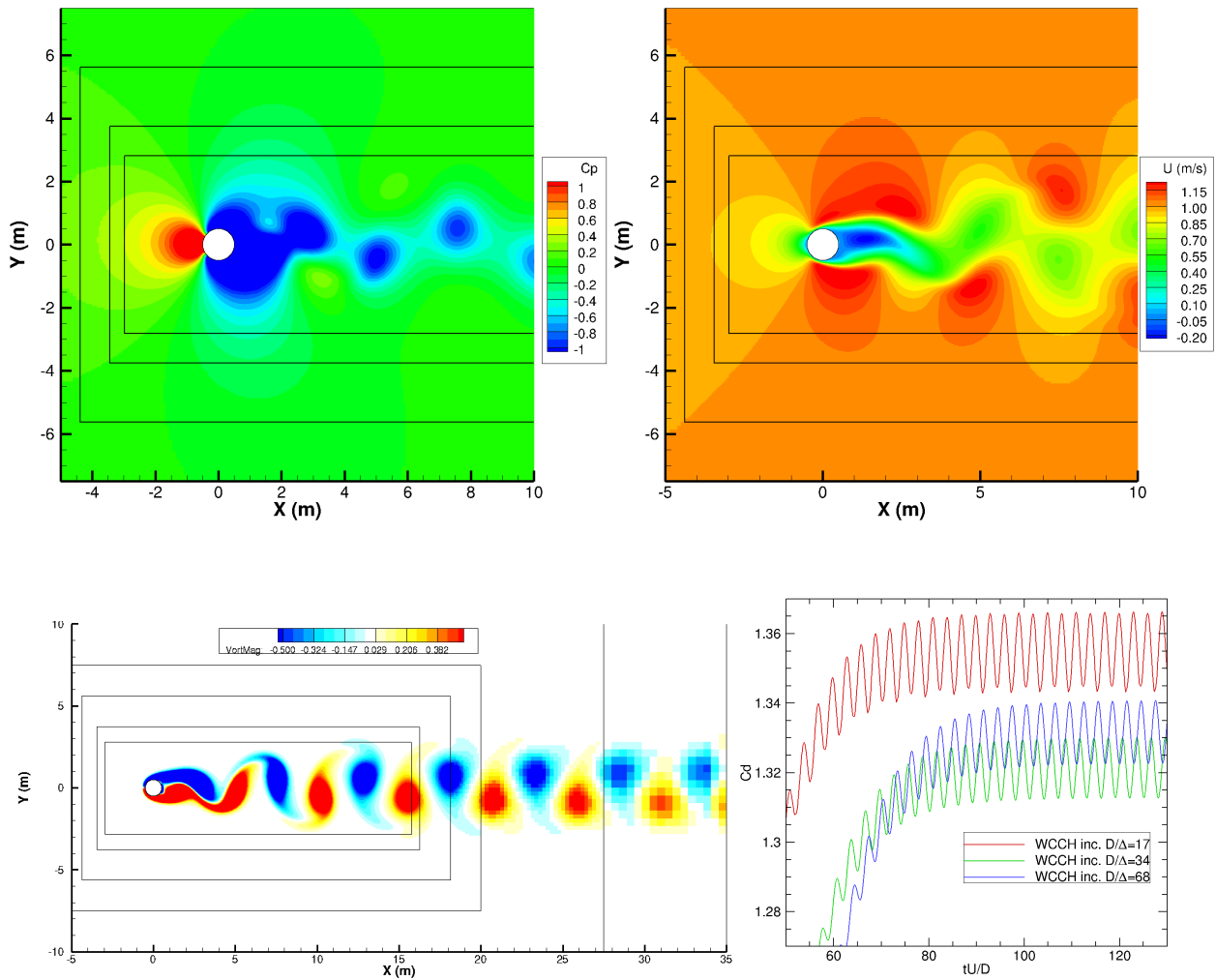


FIGURE 4.19 – Écoulement autour du cylindre à $Re = 100$ en formulation incompressible pour le maillage $\frac{D}{\Delta x} = 68$ à $\frac{tu_{\infty}}{D} = 125$.

Le tableau 4.12 présente une comparaison des coefficients de traînée, de portance et le nombre de Strouhal qui mesure la fréquence f des détachements tourbillonnaires :

$$St = \frac{fD}{u_{\infty}} \quad (4.18)$$

Ces résultats sont en accord avec la littérature. La fréquence des tourbillons est correctement prédite dès le premier maillage grossier. Pour finir, le calcul du maillage le plus fin a duré 17 heures pour des conditions

	C_d	C_l	Strouhal
Braza et al. [Braza et al., 1986]	1.364 ± 0.015	± 0.25	-
Calhoun [Calhoun, 2002]	1.33 ± 0.014	± 0.298	-
Ng et al. [Ng et al., 2009]	1.368 ± 0.01	± 0.36	-
Expérience [Wu et al., 2004]	-	-	0.164
WCCH inc. $\frac{D}{\Delta x} = 17$	1.355 ± 0.011	± 0.347	0.166
WCCH inc. $\frac{D}{\Delta x} = 34$	1.320 ± 0.009	± 0.316	0.167
WCCH inc. $\frac{D}{\Delta x} = 68$	1.332 ± 0.009	± 0.320	0.166

TABLE 4.12 – Coefficient de traînée et de portance et nombre de Strouhal pour l’écoulement autour d’un cylindre à $Re = 100$

d’exécution identiques à celle du calcul à $Re = 40$. Le temps d’exécution est plus court alors que la durée de simulation est de 200 secondes à cause des différences de valeur de la viscosité cinématique.

En conclusion, ces écoulements laminaires à différents régimes autour d’un cylindre ont permis de valider l’implémentation d’une méthode de frontière immergée en formulation incompressible. Ces calculs auraient été difficilement réalisables sur grille cartésienne uniforme (plus de 16 millions de cellules nécessaires pour une grille uniforme équivalente au maillage le plus fin testé). Le raffinement statique de maillage permet de concentrer les cellules à proximité du corps et dans le sillage pour bien capter la physique de l’écoulement.

4.3 Comparaison des formulations incompressible et faiblement-compressible

Dans cette section, nous cherchons maintenant à confronter les deux formulations, incompressible et faiblement-compressible. Dans la littérature des comparaisons entre les deux formulations ont déjà été conduites mais principalement avec la méthode SPH [Lee et al., 2008, Shadloo et al., 2012]. Nous avons identifié trois critères principaux de comparaison : la diffusion numérique, l’aptitude à simuler un écoulement laminaire instationnaire autour d’un corps et l’efficacité algorithmique/calcul haute performance. Les calculs sont effectués dans des conditions identiques pour les deux formulations. Ils sont exécutés sur des nœuds de calcul contenant deux processeurs de type Intel Xeon E5-2680 v3 Twelve-Core Haswell 2.5GHz.

4.3.1 Diffusion numérique et précision

Tourbillons de Taylor-Green

Il s’agit d’un écoulement incompressible de décroissance de tourbillons, abondamment utilisé dans la littérature pour évaluer la diffusion numérique des schémas. La solution exacte de cet écoulement vaut :

$$\begin{cases} u(x, y, t) = U_0 e^{-\frac{8\pi^2 \nu t}{L^2}} \sin(2\pi \frac{x}{L}) \cos(2\pi \frac{y}{L}) \\ v(x, y, t) = -U_0 e^{-\frac{8\pi^2 \nu t}{L^2}} \cos(2\pi \frac{x}{L}) \sin(2\pi \frac{y}{L}) \\ p(x, y, t) = \frac{\rho U_0^2}{4} e^{-\frac{16\pi^2 \nu t}{L^2}} [\cos(4\pi \frac{x}{L}) + \cos(4\pi \frac{y}{L})] \end{cases} \quad (4.19)$$

avec ν la viscosité cinématique. $U_0 = 1m/s$ est la vitesse de référence de l’écoulement, $\rho = 1000kg/m^3$ est la densité nominale du fluide et $L = 1m$ est la taille du domaine périodique bidimensionnel. En l’absence de viscosité, l’écoulement est stationnaire. Néanmoins la diffusion numérique intrinsèque des schémas est

responsable de la décroissance de l'amplitude des tourbillons au cours du temps. La dépendance au nombre de Mach de la diffusion numérique de l'approche faiblement-compressible du code WCCH a déjà été mise en évidence sur ce cas test par C. Leroy [Leroy, 2012]. Dans notre cas nous allons plus loin en la comparant avec la diffusion d'une approche purement incompressible.

Des simulations sont conduites sur trois grilles cartésiennes uniformes de résolution spatiale croissante $N_x \times N_y = 32 \times 32, 64 \times 64, 128 \times 128$. Pour la formulation faiblement-compressible, la vitesse nominale du son est fixée à $c_0 = 10m/s$.

La figure 4.20 présente les champs de pression obtenu à $\frac{tU_0}{L} = 3$ pour les deux formulations avec un maillage de 32 cellules par direction. Les champs sont en apparence très semblables mais si on regarde la décroissance d'énergie cinétique au cours du temps, on constate que l'approche faiblement-compressible est beaucoup plus diffusive que celle incompressible.

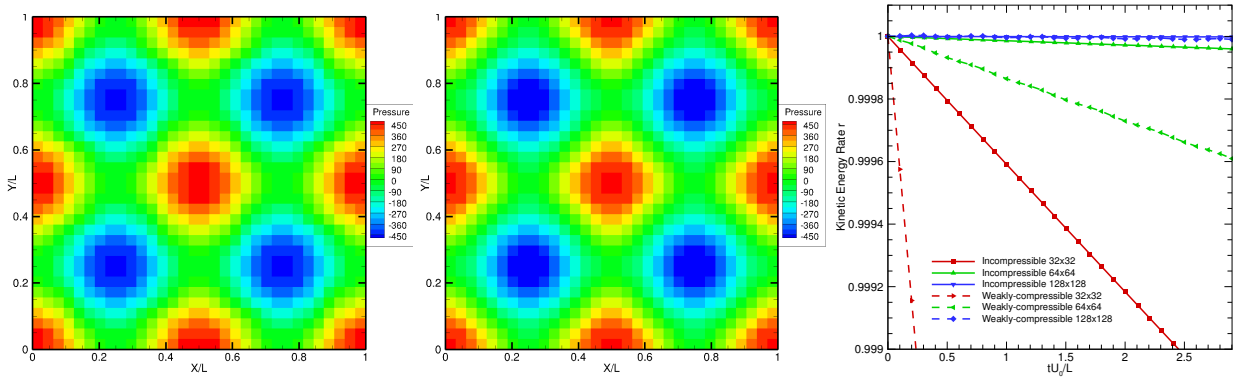


FIGURE 4.20 – Tourbillons de Taylor-Green. Comparaison de la pression à $\frac{tU_0}{L} = 3$ pour l'approche faiblement-compressible (gauche) et incompressible (milieu). Taux de dissipation de l'énergie cinétique en fonction du temps (droite)

Par ailleurs, il est aisé d'estimer une viscosité numérique par analogie avec la viscosité analytique de la solution (4.19). La décroissance exponentielle de l'énergie cinétique $r(t)$ au cours du temps est égale à :

$$r(t) = \frac{k(t)}{k(t=0)} = \frac{\frac{1}{2} \iint_D \rho(x, y, t) (u^2(x, y, t) + v^2(x, y, t)) dx dy}{\frac{1}{2} \iint_D \rho(x, y, 0) (u^2(x, y, 0) + v^2(x, y, 0)) dx dy} = e^{-\frac{16\pi^2 \nu t}{L^2}} \quad (4.20)$$

En appliquant une régression linéaire au sens des moindres carrés, des viscosités cinématiques numériques peuvent être calculées pour chaque discrétisation (Tab. 4.13). Nous constatons que les valeurs de la visco-

		Solver	
		Faiblement-compressible	Incompressible
Maillage	32x32	$2, 7.10^{-5}$	$2, 6.10^{-6}$
	64x64	$8, 6.10^{-7}$	$8, 7.10^{-8}$
	128x128	$2, 5.10^{-8}$	$3, 4.10^{-9}$

TABLE 4.13 – Tourbillons de Taylor-Green. Viscosités numériques en $m^2.s^{-1}$

sité numérique de la formulation faiblement-compressible sont globalement 10 fois supérieures à celle de l'approche incompressible, soit un rapport cohérent avec l'estimation théorique faite dans le chapitre 1 (cf. section 1.3) qui donne un ordre de grandeur autour de $O(\frac{1}{Ma})$.

Cette diffusion est principalement attribuable au décentrement effectué pour stabiliser le schéma d'advection. En pratique, sur un écoulement visqueux, elle s'ajouterait à la viscosité physique. En relevant les ordres de grandeurs du tableau 4.13, nous pouvons affirmer que pour des écoulements laminaires, les effets de la viscosité numérique seront sans doute masqués par la diffusion réelle. Pour des écoulements à très haut Reynolds, la question reste cependant ouverte.

Convection de vortex

Nous reprenons le benchmark de convection de vortex proposé par le CERFACS avec les mêmes paramètres physiques. Nous cherchons cette fois-ci à calculer un ordre de convergence et à comparer les résultats entre l'approche faiblement-compressible et celle incompressible. Les conditions aux limites périodiques sont un avantage pour le solveur incompressible qui n'est pas pénalisé avec des conditions aux limites de Dirichlet sur la vitesse.

L'étude de convergence est menée sur des grilles cartésiennes uniformes allant de 32 à 512 cellules par direction. L'erreur par rapport à la solution analytique est quantifiée par la norme L_2 après 10 passages dans la boîte périodique :

$$\begin{cases} \epsilon_{\|u\|_2} = \sqrt{\frac{\int_D |\mathbf{u} - \mathbf{u}_{ref}|^2}{\int_D |\mathbf{u}_{ref}|^2}} \\ \epsilon_{\|p\|_2} = \sqrt{\frac{\int_D |p - p_{ref}|^2}{\int_D |p_{ref}|^2}} \end{cases} \quad (4.21)$$

L'erreur est intégrée sur un cercle de rayon égal à 4 fois le rayon du vortex. En incompressible, l'erreur de la pression est mesurée sur la pression reconstruite à partir des pressions obtenues à chaque sous-étape Runge-Kutta (cf. équation (2.96)). En faiblement-compressible, la vitesse nominale du son est fixée à $c_0 = 350m/s$.

La figure 4.21 présente les champs de pression à $t = 0$ et à $t = 10 \frac{L}{U_0}$ pour les deux formulations sur un maillage 80×80 . Nous pouvons d'ores et déjà constater que le vortex de la formulation faiblement-compressible apparaît plus atténué par rapport à la formulation incompressible.

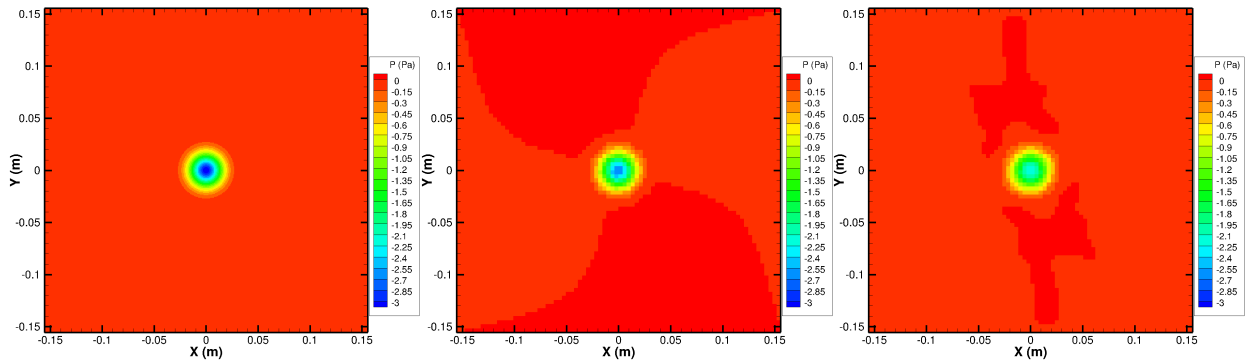


FIGURE 4.21 – Convection de vortex. Champ de pression à $t = 0$ (gauche) et $t = 10 \frac{L}{U_0}$ avec 80×80 cellules en incompressible (milieu) et faiblement-compressible (droite).

La figure 4.22 présente les résultats de convergence. Pour des maillages grossiers, le vortex est trop détérioré pour mettre en évidence un quelconque ordre de convergence. A partir de 80 cellules par direction, le taux de convergence est supérieur à 4 pour la vitesse des deux formulations. La pression est également d'ordre 4 en faiblement-compressible. Pour la formulation incompressible, la pression converge à un ordre équivalent même si à partir de 250 cellules un seuil de saturation semble atteint. Le problème persiste en diminuant le critère d'arrêt sur les résidus de la résolution de l'équation de Poisson. Il est assez remarquable

de constater une convergence à l'ordre 4 pour la pression alors que celle-ci n'est discrétisée qu'à l'ordre 2 en espace.

Par ailleurs on constate que pour des maillages d'environ 350 cellules par direction, l'erreur sur la pression et la vitesse sont globalement identiques pour les deux formulations. Cependant, l'approche incompressible est cinq fois plus rapide à exécuter une simulation pour ce maillage. A partir de 350 cellules par direction, l'approche faiblement-compressible est donc plus précise. Néanmoins, à ce niveau de discrétisation spatiale, l'erreur d'amplitude sur la pression au bout de 10 passages dans la boîte périodique est seulement de 0.1% pour les deux formulations. Il n'est donc pas pertinent de les comparer jusqu'à un tel niveau de précision qui ne sera, de toute façon, pas poursuivi pour les applications d'ingénierie que nous visons.

Enfin il apparaît que pour ce cas la vitesse du son en faiblement-compressible peut être abaissée jusqu'à $35m/s$, c'est-à-dire Mach 1 sans perte sur la précision numérique ni instabilités numériques. Ceci s'explique par l'ordre de grandeur du terme convectif qui est nettement supérieur à celui du gradient de pression. Ce cas test n'est donc pas très pertinent pour quantifier un gain CPU entre les deux formulations puisqu'il est possible d'avoir un pas de temps équivalent pour les deux approches.

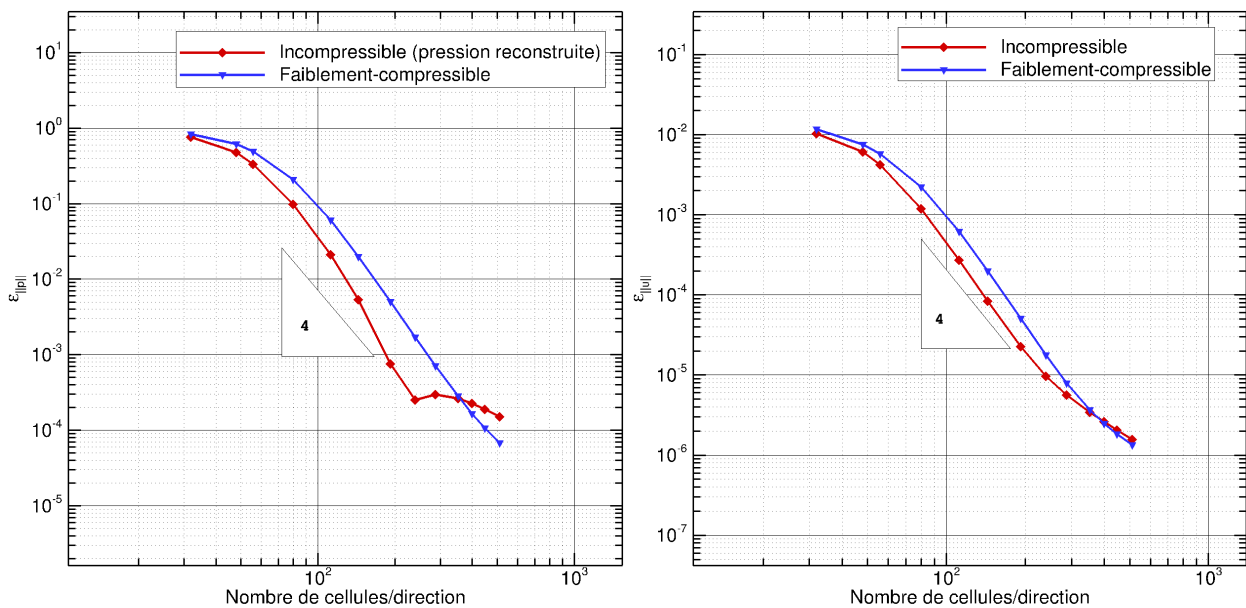


FIGURE 4.22 – Convection de vortex. Convergence en pression (gauche) et vitesse (droite).

4.3.2 Écoulement de Taylor-Couette

Ce troisième cas étudié est l'écoulement de Taylor-Couette entre deux cylindres concentriques. La géométrie est décrite par la figure 4.23, $R_1 = 0.5m$, $\Omega_1 = 1rad/s$, $R_2 = 1m$ et $\Omega_2 = 0.25rad/s$.

Dans cette configuration, l'écoulement reste stable. Sinon, la force centrifuge peut devenir prédominante par rapport aux efforts visqueux et entraîner des recirculations verticales. L'écoulement ne serait alors plus 2D. La viscosité cinématique est fixée à $\nu = 10^{-3}m^2/s$ et la densité à $\rho = 1kg/m^3$. Le nombre de Reynolds basé sur la vitesse de rotation et le rayon du cylindre intérieur vaut :

$$Re = \frac{\Omega_1 R_1^2}{\nu} = 250 \quad (4.22)$$

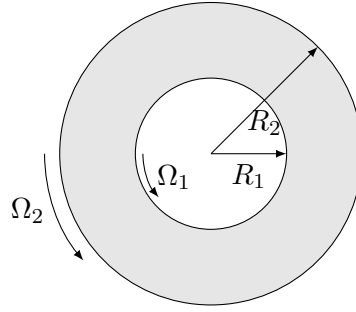


FIGURE 4.23 – Écoulement de Taylor-Couette

L'écoulement stationnaire est axisymétrique et une solution analytique est disponible pour la vitesse ortho-radiale et la pression :

$$\begin{cases} u_\theta(r) = \frac{A}{2}r + \frac{B}{r} \\ p(r) = p_0 + \rho \left(\frac{A^2}{8}r^2 + AB \ln(r) - \frac{B^2}{2r^2} \right) \end{cases} \quad (4.23)$$

où $A = 2 \frac{\Omega_2 R_2^2 - \Omega_1 R_1^2}{R_2^2 - R_1^2}$ et $B = -(\Omega_2 - \Omega_1) \frac{R_1^2 R_2^2}{R_2^2 - R_1^2}$. La pression est définie à une constante près p_0 .

Le domaine de calcul est carré, périodique, de côté $L = 3m$. Une attention particulière est prêtée à l'impact de la méthode de frontière immergée sur la précision globale. Des premières simulations sont réalisées pour des grilles uniformes de $N_x \times N_y = 64 \times 64$ cellules. Les champs de pression et de vitesse obtenus sont présentés sur la figure 4.24. Les différences entre les deux approches au voisinage des frontières solides sont notables.

Les profils de vitesse et de pression en fonction du rayon sont tracés sur la figure 4.25. Ces profils sont extraits par défaut sur la droite $y = 0$. Ils semblent indiquer que la formulation incompressible a un meilleur accord avec la solution analytique. Il est intéressant de relever que le comportement de la pression à l'intérieur des solides est bien différent pour les deux approches en particulier pour le cylindre intérieur.

En formulation incompressible la pression est constante à l'intérieur du cylindre, ce qui correspond au comportement attendu. En effet, avec la méthode de frontière immergée employée, la vitesse prédite \mathbf{u}^* à l'intérieur du cylindre est une vitesse de solide rigide en rotation et est donc à divergence nulle (cf. section 2.4). L'équation de Poisson se réduit donc à l'équation de Laplace à l'intérieur du cylindre $\Delta\psi = 0$, avec ψ l'incrément de pression. Ainsi la pression est une fonction harmonique dans le cylindre intérieur. Par ailleurs, compte-tenu du caractère axisymétrique de l'écoulement, elle supposée constante le long de la frontière solide. En vertu du principe du maximum, la pression théoriquement attendue à l'intérieur du cylindre est donc constante. En pratique la pression n'est pas strictement constante à partir de la frontière mais évolue un peu dans la région de transition qui correspond à la zone de régularisation du terme source par la fonction Heaviside lissée. Ce lissage permet de rendre la pression continue et différentiable à la traversée de la frontière.

En formulation faiblement-compressible, l'équation de continuité n'est pas contrainte par un terme source à l'intérieur du corps et s'écrit :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}_{solide}) = 0 \quad (4.24)$$

Parce que $\nabla \cdot \mathbf{u}_{solide} = 0$, ρ satisfait une équation de convection instationnaire basique :

$$\frac{\partial \rho}{\partial t} + \mathbf{u}_{solide} \cdot \nabla \rho = 0 \quad (4.25)$$

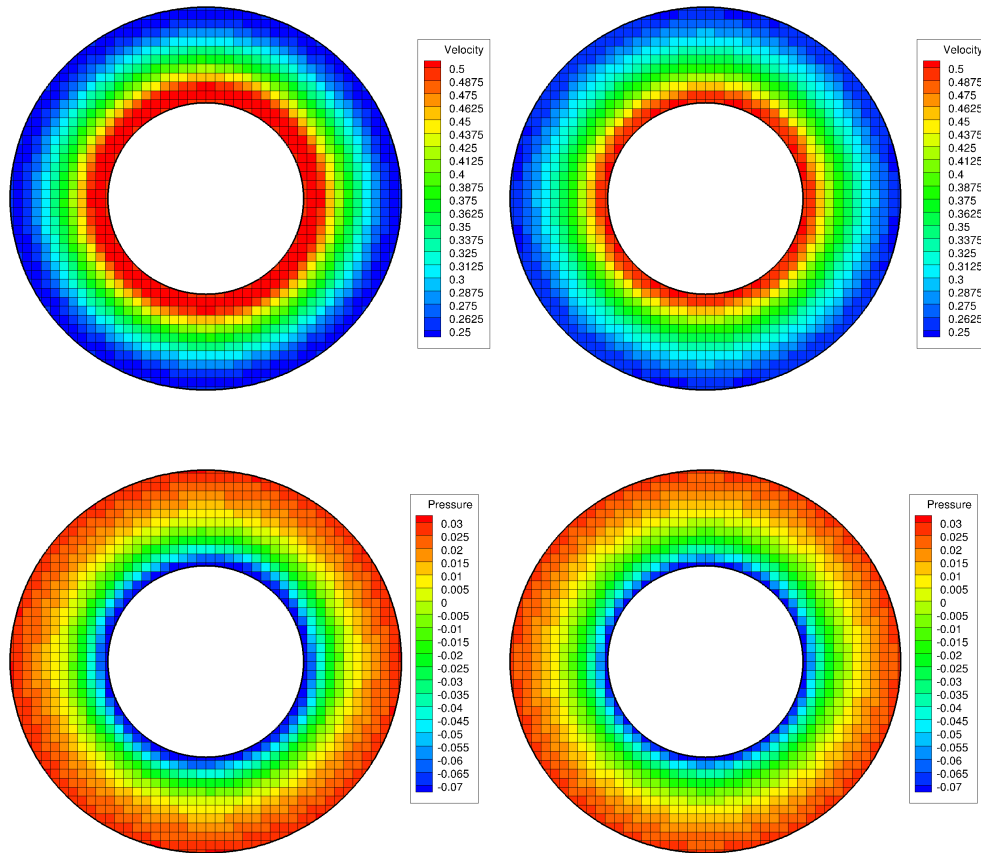


FIGURE 4.24 – Écoulement de Taylor-Couette. Comparaison des champs de vitesse et de pression à $t = 230s$ en formulation faiblement-compressible (gauche) et incompressible (droite)

En conséquence, en régime stationnaire, la pression attendue à l'intérieur du cylindre doit être constante. En pratique, le solveur de Riemann introduit de la diffusion numérique dans l'équation de continuité et explique le profil de pression observé sur la figure 4.25. Au centre du cylindre, $\mathbf{u}_{solide} = 0$ et la densité n'évolue pas en temps. La pression est donc égale à la pression de référence.

Ainsi les différences observées sur les profils de vitesses et de pression peuvent être imputées aux comportements respectifs de la pression des schémas à l'intérieur du cylindre.

Pour l'étude de convergence, on choisit des maillages de 32 à 256 cellules par direction. On mesure la norme L_2 de l'erreur relative par rapport à la solution analytique (cf. équation 4.21).

Les résultats sont résumés dans les tableaux 4.14 et 4.15 et confirment la tendance esquissée ci-dessus.

L'ordre de convergence en formulation incompressible est proche de 1.8 pour la vitesse et la pression alors qu'il n'est que de 1.35 en formulation faiblement-compressible. Outre un ordre de convergence plus élevé, l'approche incompressible donne des erreurs plus faibles.

Par ailleurs, le taux de convergence pour les deux approches est plus élevé que ce qui est attendu. Une norme plus restrictive comme la norme L_∞ , conduirait probablement à des taux de convergence proche de 1.

Pour aller plus loin dans la comparaison entre les deux approches, nous affichons le temps CPU d'exécution en fonction de l'erreur (Fig. 4.26). Compte-tenu de l'allure des courbes nous pouvons construire une

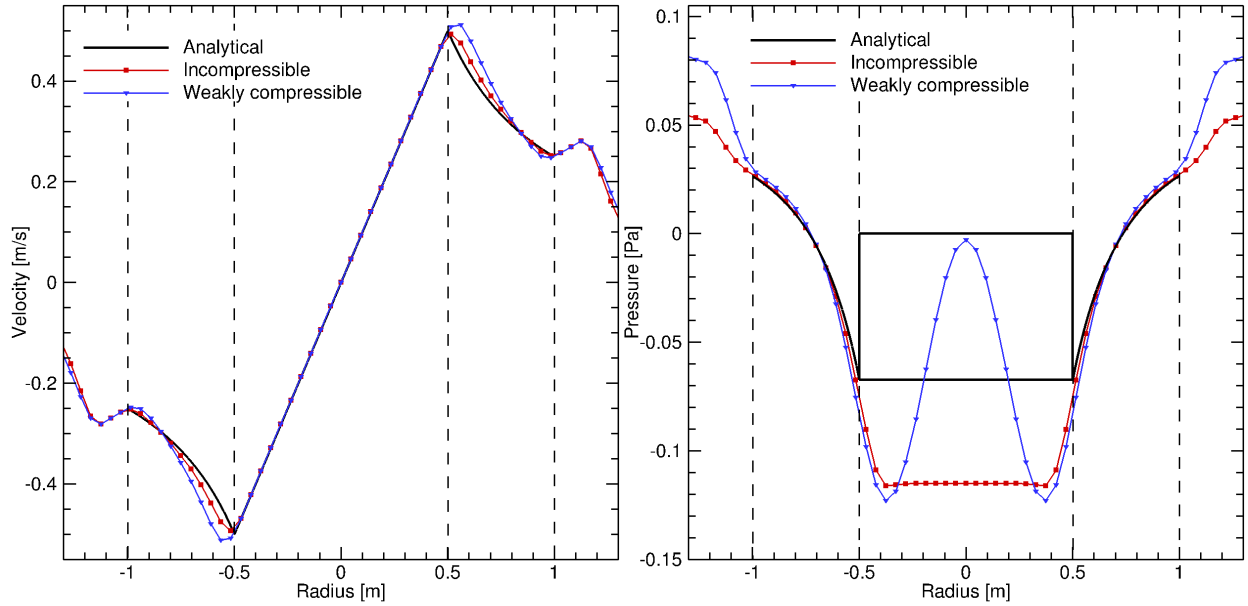


FIGURE 4.25 – Écoulement de Taylor-Couette. Comparaison des profils de vitesse (gauche) et de pression (droite) à $t = 230s$

$N_x \times N_y$	$\epsilon_{\ u\ _2}$	Ordre	$\epsilon_{\ p\ _2}$	Ordre
32×32	1.21×10^{-1}	-	2.79×10^{-1}	-
48×48	7.55×10^{-2}	1.17	1.66×10^{-1}	1.28
64×64	4.68×10^{-2}	1.66	9.44×10^{-2}	1.96
96×96	2.52×10^{-2}	1.53	5.08×10^{-2}	1.53
128×128	1.58×10^{-2}	1.62	3.08×10^{-2}	1.75
160×160	9.68×10^{-3}	2.19	1.83×10^{-2}	2.34
192×192	7.03×10^{-3}	1.76	1.44×10^{-2}	1.29

TABLE 4.14 – Écoulement de Taylor-Couette. Convergence spatiale en maillage pour la formulation incompressible

$N_x \times N_y$	$\epsilon_{\ u\ _2}$	Ordre	$\epsilon_{\ p\ _2}$	Ordre
32×32	1.97×10^{-1}	-	4.74×10^{-1}	-
48×48	1.51×10^{-1}	0.65	3.39×10^{-1}	0.83
64×64	1.07×10^{-1}	1.19	2.28×10^{-1}	1.37
96×96	6.40×10^{-2}	1.27	1.30×10^{-1}	1.39
128×128	4.32×10^{-2}	1.37	8.43×10^{-2}	1.50
160×160	3.26×10^{-2}	1.26	6.21×10^{-2}	1.37
192×192	2.57×10^{-2}	1.31	4.87×10^{-2}	1.33

TABLE 4.15 – Écoulement de Taylor-Couette. Convergence spatiale en maillage pour la formulation faiblement-compressible

fonction qui lie l'erreur au temps CPU sous la forme :

$$\text{Temps CPU} = \alpha \text{erreur}^\beta \quad (4.26)$$

Une régression linéaire au sens des moindres carrés sur $\log(\text{Temps CPU})$ et $\log(\text{erreur})$ permet de déterminer les coefficients α et β pour chaque formulation. A partir de ces fonctions nous pouvons construire une nouvelle courbe du rapport des temps CPU faiblement-compressible et incompressible en fonction de l'erreur. Cette courbe donne le gain CPU de l'approche incompressible par rapport à l'approche faiblement-compressible à précision équivalente et est présentée sur la figure 4.26.

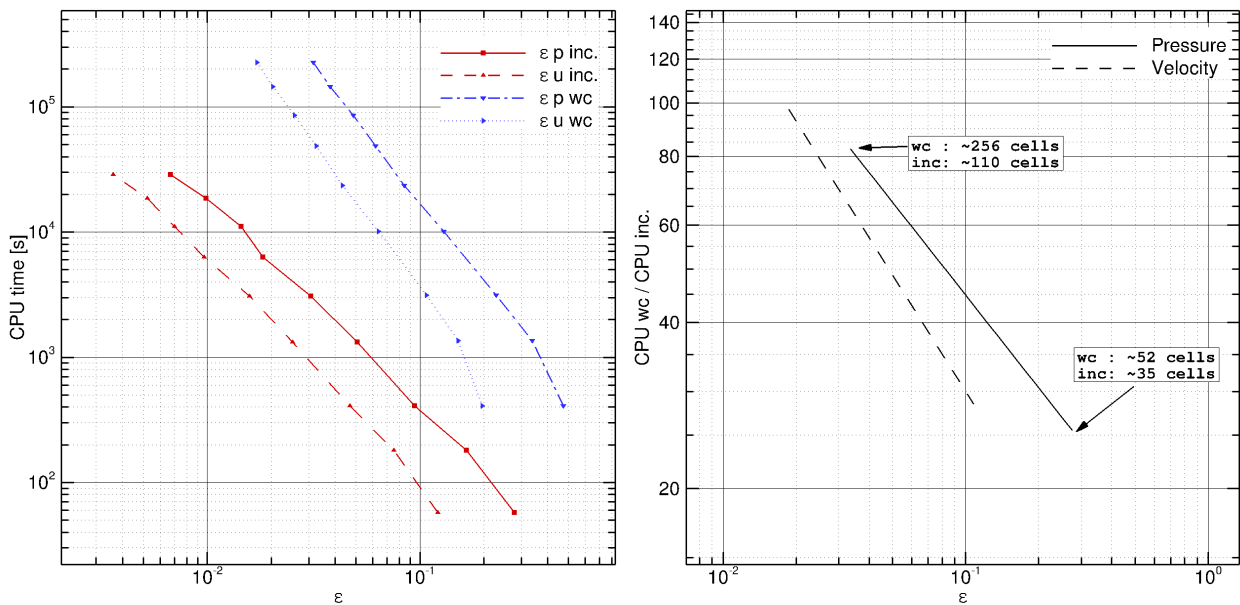


FIGURE 4.26 – Écoulement de Taylor-Couette. Comparaison du temps CPU en fonction de l'erreur (gauche) et du gain incompressible/faiblement-compressible (droite)

Ces courbes donnent clairement un avantage à l'approche incompressible avec un gain de 20 pour les maillages les plus grossiers et de près de 60 pour une erreur relative de 5%. Ce gain significatif doit être pris avec précaution et il serait dangereux de conclure à partir des résultats de ce seul cas test. La faible précision de la formulation faiblement-compressible semble liée à la configuration de l'écoulement et au champ de pression à l'intérieur du cylindre. Sans doute que la formulation faiblement-compressible n'est pas apte à simuler un écoulement interne. Le cas test suivant concerne justement un écoulement externe et instationnaire d'une complexité supérieure.

4.3.3 Écoulement autour d'un profil NACA

Le dernier cas test de comparaison étudié est celui de l'écoulement à Reynolds 1400 autour d'un profil NACA symétrique 0010 sous un angle d'incidence de 30 degrés. Le nombre de Reynolds est défini par :

$$Re = \frac{U_0 c}{\nu} \quad (4.27)$$

avec U_0 la vitesse incidente uniforme, c la corde du profil et ν la viscosité cinématique. Pour ce nombre de Reynolds, l'écoulement est instationnaire et des tourbillons alternés s'échappent dans le sillage. La paramétrisation du domaine de calcul est présentée figure 4.27. Un raffinement statique avec 6 niveaux est utilisé.

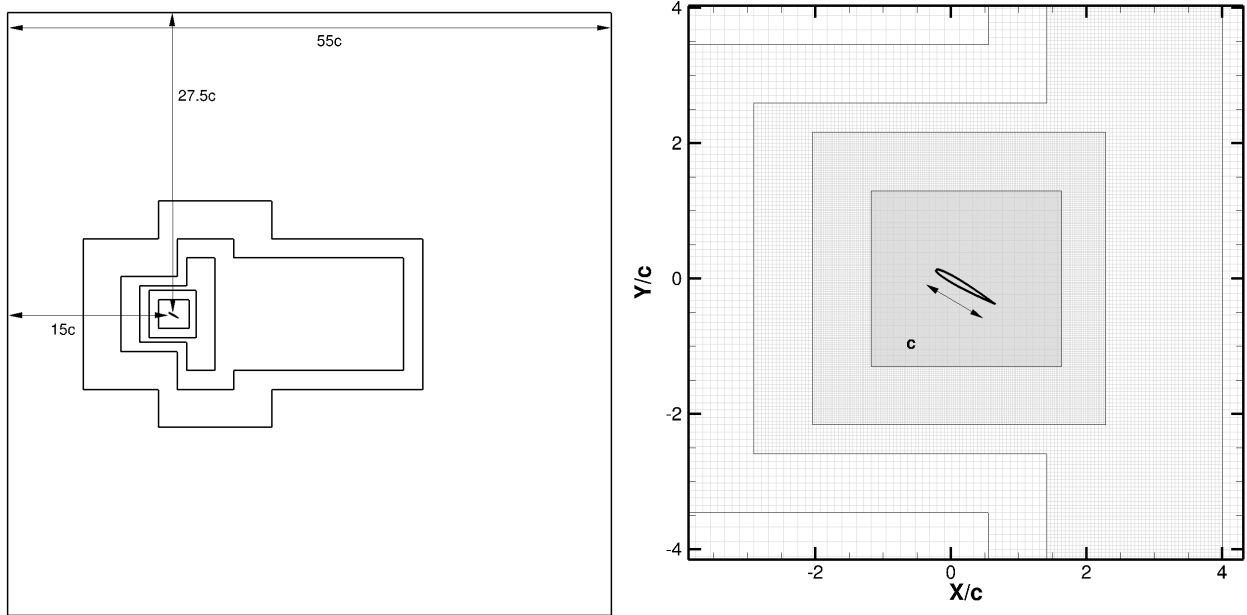


FIGURE 4.27 – Écoulement 2D autour d'un profil NACA à $Re = 1400$. Domaine de calcul avec dimensions adimensionnalisées par la corde du profil (gauche) et maillage au voisinage du solide avec $\frac{c}{\Delta x} = 75$ (droite)

Trois résolutions spatiales sont étudiées dans un premier temps $\frac{c}{\Delta x} = 75$, $\frac{c}{\Delta x} = 130$ et $\frac{c}{\Delta x} = 260$. Le raffinement par bloc est identique pour chaque discrétisation, le nombre de cellules est adapté en faisant varier le nombre de cellules par bloc et par direction : 8, 14 et 28. Ainsi le nombre d'interfaces de raffinement est constant. Encore une fois les calculs sont menés dans les mêmes conditions pour les deux formulations. La vitesse du son est fixée à $30U_0$ pour l'approche faiblement-compressible. Le temps de simulation total est de $\frac{tU_0}{c} = 100$ pour permettre au régime périodique de s'installer après une période transitoire. Les calculs sont réalisés sur 16 cœurs.

Nous nous intéressons dans un premier temps au coefficient de traînée C_d . La figure 4.28 présente l'historique du signal de traînée. Après un régime transitoire de $\frac{tU_0}{c} = 50$, l'écoulement devient périodique. Pour cet écoulement complexe il n'existe pas de solution analytique à laquelle comparer les deux approches. Nous choisissons comme référence une solution numérique donnée par la méthode DVH (Diffused Vortex Hydrodynamics) développée par Colagrossi et Graziani [Rossi et al., 2015]. Le code WCCH a déjà été comparé avec succès à la méthode DVH (en version faiblement-compressible) dans le cadre d'un partenariat académique avec l'université de Rome "La Sapienza" [Rossi et al., 2018]. La résolution spatiale du calcul par la méthode DVH est de $\frac{c}{\Delta x} = 400$ proche des frontières solides et décroît rapidement en s'éloignant du profil.

Pour le premier maillage ($\frac{c}{\Delta x} = 75$), les résultats des formulations faiblement-compressible et incompressible reproduisent bien un motif périodique. La fréquence de la première harmonique correspondant au lâcher de tourbillons dans le sillage du profil NACA semble assez bien prédite par les deux approches. Néanmoins, les amplitudes des différentes harmoniques qui composent le signal restent éloignées par rapport à la référence DVH. Les deux autres maillages plus fins, $\frac{c}{\Delta x} = 130$ et $\frac{c}{\Delta x} = 260$ donnent des résultats qui convergent bien vers le signal de référence. Il est difficile de départager les deux formulations sur la base du signal des efforts. Aucune des deux ne semble converger plus rapidement que l'autre.

Nous traçons ensuite le champ de vorticité pour les différentes discrétisations à l'instant $\frac{tU_0}{D} = 100$ (Fig. 4.29). Si le maillage le plus grossier donne un résultat qui a l'air assez éloigné du champ donné par la méthode DVH, les deux autres maillages montrent que les détachements tourbillonnaires de la couche

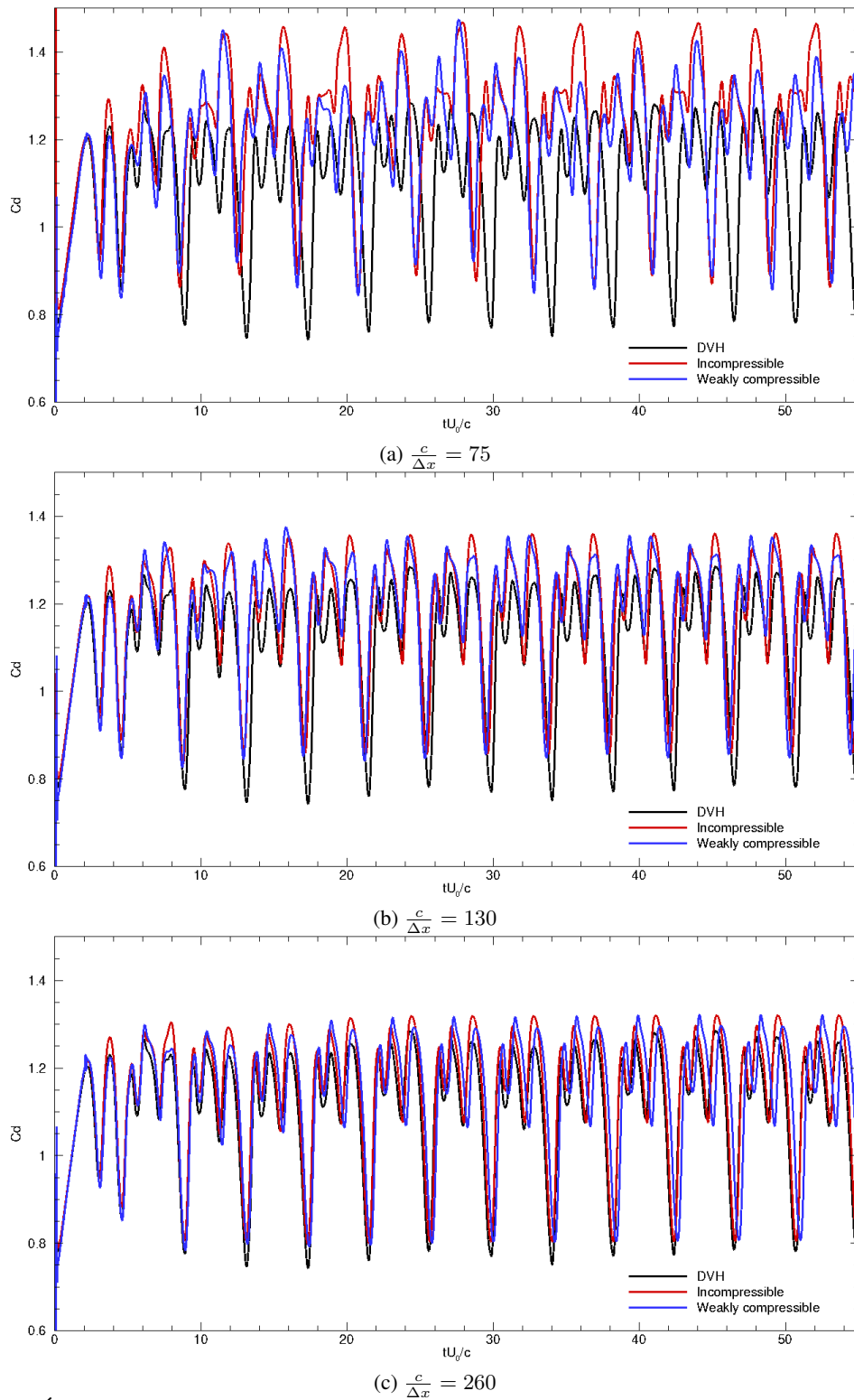


FIGURE 4.28 – Écoulement 2D autour d'un profil NACA à $Re = 1400$. Coefficient de traînée en fonction du temps.

4.3. COMPARAISON DES FORMULATIONS INCOMPRESSIBLE ET FAIBLEMENT-COMPRESSIBLE

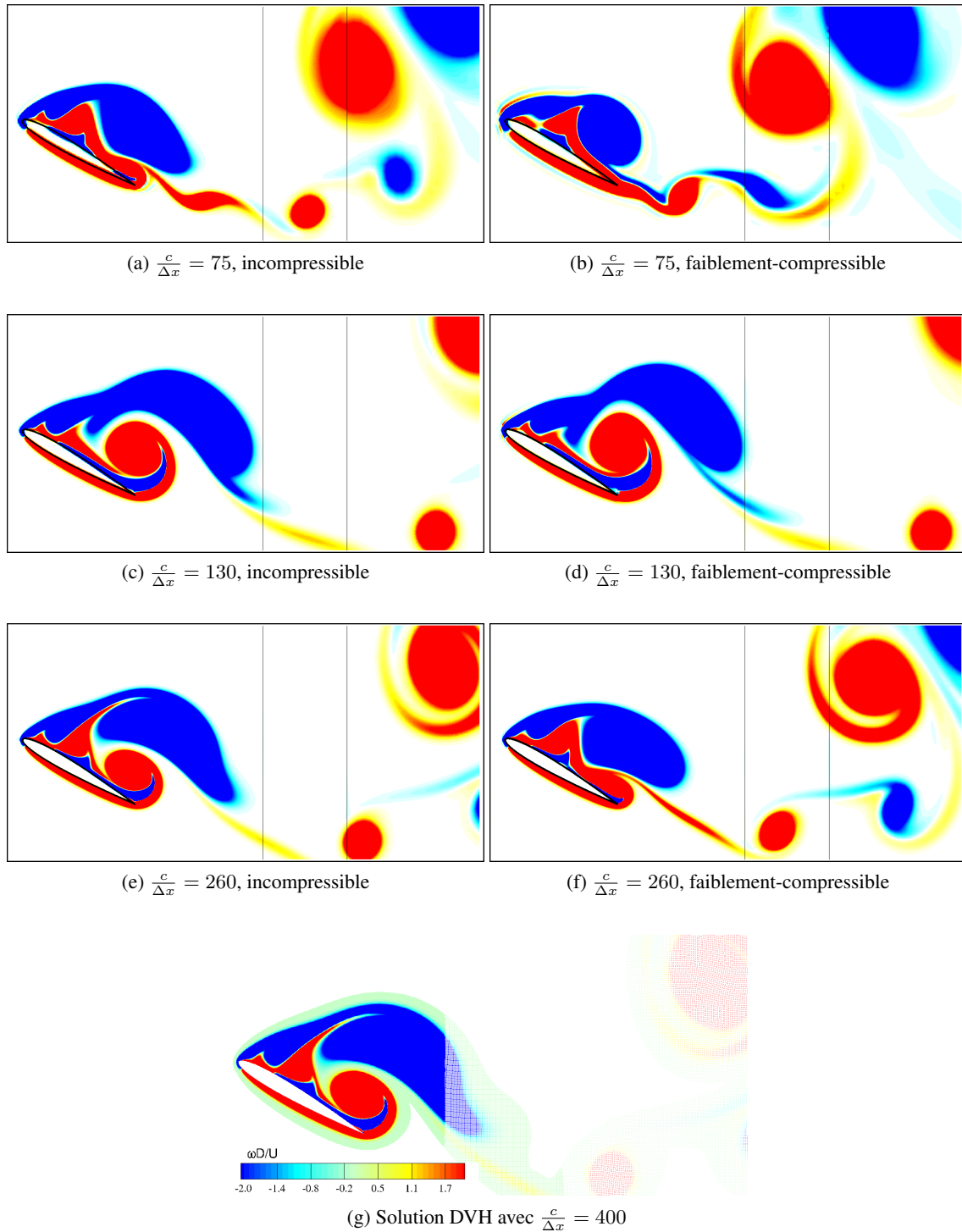


FIGURE 4.29 – Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de vorticité à $\frac{tU_0}{D} = 100$. Comparaison incompressible (gauche), faiblement-compressible (droite), DVH (bas)

limite du profil sont en accord avec la référence. Sur la grille la plus fine, la solution incompressible apparaît malgré tout mieux synchronisée avec la solution DVH que celle de la formulation faiblement-compressible.

La figure 4.30 montre la pression adimensionnalisée $\frac{P}{\frac{1}{2}\rho U_0^2}$ et confirme la ressemblance des résultats des

deux approches. Dans ce mode, les tourbillons se détachent dans deux directions distinctes [Rossi et al., 2018]. Des tourbillons de forte intensité se situent dans le sillage à 30 degrés par rapport à la direction principale de l'écoulement tandis que des tourbillons de taille plus faible sont évacués dans la direction horizontale.

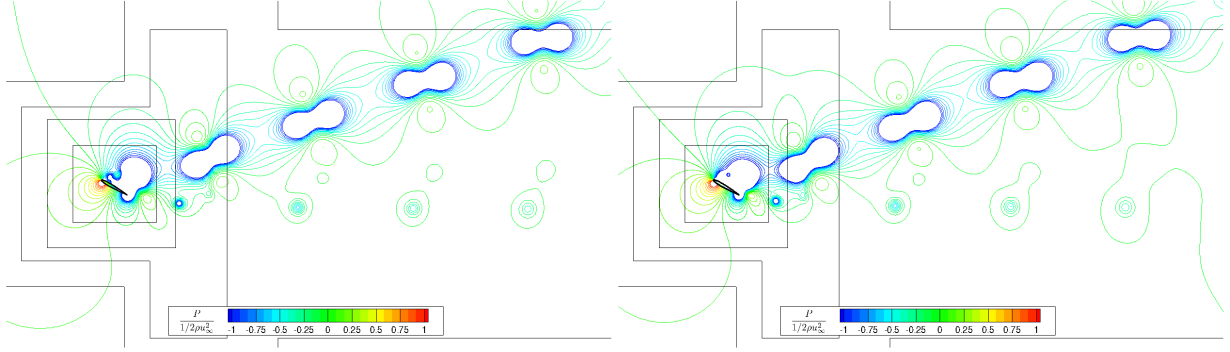


FIGURE 4.30 – Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de pression à $\frac{tU_0}{D} = 100$ pour $\frac{corde}{\Delta x} = 185$, incompressible (gauche) and faiblement-compressible (droite).

Le tableau (Tab. 4.16) reprend les temps d'exécution des calculs précédents ainsi que les pas de temps moyen. A la lecture de ces résultats nous constatons un facteur 10 sur le temps CPU total entre l'approche incompressible et l'approche faiblement-compressible.

Maillage $\frac{corde}{\Delta x}$	Nombre total de cellules	Incompressible Temps CPU (h)	Fbl.-compressible Temps CPU (h)	Incompressible Pas de temps (s)	Fbl.-compressible Pas de temps (s)
75	103552	1.5	15.9	3.20e-03	2.12e-04
130	317128	7.7	87.4	1.85e-03	1.21e-04
260	1268512	65.5	873.5 (estimation)	9.00e-04	6.05e-05

TABLE 4.16 – Écoulement 2D autour d'un profil NACA à $Re = 1400$. Paramètres et temps de simulations

Cet écart sur le temps CPU s'explique en partie par le pas de temps de moyen qui est 15 fois plus grand en incompressible. En effet le gain de l'approche incompressible sur celle faiblement-compressible pour un calcul effectué dans des conditions identiques (même maillage, mêmes paramètres numériques comme le nombre CFL) peut être estimé par :

$$\frac{TempsCPU^{comp.}}{TempsCPU^{inc}} = \frac{t_{\Delta t}^{comp} n_{\Delta t}^{comp}}{t_{\Delta t}^{inc} n_{\Delta t}^{inc}} = \frac{t_{\Delta t}^{comp} \Delta t^{inc}}{t_{\Delta t}^{inc} \Delta t^{comp}} \quad (4.28)$$

où $t_{\Delta t}$ désigne le temps CPU d'un pas de temps, Δt est la taille du pas de temps et $n_{\Delta t}$ est le nombre de pas de temps. L'indice $comp$ fait référence à l'approche faiblement-compressible et inc à l'approche incompressible. Avec un nombre CFL identique pour les deux formulations, le rapport des tailles de pas de temps donne :

$$\frac{\Delta t^{inc}}{\Delta t^{comp}} = 1 + \frac{c}{u_{max}} \quad (4.29)$$

avec c la vitesse du son et u_{max} la vitesse maximale dans l'écoulement. En considérant que $c \approx 10u_{max}$ (i.e. $Ma = 0.1$) :

$$\frac{\Delta t^{inc}}{\Delta t^{comp}} \approx 11 \quad (4.30)$$

Cela signifie que l'approche faiblement-compressible requiert 11 fois plus d'itérations du schéma d'avance en temps.

Dans le cas présent, le rapport de 15 trouvé est bien supérieur à celui attendu d'après (4.30). Cela s'explique par le nombre de Mach de l'écoulement qui se situe autour de 0.08. Néanmoins des tests avec une vitesse du son d'environ $10u_{max}$ donnaient de moins bons résultats sur l'erreur de la traînée et de la portance si bien qu'une vitesse du son supérieure et un pas de temps plus petit ont été choisis.

D'autre part, le coût d'un pas de temps du schéma incompressible est environ 1.5 fois plus élevé comme le montre la figure 4.31. Cette figure compare le coût CPU d'un pas de temps par nombre de cellules pour les deux formulations en fonction du nombre de cellules dans la corde du profil. Nous avons ajouté d'autres résolutions spatiales pour étudier la sensibilité du temps CPU au nombre total de cellules, de $\frac{c}{\Delta x} = 33$ à $\frac{c}{\Delta x} = 260$ soit de 26000 à 1260000 cellules au total. Si nous omettons le premier maillage qui correspond à un nombre de 4 cellules par bloc et par direction, nous observons que le coût CPU moyen d'un pas de temps par cellule est presque constant en formulation incompressible. En formulation faiblement-compressible, il est aussi constant jusqu'à une valeur d'environ 20 cellules par bloc par direction. Sa valeur augmente ensuite jusqu'à se stabiliser. Ce résultat est intéressant et indique que la complexité algorithmique approche $O(N)$ y compris pour la formulation incompressible.

Finalement d'après la figure 4.31, le rapport moyen des temps CPU d'un pas de temps entre les deux approches est de :

$$\frac{t_{\Delta t}^{comp}}{t_{\Delta t}^{inc}} \approx \frac{1}{1.5} \quad (4.31)$$

Le temps CPU d'un pas de temps incompressible est supérieur à cause du temps supplémentaire nécessaire à la résolution de l'équation de Poisson. Nous traçons par ailleurs le pourcentage du temps CPU consacré à cette résolution du système linéaire. Cette fraction est, là aussi, assez indépendante du nombre de cellules et se situe autour de 30%. Ce pourcentage est notoire et montre que la résolution de l'équation de pression ne constitue pas la majeure partie du temps CPU d'un pas de temps. En incompressible une partie importante du temps CPU d'un pas de temps est consacrée à la première étape du schéma de projection et au calcul du terme convectif. En effet dans le code, les reconstructions WENO sont assez coûteuses et mobilisent d'importantes ressources en mémoire puisque les coefficients WENO sont pré-calculés et stockés dans de grands tableaux. L'accès mémoire n'est pas forcément optimal¹. A noter que ce problème CPU affecte indifféremment les deux approches puisque l'algorithme est globalement le même dans les deux cas.

Finalement en faiblement-compressible le pas de temps est 15 fois plus petit alors que le code de cette formulation exécute un pas de temps 1.5 fois plus rapidement. A maillage identique l'approche incompressible fournit donc un résultat 10 fois plus rapidement ($15/1.5$), comme le confirment les temps CPU du tableau 4.16.

Plutôt qu'une comparaison à maillage identique nous cherchons maintenant à départager les approches à précision équivalente. Cette démarche, déjà mise en œuvre sur l'écoulement de Taylor-Couette est plus pertinente. Pour cela, une convergence en maillage est réalisée sur la portance et la traînée avec $\frac{c}{\Delta x}$ variant de 32 à 220. Nous intégrons en temps de $\frac{tU_0}{D} = 50$ à $\frac{tU_0}{D} = 100$ l'erreur relative des signaux temporels de portance et de traînée des deux codes par rapport à la solution de DVH :

$$\epsilon_{Cd} = \sqrt{\int_{t=50}^{t=100} \frac{|Cd - Cd_{DVH}|^2}{Cd_{DVH}^2} dt} \quad (4.32)$$

Nous avons choisi la solution DVH comme référence plutôt que la solution des maillages les plus fins de chaque formulation pour les comparer à une solution identique. La figure 4.32 présente les résultats. Nous

1. Un profilage détaillé du code pourrait être envisagé pour mettre en évidence les portions du code non optimisées et les éventuels défauts de cache préjudiciables aux performances du code.

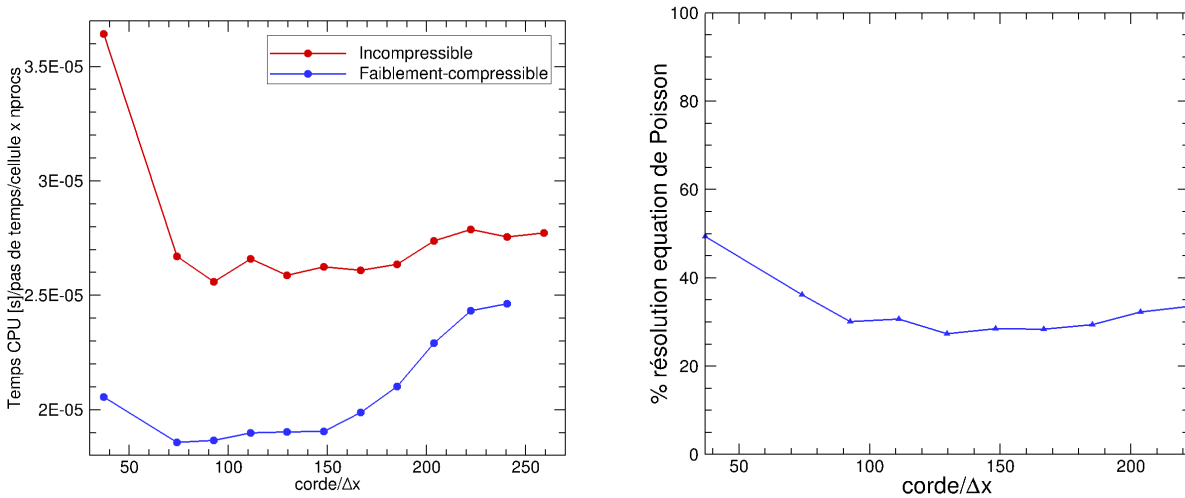


FIGURE 4.31 – Écoulement 2D autour d’un profil NACA à $Re = 1400$. Temps CPU moyen d’un pas de temps par cellule (gauche) et pourcentage du temps CPU consacré à la résolution de l’équation de Poisson en incompressible (droite).

constatons que les deux approches voient leur erreur relative diminuer à mesure que le maillage est raffiné même si la convergence n’est pas toujours régulière.

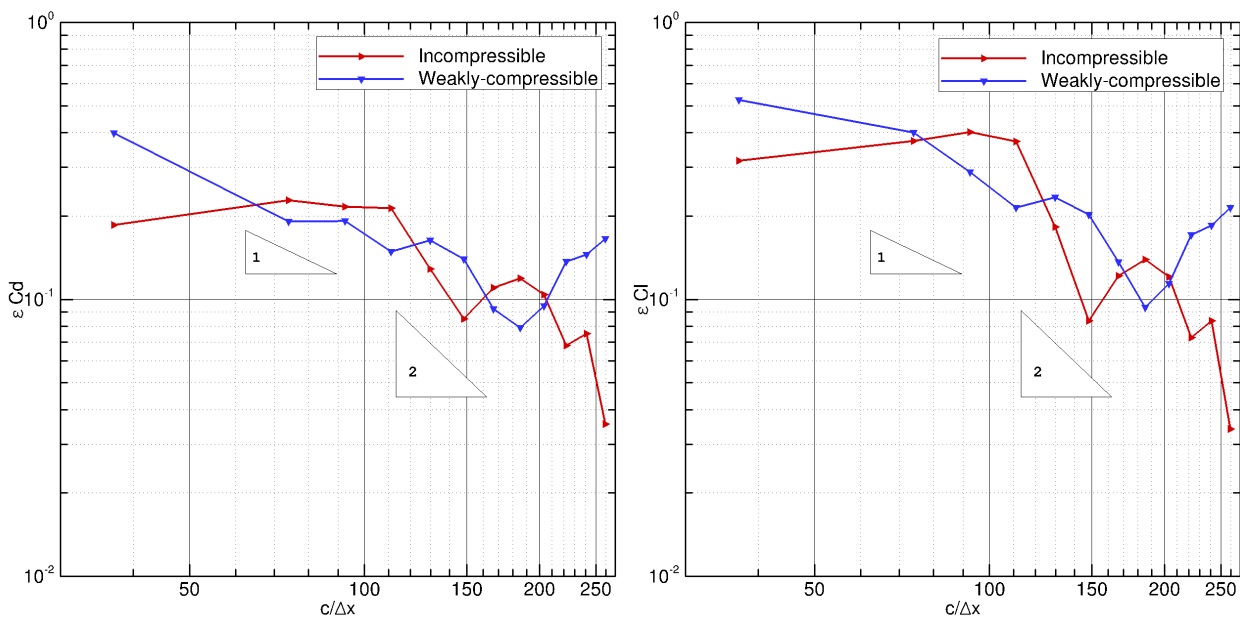


FIGURE 4.32 – Écoulement 2D autour d’un profil NACA à $Re = 1400$. Convergence spatiale sur la traînée (gauche) et sur la portance (droite)

Contrairement à l’écoulement de Taylor-Couette, nous ne pouvons effectuer une régression linéaire au sens des moindres carrés pour déterminer une fonction reliant le temps CPU et l’erreur relative.

Néanmoins nous pouvons estimer approximativement les valeurs des gains. Ils vont de 5 à 20 sur les

coefficients Cd et Cl pour les maillages intermédiaires et raffinés. Les gains sont moins spectaculaires que pour l'écoulement de Taylor-Couette mais restent en faveur de la formulation incompressible et avec des valeurs significatives.

En conclusion, ce cas test témoigne de la pertinence d'une approche complètement incompressible par rapport à celle faiblement-compressible explicite pour la résolution d'écoulements à bas nombre de Mach.

4.3.4 Discussion sur les éléments de calcul haute performance

Performance en calcul parallèle

Une rapide étude de scalabilité est réalisée sur les deux codes pour comparer les performances parallèles des deux approches. L'excellente scalabilité du code WCCH en formulation faiblement-compressible a déjà été mise en évidence par P. Bigay au cours de sa thèse [Bigay, 2015]. Pour l'implémentation incompressible, la première étape du schéma de projection qui calcule la vitesse prédite u^* est très proche de l'implémentation du calcul d'avance en temps en faiblement-compressible. Cette partie de code est donc supposée être scalable comme en faiblement-compressible. La principale incertitude en incompressible concerne la résolution de l'équation de Poisson pour laquelle nous ne maîtrisons pas finement la parallélisation (quelques paramètres avancés de PETSc permettent cependant d'assurer au mieux l'efficacité parallèle). Comme nous l'avons vu au chapitre 2, les méthodes itératives de résolution de système linéaire sont en général scalables pour une itération. Cependant le nombre d'itérations nécessaire à une décroissance suffisante des résidus augmente avec la taille du système linéaire à cause des basses fréquences spatiales qui s'amortissent difficilement. Le préconditionnement ou les méthodes multigrilles réduisent de façon notable la résolution à quelques itérations seulement. Les bibliothèques PETSc et HYPRE implémentent de façon efficace et en parallèle les méthodes de résolution de système linéaire et de préconditionnement. En particulier la bibliothèque BoomerAMG est une implémentation robuste et performante des méthodes multigrilles algébriques [Yang et al., 2002, Cleary et al., 2000]. Ces bibliothèques sont aujourd'hui massivement utilisées dans le monde du calcul numérique scientifique et sont des références en la matière. Elles démontrent une excellente scalabilité. Nous pouvons donc être relativement confiant en ce qui concerne les performances de notre version incompressible du code.

Pour l'étude de scalabilité nous avons retenu un écoulement autour d'un cylindre à Reynolds 40 sur grille uniforme. Les calculs ont été exécutés sur 1 à 10 nœuds bi-processeurs Intel Six-Core Westmere EP 2.8GHz reliés par un réseaux Infiniband (cf. Fig. 4.33). Les entrées/sorties du code ont été limitées au strict minimum pour ne pas altérer les mesures de temps CPU. Le temps de référence est celui sur 1 nœud, i.e. sur 12 cœurs.

Le solveur incompressible a une scalabilité qui n'est pas aussi parfaite que celle de l'approche faiblement-compressible, notamment quand le nombre de cellules par processeur est faible. Néanmoins ces courbes sont plus que satisfaisantes et montrent une efficacité supra-linéaire lorsque le nombre de cellules par cœur est conséquent avec 16 millions de cellules. Un constat similaire avait déjà été rapporté par Almgren et al. [Almgren et al., 2010b]. Dans cet article deux solveurs basés sur la bibliothèque AMR BoxLib [Bell et al., 2012] sont comparés : MAESTRO, un code d'hydrodynamique basé sur une approche bas Mach avec résolution de système linéaire et CASTRO, un code compressible purement explicite. CASTRO a une meilleure scalabilité que MAESTRO qui est un peu plus éloigné d'une scalabilité linéaire idéale.

Pour se faire une idée plus précise de la performance des codes, nous nous intéressons à un deuxième indicateur, "la vitesse de calcul". En effet le "speedup" masque la performance absolue du code. Par exemple il d'autant plus facile d'afficher une excellente scalabilité que le code est lent en exécution séquentielle, du fait du recouvrement des communications par du calcul. A cet égard, la vitesse de calcul exprimée en temps CPU d'un pas de temps par cellule par nombre de processeur offre une meilleure représentation de la

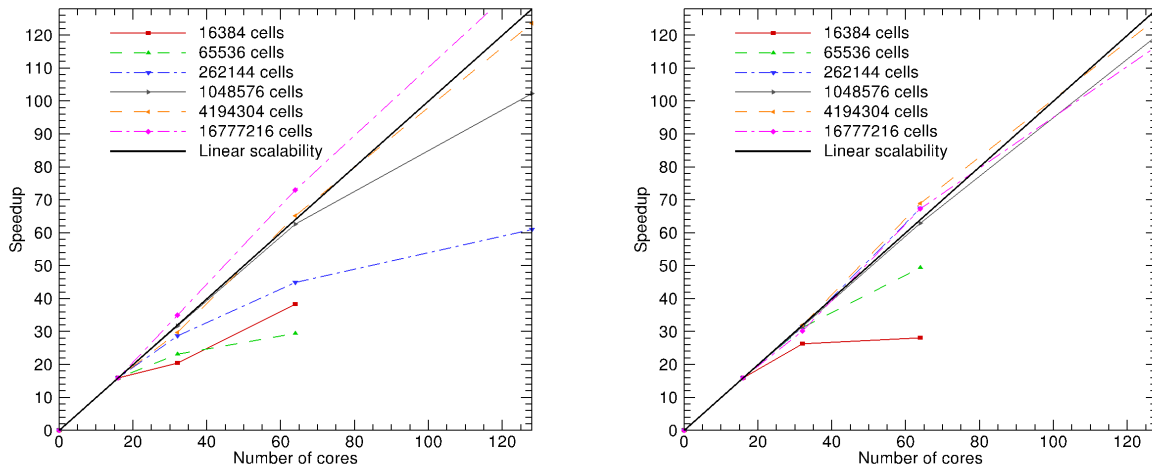


FIGURE 4.33 – Étude de scalabilité. Incompressible (gauche) et faiblement-compressible (à droite)

performance. Elle est tracée sur la figure 4.34 en fonction du nombre de cœurs.

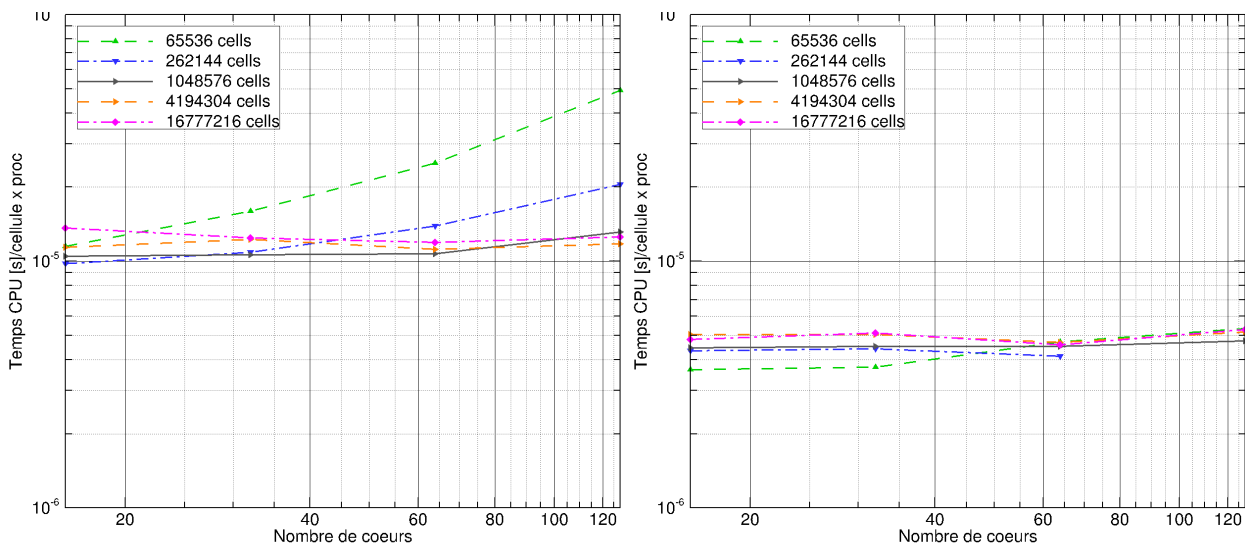


FIGURE 4.34 – Temps CPU d'un pas de temps par cellule et par processeur. Incompressible (gauche) et faiblement-compressible (à droite)

En formulation incompressible, nous constatons que pour des maillages grossiers, la vitesse de calcul augmente avec le nombre de cœurs. Pour autant, pour les maillages les plus fins, les courbes semblent s'établir autour d'une valeur constante qui donne donc la performance absolue du code à $10\mu\text{s}/\text{cellule}/\text{proc}$. L'allure des courbes est à mettre en lien avec les remarques déjà formulées précédemment. En revanche, en faiblement-compressible, la vitesse est constante quel que soit le maillage, à $5\mu\text{s}/\text{cellule}/\text{proc}$. Enfin il reste difficile de conclure sur ces valeurs de vitesse sans avoir des éléments de comparaison comme les vitesses que donneraient d'autres codes de calculs aux caractéristiques similaires (schémas d'ordre élevé notamment) sur ce cas test. On notera aussi que pour ce cas, la formulation incompressible est deux fois plus lente que

la formulation faiblement-compressible et la résolution de l'équation de Poisson représente environ 50% du temps CPU total. Ce pourcentage, plus élevé que celui relevé précédemment sur le cas d'écoulement autour du profil NACA0010, s'explique par l'absence de raffinement local de maillage qui accroît le temps CPU consacré aux reconstructions WENO5.

L'incompressible toujours plus avantageux que le faiblement-compressible ?

Nous venons de voir que l'approche incompressible offre un gain certain sur la restitution des résultats à précision équivalente par rapport à l'approche faiblement-compressible. Nous sommes en droit cependant de questionner la validité de ces conclusions étant donné qu'elles sont liées à une implémentation précise. Notamment qu'en serait-il si l'algorithme de la formulation faiblement-compressible était optimisé et accéléré ?

En pratique, accélérer le code faiblement-compressible d'un facteur donné ne suffit pas pour garantir que le gain du schéma incompressible sur la formulation faiblement-compressible sera divisé par ce facteur. En effet, comme cela a déjà été mentionné, l'étape de prédiction du schéma de projection de Chorin est très similaire à une itération en temps du solveur faiblement-compressible. Dans le code, les deux approches font appel à un nombre important de routines communes. Les quelques différences de traitement concernent la localisation des variables (maillage MAC contre maillage colocalisé) et les interfaces de raffinement.

Prenons un gain initial dans l'intervalle [5, 20] en faveur de l'incompressible (ces valeurs correspondent au gain qui a été relevé sur l'écoulement autour du profil NACA à $Re = 1400$). Pour que l'approche faiblement-compressible offre des temps de restitution équivalents à la formulation incompressible, l'optimisation des algorithmes communs doit conduire à une accélération d'un facteur réel bien supérieur, compris entre 9 et 70. En annexe C nous présentons la réflexion qui nous a conduit à l'estimation de ce facteur réel d'accélération. Deux configurations sont étudiées : une accélération des algorithmes faiblement-compressible à maillage constant et une accélération à erreur constante.

Même si à l'heure actuelle le code WCCH pourrait incontestablement bénéficier d'optimisation, le facteur d'accélération à atteindre reste conséquent. A titre d'exemple, un post-doctorant a travaillé récemment au LHEEA sur le portage GPU du code (en Cuda). Les gains obtenus sur le cas test de la convection de tourbillon se situent aux alentours de 20 à 30 pour des blocs de 4×4 cellules et sont donc du même ordre de grandeur (Fig. 4.35). Cependant il s'agit du speed-up sur un cas test périodique bien différent de l'écoulement

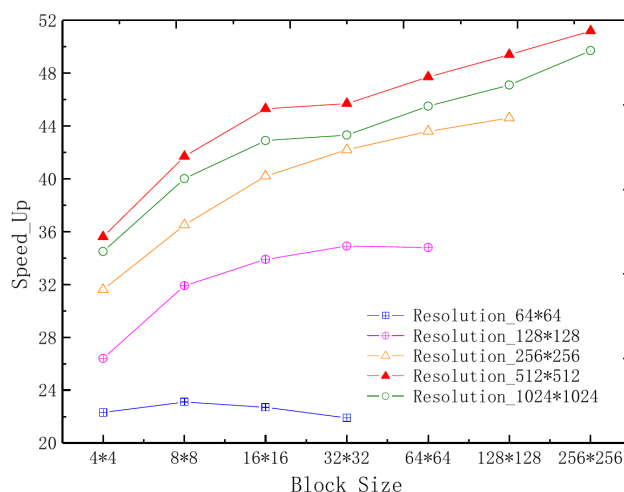


FIGURE 4.35 – Accélération du solveur faiblement-compressible de WCCH par portage du code sur GPU.

ment autour d'un profil NACA. Et porter tout le code WCCH sur GPU n'est pas forcément une perspective

envisagée par le LHEEA. Le développement et la maintenance de modèles numériques dans un code GPU restent encore, malheureusement, l'affaire de spécialistes affûtés.

Par conséquent toute accélération du code faiblement-compressible bénéficierait en partie au code incompressible et ne fait qu'accroître le facteur d'accélération à atteindre pour que la formulation faiblement-compressible surpasse la formulation incompressible. Même si les conclusions sur les gains à précision équivalente obtenus sur les cas test de Taylor-Couette ou du NACA sont liées à une implémentation particulière (choix des structures de données, de calcul des flux, etc.), il semblerait que le facteur d'accélération nécessaire à l'approche faiblement-compressible soit supérieur à 10. De plus ce facteur d'accélération requis ne tient pas compte d'une éventuelle optimisation de la résolution de l'équation de Poisson en incompressible qui accroîtrait son bénéfice sur la formulation faiblement-compressible.

4.4 Conclusion

Dans ce chapitre, nous avons vérifié et validé l'implémentation de la formulation incompressible.

Le solveur a dans un premier temps été validé sur des grilles cartésiennes uniformes puis sur des grilles raffinées localement. Nous avons pu mettre en évidence l'ordre élevé de convergence spatiale de la vitesse et de la pression pour des écoulements avec des conditions aux limites périodiques. En particulier, nous avons montré qu'une pression d'ordre élevé peut être reconstruite à l'ordre 4 à partir des pressions issues des sous-étapes du schéma Runge-Kutta 4. En présence de conditions aux limites de Dirichlet instationnaires sur la vitesse, nous avons mis en évidence la chute de l'ordre de convergence sur la pression comme cela est attendu pour le schéma de projection de Chorin. Nous avons également comparé deux techniques distinctes pour le traitement des interfaces de raffinement : l'approche guard-cells et une approche originale d'étirement (stretching) du maillage. Pour des écoulements visqueux à bas Reynolds les deux techniques donnent des résultats strictement identiques. Cependant la méthode guard-cells donne de meilleurs résultats à mesure que le nombre de Reynolds augmente, signe que le traitement de la partie convective est plus précis dans ce cas. Notre approche de stretching présente aussi l'intérêt de pouvoir reconstruire des inconnues et calculer des gradients manquants "au vol", lors du calcul du flux alors que la méthode guard-cells nécessite d'être appelée au début de chaque avance en temps. Ces deux approches ont été développées dans des codes distincts il n'est donc pas possible de comparer directement leur efficacité algorithmique. L'approche de type stretching que nous avons présentée a aussi l'avantage d'être hautement compatible avec le schéma WENO5 et un nombre de cellules par bloc et par direction pris égal à 2.

Dans un deuxième temps nous nous sommes attachés à comparer en détail les approches incompressible et faiblement-compressible. Nous avons confirmé la faible diffusion de l'incompressible. La résolution du problème de Riemann aux faces des cellules explique le taux de dissipation de la formulation faiblement-compressible. Le cas test de la convection de vortex a mis en évidence que l'ordre de convergence est élevé pour les deux formulations. L'ordre est plus marqué en faiblement-compressible en particulier sur les maillages les plus fins alors qu'une forme de saturation apparaît en incompressible. En présence de frontières immergées, le comportement des deux formulations est très différent à bas nombre de Reynolds comme l'a montré l'écoulement de Taylor-Couette. Notamment la pression à l'intérieur du corps joue un rôle différent et sans doute pénalisant en faiblement-compressible. Pour des nombres de Reynolds plus élevés, les résultats des deux approches sont plus équivalents. A maillage identique, il semble difficile de les départager. Mais à précision équivalente l'approche incompressible est 5 à 20 fois plus rapide. Enfin la scalabilité du code en formulation incompressible est excellente et presque aussi parfaite que celle du code en formulation faiblement-compressible même si cette dernière semble autoriser plus facilement des nœuds peu chargés.

En définitive, le choix initial d'un schéma faiblement-compressible plutôt qu'un schéma plus conven-

tionnel incompressible pour résoudre des écoulements à bas Mach était motivé par l'absence de système linéaire à résoudre et la scalabilité élevée de son algorithme. La vitesse du son limite cependant la taille des pas de temps et allonge les temps de restitution. Nous avons choisi d'implémenter une nouvelle formulation purement incompressible basée sur le schéma de projection de Chorin qui reste en accord avec les exigences du code WCCH : explicite en temps (pour la vitesse), compatible avec des schémas d'ordre élevé en espace et en temps, etc.

Finalement le système linéaire à résoudre ne pose que peu de difficultés puisqu'il s'agit d'un Laplacien à coefficient constant dont la résolution est bien maîtrisée. Il ne semble pas trop affecter la scalabilité globale du code. La formulation incompressible offre donc des performances HPC comparables à la formulation faiblement-compressible avec une diffusion numérique inférieure et une meilleure compatibilité avec la méthode de frontière immergée. Dans l'optique de disposer d'un code de calcul à vocation industrielle, le choix de l'approche incompressible est sans doute plus judicieux que celle faiblement-compressible.

Enfin même si nous avons cherché à être le plus exhaustif possible dans la comparaison des codes, il manque un certain nombre de considérations pour compléter le tableau de comparaison comme la prise en compte d'une surface libre, de corps mobiles ou de modèles de turbulence.

Chapitre 5

Analyse de stabilité et amélioration de la méthode de frontière immergée

Ce chapitre est consacré à l'analyse détaillée de la méthode de frontière immergée utilisée dans le code WCCH. Il vise à établir un socle théorique et bibliographique afin de placer la méthode dans le paysage des méthodes de frontière immergée et ainsi mieux cerner ses avantages et ses limitations. Nous nous intéressons notamment à la convergence de la méthode et à sa stabilité.

Dans un deuxième temps, nous proposons une piste d'amélioration de la méthode et nous étudions la faisabilité d'une montée en ordre. Enfin, cette approche est mise en application sur des écoulements laminaires stationnaires et instationnaires, en formulations faiblement-compressible et incompressible.

5.1	Introduction	145
5.2	Revue des méthodes de frontière immergée	146
5.2.1	Continuous forcing	146
5.2.2	Discrete forcing	150
5.2.3	Calcul des efforts	160
5.2.4	Frontière immergée et corps mobiles	161
5.3	Position de la méthode initialement implémentée dans WCCH	162
5.4	Consistance, stabilité et convergence de la méthode	164
5.4.1	Simplification du cas d'étude	164
5.4.2	Position du problème	164
5.4.3	Précision de la prise en compte des conditions aux limites	165
5.4.4	Validations et expériences numériques	169
5.5	Amélioration et montée en ordre de la méthode de frontière immergée	171
5.5.1	Réduction de la sensibilité au pas de temps	171
5.5.2	Adaptation de la méthode pour les schémas Runge-Kutta	173
5.5.3	Vers une montée en ordre de la méthode	175
5.5.4	Validations numériques de la montée en ordre	179
5.6	Conclusions	187

5.1 Introduction

Comme évoqué dans les chapitres précédents, le principal attrait d'une méthode de frontière immergée est qu'il n'est pas nécessaire de faire coïncider les cellules du maillage volumique fluide avec la frontière des

corps solides. A l'origine, le concept de frontière immergée a été introduit par Peskin [Peskin, 1972] pour simuler l'écoulement autour d'une valve aortique. L'intérêt majeur était de pouvoir simuler une interaction fluide/structure avec de grandes déformations d'une membrane élastique. Peu à peu, la communauté des chercheurs en CFD a vu dans ces méthodes la possibilité de calculer des écoulements autour de corps rigides sans avoir à générer des maillages adaptés. Aujourd'hui, les techniques de frontière immergée sont un sujet actif de recherche, comme en témoigne l'explosion du nombre de publications sur le sujet ces dix dernières années. Malgré tout, leur application industrielle reste encore minoritaire ; preuve que la construction d'un schéma de frontière immergée pour des écoulements multiphysiques turbulents reste un immense défi à l'heure actuelle.

5.2 Revue des méthodes de frontière immergée

La catégorisation des méthodes de frontière immergée ne fait pas toujours consensus dans la littérature tant les méthodes foisonnent et sont variées. Leur point commun est de modifier localement l'écoulement au voisinage du corps afin qu'il reconnaisse la présence du solide. Une classification souvent reprise est celle de Mittal et Iaccarino [Mittal et Iaccarino, 2005]. Elle distingue les méthodes qui prennent en compte la présence du corps à l'échelle continue des équations de Navier-Stokes, de celles qui interviennent après la discrétisation des équations en modifiant les schémas numériques pour imposer des conditions de bord. La première classe de méthodes est qualifiée de forçage continu (Continuous forcing). Le forçage prend la forme d'un terme source pour tenir compte de la condition de bord. A l'inverse, le deuxième type de méthode est appelé forçage discret (Discrete forcing) et repose sur une modification du schéma numérique discret.

Dans ce qui suit, nous allons décrire certaines des méthodes des deux classes. Nous ne retiendrons que les méthodes les plus répandues. Nous nous limiterons également à celles dont le champ d'application est proche des écoulements d'hydrodynamique marine que nous visons ou celles qui ressemblent à la méthode déjà utilisée dans le code WCCH. Cela permettra dans la section suivante de mieux inscrire la méthode du code WCCH dans le vaste spectre des méthodes de frontière immergée. La revue sera donc loin d'être exhaustive et on pourra se référer par exemple à la thèse de master de Bandringa [Bandringa, 2010] pour un inventaire plus complet.

5.2.1 Continuous forcing

Méthode de Peskin

Peskin a été le premier à introduire le concept de frontière immergée [Peskin, 1972]. La méthode originale s'applique à une frontière solide déformable. L'originalité est de proposer un couplage entre une résolution eulérienne du fluide et une description lagrangienne du solide. La frontière solide est suivie par des points lagrangiens qui se déplacent avec la vitesse locale du fluide :

$$\frac{\partial \mathbf{X}}{\partial t}(s, t) = \mathbf{u}(\mathbf{X}(s, t)) \quad (5.1)$$

avec \mathbf{X} la position du marqueur lagrangien et s son abscisse curviligne. La force $\mathbf{f}(\mathbf{x}, t)$ appliquée par la frontière sur le fluide est définie par convolution avec le noyau Dirac δ :

$$\mathbf{f}(\mathbf{x}, t) = \int_{\Gamma_b} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds \quad (5.2)$$

où $\mathbf{F}(s, t)$ est la contrainte exercée localement par un point du solide sur le fluide. Cette contrainte peut par exemple être déterminée à partir d'une loi de comportement élastique comme celle de Hooke :

$$\mathbf{F}(s, t) = \mathbf{S}(\mathbf{X}(s, t), t) \quad (5.3)$$

Comme les marqueurs lagrangiens ne suivent généralement pas exactement le maillage fluide eulérien, les forces \mathbf{f} sont réparties sur plusieurs cellules au voisinage de la frontière solide avec une fonction Dirac régularisée comme par exemple [Roma et al., 1999] :

$$\delta_h = \begin{cases} \frac{1}{6} \left(5 - 3|r/h| - \sqrt{-3(1 - |r/h|)^2 + 1} \right) & \text{si } 0.5 \leq |r/h| \leq 1.5 \\ \frac{1}{3} \left(1 + \sqrt{-3(r/h)^2 + 1} \right) & \text{si } |r/h| \leq 0.5 \\ 0 & \text{sinon} \end{cases} \quad (5.4)$$

On trouve de nombreuses fonctions régularisées dans la littérature, chacune obéissant à des propriétés bien précises. La fonction δ_h est à support compact et dépend du pas d'espace h du maillage pour limiter la diffusion de la force aux cellules proches de la frontière solide.

En pratique l'algorithme de résolution pour un pas de temps est le suivant :

1. Disposant des positions et des vitesses des marqueurs lagrangiens à la frontière du solide, il est possible de calculer et de diffuser le terme source (5.2) aux cellules du fluide qui se situent au voisinage de la membrane solide.
2. Les équations de Navier-Stokes sont ensuite résolues en espace et avancées en temps sans modification particulière, sauf l'ajout du terme source $\mathbf{f}(\mathbf{x}, t)$.
3. Les vitesses eulériennes sont ensuite interpolées au niveau des marqueurs lagrangiens.
4. Les marqueurs lagrangiens sont finalement advectés suivant les valeurs de ces vitesses interpolées. Le processus est répété à la prochaine avance en temps.

La particularité de cette méthode est qu'elle s'applique uniquement à des frontières élastiques.

Méthode pour des frontières rigides

La première tentative d'extension de la méthode originale de Peskin à des frontières rigides est attribuée à Goldstein et al. [Goldstein et al., 1993]. Ils ont suggéré de modéliser des corps rigides avec l'expression suivante de $\mathbf{F}(s, t)$:

$$\mathbf{F}(s, t) = \alpha \int_0^t \mathbf{u}(s, \tau) d\tau + \beta \mathbf{u}(s, t) \quad (5.5)$$

Cette formule peut être vue comme un régulateur PI (Proportionnel Intégral). L'intérêt de cette approche qui s'apparente à un asservissement est de prendre en compte la mémoire de la grandeur asservie et d'avoir une erreur statique nulle. Les coefficients α et β prennent des valeurs négatives très élevées pour avoir l'effet escompté. Néanmoins cette méthode souffre de problèmes de raideur et de stabilité, sources de phénomènes oscillatoires.

Méthode d'interface immergée de LeVeque

Depuis son introduction en 1972, la méthode de Peskin a trouvé un vaste champ d'application pour la simulation d'écoulements biologiques. Malgré son efficacité, cette méthode demeure d'ordre 1 en espace. Une des raisons expliquant ce faible ordre de convergence est liée au terme source qui peut prendre des valeurs très importantes lorsque la cellule fluide est située à proximité d'un marqueur lagrangien. Il en résulte que la vitesse prédite avec ce terme source prend des valeurs non physiques. En formulation incompressible, la divergence de cette vitesse est ensuite utilisée pour calculer le second membre de l'équation de Poisson conduisant à une brusque variation de pression à la traversée de la membrane. Lee et LeVeque ont suggéré une amélioration de la précision de la méthode de Peskin en gérant différemment la force singulière [Lee et LeVeque, 2003]. La discontinuité attendue en pression est prise en compte directement dans la résolution de

l'équation de Poisson. La méthode qu'ils ont proposée est issue de l'étude de LeVeque et Li sur une méthode d'interface immergée (*IIM* pour Immersed Interface Method) pour des équations elliptiques à coefficients discontinus [LeVeque et Li, 1994].

Pour les équations de Navier-Stokes incompressibles, la solution satisfait les conditions de saut :

$$\begin{cases} [p](s, t) = \mathbf{F}_n(s, t) \cdot \mathbf{n}(s, t) \\ \left[\frac{\partial p}{\partial n} \right](s, t) = \frac{\partial}{\partial s} \mathbf{F}_\tau(s, t) \cdot \mathbf{n}(s, t) \\ [\mathbf{u}(s, t)] = 0 \\ \mu \left[\frac{\mathbf{u}}{n} \right](s, t) = -(\mathbf{F}_\tau(s, t) \cdot \boldsymbol{\tau}(s, t)) \boldsymbol{\tau}(s, t) \end{cases} \quad (5.6)$$

avec $\mathbf{n}(s, t)$ le vecteur normal à l'interface et $\boldsymbol{\tau}(s, t)$ le vecteur tangentiel. $\mathbf{F}_n(s, t)$ et $\mathbf{F}_\tau(s, t)$ sont respectivement la composante normale et la composante tangentielle de la densité de force. La composante tangentielle $\mathbf{F}_\tau(s, t)$ de la densité de force est utilisée dans le calcul d'avance en temps des vitesses fluides et est donc diffusée à toutes les cellules du voisinage de la frontière solide. La composante normale $\mathbf{F}_n(s, t)$ est, quant à elle, ajoutée au second membre de l'équation de Poisson. Néanmoins comme la composante tangentielle reste diffusée par l'opérateur δ_h , la frontière reste lissée. Xu et Wang ont poussé plus loin la modification de la méthode en intégrant les conditions de saut directement dans la discrétisation des équations et se passent ainsi de l'étape de régularisation qui diffuse l'interface [Xu et Wang, 2006]. La méthode *IIM* donne de bons résultats et améliore significativement l'approche originelle de Peskin.

Méthode de projection de Taira et Colonius

Taira et Colonius ont présenté une méthode originale au formalisme mathématique élégant, et dont l'approche est semblable au schéma de projection en formulation incompressible [Taira et Colonius, 2007]. Le terme source défini sur les marqueurs lagrangiens (5.2) est vu comme un multiplicateur de Lagrange qui permet de satisfaire la condition d'adhérence à la paroi tout comme la pression peut être perçue en incompressible comme un multiplicateur de Lagrange qui assure la condition d'incompressibilité. Le système d'équations à résoudre est alors :

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{\nabla p}{\rho} + \frac{1}{\rho} \nabla \cdot \bar{\boldsymbol{\tau}} + \int_{\Gamma_b} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds \\ \mathbf{u}(\mathbf{X}(s, t)) = \int_{\Omega} \mathbf{u}(\mathbf{x}) \delta(\mathbf{x} - \mathbf{X}(s, t)) d\mathbf{x} = \mathbf{u}_{\Gamma_b}(\mathbf{X}(s, t)) \end{cases} \quad (5.7)$$

Comme dans la méthode de Peskin, la convolution avec l'opérateur Dirac δ permet d'échanger des informations entre le domaine fluide et la frontière solide. La discrétisation de ce système d'équations peut s'écrire sous forme matricielle :

$$\begin{pmatrix} A & G & -H \\ D & 0 & 0 \\ E & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ p^{n+1} \\ f \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \\ u_{\Gamma_b}^{n+1} \end{pmatrix} + \text{conditions limites} \quad (5.8)$$

E est l'opérateur qui permet d'interpoler la vitesse sur la frontière solide à partir des vitesses fluides. $-H$ est l'opérateur de diffusion (spreading) du terme source du marqueur lagrangien vers les termes sources fluides. Ces opérateurs sont symétriques et le système linéaire est résolu par une décomposition LU. Enfin, la méthode est appliquée à des frontières avec cinématique imposée. Nous pouvons aussi émettre quelques incertitudes sur le conditionnement de la matrice du système linéaire lorsque les marqueurs lagrangiens sont très proches des points du maillage fluide.

Approche DLM

L'approche DLM (Distributed Lagrange-Multiplier/fictitious domain) qualifie les méthodes d'immersion de géométrie qui consiste à résoudre les équations de Navier-Stokes sur tout le domaine tout en considérant le fluide dans le corps comme un ensemble homogène rigide. La méthode, proposée par Glowinski et al. [Glowinski et al., 2001] repose sur la notion de multiplicateur de Lagrange dans un contexte de formulation variationnelle pour imposer une contrainte de tenseur de vitesse nul à l'intérieur du corps. En effet, l'équation $\bar{\bar{\tau}} = 0$ est rigoureusement équivalente à imposer un mouvement de solide rigide comme cela est montré dans [Vincent et al., 2014]. Dans [Hachem et al., 2012], les auteurs reprennent ce principe et appliquent leur méthode, baptisée IVM (Immersed Volume Method), à des écoulements multiphasiques autour de corps rigides. Le système d'équations à résoudre est alors le suivant :

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{\nabla p}{\rho} + \frac{1}{\rho} \nabla \cdot \bar{\bar{\tau}} + \nabla \cdot \chi_s \bar{\bar{\lambda}} \\ \chi_s \bar{\bar{\tau}} = \mathbf{0} \end{cases} \quad (5.9)$$

où $\bar{\bar{\lambda}}$ est le multiplicateur de Lagrange et χ_s la fonction indicatrice du solide régularisée au voisinage de la frontière :

$$\chi_s = \begin{cases} 1 & \text{si } \alpha > \epsilon \\ \frac{1}{2} \left(1 + \frac{\alpha}{\epsilon} + \frac{1}{\pi} \sin \left(\frac{\pi \alpha}{\epsilon} \right) \right) & \text{si } |\alpha| \leq \epsilon \\ 0 & \text{si } \alpha < -\epsilon \end{cases} \quad (5.10)$$

α étant la distance signée au corps et ϵ une distance paramétrique qui définit la largeur de l'interface. Les approches monolithiques se prêtent bien aux formulations variationnelles et à la discrétisation par la méthode des éléments finis.

Méthodes de pénalisation

Enfin les méthodes de pénalisation telles que celle proposée par Arquis et Caltagirone [Arquis et Caltagirone, 1984] puis par Angot [Angot, 1989] consistent à considérer le solide comme un milieu poreux. Le modèle de Arquis et Caltagirone est un modèle de porosité de type Brinkman dont le terme source prend la forme :

$$\mathbf{f}(\mathbf{x}, t) = \frac{1}{\eta} \chi_s \mathbf{u}(\mathbf{x}, t) \quad (5.11)$$

$\frac{1}{\eta}$ est la perméabilité du milieu qui est nulle dans le fluide et tend vers l'infini dans le solide pour imposer une vitesse nulle. χ_s est la fonction indicatrice du solide qui vaut 1 dans le solide et 0 dans le fluide. Angot et al. montrent que cette méthode de pénalisation, qualifiée de L^2 , converge vers la solution des équations de Navier-Stokes avec condition de Dirichlet à la frontière du solide lorsque que η tend vers 0. L'ordre de convergence est en $O(\eta^{3/4})$ [Angot et al., 1999]. Ils proposent par ailleurs une autre méthode de pénalisation, H^1 , dont la convergence théorique est en $O(\eta)$. Cette approche similaire pénalise la vitesse mais également la dérivée de vitesse, le gradient de pression et le tenseur visqueux. La réécriture des équations de Navier-Stokes donne :

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \left(1 + \frac{1}{\eta} \chi_s \right) \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = \frac{1}{\rho} \nabla \cdot \left(1 + \frac{1}{\eta} \chi_s (-p \bar{\bar{\mathbf{I}}} + \bar{\bar{\tau}}) \right) - \frac{1}{\eta} \chi_s \mathbf{u} \end{cases} \quad (5.12)$$

Des expériences numériques montrent que les deux méthodes, L^2 et H^1 , sont d'ordre 1 en espace et que la pénalisation H^1 ne donne pas forcément de meilleurs résultats. Cela s'explique par la pénalisation du

terme visqueux qui créé une couche limite artificielle. Les méthodes de pénalisation ont été étendues pour des écoulements compressibles [Liu et Vasilyev, 2007]. L'équation de continuité est traitée en pénalisant le flux de masse pour éviter la propagation d'ondes acoustiques parasites à l'intérieur du solide :

$$\frac{\partial \rho}{\partial t} + \left(1 + \left(\frac{1}{\phi} - 1\right)\right) \nabla \cdot (\rho \mathbf{u}) = 0 \quad (5.13)$$

avec ϕ la porosité du milieu. Les méthodes de pénalisation sont aussi compatibles avec les écoulements multiphasiques autour d'obstacles comme cela est présenté dans [Doradoux, 2017].

Enfin Vincent et al. ont proposé une méthode d'ordre 2 appliquée à des écoulements multiphasiques autour de particules solides [Vincent et al., 2014]. L'originalité de leur approche est d'appliquer la pénalisation sur une décomposition spécifique du tenseur visqueux faisant apparaître les effets physiques de compressibilité, de cisaillement, d'élongation et de rotation. Ainsi, la contrainte de solide rigide est imposée de façon plus forte et plus précise.

En somme, les approches de forçage continu, à l'exception de l'approche monolithique, reposent sur l'utilisation d'une force singulière conduisant à une représentation diffuse de l'interface. L'efficacité des méthodes semblables à celle de Peskin est étroitement liée à une contrainte sur les pas d'espace des maillages eulériens et lagrangiens qui doivent être sensiblement équivalents. Elles ont trouvé de nombreuses applications en ingénierie biomédicale où les nombres de Reynolds restent modérés et où les déformations de membrane sont bien représentées. Elles sont par ailleurs très employées dans la simulation d'écoulements avec un grand nombre de particules solides ou pour des problèmes de sédimentation. Pour envisager des écoulements à Reynolds plus élevé, il apparaît nécessaire de prendre en compte la position de l'interface de façon plus précise pour mieux capter la couche limite. Plus le nombre de Reynolds est élevé, plus la couche limite est concentrée au voisinage du corps et nécessite une précision accrue. La précision locale revêt alors d'une importance cruciale. Les approches par forçage discret permettent de répondre à ces attentes. A ce titre, la méthode de Lee et Leveque ainsi que celle Xu et Wang, mentionnées dans cette partie sur les approches continues, peuvent aussi être considérées comme des méthodes rattachées au forçage discret puisqu'elles incluent une représentation plus exacte de la frontière solide.

5.2.2 Discrete forcing

Par opposition aux méthodes de type "continuous forcing" énumérées précédemment, les méthodes discrètes modifient localement la discrétisation des opérateurs au voisinage de l'interface afin de prendre en compte la condition de bord.

Rappelons les équations de Navier-Stokes en formulation incompressible sur le domaine Ω :

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} + \mathbf{f} \\ \mathbf{u} = \mathbf{u}_\Gamma \text{ sur } \partial\Omega \end{cases} \quad (5.14)$$

Dans l'approche continue, le terme de forçage \mathbf{f} était déterminé à partir d'une loi de comportement de la frontière solide comme pour la méthode de Peskin. Mohd-Yusof [Mohd-Yusof, 1997] a proposé l'introduction d'un terme source qui est directement fixé par la solution numérique. Prenons un schéma d'avance en temps Euler explicite d'ordre 1. L'équation discrète pour un volume de contrôle i du maillage s'écrit :

$$\frac{\mathbf{u}_i^{n+1} - \mathbf{u}_i^n}{\Delta t} = \mathbf{RHS}_i^n + \mathbf{f}_i^n \quad (5.15)$$

où RHS_i^n regroupe le calcul des flux convectifs et visqueux ainsi que le gradient de pression. Pour avoir $\mathbf{u} = \mathbf{u}_\Gamma$ sur $\partial\Omega$ au pas de temps t^{n+1} , le terme de forçage doit s'écrire simplement :

$$\begin{cases} \mathbf{f}_i^n = -RHS_i^n + \frac{\mathbf{u}_\Gamma^{n+1} - \mathbf{u}_i^n}{\Delta t} \text{ sur } \partial\Omega \\ 0 \text{ ailleurs} \end{cases} \quad (5.16)$$

Classiquement deux classes de forçage discret sont distinguées :

- les méthodes explicites ("indirect boundary condition imposition" dans la classification de Mittal et al. [Mittal et Iaccarino, 2005]) où la condition de bord est prise en compte directement via le terme de forçage. Celui-ci ne s'applique qu'à un nombre restreint de cellules au voisinage de la frontière.
- les méthodes implicites ("direct boundary condition imposition"). Il ne s'agit plus de déterminer explicitement un terme de forçage mais de faire évoluer les champs de vitesse et de pression directement à partir des conditions aux limites. Les méthodes ghost-cell ou cut-cell tombent dans cette catégorie.

Méthodes "discrete forcing" explicites

La première méthode de ce type que nous considérons est celle de Fadlun et al. [Fadlun et al., 2000]. Celle-ci s'inspire directement de [Mohd-Yusof, 1997] que nous ne détaillons pas dans cette partie. En effet, la méthode de Mohd-Yusof ressemble plus, à certains égards, à la méthode ghost-cell qui est détaillée par la suite. Néanmoins Fadlun et al. reprennent le calcul du terme source (5.16) de Mohd-Yusof. Ils mentionnent que ce type de forçage est direct dans la mesure où la valeur souhaitée de la vitesse est imposée explicitement sur la frontière sans processus dynamique. Ainsi à chaque pas de temps, la condition est satisfaite quelles que soient les variations temporelles de l'écoulement. Un autre avantage est qu'elle n'introduit pas de constante à ajuster suivant le type d'écoulement comme les constantes de raideur des méthodes de forçage continu. De plus, le coût CPU supplémentaire pour le calcul de \mathbf{f}_i^n est nul puisque les termes RHS_i^n ont déjà été calculés par ailleurs. Enfin, cette méthode préserve la stabilité du schéma car les opérateurs numériques restent inchangés.

La principale objection à cette méthode est qu'elle n'est valable que si les nœuds de la frontière du solide coïncident avec les points de discrétisation du maillage fluide. Fadlun envisage donc trois possibilités (Fig. 5.1). La première, et la plus simple, consiste à sélectionner les points de forçage à proximité de la frontière et de faire comme si ces points coïncidaient avec la frontière solide. Tout se passe alors comme si la frontière était décrite en forme d'escalier. La géométrie n'est donc pas lisse et pas fidèlement respectée. Une deuxième option envisageable revient à calculer le volume V_b occupé par le solide dans une cellule de volume V_{cell} . V_b/V_{cell} est la fraction de volume solide et vaut alors 1 si une cellule est complètement située à l'intérieur du solide, 0 si elle est totalement fluide, et une valeur comprise entre 0 et 1 si la frontière solide l'intersecte. Le terme source est ensuite pondéré par cette fraction volumique. La dernière possibilité est de sélectionner les cellules qui intersectent le solide et dont l'emplacement de l'inconnue de vitesse se situe dans le fluide ; puis d'interpoler en ce point une vitesse à partir de la vitesse d'une cellule fluide voisine \mathbf{u}_{i+1}^{n+1} et de la vitesse de la frontière \mathbf{u}_Γ^{n+1} (Fig. 5.1).

Fadlun compare ensuite ces trois approches sur un écoulement axisymétrique de formation de tourbillons en réalisant une étude de convergence en maillage. Sans surprise, l'erreur diminue quand la frontière est prise en compte avec une complexité croissante. La première approche n'est même pas d'ordre 1 (pour la norme 2 de l'erreur en vitesse) contrairement à l'approche avec pondération par la fraction volumique solide. Enfin la dernière approche avec interpolation offre la meilleure précision avec un ordre proche de 2 (Fig. 5.2).

Avant de poursuivre sur les méthodes d'interpolation (qui donnent donc une meilleure précision), arrêtons-nous sur la deuxième technique de Fadlun. Celle-ci est relativement simple à mettre en œuvre et est prisée

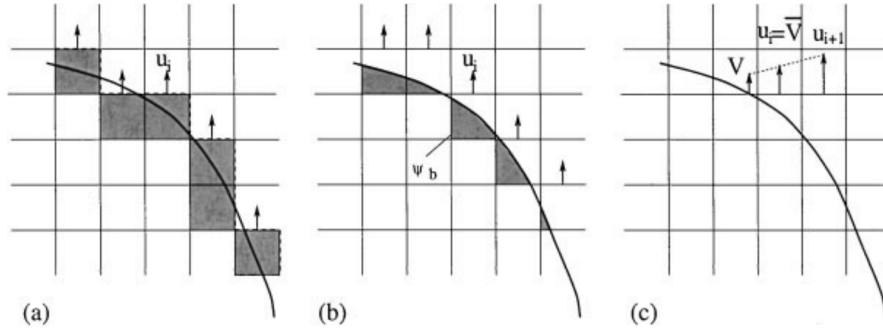


FIGURE 5.1 – Procédures d’interpolation possibles tirées de [Fadlun et al., 2000] : (a) pas d’interpolation, (b) pondération par la fraction de volume, (c) interpolation

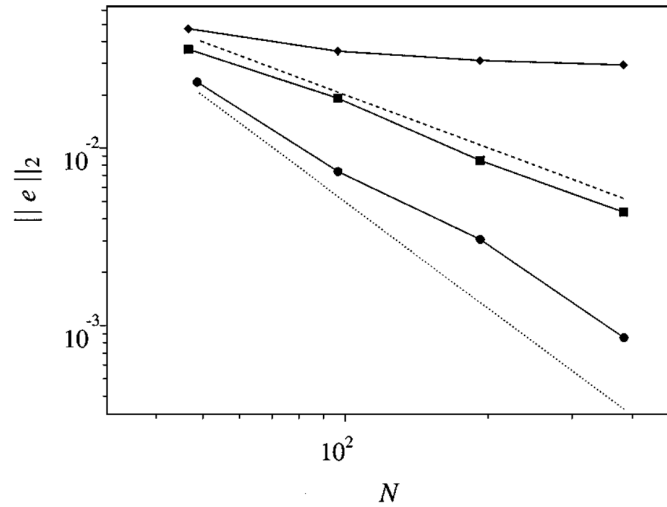


FIGURE 5.2 – Erreur de convergence sur la vitesse en fonction du nombre de points de calcul [Fadlun et al., 2000] : (◆) pas d’interpolation, (■) pondération par la fraction de volume, (●) interpolation

pour les écoulements impliquant des particules solides [Kajishima et Takiguchi, 2002, Bigot et al., 2014]. Elle permet en effet de traiter un écoulement avec un grand nombre de particules pour des applications de sédimentation par exemple. On la retrouve aussi dans la littérature sous le terme de Volume of Body ou Volume of Solid [Pan, 2006] par analogie avec la méthode Volume of Fluid pour des écoulements diphasiques. La principale différence entre toutes ces méthodes similaires tient dans la façon de traiter le couplage fluide-structure et la dynamique des particules (avec prise en compte des collisions par exemple).

Si nous reprenons l’expression des équations de Navier-Stokes avec le terme source de forçage pondéré par la fraction volumique :

$$\frac{\mathbf{u}_i^{n+1} - \mathbf{u}_i^n}{\Delta t} = \mathbf{RHS}_i^n + \frac{V_i^b}{V_i^{cell}} \mathbf{f}_i^n \quad (5.17)$$

Cela donne avec l’expression de \mathbf{f}_i^n :

$$\frac{\mathbf{u}_i^{n+1} - \mathbf{u}_i^n}{\Delta t} = \left(1 - \frac{V_i^b}{V_i^{cell}}\right) \mathbf{RHS}_i^n + \frac{V_i^b}{V_i^{cell}} \frac{\mathbf{u}_i^{body} - \mathbf{u}_i^n}{\Delta t} \quad (5.18)$$

Que l’on peut aussi mettre sous la forme

$$\mathbf{u}_i^{n+1} = (1 - \varphi_i^b)(\mathbf{u}_i^n + \Delta t \mathbf{RHS}_i^n) + \varphi_i^b \mathbf{u}_i^{body} \quad (5.19)$$

avec $\varphi_i^b = V_i^b/V_i^{cell}$. La vitesse calculée à l'instant $n+1$ apparaît donc comme une pondération d'une vitesse fluide et d'une vitesse de solide rigide. La fraction volumique solide φ_i^b assure la transition et le couplage entre la partie fluide et la partie solide. Pour évaluer cette fraction volumique, il faut être en mesure de calculer pour une cellule le volume occupé par le solide ou le fluide. Pour cela, il est possible d'utiliser par exemple une fonction de distance signée à la paroi (level-set). La figure 5.3 représente une cellule de maillage intersectée par une frontière solide. La fraction de volume solide est un polygone dont les côtés peuvent être aisément estimés avec :

$$\Delta s = \frac{\Delta y}{1 + \frac{|\phi_2|}{|\phi_1|}} \quad (5.20)$$

grâce au théorème de Thalès.

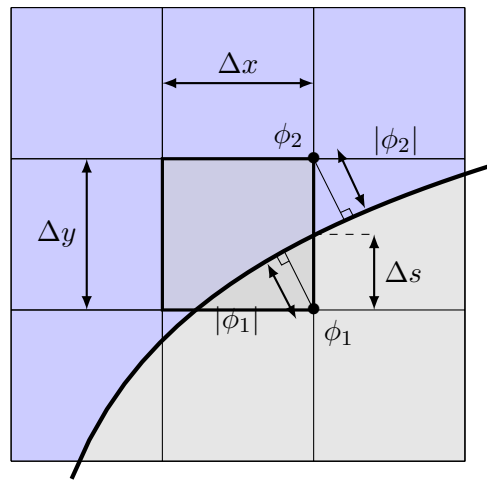


FIGURE 5.3 – Identification de l'intersection d'une cellule d'un maillage cartésien avec une frontière solide.

Si l'identification des polygones résultant de l'intersection avec la frontière solide reste relativement simple en 2D, en 3D le problème est d'une complexité bien supérieure.

Une alternative à cette démarche est l'utilisation de la formule approximative suivante, suggérée dans [Kempe et Fröhlich, 2012] :

$$\varphi_b = \frac{\sum_{m=1}^N \phi_m H(\phi_m)}{|\phi_m|} \quad (5.21)$$

avec $N = 4$ en 2D pour les quatre coins d'un volume de contrôle et H la fonction indicatrice du corps qui vaut 1 si $\phi \leq 0$ et 0 sinon.

Sans aller jusqu'à estimer finement la fraction volumique de solide, beaucoup d'auteurs se limitent à l'utilisation d'une fonction de type Heaviside régularisée qui lisse la transition solide/liquide sur quelques cellules et a le même effet escompté. Par exemple :

$$\varphi_b = \begin{cases} 1 & \text{si } \alpha > \epsilon \\ \frac{1}{2} \left(1 + \frac{\alpha}{\epsilon} + \frac{1}{\pi} \sin \left(\frac{\pi \alpha}{\epsilon} \right) \right) & \text{si } |\alpha| \leq \epsilon \\ 0 & \text{si } \alpha < -\epsilon \end{cases} \quad (5.22)$$

Quoi qu'il en soit, le type d'application visée par le recours à ce type de méthode semble surtout restreint aux écoulements avec particules solides en suspension. Nous relèverons cependant un exemple d'application navale de la méthode pour la détermination de résistance ajoutée sur houle de navires [Yang et al., 2015]. Dans cet article, les auteurs simulent le comportement dynamique d'un navire dans la houle sur grille

cartésienne étirée avec une méthode de capture de surface libre de type THINC (Tangent of Hyperbola for INterface Capturing).

Pour revenir à la méthode d'interpolation de Fadlun, son intérêt se justifie donc par sa précision d'ordre plus élevé. Dans cette approche, le choix de la direction d'interpolation est arbitraire suivant une des directions cartésiennes de l'espace. Par la suite, Balaras a suggéré une meilleure reconstruction effectuée dans la direction normale au corps [Balaras, 2004] (Fig. 5.4).

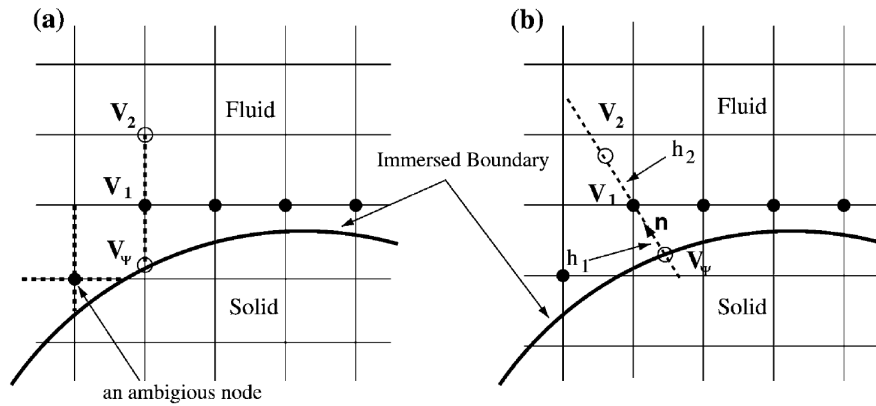


FIGURE 5.4 – Comparaison des schémas d'interpolation des vitesses au bord d'un solide par la méthode de Fadlun (a) et de Balaras (b) (extrait de [Balaras, 2004])

Balaras identifie d'abord les points de maillage fluide en contact avec l'interface solide (point 1 par exemple sur la figure 5.4(b)). Ces points sont, comme pour Fadlun, les points de forçage. Il définit ensuite des droites normales au solide passant par ces points (ligne en pointillés sur la figure). Cette droite intersecte la frontière solide (en Ψ par exemple). Un point fictif (point 2) est créé sur cette droite en s'éloignant de la frontière solide. En ce point, une vitesse est calculée à l'aide d'un schéma d'interpolation s'appuyant sur les quatre points voisins. Connaissant donc la vitesse en ce point fictif, il est ensuite possible d'interpoler une vitesse au point de forçage, avec par exemple l'expression :

$$V_1 = \frac{h_2}{h} V_\Psi + \frac{h_1}{h} V_2 \quad (5.23)$$

Avec $h = h_1 + h_2$. Par défaut, le point fictif est construit par symétrie et $h_1 = h_2$. Mais il peut être nécessaire d'aller chercher un point fictif plus loin, notamment si parmi les quatre points entourant le point fictif certains sont eux-mêmes des points de forçage. Dans ce cas $h_1 \neq h_2$. On note que h est nécessairement strictement positif et ne s'annule jamais. Ainsi même si le point de forçage est confondu avec la paroi solide, $h \neq 0$. Ce qui évite à la vitesse prédite V_1 de prendre des valeurs non-admissibles dans de tels cas. De nombreuses méthodes proposées dans la littérature sont directement issues de l'approche de Balaras. Ce qui les distingue est l'ordre d'interpolation et la sélection des stencils. Récemment, cette méthode a été appliquée à des problèmes de complexité industrielle comme l'écoulement turbulent 3D autour d'un sous-marin en auto-propulsion [Posa et Balaras, 2018]. La simulation décrite dans l'article mêle frontière immergée et modèle de turbulence de type LES avec loi de paroi. Elle s'appuie en particulier sur l'extension de la méthode originale effectuée par Yang et Balaras pour traiter des frontières mobiles [Yang et Balaras, 2006]. Enfin Yang et Stern ont intégré cette méthode dans une approche non-itérative pour résoudre des problèmes de couplage fluide-structure [Yang et Stern, 2015].

Nous résumons ci-après le schéma d'avance en temps de ces méthodes de calcul explicites du terme de forçage pour des géométries fixes et dans le cadre d'un schéma incompressible de projection de Chorin :

1. Calculer une vitesse prédite \mathbf{u}_i^* en omettant le terme source de forçage.

$$\mathbf{u}_i^* = \mathbf{u}_i^n + \Delta t R H S_i^n \quad (5.24)$$

2. Identifier les points de forçage à la frontière du domaine fluide (avec une fonction level-set par exemple). Cette étape peut être réalisée une fois pour toutes en début de simulation dans le cas de géométries fixes.
3. Calculer une vitesse interpolée $\mathbf{u}_i^{\text{interp}}$ en ces points de forçage (à partir des valeurs de \mathbf{u}^*). En déduire le terme source \mathbf{f}_i^n pour ces points :

$$\mathbf{f}_i^n = -R H S_i^n + \frac{\mathbf{u}_i^{\text{interp}} - \mathbf{u}_i^n}{\Delta t} \quad (5.25)$$

4. Mettre à jour \mathbf{u}_i^* . Cette étape ne concerne que les cellules forcées :

$$\mathbf{u}_i^* = \mathbf{u}_i^* + \Delta t \mathbf{f}_i^n \quad (5.26)$$

5. Résoudre l'équation de Poisson.
6. Ajouter le gradient de pression à la vitesse prédite pour la rendre à divergence nulle.

Cet algorithme soulève la question du rôle de la pression au point de forçage. Balaras précise que la condition de bord sur la pression est implicitement prise en compte dans le second membre de l'équation de Poisson puisque ce dernier dépend de $\mathbf{u}_i^{\text{interp}}$ et donc de la vitesse à la paroi pour laquelle la condition de bord en pression est respectée. Enfin, sur le fait que la vitesse est forcée sur \mathbf{u}_i^* et non sur \mathbf{u}_i^{n+1} , Balaras explique que la précision du schéma temporel ne s'en trouve pas affectée car $\mathbf{u}_i^{n+1} = \mathbf{u}_i^* + O(\Delta t^2)$ (cf. [Kim et al., 2001]).

Par ailleurs, nous remarquerons que sans traitement particulier, la vitesse à l'instant t^{n+1} ne respecte donc pas simultanément la condition de divergence nulle et de non-glissement à la paroi. Pour résoudre ce problème Kim et al. ont proposé un modèle de sources/puits dans le traitement de l'équation de pression pour respecter les deux conditions à la fois [Kim et al., 2001].

Pour finir, un dernier point sur ces méthodes explicites concerne le traitement de l'intérieur du corps. Fadlun et al. [Fadlun et al., 2000] évoquent trois façons possibles de traiter l'intérieur du solide pour des géométries fixes. La première est d'appliquer un terme source à tous les points intérieurs sans lissage particulier. Le champ de vitesse est donc imposé mais la pression est laissée libre et s'ajuste pour imposer la conservation de la masse. Une deuxième option est de laisser l'intérieur du corps libre de développer un écoulement fluide. Le troisième choix possible est d'inverser la vitesse pour les premières cellules solides et d'imposer ainsi la condition de bord à la manière des cellules fantômes. Les différents tests qu'ils ont conduits sur ces trois options ont montré qu'aucune ne se distinguait particulièrement d'une autre. Le choix de traitement de l'intérieur du corps peut donc être arbitraire. Pour des corps mobiles en revanche, la façon de prendre en compte l'intérieur du corps a un effet important, en particulier quand une cellule initialement solide devient fluide. Un traitement spécifique est alors nécessaire comme celui proposé dans [Yang et Balaras, 2006].

Méthodes "discrete forcing" implicites

Le propre des méthodes de forçage discret implicites est de faire évoluer les champs de vitesse et de pression en prenant en compte la condition de bord en amont de l'avance en temps. Cette approche est donc plus dynamique que la méthode explicite précédente qui impose un champ de vitesse *a posteriori*.

Ghost-cell La méthode ghost-cell telle qu'elle a été proposée par Tseng et al. [Tseng et Ferziger, 2003] s'inspire aussi de Mohd-Yusof [Mohd-Yusof, 1997]. Ce dernier a proposé l'introduction d'un terme source qui est fixé par la différence entre une vitesse interpolée et la vitesse à imposer. Pour que le terme de forçage soit régulier à la traversée de la frontière, il est défini sur la frontière mais aussi en quelques points intérieurs au solide. La vitesse tangentielle du premier point à l'intérieur du solide est l'opposée de la vitesse du premier point fluide selon le principe du miroir. Cette technique permet de minimiser l'erreur sur l'estimation du terme diffusif (Fig. 5.5).

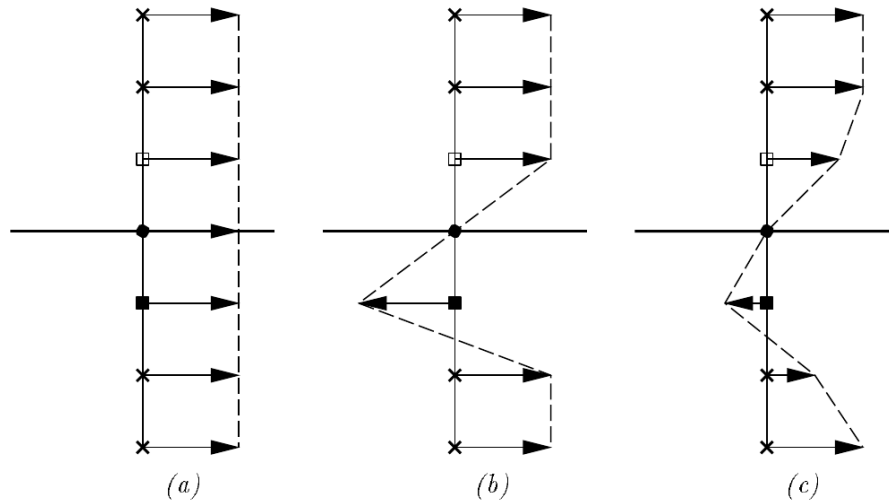


FIGURE 5.5 – Représentation du champ de vitesse initial (a), du champ de vitesse imposé après forçage (b) et du champ de vitesse après avance en temps (c) (extrait de [Mohd-Yusof, 1997]). Le point (■) représente le point où la vitesse est inversée, (□) est le point fluide et (●) le point frontière sur lequel la vitesse est imposée.

Plus généralement, le terme "cellule fantôme" ou "ghost-cell" désigne une cellule fictive à l'intérieur du corps qui est en contact avec au moins une cellule fluide et à laquelle on affecte des valeurs de vitesse et de pression de façon à imposer la condition de bord. Le schéma numérique n'a donc plus à être modifié au voisinage de la frontière. Les cellules fantômes étendent de manière naturelle le domaine de calcul à la frontière. Elles supposent donc qu'une solution fluide continue existe à la traversée de la frontière. La procédure numérique d'un pas de temps de la méthode ghost-cell est la suivante :

1. Détecter la frontière et identifier les cellules fantômes.
2. Extrapoler pour trouver les valeurs de vitesse et de pression des cellules fantômes et s'assurer que les conditions de bord sont satisfaites.
3. Résoudre les équations de Navier-Stokes avec un schéma de projection de Chorin par exemple.

Il existe de nombreuses méthodes pour extrapoler les champs du fluide aux cellules fantômes. Une première approche retenue par Tseng et al. [Tseng et Ferziger, 2003] est de construire un triangle qui a pour sommet le nœud de la cellule fantôme et les deux cellules fluides les plus proches. Ce choix minimise la probabilité d'instabilités numériques. Sur la figure 5.6, G est le point fantôme et X_1 et X_2 sont les points du fluide les plus proches. O est le point à la paroi sur lequel la condition de bord doit être imposée. La formule d'interpolation bilinéaire en 2D est :

$$\phi = a_0 + a_1x + a_2y \quad (5.27)$$

avec ϕ une variable locale. La valeur de ϕ sur la cellule fantôme est fonction des valeurs aux points X_1 , X_2 et O :

$$\phi_G = \omega_1\phi_1 + \omega_2\phi_2 + \omega_0\phi_0 \quad (5.28)$$

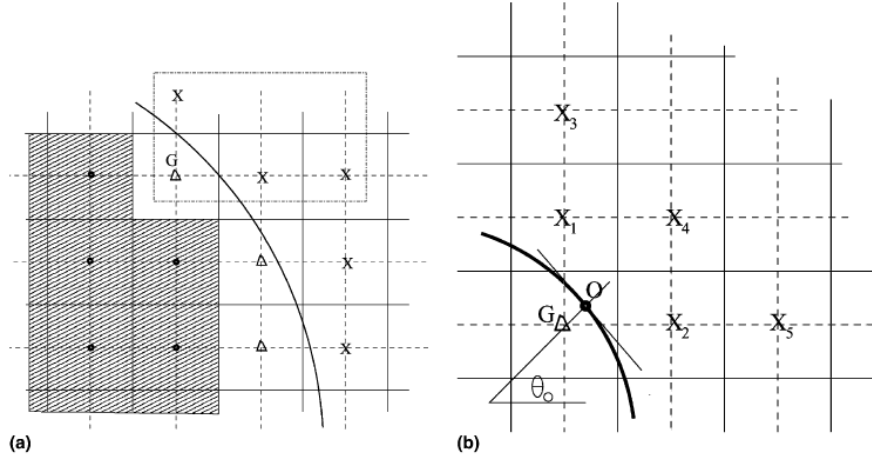


FIGURE 5.6 – (a) Représentation du domaine de calcul avec les points de calcul fluide (\times) et les cellules fantômes (Δ). (b) Représentation des points d'appui utilisés pour l'extrapolation (extrait de [Tseng et Ferziger, 2003])

Les coefficients de pondération ω s'obtiennent à partir des valeurs aux nœuds :

$$[\omega_1, \omega_2, \omega_0]^T = B^{-1}[x_G, y_G, 1]^T \quad (5.29)$$

avec B une matrice de Vandermonde :

$$B = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_0 & y_0 \end{pmatrix} \quad (5.30)$$

L'inconvénient majeur de ce type d'extrapolation est que les coefficients de pondération ω peuvent prendre des valeurs négatives très grandes si O est proche de l'un des points du fluide. Bien que cela soit correct et attendu d'un point de vue algébrique, des instabilités numériques peuvent apparaître. Tseng et al. proposent alors deux parades. La première est d'utiliser le point I symétrique de G par rapport à O (Fig. 5.7). ϕ_I est déterminée à partir de la procédure d'interpolation décrite ci-dessus et $\phi_G = 2\phi_O - \phi_I$. L'autre solution est de modifier localement la frontière solide en la déplaçant jusqu'au point situé très près de l'interface. Par ailleurs Tseng et al. donnent aussi une procédure pour traiter une condition homogène de Neumann comme celle sur la pression.

Mittal et al. ont appliqué cette méthode en 3D tout en améliorant les opérateurs d'interpolation normaux pour déterminer les valeurs des cellules fantômes [Mittal et al., 2008]. Berthelsen et Faltinsen ont adapté la méthode pour prendre en compte des géométries irrégulières avec des opérateurs d'ordre élevé et une approche multidirectionnelle [Berthelsen et Faltinsen, 2008].

Des méthodes ghost-cell ont également été développées en formulation compressible [Ghias et al., 2007, Gorsse et al., 2012]. Le traitement des frontières en présence de chocs représente une difficulté supplémentaire par rapport à la formulation incompressible.

Méthode de type "cut-cell" Aucune des méthodes présentées jusqu'ici ne permet de garantir la conservation de la masse et de la quantité de mouvement au voisinage du solide. Les méthodes de type cut-cell répondent à cette exigence. La méthode proposée par Clarke et al. était pionnière dans le domaine et s'intéressait à des écoulements non visqueux [Clarke et al., 1986]. Globalement, ces méthodes consistent à inscrire la géométrie du corps directement dans le maillage cartésien. En supposant que l'intersection de la frontière dans une cellule est un segment en 2D, les cellules intersectées par la géométrie sont découpées en

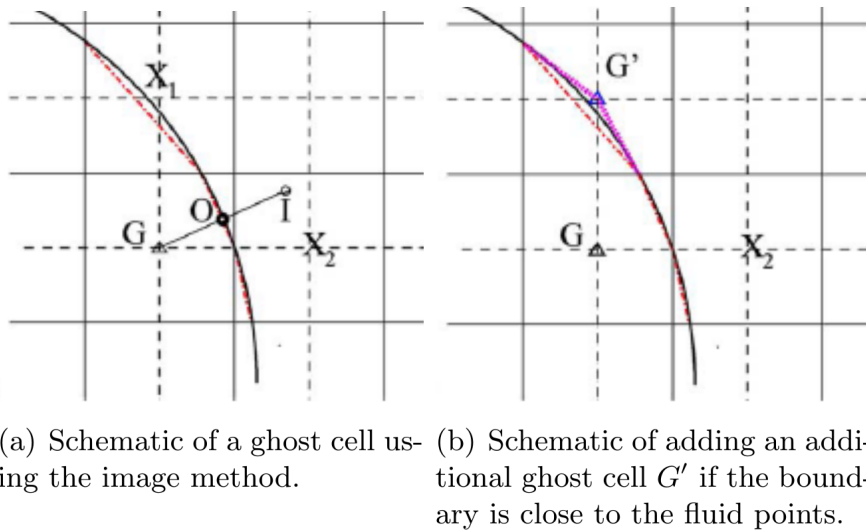


FIGURE 5.7 – Deux solutions pour traiter les points du fluide proches de la frontière solide (extrait de [Tseng et Ferziger, 2003])

cellules polygonales de différentes natures (triangles, trapèzes, pentagones... en 2D) (Fig. 5.8). La méthode cut-cell est indissociable du formalisme des volumes finis. Des flux sont ensuite calculés sur chaque segment d'intersection. Les méthodes cut-cell peuvent être complexes à mettre en œuvre, notamment parce que les configurations de découpe possibles sont nombreuses, chacune nécessitant un traitement particulier. De plus, l'intersection peut faire apparaître des cellules de faible volume qui pénalisent l'avance en temps par imposition de pas de temps éventuellement très petits à travers la condition CFL. La méthode employée par Ye et al. pour pallier cette contrainte est de fusionner les cellules intersectées dont le centre se situe dans le solide avec leurs cellules voisines [Ye et al., 1999].

Le code open-source Gerris [Popinet, 2003b] déjà mentionné auparavant et le code commercial CONVERGE CFD [Science, 1997] sont des exemples de code qui implémentent la méthode cut-cell. Gerris prend en compte les corps avec une méthode de type cut-cell dont les opérations géométriques de découpe ont été implémentées dans la librairie tierce open-source GTS [Popinet, 2000]. Cependant, dans sa version actuelle et disponible à tous, le code ne peut gérer les corps mobiles. Un premier travail de thèse en 2009 [Delaux, 2009] a permis d'introduire des corps à mouvement imposé et un second en 2015 [Shui, 2015] des corps dynamiques.

La méthode de Clarke et al. supposait une disposition des variables colocalisées au centre de la cellule. Sur maillage MAC, il faut veiller à redéfinir une vitesse sur le nouveau segment résultant de l'intersection de la cellule avec la géométrie. Cheny et Botella ont proposé récemment la méthode LS-STAG (pour Level-Set Staggered) [Cheny et Botella, 2010]. Ils utilisent une fonction distance pour définir l'interface fluide-solide. L'idée n'est plus de calculer un flux sur le segment d'intersection mais de déplacer la vitesse qui se situe au milieu d'une face intersectée au milieu de la partie fluide de la face. Les opérateurs doivent à leur tour être modifiés (Fig. 5.9).

Cette méthode a été depuis activement reprise par les chercheurs de l'Université de Groningen pour une implémentation dans le code de calcul d'hydrodynamique marine ComFLOW et étendue partiellement en 3D [Bandringa, 2010, H.J.L. van der Heiden et Luppés, 2013]. Enfin si la méthode cut-cell est complexe à implémenter en "collocated" et coûteuse en temps CPU, elle l'est davantage en "staggered".

En conclusion les méthodes de forçage discret proposent une prise en compte plus ou moins exacte de

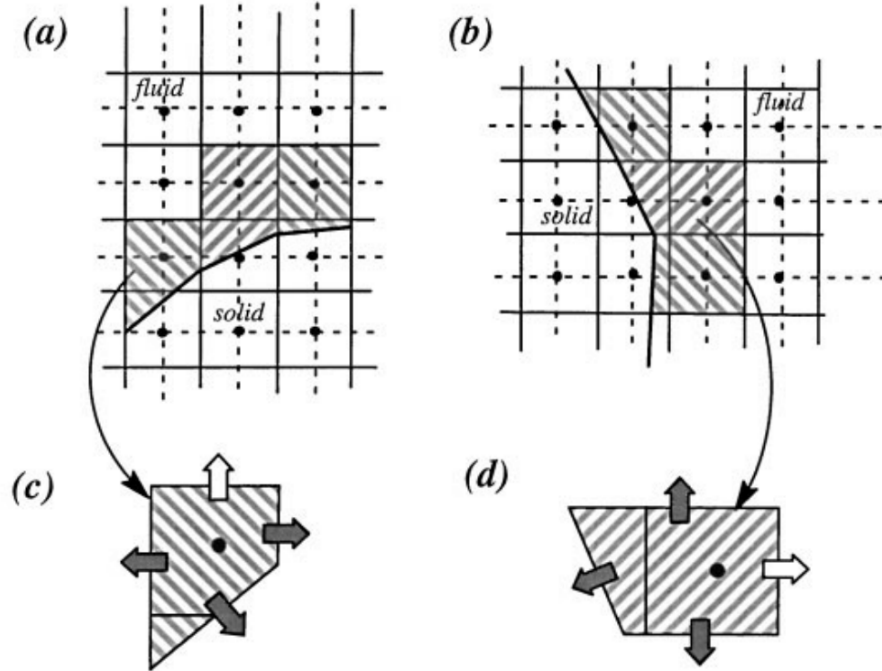


FIGURE 5.8 – Exemple de redécoupage géométrique de cellules intersectées par la frontière (extrait de [Ye et al., 1999])

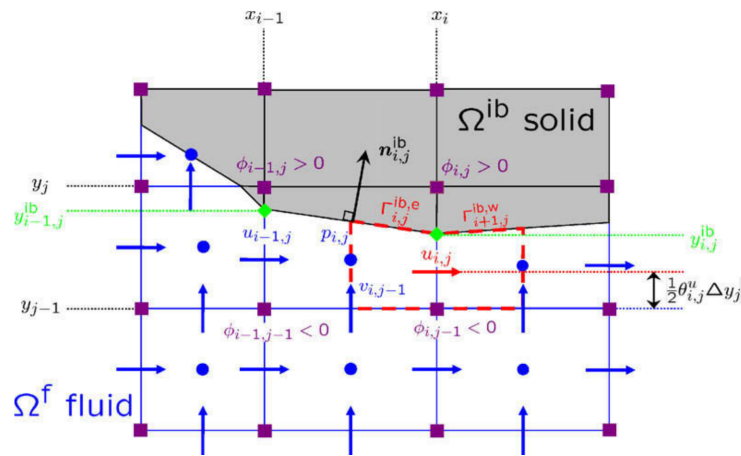


FIGURE 5.9 – Méthode LS-STAG (extrait de [Cheny et Botella, 2010])

la frontière solide et rendent possible des simulations à haut Reynolds. Ce gain en précision et en stabilité par rapport au forçage continu s'accompagne d'une complexité algorithmique bien supérieure, qui peut rapidement devenir délicate comme pour la méthode cut-cell.

5.2.3 Calcul des efforts

D'une façon générale, les efforts exercés par le fluide sur le corps, notés \mathbf{F} se calculent en intégrant le tenseur des contraintes locales sur la surface du corps :

$$\mathbf{F} = \iint_S (-p\bar{\mathbf{I}} + \bar{\boldsymbol{\tau}}) \cdot \mathbf{n} dS = F_p + F_v \quad (5.31)$$

avec \mathbf{n} le vecteur normal à la surface du corps, qui par convention est orienté vers le milieu fluide. L'effort se décompose en une contribution de pression et une partie visqueuse. Pour les approches "body-fitted" où le maillage est adapté au corps, c'est cette technique de calcul qui est retenue.

En frontière immergée, la non-coïncidence entre maillage volumique fluide et maillage surfacique solide ne permet pas d'intégrer directement le tenseur des contraintes sur le corps.

Pour les méthodes de forçage continu, la détermination des efforts reste simple puisqu'il suffit d'intégrer les termes source de forçage calculés au niveau des marqueurs lagrangiens. Pour les méthodes à forçage discret autres que celles de type cut-cell, les forces et moments appliqués au corps peuvent être calculés à l'aide des équations de conservation de la quantité de mouvement et du moment cinétique [Balaras, 2004, Berthelsen et Faltinsen, 2008]. Prenons par exemple une surface Σ_∞ qui englobe un volume de fluide V autour du corps solide de surface S (Fig. 5.10).

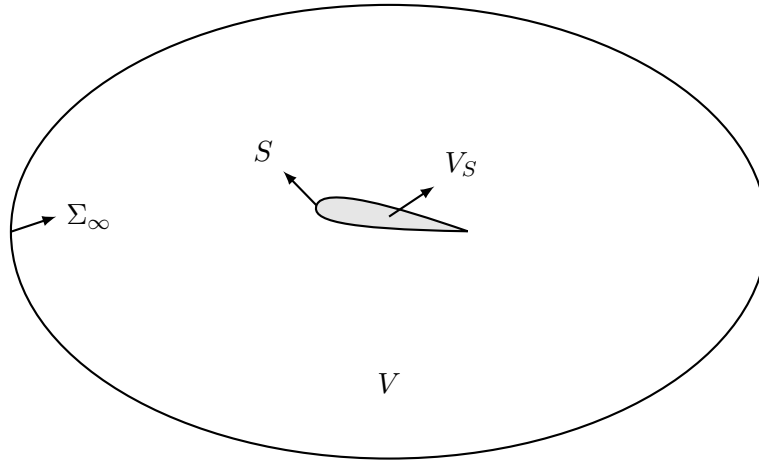


FIGURE 5.10 – Calcul des efforts appliqués à un corps par application de la conservation de la quantité de mouvement

L'intégration de l'équation de conservation de la quantité de mouvement sur le volume V et l'application du théorème de Green-Ostrogradski donnent :

$$\mathbf{F} = \frac{d}{dt} \iiint_V \rho \mathbf{u} dV + \iint_{\Sigma_\infty} ((\rho \mathbf{u} \otimes \mathbf{u}) + p\bar{\mathbf{I}} - \bar{\boldsymbol{\tau}}) \cdot \mathbf{n} dS \quad (5.32)$$

puisque la condition de non-pénétration du fluide donne :

$$\iint_S (\rho \mathbf{u} \otimes \mathbf{u}) \cdot \mathbf{n} dS = 0 \quad (5.33)$$

Une autre façon de procéder est de considérer le volume intérieur V_S du solide comme un fluide. L'intégration des équations de Navier-Stokes continues :

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot (\bar{\boldsymbol{\tau}}) + \rho \mathbf{f} \quad (5.34)$$

sur le volume V_S et l'application du théorème de Green-Ostrogradski donnent :

$$\mathbf{F} = - \iiint_{V_S} \rho \mathbf{f} dV + \iiint_{V_S} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) \right) dV \quad (5.35)$$

où \mathbf{f} est une représentation du terme source discret. \mathbf{F} est issue de (5.31). Cette formule peut s'écrire plus simplement :

$$\mathbf{F} = - \iiint_{V_S} \rho \mathbf{f} dV + \frac{d}{dt} \iiint_{V_S} \rho \mathbf{u} dV \quad (5.36)$$

La conservation de la masse du solide rigide implique que :

$$\frac{d}{dt} \iiint_{V_S} \rho \mathbf{u} dV = 0 \quad (5.37)$$

et \mathbf{F} se résume ainsi à :

$$\mathbf{F} = - \iiint_{V_S} \rho \mathbf{f} dV \quad (5.38)$$

Ainsi dans beaucoup d'applications de méthodes de frontière immergée, les forces \mathbf{F} et moments \mathbf{M}_G sont calculés en sommant les termes sources discrets internes au solide, comme par exemple dans [Bigot et al., 2014]. Le torseur des efforts extérieurs s'exprime par :

$$\left\{ \begin{array}{l} \mathbf{F} = - \sum_{i \in V_S} \rho_i \mathbf{f}_i = \sum_{i \in V_S} \rho_i \left(- \frac{\mathbf{u}_{solide} - \mathbf{u}_i}{\Delta t} + (-\mathbf{C}_i + \mathbf{D}_i - \mathbf{G} \mathbf{p}_i) \right) \\ \mathbf{M}_G = - \sum_{i \in V_S} \rho_i (\mathbf{x}_i \wedge \mathbf{f}_i) \end{array} \right. \quad (5.39)$$

avec \mathbf{C}_i la discrétisation du terme convectif, \mathbf{D}_i celui du terme visqueux et $\mathbf{G} \mathbf{p}_i$ le terme de pression. G désigne le centre de gravité du corps et \mathbf{x}_i est le vecteur distance par rapport à G . Belliard et al. ont par ailleurs montré, dans leur étude de frontière immergée appliquée à des corps mobiles, que le terme $-\frac{\mathbf{u}_{solide} - \mathbf{u}_i}{\Delta t} - \mathbf{C}_i$ est négligeable par rapport aux autres contributions [Belliard et al., 2016].

Enfin Takiguchi et al. ont une approche mixte qui consiste à déterminer d'abord la contribution de pression en sommant les gradients de pression volumique interne au solide pour minimiser l'erreur de transfert de quantité de mouvement entre la phase liquide et la phase solide [Takiguchi et al., 1999]. La contribution visqueuse est ensuite calculée en évaluant le tenseur visqueux sur le maillage surfacique du corps.

5.2.4 Frontière immergée et corps mobiles

La bonne gestion des corps mobiles dans les méthodes de frontière immergée est un point crucial et reste un véritable défi à l'heure actuelle. En effet, l'un des intérêts majeurs des méthodes de frontière immergée est de ne pas avoir à gérer un maillage adapté au corps sur lequel l'écoulement est simulé. Et plus précisément, elles permettent de se passer des étapes de déformation du maillage adapté avec des approches ALE (Arbitrary Lagrangian–Eulerian) par exemple. Elles semblent donc parfaitement adaptées aux interactions fluide-structure avec grandes déformations.

Néanmoins un problème récurrent pour les méthodes de frontière immergée à forçage discret est celui de la génération d'oscillations d'efforts parasites (SFO dans la littérature, pour Spurious Force Oscillations) lorsque les corps sont mis en mouvement. Le phénomène se produit aussi bien pour des corps à cinématique imposée que libres de mouvement. Il est lié à l'apparition de cellules initialement fluides qui se retrouvent solides (dead cells) et inversement de cellules solides qui deviennent fluides (fresh cells). Lee et al. montrent que c'est plus précisément les dead cells qui sont principalement responsables des oscillations par la discontinuité temporelle de leur valeur de vitesse et de pression [Lee et al., 2011]. Il a aussi été mis évidence

que l'amplitude des oscillations décroît à mesure que le pas d'espace Δx décroît, mais croît lorsque le pas de temps Δt décroît suivant une loi en $O(\frac{\Delta x^\alpha}{\Delta t})$ [Lee et al., 2011, Seo et Mittal, 2011]. Belliard et al. ont formellement établi que pour une méthode d'ordre 2 avec un forçage direct, l'amplitude des oscillations est en $O(\frac{\Delta x^2}{\Delta t}) + O(\Delta x)$ [Belliard et al., 2016].

La méthode des sources/puits de Kim et al. permet de réduire en partie l'ampleur du phénomène des SFO en traitant la discontinuité de pression [Kim et al., 2001]. Pour les méthodes de forçage basées sur la fraction de volume ou l'utilisation de la fonction Heaviside, de nombreux auteurs ont proposé des formulations de fonctions régularisées pour limiter les oscillations [Uhlmann, 2005, Yuki et al., 2007]. En particulier Belliard et al. proposent une approche hybride intéressante mêlant interpolation d'ordre 2 et lissage par une fonction Heaviside [Belliard et al., 2016]. Elle permet de ramener l'amplitude des oscillations en $O(\Delta t) + O(\Delta x)$, c'est-à-dire consistante en espace et en temps.

Une autre problématique concerne les corps libres en mouvement dont la cinématique n'est pas imposée mais dont la dynamique est régie par les lois de la mécanique de Newton. La difficulté est alors d'ajuster vitesse et déplacement du solide rigide pour les faire correspondre aux efforts hydrodynamiques. Un couplage faible, explicite est souvent adopté en première approche et consiste à déterminer les efforts hydrodynamiques puis déplacer le corps rigide selon la 2^{ème} loi de Newton. Un couplage fort, implicite, est cependant recommandé dans certaines situations où le couplage faible est instable. Pour plus de détails sur cette problématique, le lecteur pourra se référer à [Bigot et al., 2014, Takiguchi et al., 1999] pour des exemples de couplage faible ou à [Yang et Stern, 2012, Yang et Stern, 2015] pour des exemples de couplage fort.

Bilan de l'état de l'art

Cette revue partielle des méthodes de frontière immergée a permis de mettre en lumière le large éventail des méthodes de frontière immergée. Pour chaque grande méthode numérique, de nombreuses déclinaisons ont été mises en œuvre pour répondre à des exigences de précision et de stabilité. Nous retiendrons que pour la simulation autour de corps rigides à des nombres de Reynolds élevés, les méthodes de forçage discret avec schéma d'interpolation d'ordre élevé pour faire apparaître de façon exacte l'interface sont de plus en plus prisées.

5.3 Position de la méthode initialement implémentée dans WCCH

Au cours de sa thèse, P. Bigay a testé trois méthodes de forçage discret pour un solveur volumes finis en formulation faiblement-compressible [Bigay, 2015] : une méthode cut-cell, une approche ghost-cell et une méthode de forçage direct explicite. Les résultats issus des deux premières méthodes n'étaient pas à la hauteur des attentes fixées. La méthode cut-cell n'a été implémentée qu'en 2D mais elle a permis d'appréhender toute la complexité algorithmique de l'intersection des maillages volumiques par des géométries surfaciques. La méthode ghost-cell n'a pas été retenue du fait de son manque de précision et des coûts (certes moindres que ceux de la méthode cut-cell mais non négligables) à entrevoir pour une utilisation 3D. C'est finalement la méthode de forçage direct qui a été choisie pour sa simplicité algorithmique et la possibilité de traiter des frontières mobiles.

Dans le paragraphe suivant, nous tentons de positionner cette méthode par rapport aux méthodes existant dans la littérature. Il vise à compléter la présentation succincte de la méthode dans [Bigay et al., 2017, Bigay, 2015].

Nous rappelons d'abord la méthode telle qu'elle est décrite dans [Bigay et al., 2017] (et précédemment exposée Sec. 1.2.4) : l'accélération d'une cellule i est décomposée en une accélération fluide et une accélération solide,

$$\left(\frac{d\mathbf{u}_i}{dt}\right)^n = (1 - H_i) \left(\frac{d\mathbf{u}_i}{dt}\right)_{fluide}^n + H_i \left(\frac{d\mathbf{u}_i}{dt}\right)_{solide}^n \quad (5.40)$$

H_i étant donc la fonction Heaviside régularisée (Fig. 1.6). En discrétisant les dérivées temporelles avec un schéma Euler explicite d'ordre 1 :

$$\left(\frac{d\mathbf{u}_i}{dt}\right)_{\text{fluide}}^n \approx \frac{\mathbf{u}_{i,\text{fluide}}^{n+1} - \mathbf{u}_i^n}{\Delta t} \quad (5.41)$$

et

$$\left(\frac{d\mathbf{u}_i}{dt}\right)_{\text{solide}}^n \approx \frac{\mathbf{u}_{i,\text{solide}}^{n+1} - \mathbf{u}_i^n}{\Delta t} \quad (5.42)$$

Nous obtenons la formulation en vitesse :

$$\mathbf{u}_i^{n+1} = (1 - H_i)\mathbf{u}_{i,\text{fluide}}^{n+1} + H_i\mathbf{u}_{i,\text{solide}}^{n+1} \quad (5.43)$$

avec :

$$\mathbf{u}_{i,\text{fluide}}^{n+1} = \mathbf{u}_i^n + \Delta t \mathbf{RHS}_i^n \quad (5.44)$$

Cette formulation en vitesse nous indique que la méthode initiale dans le code WCCH est clairement à rattacher aux méthodes de forçage direct explicite avec pondération du terme source par fraction de volume solide (et autres équivalents) comme proposées dans [Takiguchi et al., 1999, Fadlun et al., 2000, Yuki et al., 2007, Bigot et al., 2014, Belliard et al., 2016] par exemple. Yuki et al. [Yuki et al., 2007] proposent par exemple une fonction permettant d'estimer la fraction de volume α et qui ressemble à (1.27) :

$$\begin{aligned} \alpha &= \frac{1}{2} \left(1 - \tanh \left(\frac{\text{dist}}{\sigma \lambda \Delta} \right) \right) \\ \lambda &= |n_x| + |n_y| + |n_z| \\ \sigma &= 0.05(1 - \lambda^2) + 0.3 \end{aligned} \quad (5.45)$$

avec $\mathbf{n} = (n_x, n_y, n_z)$ le vecteur unitaire normal à la frontière et Δ un élément caractéristique de la taille du maillage qui vaut $\sqrt{2}\Delta x$ sur grille uniforme. En pratique la valeur de σ peut être augmentée pour lisser la valeur de α sur 1 à 3 cellules comme dans [Bigot et al., 2014] et éviter l'apparition de pics d'oscillation parasites lors de simulations de corps mobiles. L'expression de la fraction volumique (5.45) est à mettre en lien avec le choix de la fonction Heaviside lissée (1.27) qui a été choisie dans le code WCCH. Néanmoins, l'originalité de la méthode employée dans WCCH est la résolution fluide effectuée en formulation faiblement-compressible alors que la majorité des auteurs décrivant ce type de méthode ont recours à des approches incompressibles.

Par ailleurs, la formulation en accélération (5.40) semble plus générique au premier abord que la formulation en vitesse (5.43). Cependant, elle ne présente pas d'intérêt pratique. Par exemple supposons que nous souhaitions simuler l'écoulement autour d'un corps se déplaçant à vitesse constante. L'accélération a_i^n de ce corps est nulle. La discrétisation de (5.40) en intégrant cette condition donne :

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + (1 - H_i)\Delta t \mathbf{RHS}_i^n + H_i \Delta t a_i^n = \mathbf{u}_i^n + (1 - H_i)\Delta t \mathbf{RHS}_i^n \quad (5.46)$$

Si une cellule est initialement dans le fluide puis se retrouve dans le corps alors $\mathbf{u}_i^{n+1} = \mathbf{u}_i^n$ puisque $H_i \equiv 1$ dans le corps. La vitesse imposée ne sera alors pas égale à celle du corps en mouvement. C'est donc bien sous l'angle de la décomposition de la vitesse en une vitesse fluide et une vitesse solide qu'il faut envisager la méthode (cf. Eq. (5.43)).

5.4 Consistance, stabilité et convergence de la méthode

5.4.1 Simplification du cas d'étude

Dans cette partie nous souhaitons étudier quelques propriétés de la méthode de frontière immergée. Afin de mieux cerner la convergence et la stabilité de la méthode, nous l'étudions sur un cas simplifié d'une équation linéaire d'advection-diffusion monodimensionnelle.

Une équation linéaire d'advection-diffusion 1D d'une grandeur ϕ prend la forme suivante :

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} = \nu \frac{\partial^2 \phi}{\partial x^2} \quad (5.47)$$

Cette équation modélise par exemple la dispersion d'un polluant transporté par un fluide se déplaçant à la vitesse constante u et où ν est le coefficient de diffusion. Physiquement, elle exprime que l'advection est contrebalancée par la diffusion dans la direction de l'écoulement. Dans ce sens elle s'approche des équations de Navier-Stokes même si pour ces dernières, le gradient de pression et la non-linéarité du terme convectif ont une importance non négligeable et bien plus complexe dans l'équilibre des différents phénomènes. En passant des équations de Navier-Stokes à cette équation linéaire, l'hypothèse simplificatrice est forte. Néanmoins, l'équation linéaire d'advection-diffusion présente l'avantage d'être beaucoup plus simple à appréhender et de disposer d'une solution analytique. L'extrapolation des conclusions de l'étude sur cette équation au système complet des équations de Navier-Stokes nécessite des vérifications heuristiques.

Si les conditions aux limites sont stationnaires, ϕ est solution de l'équation :

$$u \frac{\partial \phi}{\partial x} = \nu \frac{\partial^2 \phi}{\partial x^2} \quad (5.48)$$

Avec les conditions aux limites de Dirichlet, $\phi = \phi_L$ en $x = 0$ et $\phi = \phi_R$ en $x = L$, la solution de (5.48) est alors :

$$\phi = \phi_L + \frac{e^{Pe \frac{x}{L}} - 1}{e^{Pe} - 1} (\phi_R - \phi_L) \quad (5.49)$$

où Pe définit le nombre de Peclet :

$$Pe = \frac{uL}{\nu} \quad (5.50)$$

Ce nombre tient un rôle équivalent au nombre de Reynolds pour les équations de Navier-Stokes.

5.4.2 Position du problème

La méthode de frontière immergée décompose la résolution des équations en un problème fluide et un problème solide (cf. équation (5.40)). Les deux sont liés par une combinaison convexe dont la somme des pondérations vaut 1 :

$$\begin{aligned} \left(\frac{\partial \phi}{\partial t} \right)_{fluide} &= -u \frac{\partial \phi}{\partial x} + \nu \frac{\partial^2 \phi}{\partial x^2} \times (1 - H) \\ \left(\frac{\partial \phi}{\partial t} \right)_{solide} &= \frac{\phi_c^{n+1} - \phi_c^n}{\Delta t} \times H \end{aligned} \quad (5.51)$$

$$\Rightarrow \left(\frac{\partial \phi}{\partial t} \right) = (1 - H) \left(-u \frac{\partial \phi}{\partial x} + \nu \frac{\partial^2 \phi}{\partial x^2} \right) + H \left(\frac{\phi_c^{n+1} - \phi_c^n}{\Delta t} \right)$$

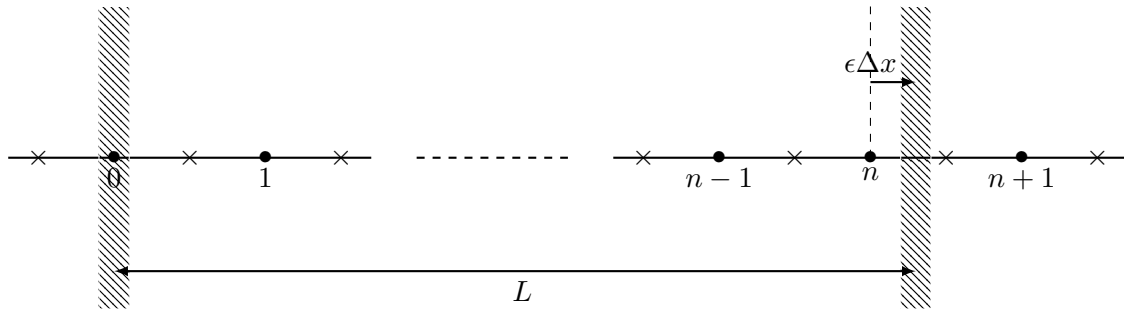


FIGURE 5.11 – Discrétisation du domaine pour l'analyse de la méthode de frontière immergée.

La consistance et la stabilité des schémas de discrétisation de l'équation d'advection-diffusion peuvent être démontrées dans certains cas (voir [Hirsch, 2007] pour des exemples détaillés).

Pour la partie "solide", la consistance et la stabilité n'ont pas réellement de sens dans la mesure où cette équation n'est pas une équation régissant l'évolution d'une quantité mais visant à forcer la valeur de ϕ au voisinage du corps après un pas de temps :

$$\phi_c^{n+1} = \phi^{n+1} \quad (5.52)$$

L'application à chaque contribution d'une pondération comprise entre 0 et 1 préserve la stabilité de la méthode globale.

Même si ceci ne tient pas lieu d'une démonstration rigoureuse, nous pouvons affirmer que les schémas numériques consistants et stables pour la discrétisation de l'équation d'advection-diffusion le resteront en présence de la méthode de frontière immergée.

Ainsi (5.40) admet une solution convergente.

Pour autant nous pouvons nous demander si la solution de l'équation (5.40) converge bien vers la solution physiquement attendue et si elle respecte les conditions aux limites.

5.4.3 Précision de la prise en compte des conditions aux limites

Pour étudier la précision de la prise en compte des conditions aux limites, un certain nombre d'hypothèses simplificatrices sont effectuées :

- nous cherchons la solution de l'équation stationnaire (5.48).
- (5.48) est discrétisée avec un schéma décentré d'ordre 1 (en considérant $u > 0$) pour le terme advectif et un schéma centré d'ordre 2 pour le terme diffusif, nous avons alors :

$$u \frac{\phi_i - \phi_{i-1}}{\Delta x} = \nu \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \quad (5.53)$$

- le domaine 1D est discrétisé comme représenté sur la figure 5.11. Δx est le pas d'espace tel que $\Delta x = \frac{L}{n+\epsilon}$, avec n le nombre total de points de discrétisation à l'intérieur du domaine. L'emplacement des limites du domaine est fixé de sorte que la frontière gauche coïncide exactement avec le premier point de discrétisation. Ainsi l'imposition de la condition aux limites de ce côté est exacte. Pour la frontière à droite en $x = L$, on choisit une position arbitraire entre les deux points de discrétisation n et $n + 1$. Cette condition aux limites sera imposée avec la méthode de frontière immergée. Nous supposons que celle-ci n'impacte que les points n et $n + 1$. Avant ces points, le terme source n'intervient pas. Au-delà, la valeur de ϕ est fixée à ϕ_R . L'espace entre les points n et $n + 1$ représente donc la zone de transition entre le domaine "fluide" et le domaine "solide".

Les hypothèses choisies et le contexte dans lequel elles s'inscrivent nous éloignent un peu de l'environnement du code WCCH mais elles sont nécessaires pour mettre en évidence certains aspects de la méthode et proposer ensuite des améliorations.

Le système d'équations à résoudre se résume donc à :

$$\begin{cases} \phi_i = \phi_L & \text{si } i = 0 \\ \sigma_c(\phi_i - \phi_{i-1}) = \sigma_v(\phi_{i+1} - 2\phi_i + \phi_{i-1}) & \text{si } 1 \leq i \leq n-1 \\ (1 - H_i)(\sigma_v\phi_{i+1} - (2\sigma_v + \sigma_c)\phi_i + (\sigma_v + \sigma_c)\phi_{i-1}) + H_i(\phi_R - \phi_i) = 0 & \text{si } n \leq i \leq n+1 \\ \phi_i = \phi_R & \text{si } i > n+1 \end{cases} \quad (5.54)$$

avec

$$\sigma_c = \frac{u\Delta t}{\Delta x} \text{ et } \sigma_v = \frac{\nu\Delta t}{\Delta x^2} \quad (5.55)$$

Bien que le problème étudié soit stationnaire, le pas de temps Δt intervient dans le système à cause du terme source. En effet le principe même de la méthode est de forcer la vitesse a posteriori, au terme de chaque pas de temps. Ainsi quel que soit le schéma d'intégration temporelle employé pour atteindre l'état stationnaire (Euler, Runge-Kutta, explicite ou implicite), le terme source subsiste et avec lui Δt . Par ailleurs H_i est le coefficient qui permet de progressivement imposer la bonne valeur de ϕ . Par principe $0 \leq H_n \leq H_{n+1} \leq 1$.

Dans la suite nous dissocions le cas purement advectif du cas purement diffusif pour des raisons de simplicité.

- Étude du cas purement diffusif, $u = 0$.

La solution exacte attendue est une droite affine :

$$\tilde{\phi}(x) = \frac{x - x_L}{x_R - x_L}(\phi_R - \phi_L) + \phi_L \quad (5.56)$$

ou d'un point de vue discret :

$$\tilde{\phi}_i = \frac{n + \epsilon - i}{n + \epsilon}\phi_L + \frac{i}{n + \epsilon}\phi_R \quad (5.57)$$

Si les conditions aux limites étaient prises en compte de façon exacte, le résultat numérique que l'on obtiendrait après résolution serait exactement égal à la solution continue. En effet, cette solution étant linéaire et la dérivée seconde de ϕ étant discrétisée à l'ordre 2 sur un maillage uniforme, nous nous attendrions à avoir $\phi_i = \tilde{\phi}_i$ pour tout i . Cependant, avec la méthode de frontière immergée, la condition de bord n'est prise en compte qu'à l'ordre 1. Une erreur non nulle est donc attendue.

Pour la partie du domaine loin de la frontière droite, c'est-à-dire $1 \leq i \leq n-1$, la solution numérique de $\phi_{i+1} - 2\phi_i + \phi_{i-1} = 0$ est une droite affine :

$$\phi_i = \frac{n-i}{n}\phi_L + \frac{i}{n}\phi_n \quad (5.58)$$

Ce qui nous permet de dire que :

$$\phi_{n-1} = \frac{1}{n}\phi_L + \left(1 - \frac{1}{n}\right)\phi_n \quad (5.59)$$

En résumé, les équations vérifiées aux points $n-1$, n et $n+1$ se résument au système :

$$\begin{pmatrix} 1 & -(1 - \frac{1}{n}) & 0 \\ \sigma_v(1 - H_n) & -2\sigma_v(1 - H_n) - H_n & \sigma_v(1 - H_n) \\ 0 & \sigma_v(1 - H_{n+1}) & -2\sigma_v(1 - H_{n+1}) - H_{n+1} \end{pmatrix} \begin{pmatrix} \phi_{n-1} \\ \phi_n \\ \phi_{n+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{n} \\ 0 \\ 0 \end{pmatrix} \phi_L + \begin{pmatrix} 0 \\ -H_n \\ -H_{n+1} - \sigma_v(1 - H_{n+1}) \end{pmatrix} \phi_R \quad (5.60)$$

La résolution du système (5.60), permet d'en déduire ϕ_n et, par là même, ϕ_i pour tout $i \in [1, n]$ d'après (5.58).

Cependant, nous allons nous intéresser uniquement à ϕ_n qui est le seul point du domaine "fluide" affecté par le terme source. En effet si le schéma converge en ce point, alors il sera alors globalement convergent. On suppose donc que l'erreur sera maximale en n , ce que confirme l'expérience numérique pour ce cas.

Après résolution de (5.60), il vient :

$$\phi_n = \frac{1}{1 + n/\beta} \phi_L + \frac{1}{1 + \beta/n} \phi_R \quad (5.61)$$

avec β un paramètre dépendant de σ_v , H_n et H_{n+1} tel que :

$$\beta = \frac{1}{\frac{-H_n}{(H_n-1)\sigma_v} + \frac{H_{n+1}-\sigma_v(H_{n+1}-1)}{H_{n+1}-2\sigma_v(H_{n+1}-1)}} \quad (5.62)$$

Ce résultat est à comparer avec la solution exacte en n donnée par (5.57) :

$$\tilde{\phi}_n = \frac{1}{1 + n/\epsilon} \phi_L + \frac{1}{1 + \epsilon/n} \phi_R \quad (5.63)$$

L'erreur commise par la résolution discrète des équations en ce point est donc :

$$e = \tilde{\phi}_n - \phi_n = \left(\frac{1}{1 + n/\epsilon} - \frac{1}{1 + n/\beta} \right) \phi_L + \left(\frac{1}{1 + \epsilon/n} - \frac{1}{1 + \beta/n} \right) \phi_R \quad (5.64)$$

On comprend qu'il faudrait faire en sorte que $\beta = \epsilon$ pour que l'erreur commise soit nulle. La méthode de frontière immergée serait alors au moins d'ordre 2. De plus, sans pour autant chercher à avoir une égalité stricte, l'erreur pourrait être minimisée en cherchant H_n et H_{n+1} solutions du problème d'optimisation non-linéaire sous contrainte :

$$\begin{cases} \min(|\beta - \epsilon|) \\ 0 < H_n < 1 \\ 0 < H_{n+1} < 1 \end{cases} \quad (5.65)$$

Cette démarche est possible puisque la solution est connue ici. Mais elle n'est nullement applicable et n'aurait pas de sens dans un cas général où la solution n'est pas connue a priori.

Nous étudions ensuite la convergence spatiale du schéma ainsi que la dépendance au pas de temps.

- **Convergence spatiale avec σ_v fixé.** Ce cas se produit si un pas de temps adaptatif est choisi $\Delta t = CFL \frac{\Delta x^2}{\nu}$. On a alors $\sigma_v = CFL$. Dans ce cas β est constant et indépendant de Δx . Un développement à l'ordre 2 en considérant n suffisamment grand donne :

$$e = \left(\frac{\beta - \epsilon}{n} \right) (\phi_R - \phi_L) + O\left(\frac{1}{n^2} \right) \quad (5.66)$$

La méthode est donc bien d'ordre 1 dans ce cas.

- **Convergence spatiale avec Δt fixé.** Dans ce cas, σ_v tend vers $+\infty$ quand n devient très grand. On a alors

$$\beta = 2 + O\left(\frac{1}{n} \right) \quad (5.67)$$

et l'erreur reste la même :

$$e = \left(\frac{\beta - \epsilon}{n} \right) (\phi_R - \phi_L) + O\left(\frac{1}{n^2} \right) \quad (5.68)$$

Ainsi quelle que soit la valeur de Δt , l'erreur locale sur ϕ_n est d'ordre 1 en espace.

- **Dépendance au pas de temps.** Faisons tendre Δt vers 0 avec n fixé. Alors :

$$\beta = \left(\frac{1}{H_n} - 1 \right) \frac{\nu}{\Delta x^2} \Delta t + O(\Delta t^2) \quad (5.69)$$

L'erreur sur ϕ_n se simplifie en :

$$e = \left(\frac{1}{1 + n/\epsilon} - \frac{\beta}{n} \right) (\phi_L - \phi_R) = \left(\frac{1}{1 + n/\epsilon} - \left(\frac{1}{H_n} - 1 \right) \frac{\nu}{L^2} n \Delta t \right) (\phi_L - \phi_R) + O(\Delta t^2) \quad (5.70)$$

$$e = K \left(\frac{1}{n} \right) + O(\Delta t) \quad (5.71)$$

Ceci confirme que l'erreur n'est pas indépendante du pas de temps.

- Étude du cas purement advectif, $\nu = 0$.

La solution attendue est une valeur constante égale à ϕ_L avec donc une discontinuité à la frontière droite d'amplitude $\phi_R - \phi_L$. Ce cas test représente la limite quand le nombre de Peclet tend vers l'infini. Il est singulier car la caractéristique principale est sortante en $x = L$ (u étant positif). Aucune condition ne devrait donc être imposée en sortie, l'information vient du domaine numérique. On peut donc se douter que la méthode ne va pas être efficace pour traiter la frontière à droite.

$$\begin{cases} \phi_i = \phi_L & \text{si } 0 \leq i \leq n-1 \\ \sigma_c(1 - H_i)(\phi_i - \phi_{i-1}) = H_i(\phi_R - \phi_i) & \text{si } n \leq i \leq n+1 \\ \phi_i = \phi_R & \text{si } i > n+2 \end{cases} \quad (5.72)$$

D'après (5.72), ϕ_n vaut :

$$\phi_n = \frac{1}{1 + \sigma_c \frac{1-H_n}{H_n}} \phi_R + \frac{1}{1 + \frac{H_n}{\sigma_c(1-H_n)}} \phi_L \quad (5.73)$$

La solution étant,

$$\tilde{\phi}_n = \phi_L \quad (5.74)$$

l'erreur vaut donc :

$$e = \tilde{\phi}_n - \phi_n = \frac{1}{1 + \sigma_c \frac{1-H_n}{H_n}} (\phi_L - \phi_R) \quad (5.75)$$

- **Convergence spatiale avec σ_c fixé.** Alors l'erreur e est indépendant de Δx et il n'y a pas convergence.
- **Convergence spatiale avec Δt fixé.** Alors $\sigma_c \rightarrow +\infty$ et l'erreur tend bien vers 0 à l'ordre 1 :

$$e = \frac{H_n}{1 - H_n} \frac{L}{u \Delta t} \frac{\phi_L - \phi_R}{n} + O\left(\frac{1}{n^2}\right) \quad (5.76)$$

- **Dépendance au pas de temps.** En faisant tendre Δt vers 0 avec Δx fixé :

$$e = \left(1 - \frac{u \Delta t}{\Delta x} \frac{1 - H_n}{H_n} \right) (\phi_L - \phi_R) + O(\Delta t^2) \quad (5.77)$$

Encore une fois l'erreur n'est pas indépendante de Δt .

En résumé, parmi toutes les formes de convergence envisagées, la convergence spatiale avec σ fixé est la plus représentative de ce qui est fait en pratique. Sur un cas purement diffusif, la méthode est bien d'ordre 1 localement. Pour l'advection seule, elle est d'ordre 0. Mais on s'aperçoit dans ce cas que l'erreur est d'autant plus faible que σ_c (et donc le nombre CFL) est grand. Cette étude nous a aussi permis de mettre en évidence la sensibilité de la méthode au pas de temps.

5.4.4 Validations et expériences numériques

Afin de valider les énoncés théoriques précédents, nous évaluons numériquement les différentes convergences et étudions l'influence de certains paramètres sur la précision de la méthode. Nous testons trois valeurs du nombre de Peclet : $Pe = 0$ qui correspond donc au cas où l'équation (5.47) se résume à une équation de diffusion, $Pe = 10$ et $Pe = \infty$ pour un problème d'advection pure. Les schémas numériques employés sont les mêmes que ceux que l'on retrouve dans le code WCCH : WENO5 pour le calcul du terme advectif, schéma centré d'ordre 2 pour le terme diffusif et avance en temps explicite avec un schéma Runge-Kutta d'ordre 4. Le nombre CFL de la partie advective est fixé à 0.7, celui de la partie visqueuse à 0.125. Le domaine discret est de longueur $L = 1.5$. En $x = 0.25$, une condition homogène de Dirichlet est imposée. En $x = 1.25$, on fixe la valeur de ϕ à 1. Ainsi pour $x \in [0; 0.25]$ et $x \in [1.25; 1.5]$, la grandeur ϕ est fixée au moyen de la méthode de frontière immergée. La longueur effective est donc de 1.

Pour les convergences spatiales, le nombre de volumes de contrôle évolue de 8 à 256 cellules. Pour l'étude de sensibilité au pas de temps, le nombre CFL de la partie advective varie de 0.01 à 1, et de 0.002 à 0.5 pour la partie visqueuse.

Une erreur absolue par rapport à la solution analytique est calculée en normes 1, 2 et infinie. Le calcul de l'erreur est effectué uniquement dans la zone "résolue", c'est-à-dire pour $0.25 \leq x \leq 1.25$.

Enfin nous avons vu que la méthode est sensible à la position de la frontière dans le domaine discret. Pour s'affranchir de cette dépendance, on effectue plusieurs simulations pour un même maillage (environ 40) en modifiant à chaque fois la position de l'interface par rapport au centre des volumes de contrôle. La longueur effective reste la même, la position des frontières est juste décalée d'une fraction de Δx . Cela revient à faire varier ϵ de 0 à 1. Une moyenne sur l'erreur des 40 simulations est ensuite réalisée pour ne pas faire apparaître la sensibilité à la position de la frontière sur la courbe de convergence.

- Convergence en espace en fixant le nombre CFL .

Le résultat de la convergence est tracé sur la figure 5.13a. Il s'agit du résultat de référence pour ensuite étudier l'influence de certains paramètres. Pour les nombres de Peclet égaux à 0 et 10, nous constatons que la méthode est d'ordre 1 pour toutes les normes comme cela est attendu. Pour le nombre de Peclet infini, c'est-à-dire en présence d'une advection pure, on retrouve bien le fait que la méthode ne converge pas (en norme L_∞) et qu'elle n'est pas consistante. Les normes L_1 et L_2 semblent converger mais à un ordre inférieur à 1, ce qui témoigne d'une non-convergence locale.

- Convergence en espace en fixant le pas de temps Δt .

Le pas de temps est choisi de sorte à respecter exactement la condition CFL sur le maillage le plus fin. Les courbes sont présentées figure 5.13b. Étonnamment, l'ordre de convergence est parfois supérieur à 1 notamment pour les normes L_1 et L_2 , alors que l'étude théorique prévoit une convergence d'ordre 1 pour toutes valeurs du nombre de Peclet. Pour les maillages les plus fins, l'erreur à Δt fixe est très proche de celle à CFL fixe. Rien de surprenant puisque les pas de temps sont alors équivalents. Mais sur un maillage plus grossier, l'erreur à Δt fixe est bien plus importante qu'à CFL fixe. La seule différence entre les deux est que dans le premier cas le pas de temps est très petit, alors que dans le second il est fixé par la condition CFL du maillage, et est donc bien plus élevé. De faibles valeurs de pas de temps sont donc préjudiciables à la convergence. Ceci est conforté par l'étude de convergence suivante.

- Convergence en temps avec Δx fixé.

La figure 5.13c présente l'étude de sensibilité au pas de temps pour un maillage fixe avec $n = 64$ cellules. Les courbes montrent que l'erreur dépend du pas de temps, mais pas suivant une loi en puissance comme l'étude théorique l'a mis en évidence. Pour $Pe = 0$ et $Pe = 10$, il existe une valeur de Δt (égale environ à 1/10 de la valeur de la CFL) pour laquelle l'erreur est minimale. Pour de faibles valeurs de pas de temps, l'erreur semble converger vers une valeur asymptotique.

Dans la suite, nous testons l'influence de la régularisation Heaviside et du calcul des indicateurs de régularité du schéma WENO. Toutes les convergences spatiales sont réalisées à CFL fixée.

- Influence de la régularisation de la fonction Heaviside pour imposer le champ de vitesse

L'utilisation d'une fonction Heaviside lissée dans la méthode de frontière immergée permet de progressivement imposer le terme source qui fixe la condition de bord de Dirichlet. Sans cela, le terme source est brutalement imposé et les domaines fluide et solide sont complètement découplés. Les convergences précédentes ont été conduites avec un Heaviside lissé qui assure la transition sur environ deux cellules. La figure 5.13d présente les courbes de convergence dans le cas où un Heaviside singulier est appliqué, c'est-à-dire sans transition dans l'application des conditions de bord. L'effet est négatif puisque, même si l'ordre de convergence reste à 1, le niveau d'erreur est environ 5 fois plus élevé que lorsque le Heaviside est régularisé. Fadlun et al. ont quant à eux constaté numériquement (sur les équations de Navier-Stokes) que l'imposition brutale du terme source donne dans leur cas un ordre de convergence inférieur à 1 [Fadlun et al., 2000]. Sans surprise l'erreur est nulle pour le nombre de Peclet infini. En effet, le schéma WENO arrive à capter précisément la discontinuité qui n'est pas atténuée par le Heaviside.

Par ailleurs, sur la figure 5.12 est tracée la convergence en espace pour $Pe = 10$ avec les erreurs de l'ensemble des simulations qui sont testées à chaque pas d'espace. L'écart-type des erreurs est nettement réduit avec régularisation. En l'absence d'effet lisseur, les erreurs présentent des extrema plus importants, les données sont plus étalées. Mais en valeur moyenne l'erreur est inférieure avec lissage.

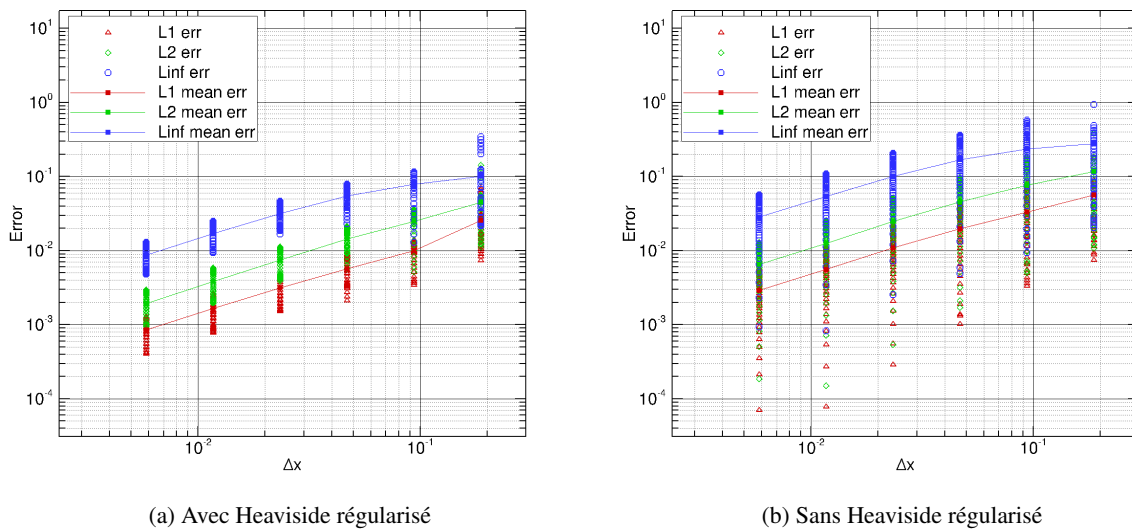


FIGURE 5.12 – Comparaison de la sensibilité à la position de la frontière avec et sans régularisation du Heaviside

- Influence des indicateurs de régularité du schéma WENO

A l'origine, les schémas WENO sont destinés à conserver un ordre de précision élevé sauf au voisinage des discontinuités pour prévenir l'apparition d'oscillations. La reconstruction des flux aux faces des volumes de contrôle s'effectue par une interpolation polynomiale avec une approche hiérarchique qui choisit parmi plusieurs stencils celui sur lequel le champ reconstruit est le plus régulier (cf. introduction du schéma WENO dans le Chapitre 1). Des indicateurs de régularité du champ sont donc calculés. En l'absence de discontinuité et pour des champs réguliers, la nécessité de calculer ces indicateurs pose question. En effet, leur calcul a un coût CPU non négligeable, il est alors tentant de "shunter" cette étape. Cependant de forts gradients sont

attendus au voisinage du corps, notamment à haut Reynolds, avec le risque de voir apparaître des oscillations. Les courbes de convergence spatiale sont représentées sur la figure 5.13e. Les résultats avec et sans calcul des indicateurs de régularité sont sensiblement les mêmes hormis pour $Pe \rightarrow 0$. Pour ce cas la méthode sans indicateur donne bien évidemment de très mauvais résultats car la discontinuité est mal capturée.

En définitive, cette étude a permis de mettre en évidence que, dans le cas d'une équation linéaire d'advection-diffusion, la méthode de frontière immergée originelle du code WCCH est d'ordre 1. Il semble possible de construire une fonction Heaviside pour passer à l'ordre 2 (cf. étude sur un cas diffusif pur). Néanmoins l'intérêt pratique est nul puisqu'il suppose de connaître a priori la solution de l'écoulement. Il semble préférable de passer à des méthodes d'interpolation pour prendre en compte de façon exacte la position de la frontière, afin d'améliorer la précision de la méthode. Enfin, une autre limitation de cette méthode est qu'elle convient uniquement pour des conditions aux limites de Dirichlet sur la vitesse.

5.5 Amélioration et montée en ordre de la méthode de frontière immergée

5.5.1 Réduction de la sensibilité au pas de temps

Comme nous l'avons vu, la méthode de frontière immergée dépend du pas de temps employé par le schéma d'intégration temporelle. Ce constat est un peu déroutant pour des simulations stationnaires. Cette dépendance est en fait introduite par le lissage de la fonction Heaviside qui crée une région de transition où cohabitent la résolution fluide et l'imposition du terme source. Si on reprend la formulation en vitesse de la méthode (5.43) :

$$\mathbf{u}_i^{n+1} = (1 - H_i)\mathbf{u}_{i,fluide}^{n+1} + H_i\mathbf{u}_{i,solide}^{n+1} \quad (5.78)$$

avec, pour un schéma d'avance en temps Euler d'ordre 1 :

$$\mathbf{u}_{i,fluide}^{n+1} = \mathbf{u}_{i,fluide}^n + \Delta t \mathbf{RHS}_i^n \quad (5.79)$$

En régime stationnaire, si Δt varie plus ou moins brutalement alors $\mathbf{u}_{i,fluide}^{n+1}$ va varier, ainsi que \mathbf{u}_i^{n+1} . En effet, au voisinage du corps ($0 < H < 1$), \mathbf{RHS}_i^n n'est pas nul, même en régime stationnaire. Une variation soudaine interviendra alors sur les champs de vitesse et de pression, visible par un pic sur le signal d'effort calculé. La solution va ensuite converger vers un nouvel état stationnaire assez proche de l'état précédent. En pratique, comme nous utilisons une procédure de pas de temps adaptatif,

$$\Delta t_c = CFL_{c,\min} \frac{\Delta x_i}{|\mathbf{u}_i| + c_i} \text{ ou } \Delta t_c = CFL_{c,\min} \frac{\Delta x_i}{|\mathbf{u}_i|} \quad (5.80)$$

ce type de problème peut se manifester dès qu'il y a une variation de vitesse dans l'écoulement ou un changement de taille de maillage. C'est par exemple le cas après l'appel aux procédures de raffinement/dé-raffinement en cours de calcul.

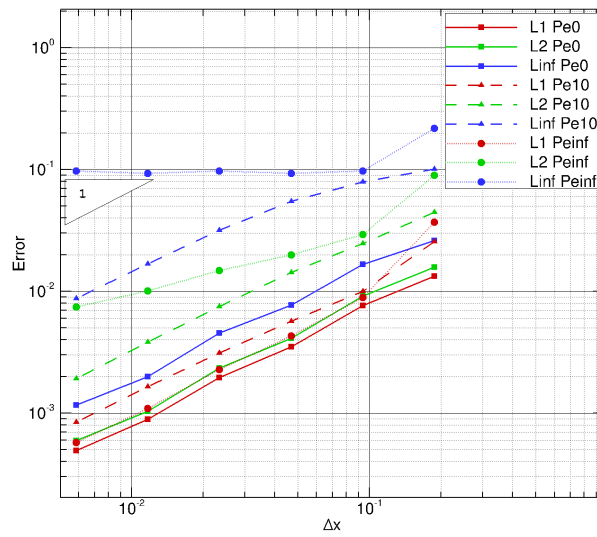
Pour réduire la sensibilité au pas de temps, nous souhaitons trouver une nouvelle fonction H_i qui puisse assurer l'indépendance de la solution vis-à-vis de Δt en régime stationnaire. Pour cela, il faut envisager H_i comme dépendante de Δt .

Soit un état stationnaire noté 1, il vérifie d'après (5.78) :

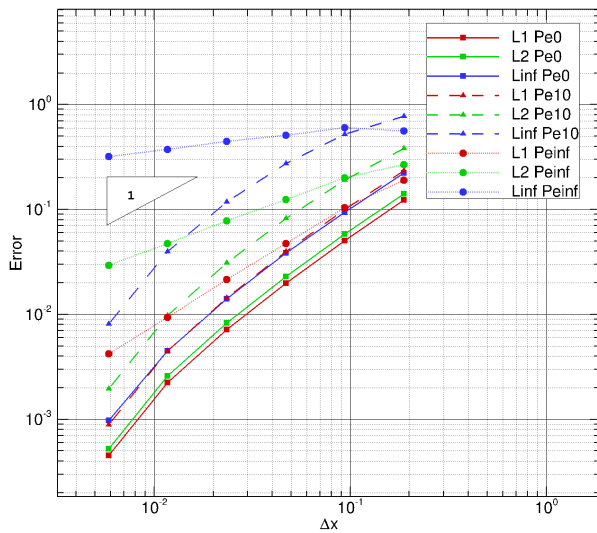
$$\Delta t_1(1 - H_i^1)\mathbf{RHS}_i^n + H_i^1(\mathbf{u}_{i,solide} - \mathbf{u}_i^n) = 0 \quad (5.81)$$

Si on considère qu'avec un autre pas de temps Δt_2 , la solution doit rester inchangée (même valeur de \mathbf{RHS}_i^n et \mathbf{u}_i^n), alors :

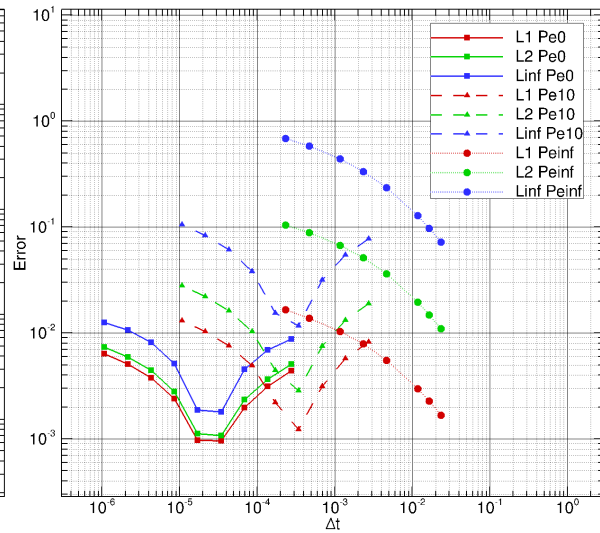
$$\Delta t_2(1 - H_i^2)\mathbf{RHS}_i^n + H_i^2(\mathbf{u}_{i,solide} - \mathbf{u}_i^n) = 0 \quad (5.82)$$



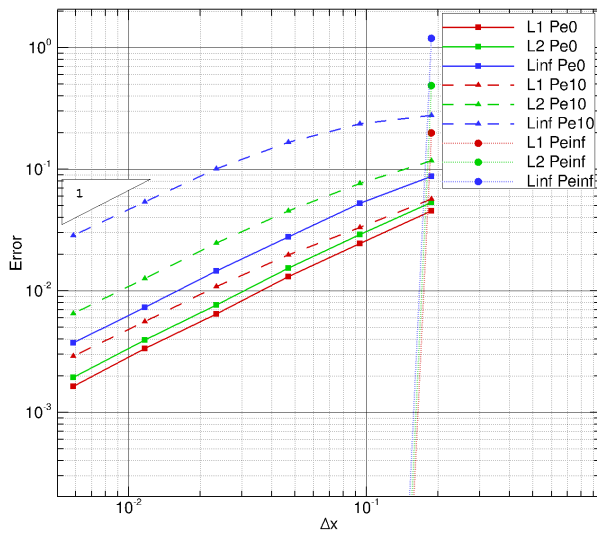
(a) Convergence spatiale à CFL fixe



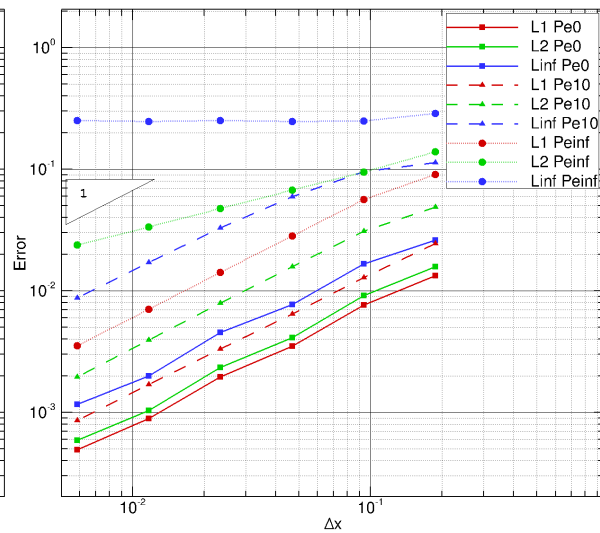
(b) Convergence spatiale à Δt fixe



(c) Convergence temporelle à Δx fixe



(d) Convergence spatiale sans régularisation du Heaviside (à CFL fixe)



(e) Convergence spatiale sans indicateurs de régularité du schéma WENO (à CFL fixe)

FIGURE 5.13 – Courbes de convergence de la méthode de frontière immergée pour différentes configurations et différentes valeurs de Pe

Ces deux équations nous amènent à la relation suivante entre H_i^1 et H_i^2 :

$$H_i^2 = \frac{H_i^1}{H_i^1 + (1 - H_i^1) \frac{\Delta t_1}{\Delta t_2}} \quad (5.83)$$

Ainsi nous pouvons proposer une nouvelle formulation de la fonction Heaviside \tilde{H} fonction de l'ancienne H :

$$\tilde{H} = \frac{H}{H + (1 - H) \frac{\Delta t_0}{\Delta t}} \quad (5.84)$$

Δt_0 est un pas de temps arbitraire qui pourra être fixé en début de simulation. Si $\Delta t = \Delta t_0$ alors $\tilde{H} = H$. Plus Δt est proche de Δt_0 , plus \tilde{H} s'approchera de H . La figure 5.14 représente \tilde{H} pour différentes valeurs du ratio $\frac{\Delta t_0}{\Delta t}$ avec H définie par (1.27) et avec $\alpha = 3$. On remarque que le principal effet introduit par la nouvelle formulation est de déplacer \tilde{H} autour de la position originale. Par exemple le cas $\frac{\Delta t_0}{\Delta t} = 2$ correspond à un déplacement de la frontière solide vers la droite. Les valeurs asymptotiques restent inchangées : loin du corps \tilde{H} est nul et $\tilde{H} = 1$ loin du fluide. On trace également sur la figure 5.14 l'erreur au cours du temps dans le cas où la simulation est réalisée avec un pas de temps oscillant (d'amplitude relative de 40% et de période $\sim 440\Delta t_0$) pour l'équation linéaire d'advection-diffusion ($Pe = 10$, RK4, $n = 64$, $CF L_v = 0.125$, $\Delta t_0 = 0.0007$). La nouvelle fonction Heaviside permet ainsi de lisser les champs et d'éviter les oscillations sans changer l'erreur moyenne malgré le "déplacement" de la frontière induit par cette nouvelle fonction.

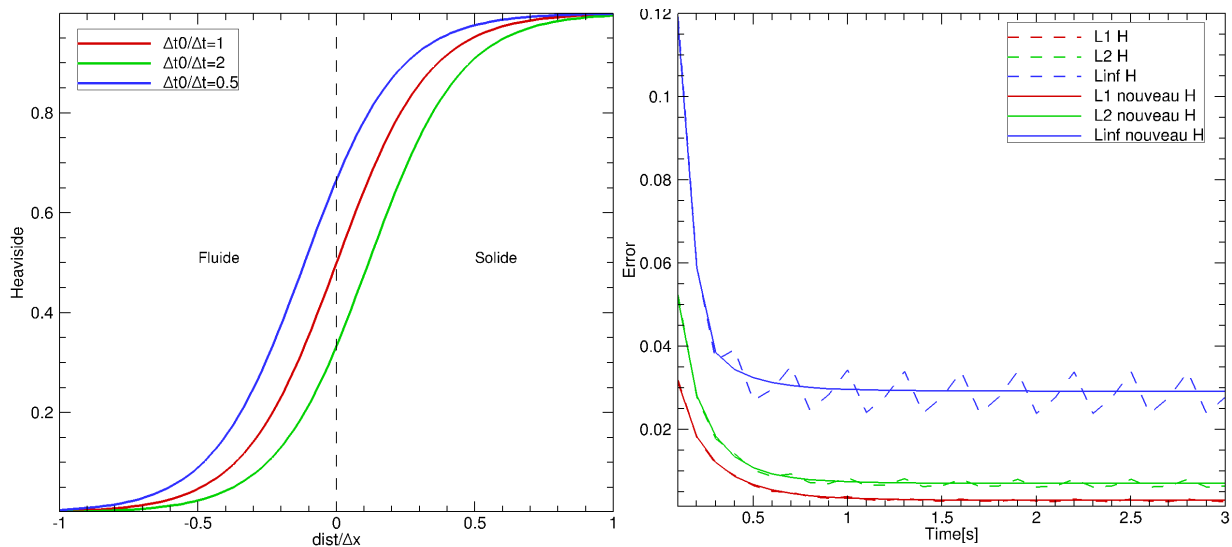


FIGURE 5.14 – Nouvelle fonction Heaviside pour différentes valeurs du ratio $\frac{\Delta t_0}{\Delta t}$ (gauche) et effet lisseur pour un pas de temps oscillant (droite)

5.5.2 Adaptation de la méthode pour les schémas Runge-Kutta

Jusqu'à présent, la méthode n'a été considérée qu'avec un schéma d'avance en temps explicite Euler d'ordre 1. Pour autant, la méthode de forçage discret explicite n'est pas limitée à ce type de schéma. Mohd-Yusof [Mohd-Yusof, 1997] utilise par exemple sa méthode avec un schéma Runge-Kutta d'ordre 3. Kim et al. [Kim et al., 2001] recourent également à ce schéma d'intégration temporelle. L'une des principales interrogations avec l'utilisation des schémas Runge-Kutta concerne la forme du terme source. Quelle forme

doit-il prendre ? En particulier quel pas de temps choisir à chaque sous-itération pour le terme source :

$$\mathbf{f}^n = -\mathbf{RHS}^n + \frac{\mathbf{u}_{solide}^{n+1} - \mathbf{u}^n}{\Delta t} \quad (5.85)$$

Les auteurs qui appliquent ce type de méthode de frontière immergée avec des schémas Runge-Kutta [Mohd-Yusof, 1997, Kim et al., 2001, Bigot et al., 2014, Uhlmann, 2005] fixent le pas de temps du terme source au pas de temps global de l'itération. Ce choix a du sens en considérant la décomposition en vitesse fluide et en vitesse solide (cf. (5.43)) où il apparaît que le schéma d'intégration temporelle des équations de Navier-Stokes ne concerne que la partie fluide.

En conséquence un schéma Runge-Kutta avec terme source doit s'écrire :

$$\forall i = 1, \dots, s \begin{cases} t^i = t^n + c_i \Delta t \\ \mathbf{u}^i = \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \left((1-H)\mathbf{RHS}^j + H \frac{\mathbf{u}_{solide}^{n+1} - \mathbf{u}^n}{\Delta t} \right) \end{cases} \quad (5.86)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \sum_{j=1}^s b_j \left((1-H)\mathbf{RHS}^j + H \frac{\mathbf{u}_{solide}^{n+1} - \mathbf{u}^n}{\Delta t} \right)$$

Cette méthode permet d'imposer correctement le champ de vitesse après la dernière étape du schéma Runge-Kutta. En effet $\sum_{j=1}^s b_j = 1$ et ainsi

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \sum_{j=1}^s b_j \left((1-H)\mathbf{RHS}^j \right) + \Delta t H \frac{\mathbf{u}_{solide}^{n+1} - \mathbf{u}^n}{\Delta t} \quad (5.87)$$

que l'on peut réécrire sous la forme d'une décomposition de vitesse fluide et solide :

$$\mathbf{u}^{n+1} = (1-H) \left[\mathbf{u}^n + \Delta t \sum_{j=1}^s b_j \left((1-H)\mathbf{RHS}^j \right) \right] + H \mathbf{u}_{solide}^{n+1} \quad (5.88)$$

Ainsi la bonne valeur du champ de vitesse solide est imposée au terme du schéma Runge-Kutta. En revanche la vitesse solide imposée à chaque sous itération n'est pas la bonne. Ceci conduit à une dégradation de l'ordre de convergence temporelle du schéma au voisinage de la frontière.

Pour la version précédente du code WCCH, le choix avait été fait d'imposer le bon champ de vitesse à chaque sous-étape [Bigay, 2015, Bigay et al., 2017] :

$$\forall i = 1, \dots, s \begin{cases} t^i = t^n + c_i \Delta t \\ \mathbf{u}^i = \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \left((1-H)\mathbf{RHS}^j + H \frac{\mathbf{u}_{solide}^j - \mathbf{u}^n}{c_j \Delta t} \right) \end{cases} \quad (5.89)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \sum_{j=1}^s b_j \left((1-H)\mathbf{RHS}^j + H \frac{\mathbf{u}_{solide}^j - \mathbf{u}^n}{c_j \Delta t} \right)$$

Mais, après la dernière étape, à l'intérieur du corps où $H \equiv 1$:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \sum_{j=1}^s \frac{b_j}{c_j} (\mathbf{u}_{solide}^j - \mathbf{u}^n) \quad (5.90)$$

soit

$$\mathbf{u}^{n+1} = -\frac{1}{2} \mathbf{u}^n + \sum_{j=1}^s \frac{b_j}{c_j} \mathbf{u}_{solide}^j \quad (5.91)$$

La vitesse n'est donc pas exactement celle qui est attendue. Si on envisage une vitesse \mathbf{u}_{solide} stationnaire, à convergence on aura bien $\mathbf{u}^{n+1} = \mathbf{u}_{solide}$. Avec cette approche on s'attend aussi à ce que l'ordre de convergence temporelle soit dégradé au voisinage du corps.

En conclusion, la décomposition de la vitesse en vitesse fluide et vitesse solide conduit à fixer le terme source sous la forme :

$$\mathbf{f}^n = -\mathbf{RHS}^n + \frac{\mathbf{u}_{solide}^{n+1} - \mathbf{u}^n}{\Delta t} \quad (5.92)$$

avec Δt le pas de temps global d'une avance en temps. C'est l'approche retenue dans la littérature pour traiter des écoulements autour de solide avec un forçage discret explicite impliquant la fraction volumique solide. Dans l'approche originale de WCCH, c'est le pas de temps de chaque sous-étape du schéma Runge-Kutta qui était choisi.

Cependant, différents tests numériques conduits sur l'une ou l'autre des deux approches n'ont pas permis de les départager. Elles donnent des résultats visiblement équivalents, du moins dans le cadre de vitesses constantes du corps.

5.5.3 Vers une montée en ordre de la méthode

Nous l'avons vu, les méthodes basées sur l'approche de forçage discret utilisant la fraction de volume pour progressivement imposer le champ de vitesse solide s'appliquent principalement à des écoulements laminaires ou des problèmes de sédimentation de particules. La nécessité de prendre en compte précisément la position de l'interface pour viser des écoulements à haut Reynolds semblent faire consensus auprès de la communauté.

Parmi toutes les méthodes listées dans l'état de l'art, la méthode de Balaras [Balaras, 2004] représente l'extension plus ou moins naturelle de la méthode originelle utilisée dans WCCH vers un ordre de convergence plus élevé. De plus les déconvenues rencontrées avec les méthodes cut-cell et ghost-cell en formulation faiblement-compressible au cours de la thèse de P. Bigay nous ont confortés dans le choix de cette méthode. Dans ce qui suit, nous proposons une formulation qui s'inspire de [Balaras, 2004].

Nous étudierons d'abord la méthode sur l'équation linéaire d'advection-diffusion pour vérifier son ordre et la comparer à la méthode originelle d'ordre 1. Ensuite nous proposerons une extension 2D (l'extension en 3D est triviale) pour les équations de Navier-Stokes. Pour la direction d'interpolation, contrairement à Balaras, nous avons fait le choix de privilégier une pondération des interpolations dans les directions cartésiennes, comme l'a proposée Berthelsen [Berthelsen et Faltinsen, 2008] pour sa méthode ghost-cell.

Principe en 1D

Le terme source est toujours calculé de la même façon :

$$\mathbf{f}_i^n = \frac{\mathbf{u}_{i,solide}^{n+1} - \mathbf{u}_i^n}{\Delta t} - \mathbf{RHS}_i^n \quad (5.93)$$

où \mathbf{u}_i est la vitesse que l'on souhaite imposer. A l'aide d'un développement en série de Taylor à l'ordre 1 au voisinage de la frontière solide :

$$\mathbf{u}_i = \mathbf{u}_{i,solide} + (\nabla \mathbf{u}|_{i,solide})d + O(d^2) \quad (5.94)$$

où d est la distance à la frontière.

La figure (5.15) présente en 1D un domaine discrétisé au voisinage d'une frontière. Pour les volumes de contrôle verts, la résolution des équations est effectuée normalement, sans terme source. La cellule rouge

est la première cellule fluide sur laquelle le terme source est ajouté. Pour calculer le gradient de vitesse à la paroi (décentré), on s'appuie sur la première cellule fluide (verte) :

$$\mathbf{u}_i = \mathbf{u}_{i,\text{solide}} - \alpha \frac{\mathbf{u}_{i,\text{solide}} - \mathbf{u}_{i-1}}{1 + \alpha} \quad (5.95)$$

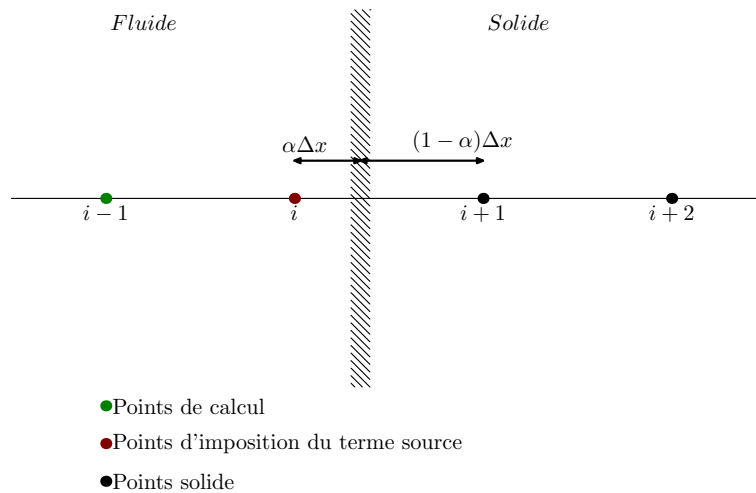


FIGURE 5.15 – Position de la frontière par rapport au domaine discrétisé

Pour l'évaluation numérique de la précision de cette méthode, nous reprenons l'équation linéaire d'advection-diffusion pour différentes valeurs du nombre de Peclet. La figure 5.16 présente les courbes de convergence pour la méthode d'ordre 2. Les résultats pour la méthode d'ordre 1 sont aussi rappelés. Dans la suite, tout ce qui est marqué "ordre 1" fait référence à la méthode de forçage discret avec lissage du terme source telle qu'elle est utilisée dans WCCH. "ordre 2" renvoie à la méthode que nous mettons au point à partir de celle de Fadlun et al. et de Balaras (5.95). Pour l'écoulement avec $Pe = 10$, la méthode d'ordre 2 donne bien le résultat attendu et une convergence d'ordre 2 pour toutes les normes. Pour l'écoulement purement diffusif, $Pe = 0$, l'erreur est nulle (10^{-13}). La solution étant linéaire, il est logique que la méthode d'ordre 2 soit en mesure de la déterminer exactement.

Il est intéressant de noter que sur des maillages relativement "grossiers", l'erreur de la méthode d'ordre 1 est inférieure à celle de la méthode d'ordre 2. Cependant la pente de convergence étant plus raide avec la méthode d'ordre 2, la tendance s'inverse sur les maillages les plus fins. Enfin pour $Pe \rightarrow \infty$, la même inconsistance qu'avec la méthode d'ordre 1 est observée.

Extension aux équations de Navier-Stokes multidimensionnelles

La méthode décrite précédemment est facilement extensible en 3D. Il reste cependant à choisir la direction d'interpolation. Fadlun et al. la choisissent de façon purement arbitraire, mais Balaras a montré que les meilleurs résultats étaient obtenus en calculant des gradients dans la direction normale à l'interface. En pratique cette condition nécessite la recherche d'un voisinage sur des cellules diagonales. En l'état actuel du code WCCH, il est aisé de déterminer les cellules voisines d'une cellule dans les directions cartésiennes de l'espace. L'accès aux cellules diagonales n'est en revanche pas immédiat. Ceci est aussi lié à la stratégie de parallélisation. Un processeur héberge toutes les cellules OUTER (cf. Chapitre 1 d'introduction au code WCCH) nécessaires pour compléter les calculs de flux aux frontières du domaine hébergé par le processeur. Ces cellules OUTER ne sont définies que pour les calculs de flux dans les directions cartésiennes de l'espace. Ainsi un processeur qui héberge une cellule en frontière de domaine (cellule INNER) n'héberge

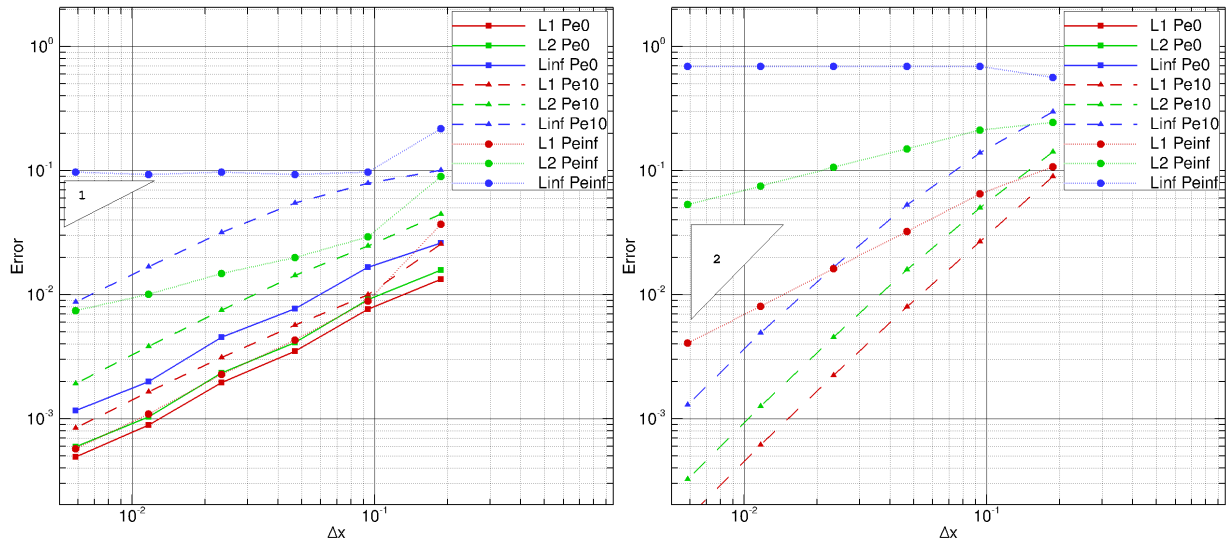


FIGURE 5.16 – Comparaison des méthodes de frontière immergée, ordre 1 (à gauche) ordre 2 (à droite)

pas forcément toutes ses voisines dans les directions diagonales. Des balayages et communications MPI supplémentaires permettraient de résoudre cette lacune. Mais le coût CPU de l'interpolation s'en trouverait grandement alourdi, surtout si la procédure doit être appelée à chaque pas de temps. Pour pallier ce problème, on ne calcule des gradients que dans les directions de balayage et on pondère leur somme suivant l'orientation de la facette du solide. Cette méthode est directement inspirée de Berthelsen et al. [Berthelsen et Faltinsen, 2008] qui l'utilisent pour pondérer les interpolations de cellules fantômes.

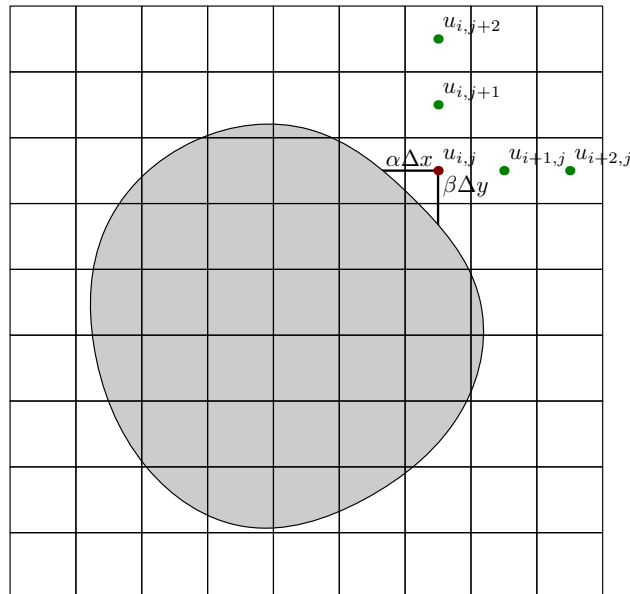


FIGURE 5.17 – Reconstruction des points aux frontières (●) à partir des points de calcul (●)

La vitesse interpolée s'écrit (Fig. 5.17) :

$$\mathbf{u}_{i,j} = \frac{1}{1 + \left(\frac{\alpha\Delta x}{\beta\Delta y}\right)^2} \left(\mathbf{u}_{solide} - \alpha \frac{\mathbf{u}_{i+1,j} - \mathbf{u}_{solide}}{1 + \alpha} \right) + \frac{1}{1 + \left(\frac{\beta\Delta y}{\alpha\Delta x}\right)^2} \left(\mathbf{u}_{solide} - \beta \frac{\mathbf{u}_{i,j+1} - \mathbf{u}_{solide}}{1 + \beta} \right) \quad (5.96)$$

Avec cette pondération, seule la direction verticale ou horizontale est privilégiée lorsque la facette du corps est horizontale ou verticale.

La figure 5.18 présente pour un cylindre et un profil NACA la première couche de cellules de fluide auxquelles on applique le terme source pour imposer la vitesse. Les cellules bleues sont les cellules intérieures au corps. Aucun traitement particulier n'est fait sur ces dernières, elles sont justes présentées pour mettre en évidence les premières.

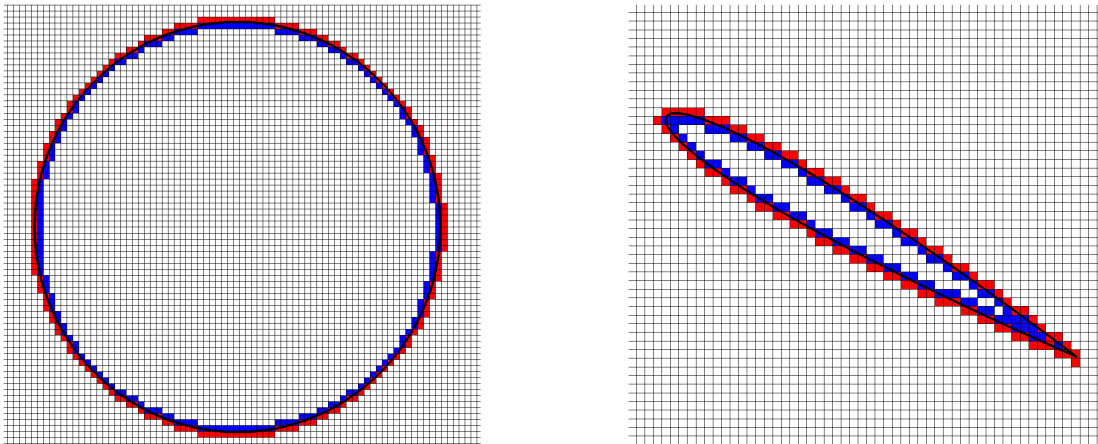


FIGURE 5.18 – ■ : 1^{ère} couche de cellules fluide, ■ : 1^{ère} couche de cellules solide

La figure 5.19 représente les valeurs des coefficients de pondération évalués au centre des volumes de contrôle pour un cylindre. On voit que pour les facettes du corps presque verticales, la direction d'interpolation privilégiée est la direction horizontale (coefficient de pondération proche de 1). Pour des facettes presque horizontales, le coefficient de pondération est quasi nul et c'est la direction verticale qui est privilégiée pour l'interpolation.

Détails sur l'implémentation

- **Faiblement-compressible** : pour cette formulation, seule la vitesse est imposée. Comme pour la méthode d'ordre 1, la pression est laissée libre au voisinage et à l'intérieur du corps. Aucun terme source n'est ajouté dans l'équation de continuité. Par ailleurs, la vitesse n'est imposée que dans le voisinage de la frontière. A l'intérieur du solide (à environ $2\Delta x$ de la frontière), aucun terme source n'est ajouté, les équations de Navier-Stokes sont pleinement résolues. Cela permet de décontraindre le système et d'apporter plus de flexibilité.
- **Incompressible** : nous reprenons la formulation incrémentale du schéma de projection (cf. Chapitre 2 équation (5)), à ceci près que la pression n'est pas incrémentée à l'intérieur du solide. En effet l'expérience numérique a montré que la pression tend à augmenter indéfiniment dans le corps au cours du calcul. Ce problème est lié au fait que les deux contraintes : cinématique sur la vitesse interpolée et de divergence nulle ne sont pas compatibles. Par ailleurs ce parti pris n'a pas d'incidence sur la consistance et la précision du schéma puisque, dans et au voisinage du corps, la cinématique est imposée sans faire appel à la résolution des équations de Navier-Stokes. Enfin une conséquence directe est que la contrainte cinématique est privilégiée au détriment de la garantie de la divergence nulle de l'écoulement.

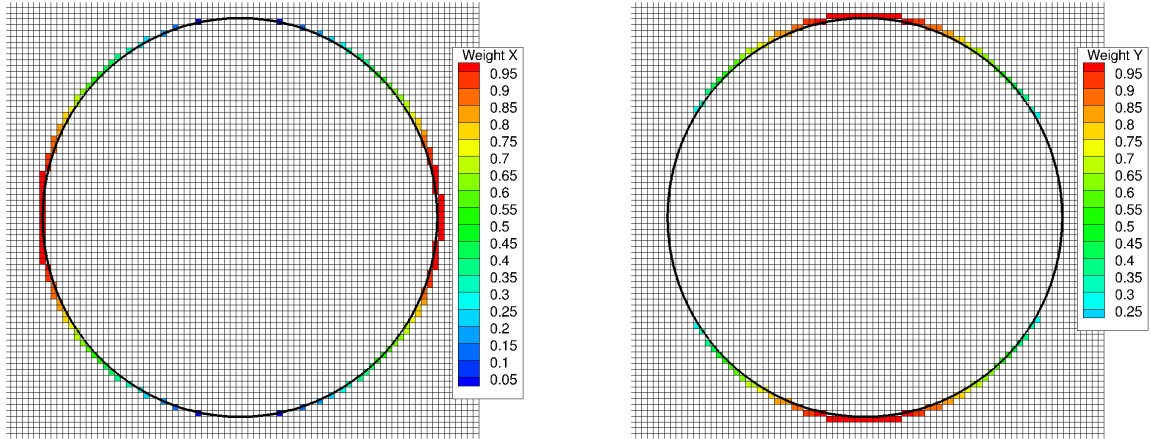


FIGURE 5.19 – Valeurs des coefficients de pondération pour la direction horizontale (gauche) et verticale (droite).

- **Régularisation au voisinage de la frontière** : nous avons également envisagé de lisser l'imposition du terme source à la traversée de la frontière solide, à la manière de ce qui se fait pour l'ordre 1 (1.27). La fonction (tracée sur la figure 5.20) prend la forme :

$$H\left(\frac{\phi(\mathbf{x})}{\Delta x}\right) = \frac{1}{2} \left(1 + \tanh \left(-4 \left(\frac{\phi(\mathbf{x})}{\Delta x} - \frac{1}{2} \right) \right) \right) \quad (5.97)$$

Par rapport à celle d'ordre 1, elle est décalée en espace côté fluide pour s'appliquer aux points de forçages qui sont situés en dehors de la région solide, près de la frontière.

Nous pouvons nous attendre à ce que cette régularisation pénalise l'ordre de la méthode et le fasse chuter à 1. Mais nous pouvons aussi espérer un effet bénéfique sur la régularité des champs au voisinage du corps, en particulier pour la formulation faiblement-compressible où le lissage du terme source a déjà montré son intérêt avec la méthode d'ordre 1. Une telle approche avec une méthode d'ordre 2 ne semble d'ailleurs pas saugrenue et est retenue par exemple par [Belliard et al., 2016] pour limiter les oscillations parasites en présence de corps mobiles (SFO).

5.5.4 Validations numériques de la montée en ordre

Écoulement autour d'un cylindre à $Re=40$

Nous reprenons ici l'écoulement 2D autour d'un cylindre à $Re = 40$ avec la même mise en données que dans le Chapitre 4. Une discrétisation assez grossière ($\frac{D}{\Delta x} = 17$) a été retenue avec pour objectif de mettre en évidence le gain sur la précision des résultats grâce à la montée en ordre. La figure 5.21 présente les champs de pression adimensionnés pour les différentes méthodes de frontière immergée (ordre 1 lissé, ordre 2 et ordre 2 lissé) en formulations incompressible et faiblement-compressible.

A l'ordre 1 lissé, les champs de pression sont semblables à l'extérieur du corps pour les deux formulations. A l'intérieur, ils diffèrent quelque peu. En particulier, le champ à la traversée de la frontière a l'air moins régulier en formulation faiblement-compressible.

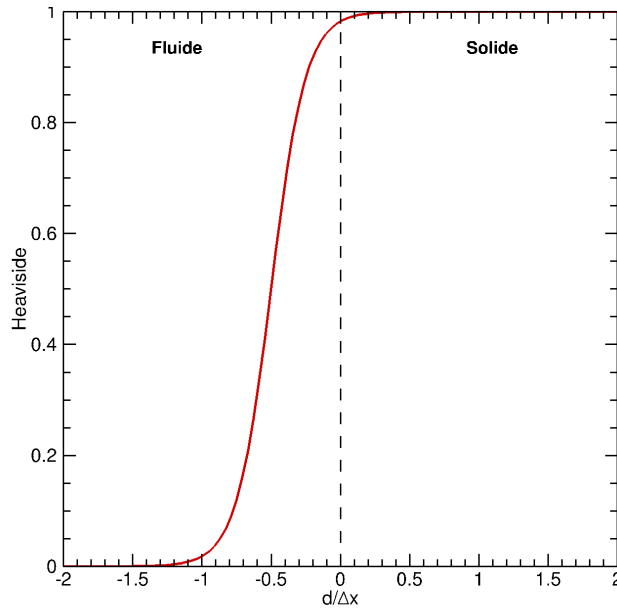


FIGURE 5.20 – Lissage de l'imposition du terme au source au voisinage du corps pour une méthode d'ordre 2.

A l'ordre 2, les deux formulations donnent des résultats sensiblement différents. En formulation incompressible, le champ de pression est régulier à l'extérieur du corps mais nul à l'intérieur de celui-ci, comme attendu puisque le champ de pression n'y est pas incrémenté. En revanche, pour la formulation faiblement-compressible, l'interpolation des vitesses sur les premières cellules du fluide du domaine a un effet désastreux sur la régularité du champ. Pour les cellules solides en contact avec le fluide, la pression prend des valeurs jusqu'à trois fois la valeur au point d'arrêt de l'écoulement. Le champ extérieur au cylindre s'en trouve fortement perturbé également. Les variations de densité au cours de la simulation sont de l'ordre de 1%, ce qui permet de valider l'hypothèse faiblement-compressible malgré tout. Pour les cellules intérieures au cylindre, qui ne voient pas leur vitesse imposée par la contrainte cinématique, le champ de pression est uniforme.

Pour la méthode d'ordre 2 lissée, le champ de pression est fortement atténué pour les deux formulations, avec une valeur au point d'arrêt très inférieure à la valeur attendue. Le lissage avec la fonction Heaviside régularisée détériore donc les résultats.

Pour mieux cerner les différences entre les méthodes de frontière immergée et les formulations compressible et incompressible, nous traçons les profils de pression et de vitesse le long de l'axe horizontal médian, c'est-à-dire pour $y = 0$ (Fig. 5.22). La partie $-0.5 < X/D < 0.5$ correspond au domaine solide du cylindre. Nous choisissons comme référence un calcul effectué avec le logiciel commercial Star-CCM+. Pour avoir une solution convergée spatialement, un maillage avec $\frac{D}{\Delta x} = 120$ au voisinage du corps (concentré dans un rayon d'environ $1 \times D$) a été retenu. Avec ce calcul, nous disposons d'une référence "body-fitted" dont on sait que les conditions de bords sont imposées de façon exacte.

Les profils de pression confirment les résultats précédemment énoncés. En incompressible à l'ordre 2, la première cellule fluide a bien une pression nulle même si ce n'est pas clairement mis en évidence sur la courbe. En effet les profils sont obtenus à partir du logiciel de post-traitement qui extrait les données du domaine en les interpolant le long d'une ligne. En faiblement compressible nous constatons que le champ de pression est fortement irrégulier à l'ordre 2. Finalement la méthode d'ordre 1 lissée s'approche le plus de la solution "body-fitted", que ce soit en amont ou en aval du cylindre. Elle donne des champs continus et réguliers à la traversée de la frontière. Si les efforts appliqués au cylindre devait être évalués en intégrant le

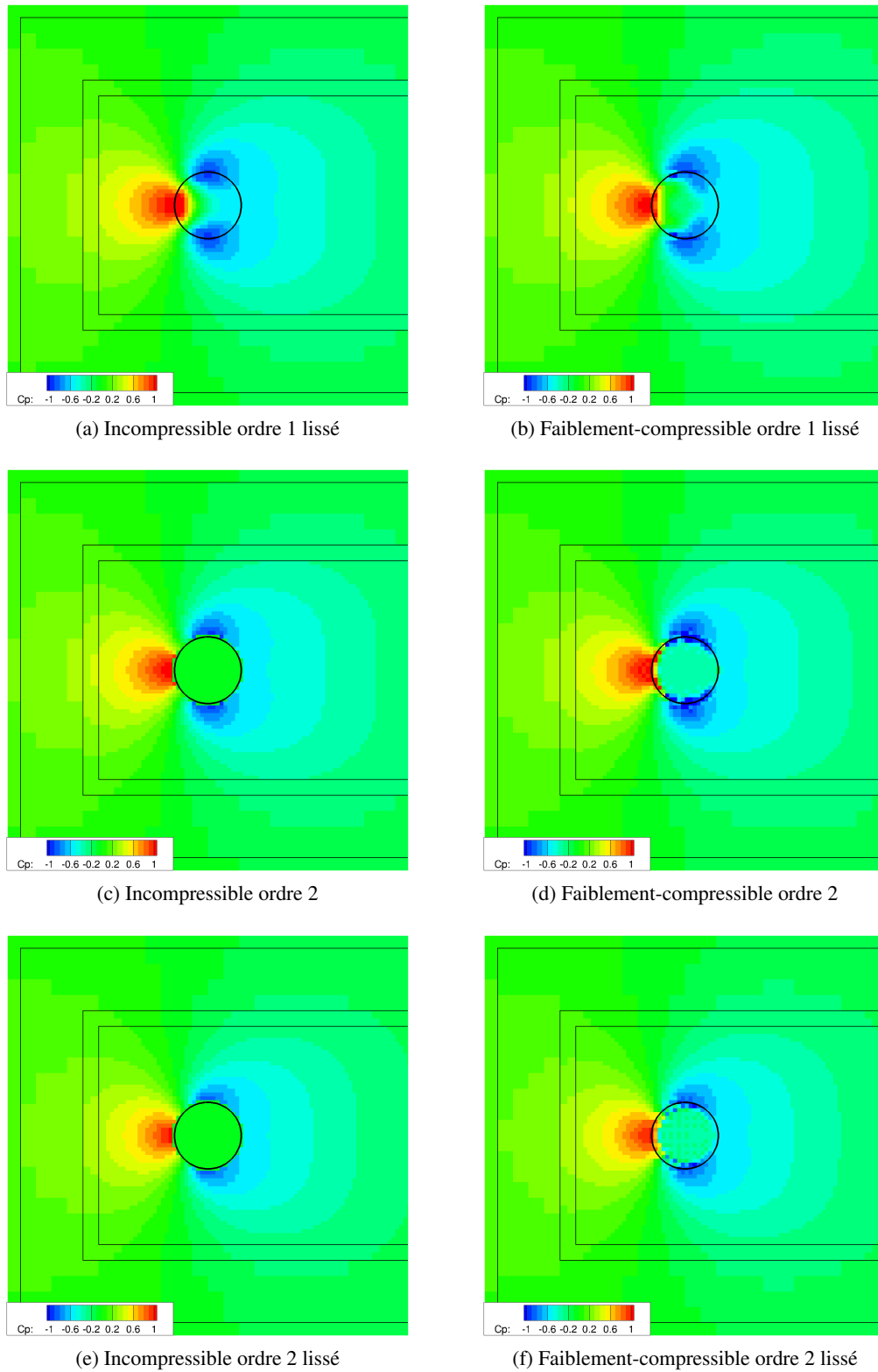


FIGURE 5.21 – Écoulement autour d'un cylindre à $Re = 40$: comparaison des champs de pressions pour les formulations incompressible (gauche) et faiblement-compressible (droite) pour les différentes méthodes de frontière immergée à $\frac{D}{\Delta x} = 17$.

champ de pression sur la frontière solide, la méthode d'ordre 1 lissé s'y prêterait bien mieux que les autres méthodes.

Les profils de vitesse horizontale confirment ces remarques.

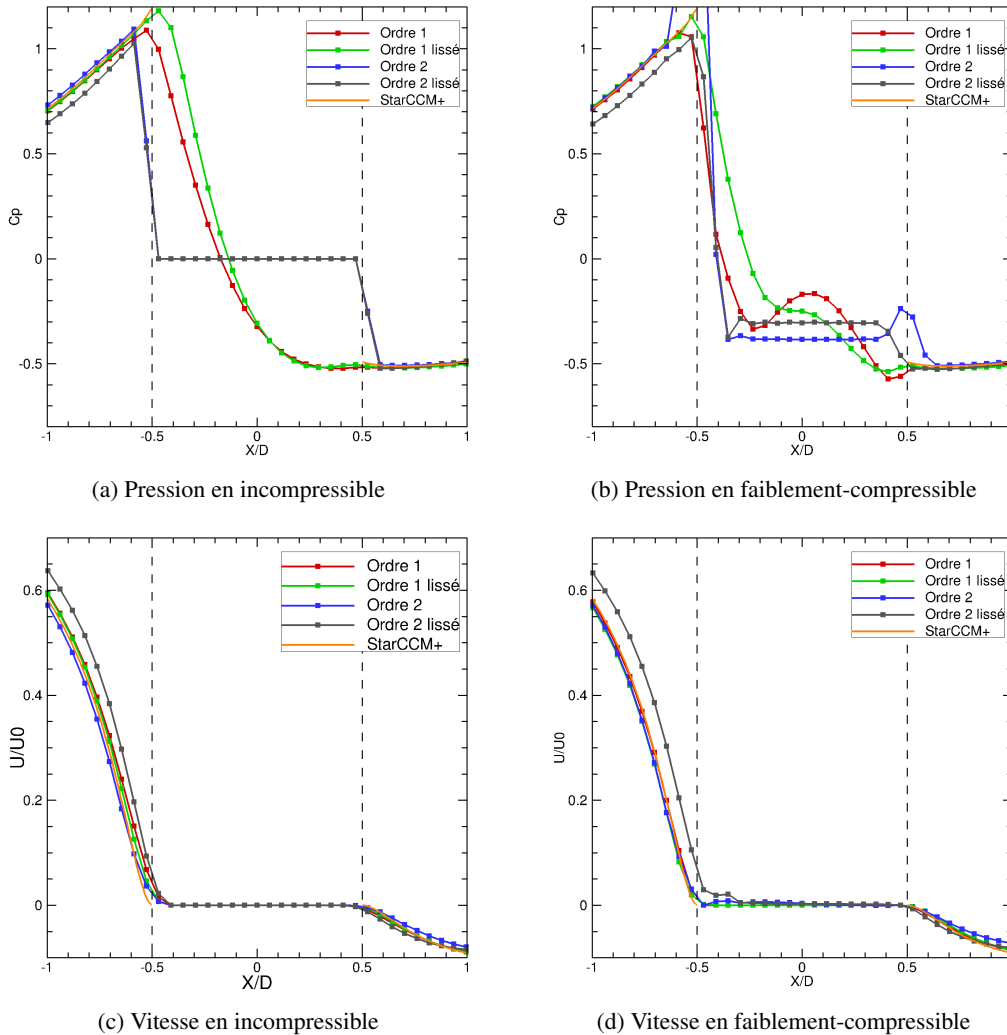


FIGURE 5.22 – Écoulement autour d'un cylindre à $Re = 40$: comparaison des profils de pression (haut) et de vitesse horizontale (bas) sur l'axe horizontal médian en incompressible (gauche) et faiblement-compressible (droite), $\frac{D}{\Delta x} = 17$.

Par ailleurs, afin d'illustrer l'incompatibilité de la contrainte cinématique et de la divergence nulle du champ de vitesse en formulation incompressible, nous traçons sur la figure 5.23 l'incrément de pression.

Nous pouvons constater que dans le cas de la méthode d'ordre 2, l'incrément n'est pas nul en régime stationnaire pour les cellules qui voient leur vitesse u^* fixée à chaque pas de temps. En n'incrémentant pas ces cellules, nous privilégions la contrainte cinématique au détriment de la contrainte de divergence nulle. Ce choix explique peut-être en partie les résultats inférieurs de l'ordre 2.

Nous regroupons ensuite les valeurs de coefficients de traînée dans le tableau 5.1. Le calcul des efforts est effectué via l'approche par volume de contrôle décrite par l'équation (5.32). L'avantage est d'évaluer l'effort de façon identique pour les différents ordres de la méthode de frontière immergée.

A l'exception de l'ordre 2 lissé, les différentes méthodes et formulations donnent une valeur de Cd en

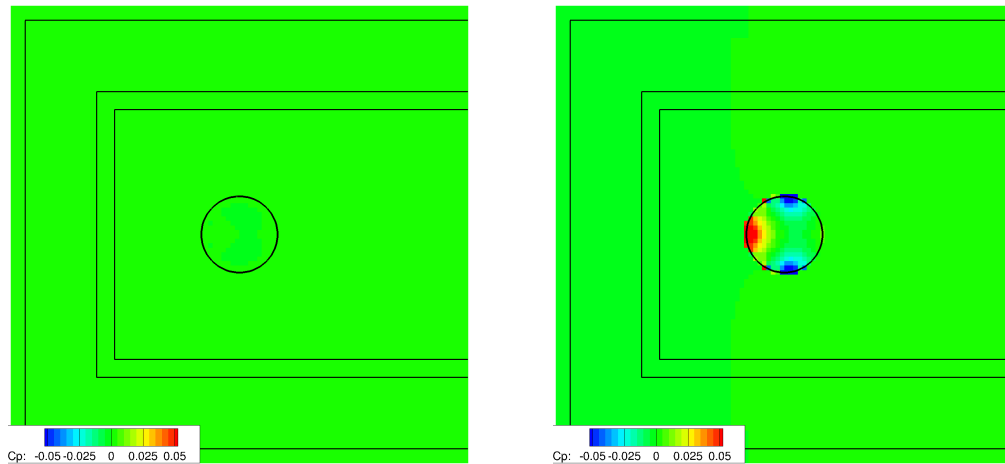


FIGURE 5.23 – Écoulement autour d'un cylindre à $Re = 40$: incrément de pression adimensionné en régime stationnaire pour la méthode d'ordre 1 lissé (gauche) et la méthode d'ordre 2 (droite)

	Cd	
	Incompressible	Faiblement-compressible
Ordre 1	1.54	1.56
Ordre 1 lissé	1.55	1.58
Ordre 2	1.57	1.54
Ordre 2 lissé	1.45	1.46

TABLE 5.1 – Écoulement autour d'un cylindre à $Re = 40$: comparaison des coefficients de traînée pour les différents ordres de frontière immergée

accord avec celles de la littérature, comprises entre 1.5 et 1.6 (cf. étude de l'écoulement laminaire autour d'un cylindre à $Re = 40$ au Chapitre 4). L'écart relatif entre les valeurs est de l'ordre de quelques pourcents.

Au regard de ces différents résultats, l'ordre 2 ne semble pas améliorer de façon nette les résultats par rapport à l'ordre 1. Notons enfin que pour des résolutions spatiales d'un niveau supérieur ($\frac{D}{\Delta x} = 68$), les méthodes d'ordre 1 et 2 donnent des résultats identiques qui se confondent à celui de référence donné par Star-CCM+.

Écoulement autour du profil NACA0010 à $Re=1400$

Pour départager un peu plus les deux méthodes de frontière immergée, nous poursuivons les comparaisons avec un écoulement laminaire (mais instationnaire) autour d'un profil NACA à $Re = 1400$, c'est-à-dire à un nombre de Reynolds plus élevé que dans le cas du cylindre. Nous reprenons encore une fois la configuration décrite au chapitre précédent.

Deux discrétisations ont été retenues : un maillage grossier avec $\frac{c}{\Delta x} = 37$ et un maillage fin $\frac{c}{\Delta x} = 150$ dont on sait qu'il donne de bons résultats en comparaison avec la solution du solveur DVH [Rossi et al., 2018].

La figure 5.24 présente le champ de vorticit  à $\frac{tU_0}{c} = 100$ pour les différents ordres de la méthode en formulations incompressible et faiblement-compressible. En incompressible, la méthode d'ordre 1 présente la physique la plus proche de la solution DVH, avec un détachement de tourbillons dans la direction horizontale et un autre à 30 degrés. En revanche, la méthode d'ordre 2 ne fait pas apparaître le détachement à 30

degrés. L'ordre 2 lissé donne un résultat un peu plus cohérent.

En faiblement-compressible, aucune méthode n'est en mesure de fournir les sillages attendus : la direction et la fréquence des tourbillons ne correspondent pas du tout avec celle de la solution DVH.

La figure 5.25 présente les mêmes résultats que la figure 5.24 mais pour le maillage le plus fin. L'ordre 2 incompressible donne pour cette discrétisation un résultat cohérent avec DVH. Pourtant, la forme de la recirculation et la position des tourbillons semblent mieux prédites par l'ordre 1 lissé que par l'ordre 2. En faiblement-compressible, l'ordre 2 ne fournit toujours pas le bon comportement du sillage alors que l'ordre 2 lissé donne à présent un excellent résultat.

Enfin, la figure 5.26 expose les champs de pression adimensionnée à $\frac{tU_0}{c} = 100$ pour $\frac{c}{\Delta x} = 150$ (une comparaison avec DVH n'est pas possible car la méthode ne permet pas le calcul direct de la pression). Ce qui interpelle le plus est l'irrégularité du champ de pression au voisinage du bord d'attaque en formulation faiblement-compressible.

Pour conclure sur ce cas test, même si notre démarche d'évaluation de la méthode d'ordre 2 a été purement qualitative en comparant des champs de vorticité et de pression, il apparaît que la méthode d'ordre 2 converge paradoxalement moins rapidement que la méthode originale d'ordre 1. Pour les deux méthodes, il semblerait que l'interpolation du champ de vitesse au voisinage de la frontière ne soit pas pleinement compatible avec un champ de pression laissé libre. L'équation de continuité devrait également être modifiée pour améliorer la précision des résultats.

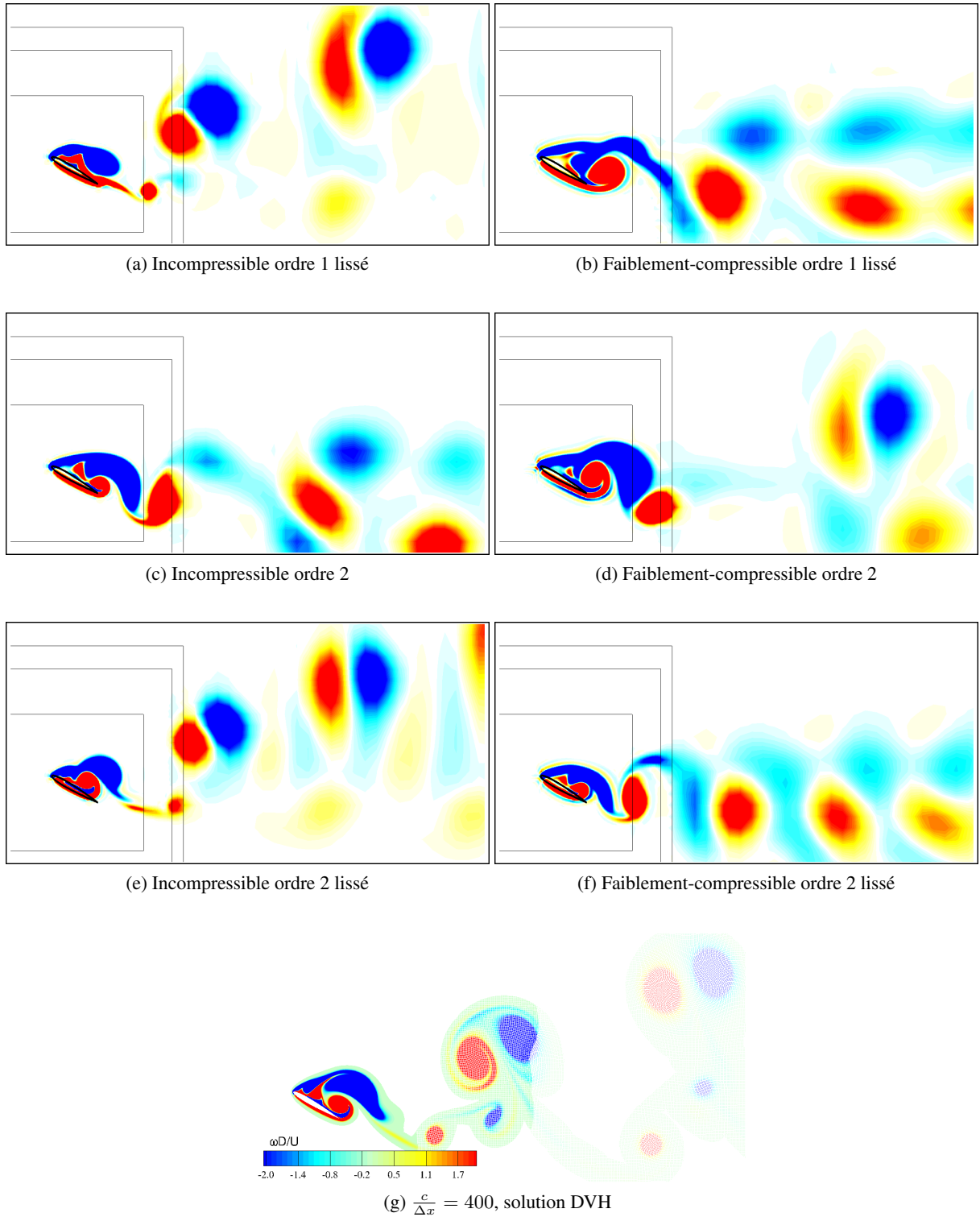


FIGURE 5.24 – Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de vorticité à $\frac{tU_0}{c} = 100$. Comparaison des formulations incompressible (gauche) et faiblement-compressible (droite), $\frac{c}{\Delta x} = 37$, DVH (bas)

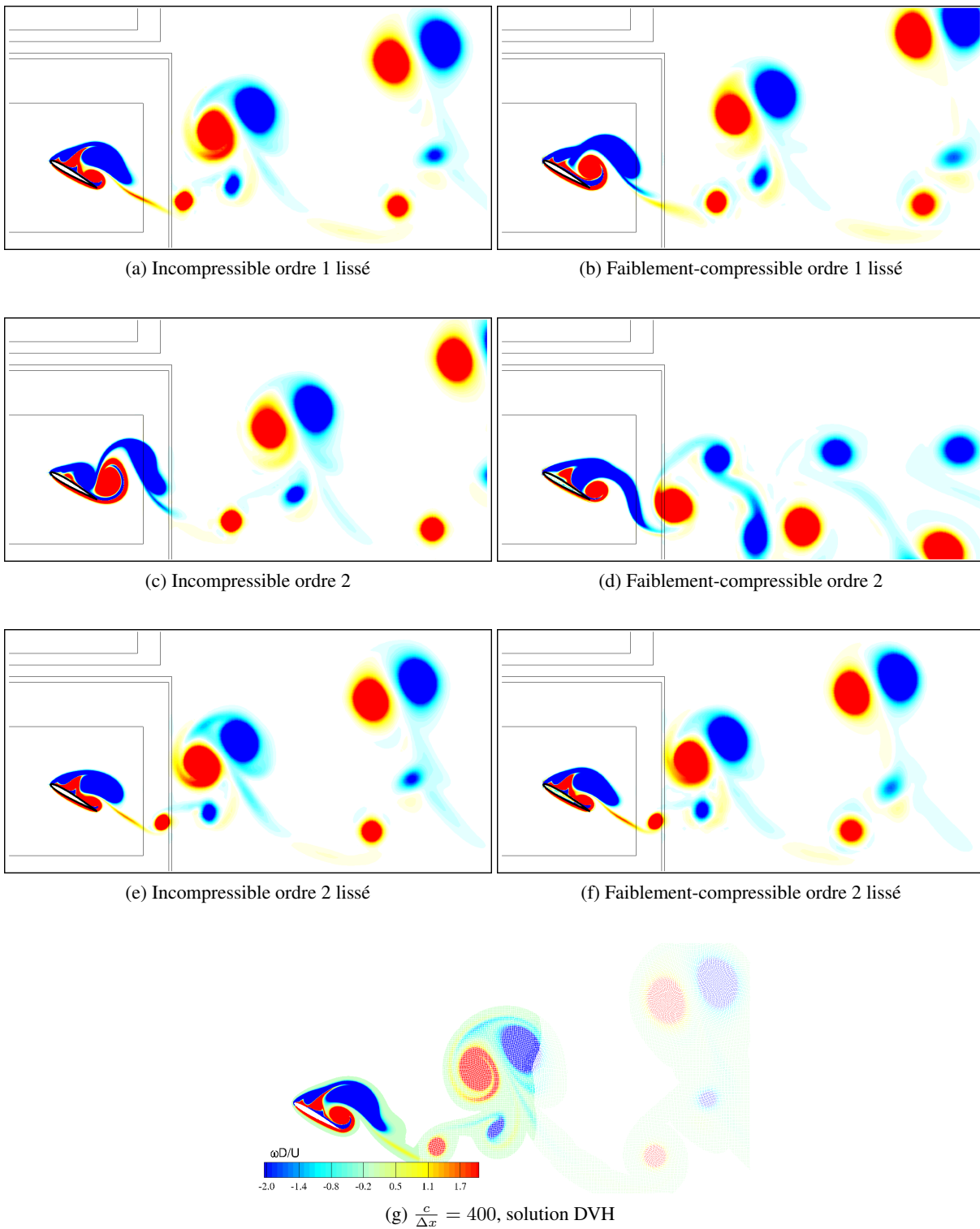


FIGURE 5.25 – Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de vorticité à $\frac{tU_0}{c} = 100$. Comparaison des formulations incompressible (gauche) et faiblement-compressible (droite), $\frac{c}{\Delta x} = 150$, DVH (bas)

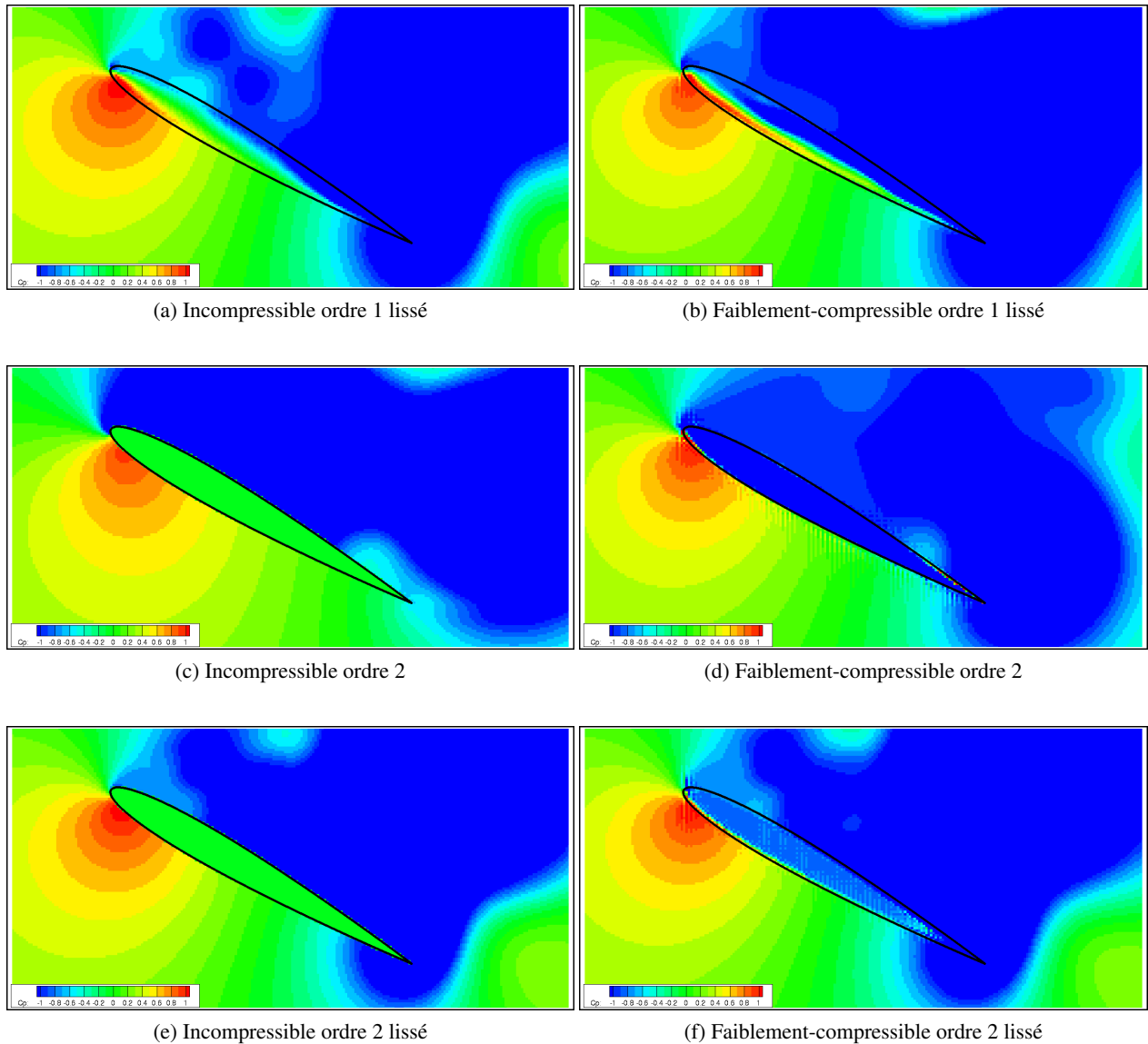


FIGURE 5.26 – Écoulement 2D autour d'un profil NACA à $Re = 1400$. Champ de pression à $\frac{tU_0}{c} = 100$. Comparaison des formulations incompressible (gauche) et faiblement-compressible (droite), $\frac{c}{\Delta x} = 150$

5.6 Conclusions

Dans ce chapitre, nous avons dressé un état de l'art partiel des différentes méthodes de frontière immergée existantes dans la littérature. Cela nous a permis de rattacher la méthode originale utilisée dans le code WCCF aux méthodes de type "discrete forcing" explicites utilisant la pondération du terme source par la fraction de volume solide. Ces méthodes restent assez simples à mettre en œuvre, et trouvent un intérêt pratique pour des simulations impliquant un grand nombre de solides (sédimentation par exemple). Elles sont bien référencées pour des approches incompressibles, mais leur utilisation en compressible reste inconnue.

Grâce à l'étude d'une équation linéaire d'advection-diffusion nous avons pu mettre en évidence que ce type de méthode converge à l'ordre 1 quel que soit l'ordre des opérateurs spatiaux-temporels employés. Par ailleurs, elle introduit une dépendance de la solution au pas de temps pour les écoulements stationnaires.

Afin de limiter ce phénomène, il a été proposé une nouvelle formulation de la fonction Heaviside régularisée qui vient progressivement imposer le terme source.

La volonté de modéliser des écoulements à nombres de Reynolds élevés a motivé une tentative de montée en ordre de la méthode originale. Cette démarche s'est effectuée naturellement en faisant appel à une méthode qui vient forcer explicitement (et a posteriori) la vitesse des cellules du fluide situées dans le voisinage immédiat de la frontière. Ces vitesses sont interpolées à partir des cellules fluides voisines et de la vitesse de l'interface solide, introduisant ainsi une prise en compte exacte de la frontière.

Les résultats obtenus avec la méthode d'ordre 2 ont pu être validés sur un écoulement laminaire autour d'un cylindre à $Re = 40$. Malheureusement, les solutions obtenues sur un écoulement visqueux instationnaire autour d'un profil NACA à $Re = 1400$ ne sont pas à la hauteur des attentes et ne montrent pas une amélioration tangible de la précision des résultats.

En formulation incompressible, la qualité des résultats peut être attribuée à l'incompatibilité des contraintes cinématique et de divergence nulle du champ de vitesse. De plus, l'interpolation effectuée est semblable à une condition de Dirichlet sur la vitesse, dont on a vu qu'elle était responsable d'une couche limite artificielle avec le schéma de projection que nous utilisons.

En formulation faiblement-compressible, la méthode d'ordre 2 manque manifestement de robustesse. Cela se traduit par un champ de pression fortement irrégulier au voisinage de la frontière. Nous pouvons supposer que le forçage de la vitesse, en faisant fi du signe et des valeurs des caractéristiques, détériore la qualité de la solution. De plus, le caractère explicite de la résolution des équations est exigeant et ne "pardonne" pas la moindre imprécision de traitement du système d'ondes.

Des pistes d'amélioration pourront être envisagées à l'avenir. Des investigations plus poussées devront être conduites sur la prise en compte de la pression et son influence sur la qualité des résultats. A travers l'étude conduite ici, il apparaît que l'équation de continuité joue un rôle important et que celle-ci doit sans doute être modifiée au voisinage de la frontière pour accroître la précision et la robustesse de la méthode. En formulation incompressible, la méthode des sources et puits de Kim et al. [Kim et al., 2001] pourrait peut-être contribuer à l'amélioration du traitement de la pression à la traversée de la frontière. En formulation faiblement-compressible, il faudrait s'attarder sur le problème de Riemann au voisinage de l'interface et par exemple le modifier pour introduire plus de diffusion numérique dans l'équation de continuité discrète pour la région du corps immergée. Cela ne devrait que peu dégrader la précision numérique pour cette région de l'écoulement mais favoriserait le lissage du champ de pression. Une autre solution, plus radicale, pourrait consister à impliciter la contribution acoustique de l'écoulement.

Pour clore ce chapitre, même si les résultats que nous avons obtenus ne sont pas encore pleinement satisfaisants, la démarche suivie semble cohérente avec l'objectif visé de simuler des écoulements à haut Reynolds autour de corps mobiles.

Chapitre 6

Implémentation d'une méthode d'immersion de géométries

Ce dernier chapitre est consacré à l'immersion de géométries complexes dans WCCH. Afin d'envisager des écoulements plus complexes et plus proches des besoins industriels, il est nécessaire de disposer d'une procédure "automatique", efficace et robuste pour immerger une géométrie issue d'un logiciel de CAO. Pour appliquer les méthodes de frontières décrites dans le chapitre précédent, la position de chaque centre de volume de contrôle est requise : dans ou hors le solide. Il s'agit du problème intérieur/extérieur. D'autre part, la distance au corps est indispensable, soit pour évaluer la fonction Heaviside lissée de la méthode originale d'ordre 1, soit pour les interpolations de la méthode d'ordre 2. En somme, l'opération se résume au calcul d'une distance signée.

Après une revue des différentes méthodes qui permettent de calculer des distances signées, nous présentons l'intégration de l'une d'entre elle dans le code WCCH. Son cadre d'application se limite à la méthode originelle d'ordre 1. La qualité et l'efficacité de cette méthode sont ensuite évaluées sur un cas complexe d'application : la simulation numérique directe d'écoulement en eau libre autour d'un propulseur de navire en rotation.

6.1	État de l'art	190
6.1.1	Problématique	190
6.1.2	Algorithmes de calcul de distance signée	190
6.1.3	Structure de partitionnement de la géométrie	196
6.1.4	Conclusion et choix d'une méthode	199
6.2	Implémentation de la librairie <i>libOL</i> dans WCCH	200
6.2.1	Description de la librairie <i>libOL</i>	200
6.2.2	Utilisation de <i>libOL</i> et calcul de distance signée dans WCCH	201
6.2.3	Prise en compte des corps mobiles	205
6.3	Applications à un écoulement d'hydrodynamique navale	206
6.3.1	Introduction	206
6.3.2	Paramétrisation du calcul	207
6.3.3	Résultats	209
6.4	Conclusion	215

6.1 État de l'art

6.1.1 Problématique

Pour une méthode de frontière immergée, il reste assez facile de déterminer l'intersection entre la grille volumique sous-jacente et la frontière géométrique, tant que les géométries peuvent être décrites par des formulations analytiques comme par exemple des cylindres ou des sphères. Le repérage dans le maillage volumique des zones intérieure et extérieure à la géométrie ne pose alors pas trop de difficultés. A l'inverse, les géométries plus complexes sont souvent décrites à partir d'une représentation discrète, souvent un maillage surfacique fait de triangles qui offre un bon compromis simplicité/précision : le repérage de la géométrie par rapport au maillage volumique n'a alors plus rien d'évident.

L'objectif poursuivi ici est d'introduire dans le code WCCH une fonctionnalité de calcul de distance signée pour la méthode originale de frontière immergée. Nous considérerons que le maillage surfacique de la géométrie 3D sera représenté selon un format nœud/connectivité. La table des nœuds liste les coordonnées de l'ensemble des sommets de triangles et la table de connectivité définit pour chaque triangle les sommets qui le constituent. Cette représentation spatiale du solide est assez générique et peut être construite par n'importe quel logiciel de CAO incluant un module de maillage en triangles (tesselation en anglais).

Nous présentons dans la suite un certain nombre de techniques et d'algorithmes qui pour beaucoup sont issues de la représentation graphique ou du traitement d'image. En effet le calcul d'une distance signée est par exemple essentiel en visualisation graphique ou dans les jeux vidéo pour afficher des objets avec un rendu qui soit le plus réaliste possible. Il permet d'afficher un objet tel qu'il serait perçu en réalité, avec par exemple des faces cachées du point de vue d'un observateur ou l'ombre produite par un éclairage (rendering/shading en anglais).

Le problème d'immersion de géométries complexes reste souvent peu abordé en détails dans la littérature associée. Il est souvent confié à une librairie tierce. Dans leur étude [Yang et Stern, 2013], Yang et Stern dressent un état des lieux partiel des différentes techniques existantes de calcul de distance et de résolution du problème intérieur/extérieur.

6.1.2 Algorithmes de calcul de distance signée

Nous proposons par la suite une classification qui distingue les méthodes de calcul qui reposent sur la résolution d'une équation Eikonale et d'autres méthodes qui ne s'appuient pas sur ce type d'équation. On peut résumer les approches de la première catégorie comme des méthodes qui parcourent les éléments de la géométrie pour assigner des valeurs de distance au maillage volumique, et celles de la seconde comme des méthodes qui parcourent le maillage volumique et cherchent une distance par rapport à la géométrie.

Résolution d'une équation Eikonale

Ce type de méthode s'appuie sur le formalisme "level-set" et l'équation Eikonale. Cette dernière est une équation fondamentale qui régit la propagation de certains phénomènes comme la lumière dans un milieu. Elle s'écrit :

$$|\nabla\phi(\mathbf{x})| = \frac{1}{f(\mathbf{x})}, \mathbf{x} \in \Omega \quad (6.1)$$

avec $\phi(\mathbf{x})|_{\partial\Omega} = 0$ et f une fonction dépendant de la nature physique du problème traité (propagation d'onde, électromagnétisme, etc.). Physiquement, $\phi(\mathbf{x})$ est le temps le plus court nécessaire au parcours de la frontière $\partial\Omega$ jusqu'au point \mathbf{x} et f est la vitesse de propagation du phénomène. Dans le cas où $f = 1$, la solution est la distance signée à $\partial\Omega$. Dans notre contexte d'application, $\phi(\mathbf{x})$ désigne la distance signée.

Fast marching method (FMM) La méthode FMM a été introduite par Sethian [Sethian, 1996] pour résoudre numériquement l'équation (6.1) par la méthode des caractéristiques. Son principe repose sur la propagation de front. L'information de distance est propagée de proche en proche à partir de la géométrie puis jusqu'à couvrir l'ensemble du domaine de calcul.

En pratique, tous les points du maillage fluide sont répartis en trois catégories :

- *Frozen points* : points figés pour lesquels la distance signée a déjà été calculée. Ces points sont inclus dans le front.
- *Narrow band* : points situés en amont du front, au voisinage des points déjà calculés.
- *Far away* : les autres points.

Connaissant la position du front, une distance à ce front est calculée pour tous les points *Narrow band*. Cette distance est évaluée à partir d'une discrétisation de l'équation Eikonale. Le gradient est en général discrétisé à partir d'une expression décentrée qui s'appuie sur les *Frozen point*. Parmi tous ces points *Narrow band*, celui qui est le plus proche du front à une distance d devient un *Frozen points*. Tous ses voisins, qui sont catégorisés *far away*, appartiennent désormais à la *Narrow band*. Le front est ensuite avancé d'une distance d . L'algorithme est ensuite répété. La figure 6.1 illustre le principe de la méthode.

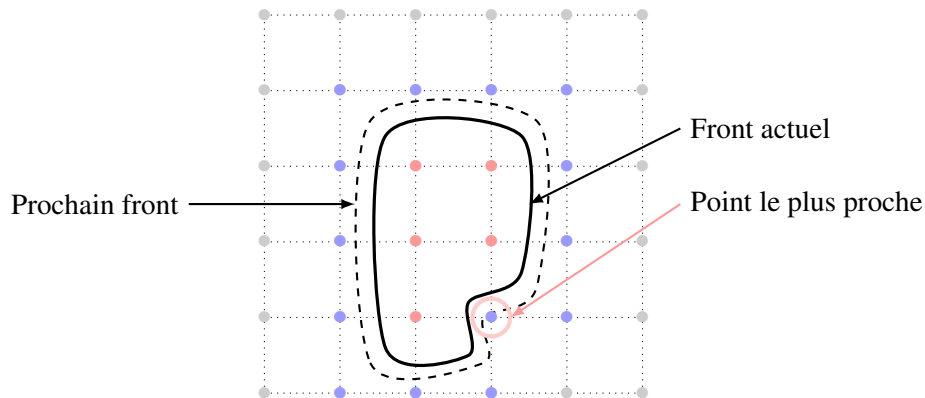


FIGURE 6.1 – Illustration du principe de la fast marching method : points *Far away* (●), points *Narrow band* (●), points *Frozen points* (●)

Pour réduire le temps de recherche du point minimum, un algorithme de tri par tas réduit la complexité algorithmique du processus en $O(\log(N_{NB}))$, avec N_{NB} le nombre de points de *Narrow band*.

D'un point de vue pratique, la mise en œuvre de cette méthode suppose déjà que le front soit initialisé. Ceci pourrait être fait en sélectionnant toutes les cellules volumiques qui intersectent le maillage. On pourrait imaginer parcourir tous les éléments géométriques du maillage surfacique (sommets, arêtes, triangles) et chercher à savoir quelle cellule volumique les contient en les étiquetant. Avec le repérage en code de Morton, cela ne pose pas trop de difficultés mais encore faut-il s'assurer que les cellules volumiques étiquetées forment une surface fermée.

Par ailleurs, la procédure globale doit être appelée deux fois : une pour les distances positives et une pour les distances négatives, autrement dit une pour l'extérieur et une pour l'intérieur du corps.

Méthode Closest Point Transform (CPT) Cette méthode proposée par Mauch [Mauch, 2000] résout aussi une équation Eikonale par la méthode des caractéristiques. Elle est implémentée efficacement avec une méthode de rasterisation ou matrialisation (scan conversion en anglais). Le principe de la méthode CPT est de définir des polygones créés par extrusion des éléments géométriques dans la direction normale au corps. En 2D, les principaux éléments d'une géométrie sont le segment et le point. Pour le premier, son extrusion dans la direction normale au corps crée un rectangle de côté $l \times L$ avec l la longueur du segment

et L la hauteur d'extrusion. Pour un point, la figure générée est un cône dont le sommet principal est le point lui-même à partir duquel on peut définir des demi-droites orientées suivant les vecteurs normaux des deux segments qui partagent le point en commun (Fig. 6.2). L'extrusion est définie vers l'extérieur et l'intérieur du corps.

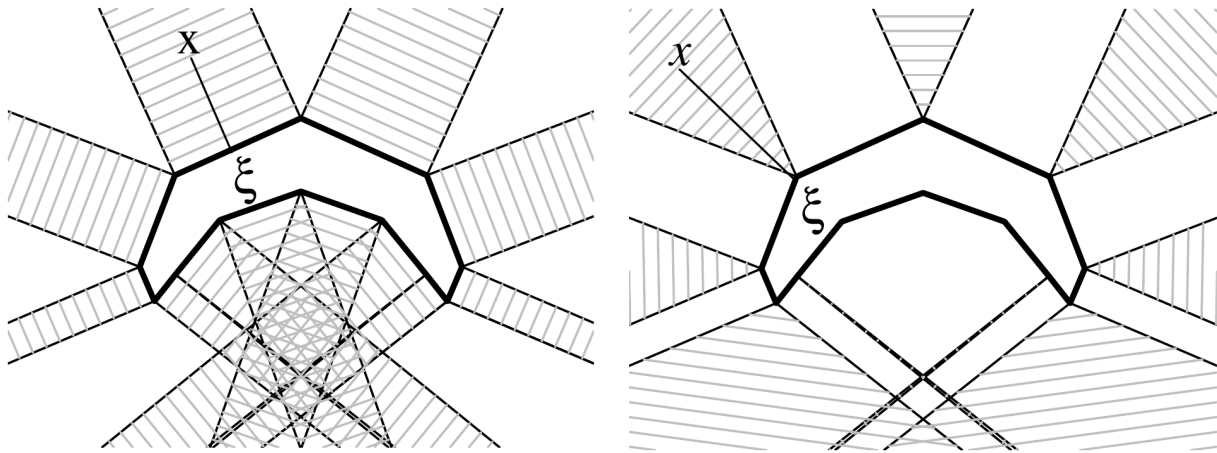


FIGURE 6.2 – Rectangles et cônes définis par extrusion de la géométrie en 2D, extrait de [Mauch, 2000]

EN 3D, des polyèdres sont générés par extrusion des faces triangulaires, des arêtes et des sommets (Fig. 6.3).

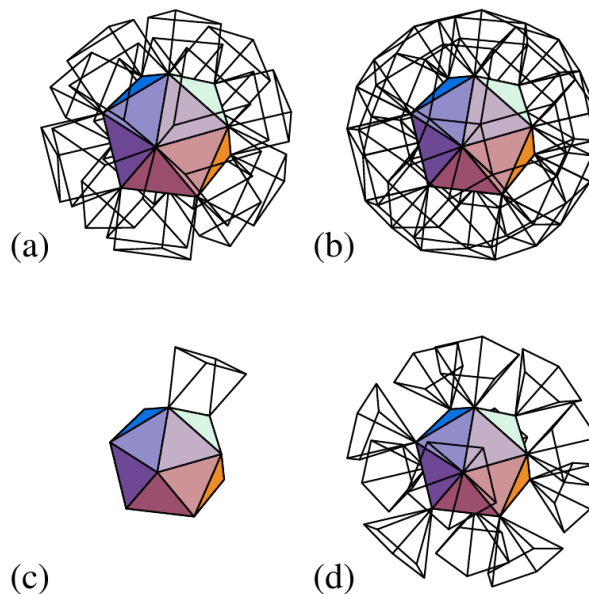


FIGURE 6.3 – (a) Polyèdres des faces triangulaires, (b) polyèdres des arêtes, (c) polyèdre d'une arête, (d) polyèdres des sommets, extrait de [Mauch, 2000]

Par ailleurs, tout point de l'espace appartient à au moins une figure géométrique générée par le procédé d'extrusion. Calculer la distance d'un point à la géométrie est alors simple. Une fois qu'un point est rattaché à un élément géométrique en identifiant son appartenance à une des figures extrudées, la distance est directement obtenue en calculant une distance à un point, un segment ou une face. Par exemple en 2D, si un point

appartient à un cône extrudé, la distance du point à la géométrie est la distance du point au sommet du cône. S'il appartient à un rectangle d'extrusion, la distance est évaluée en projetant orthogonalement le point sur le segment extrudé.

Enfin pour déterminer l'appartenance d'un point à un polygone ou polyèdre extrudé, Mauch propose d'utiliser une méthode de rasterisation. Considérons un polygone convexe et une grille cartésienne. La méthode de rasterisation permet de déterminer quels points de la grille cartésienne appartiennent au polygone en intersectant chaque ligne de la grille avec les côtés du polygone (cf. Fig. 6.4). La rasterisation suppose donc le recours à une grille cartésienne.

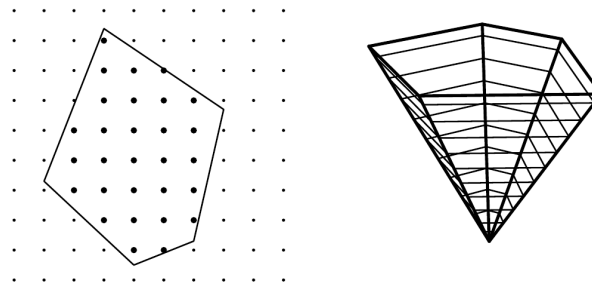


FIGURE 6.4 – Principe de la rasterisation (ou scan conversion en anglais), extrait de [Mauch, 2000]

En pratique, la méthode CPT génère d'abord les polygones d'extrusion. Puis étant donné une grille cartésienne englobant la géométrie, elle identifie pour chaque polygone d'extrusion les points de la grille qu'il contient. Pour tous ces points, la distance est ensuite calculée simplement. La méthode CPT tient son efficacité de la méthode de rasterisation qui s'applique de façon simple pour les polygones extrudés.

En résumé, les méthodes de calculs de distance signée par résolution d'une équation Eikonale semblent efficaces et robustes. Un avantage certain de ces techniques est de pouvoir limiter la distance jusqu'à laquelle nous souhaitons les appliquer. Finalement puisque le calcul de la distance n'est nécessaire qu'au voisinage du corps, limiter le calcul à cette zone permet d'économiser du temps de calcul.

Néanmoins, le principal inconvénient de ces techniques est de supposer que l'on souhaite calculer des distances sur une grille cartésienne. Ceci s'explique par le fait qu'elles ont été, pour la plupart, développées pour le traitement numérique des images dont les pixels sont nécessairement organisés sur une grille cartésienne. Pour un maillage volumique à raffinement local comme celui de WCCH, ces méthodes semblent difficilement applicables même si nous pouvons envisager par exemple une utilisation du code de Morton pour tenter de maintenir l'efficacité de la rasterisation pour la méthode CPT.

Méthodes de calcul du maillage fluide vers la géométrie

La résolution d'une équation Eikonale ne constitue pas la seule possibilité de calcul de distance signée. Une autre approche est de partir du maillage volumique fluide et d'aller vers la géométrie. L'approche classique consiste alors à décomposer le calcul de la distance signée en deux étapes : le calcul de la distance puis de son signe dans un deuxième temps.

Calcul de distance absolue Pour le calcul de la distance, une méthode naïve consisterait à parcourir l'ensemble des éléments géométriques pour déterminer lequel est le plus proche. Pour les maillages surfaciques que nous envisageons, l'élément renvoyé comme étant le plus proche sera un triangle. La complexité algorithmique d'une telle approche est en $O(N_{fluide}N_{geo})$ avec N_{fluide} le nombre de points du maillage fluide pour lesquels nous voulons calculer une distance et N_{geo} le nombre d'éléments géométriques du maillage surfacique. En pratique cette technique n'est pas retenue et présente peu d'intérêt avec N_{fluide}

et $N_{geo} \sim 10^6$, surtout s'il est envisagé de l'appeler régulièrement comme à chaque pas de temps. Des structures de données optimisées sont en général employées pour stocker les éléments de la géométrie et y accéder rapidement. Le calcul d'une distance est alors possible sans avoir à parcourir l'ensemble des éléments géométriques du solide. Ces structures seront détaillées par la suite.

La deuxième étape est donc la détermination du signe. Plus simplement, il s'agit de savoir si un point de l'espace se situe dans ou en dehors de la géométrie.

Calcul de signe par les vecteurs normaux à la géométrie La manière la plus simple de procéder au calcul du signe est de le faire au moyen de produits scalaires avec les vecteurs normaux de l'élément géométrique le plus proche. Pour cela, il faut projeter le point M duquel la distance est calculée sur le plan contenant le triangle. Le projeté M' peut se retrouver dans trois configurations différentes : directement dans le triangle, face à une arête ou face à un sommet (Fig. 6.5). Ces trois configurations correspondent en fait aux trois types de polygones extrudés qui peuvent être construits pour l'application de la méthode CPT précédemment énoncée.

- Si M' tombe dans le triangle alors le calcul du signe est immédiat et donné par le signe du produit scalaire $M'M \cdot n$, n désignant le vecteur normal au triangle.
- Si M' est face à une arête, le point M est à nouveau projeté, mais sur l'arête. Si ce point projeté, noté M'' , se situe entre les deux sommets de l'arête alors le signe peut être déterminé par le signe du produit scalaire $M''M \cdot n$. Le vecteur n désigne alors le vecteur normal à l'arête obtenu en moyennant les vecteurs normaux des deux triangles partageant l'arête. Si le point M'' se trouve en dehors de l'arête, alors le problème se ramène à la dernière configuration qui est celle d'un point projeté face à un sommet.
- Si M' est face à un sommet alors la plus robuste des façons de procéder est d'utiliser la méthode *Angle Weighted Pseudo-normal* proposée par Bærentzen et Aanaes [Bærentzen et Aanaes, 2005].

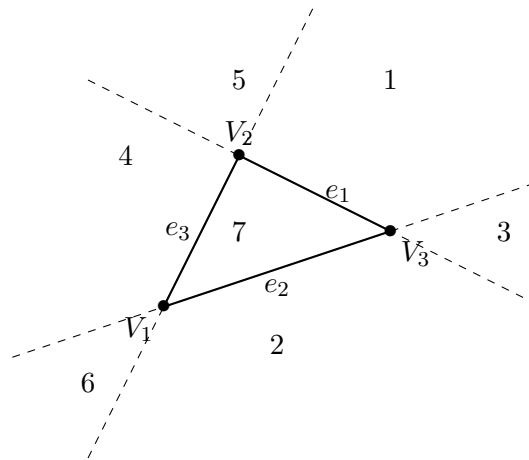


FIGURE 6.5 – Configurations possibles de la projection d'un point sur la surface d'un triangle pour calculer le signe

La méthode *Angle Weighted Pseudo-normal* consiste à définir pour chaque sommet du maillage triangulaire un vecteur normal calculé par :

$$n_\alpha = \sum_i \alpha_i n_i \quad (6.2)$$

n_i étant les vecteurs normaux des triangles partageant le sommet en commun. α_i est l'angle d'incidence de chaque sommet (Fig. 6.6).

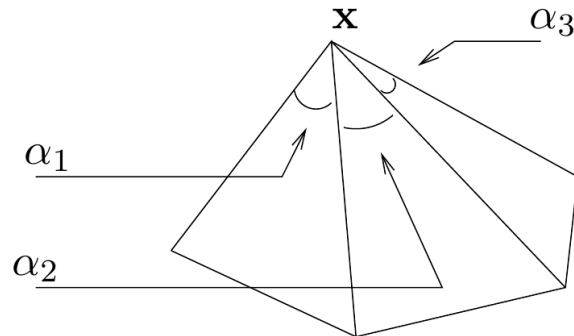


FIGURE 6.6 – Angle d’incidence de chaque sommet dans la méthode *Angle Weighted Pseudo-normal*, extrait de [Bærentzen et Aanaes, 2005]

Ensuite, le signe du produit scalaire entre le vecteur pointant du sommet vers le point M et le vecteur \mathbf{n}_α permet de conclure sur la position de M par rapport au corps. Bærentzen et Aanaes ont montré que leur méthode est exacte.

La technique présentée ici a l’avantage d’avoir un coût algorithmique insignifiant.

Cependant, elle peut être rapidement mise en échec si le maillage surfacique contient des triangles qui s’intersectent. Ce problème est produit à la génération du maillage surfacique à partir de la CAO. Si la taille cible des triangles n’est pas suffisamment petite, des triangles peuvent s’intersecter comme illustré sur la figure 6.7.

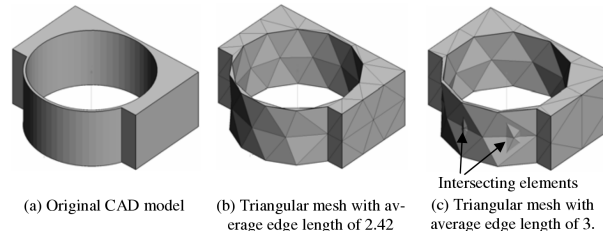


FIGURE 6.7 – Exemple de triangles qui s’intersectent (ou *self-intersected triangles* en anglais), extrait de [Yamakawa et Shimada, 2009]

Le phénomène a une forte probabilité d’occurrence au niveau des bords minces (coques) où les faces interne et externe peuvent se couper comme sur la figure 6.7. Parfois, la mauvaise qualité de la CAO et des courbes qui la définissent (courbes de Bézier ou NURBS) peuvent aussi en être responsables.

Si un triangle identifié comme étant le plus proche élément présente ce problème et appartient à une protubérance comme celle de la figure 6.7, alors le signe calculé par le produit scalaire avec le vecteur normal sera erroné.

Enfin, pour des problèmes de frontière immergée, la méthode *Angle Weighted Pseudo-normal* est par exemple mise en œuvre dans [Yang et al., 2015] et bien détaillée dans [Choi et al., 2007].

Calcul du signe par la méthode *generalized winding number* Une autre approche intéressante pour savoir si un point est dans ou hors d’une géométrie fermée est celle décrite dans [Jacobson et al., 2013], et fait appel à la notion mathématique d’indice ou *generalized winding number* en anglais. Le principe est simple et repose sur le calcul de ce nombre par (Fig. 6.8) :

$$w(\mathbf{p}) = \frac{1}{2\pi} \sum_{i=1}^n \theta_i \quad (6.3)$$

ou en 3D :

$$w(\mathbf{p}) = \sum_{f=1}^n \frac{1}{4\pi} \Omega_f(\mathbf{p}) \quad (6.4)$$

où n est le nombre de segments (en 2D) ou de faces triangulaires (3D), θ_i est l'angle signé entre deux sommet consécutifs en 2D et Ω_f est l'angle solide de la surface d'un triangle en 3D.

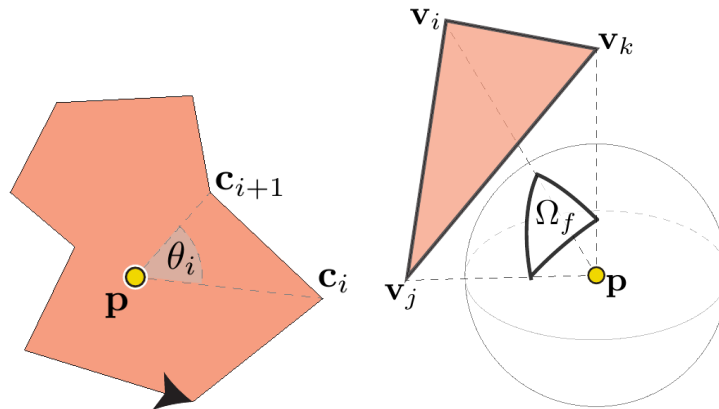


FIGURE 6.8 – Principe du calcul de l'indice *generalized winding number* en 2D et 3D, extrait de [Jacobson et al., 2013]

Si l'indice vaut 0 alors le point p est hors de la géométrie, s'il vaut 1, il est dedans. L'avantage d'une telle approche est sa robustesse pour gérer des surfaces non-connexes et ouvertes. Elle donne de bons résultats même si la géométrie présente des défauts comme des trous ou des triangles retournés. Sachant que la génération d'une triangulation "propre" de la géométrie à partir d'une CAO imparfaite est un problème parfois épineux, et qui demande du temps ingénieur, la méthode de Jacobson peut être intéressante de ce point de vue. Toutefois, l'indice est calculé en sommant les contributions de chaque triangle, ce qui peut rendre le coût de calcul rapidement limitant.

Calcul du signe par lancer de rayon Une méthode répandue en infographie pour savoir si un point se trouve à l'intérieur ou à l'extérieur d'une frontière consiste à tracer une demi-droite (rayon) issue du point en question puis à compter le nombre de fois qu'elle intersecte la géométrie. Si ce nombre est pair alors le point se trouve à l'extérieur, s'il est impair, le point est à l'intérieur (voir Fig. 6.9).

Cette approche est plus coûteuse que la première approche présentée, qui s'appuie sur le signe de produits scalaires. Son application par force brute est tout aussi coûteuse que la méthode de Jacobson puisque chaque triangle est testé pour savoir s'il est traversé par le rayon. Cependant, il est possible de réduire drastiquement ce coût si elle est employée avec une structure de partitionnement de la géométrie. Par ailleurs, elle est plus robuste que la première méthode car elle n'est pas mise en défaut si la géométrie présente des triangles qui s'intersectent. Enfin il est nécessaire d'appliquer une procédure particulière si le rayon intersecte une arête ou le sommet d'un triangle pour ne pas compter plusieurs fois le point d'intersection.

6.1.3 Structure de partitionnement de la géométrie

Nous avons vu que le calcul d'une distance par les méthodes autres que la résolution d'une équation Eikonale nécessitent de parcourir tous les éléments géométriques qui constituent le maillage surfacique du solide. Pour éviter cela, des structures de partitionnement de la géométrie permettent de ranger les éléments

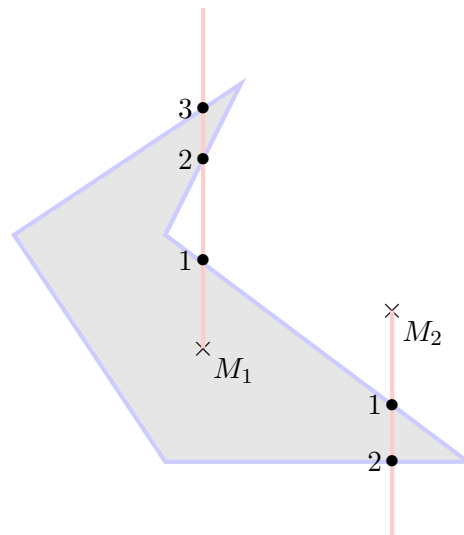


FIGURE 6.9 – Test intérieur/extérieur par comptage des intersections de rayon

géométriques dans une structure de données, souvent des arbres, pour accéder facilement aux éléments. Elles obéissent au principe "diviser pour régner". Ces structures accélératrices renseignent également sur la proximité des éléments. Pour le calcul d'une distance, elles permettent d'exclure rapidement les éléments qui sont trop éloignés. La classification des structures présentée par la suite s'inspire de [Doran, 2013].

Hierarchie de volumes englobants

La hiérarchie de volumes englobants (ou BVH en anglais pour Bounding Volume Hierarchy) est une structure arborescente d'éléments géométriques dont les nœuds représentent le volume occupé par un sous-ensemble d'éléments géométriques. Les nœuds stockent le volume englobant alors que les feuilles, c'est-à-dire les nœuds sans descendant, contiennent la liste des éléments géométriques présents dans leur volume englobant. Elles peuvent contenir un à plusieurs centaines de triangles (tout dépend de la profondeur de l'arbre). Enfin, le volume englobant d'un nœud est également un englobant de ses "fils" (Fig. 6.10). L'arbre peut être construit à partir de volumes englobants cubiques, sphériques, etc. Les volumes englobants peuvent être alignés aux axes (Axis-Aligned Bounding Boxes – AABB) ou orientés par rapport aux éléments géométriques (Oriented Bounding Boxes – OBB).

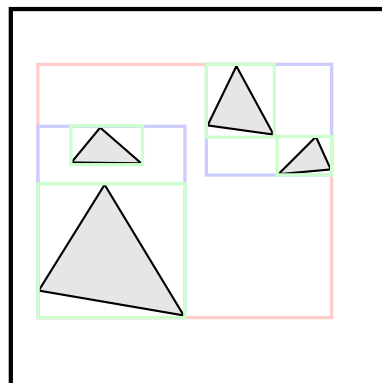


FIGURE 6.10 – Partitionnement par volume englobant avec des AABB

La structure BVH est principalement exploitée pour les rendus graphiques. Elle supporte efficacement

la détection de collisions entre objets et la méthode de lancer de rayon.

Grille cartésienne

Le principe de la grille cartésienne est simple et réside dans le partitionnement de l'espace en cellules selon une division régulière, ou non (Fig. 6.11). Les éléments géométriques sont rangés dans les cellules. L'avantage de cette structure est d'avoir un coût algorithmique de construction relativement faible et le déplacement dans la grille est très simple. Cependant, n'étant pas hiérarchique, le coût de recherche et de parcours des éléments est bien plus élevé que pour les autres structures de partitionnement.

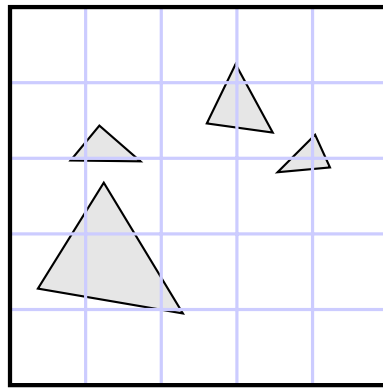


FIGURE 6.11 – Partitionnement par grille cartésienne

Octree

Sans rappeler la définition d'un octree qui a déjà été donnée dans la Section 1.2.5, son utilisation dans le contexte de la partition de géométries revient à stocker les éléments géométriques dans les feuilles de l'arbre (Fig. 6.12). Partant d'une géométrie, l'octree est subdivisé de manière récursive. La subdivision est arrêtée quand chaque nœud ne contient qu'un nombre limité de triangles. De plus le coût de parcours de l'arbre est globalement constant. Le risque avec cette structure est que la profondeur de l'arbre (i.e. le nombre de niveaux de récursion) peut devenir très importante et déséquilibrer l'arbre si la répartition des triangles n'est pas uniforme. Le parcours de l'arbre sera alors ralenti.

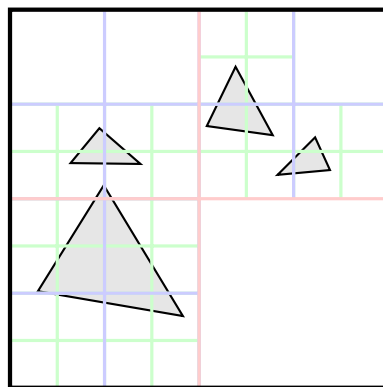


FIGURE 6.12 – Partitionnement par octree

Kd-tree

Enfin le kd-tree résout ce problème d'équilibrage. Les arbres kd sont des arbres binaires dans lesquels chaque nœud contient un point en dimension k . Ces points divisent l'espace en deux demi-espaces. Les points situés dans le demi-espace de gauche (respectivement de droite) sont stockés dans la branche gauche (respectivement droite) du nœud courant. Par exemple, si un nœud particulier divise l'espace selon un plan normal à la direction x , tous les points de coordonnée x inférieure à la coordonnée du point associé au nœud seront stockés dans la branche gauche du nœud. De manière similaire, les points de coordonnée x supérieure à celle du point considéré seront stockés dans la branche droite du nœud. La façon de produire les subdivisions assure donc un partage des éléments géométriques optimal entre les différents nœuds.

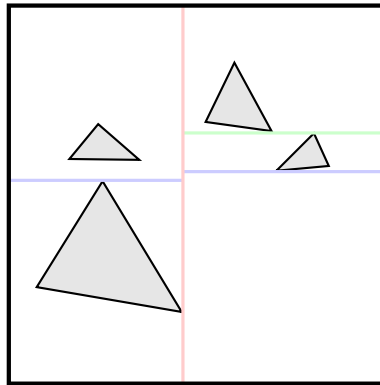


FIGURE 6.13 – Partitionnement par kd-tree

6.1.4 Conclusion et choix d'une méthode

Eu égard à cette succincte revue des méthodes de calcul de distance signée, il a été choisi de mettre de côté les méthodes basées sur la résolution de l'équation Eikonale. La principale raison est leur difficile application sur un maillage volumique à raffinement local comme dans WCCH. De plus, l'effort à fournir pour paralléliser ces algorithmes paraît non négligeable. L'avantage de la deuxième catégorie de méthodes, qui ne sont donc pas basées sur cette approche Eikonale, est qu'elles sont beaucoup plus générales et moins intrusives sur le maillage volumique. Il est tout à fait possible d'imaginer que les tâches de calcul de distance et de signe soient confiées à une librairie tierce indépendante du solveur fluide.

En ce qui concerne le calcul du signe, l'approche basée sur le calcul d'indice *generalized winding number*, permet certes de traiter des géométries dégradées et imparfaites, mais nécessite un parcours complet des éléments géométriques du solide sans recours possible aux structures de partitionnement. Nous avons sélectionné en première approche la méthode basée sur le signe des produits scalaires avec les vecteurs normaux de la géométrie (qui utilise la méthode de Bærentzen et Aanaes). Toutefois, certaines difficultés rencontrées avec des géométries comportant des triangles intersectés nous ont conduits à développer aussi une méthode de lancer de rayon.

Pour le calcul de la distance absolue, il va de soi qu'une structure de partitionnement est indispensable pour éviter le parcours de l'ensemble des éléments géométriques. Cette structure est comparable, de façon imagée, à un ensemble de tiroirs de rangement des éléments de la géométrie dans lesquels il est ensuite facile de trouver les éléments les plus proches par exemple. Nous avons éliminé la grille cartésienne qui ne répond pas aux exigences de rapidité de recherche. Il reste les structures hiérarchiques BVH, octree et kd-tree. D'un côté l'octree est relativement simple à générer mais peut se trouver grandement déséquilibré si la géométrie n'est pas homogène et qu'un grand nombre de triangles se trouvent concentrés dans un coin de l'espace. D'un autre côté, la flexibilité des BVH et kd-tree leur permet d'éviter ce problème mais encore

faut-il construire ces arbres de façon optimisée pour atteindre l'équilibrage idéal. Finalement nous avons sélectionné l'octree, notamment parce que nous disposons déjà d'une certaine expérience sur leur utilisation dans le code WCCH.

D'un point de vue pratique, il serait possible d'utiliser directement l'octree de WCCH pour stocker les éléments géométriques. Toutefois, cela conduirait à une incompatibilité entre un octree originellement conçu pour s'adapter à la dynamique de l'écoulement et un octree destiné à recueillir les éléments géométriques. Surtout que l'octree du maillage surfacique n'a pas d'exigence de différence de niveau alors que l'octree de WCCH doit respecter la règle $2:1$, c'est-à-dire que chaque bloc ne peut avoir plus d'un niveau de différence avec son voisin.

Enfin plutôt que de développer nous-mêmes un algorithme de partitionnement géométrique par octree, nous avons fait le choix d'utiliser une librairie tierce. Des bibliothèques "open-source" sont disponibles en ligne pour la gestion de maillages en mécanique des fluides comme par exemple, GTS développée pour le solveur Gerris [Popinet, 2000], Cassiopée de l'ONERA [Benoit et al., 2015], etc. D'autres sont plus généralistes comme CGAL (Computational Geometry Algorithms Library) [The CGAL Project, 2018] et donc beaucoup plus complètes. Notre choix s'est finalement porté sur *libOL*, développée par Loïc Maréchal [Marechal, 2016] pour sa concision et relative simplicité de mise en œuvre (un interfaçage avec le Fortran étant déjà en place).

Dans la section suivante nous décrivons l'intégration de *libOL* dans WCCH.

6.2 Implémentation de la librairie *libOL* dans WCCH

6.2.1 Description de la librairie *libOL*

libOL est une librairie dont le but est de générer une structure octree à partir d'un maillage surfacique constitué de triangles. Elle fournit également un certain nombre d'opérations telles que la recherche du triangle le plus proche d'un point donné de l'espace ou bien la localisation des triangles intersectant une région donnée.

Pour utiliser *libOL*, la donnée à fournir est un maillage surfacique au format nœud/connectivité. A partir de cela la librairie construit un octree en suivant la procédure suivante :

1. Un octant *root* est créé comme étant la boîte englobante (bounding box) de la géométrie.
2. La liste des entités (sommets, arêtes ou triangles) est parcourue pour construire l'octree. Si un sommet ou un triangle est contenu dans une feuille alors il y est ajouté. Si le nombre d'entités contenu dans une feuille dépasse 20 alors la feuille est subdivisée en 8 enfants (en 3D) qui se répartissent les entités de leur parent. Les entités sont ensuite chaînées entre elles pour pouvoir les parcourir facilement par la suite. Cette construction nécessite le calcul de l'intersection entre un parallélépipède et un triangle. Dans *libOL*, cette intersection est réalisée de manière exacte. Autrement dit soit l'un des sommets du triangle se situe dans le parallélépipède, soit une des arêtes du triangle intersecte une face, soit le plan du triangle intersecte une arête. Certes, cela demande plus de ressources CPU mais le calcul est rigoureux. En conséquence un triangle appartient à tous les octants qu'il intersecte et est dupliqué autant de fois.
3. Le niveau maximum atteint par l'arbre est limité à 10. Ainsi une feuille située au niveau 10 peut contenir plus de 20 entités. Ce choix est arbitraire mais limite la profondeur de l'arbre.

Pour accélérer l'accès aux octants de l'arbre, une grille cartésienne uniforme de taille N_{buc}^3 est créée. La taille de cette grille est définie par :

$$N_{\text{buc}} = 2^{\text{MinLvl}} \quad (6.5)$$

avec *MinLvl* calculé selon :

$$\text{MinLvl} = \max\left(\frac{\log(N_{\text{entités}}/100)}{\log(8)}, 3\right) \quad (6.6)$$

$\frac{\log(N_{\text{entités}}/100)}{\log(8)}$ définit le niveau de la taille de la grille qui serait nécessaire pour stocker 100 entités par cellule de la grille en supposant une répartition homogène du nombre total d'entités $N_{\text{entités}}$. Chaque élément de la grille accélératrice pointe vers l'octant de niveau *MinLvl* de l'arbre.

Cette gestion des données permet d'accélérer les accès mémoire dans l'arbre, en particulier à l'initialisation d'une procédure de parcours de l'arbre. Par exemple, pour chercher la distance à la géométrie en un point de l'espace, ce point est d'abord repéré dans la grille accélératrice puis positionné dans l'arbre au niveau *MinLvl*. Les octants de niveau inférieur à *MinLvl* ne sont donc, en fait, jamais parcourus.

De plus le stockage des entités dans les feuilles de l'arbre ne se fait que si celles-ci sont de niveau au moins *MinLvl*.

La figure 6.14 montre l'octree générée par la librairie *libOL* autour d'une géométrie de voiture. Le maillage surfacique contient 33126 sommets et 66240 triangles. Les niveaux de raffinement vont de 2 à 10. Le nombre de feuilles de l'arbre contenant des entités est de 25264. Le nombre maximum de triangles contenu au niveau 10 par une feuille est de 203. Enfin le nombre total de triangles est de 202731 à cause du processus de duplication des triangles intersectant plusieurs octants. C'est près de 3 fois le nombre initial de triangles.

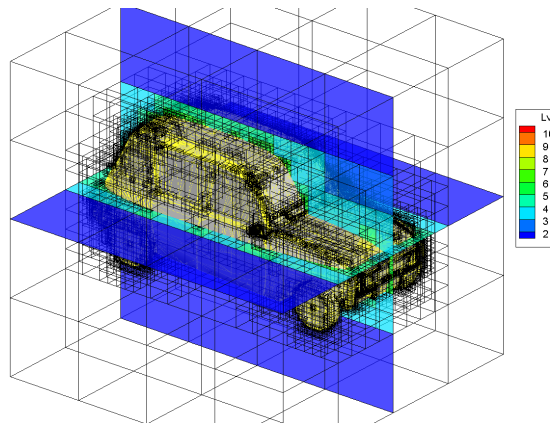


FIGURE 6.14 – Octree générée à partir du maillage surfacique d'une voiture

Le calcul d'une distance d'un point de l'espace vers la géométrie contenue dans l'octree s'effectue selon l'algorithme 1. L'utilisateur spécifie comme données d'entrée : l'octree et toutes les entités géométriques qu'il contient, les coordonnées *VerCrd* du point pour lequel on cherche une distance et une distance maximale *MaxDis* au-delà de laquelle la recherche est interrompue. En retour *libOL* fournit l'élément (sommets ou triangle) identifié comme étant le plus proche *MinItm* ainsi que la distance *MinDis* à laquelle il se trouve.

Le parcours des octants de l'arbre est effectué de façon récursive.

6.2.2 Utilisation de *libOL* et calcul de distance signée dans WCCH

Dans cette partie nous évaluons dans un premier temps les performances de la librairie *libOL*. Nous cherchons à évaluer le coût CPU de la construction de l'octree géométrique et le coût de calcul de la distance. La géométrie retenue pour cette étude est une sphère de rayon 1 avec des maillages allant de 32 à 8388608 triangles (Fig. 6.15). L'étude de qualification de la méthode que nous effectuons ici est similaire à celle de Yang et Stern [Yang et Stern, 2013].

Le tableau 6.1 résume les caractéristiques des octrees générés par *libOL* : nombre initial de triangles, nombre total de triangles après génération de l'octree (duplication des éléments), niveau minimum de l'octree, niveau maximum et nombre maximum de triangles par octant.

```

Données : octree, VerCrd, MaxDis
Résultat : MinDis, MinItm
MinDis ← MaxDis;
localisation de l'octant de niveau minimal contenant le point de coordonnées VerCrd (grille
accélétratrice);
si le point est hors de la grille alors
| projection du point sur la boîte englobante;
fin
initialiser la pile avec l'octant qui contient le point;
tant que tant que la pile n'est pas vide faire
| prendre le premier octant sur la pile;
| évaluer MinDis pour toutes les entités contenues dans les octants enfants;
| pour les 6 octants voisins faire
| | si l'octant n'existe pas ou a déjà été mis dans la pile alors
| | | continuer;
| | sinon
| | | si l'octant est à une distance < MinDis alors
| | | | ajouter l'octant à la pile;
| | | fin
| | fin
| fin
fin

```

Algorithme 1 : Résumé de l'algorithme du calcul de la distance d'un point de l'espace à la géométrie dans *libOL*

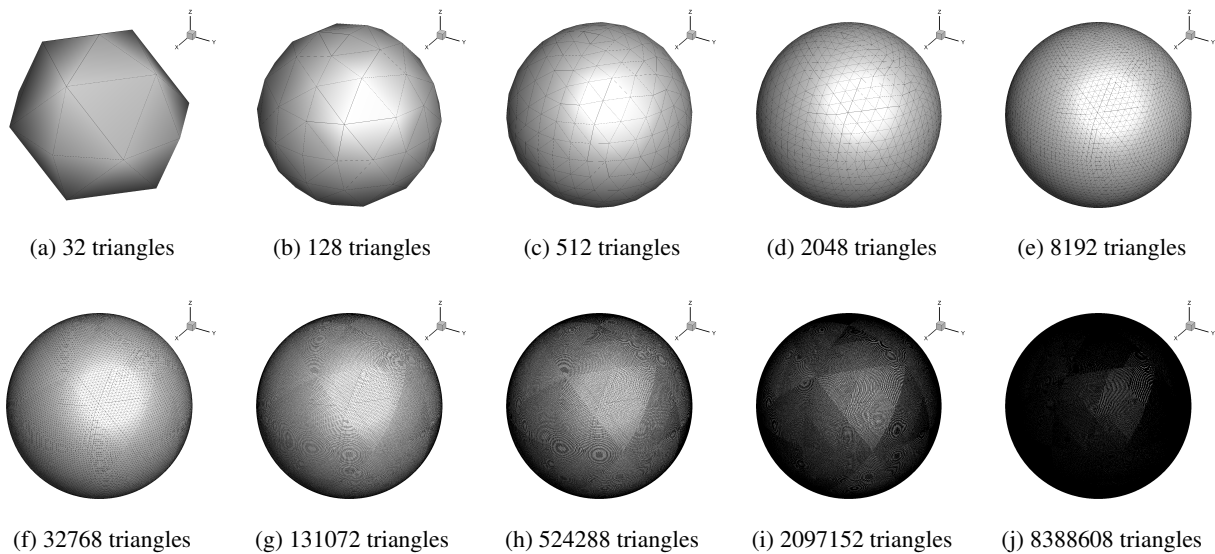


FIGURE 6.15 – Maillage surfacique d'une sphère

Pour les trois maillages les plus grossiers, l'octree n'est pas surchargé et la profondeur de l'arbre correspond à la profondeur minimale prévue par la librairie. En revanche le nombre de triangles dupliqués est important. Sur les maillages suivants, la profondeur de l'arbre augmente progressivement avec un nombre maximum de triangles par octant qui correspond à la limite fixée d'environ 20. Le rapport du nombre total

Maillage	Nombre de triangles	Nombre total de triangles	Niveau min	Niveau max	Nombre max de triangles par octant
1	32	506	2	3	6
2	128	890	2	3	6
3	512	1754	2	3	14
4	2048	5968	2	4	19
5	8192	24152	2	5	19
6	32768	98379	2	6	20
7	131072	392549	2	7	20
8	524288	1553529	2	8	20
9	2097152	6228518	2	9	22
10	8388608	24879261	2	10	22

TABLE 6.1 – Caractéristiques des octrees générés

de triangles sur le nombre initial se stabilise autour de 3.

Ensuite le temps de construction de l'octree est relevé pour ces 10 maillages. Les évaluations des temps CPU ont été effectuées sur un cœur d'un nœud de calcul contenant deux processeurs Intel Xeon E5-2680 v3 Twelve-Core Haswell 2.5GHz. La figure 6.16 montre le temps CPU en fonction du nombre initial de triangles sur un graphique en échelle log-log. Nous remarquons que ce temps de calcul est très faible pour les premiers maillages. Cela s'explique par le fait que la profondeur de l'arbre est fixée à 3 au début de l'algorithme et qu'elle n'évolue pas à mesure que les éléments géométriques sont rangés dans la structure. En revanche, quand le niveau maximum augmente sur les autres maillages, l'augmentation du temps CPU est plus conséquente. L'étape de l'algorithme de construction la plus coûteuse pour ces maillages est la redistribution des éléments géométriques aux octants enfants quand la profondeur de l'arbre gagne un niveau. Pour le maillage 7, contenant environ 100000 triangles, le temps de construction est de l'ordre d'une seconde. Sur le maillage le plus fin à 8000000 de triangles, il est d'environ 14 secondes. La complexité algorithmique de cette étape de génération de l'octree semble être en $O(N_{tri})$.

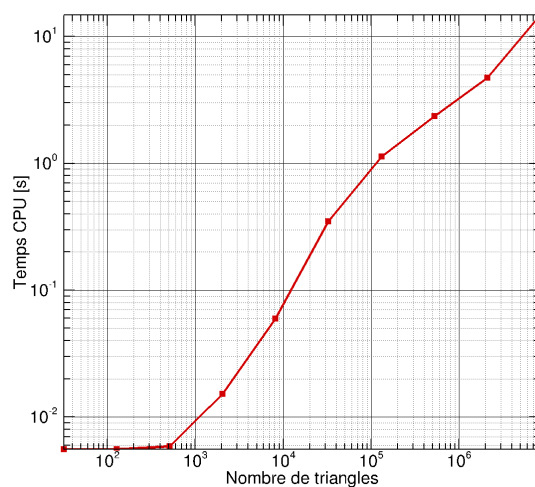


FIGURE 6.16 – Temps CPU de la construction de l'octree

A noter que si ces temps de construction de l'octree peuvent sembler importants, c'est aussi parce que la librairie *libOL* a avant tout été pensée pour offrir une empreinte mémoire la plus faible possible. En particulier la quantité de données enregistrées en mémoire est réduite au strict minimum, tout le reste est recalculé au vol quand cela est nécessaire.

Ensuite, *libOL* est intégrée dans WCCH avec le parti pris que le maillage géométrique n'est pas distribué sur les processeurs pour les calculs parallèles. Chaque cœur de calcul possède donc l'intégralité de la géométrie. On note qu'envisager une stratégie de parallélisation n'est pas un impératif immédiat.

Nous générons un maillage volumique 3D cartésien uniforme de taille $[-2; 2] \times [-2; 2] \times [-2; 2]$ et une distance à la géométrie est calculée depuis tous les volumes de contrôle de ce maillage. 9 maillages volumiques de 32768 à 16777216 cellules sont créés. Les figures 6.17 présentent le coût CPU de calcul de la distance par nombre de cellules par nombre de triangle en fonction du nombre de cellules et du nombre de triangles. Elles nous montrent que le temps de calcul est linéairement dépendant du nombre total de cellules du maillage volumique. Ce comportement est attendu puisque le temps de calcul est supposé être globalement le même quelle que soit la position du point pour lequel la distance est calculée. Nous remarquons également que plus le maillage géométrique est dense, plus le temps CPU est important. En effet, la profondeur de l'arbre est alors plus grande et le parcours dans la structure est rallongé. Pour un maillage volumique fixe, le temps de calcul semble proportionnel à $\sqrt{N_{tri}}$. En résumé la complexité algorithmique du calcul de la distance serait en $O(N_{cellules}\sqrt{N_{tri}})$.

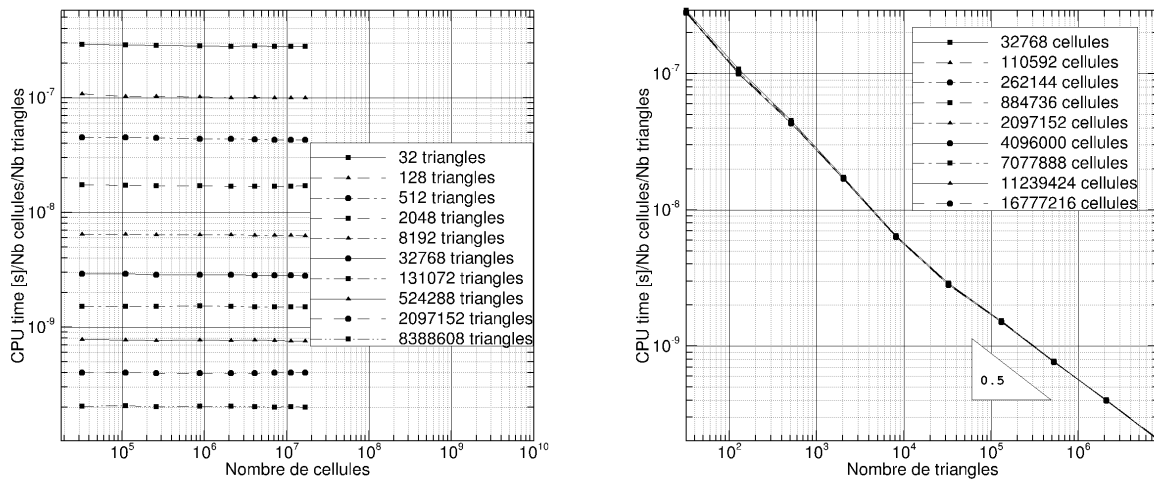


FIGURE 6.17 – Temps CPU du calcul de distance par cellule par triangle en fonction du nombre de cellules (à gauche) et en fonction du nombre de triangles (à droite)

Enfin, le calcul du signe est effectué directement dans le code WCCH selon la méthode basée sur le signe des produits scalaires. Le coût de calcul du signe avec cette méthode est négligeable et ne représente que quelques pourcents du temps CPU consacré au calcul de la distance. La figure 6.18 illustre le calcul du signe pour la géométrie de la voiture de la figure 6.14. L'isosurface de distance nulle est tracée par le logiciel de post-traitement. Les détails apparaissent au-delà du 10^{ème} niveau de raffinement.

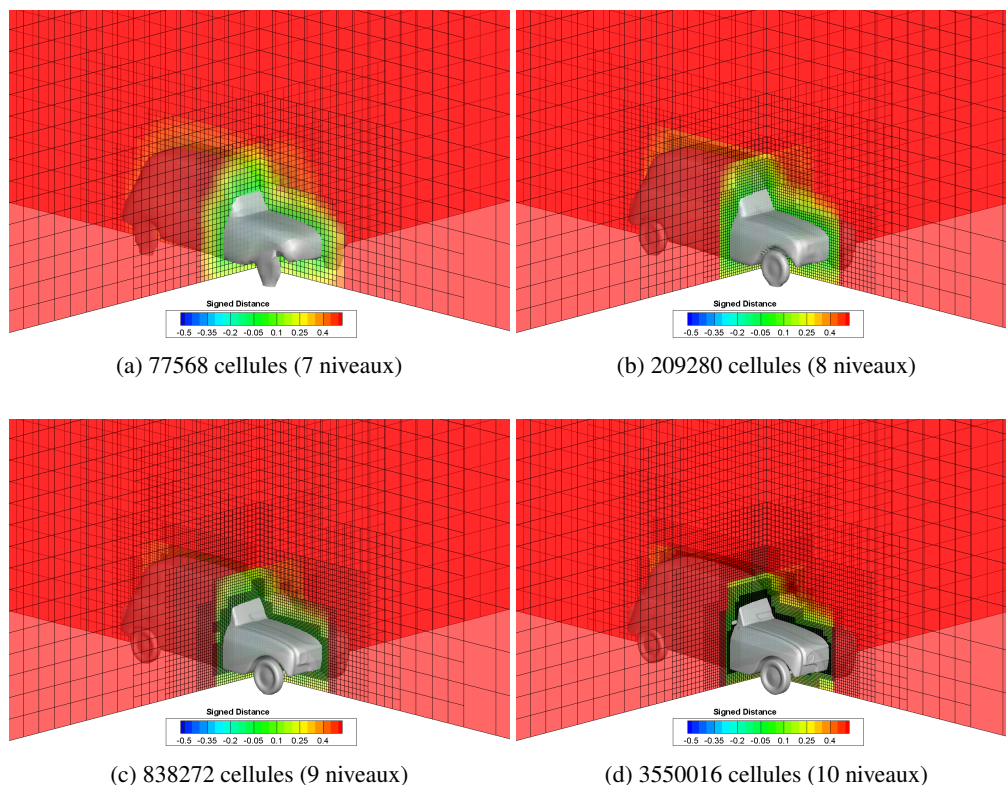


FIGURE 6.18 – Distance signée autour d’une voiture pour différents maillages.

6.2.3 Prise en compte des corps mobiles

Nous souhaitons que la méthode choisie reste efficace en présence de corps mobiles. En particulier, il est exclu d’envisager une reconstruction de l’octree géométrique à chaque fois que le corps bouge. Le coût CPU serait beaucoup trop important si, à chaque étape du schéma Runge-Kutta, celui-ci était détruit puis reconstruit pour suivre les déplacements de la géométrie (cf. étude CPU de la construction de l’octree Fig. 6.16). La technique adoptée est de construire une fois pour toutes un octree qui reste lié au repère de la géométrie. Ainsi, pour calculer la distance d’un point M de l’espace à la géométrie, il est nécessaire de passer les coordonnées de M exprimées dans le repère (X_0, Y_0) lié au maillage volumique dans le repère (X_S, Y_S) lié à la géométrie (Fig. 6.19). La représentation des rotations spatiales d’une base à l’autre est réalisée par des quaternions.

Par ailleurs, de nombreuses stratégies sont possibles pour limiter l’impact du calcul de la distance à chaque itération temporelle. La principale est de restreindre le nombre de cellules pour lesquelles le calcul de la distance est nécessaire à chaque pas de temps. En fait, ce calcul n’a d’intérêt qu’au voisinage du corps mobile pour lisser l’imposition du terme source via l’utilisation de la fonction Heaviside régularisée H . Loin des frontières solides, la valeur de la distance n’a pas besoin d’être très précise à condition qu’elle donne une valeur de H très proche de 0 ou 1. Il est alors primordial de déterminer correctement le signe ; ce qui peut être fait en début de simulation si on suppose que les mouvements du corps seront restreints à une certaine région.

En pratique, le calcul de la distance à chaque pas de temps peut ne concerner que les cellules dont la distance au pas de temps précédent était inférieure (en valeur absolue) à une certaine tolérance. Puis à une fréquence qui reste à fixer par l’utilisateur et selon la cinématique du corps en mouvement, le calcul est effectué pour toutes les cellules.

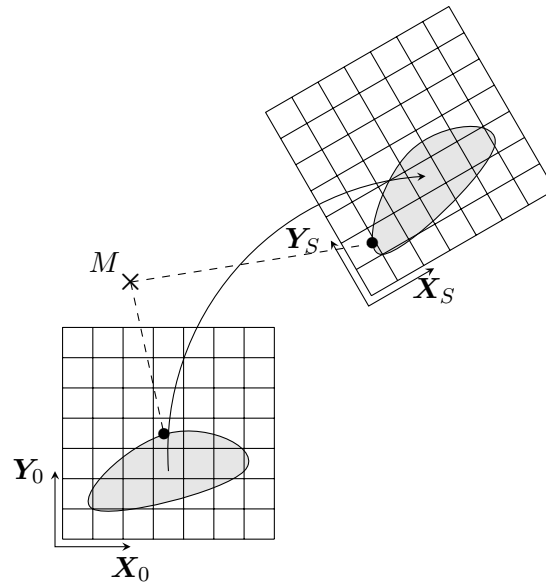


FIGURE 6.19 – Changement de repère pour le calcul de distance à des corps mobiles

Cette approche permet, à chaque pas de temps, de restreindre largement le nombre de cellules pour lesquelles il faut calculer la distance. En appliquant cette méthode, il a été observé une réduction drastique du temps CPU par rapport à l'approche classique n'excluant aucune cellule. Naturellement, plus la fréquence d'actualisation globale de la distance est faible, plus le gain CPU est important.

6.3 Applications à un écoulement d'hydrodynamique navale

6.3.1 Introduction

Cette dernière section présente un exemple de simulation numérique directe d'un écoulement autour d'un propulseur de navire en rotation. L'objectif premier est avant tout d'illustrer comment le calcul de la fonction distance avec *libOL* permet de s'approcher un peu plus de la finalité industrielle du solveur. Toutefois nous profitons du baptême du feu du code sur un écoulement complexe pour également le comparer à des références et illustrer concrètement certaines limitations déjà identifiées.

L'étude d'hélices dans WCCH a déjà fait l'objet d'une thèse [Elie, 2015]. Cependant dans ce travail de recherche, la prise en compte des pales n'était pas effectuée via la méthode de frontière immergée (ou partiellement pour simuler le moyeu par exemple) mais par des méthodes actuatrices (actuator disk ou line). Le principe de ces méthodes est de modéliser la présence d'un propulseur par des termes sources qui ralentissent le fluide en extrayant son énergie cinétique. Elles ont été appliquées à la simulation d'écoulement autour d'hydroliennes avec de bons résultats en termes de sillage [Elie et al., 2017].

En hydrodynamique navale, les performances d'un propulseur sont fournis par deux coefficients adimensionnels :

- le coefficient de poussée (thrust coefficient en anglais)

$$K_T = \frac{T}{\rho n^2 D^4} \quad (6.7)$$

— et le coefficient de couple (torque coefficient) :

$$K_Q = \frac{Q}{\rho n^2 D^5} \quad (6.8)$$

avec T la poussée délivrée par l'hélice, Q son couple. ρ la densité de l'eau. n la vitesse de rotation de l'hélice (en tr/s) et D le diamètre de l'hélice.

Ces coefficients sont mesurés expérimentalement ou calculés numériquement dans des conditions d'eau libre c'est-à-dire sans le bateau que l'hélice propulse, ni la surface libre en général. Ils sont fournis sous forme d'une courbe en fonction d'un autre coefficient adimensionnel, le paramètre d'avance noté J :

$$J = \frac{U}{nD} \quad (6.9)$$

avec U la vitesse du courant vue par l'hélice (proche de la vitesse d'avance du bateau dans le cas d'un navire auto-propulsé).

Enfin le rendement en eau libre de l'hélice vaut :

$$\eta = \frac{TU}{2\pi nQ} = \frac{JK_T}{2\pi K_Q} \quad (6.10)$$

Pour notre mise en application, nous avons choisi l'hélice du porte-conteneur KCS (Kriso Container Ship). Bien que ce bateau prototype n'ait jamais été construit à l'échelle réelle, des maquettes du navire et de l'hélice ont été abondamment testées dans différents bassins d'essais pour fournir des données de référence. Elles permettent de valider des codes CFD lors de workshops comme celui de SIMMAN 2008 ou CFD Workshop Tokyo 2015. La configuration retenue est celle de l'expérience réalisée au bassin du KRISO en Corée du Sud et à celui du SRI (NMRI maintenant) au Japon [Fujisawa et al., 2000].

6.3.2 Paramétrisation du calcul

L'hélice fait $D = 0.25m$ de diamètre. Elle est reliée à un arbre de diamètre $d = 0.047m$. Le maillage surfacique est généré à partir de 17022 triangles dont près de 8900 pour l'hélice (Fig. 6.20).

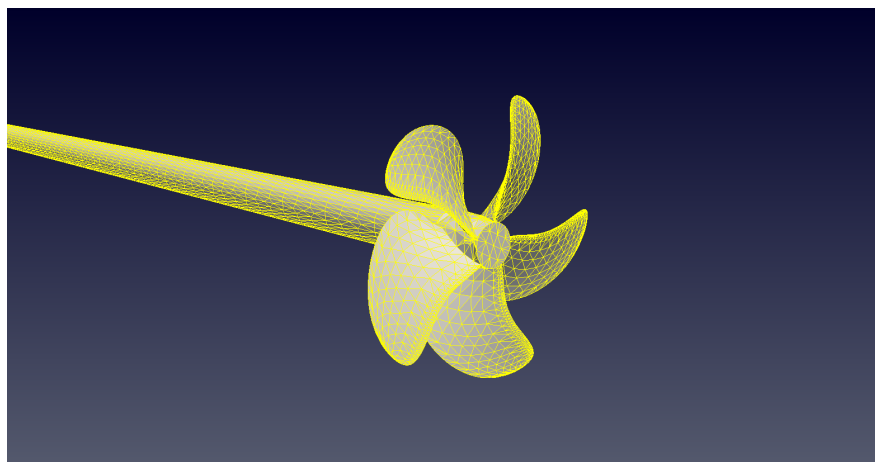


FIGURE 6.20 – Maillage surfacique de l'hélice KCS

Ce maillage présente quelques défauts avec des triangles qui s'intersectent. Les nombreuses tentatives pour améliorer sa qualité sont restées malheureusement infructueuses et ont conduit à l'utilisation de la méthode de lancer de rayon pour le calcul du signe plutôt que celle du signe du produit scalaire avec les vecteurs normaux.

Le maillage volumique est initialement généré en spécifiant un critère de raffinement géométrique pour ne raffiner le maillage qu'autour du corps (Fig. 6.21). Le domaine de calcul est un cube de côté $10D$ et son centre coïncide avec le centre de rotation de l'hélice.

$$\left\{ \begin{array}{l} \text{Taille du domaine : } [-0.75; 1.75] \times [-1.25; 1.25] \times [-1.25; 1.25] \\ \text{Niveaux de raffinement : } 12 \\ \text{Nombre de cellules par bloc : } 2 \times 2 \times 2 \\ \Delta x_{min} = 0.00061m \\ \text{Nombre initial total de cellules : } 8052360 \end{array} \right. \quad (6.11)$$

Pour préciser la résolution spatiale adoptée, la corde d'une pale de l'hélice à $0.7r$ (où r désigne le rayon de l'hélice, $r = D/2$) est $c = 0.092m$ et l'épaisseur $e = 0.0033m$. Ainsi :

$$\frac{c}{\Delta x_{min}} = 150 \text{ et } \frac{e}{\Delta x_{min}} = 5.4 \quad (6.12)$$

Ce niveau de raffinement est bien insuffisant pour capter le développement ou le décollement de la couche limite mais encore une fois, là n'est pas l'objectif. Il faudrait au moins une dizaine de cellules dans l'épaisseur de la pale pour imposer correctement les conditions aux limites en vitesse sur l'intrados et l'extrados des pales.

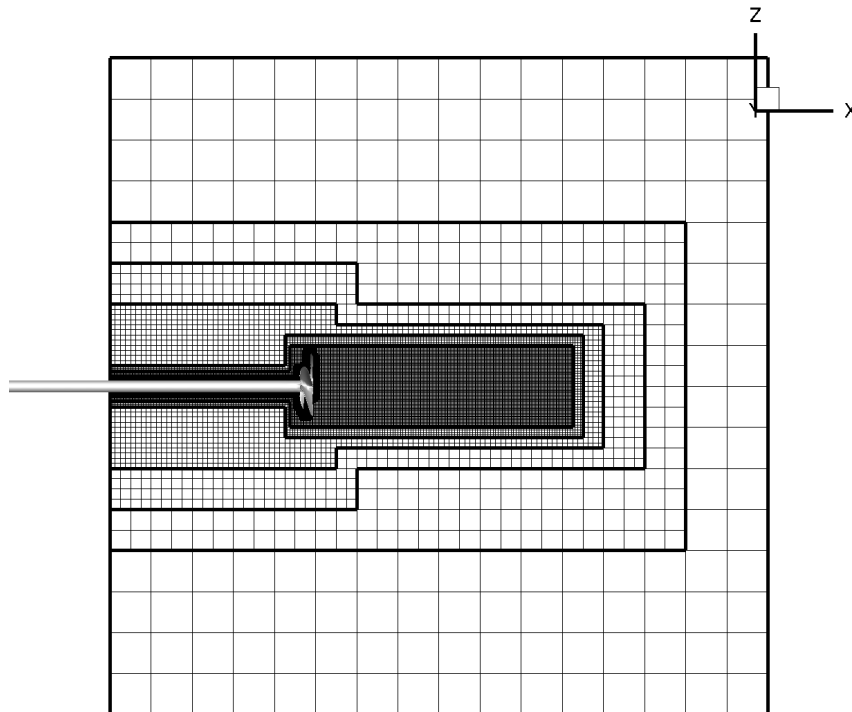


FIGURE 6.21 – Plan de coupe du maillage volumique

La figure 6.22 présente l'iso-surface 0 de la fonction distance. Elle est supposée représenter l'enveloppe du corps solide vue par la méthode de frontière immergée. On peut constater que les bords de fuite des pales sont "grignotés" et mal représentés, conséquence de la sous-résolution spatiale du maillage volumique.

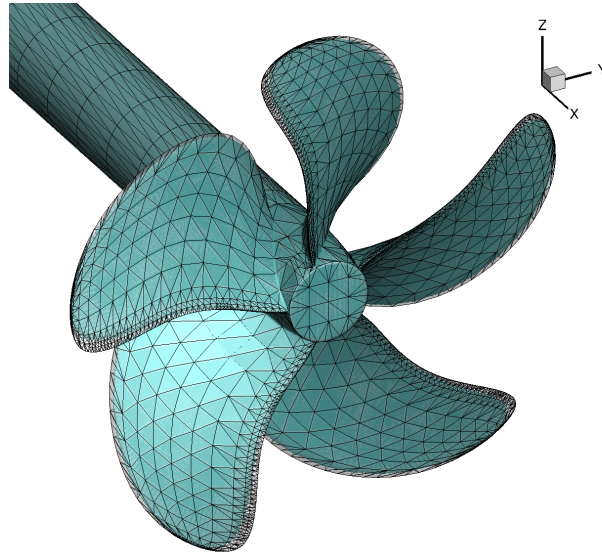


FIGURE 6.22 – Iso-surface de la fonction distance pour une valeur de 0.

Ce cas d'application ayant été réalisé en parallèle des développements sur la formulation incompressible et la montée en ordre de la méthode de frontière immergée, l'approche faiblement-compressible et la méthode initiale de frontière immergée ont été privilégiées. Les paramètres physiques et numériques de calcul sont les suivants :

$$\left\{ \begin{array}{l} \text{Vitesse du courant : } U = 2.1m/s \\ \text{Vitesse de rotation de l'hélice : } \Omega = -2\pi n e_x \text{ avec } n = 12tr/s \\ \text{Vitesse à } 0.7r : V = \sqrt{U^2 + (0.7r\Omega)^2} = 6.9m/s \\ \text{Paramètre d'avance : } J = 0.7 \text{ (point d'auto-propulsion à } 0.73 \text{ pour le bateau)} \\ \text{Masse volumique : } \rho = 998kg/m^3 \\ \text{Viscosité dynamique : } \mu = 10^{-3}kg/(m.s) \\ \text{Vitesse du son : } c_0 = 90m/s \\ \text{CFL : } 0.7 \\ \text{Temps physique de simulation : } 0.6s \approx 7 \text{ rotations d'hélice} \end{array} \right. \quad (6.13)$$

Les 0.6 secondes de simulation physique semblent suffisantes pour stabiliser les efforts hydrodynamiques appliqués sur l'hélice.

Une condition aux limites entrante permet de générer le courant dans lequel l'hélice est placé. Toutes les autres frontières du domaine de calcul sont de type sortante (cf. Fig. 6.23). Une condition d'adhérence est spécifiée sur l'arbre et l'hélice.

Même si jusqu'à présent les maillages considérés étaient uniquement statiques, nous avons choisi pour ce cas d'appliquer un critère dynamique de raffinement (AMR) basé sur la vorticit .

La simulation s'est ex cut e en 190 heures sur 192 c eurs Intel Xeon E5-2680 Haswell cadenc s   2.5GHz. Le temps CPU d di  au calcul de la distance sign e repr sente moins de 2% du temps de calcul d'un pas de temps.

6.3.3 R sultats

La figure 6.24 montre le d veloppement des structures tourbillonnaires en aval de l'h lice.

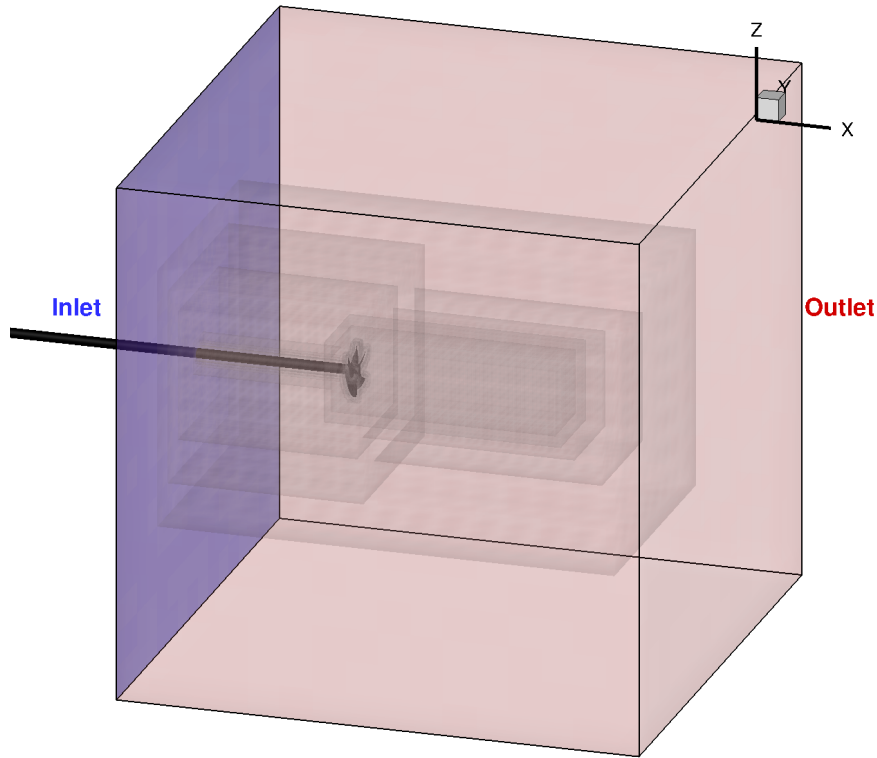


FIGURE 6.23 – Domaine de calcul et conditions aux limites

A titre de comparaison, nous présentons d'autres résultats de simulations de l'hélice réalisées avec le logiciel commercial Star-CCM+ dans un contexte industriel. Ces résultats ont été produits par la société HydrOcean, dans le cadre du projet collaboratif "Bassin numérique" porté par l'Institut de Recherche Technologique Jules Verne. Dans un contexte d'exploitation des technologies numériques pour réduire les temps de conception de nouvelles carènes de navires ou de propulseurs, ce projet a conduit plusieurs acteurs de l'hydrodynamique navale (STX, Sirehna, HydrOcean, Bureau Veritas, Centrale Nantes) à collaborer sur la validation de leur savoir-faire en simulation numérique.

La figure 6.25 expose les maillages polyédriques utilisés par HydrOcean. Nous pouvons constater que les mailles structurées de la zone de couche limite ont une épaisseur très fine. Cela est imposé par l'utilisation d'un modèle de turbulence de type RANSE sans loi de paroi, i.e. avec $y^+ = 1$, où y^+ est une distance non dimensionnelle telle que :

$$y^+ = \frac{u_* y}{\nu} \quad (6.14)$$

avec u_* la vitesse de friction $u_* = \sqrt{\frac{\tau_w}{\rho}}$, τ_w la contrainte de cisaillement à la paroi, et y la distance à la paroi. y^+ est utilisée comme grandeur de référence en théorie de couche limite. Une valeur de 1 pour la première maille en contact avec la paroi est nécessaire pour capter fidèlement une couche limite. La taille de cette première cellule est donc dictée par cette contrainte qui peut être relâchée si des lois de paroi sont employées conjointement avec un modèle de turbulence.

La taille de la première cellule de couche limite que l'on peut voir sur la figure 6.25 fixe les idées sur la taille que devraient avoir les cellules de notre maillage volumique WCCH près de la frontière solide. Nous sommes loin de respecter cette contrainte comme le montre le calcul de y^+ présenté sur la figure 6.26. y^+ a été évalué numériquement en calculant τ_w pour les cellules fluides les plus proches de la paroi solide tel

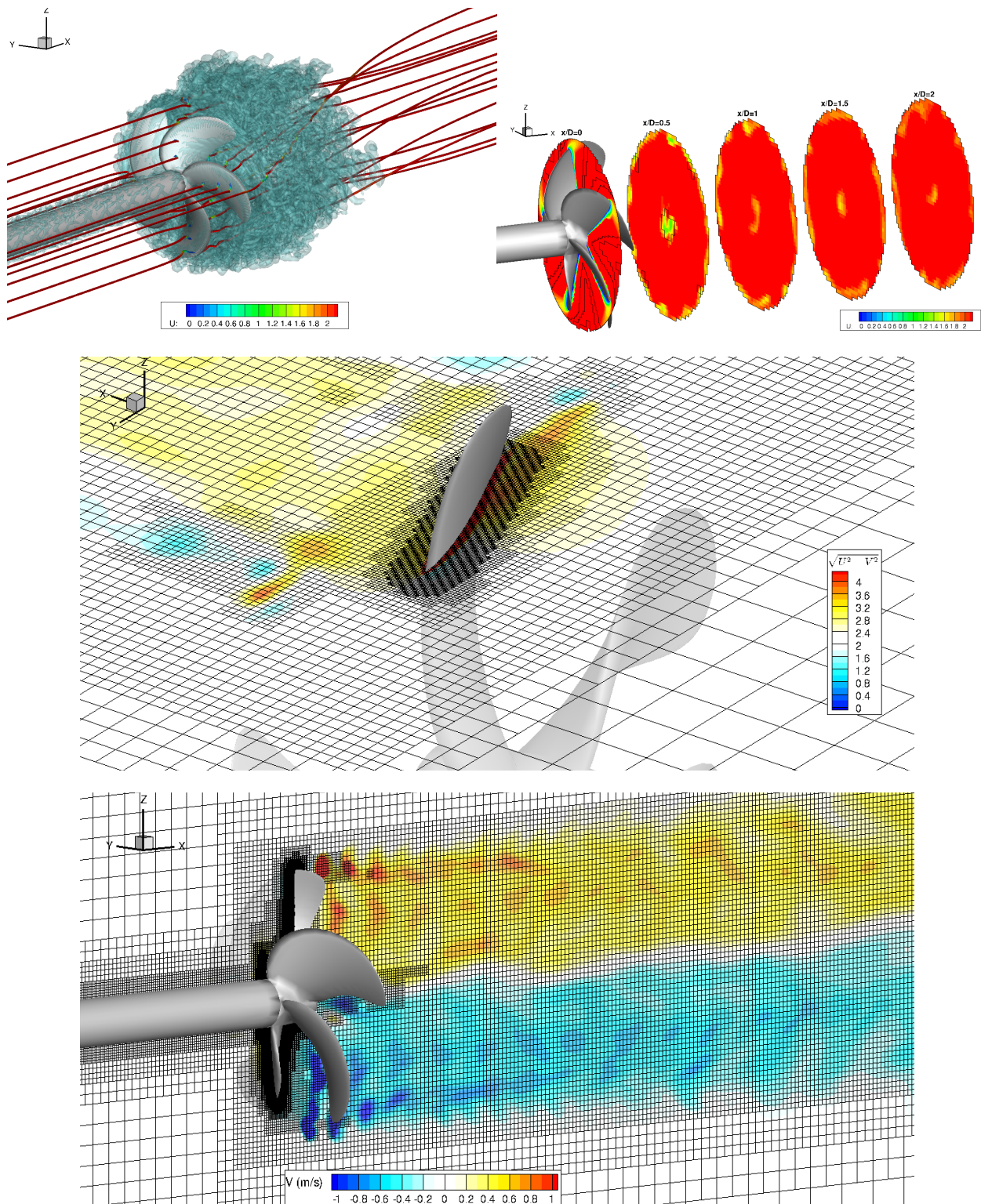


FIGURE 6.24 – Isocontour 1000 du critère Q et lignes de courant (en haut à gauche), champ de vitesse horizontale dans des plans de coupe en aval de l'hélice (en haut à droite), champ de vitesse dans un plan horizontal (milieu), champ de vitesse transversale (en bas).

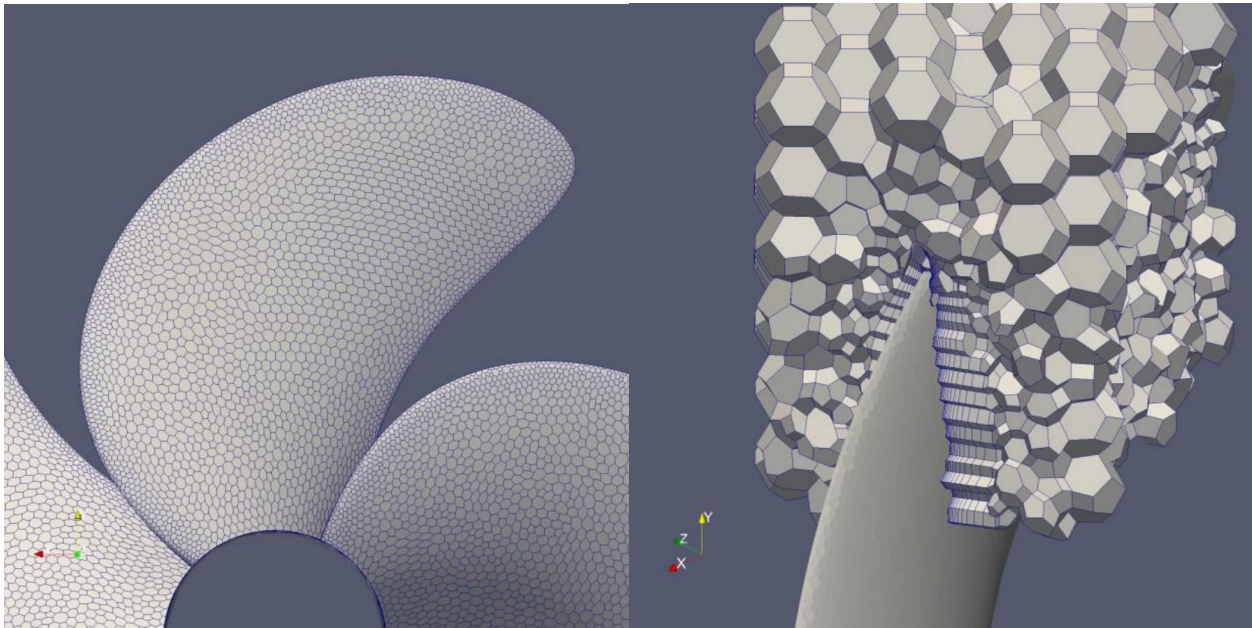


FIGURE 6.25 – Vues du maillage Star-CCM+. Maillage surfacique (à gauche) et maillage volumique avec couche limite (à droite)

que :

$$\tau_w = \mu \frac{u_{fluide} - u_{solide}}{y} \quad (6.15)$$

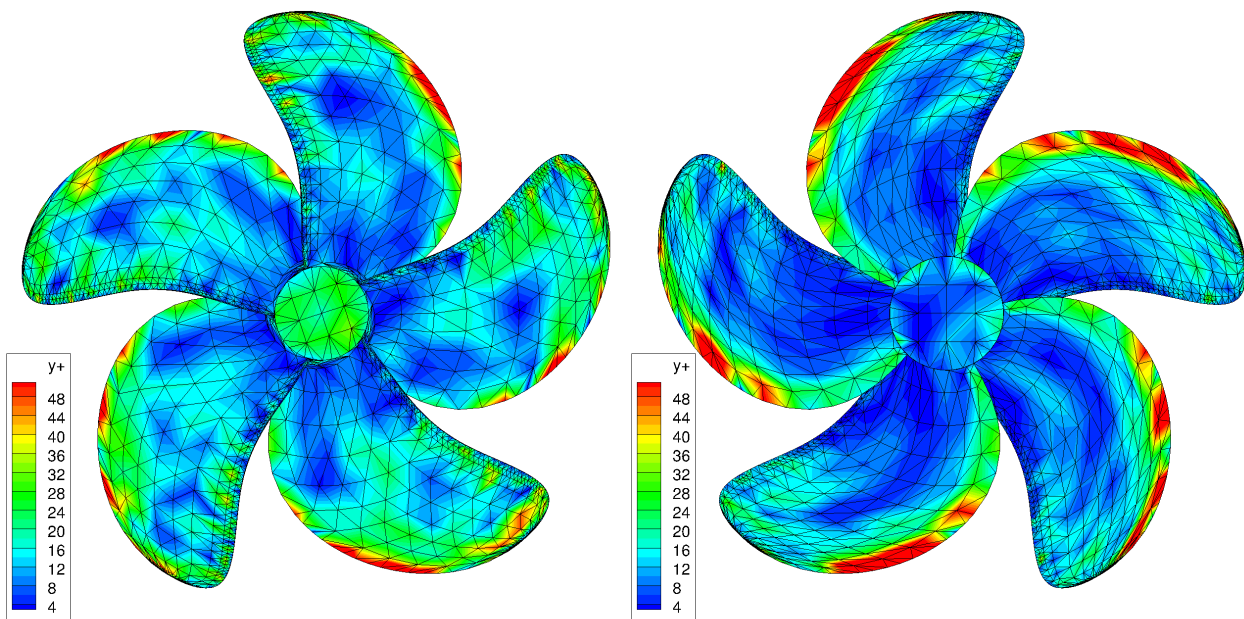


FIGURE 6.26 – Vues de $y+$ pour le maillage WCCH

$y+$ prend des valeurs comprises entre 0 et 100. Les valeurs les plus élevées sont relevées aux extrémités des pales près des bords de fuite. Cela coïncide avec les régions du maillage où l'hélice est la moins bien

représentée du fait de la faible épaisseur des profils et d'une sous-discrétisation spatiale.

La figure 6.27 expose les champs de pression adimensionnels sur l'intrados et l'extrados des pales pour les simulations faites avec WCCH et Star-CCM+.

$$C_p = \frac{P}{1/2\rho n^2 D^2} \quad (6.16)$$

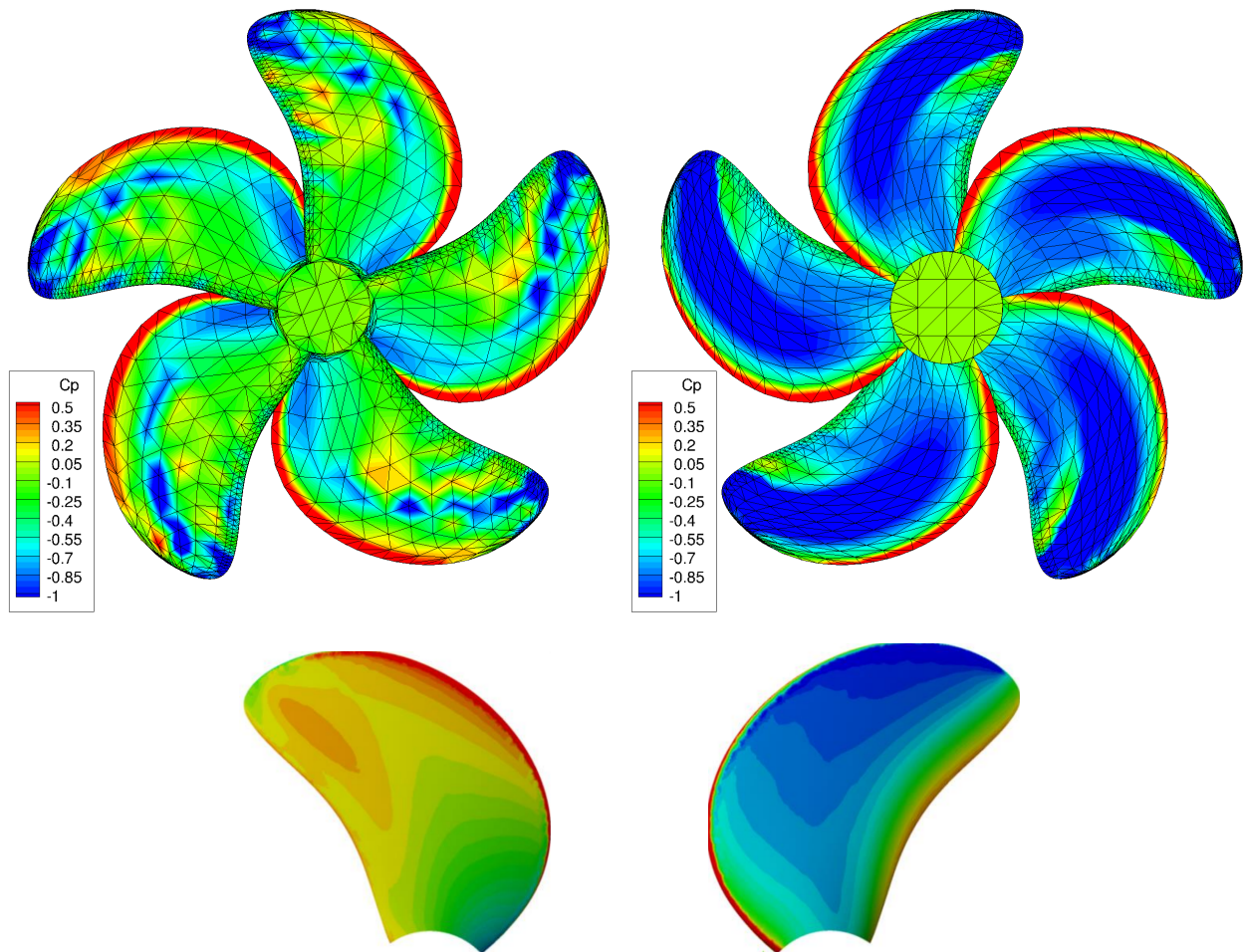


FIGURE 6.27 – Champ de pression sur l'intrados et l'extrados des pales. Résultats WCCH (haut, hélice complète) résultats Star-CCM+ (bas, une seule pale)

Nous pouvons constater que même si la tendance est respectée entre le résultat WCCH et le résultat Star-CCM+, les écarts de valeurs sont relativement importants.

Enfin, la figure 6.28 compare le point de fonctionnement issu du calcul WCCH avec les résultats issus des essais expérimentaux en bassin. Nous pouvons constater que les valeurs des coefficients K_T et K_Q obtenus sont loin des références, avec un écart relatif de près de 50%. Cette différence importante n'est pas étonnante compte-tenu des valeurs de y^+ relevées ci-dessus. L'absence d'utilisation d'un modèle de turbulence avec WCCH nous amène une simulation DNS très sous-résolue. Toute comparaison avec l'écoulement réel a peu de sens a priori.

Pour conclure, ce cas test a permis d'illustrer la mise en œuvre d'une technique d'immersion de géométrie complexe dans le code WCCH et une application concrète de la méthode des frontières immergées.

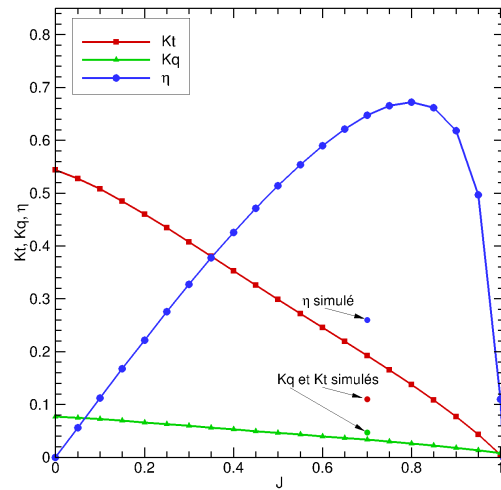


FIGURE 6.28 – Courbe de performance de l’hélice issue des expériences et point de fonctionnement simulé

Même si l’écoulement n’est pas reproduit fidèlement, la démarche a mis en évidence la possibilité de simuler des écoulements autour de géométries complexes mobiles avec un coût CPU acceptable.

6.4 Conclusion

En définitive, une technique robuste et efficace de calcul de distance signée a été implémentée dans le code WCCH dans sa version faiblement-compressible. Elle utilise notamment la librairie *libOL* pour le calcul de la distance absolue. Avec cette librairie, le maillage surfacique de la géométrie est rangé dans un octree pour accélérer les temps d'accès aux éléments géométriques et ne pas avoir à les parcourir dans l'ensemble. Comme la distance à la géométrie ne doit être précise qu'au voisinage de la frontière, il est possible de restreindre le nombre de cellules du maillage volumique qui ont besoin de cette valeur. Pour le calcul du signe une première méthode utilisant le signe du produit scalaire avec le vecteur normal de l'élément géométrique le plus proche a donné de bons résultats pour des maillages surfaciques de bonne qualité. Une fois que l'élément géométrique le plus proche a été identifié avec le calcul de la distance, la détermination du signe avec cette méthode est fiable, mais surtout immédiate avec un coût CPU acceptable. Cependant, dès que la géométrie comporte des irrégularités comme des triangles qui s'intersectent, alors elle peut échouer à donner le bon signe. Une deuxième méthode basée sur un lancer de rayon qui compte le nombre d'intersections avec la géométrie offre une robustesse nettement supérieure mais avec un coût CPU bien plus important.

La méthode a ensuite été mise en application sur un écoulement en eau libre autour d'une hélice de bateau. Cet exemple d'application a permis de montrer que la technique mise en place est tout à fait compatible avec la simulation d'écoulements autour de géométries complexes en mouvement. Elle a aussi mis en lumière les faiblesses du solveur pour traiter des problèmes classiques de l'hydrodynamique marine qui sont déjà bien résolus par les outils CFD actuels. Mais il faut aussi reconnaître que l'écoulement choisi n'était pas forcément celui susceptible de démontrer pleinement les atouts du code (ordre élevé, frontière immergée, etc.) Sans changer les conclusions, une approche purement incompressible aurait sans doute permis de réduire les temps de restitution.

Conclusions générales et perspectives

L'objectif global poursuivi lors de ce travail de thèse était la réduction des temps de restitution d'un code résolvant les équations de Navier-Stokes par une méthode de volumes finis sur des grilles cartésiennes raffinées localement. L'originalité de ce code développé au LHEEA depuis bientôt une dizaine d'années est de reposer sur une approche faiblement-compressible. Une résolution purement explicite des équations facilite alors le recours à des schémas numériques d'ordre élevé et convient parfaitement aux architectures de calcul actuelles massivement parallèles.

Trois axes d'études ont été abordés pour répondre à la problématique fixée.

Le premier a concerné l'implémentation d'une version purement incompressible. En effet la résolution explicite des équations compressibles impose une contrainte forte sur la taille du pas de temps qui doit tenir compte de la vitesse de propagation du son dans l'écoulement. Bien que l'approche faiblement-compressible consiste à abaisser la vitesse du son tant que les effets de compressibilité restent négligeables, cela reste une limitation importante en comparaison avec l'approche complètement incompressible. Celle-ci permet de s'affranchir de la limitation de la vitesse du son et autorise des pas de temps plus élevés. En revanche le prix à payer est un système linéaire à résoudre pour déterminer la pression. Même si celui-ci est creux, il représente un coût CPU supplémentaire par rapport à la formulation faiblement-compressible. Le gain d'une formulation par rapport à l'autre n'est donc pas si évident de prime abord, notamment dans le contexte particulier du code développé : schémas d'ordre élevé en espace et en temps, méthode de frontière immergée pour la prise en compte des corps solides, grille cartésienne raffinée localement, etc. Le choix de la méthode de résolution des équations incompressible s'est finalement porté sur un schéma de projection de Chorin qui permet de respecter au mieux les exigences de précision et d'efficacité. Ce choix a permis de conserver certains schémas numériques (WENO5, Runge-Kutta) déjà utilisés en approche faiblement-compressible, à ceci près que quelques modifications ont dû être apportées comme le changement de l'emplacement des inconnues de vitesse aux faces des volumes de contrôle pour des questions de stabilité. Par ailleurs la méthode de frontière immergée présente initialement dans le code a été étendue à l'approche incompressible. Une technique originale de traitement des interfaces de raffinement, basée notamment sur une méthode d'étirement des volumes de contrôle, a aussi été proposée pour adapter les reconstructions WENO dans ces régions du maillage. Elle a été comparée à l'approche "guard-cells" plus commune dans la littérature. En faisant coexister les deux formulations incompressible et faiblement-compressible dans le même code il a été ainsi possible de les comparer de manière approfondie sur différents aspects : la précision numérique, la complexité algorithmique, la scalabilité des codes, etc. Au regard des comparaisons effectuées sur différents cas tests il apparaît que l'approche incompressible présente un gain de temps de calcul compris entre 5 et 20 pour les estimations les plus basses par rapport au faiblement-compressible. Ce bénéfice calculé à précision équivalente est significatif. Pour autant, la formulation faiblement-compressible possède l'avantage d'offrir une algorithmique relativement simple. De plus elle semble plus tolérante au sous-chargement des processeurs lors de la parallélisation du calcul. Enfin il convient de rester prudent sur ces gains en précision et de ne pas occulter le cadre d'application dans lequel ils ont été mis évidence.

Le deuxième axe a porté sur l'amélioration de la précision de la méthode de frontière immergée. Un état de l'art ciblé a permis de positionner la méthode originelle du code par rapport aux méthodes proposées dans la littérature. L'étude de l'application de la méthode sur une équation linéaire d'advection-diffusion a permis de mettre en évidence l'ordre de la méthode ainsi que certains aspects comme la dépendance à la taille du pas de temps d'une solution pourtant stationnaire. Toujours dans l'optique de réduction des temps de restitution par l'amélioration de la précision, une méthode d'ordre théoriquement supérieur a été suggérée pour les formulations incompressible et faiblement-compressible. Le principe est de prendre en compte la position exacte de la frontière solide. Les premiers résultats obtenus ne donnent pas entière satisfaction puisque la qualité des solutions est inférieure en comparaison avec la méthode initiale. Ce constat n'est pas pessimiste mais témoigne de la difficulté à proposer une méthode de frontière immergée précise et robuste. De plus, la méthode proposée n'est qu'une ébauche de la démarche à suivre pour monter en ordre le schéma de frontière immergée. Les perspectives d'amélioration de la méthode listées dans le Chapitre 5 sont nombreuses et laissent entrevoir des solutions donnant des résultats plus précis. Par ailleurs, il a été constaté que le développement d'un schéma de frontière immergée est plus aisé en formulation incompressible. En faiblement-compressible, la moindre perturbation du champ de vitesse par les termes sources conduit à une irrégularité du champ de pression. Avec la méthode de frontière immergée initiale, le champ de pression est bien régulier au voisinage des frontières solides grâce à la transition effectuée par la fonction Heaviside lissée. La zone de transition permet d'absorber la partie acoustique ainsi générée et de conserver des champs réguliers. Avec la nouvelle méthode proposée, les ondes acoustiques ne sont plus amorties ou filtrées, rendant la tâche de montée en ordre particulièrement ardue. En incompressible, le caractère implicite de la pression permet de lisser les champs et offre donc plus de flexibilité. Une difficulté supplémentaire est aussi l'incompatibilité entre la contrainte cinématique des conditions de bords et la contrainte de divergence nulle du champ de vitesse si elles ne sont pas prises en compte à la fois lors l'étape de prédiction du schéma de projection et lors de la résolution de l'équation de Poisson.

Enfin le dernier axe a concerné l'immersion rapide et robuste de géométries complexes. Une procédure a été mise en place pour calculer efficacement la distance signée nécessaire à la méthode de frontière immergée. Le calcul de la distance repose sur la localisation géométrique par arbre octal des triangles du maillage surfacique du corps. Pour le calcul du signe, si la géométrie est de bonne qualité alors le signe du produit scalaire avec les vecteurs normaux des éléments les plus proches renseigne sur la position intérieure ou extérieure des cellules par rapport au corps. Cependant si elle présente quelques défauts comme des triangles qui s'intersectent alors une méthode de lancer de rayon doit être appliquée pour donner un résultat exact. Dans le deuxième cas, le calcul du signe est bien évidemment plus coûteux puisqu'il nécessite un parcours de l'arbre en longueur. Par ailleurs, afin de réduire les temps de calcul, il est possible de restreindre le nombre de cellules du maillage volumique calculant la distance signée à chaque pas de temps. En effet l'information de distance au corps n'est indispensable qu'au voisinage du corps. Toutes les cellules du domaine de calcul n'ont donc pas nécessairement besoin de faire appel aux routines de calcul de distance signée à chaque pas de temps. Une bonne efficacité de calcul peut donc être conservée pour des corps mobiles. La technique mise en place a été évaluée sur une simulation numérique directe sous-résolue d'un écoulement autour d'une hélice en eau libre avec un objectif d'illustration plus que de validation.

Recommandations

A l'issue de cette thèse, les perspectives d'amélioration de la méthode ou de développements ne manquent pas. Nous pourrions envisager entre autres :

- Évaluer le raffinement dynamique de maillage sur la base de critères physiques (gradient de pression, de vitesse, vorticité, etc.) pour la formulation incompressible. Le raffinement adaptatif est déjà fonctionnel pour la formulation faiblement-compressible, il ne reste qu'à l'étendre à la formulation incompressible en définissant des opérateurs appropriés de restriction et de prolongation.

- Quantifier le gain de la méthode guard-cells par rapport à l'approche originale présenté dans ce manuscrit pour traiter les interfaces de raffinement. Il a été montré que la méthode guard-cells donne des résultats plus précis (cf. Chapitre 4). Cependant la méthode d'étirement que nous avons proposée semble mieux s'adapter aux maillages avec un nombre de cellules par bloc et par direction de 2. De plus elle offre une empreinte mémoire plus légère puisqu'elle ne nécessite pas le stockage en mémoire de cellules supplémentaires. Les temps CPU des deux méthodes devraient être comparés à précision équivalente et pour des maillages très hétérogènes (avec un grand nombre d'interfaces de raffinement).
- Si un niveau de performance supplémentaire était attendu pour la résolution de l'équation de Poisson, l'implémentation d'une méthode multigrille géométrique pourrait être envisagée. Deux choix s'offriraient alors suivant le niveau de performance exigé et le temps alloué au développement : continuer à utiliser PETSc en ayant recours aux outils de la librairie qui permettent de gérer la hiérarchie des niveaux récursifs de la méthode multigrille (décomposition de domaine *DMPlex* par exemple) ou s'en passer totalement.
- Implémenter une méthode pour la simulation d'écoulements diphasiques en formulation incompressible. Elle fournirait un nouveau critère de comparaison sur l'aptitude des formulations incompressible et faiblement-compressible à simuler un écoulement à surface libre.
- Poursuivre la montée en ordre de la méthode de frontière immergée en se concentrant sur l'équation de continuité et uniquement en formulation incompressible dans un premier temps.

Enfin concernant l'arbitrage entre l'approche incompressible et faiblement-compressible ; il est apparu au cours de cette thèse que faire coexister deux approches présentant quelques divergences n'est pas une tâche aisée. A court terme il reste possible de maintenir les deux formulations. A plus long terme, il serait judicieux de ne conserver plus qu'une seule formulation pour faciliter la maintenance et l'évolution du code. Le choix de l'une ou l'autre des formulations pourra dépendre de l'importance accordée aux avantages de chacune et des champs applicatifs visés pour le code in fine.

Pour conclure, le code WCCH est arrivé à un stade où il doit être éprouvé sur des écoulements d'intérêt industriel pour mettre en évidence les axes de recherche. Le cas test présenté dans le dernier chapitre s'inscrit dans cette veine mais les cas envisagés devront justifier l'emploi de la méthode (schémas d'ordre élevé avec frontière immergée). L'objectif sera de piloter les développements futurs en lien avec les attentes de l'industrie navale. En l'état actuel du code, les applications envisageables à court terme pourraient être les simulations : d'écoulements aérodynamiques autour de structures complexes, de ventilation de cales des navires, des phénomènes de sillage (signature d'un sous-marin par exemple), d'interaction fluide-structure (interaction coque-hélice, pose de colis, lancement d'armes de type torpilles, etc.).

Bibliographie

- [Almgren et al., 2010a] Almgren, A., Beckner, V., Bell, J., Day, M., Howell, L., Joggerst, C., Lijewski, M., Nonaka, A., Singer, M., and Zingale, M. (2010a). Castro : A new compressible astrophysical solver. i. hydrodynamics and self-gravity. *The Astrophysical Journal*, 715(2) :1221.
- [Almgren et al., 2010b] Almgren, A., Bell, J., Kasen, D., Lijewski, M., Nonaka, A., Nugent, P., Rendleman, C., Thomas, R., and Zingale, M. (2010b). Maestro, castro, and sedona–petascale codes for astrophysical applications. *arXiv preprint arXiv :1008.2801*.
- [Almgren et al., 2013a] Almgren, A. S., Aspden, A., Bell, J. B., and Minion, M. L. (2013a). On the use of higher-order projection methods for incompressible turbulent flow. *SIAM Journal on Scientific Computing*, 35(1) :B25–B42.
- [Almgren et al., 2000] Almgren, A. S., Bell, J. B., and Crutchfield, W. Y. (2000). Approximate projection methods : Part i. inviscid analysis. *SIAM Journal on Scientific Computing*, 22(4) :1139–1159.
- [Almgren et al., 2013b] Almgren, A. S., Bell, J. B., Lijewski, M. J., Lukić, Z., and Van Andel, E. (2013b). Nyx : A massively parallel amr code for computational cosmology. *The Astrophysical Journal*, 765(1) :39.
- [Almgren et al., 1996] Almgren, A. S., Bell, J. B., and Szymczak, W. G. (1996). A numerical method for the incompressible navier-stokes equations based on an approximate projection. *SIAM Journal on Scientific Computing*, 17(2) :358–369.
- [Angot, 1989] Angot, P. (1989). *Contribution à l'étude des transferts thermiques dans des systèmes complexes : application aux composants électroniques*. Thèse de doctorat, Bordeaux 1.
- [Angot et al., 1999] Angot, P., Bruneau, C.-H., and Fabrie, P. (1999). A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81(4) :497–520.
- [Angot et Laugier, 1994] Angot, P. and Laugier, M. (1994). La methode fic de raccordement conservatif de sous-domaines emboites pour un modele de circulation oceanique. *Comptes rendus de l'Académie des sciences. Série II, Mécanique, physique, chimie, astronomie*, 319(9) :993–1000.
- [ANL, 1997] ANL (1997). PETSc Web page. <http://www.mcs.anl.gov/petsc>.
- [Arquis et Caltagirone, 1984] Arquis, E. and Caltagirone, J. (1984). Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux : application a la convection naturelle. *CR Acad. Sci. Paris II*, 299 :1–4.
- [Ascher et Petzold, 1998] Ascher, U. M. and Petzold, L. R. (1998). *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam.
- [Bærentzen et Aanaes, 2005] Bærentzen, J. A. and Aanaes, H. (2005). Signed distance computation using the angle weighted pseudonormal. *IEEE Transactions on Visualization and Computer Graphics*, 11(3) :243–253.
- [Bai et Brandt, 1987] Bai, D. and Brandt, A. (1987). Local mesh refinement multilevel techniques. *SIAM Journal on Scientific and Statistical Computing*, 8(2) :109–134.

- [Balaras, 2004] Balaras, E. (2004). Modeling complex boundaries using an external force field on fixed cartesian grids in large-eddy simulations. *Computers & Fluids*, 33(3) :375–404.
- [Balay et al., 2017] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H. (2017). PETSc users manual. Technical Report ANL-95/11 - Revision 3.8, Argonne National Laboratory.
- [Balay et al., 1997] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. (1997). Efficient management of parallelism in object oriented numerical software libraries. In Arge, E., Bruaset, A. M., and Langtangen, H. P., editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press.
- [Balsara, 2001] Balsara, D. S. (2001). Divergence-free adaptive mesh refinement for magnetohydrodynamics. *Journal of Computational Physics*, 174(2) :614–648.
- [Bandringa, 2010] Bandringa, H. (2010). Immersed boundary methods. Master’s thesis, University of Groningen.
- [Bardin, 2015] Bardin, A. (2015). *Développement de méthodes d’interface dans un solveur hydrodynamique basé sur une résolution volumes finis sur maillage cartésien en hypothèse faiblement compressible*. Thèse de doctorat, Ecole Centrale de Nantes.
- [Bell et al., 2012] Bell, J., Almgren, A., Beckner, V., Day, M., Lijewski, M., Nonaka, A., and Zhang, W. (2012). Boxlib user’s guide. github.com/BoxLib-Codes/BoxLib.
- [Bell et al., 1989] Bell, J. B., Colella, P., and Glaz, H. M. (1989). A second-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 85(2) :257–283.
- [Bell et al., 1991] Bell, J. B., Colella, P., and Howell, L. H. (1991). An efficient second-order projection method for viscous incompressible flow. In *Proceedings of the Tenth AIAA Computational Fluid Dynamics Conference, AIAA*, page 24.
- [Belliard et al., 2016] Belliard, M., Chandesris, M., Dumas, J., Gorse, Y., Jamet, D., Josserand, C., and Mathieu, B. (2016). An analysis and an affordable regularization technique for the spurious force oscillations in the context of direct-forcing immersed boundary methods. *Computers & Mathematics with Applications*, 71(5) :1089–1113.
- [Benoit et al., 2015] Benoit, C., Péron, S., and Landier, S. (2015). Cassiopee : a cfd pre-and post-processing tool. *Aerospace Science and Technology*, 45 :272–283.
- [Berger et Rigoutsos, 1991] Berger, M. and Rigoutsos, I. (1991). An algorithm for point clustering and grid generation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5) :1278–1286.
- [Berger et Colella, 1989] Berger, M. J. and Colella, P. (1989). Local adaptive mesh refinement for shock hydrodynamics. *Journal of computational Physics*, 82(1) :64–84.
- [Berger et Olinger, 1984] Berger, M. J. and Olinger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of computational Physics*, 53(3) :484–512.
- [Berthelsen et Faltinsen, 2008] Berthelsen, P. A. and Faltinsen, O. M. (2008). A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries. *Journal of computational physics*, 227(9) :4354–4397.
- [Bigay, 2015] Bigay, P. (2015). *Développement d’un solveur faiblement compressible sur maillage cartésien pour les écoulements hydrodynamiques autour de corps*. Thèse de doctorat, Ecole Centrale de Nantes.
- [Bigay et al., 2017] Bigay, P., Oger, G., Guilcher, P.-M., and Le Touzé, D. (2017). A weakly-compressible cartesian grid approach for hydrodynamic flows. *Computer Physics Communications*, 220 :31–43.

- [Bigot et al., 2014] Bigot, B., Bonometti, T., Lacaze, L., and Thual, O. (2014). A simple immersed-boundary method for solid–fluid interaction in constant-and stratified-density flows. *Computers & Fluids*, 97 :126–142.
- [Botella et Peyret, 1998] Botella, O. and Peyret, R. (1998). Benchmark spectral results on the lid-driven cavity flow. *Computers & Fluids*, 27(4) :421–433.
- [Brandt, 1977] Brandt, A. (1977). Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138) :333–390.
- [Braza et al., 1986] Braza, M., Chassaing, P., and Minh, H. H. (1986). Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of fluid mechanics*, 165 :79–130.
- [Briggs et al., 2000] Briggs, W. L., Henson, V. E., and McCormick, S. F. (2000). *A multigrid tutorial*. SIAM.
- [Brown et al., 2001] Brown, D. L., Cortez, R., and Minion, M. L. (2001). Accurate projection methods for the incompressible navier–stokes equations. *Journal of computational physics*, 168(2) :464–499.
- [Brown et Lowe, 2005] Brown, J. D. and Lowe, L. L. (2005). Multigrid elliptic equation solver with adaptive mesh refinement. *Journal of Computational Physics*, 209(2) :582–598.
- [Bruneau et Saad, 2006] Bruneau, C.-H. and Saad, M. (2006). The 2d lid-driven cavity problem revisited. *Computers & Fluids*, 35(3) :326–348.
- [Burstedde et al., 2011] Burstedde, C., Wilcox, L. C., and Ghattas, O. (2011). p4est : Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3) :1103–1133.
- [Calhoun, 2002] Calhoun, D. (2002). A cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. *Journal of computational physics*, 176(2) :231–275.
- [Capdeville, 2006] Capdeville, G. (2006). *Modélisation numérique d’écoulements compressibles, partie I et II*. Cours de l’Ecole Centrale de Nantes.
- [CCSE, 2017] CCSE (2017). Amrex : Block-structured amr framework. <https://ccse.lbl.gov/AMReX>. Accessed :2017-01-12.
- [CERFACS, 2011] CERFACS (2011). The cerfacs co-vo test (convection of a vortex) for dns and les codes. <http://elearning.cerfacs.fr/numerical/benchmarks/vortex2d/index.php>. Accessed :2017-09-01.
- [Chella et al., 2015] Chella, M. A., Bihs, H., Myrhaug, D., and Muskulus, M. (2015). Breaking characteristics and geometric properties of spilling breakers over slopes. *Coastal Engineering*, 95 :4–19.
- [Chénier et al., 2015] Chénier, E., Eymard, R., Gallouët, T., and Herbin, R. (2015). An extension of the mac scheme to locally refined meshes : convergence analysis for the full tensor time-dependent navier–stokes equations. *Calcolo*, 52(1) :69–107.
- [Chénier et al., 2011] Chénier, R., Eymard, R., and Herbin, R. (2011). An extension of the mac scheme to some unstructured meshes. *Finite volumes for complex applications VI*, 1 :253–261.
- [Cheny et Botella, 2010] Cheny, Y. and Botella, O. (2010). 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.
- [Choi et Merkle, 1985] Choi, D. and Merkle, C. (1985). Application of time-iterative schemes to incompressible flow. *AIAA Journal*, 23 :1518–1524.

- [Choi et al., 2007] Choi, J.-I., Oberoi, R. C., Edwards, J. R., and Rosati, J. A. (2007). An immersed boundary method for complex incompressible flows. *Journal of Computational Physics*, 224(2) :757–784.
- [Choi et Merkle, 1993] Choi, Y.-H. and Merkle, C. L. (1993). The application of preconditioning in viscous flows. *Journal of Computational Physics*, 105(2) :207–223.
- [Chorin, 1967] Chorin, A. J. (1967). A numerical method for solving incompressible viscous flow problems. *J. Comp. Physics*, 2 :12–26.
- [Chorin, 1968] Chorin, A. J. (1968). Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104) :745–762.
- [Clain et al., 2011] Clain, S., Diot, S., and Loubère, R. (2011). A high-order finite volume method for systems of conservation laws—multi-dimensional optimal order detection (mood). *Journal of computational Physics*, 230(10) :4028–4050.
- [Clarke et al., 1986] Clarke, D. K., Hassan, H., and Salas, M. (1986). Euler calculations for multielement airfoils using cartesian grids. *AIAA journal*, 24(3) :353–358.
- [Cleary et al., 2000] Cleary, A. J., Falgout, R. D., Henson, V. E., Jones, J. E., Manteuffel, T. A., McCormick, S. F., Miranda, G. N., and Ruge, J. W. (2000). Robustness and scalability of algebraic multigrid. *SIAM Journal on Scientific Computing*, 21(5) :1886–1908.
- [Coirier, 1994] Coirier, W. J. (1994). *An adaptively-refined, Cartesian cell-based scheme for the Euler and Navier-Stokes equations*, volume 106754. University of Michigan PhD thesis.
- [Colella et Graves, 2000] Colella, P. and Graves, D. (2000). Chombo software package for amr applications-design document.
- [Couderc, 2007] Couderc, F. (2007). *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, Ecole nationale supérieure de l’aéronautique et de l’espace.
- [Coutanceau et Bouard, 1977] Coutanceau, M. and Bouard, R. (1977). Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. part 1. steady flow. *Journal of Fluid Mechanics*, 79(2) :231–256.
- [Dauvergne et al., 2008] Dauvergne, F., Ghidaglia, J.-M., Pascal, F., and Rovarch, J.-M. (2008). Renormalization of the numerical diffusion for an upwind finite volume method. application to the simulation of kelvin-helmholtz instability. In Eymard, R. and Hérard, J.-M., editors, *Finite Volumes for Complex Applications V*. John Wiley & Sons.
- [Delaux, 2009] Delaux, S. (2009). *Physical and numerical modelling of particle settlement in a turbulent flow : implication for the settlement of algal propagules*. Thèse de doctorat, University of Canterbury.
- [Dennis et Chang, 1970] Dennis, S. and Chang, G.-Z. (1970). Numerical solutions for steady flow past a circular cylinder at reynolds numbers up to 100. *Journal of Fluid Mechanics*, 42(3) :471–489.
- [DeZeeuw et Powell, 1993] DeZeeuw, D. and Powell, K. G. (1993). An adaptively refined cartesian mesh solver for the euler equations. *Journal of Computational Physics*, 104(1) :56–68.
- [Di Mascio et al., 2007] Di Mascio, A., Broglia, R., and Muscari, R. (2007). On the application of the single-phase level set method to naval hydrodynamic flows. *Computers & fluids*, 36(5) :868–886.
- [Di Mascio et al., 2014] Di Mascio, A., Muscari, R., and Dubbioso, G. (2014). On the wake dynamics of a propeller operating in drift. *Journal of Fluid Mechanics*, 754 :263–307.
- [Doradoux, 2017] Doradoux, A. (2017). *Simulation numérique d’écoulements diphasiques autour d’un solide mobile*. Thèse de doctorat, Université de Bordeaux.
- [Doran, 2013] Doran, A. (2013). *Occlusion culling et pipeline hybride CPU/GPU pour le rendu temps réel de scènes complexes pour la réalité virtuelle mobile*. Thèse de doctorat, Université de Toulouse, Université Toulouse III-Paul Sabatier.

- [Dubey et al., 2014] Dubey, A., Almgren, A., Bell, J., Berzins, M., Brandt, S., Bryan, G., Colella, P., Graves, D., Lijewski, M., Löffler, F., et al. (2014). A survey of high level frameworks in block-structured adaptive mesh refinement packages. *Journal of Parallel and Distributed Computing*, 74(12) :3217–3227.
- [Ducros et al., 1999] Ducros, F., Ferrand, V., Nicoud, F., Weber, C., Darracq, D., Gacherieu, C., and Poinsot, T. (1999). Large-eddy simulation of the shock/turbulence interaction. *Journal of Computational Physics*, 152(2) :517–549.
- [Ducros et al., 2000] Ducros, F., Laporte, F., Souleres, T., Guinot, V., Moinat, P., and Caruelle, B. (2000). High-order fluxes for conservative skew-symmetric-like schemes in structured meshes : application to compressible flows. *Journal of Computational Physics*, 161(1) :114–139.
- [Elie, 2015] Elie, B. (2015). *Modélisation numérique de sillages lointains d’hydroliennes par une approche volumes finis faiblement compressible*. Thèse de doctorat, Ecole Centrale de Nantes.
- [Elie et al., 2017] Elie, B., Oger, G., Guillermin, P.-E., and Alessandrini, B. (2017). Simulation of horizontal axis tidal turbine wakes using a weakly-compressible cartesian hydrodynamic solver with local mesh refinement. *Renewable Energy*, 108(Supplement C) :336 – 354.
- [Fadlun et al., 2000] Fadlun, E., Verzicco, R., Orlandi, P., and Mohd-Yusof, J. (2000). Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1) :35–60.
- [Falgout et Yang, 2002] Falgout, R. D. and Yang, U. M. (2002). hypre : A library of high performance preconditioners. In *International Conference on Computational Science*, pages 632–641. Springer.
- [Ferziger et Perić, 1996] Ferziger, J. and Perić, M. (1996). *Computational Methods for Fluid Dynamics*. Numerical methods : research and development. Springer.
- [Fujisawa et al., 2000] Fujisawa, J., Ukon, Y., Kume, K., and Takeshi, H. (2000). Local velocity field measurements around the kcs model (sri m.s.no.631) in the sri 400m towing tank. Ship performance division report no. 00-003-02, The Ship Research Institute of Japan.
- [Ghia et al., 1982] Ghia, U., Ghia, K. N., and Shin, C. (1982). High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of computational physics*, 48(3) :387–411.
- [Ghias et al., 2007] Ghias, R., Mittal, R., and Dong, H. (2007). A sharp interface immersed boundary method for compressible viscous flows. *Journal of Computational Physics*, 225(1) :528–553.
- [Ghidaglia, 2001] Ghidaglia, J.-M. (2001). Flux schemes for solving nonlinear systems of conservation laws. *Innovative Methods for Numerical Solution of Partial Differential Equations*, Chattot JJ and Hafez M. Eds, pages 232–242.
- [Ghidaglia et al., 1996] Ghidaglia, J.-M., Kumbaro, A., and Le Coq, G. (1996). Une méthode volumes finis à flux caractéristiques pour la résolution numérique des systèmes hyperboliques de lois de conservation. *Comptes rendus de l’Académie des sciences. Série 1, Mathématique*, 322(10) :981–988.
- [Ghosh et Baeder, 2012] Ghosh, D. and Baeder, J. D. (2012). Compact reconstruction schemes with weighted eno limiting for hyperbolic conservation laws. *SIAM Journal on Scientific Computing*, 34(3) :A1678–A1706.
- [Glowinski et al., 2001] Glowinski, R., Pan, T., Hesla, T., Joseph, D., and Periaux, J. (2001). A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies : application to particulate flow. *Journal of Computational Physics*, 169(2) :363–426.
- [Goda, 1979] Goda, K. (1979). A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows. *Journal of Computational Physics*, 30(1) :76–95.
- [Godunov, 1959] Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3) :271–306.

- [Goldstein et al., 1993] Goldstein, D., Handler, R., and Sirovich, L. (1993). Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105(2) :354–366.
- [Gorsse et al., 2012] Gorsse, Y., Iollo, A., Telib, H., and Weynans, L. (2012). A simple second order cartesian scheme for compressible euler flows. *Journal of Computational Physics*, 231(23) :7780–7794.
- [Grove et al., 1964] Grove, A. S., Shair, F., and Petersen, E. (1964). An experimental investigation of the steady separated flow past a circular cylinder. *Journal of Fluid Mechanics*, 19(1) :60–80.
- [Guermont et al., 2006] Guermont, J.-L., Mineev, P., and Shen, J. (2006). An overview of projection methods for incompressible flows. *Computer methods in applied mechanics and engineering*, 195(44) :6011–6045.
- [Guillard et Viozat, 1999] Guillard, H. and Viozat, C. (1999). On the behaviour of upwind schemes in the low mach number limit. *Computers & fluids*, 28(1) :63–86.
- [Guittet et al., 2015] Guittet, A., Theillard, M., and Gibou, F. (2015). A stable projection method for the incompressible navier–stokes equations on arbitrary geometries and adaptive quad/octrees. *Journal of Computational Physics*, 292 :215–238.
- [Hachem et al., 2012] Hachem, E., Kloczko, T., Dignonnet, H., and Coupez, T. (2012). Stabilized finite element solution to handle complex heat and fluid flows in industrial furnaces using the immersed volume method. *International Journal for Numerical Methods in Fluids*, 68(1) :99–121.
- [Hackbusch, 2013] Hackbusch, W. (2013). *Multi-grid methods and applications*, volume 4. Springer Science & Business Media.
- [Harlow et Welch, 1965] Harlow, F. and Welch, J. (1965). Numerical calculation of time–dependent viscous incompressible flow of fluid with a free surface. *The physics of fluids*, 8 :2182–2189.
- [Harten et al., 1987] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. R. (1987). Uniformly high order accurate essentially non-oscillatory schemes, iii. *Journal of computational physics*, 71(2) :231–303.
- [Herbin, 2011] Herbin, R. (2011). Analyse numérique des équations aux dérivées partielles.
- [Hermeline, 2000] Hermeline, F. (2000). A finite volume method for the approximation of diffusion operators on distorted meshes. *Journal of computational Physics*, 160(2) :481–499.
- [Hines, 2008] Hines, J. (2008). A comparative study of the simple and fractional step time integration methods for transient incompressible flows. Master’s thesis, University of Waterloo.
- [Hirsch, 2007] Hirsch, C. (2007). *Numerical computation of internal and external flows : The fundamentals of computational fluid dynamics*. Butterworth-Heinemann.
- [H.J.L. van der Heiden et Luppès, 2013] H.J.L. van der Heiden, A. V. and Luppès, R. (2013). A quasi-3d extension of the ls stag cut-cell immersed boundary method. In *Joint EUROMECH/ERCOTAC Colloquium 549*, Leiden, The Netherlands.
- [Hubert et Tesson, 2017] Hubert, F. and Tesson, R. (2017). Weno scheme for transport equation on unstructured grids with a ddfv approach. In *International Conference on Finite Volumes for Complex Applications*, pages 13–21. Springer.
- [Issa, 1986] Issa, R. I. (1986). Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of computational physics*, 62(1) :40–65.
- [Ivan et Groth, 2007] Ivan, L. and Groth, C. (2007). High-order central eno finite-volume scheme with adaptive mesh refinement. In *18th AIAA Computational Fluid Dynamics Conference*, page 4323.
- [Jacobson et al., 2013] Jacobson, A., Kavan, L., and Sorkine-Hornung, O. (2013). Robust inside-outside segmentation using generalized winding numbers. *ACM Transactions on Graphics (TOG)*, 32(4) :33.

- [Jameson, 1995] Jameson, A. (1995). Analysis and design of numerical schemes for gas dynamics, 1 : artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *International Journal of Computational Fluid Dynamics*, 4(3-4) :171–218.
- [Jameson et al., 1981] Jameson, A. et al. (1981). Numerical solutions of the euler equations by finite volume methods using runge-kutta time-stepping schemes. In *AIAA, Fluid and Plasma Dynamics Conference, 14th, Palo Alto, CA, June 23-25, 1981*. 15 p.
- [Jiang et Shu, 1996] Jiang, G.-S. and Shu, C.-W. (1996). Efficient implementation of weighted eno schemes. *Journal of computational physics*, 126(1) :202–228.
- [Johansen et Colella, 1998] Johansen, H. and Colella, P. (1998). A cartesian grid embedded boundary method for poisson’s equation on irregular domains. *Journal of Computational Physics*, 147(1) :60–85.
- [Kadioglu et al., 2008] Kadioglu, S. Y., Klein, R., and Minion, M. L. (2008). A fourth-order auxiliary variable projection method for zero-mach number gas dynamics. *Journal of Computational Physics*, 227(3) :2012–2043.
- [Kajishima et Takiguchi, 2002] Kajishima, T. and Takiguchi, S. (2002). Interaction between particle clusters and particle-induced turbulence. *International Journal of Heat and Fluid Flow*, 23(5) :639–646.
- [Kampanis et Ekaterinaris, 2006] Kampanis, N. A. and Ekaterinaris, J. A. (2006). A staggered grid, high-order accurate method for the incompressible navier–stokes equations. *Journal of Computational Physics*, 215(2) :589–613.
- [Karypis et Kumar, 1998] Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1) :359–392.
- [Kempe et Fröhlich, 2012] Kempe, T. and Fröhlich, J. (2012). An improved immersed boundary method with direct forcing for the simulation of particle laden flows. *Journal of Computational Physics*, 231(9) :3663–3684.
- [Khadra et al., 1996] Khadra, K., Angot, P., Caltagirone, J., and Morel, P. (1996). Concept de zoom adaptatif en architecture multigrille locale ; étude comparative des méthodes ldc, fac et fic. *ESAIM : Mathematical Modelling and Numerical Analysis*, 30(1) :39–82.
- [Khokhlov, 1998] Khokhlov, A. M. (1998). Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations. *Journal of Computational Physics*, 143(2) :519–543.
- [Kim et al., 2001] Kim, J., Kim, D., and Choi, H. (2001). An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, 171(1) :132–150.
- [Kim et Moin, 1985] Kim, J. and Moin, P. (1985). Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics*, 59(2) :308–323.
- [Kleefsman et al., 2005] Kleefsman, K., Fekken, G., Veldman, A., Iwanowski, B., and Buchner, B. (2005). A volume-of-fluid based simulation method for wave impact problems. *Journal of computational physics*, 206(1) :363–393.
- [Lax et Richtmyer, 1956] Lax, P. D. and Richtmyer, R. D. (1956). Survey of the stability of linear finite difference equations. *Communications on pure and applied mathematics*, 9(2) :267–293.
- [Le et Moin, 1991] Le, H. and Moin, P. (1991). An improvement of fractional step methods for the incompressible navier-stokes equations. *Journal of computational physics*, 92(2) :369–379.
- [Le Touzé et al., 2008] Le Touzé, D., Oger, G., and Alessandrini, B. (2008). Smoothed particle hydrodynamics simulation of fast ship flows. In *Proceedings of the 27th Symposium on Naval Hydrodynamics*.
- [Lee et al., 2008] Lee, E.-S., Moulinec, C., Xu, R., Violeau, D., Laurence, D., and Stansby, P. (2008). Comparisons of weakly compressible and truly incompressible algorithms for the sph mesh free particle method. *Journal of computational physics*, 227(18) :8417–8436.

- [Lee et al., 2011] Lee, J., Kim, J., Choi, H., and Yang, K.-S. (2011). Sources of spurious force oscillations from an immersed boundary method for moving-body problems. *Journal of computational physics*, 230(7) :2677–2695.
- [Lee et LeVeque, 2003] Lee, L. and LeVeque, R. J. (2003). An immersed interface method for incompressible navier–stokes equations. *SIAM Journal on Scientific Computing*, 25(3) :832–856.
- [Leroy, 2012] Leroy, C. (2012). *Développement d’une méthode volumes finis à flux caractéristiques pour la simulation d’écoulements en hydrodynamique*. Thèse de doctorat, Ecole Centrale de Nantes.
- [LeVeque et Li, 1994] LeVeque, R. J. and Li, Z. (1994). The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4) :1019–1044.
- [Levy et al., 1999] Levy, D., Puppo, G., and Russo, G. (1999). Central weno schemes for hyperbolic systems of conservation laws. *ESAIM : Mathematical Modelling and Numerical Analysis*, 33(3) :547–571.
- [Linnick et Fasel, 2005] Linnick, M. N. and Fasel, H. F. (2005). A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*, 204(1) :157–192.
- [Liu, 2011] Liu, Q. (2011). A stable and accurate projection method on a locally refined staggered mesh. *International Journal for Numerical Methods in Fluids*, 67(1) :74–92.
- [Liu et Vasilyev, 2007] Liu, Q. and Vasilyev, O. V. (2007). A brinkman penalization method for compressible flows in complex geometries. *Journal of Computational Physics*, 227(2) :946–966.
- [Liu et al., 1994] Liu, X.-D., Osher, S., and Chan, T. (1994). Weighted essentially non-oscillatory schemes. *Journal of computational physics*, 115(1) :200–212.
- [LLNL, 1997] LLNL (1997). Samrai : Structured adaptive mesh refinement application infrastructure. <http://www.llnl.gov/CASC/SAMRAI>. Accessed :2017-12-01.
- [Loppi et al., 2018] Loppi, N., Witherden, F., Jameson, A., and Vincent, P. (2018). A high-order cross-platform incompressible navier–stokes solver via artificial compressibility with application to a turbulent jet. *Computer Physics Communications*, 233 :193 – 205.
- [Losasso et al., 2004] Losasso, F., Gibou, F., and Fedkiw, R. (2004). Simulating water and smoke with an octree data structure. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 457–462. ACM.
- [MacNeice et al., 2000] MacNeice, P., Olson, K. M., Mobarry, C., De Fainchtein, R., and Packer, C. (2000). Paramesh : A parallel adaptive mesh refinement community toolkit. *Computer physics communications*, 126(3) :330–354.
- [Marechal, 2016] Marechal, L. (2016). Quick and easy spatial localization with octree. <https://github.com/LoicMarechal/libOL>. Accessed :2018-03-30.
- [Martin et Colella, 2000] Martin, D. F. and Colella, P. (2000). A cell-centered adaptive projection method for the incompressible euler equations. *Journal of computational Physics*, 163(2) :271–312.
- [Mauch, 2000] Mauch, S. (2000). A fast algorithm for computing the closest point and distance transform.
- [McCormick, 1984] McCormick, S. (1984). Fast adaptive composite grid (fac) methods : Theory for the variational case. In *Defect correction methods*, pages 115–121. Springer.
- [McCormick et Thomas, 1986] McCormick, S. and Thomas, J. (1986). The fast adaptive composite grid (fac) method for elliptic equations. *Mathematics of Computation*, 46(174) :439–456.
- [Merkle, 1987] Merkle, C. (1987). Time-accurate unsteady incompressible flow algorithms based on artificial compressibility. In *8th Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics.

- [Min et Gibou, 2006] Min, C. and Gibou, F. (2006). A second order accurate projection method for the incompressible navier–stokes equations on non-graded adaptive grids. *Journal of Computational Physics*, 219(2) :912–929.
- [Min et al., 2006] Min, C., Gibou, F., and Cenicerros, H. D. (2006). A supra-convergent finite difference scheme for the variable coefficient poisson equation on non-graded grids. *Journal of Computational Physics*, 218(1) :123–140.
- [Minion, 1996] Minion, M. L. (1996). A projection method for locally refined grids. *Journal of Computational Physics*, 127(1) :158–178.
- [Minion, 2004] Minion, M. L. (2004). Semi-implicit projection methods for incompressible flow based on spectral deferred corrections. *Applied numerical mathematics*, 48(3) :369–387.
- [Minion et al., 2003] Minion, M. L. et al. (2003). Semi-implicit spectral deferred correction methods for ordinary differential equations. *Communications in Mathematical Sciences*, 1(3) :471–500.
- [Mittal et al., 2008] Mittal, R., Dong, H., Bozkurtas, M., Najjar, F., Vargas, A., and von Loebbecke, A. (2008). A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of computational physics*, 227(10) :4825–4852.
- [Mittal et Iaccarino, 2005] Mittal, R. and Iaccarino, G. (2005). Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37 :239–261.
- [Mohd-Yusof, 1997] Mohd-Yusof, J. (1997). Combined immersed–boundary /b–spline methods for simulations of flow in complex geometries. *Annual Research Briefs*, 317.
- [Monaghan, 1992] Monaghan, J. J. (1992). Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30(1) :543–574.
- [Morinishi et al., 1998] Morinishi, Y., Lund, T. S., Vasilyev, O. V., and Moin, P. (1998). Fully conservative higher order finite difference schemes for incompressible flow. *Journal of computational physics*, 143(1) :90–124.
- [Ng et al., 2009] Ng, Y. T., Min, C., and Gibou, F. (2009). An efficient fluid–solid coupling algorithm for single-phase flows. *Journal of Computational Physics*, 228(23) :8807–8829.
- [Nonaka et al., 2010] Nonaka, A., Almgren, A., Bell, J., Lijewski, M., Malone, C., and Zingale, M. (2010). Maestro : An adaptive low mach number hydrodynamics algorithm for stellar flows. *The Astrophysical Journal Supplement Series*, 188(2) :358.
- [Oger et al., 2016] Oger, G., Le Touzé, D., Guibert, D., De Lefte, M., Biddiscombe, J., Soumagne, J., and Piccinalli, J.-G. (2016). On distributed memory mpi-based parallelization of sph codes in massive hpc context. *Computer Physics Communications*, 200 :1–14.
- [Palacios et al., 2013] Palacios, F., Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economon, T. D., Lonkar, A. K., Lukaczyk, T. W., Taylor, T. W., and Alonso, J. J. (2013). Stanford university unstructured (su2) : An open-source integrated computational environment for multi-physics simulation and design. *AIAA Paper*, 287 :2013.
- [Pan, 2006] Pan, D. (2006). An immersed boundary method for incompressible flows using volume of body function. *International journal for numerical methods in fluids*, 50(6) :733–750.
- [Patankar, 1980] Patankar, S. (1980). *Numerical heat transfer and fluid flow*. CRC press.
- [Patankar et Spalding, 1972] Patankar, S. V. and Spalding, D. B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International journal of heat and mass transfer*, 15(10) :1787–1806.
- [Pellegrini et Roman, 1996] Pellegrini, F. and Roman, J. (1996). Scotch : A software package for static mapping by dual recursive bipartitioning of process and architecture graphs. In *High-Performance Computing and Networking*, pages 493–498. Springer.

- [Pereira et al., 2001] Pereira, J. M., Kobayashi, M., and Pereira, J. C. (2001). A fourth-order-accurate finite volume compact method for the incompressible navier–stokes solutions. *Journal of Computational Physics*, 167(1) :217–243.
- [Peskin, 1972] Peskin, C. S. (1972). Flow patterns around heart valves : a numerical method. *Journal of computational physics*, 10(2) :252–271.
- [Popinet, 2000] Popinet, S. (2000). The GNU Triangulated Surface Library. <http://gts.sourceforge.net>. Accessed :2017-01-09.
- [Popinet, 2003a] Popinet, S. (2003a). Gerris : a tree-based adaptive solver for the incompressible euler equations in complex geometries. *Journal of Computational Physics*, 190(2) :572–600.
- [Popinet, 2003b] Popinet, S. (2003b). Gerris Flow Solver. <http://gfs.sourceforge.net>. Accessed :2017-01-12.
- [Posa et Balaras, 2018] Posa, A. and Balaras, E. (2018). Large-eddy simulations of a notional submarine in towed and self-propelled configurations. *Computers & Fluids*, 165 :116–126.
- [Rannacher, 1992] Rannacher, R. (1992). On chorin’s projection method for the incompressible navier–stokes equations. *Lecture notes in mathematics*, 1530 :167–183.
- [Rider, 1995] Rider, W. J. (1995). Approximate projection methods for incompressible flow : Implementation, variants and robustness. In *LANL UNCLASSIFIED REPORT LA-UR-94-2000, LOS ALAMOS NATIONAL LABORATORY*. Citeseer.
- [Roma et al., 1999] Roma, A. M., Peskin, C. S., and Berger, M. J. (1999). An adaptive version of the immersed boundary method. *Journal of computational physics*, 153(2) :509–534.
- [Rossi et al., 2015] Rossi, E., Colagrossi, A., Bouscasse, B., and Graziani, G. (2015). The diffused vortex hydrodynamics method. *Communications in Computational Physics*, 18(2) :351–379.
- [Rossi et al., 2018] Rossi, E., Colagrossi, A., Oger, G., and Le Touzé, D. (2018). Multiple bifurcations of the flow over stalled airfoils when changing the reynolds number. *Journal of Fluid Mechanics*, 846 :356–391.
- [Ruge et Stüben, 1987] Ruge, J. W. and Stüben, K. (1987). Algebraic multigrid. *Multigrid methods*, 3(13) :73–130.
- [Rycroft, 2009] Rycroft, C. H. (2009). voro++ : A three-dimensional voronoi cell library in c++. *Chaos : An Interdisciplinary Journal of Nonlinear Science*, 19(4) :041111.
- [Saad et Schultz, 1986] Saad, Y. and Schultz, M. H. (1986). Gmres : A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3) :856–869.
- [Sampath et al., 2008] Sampath, R. S., Adavani, S. S., Sundar, H., Lashuk, I., and Biros, G. (2008). Dendro : parallel algorithms for multigrid and amr methods on 2 : 1 balanced octrees. In *High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for*, pages 1–12. IEEE.
- [Sanderse et Koren, 2012] Sanderse, B. and Koren, B. (2012). Accuracy analysis of explicit runge–kutta methods applied to the incompressible navier–stokes equations. *Journal of Computational Physics*, 231(8) :3041–3063.
- [Science, 1997] Science, C. (1997). CONVERGE CFD. <https://convergecf.com/>. Accessed :2018-01-06.
- [Seo et Mittal, 2011] Seo, J. H. and Mittal, R. (2011). A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *Journal of computational physics*, 230(19) :7347–7363.

- [Sethian, 1996] Sethian, J. A. (1996). A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4) :1591–1595.
- [Shadloo et al., 2012] Shadloo, M. S., Zainali, A., Yildiz, M., and Suleman, A. (2012). A robust weakly compressible sph method and its comparison with an incompressible sph. *International Journal for Numerical Methods in Engineering*, 89(8) :939–956.
- [Shui, 2015] Shui, P. (2015). *A novel immersed boundary method for direct numerical simulations of solid-fluid flows*. Thèse de doctorat, University of Edinburgh.
- [Stone, 1968] Stone, H. L. (1968). Iterative solution of implicit approximations of multidimensional partial differential equations. *SIAM Journal on Numerical Analysis*, 5(3) :530–558.
- [Sussman et Dommermuth, 2014] Sussman, M. and Dommermuth, D. G. (2014). The numerical simulation of ship waves using cartesian grid methods. *arXiv preprint arXiv :1410.1952*.
- [Taira et Colonius, 2007] Taira, K. and Colonius, T. (2007). The immersed boundary method : a projection approach. *Journal of Computational Physics*, 225(2) :2118–2137.
- [Takiguchi et al., 1999] Takiguchi, S., Kajishima, T., and Miyake, Y. (1999). Numerical scheme to resolve the interaction between solid particles and fluid turbulence. *JSME International Journal Series B Fluids and Thermal Engineering*, 42(3) :411–418.
- [Temam, 1969] Temam, R. (1969). Sur l’approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (ii). *Archive for Rational Mechanics and Analysis*, 33(5) :377–385.
- [The CGAL Project, 2018] The CGAL Project (2018). *CGAL User and Reference Manual*. CGAL Editorial Board, 4.12 edition.
- [Timmermans et al., 1996] Timmermans, L., Mineev, P., and Van De Vosse, F. (1996). An approximate projection scheme for incompressible flow using spectral elements. *International journal for numerical methods in fluids*, 22(7) :673–688.
- [Tóth et Roe, 2002] Tóth, G. and Roe, P. (2002). Divergence-and curl-preserving prolongation and restriction formulas. *Journal of Computational Physics*, 180(2) :736–750.
- [Tritton, 1959] Tritton, D. (1959). Experiments on the flow past a circular cylinder at low reynolds numbers. *Journal of Fluid Mechanics*, 6(4) :547–567.
- [Tseng et Ferziger, 2003] Tseng, Y.-H. and Ferziger, J. H. (2003). A ghost-cell immersed boundary method for flow in complex geometry. *Journal of computational physics*, 192(2) :593–623.
- [Tu et al., 2005] Tu, T., O’Hallaron, D. R., and Ghattas, O. (2005). Scalable parallel octree meshing for terascale applications. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pages 4–4. IEEE.
- [Tukel, 1987] Turkel, E. (1987). Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of computational physics*, 72(2) :277–298.
- [Uhlmann, 2005] Uhlmann, M. (2005). An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2) :448–476.
- [van der Plas, 2016] van der Plas, P. (2016). *Local grid refinement for free-surface flow simulations*. Thèse de doctorat, University of Groningen.
- [Van Doormaal et Raithby, 1984] Van Doormaal, J. and Raithby, G. (1984). Enhancements of the simple method for predicting incompressible fluid flows. *Numerical heat transfer*, 7(2) :147–163.
- [Van Kan, 1986] Van Kan, J. (1986). A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing*, 7(3) :870–891.
- [Versteeg et Malalasekera, 2007] Versteeg, H. K. and Malalasekera, W. (2007). *An introduction to computational fluid dynamics : the finite volume method*. Pearson Education.

- [Vincent et al., 2014] Vincent, S., De Motta, J. C. B., Sarthou, A., Estivalezes, J.-L., Simonin, O., and Clement, E. (2014). A lagrangian vof tensorial penalty method for the dns of resolved particle-laden flows. *Journal of Computational Physics*, 256 :582–614.
- [Vittoz et al., 2017] Vittoz, L., Oger, G., Li, Z., de Leffe, M., and Le Touzé, D. (2017). *A High-Order Finite Volume Solver on Locally Refined Cartesian Meshes—Benchmark Session*, pages 73–89. Springer International Publishing.
- [VonNeumann et Richtmyer, 1950] VonNeumann, J. and Richtmyer, R. D. (1950). A method for the numerical calculation of hydrodynamic shocks. *Journal of applied physics*, 21(3) :232–237.
- [Wackers et al., 2013] Wackers, J., Deng, G., Guilmineau, E., Leroyer, A., Queutey, P., and Visonneau, M. (2013). Anisotropic mesh refinement in ship flow simulation with free surface. In *MARINE 2011, IV International Conference on Computational Methods in Marine Engineering*, pages 273–284. Springer.
- [Wackers et al., 2014a] Wackers, J., Deng, G., Guilmineau, E., Leroyer, A., Queutey, P., and Visonneau, M. (2014a). Combined refinement criteria for anisotropic grid refinement in free-surface flow simulation. *Computers & Fluids*, 92 :209–222.
- [Wackers et al., 2015] Wackers, J., Deng, G., Guilmineau, E., Leroyer, A., Queutey, P., and Visonneau, M. (2015). Creating free-surface flow grids with automatic grid refinement. In *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, pages 305–325. Springer.
- [Wackers et al., 2014b] Wackers, J., Guilmineau, E., Palmieri, A., Queutey, P., della Nautica, C.-M. T., and UNIBO, V. S. (2014b). Hessian-based grid refinement for the simulation of surface-piercing hydrofoils. In *17th Numerical Towing Tank Symposium (ed. R. Bensow)*, Marstrand, Sweden.
- [Weinan et Liu, 1995] Weinan, E. and Liu, J.-G. (1995). Projection method i : convergence and numerical boundary layers. *SIAM journal on numerical analysis*, pages 1017–1057.
- [Wesseling, 1995] Wesseling, P. (1995). Introduction to multigrid methods. Technical report, INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING HAMPTON VA.
- [Wu et al., 2004] Wu, M.-H., Wen, C.-Y., Yen, R.-H., Weng, M.-C., and Wang, A.-B. (2004). Experimental and numerical study of the separation angle for flow around a circular cylinder at low reynolds number. *Journal of Fluid Mechanics*, 515 :233–260.
- [Xu et Wang, 2006] Xu, S. and Wang, Z. J. (2006). An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics*, 216(2) :454–493.
- [Yamakawa et Shimada, 2009] Yamakawa, S. and Shimada, K. (2009). Removing self intersections of a triangular mesh by edge swapping, edge hammering, and face lifting. In *Proceedings of the 18th International Meshing Roundtable*, pages 13–29. Springer.
- [Yang et Balaras, 2006] Yang, J. and Balaras, E. (2006). An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics*, 215(1) :12–40.
- [Yang et Stern, 2012] Yang, J. and Stern, F. (2012). A simple and efficient direct forcing immersed boundary framework for fluid–structure interactions. *Journal of Computational Physics*, 231(15) :5029–5061.
- [Yang et Stern, 2013] Yang, J. and Stern, F. (2013). Robust and efficient setup procedure for complex triangulations in immersed boundary simulations. *Journal of Fluids Engineering*, 135(10) :101107.
- [Yang et Stern, 2015] Yang, J. and Stern, F. (2015). A non-iterative direct forcing immersed boundary method for strongly-coupled fluid–solid interactions. *Journal of Computational Physics*, 295 :779–804.
- [Yang et al., 2015] Yang, K.-K., Kim, Y., and Nam, B.-W. (2015). Cartesian-grid-based computational analysis for added resistance in waves. *Journal of Marine Science and Technology*, 20(1) :155–170.

- [Yang et al., 2002] Yang, U. M. et al. (2002). Boomeramg : a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41(1) :155–177.
- [Ye et al., 1999] Ye, T., Mittal, R., Udaykumar, H., and Shyy, W. (1999). An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of computational physics*, 156(2) :209–240.
- [Young, 2014] Young, D. M. (2014). *Iterative solution of large linear systems*. Elsevier.
- [Yuki et al., 2007] Yuki, Y., Takeuchi, S., and Kajishima, T. (2007). Efficient immersed boundary method for strong interaction problem of arbitrary shape object with the self-induced flow. *Journal of Fluid Science and Technology*, 2(1) :1–11.
- [Zhang, 2014] Zhang, Q. (2014). A fourth-order approximate projection method for the incompressible navier–stokes equations on locally-refined periodic domains. *Applied Numerical Mathematics*, 77 :16–30.
- [Zuzio, 2010] Zuzio, D. (2010). *Direct numerical simulation of two phase flows with adaptive mesh refinement*. PhD thesis, Ecole nationale superieure de l’aeronautique et de l’espace.
- [Zuzio et Estivalezes, 2011] Zuzio, D. and Estivalezes, J. (2011). An efficient block parallel amr method for two phase interfacial flow simulations. *Computers & Fluids*, 44(1) :339–357.

Annexe A

Schéma JST et schémas similaires

Schéma Jameson-Schmidt-Turkel

Le schéma JST - du nom des trois collaborateurs qui ont proposé ce schéma : Jameson, Schmidt et Turkel - a été introduit au début des années 1980 afin de résoudre les équations d'Euler [Jameson et al., 1981]. L'objectif initial était de proposer un schéma de type volumes finis à même de pouvoir simuler un écoulement transsonique avec la formation éventuelle de chocs. Il s'agit là d'une rupture dans l'évolution de la mécanique des fluides numérique. En effet la résolution d'écoulements aérodynamiques transsoniques autour de géométries complexes n'était envisagée alors qu'avec des méthodes potentielles. Les quelques schémas employés à cette époque pour résoudre les équations d'Euler produisaient des oscillations non voulues au voisinage des chocs, comme le schéma populaire de MacCormack. Le schéma JST a alors connu un engouement fulgurant pour la capture de chocs. Au point d'être adopté massivement dans la communauté compressible, dépassant par là-même le cadre d'utilisation pour lequel il a été conçu. On le retrouve dans de nombreux codes commerciaux ou open source (comme SU2, développé principalement à Stanford [Palacios et al., 2013]).

L'idée du schéma est d'introduire de la viscosité artificielle pour stabiliser le calcul centré à l'ordre 2 du flux. Le concept n'est en soit pas novateur puisque la notion de viscosité artificielle comme élément de stabilisation remonte aux années 1950 et aux travaux de Richtmyer et Von Neumann [VonNeumann et Richtmyer, 1950]. L'originalité du schéma JST est d'introduire simultanément une viscosité d'ordre 2 et d'ordre 4 pour favoriser la convergence et prévenir l'apparition d'instabilités numériques.

Prenons l'équation d'Euler en dimension 1 exprimée sous forme conservative :

$$\frac{\partial w}{\partial t} + \frac{\partial F(W)}{\partial x} = 0 \quad (\text{A.1})$$

En volumes finis, la forme semi-discrète s'écrit :

$$\frac{\partial w_i}{\partial t} + \frac{F_{i+1/2} - F_{i-1/2}}{\Delta x} = 0 \quad (\text{A.2})$$

Le flux numérique à l'interface $i + 1/2$ est calculé par un schéma centré d'ordre 2 et un terme additionnel :

$$F_{i+1/2} = \frac{1}{2}(F_i + F_{i+1}) - d_{i+1/2} \quad (\text{A.3})$$

$d_{i+1/2}$ est le terme de dissipation. Il est décomposé en un terme d'ordre 2 et un terme d'ordre 4 :

$$d_{i+1/2} = d_{i+1/2}^{(2)} + d_{i+1/2}^{(4)} \quad (\text{A.4})$$

L'équation semi-discrète devient :

$$\frac{\partial w_i}{\partial t} + \frac{F_{i+1} - F_{i-1}}{2\Delta x} = \frac{d_{i+1/2}^{(2)} - d_{i-1/2}^{(2)}}{\Delta x} + \frac{d_{i+1/2}^{(4)} - d_{i-1/2}^{(4)}}{\Delta x} \quad (\text{A.5})$$

Le flux de viscosité artificielle d'ordre 2 est calculé par la formule :

$$d_{i+1/2}^{(2)} = \epsilon_{i+1/2}^{(2)} \lambda_{i+1/2} (u_{i+1} - u_i) \quad (\text{A.6})$$

Ce qui donne :

$$\frac{d_{i+1/2}^{(2)} - d_{i-1/2}^{(2)}}{\Delta x} = \epsilon^{(2)} \lambda \Delta x \frac{\partial^2 u}{\partial x^2} \quad (\text{A.7})$$

λ est la vitesse maximale de propagation des ondes acoustiques et tient le rôle de viscosité numérique comme pour les solveurs de Riemann :

$$\lambda_{i+1/2} = \max(|u_i| + c_i, |u_{i+1}| + c_{i+1}) \quad (\text{A.8})$$

ϵ est un paramètre sensible aux variations de pression.

$$\epsilon_{i+1/2}^{(2)} = \kappa^{(2)} \max(\nu_i, \nu_{i+1}) \quad (\text{A.9})$$

avec

$$\nu_i = \frac{|p_{i-1} - 2p_i + p_{i+1}|}{|p_{i-1} + 2p_i + p_{i+1}|} \quad (\text{A.10})$$

$\kappa^{(2)}$ est une constante prise égale à 1/4. Le flux visqueux d'ordre 4 s'écrit de façon analogue :

$$d_{i+1/2}^{(4)} = -\epsilon_{i+1/2}^{(4)} \lambda_{i+1/2} (u_{i+2} - 3u_{i+1} + 3u_j - u_{j-1}) \quad (\text{A.11})$$

Soit

$$\frac{d_{i+1/2}^{(4)} - d_{i-1/2}^{(4)}}{\Delta x} = -\epsilon^{(4)} \lambda \Delta x \frac{\partial^4 u}{\partial x^4} \quad (\text{A.12})$$

avec

$$\epsilon_{i+1/2}^{(4)} = \max(0, (\kappa^{(4)} - \epsilon_{i+1/2}^{(2)})) \quad (\text{A.13})$$

et $\kappa^{(4)} = 1/256$

La viscosité d'ordre 2 permet de faire chuter localement l'ordre de convergence du schéma à l'ordre 1 au voisinage d'un choc pour le capturer. Le terme de viscosité d'ordre 4 permet de filtrer les hautes fréquences et de favoriser la convergence.

Comme tout schéma à viscosité artificielle, l'efficacité du schéma JST repose sur le contrôle de la diffusion introduite. Elle doit être suffisante pour stabiliser le schéma centré d'ordre 2 mais pas trop importante pour ne pas freiner la convergence. Le paramètre ν permet de contrôler cette diffusion. Là où l'écoulement est régulier, $\nu \approx 0$ et le schéma est pleinement d'ordre 2. Au voisinage d'un choc, la quantité de viscosité numérique introduite est considérablement augmentée afin de le capturer correctement.

Par ailleurs, le recours à une méthode accélération de type multigrille, permet au schéma JST de converger rapidement tout en amortissant les instabilités numériques qui peuvent apparaître [Jameson, 1995]. Bien que conçu originellement pour résoudre des écoulements stationnaires, le schéma s'adapte parfaitement dans un contexte de résolution instationnaire. Des schémas particuliers de type Runge-Kutta sont privilégiés pour renforcer le domaine de stabilité du schéma JST qui ne peut être utilisé avec un simple schéma d'Euler explicite.

Le schéma JST a fait l'objet de nombreuses études qui ont engendré autant de variantes.

Schémas d'ordre supérieur

Plus généralement, il est possible de montrer que le schéma JST appartient à une famille générique de schémas dits "skew-symmetric" (anti-symétriques). La particularité de ces schémas skew-symmetric est de chercher à préserver le caractère anti-symétrique du terme convectif des équations de Navier-Stokes au niveau discret. Conserver cette caractéristique à l'échelle discrète permet de garantir naturellement la conservation de l'énergie cinétique. Ces schémas sont attrayants dans le contexte d'écoulements turbulents fortement instationnaires. Ducros et al. [Ducros et al., 1999] ont montré le caractère skew-symmetric du schéma JST avant d'en proposer une version d'ordre 4 et 6 [Ducros et al., 2000]. Pour la version d'ordre 4, le calcul du flux convectif en dimension 2 pour une face de normale x en $i + 1/2$ s'écrit par exemple :

$$\begin{aligned}
 uv|_{i+1/2} = & \frac{1}{3}(u_i + u_{i+1})(v_i + v_{i+1}) \\
 & - \frac{1}{24}(u_{i-1}v_{i-1} + u_{i-1}v_{i+1} + u_i v_i + u_i + v_{i+2} + u_{i+1}v_{i+1} + u_{i+2}v_i + u_{i+2}v_{i+2}) \\
 & + \frac{1}{3}\left(\frac{1}{2}(u_{i+1}v_{i+1} + u_i v_i) - \frac{1}{4}(u_{i+1}u_i)(v_{i+1} + v_i)\right)
 \end{aligned} \quad (\text{A.14})$$

Evaluation des schémas numériques de Jameson et al. et Ducros et al.

Les schémas de Jameson et al. et Ducros et al. ont été implémentés dans WCCH en début de thèse avant de s'orienter vers le développement d'un solveur incompressible. L'objectif était d'évaluer la pertinence de ces schémas dans un contexte de réduction de la diffusion numérique.

Evaluation de la diffusion numérique sur le cas test Taylor-Green

Pour évaluer la diffusion numérique, nous nous intéressons d'abord à l'écoulement des tourbillons de Taylor-Green (cf. Section 4.3.1). Nous comparons le schéma JST (avec $\kappa^{(2)} = 0.25$ et $\kappa^{(4)} = 0.02$) au schéma MUSCL (limiteur Van Leer avec solveur VFFC) et au schéma WENO5 (+ VFFC). Les calculs sont réalisés sur un maillage 128×128 pendant 3s de temps physique avec le schéma RK4 et une CFL fixée à 0.35. Les schémas JST et MUSCL sont tous les deux d'ordre 2 en espace mais nous constatons que le schéma JST est bien moins diffusif que MUSCL (Fig. A.1). Le schéma WENO d'ordre supérieur est encore nettement moins diffusif que le schéma JST.

Evaluation de la diffusion numérique sur le cas test de convection de tourbillon du CERFACS

Nous utilisons le cas test de la convection de tourbillon proposé par le CERFACS (cf. Section 4.1.1 pour la description du cas test) pour comparer maintenant les schémas JST, MUSCL (Van Leer), WENO et Ducros et al. . Nous regardons les profils de vitesse et de pression au bout de 10 passages du tourbillon dans la boîte périodique avec 128×128 cellules (Fig. A.2).

Nous constatons d'abord que l'amplitude du vortex est atténuée de façon similaire pour les schémas JST et MUSCL. Néanmoins le profil de vitesse présente des oscillations marquées dans le sillage du tourbillon pour le schéma JST, alors que le schéma MUSCL préserve relativement bien la localisation des extrema de vitesse. Il serait sans doute nécessaire d'augmenter les paramètres de diffusion numérique pour lisser la solution.

Par ailleurs le schéma de Ducros et al. donne d'excellents résultats, meilleurs que WENO notamment sur le pic de pression. Après 10 passages consécutifs, le vortex est très peu altéré. De plus il est apparu possible de ne pas recourir à la diffusion numérique pour stabiliser ce schéma. La précision du schéma et la convection dominante du cas test font que le schéma centré ne présente pas d'instabilités. Nous remarquerons

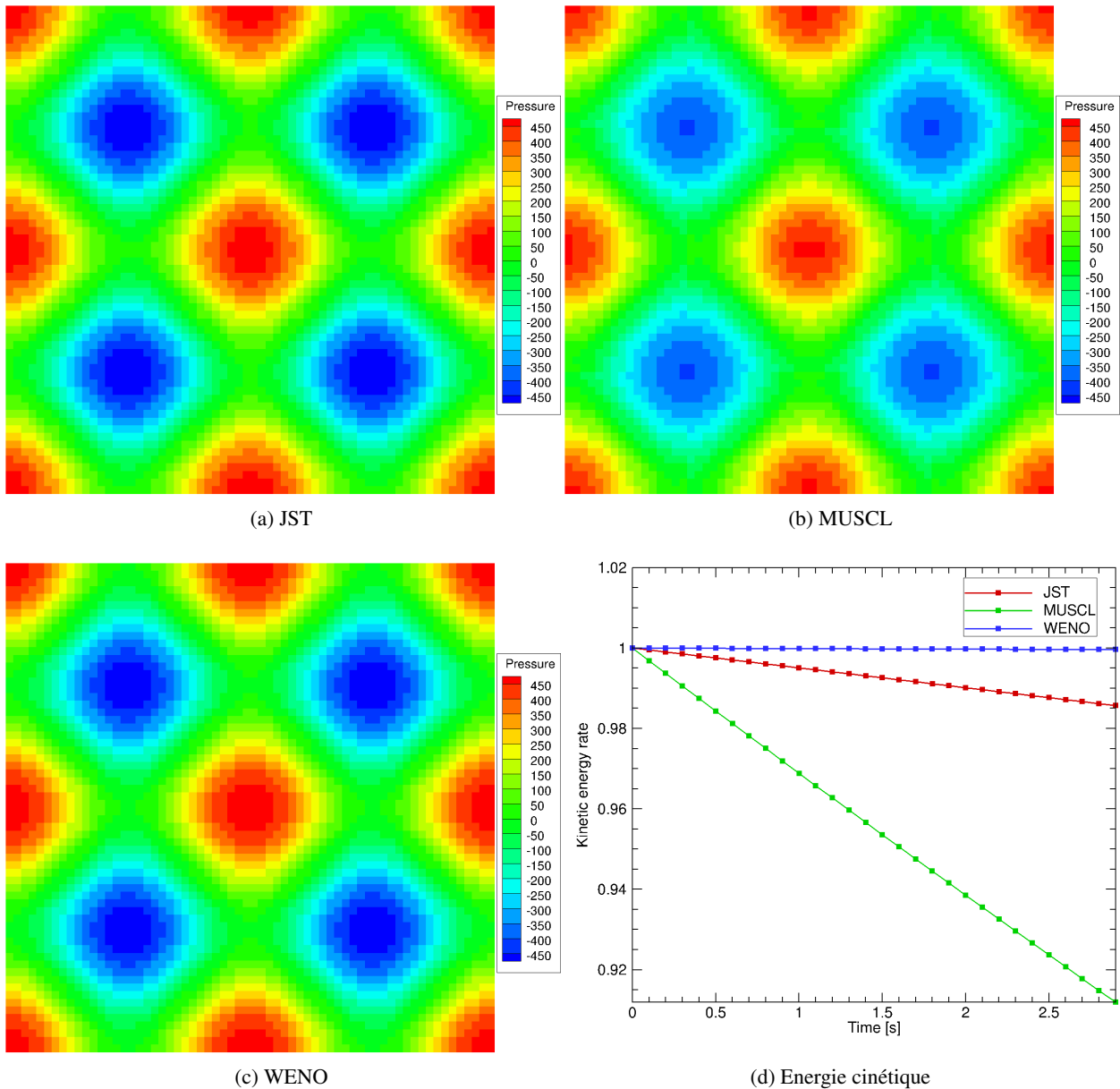


FIGURE A.1 – Comparaison des champs de pression à $t=3s$ et le taux de décroissance d'énergie cinétique pour un maillage 128×128

cependant de légères oscillations de vitesse en aval du vortex, signe que de très faibles instabilités persistent. Sans doute que l'introduction d'une légère diffusion permettrait de corriger ce problème.

Pour conclure, les schémas à viscosité artificielle peuvent être efficaces et sont finalement assez simples à mettre en œuvre. En revanche la paramétrisation des coefficients de diffusion est assez délicate. Avec une diffusion trop élevée, la convergence s'en trouve ralentie. A l'inverse si la diffusion n'est pas assez importante, le calcul peut rapidement diverger si les instabilités ne sont pas contrôlées. Les valeurs des paramètres sont à ajuster au cas par cas suivant la nature des écoulements rencontrés. Les schémas de décentrement basés sur les solveurs de Riemann offrent des alternatives plus robustes et qui permettent de bien contrôler la diffusion numérique. De plus, les schémas JST et Ducros et al. sont dédiés à la capture de choc et trouvent

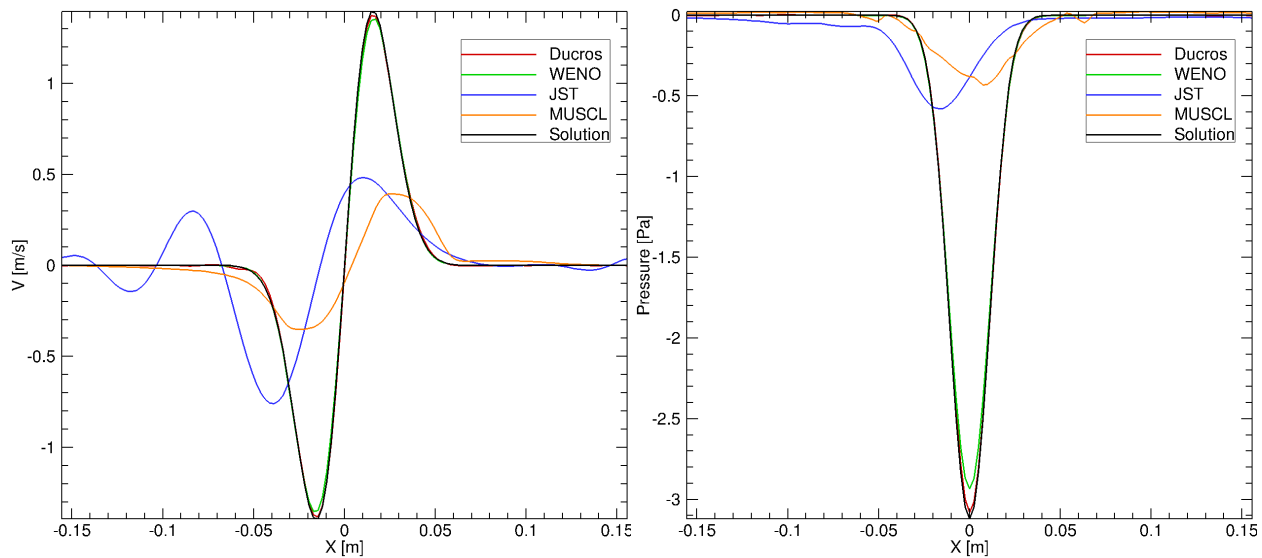


FIGURE A.2 – Profils de vitesse verticale et de pression au bout de 10 passages dans la boîte périodique de 128×128 cellules.

peu d'intérêt dans les applications hydrodynamiques quasi-incompressibles.

Finalement, nous nous sommes plutôt orientés vers l'implémentation d'une formulation incompressible pour la réduction des temps de calcul.

Annexe B

Implémentation de PETSc dans le code WCCH

PETSc est une bibliothèque dédiée au calcul scientifique parallèle en offrant un ensemble de structures et de routines pour la résolution d'équations aux dérivées partielles. Cette librairie open source est financée par plusieurs organismes d'état américain dont le Département de l'Énergie des États-Unis via le Argonne National Laboratory. Son développement initié dans le milieu des années 1990 [Balay et al., 1997] en fait aujourd'hui un outil abouti, robuste et répandu dans les codes CFD. PETSc est développée en C mais s'interface sans difficulté avec les langages C++, Fortran et Python. Elle est parallélisée suivant le protocole MPI.

En offrant tout un ensemble de structures de données (vecteurs, matrices, décomposition de domaine, etc.) et de routines récurrentes en informatique scientifique (résolution de systèmes linéaires, schémas d'avance en temps, etc.), PETSc facilite et accélère le développement d'un solveur. C'est une véritable boîte à outils permettant de tester rapidement différents schémas. Sa popularité s'explique sans doute aussi par le fait qu'elle s'adresse à un public large. En effet, une fois installée, elle est d'usage assez simple. Mais, pour l'utilisateur avancé, il reste possible de paramétrer soigneusement chaque fonctionnalité de PETSc pour contrôler en profondeur le processus de calcul et maintenir ainsi les performances du code. Par exemple la stratégie de recouvrement des communications par le calcul est tout à fait possible.

Par ailleurs PETSc s'interface avec de nombreuses autres bibliothèques d'algèbre linéaire comme HYPRE, MUMPS, etc.

Enfin elle dispose d'une communauté active de développeurs et d'utilisateurs. Une documentation détaillée est fournie avec de nombreux exemples d'illustration [Balay et al., 2017].

Dans la suite nous donnons quelques exemples d'utilisation de structures et de routines PETSc en Fortran, telles qu'elles peuvent être utilisées dans WCCH. Ces exemples ne sont pas représentatifs de la variété des fonctionnalités disponibles dans PETSc mais donnent un aperçu de l'intégration de la librairie dans WCCH.

Initialisation de l'environnement PETSc

En début de programme, il est nécessaire d'initialiser l'environnement d'exécution de PETSc. La routine `PetscInitialize` fait appel à la routine `MPI_Init` de la librairie MPI. Sans précision particulière, le communicateur par défaut est `MPI_COMM_WORLD`. En fin de programme, `PetscFinalize` clôt cet environnement.

```
#include "petsc/finclude/petscdef.h"
call PetscInitialize(PETSC_NULL_CHARACTER, ierr)
...
```

```
call PetscFinalize(ierr)
```

Vecteurs

Les vecteurs de PETSc sont définis à partir de l'objet `Vec`. La routine `VecCreateMPI` permet d'allouer de la mémoire pour la création d'un vecteur parallèle. Dans PETSc, la dimension d'un objet parallèle peut être spécifiée de deux façons distinctes : en précisant sa taille locale pour le processeur qui appelle la routine ou sa taille globale commune à tous les processeurs. Dans le premier cas, PETSc en déduit facilement la taille globale. Dans le deuxième, la taille locale pour chaque processeur est décidée par PETSc. Dans le cas de WCCH, la partition du maillage est déjà effectuée par le code. La première technique est donc employée.

```
call VecCreateMPI(MPI_COMM_WORLD, nLocalRows, PETSC_DETERMINE, x, ierr)
```

`nLocalRows` indique le nombre d'éléments du vecteur stockés par chaque processeur.

Le mot clé `PETSC_DETERMINE` indique que le calcul de la taille globale du vecteur est laissé à PETSc. Un vecteur peut également être défini par duplication d'un vecteur préalablement créé :

```
call VecDuplicate(x, y, ierr)
```

Le remplissage d'un vecteur peut se faire au moyen de la fonction suivante.

```
call VecSetValue(x, indice, value, INSERT_VALUES, ierr)
```

`VecSetValue` affecte à la ligne `indice` du vecteur `x`, la valeur `value`. `INSERT_VALUES` permet d'insérer la valeur. Pour l'ajouter, on utilise le mot-clé `ADD_VALUES`. Pour remplir tout un vecteur de façon efficace, on préfère la fonction `VecSetValues` :

```
call VecSetValues(x, nValues, indices(:), values(:), INSERT_VALUES, ierr)
```

`VecSetValues` affecte un nombre `nValues` de valeurs stockées dans le tableau `values` aux indices `indices` de `x`.

Les valeurs remplies avec `VecSetValue` ou `VecSetValues` sont stockées dans un buffer. Il est nécessaire d'appeler les deux routines suivantes avant de pouvoir utiliser le vecteur.

```
call VecAssemblyBegin(x, ierr)
```

```
call VecAssemblyEnd(x, ierr)
```

`VecGetArrayReadF90` permet ensuite de lire un vecteur en vue de récupérer ses valeurs. Un pointeur permet d'y accéder :

```
double precision, pointer :: xPointer(:)
call VecGetArrayReadF90(x, xPointer, ierr)
call VecRestoreArrayReadF90(x, xPointer, ierr)
```

En fin de programme, lorsque le vecteur n'est plus utilisé, il est détruit avec la commande :

```
call VecDestroy(x, ierr)
```

Matrices

La représentation par défaut des matrices dans PETSc est la matrice creuse stockée au format CSR (Compressed Sparse Row). Ce format répandu permet de ne stocker que les éléments non nuls. La commande suivante permet de créer une matrice `A` creuse et parallèle :

```
call MatCreateAIJ( MPI_COMM_WORLD, nLocalRows, nLocalColumns, &
                  PETSC_DETERMINE, PETSC_DETERMINE, &
                  PETSC_NULL_INTEGER, d_nnz( : ), PETSC_NULL_INTEGER, o_nnz( : ), &
                  A, ierr )
```

$nLocalRows$ est le nombre de lignes de la sous-matrice locale. Il doit être le même que le nombre de lignes d'un vecteur y stockées par un processeur pour le calcul de $y = Ax$. $nLocalColumns$ est le nombre de colonnes de la sous-matrice locale. Il doit être le même que le nombre de lignes d'un vecteur x stockées par un processeur pour le calcul de $y = Ax$. Ainsi chaque processeur héberge $nLocalRows$ de la matrice (Fig. B.1). `PETSC_DETERMINE` laisse à PETSc le soin de calculer la taille globale de la matrice. $d_nnz(:)$ est un tableau de longueur $nLocalRows$ qui contient le nombre d'éléments non nuls de la sous-matrice locale. $o_nnz(:)$ est le nombre d'éléments non nuls pour la matrice extra-diagonale. $d_nnz(:)$ et $o_nnz(:)$ permettent de préallouer la mémoire nécessaire au stockage de la matrice. Une autre façon de procéder est de donner un nombre identique d'éléments non nuls pour toutes les lignes de la sous-matrice locale et la matrice extra-diagonale. Néanmoins le risque est de voir les performances de calcul s'effondrer. La technique de préallocation fine est donc employée dans WCCH.

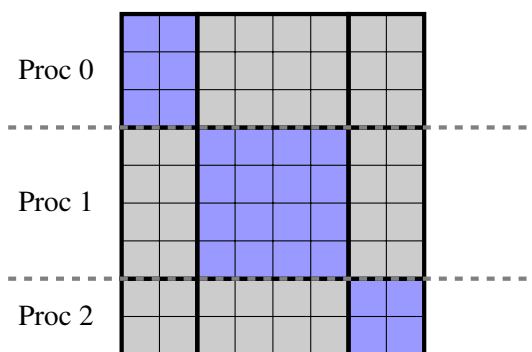


FIGURE B.1 – Définition d'une matrice parallèle en sous-matrices locales ■ et sous-matrices extra-diagonales ■

`MatSetValue` permet d'affecter une valeur à un élément ij d'une matrice A :

```
call MatSetValue(A, i, j, value, ADD_VALUES, ierr)
```

Il existe une version équivalente pour affecter un tableau de valeurs :

```
call MatSetValues(A, nLocalRows, indicesRow( : ), &
                 nLocalColumns, indicesColumn( : ), &
                 values( : , : ), ADD_VALUES, ierr)
```

`MatSetValue` et `MatSetValues` ne font que stocker les valeurs en cache. La matrice n'est utilisable qu'après avoir appelé :

```
call MatAssemblyBegin(A, MAT_FINAL_ASSEMBLY, ierr)
call MatAssemblyEnd(A, MAT_FINAL_ASSEMBLY, ierr)
```

Il est toujours possible de mettre des instructions entre `MatAssemblyBegin` et `MatAssemblyEnd` pour faire du recouvrement calcul communication.

En fin de programme, une matrice est détruite avec la commande :

```
call MatDestroy(A, ierr)
```

Solveur et préconditionneur

Pour résoudre un système linéaire avec une méthode des sous-espaces de Krylov, il faut créer un objet PESTc appelé KSP :

```
KSP :: ksp_poisson  
call KSPCreate(MPI_COMM_WORLD, ksp_poisson, ierr)
```

KSPSetOperators permet de lier une matrice de système linéaire à un objet KSP :

```
call KSPSetOperators(ksp_poisson, LHS_poisson, LHS_poisson, ierr)
```

ksp_poisson est l'objet KSP défini au préalable. L'argument qui suit dans la commande est la matrice du système linéaire (ici *LHS_poisson*). Vient ensuite la matrice associée au préconditionneur qui est en général la même (*LHS_poisson*).

Le solveur utilisé par défaut est un GMRES relancé (restart GMRES) mais il est possible de spécifier un solveur comme par exemple :

```
call KSPSetType(ksp_poisson, KSPCG, ierr)
```

pour recourir à un gradient conjugué. De nombreux autres solveurs sont disponibles et détaillés dans la documentation.

Le préconditionneur utilisé par défaut est une méthode de type block Jacobi (un block pour chaque processeur et une factorisation ILU pour chaque block). Pour préciser un préconditionneur particulier de la librairie HYPRE par exemple :

```
call KSPGetPC(ksp_poisson, pc_poisson, ierr)  
call PCSetType(pc_poisson, PCHYPRE, ierr)
```

Enfin il est possible de laisser libre le choix d'un certain nombre de paramètres à l'exécution avec la commande

```
call KSPSetFromOptions(ksp_poisson, ierr)
```

Par exemple l'option *-ksp_max_it* permet de spécifier le nombre maximum d'itérations, *-ksp_rtol* le seuil de tolérance du résidu de convergence, etc.

Pour résoudre le système linéaire, on appelle :

```
call KSPSolve(ksp_poisson, RHS_poisson, pressure, ierr)
```

RHS_poisson désigne le second membre du système linéaire et *pressure* est le vecteur contenant le résultat.

De nombreuses options permettent de monitorer la résolution du système linéaire et d'afficher les résultats de chaque itération.

Enfin l'objet KSP créé est détruit par :

```
call KSPDestroy(ksp_poisson, ierr)
```

Remarques

L'intégration de PETSc dans WCCH est assez légère. mais elle offre un grand nombre de structures de données et de fonctionnalités. La hiérarchie du maillage peut être entièrement gérée par PETSc (avec le DM, data management pour les maillages structurés, ou DMplex pour les maillages déstructurés). La parallélisation est ensuite complètement prise en charge par PETSc. Par ailleurs, la librairie permet l'intégration facile de schémas d'avance en temps, de résolution de systèmes non-linéaires, etc.

Annexe C

Impact d'une accélération ou de l'optimisation des algorithmes de WCCH

La Section 4.3 du Chapitre 4 est consacrée à la comparaison entre les formulations incompressible et faiblement-compressible. Il a été mis en évidence que l'approche incompressible offre un gain certain sur la restitution des résultats à précision équivalente par rapport à l'approche faiblement-compressible (gain compris entre 5 et 20 pour l'écoulement autour du profil NACA à $Re = 1400$). Cependant, nous sommes en droit de questionner la validité de cette conclusion étant donné qu'elle est liée à une implémentation donnée. Notamment qu'en est-il si l'algorithme faiblement-compressible était optimisé et accéléré ?

Nous étudions d'abord l'impact de l'accélération de l'algorithme faiblement-compressible sur le gain CPU incompressible/faiblement-compressible pour des maillages identiques. Le temps CPU total incompressible t_{CPU}^{inc} peut s'exprimer comme le produit du coût CPU moyen d'un pas de temps $t_{CPU/\Delta t}^{inc}$ et du nombre total de pas de temps $n_{\Delta t}^{inc}$:

$$t_{CPU}^{inc} = t_{CPU/\Delta t}^{inc} n_{\Delta t}^{inc} \quad (C.1)$$

De même en faiblement-compressible :

$$t_{CPU}^{comp} = t_{CPU/\Delta t}^{comp} n_{\Delta t}^{comp} \quad (C.2)$$

Par ailleurs le nombre total de pas de temps $n_{\Delta t}^{inc}$ peut être calculé à partir du rapport du temps physique simulé $t_{physique}$ et de la taille du pas de temps moyen du schéma Δt :

$$n_{\Delta t} = \frac{t_{physique}}{\Delta t} \quad (C.3)$$

Le gain CPU incompressible/faiblement-compressible vaut alors :

$$\frac{t_{CPU}^{comp}}{t_{CPU}^{inc}} = \frac{t_{CPU/\Delta t}^{comp}}{t_{CPU/\Delta t}^{inc}} \frac{\Delta t^{comp}}{\Delta t^{inc}} \quad (C.4)$$

Si nous considérons que le nombre CFL est identique pour les deux formulations :

$$\frac{\Delta t^{comp}}{\Delta t^{inc}} = 1 + \frac{1}{Ma} \quad (C.5)$$

Dans une approche simplificatrice nous supposons ensuite que le coût d'un pas de temps incompressible se décompose en un temps de résolution explicite directement proportionnel au temps CPU faiblement-compressible et un temps de résolution de système linéaire :

$$t_{CPU/\Delta t}^{inc} = \alpha t_{CPU/\Delta t}^{comp} + t_{CPU/\Delta t}^{Poisson} \quad (C.6)$$

Bien que sommaire, cette décomposition a du sens puisque la première étape du schéma de projection de Chorin correspond quasiment à une étape d'avance en temps du solveur faiblement-compressible. Des différences subsistent, notamment pour la résolution du problème de Riemann, pour le calcul de la vitesse d'advection ou le traitement des interfaces de raffinement. L'expérience numérique nous a montré que $\alpha \approx 1.0$ mais pour plus de sécurité nous prendrons $0.8 \leq \alpha \leq 1.2$ par la suite. Enfin le temps de résolution du système linéaire représente une fraction p du coût CPU d'un pas de temps :

$$t_{CPU/\Delta t}^{Poisson} = p t_{CPU/\Delta t}^{inc} \quad (C.7)$$

Si bien que :

$$t_{CPU/\Delta t}^{Poisson} = \alpha \frac{p}{1-p} t_{CPU/\Delta t}^{comp} \quad (C.8)$$

L'expérience numérique, encore une fois, nous a montré que $0.3 \leq p \leq 0.5$. Finalement le gain vaut :

$$\frac{t_{CPU}^{comp}}{t_{CPU}^{inc}} = \frac{t_{CPU/\Delta t}^{comp}}{\alpha t_{CPU/\Delta t}^{comp} + \alpha \frac{p}{1-p} t_{CPU/\Delta t}^{comp}} \left(1 + \frac{1}{Ma}\right) \quad (C.9)$$

Définissons ensuite γ le facteur d'accélération potentielle de l'algorithme faiblement-compressible. Le nouveau coût CPU d'un pas de temps est alors :

$$\tilde{t}_{CPU/\Delta t}^{comp} = \frac{1}{\gamma} t_{CPU/\Delta t}^{comp} \quad (C.10)$$

où $\gamma > 1$ si le temps CPU est diminué. Le nouveau gain CPU est alors :

$$\frac{t_{CPU}^{comp}}{t_{CPU}^{inc}} = \frac{\tilde{t}_{CPU/\Delta t}^{comp}}{\alpha \tilde{t}_{CPU/\Delta t}^{comp} + \alpha \frac{p}{1-p} t_{CPU/\Delta t}^{comp}} \left(1 + \frac{1}{Ma}\right) \quad (C.11)$$

En effet l'accélération de l'algorithme faiblement-compressible profite aussi à l'incompressible et à la première étape du schéma de projection. En revanche la résolution de l'équation de Poisson effectuée par PETSc n'en bénéficie pas et son coût CPU reste inchangé. Après simplification il vient :

$$\frac{t_{CPU}^{comp}}{t_{CPU}^{inc}} = \frac{1}{\alpha(1 + \gamma \frac{p}{1-p})} \left(1 + \frac{1}{Ma}\right) \quad (C.12)$$

Nous traçons alors sur la figure C.1 le gain CPU en fonction du facteur d'accélération γ . Nous fixons le nombre de Mach à 0.1. Deux valeurs de p sont choisies, $p = 0.3$ et $p = 0.5$. Enfin α varie entre 0.8 et 1.2 ce qui explique l'allure épaisse de la courbe.

D'après ce graphique nous pouvons conclure que les algorithmes faiblement-compressible doivent être accéléré d'un facteur compris entre 4 et 30 pour donner des temps de restitution identique à la formulation incompressible sur les mêmes maillages.

Raisonnons maintenant à précision équivalente. Si l'on note G le gain CPU initial de l'incompressible sur le faiblement-compressible à précision équivalente :

$$G = \frac{t_{CPU/\Delta t}^{comp} \Delta t^{comp}}{t_{CPU/\Delta t}^{inc} \Delta t^{inc}} \quad (C.13)$$

Ou encore :

$$G = \frac{t_{CPU/\Delta t}^{comp}}{\alpha t_{CPU/\Delta t}^{comp} + \alpha \frac{p}{1-p} t_{CPU/\Delta t}^{comp}} \frac{\Delta t^{comp}}{\Delta t^{inc}} \quad (C.14)$$

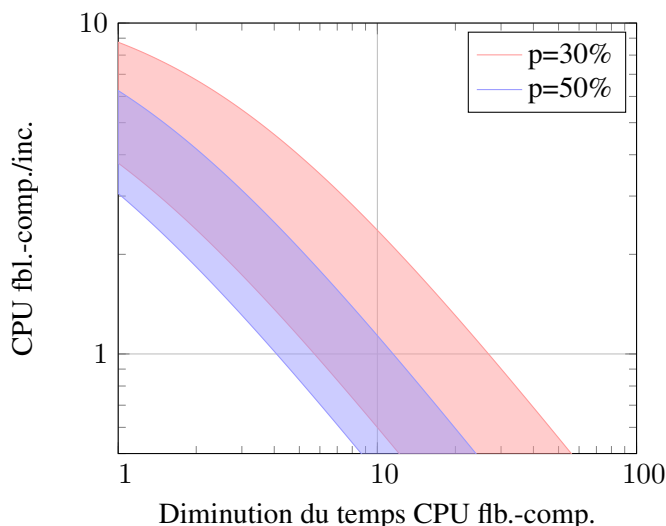


FIGURE C.1 – Gain CPU incompressible/faiblement-compressible en fonction de l'accélération de l'algorithme faiblement-compressible à maillage constant.

Soit :

$$G = \frac{\alpha}{1-p} \frac{\Delta t^{comp}}{\Delta t^{inc}} \quad (\text{C.15})$$

Nous pouvons exprimer le rapport $\frac{\Delta t^{comp}}{\Delta t^{inc}}$ en fonction de G

$$\frac{\Delta t^{comp}}{\Delta t^{inc}} = G \frac{1-p}{\alpha} \quad (\text{C.16})$$

D'après le calcul précédent, le nouveau gain suite à l'optimisation du code faiblement-compressible vaut :

$$\frac{t_{CPU}^{comp}}{t_{CPU}^{inc}} = \frac{1}{\alpha(1 + \gamma \frac{p}{1-p})} \frac{\Delta t^{comp}}{\Delta t^{inc}} \quad (\text{C.17})$$

$\frac{\Delta t^{comp}}{\Delta t^{inc}}$ reste inchangé puisqu'il est lié à la physique de l'écoulement. Donc finalement :

$$\frac{t_{CPU}^{comp}}{t_{CPU}^{inc}} = \frac{G}{1 + p(\gamma - 1)} \quad (\text{C.18})$$

Ce gain est indépendant de α . Pour $\gamma = 1$, on retrouve le gain initial G avant optimisation.

La figure C.2 présente les nouveaux gains en fonction de l'accélération γ pour $p = 0.3$ et $p = 0.5$. Le gain initial est fixé à $5 < G < 20$ ce qui correspond aux valeurs relevées sur le cas test de l'écoulement autour du profil NACA.

Nous remarquons que si le gain initial est compris entre 5 et 20, le solveur faiblement-compressible doit être accéléré d'un facteur compris entre 9 et 70 pour devenir compétitif.

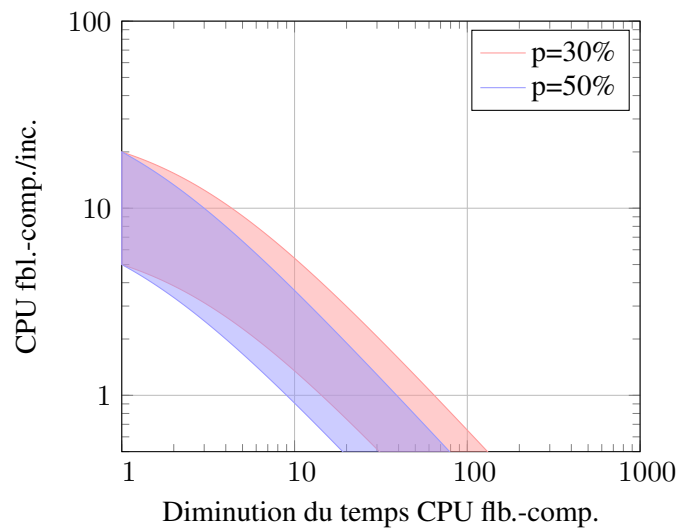


FIGURE C.2 – Gain CPU incompressible/faiblement-compressible en fonction de l'accélération potentielle de l'algorithme faiblement-compressible à précision constante.

Annexe D

Implémentation d'un schéma PIMPLE incompressible dans WCCH

Schéma PIMPLE

Les schémas de type prédiction-correction comme SIMPLE et autres (SIMPLEC, PISO, PIMPLE, etc.) sont largement utilisés dans les codes incompressibles d'hydrodynamique navale. Initialement ces schémas ont été développés pour simuler des écoulements stationnaires. Des études récentes tendent à montrer que sur des écoulements fortement instationnaires, les schémas de projection de type Chorin sont plus avantageux d'un point de vue coût CPU [Hines, 2008].

Afin d'appréhender quels pourraient être les avantages et inconvénients des approches de type SIMPLE, il est apparu intéressant d'implémenter un algorithme PIMPLE dans le code WCCH. Le PIMPLE connaît un engouement particulier auprès des utilisateurs du code OpenFOAM. Il semble en mesure de traiter efficacement les écoulements instationnaires tout en autorisant des nombres CFL bien supérieurs à 1. L'idée de base du PIMPLE est de faire converger la solution vers un état stationnaire à chaque pas de temps.

Rappelons d'abord la forme matricielle des équations de Navier-Stokes discrétisées :

$$\begin{pmatrix} E - Q & G \\ D & O \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \quad (\text{D.1})$$

$E - Q$ désigne la matrice résultant de la discrétisation des équations de quantité de mouvement. E est sa portion diagonale alors que Q est sa partie extra-diagonale. f est le terme source. G est l'opérateur gradient et D l'opérateur divergence.

La figure D.1 détaille l'algorithme PIMPLE pour un pas de temps.

Pour l'intégration en temps, notre choix s'est porté sur un schéma Crank-Nicolson d'ordre 2 :

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \beta C(\mathbf{u}^{n+1}) + (1 - \beta)C(\mathbf{u}^n) = \beta D(\mathbf{u}^{n+1}) + (1 - \beta)D(\mathbf{u}^n) \quad (\text{D.2})$$

où C fait référence à l'opérateur convectif et D l'opérateur diffusif. β est un paramètre entre 0 et 1. Pour $\beta = 0$, on obtient un schéma purement explicite d'Euler d'ordre 1, $\beta = 1$ correspond à la version implicite et $\beta = 0.5$ donne le schéma de Crank-Nicolson.

Pour la discrétisation en espace de l'opérateur convectif, nous utilisons un schéma de type *deferred correction* mêlant un calcul de flux à l'ordre 2 par un schéma centré et un calcul à l'aide du schéma WENO 5. La partie d'ordre 2 est résolue implicitement alors que la différence du flux WENO5 et du flux d'ordre 2 est déterminée explicitement. A convergence, les contributions d'ordre 2 s'annulent pour ne laisser que la

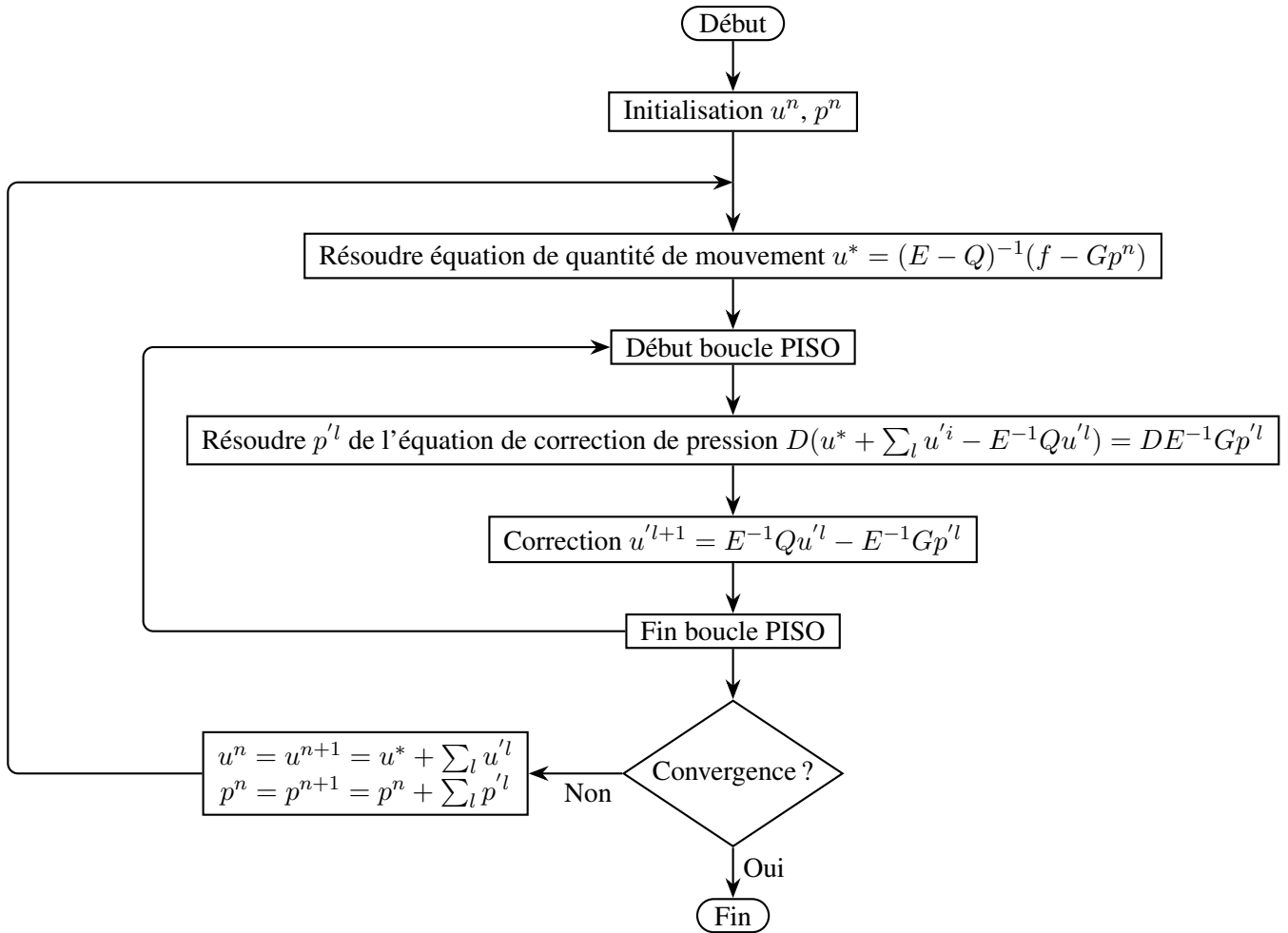


FIGURE D.1 – Algorithme PIMPLE d'un pas de temps

contribution d'ordre élevé :

$$\frac{\mathbf{u}^m - \mathbf{u}^n}{\Delta t} + \beta \left(C_L(\mathbf{u}^m) + C_H(\mathbf{u}^{m-1}) - C_L(\mathbf{u}^{m-1}) \right) + (1-\beta) \left(C(\mathbf{u}^n) + C_H(\mathbf{u}^n) - C_L(\mathbf{u}^n) \right) = \dots \quad (\text{D.3})$$

C_H correspond au calcul du flux convectif avec le WENO5 alors que C_L est calculé avec un schéma d'ordre 2. m correspond à l'itération non-linéaire.

L'implémentation du PIMPLE dans le code WCCH n'a pas représenté une tâche de développement importante dans la mesure où seule la résolution implicite des équations de quantité de mouvement restait à développer par rapport à la version incompressible avec projection de Chorin. Les systèmes linéaires associés à chaque composante de vitesse des équations de quantité de mouvement sont résolus avec un algorithme GMRES redémarré sans préconditionnement.

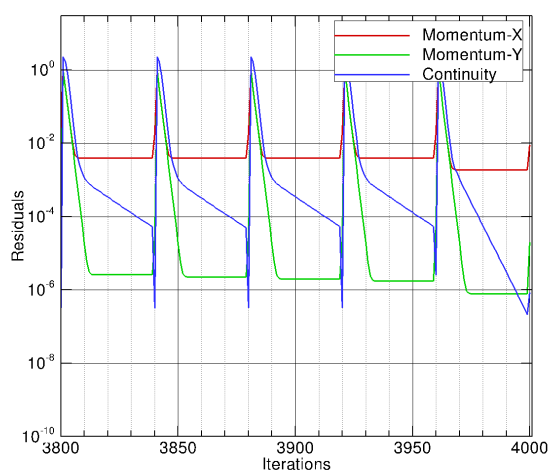
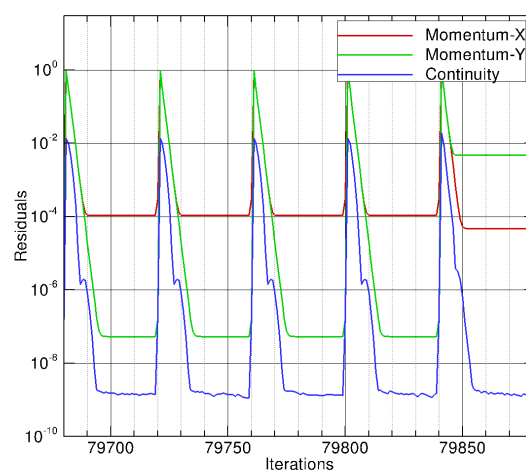
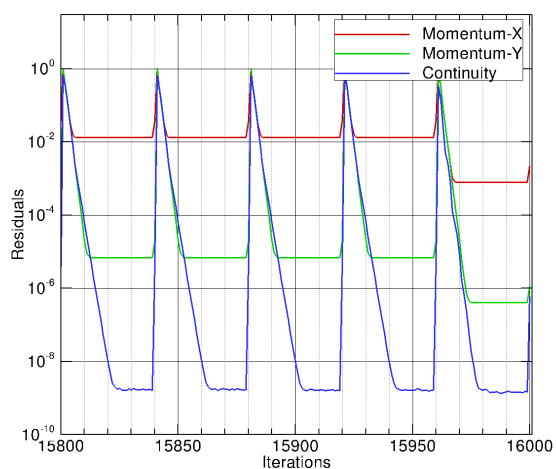
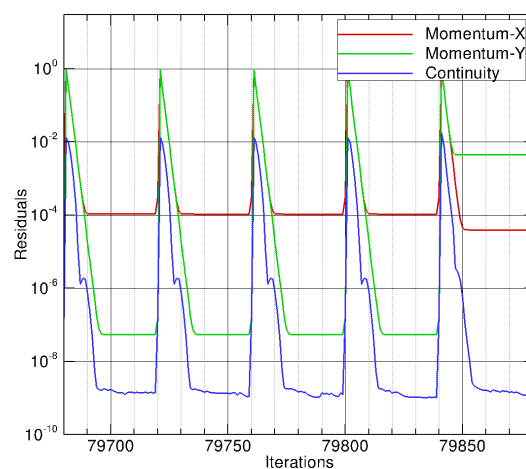
Validations numériques

Convection de tourbillon

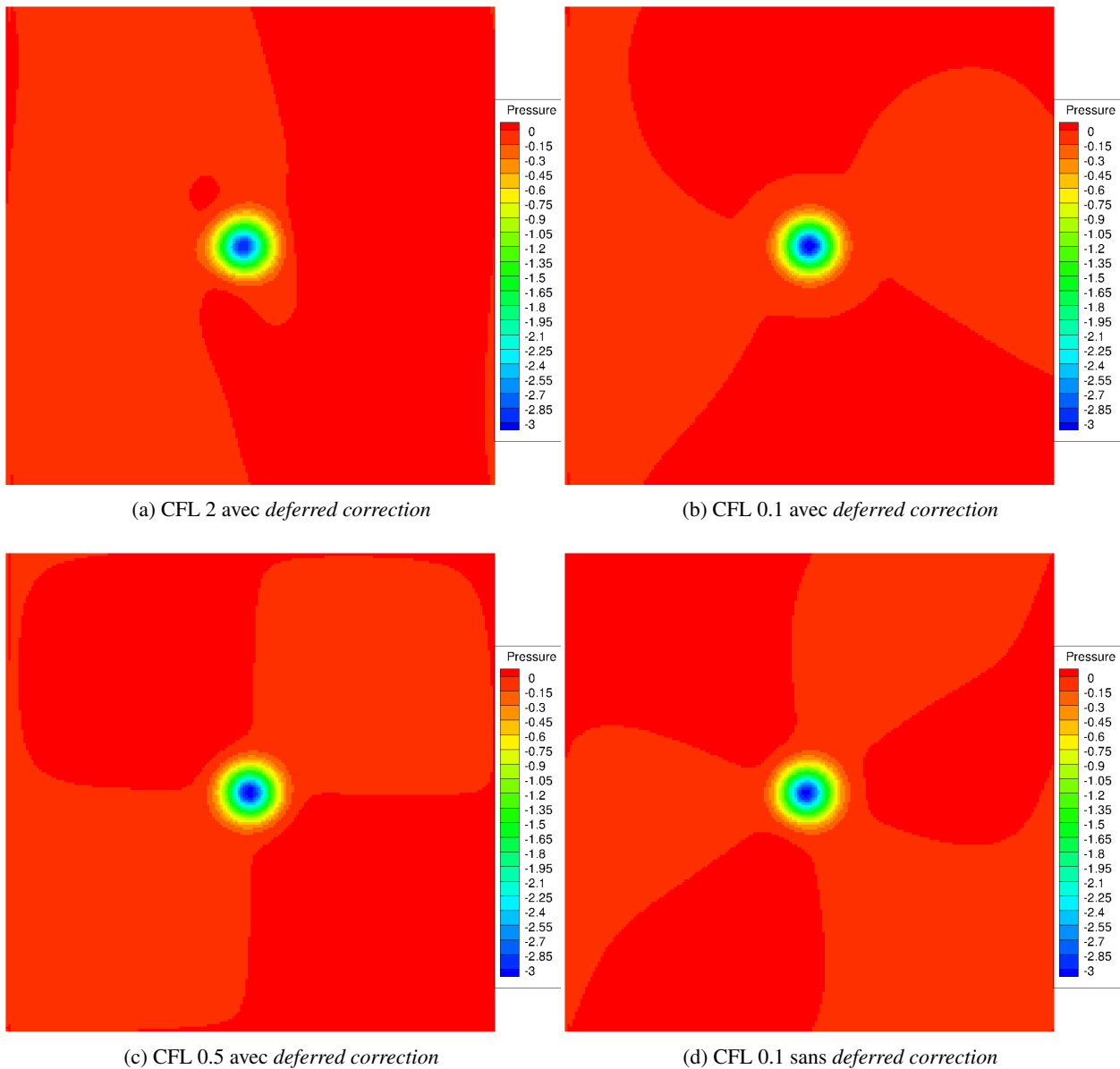
Dans un premier temps, nous testons le schéma PIMPLE sur le cas test de convection de tourbillon du CERFACS.

Nous effectuons une convergence en maillage pour des grilles allant de 32×32 à 512×512 cellules. L'écoulement est simulé sur un passage de tourbillon. Le nombre d'itérations non-linéaires est fixé à 40 (pas d'optimisation sur la tolérance des résidus par exemple) et celui des itérations de correction de pression (boucle PISO) à 2. Les équations de pression et de quantité de mouvement sont relaxées par un facteur 0.7. Nous testons différentes valeurs du nombre CFL avec et sans *deferred correction*.

La figure D.2 présente la décroissance des résidus pour quelques pas de temps sur un maillage 192×192 . Les résidus sont calculés en norme L_2 sous forme adimensionnelle. Entre chaque pas de temps, leur décroissance est suffisante pour considérer que la solution est convergée à la fin des itérations non-linéaires.

(a) CFL 2 avec *deferred correction*(b) CFL 0.1 avec *deferred correction*(c) CFL 0.5 avec *deferred correction*(d) CFL 0.1 sans *deferred correction*FIGURE D.2 – Convergence des résidus pour un maillage 192×192

La figure D.3 présente le champ de pression après un passage du tourbillon dans le domaine périodique pour le maillage 192×192 . Aucune différence notable n'est à signaler entre les différents calculs. Même pour celui sans *deferred correction* dont l'ordre de convergence théorique n'est que de 2.

FIGURE D.3 – Champ de pression après un passage du tourbillon pour un maillage 192×192

Enfin sur la figure D.4 nous traçons les courbes de convergence en vitesse et en espace. L'erreur est calculée par rapport à une solution analytique en norme L_2 comme détaillée Section 4.2.1. Elle n'est pas intégrée en temps. Pour les calculs avec l'algorithme PIMPLE, le nombre CFL influence grandement les résultats. Pour $CFL = 2$ la convergence est globalement d'ordre 2. Pour $CFL = 0.1$, l'ordre s'approche de sa valeur théorique de $4 \sim 5$. Cet exemple illustre que la convergence en espace et la convergence en temps sont intimement liées sur ce cas test. L'ordre du schéma d'avance en temps n'est sans doute pas assez élevé pour simuler correctement cet écoulement dont la vitesse de convection est, rappelons-le, de $35m/s$ alors que l'amplitude de la vitesse du vortex n'est que de $1.4m/s$. Si on compare le résultat du PIMPLE avec et sans *deferred correction* pour $CFL = 0.1$, on remarque que l'utilisation de ce type de technique à un intérêt certain. Par ailleurs le calcul avec *deferred correction* à $CFL = 0.1$ donne des résultats bien meilleurs que les formulations faiblement-compressible et incompressible de Chorin. Le terme

"Chorin_RK4_cfl0.6++" en légende désigne un schéma de projection de Chorin avec intégration en temps par une méthode de Runge-Kutta d'ordre 4 où la pression finale n'est pas la pression de la dernière sous-étape mais une pression reconstruite à partir des pressions des 4 sous-étapes (cf. Section 2.3.3).

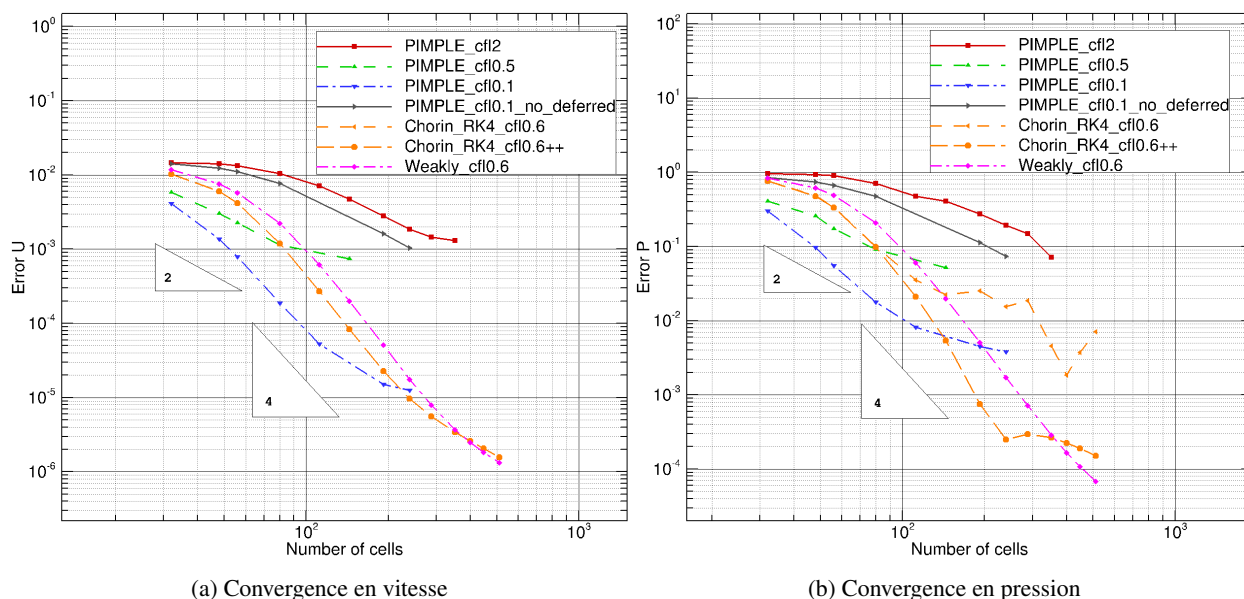


FIGURE D.4 – Comparaison des convergences en vitesse et en pression de différents schémas, PIMPLE, projection de Chorin, faiblement-compressible, etc.

En définitive, bien que montrant de bons de résultats de convergence, l'algorithme PIMPLE a un coût CPU très important. Sur un écoulement très instationnaire, il faut abaisser la CFL à des valeurs bien inférieures à 1 pour faire converger les résultats. Dans ce cas, le PIMPLE perd de son intérêt par rapport à des méthodes purement explicites comme en faiblement-compressible ou semi-implicites comme la projection de Chorin.

Cavité entraînée

Dans un deuxième temps, nous évaluons le schéma PIMPLE sur le cas de test de la cavité entraînée à $Re = 1000$. Pour ce nombre de Reynolds, l'écoulement est stationnaire. Les paramètres de calcul du schéma PIMPLE sont :

$$\left\{ \begin{array}{l} \text{Taille du maillage : } 64 \times 64 \\ \text{Nombre total d'itérations non-linéaires : } 5000 \\ \text{Coefficient de relaxation équation de quantité de mouvement : } 0.2 \text{ (avec } deferred \text{ correction) } 0.7 \text{ sinon} \\ \text{Coefficient de relaxation équation de quantité de pression : } 0.2 \text{ (avec } deferred \text{ correction) } 0.3 \text{ sinon} \end{array} \right. \quad (D.4)$$

La figure D.5 présente la décroissance des résidus. Plus que la valeur des résidus atteinte, c'est leur stabilisation qui est importante. On remarque que les résidus convergent plus rapidement sans *deferred correction*. Ceci n'est pas surprenant compte-tenu des valeurs coefficients de relaxation choisies par rapport au cas avec *deferred correction*. Le cas avec *deferred correction* aurait mérité d'être prolongé de quelques centaines d'itérations non-linéaires pour qu'il converge. Néanmoins, les variations sur les champs de pression après 5000 itérations n'étaient pas perceptibles visuellement ce qui a conduit à stopper le calcul à cet instant.

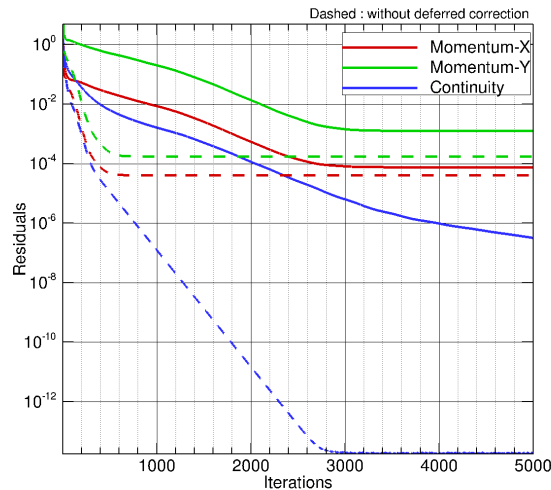


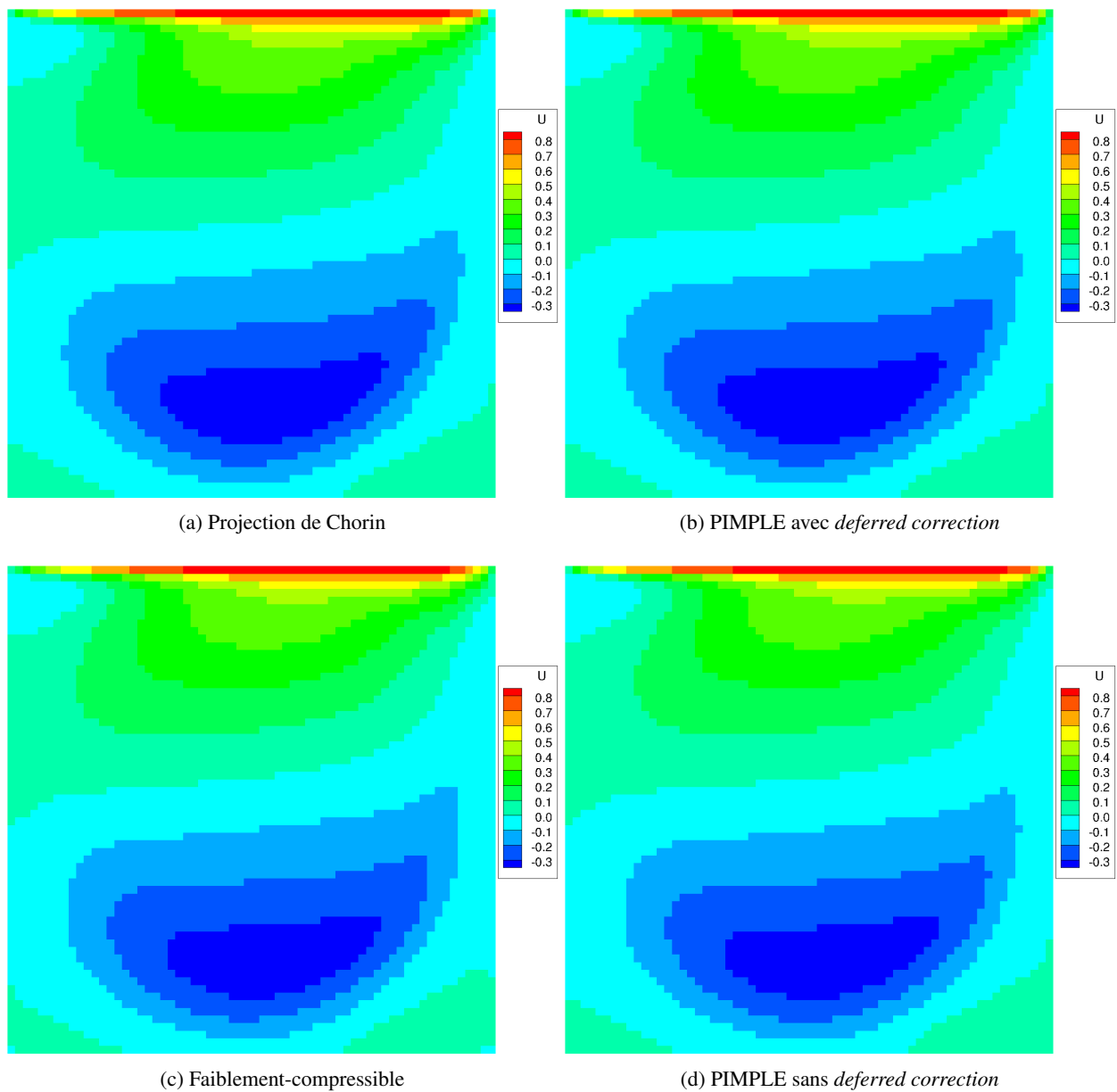
FIGURE D.5 – Convergence des résidus du schéma PIMPLE (tirets : sans *deferred correction*, trait plein : avec *deferred correction*)

Comme pour le cas test précédent, la simulation est aussi effectuée avec les schémas de projection de Chorin et de faiblement-compressible. Pour ces calculs, le temps physique de simulation est fixé à 50s et $cfl = 0.7$. Cela représente environ 5000 itérations temporelles de schéma RK4 pour l'incompressible et 50000 pour le faiblement-compressible. Les champs de vitesse sont affichés pour les différents schémas sur la figure D.6. Le résultat du schéma de projection de Chorin et celui du PIMPLE avec *deferred correction* sont quasiment identiques comme cela est attendu. En effet en régime stationnaire, les différents termes des équations de Navier-Stokes sont discrétisés par les mêmes opérateurs avec le schéma PIMPLE et avec le schéma de projection.

Ensuite les champs de pression sont tracés sur la figure D.7. Pour rendre les résultats comparables, le niveau de pression est harmonisé entre les différents calculs en fixant la pression d'une cellule. Globalement les champs sont similaires.

Pour départager plus finement les différents schémas, nous traçons les profils verticaux de vitesse horizontale au milieu de la cavité (Fig. D.8). Ce résultat confirme ce qui a déjà été énoncé sur le schéma de projection et le PIMPLE avec *deferred correction*. Il met aussi en évidence que l'utilisation de la méthode *deferred correction* améliore sensiblement les résultats. Le résultat faiblement-compressible paraît plus éloigné de la solution de Ghia et al. [Ghia et al., 1982]

En conclusion, un schéma PIMPLE a été implémenté avec succès dans le code WCCH. Plus que de disposer d'un schéma incompressible supplémentaire, l'objectif était avant tout de se rendre compte de la complexité algorithmique d'un tel schéma et de sa compatibilité avec les caractéristiques attendues du code WCCH (schémas d'ordre élevé, simplicité algorithmique, HPC, etc.). L'implémentation et l'utilisation n'ont sans doute pas été optimales, loin s'en faut. Sur un cas test stationnaire, comme celui de la cavité entraînée, il semblerait que le PIMPLE (avec *deferred correction*) donne des temps de calcul comparables au schéma de projection de Chorin utilisé avec un RK4 pour un maillage identique. Le PIMPLE sans *deferred correction* est en revanche plus rapide (difficile d'estimer cependant un gain car les deux approches restent sujettes à un seuil de convergence : temporelle pour l'une et liée au résidu pour l'autre). Pour un écoulement très instationnaire comme celui du cas test du CERFACS, les schémas de projection de Chorin et faiblement-compressible démontrent pleinement leurs aptitudes.

FIGURE D.6 – Comparaison des champs de vitesse de la cavité à $Re = 1000$ pour un maillage 64×64

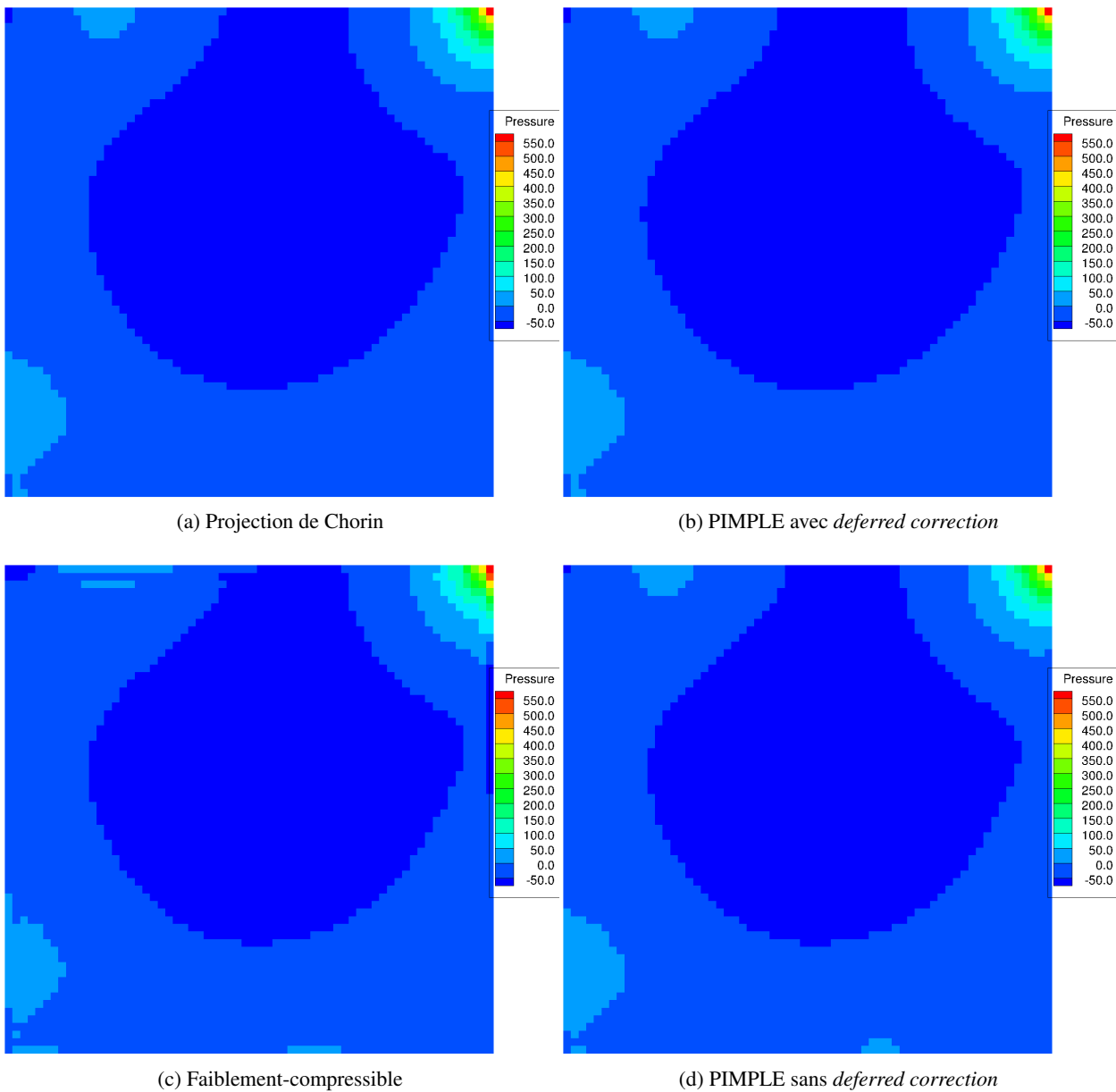


FIGURE D.7 – Comparaison des champs de pression de la cavité à $Re = 1000$ pour un maillage 64×64

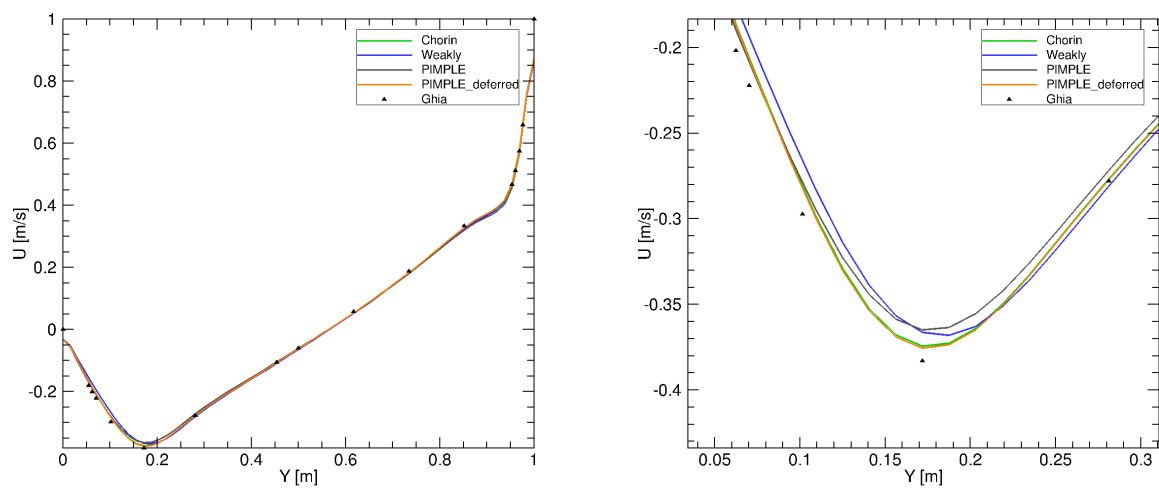


FIGURE D.8 – Comparaison des profils verticaux en $x = L/2$ de la vitesse horizontale de la cavité à $Re = 1000$ pour un maillage 64×64

Annexe E

Implémentation d'un critère de raffinement basé sur la matrice hessienne de pression

Théorie

La matrice hessienne de pression est la matrice carrée des dérivées partielles secondes de pression :

$$\mathbf{H}_p = \begin{pmatrix} \frac{\partial^2 p}{\partial x^2} & \frac{\partial^2 p}{\partial x \partial y} & \frac{\partial^2 p}{\partial x \partial z} \\ \frac{\partial^2 p}{\partial z \partial x} & \frac{\partial^2 p}{\partial y^2} & \frac{\partial^2 p}{\partial y \partial z} \\ \frac{\partial^2 p}{\partial z \partial x} & \frac{\partial^2 p}{\partial z \partial y} & \frac{\partial^2 p}{\partial z^2} \end{pmatrix} \quad (\text{E.1})$$

Les critères de raffinement de maillage qui sont basés sur des dérivées secondes sont plus sensibles aux variations de champ. Une matrice hessienne de pression va donc permettre de raffiner les principales structures d'un écoulement plus finement que ne le ferait un critère basé sur une dérivée d'ordre 1 comme le gradient de pression. Les couches limites, où le champ de pression évolue peu à haut Reynolds, restent préservées. Un autre avantage de recourir à un tel critère est la possibilité d'introduire du raffinement anisotrope.

Dans le code WCCH, les critères de raffinement possibles suivant la physique de l'écoulement sont variés : critère sur la vorticit , le gradient de pression ou le gradient des vitesses. Ils reposent sur des d riv es spatiales d'ordre 1.

Pour  toffer les types de crit res disponibles, il a  t  choisi d'impl menter un crit re bas  sur la matrice hessienne de pression. Non pas pour la possibilit  de raffinement anisotrope qu'il offre mais plus pour ses qualit s de crit re discriminant pour les structures d' coulement.

Bien que moins r pandu que les autres crit res d'ordre 1, ce crit re est par exemple employ  en hydrodynamique navale par le code ISIS-CFD avec de bons r sultats au voisinage de la surface libre. [Wackers et al., 2013, Wackers et al., 2014a, Wackers et al., 2014b, Wackers et al., 2015].

Il existe diff rentes proc dures de calcul du crit re   partir de la matrice hessienne. Celle que l'on a retenue reste simple et utilise la norme au sens de Frobenius :

$$\text{Crit} = \sqrt{\mathbf{H}_p \otimes \mathbf{H}_p^T} \quad (\text{E.2})$$

Ensuite le calcul des d riv es d'ordre 2 repr sente une op ration bien plus complexe que celle des d riv es d'ordre 1. Pour  tre consistant, il n cessite une reconstruction des champs de pression   l'ordre 3, ou un calcul des gradients de pression   l'ordre 2, tout d pend de la proc dure adopt e. Dans [Wackers et al., 2014a], les champs de pression sont calcul s   l'aide d'une approximation d'ordre 3 au sens des

moindres carrés. Les termes de la matrice hessienne se déduisent alors facilement à partir des coefficients du polynôme d'ordre 3. Nous suivons ici une approche différente qui sépare le calcul du hessien en deux temps : calcul des gradients de pression aux centres des faces des cellules puis calcul des dérivées secondes. Sur grille cartésienne uniforme, cette approche est consistante puisque les gradients de pression peuvent être déterminés à l'ordre 2 à partir de schémas centrés. En revanche, aux interfaces de raffinement, le calcul d'un gradient de pression à l'ordre 2 n'est plus immédiat.

Dans la suite nous choisissons de présenter trois méthodes de calcul des dérivées spatiales dans la direction normale à l'interface de raffinement. Ces trois méthodes sont de précision et de complexité croissantes. Elles ne concernent que les cellules fines. Le calcul pour les cellules grossières est moins problématique. La précision et l'ordre de convergence des dérivées spatiales ont été validés sur un maillage raffiné localement avec un champ de pression initialisé selon l'écoulement stationnaire non-visqueux de Taylor-Green (voir Fig. E.1)

$$p(x, y, t) = \frac{\rho U_0^2}{4} e^{-\frac{16\pi^2 \nu t}{L^2}} \left[\cos\left(4\pi \frac{x}{L}\right) + \cos\left(4\pi \frac{y}{L}\right) \right] \quad (\text{E.3})$$

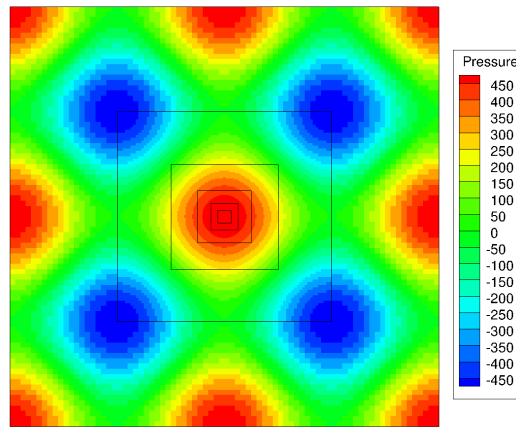


FIGURE E.1 – Champ de pression servant à la validation du calcul des dérivées spatiales

6 niveaux de raffinement ont été utilisés pour un nombre de 76 blocs au total. Le nombre de cellules par bloc est ensuite adapté (4, 8, 10, 12, 14, 16 et 20) pour effectuer une convergence en maillage. L'erreur de calcul des dérivées spatiales par rapport à la solution analytique est évaluée selon les normes L_1 , L_2 et L_∞ .

$$\begin{aligned} \epsilon_{L_1} &= \frac{\int |p - p_{sol}|}{\int |p_{sol}|} \\ \epsilon_{L_2} &= \frac{\int |p - p_{sol}|^2}{\int |p_{sol}|^2} \\ \epsilon_{L_\infty} &= \max \frac{|p - p_{sol}|}{|p_{sol}|} \end{aligned} \quad (\text{E.4})$$

La figure E.2 montre la configuration d'une interface de raffinement de maillage avec la localisation des inconnues.

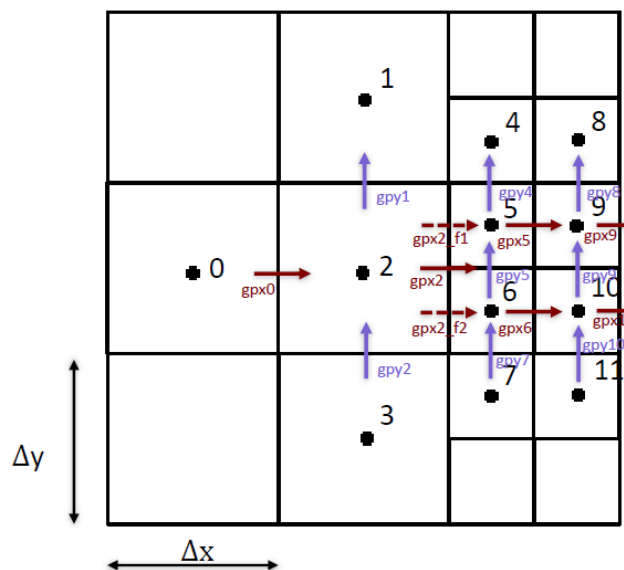


FIGURE E.2 – Emplacement des inconnues à une interface de raffinement de maillage.

Méthode 1

Cette première méthode s'appuie sur le calcul d'un gradient tel qu'il est fait dans WCCH pour évaluer par exemple le flux visqueux.

$$gpx2_{f1} = \frac{p_5 - p_2}{3/4\Delta x} \quad (E.5)$$

Un développement en série de Taylor montre que ce calcul n'est pas consistant :

$$\frac{p_5 - p_2}{3/4\Delta x} = \frac{\partial p}{\partial x} + \frac{\Delta y}{3\Delta x} \frac{\partial p}{\partial y} + O(\Delta x, \Delta y) \quad (E.6)$$

Ce résultat soulève des questions quant à la convergence du solveur volumes finis pour le calcul des flux visqueux. Néanmoins nous n'avons pas observé de problème de convergence en lien avec cette inconsistance de flux. Dans [Losasso et al., 2004] les auteurs font mention du même constat.

Le calcul d'une dérivée seconde avec une telle approximation du gradient diverge :

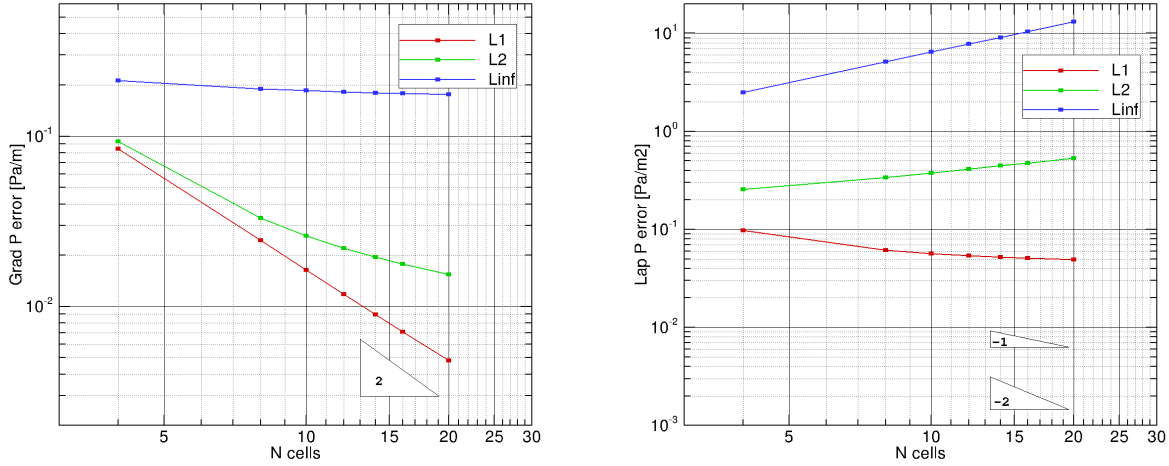
$$\frac{gpx5 - gpx2_{f1}}{1/2\Delta x} = \frac{5}{4} \frac{\partial^2 p}{\partial x^2} - \frac{2\Delta y}{3\Delta x^2} \frac{\partial p}{\partial y} + \frac{\Delta y^2}{12\Delta x^2} \frac{\partial^2 p}{\partial y^2} + \frac{\Delta y}{3\Delta x} \frac{\partial^2 p}{\partial x \partial y} + O(\Delta x, \Delta y) \quad (E.7)$$

Les résultats de convergence sont présentés sur la figure E.3. Seule la norme L_∞ a du sens et est révélatrice de la situation étant donné que le nombre d'interfaces de raffinement est fixe pour les différents maillages. On remarque que les calculs de dérivées première et seconde ne sont pas consistants voire divergent pour $\frac{\partial^2 p}{\partial x^2}$

Méthode 2

La deuxième méthode de calcul de gradient repose sur une dérivée seconde dans la direction tangentielle à l'interface de raffinement.

$$gpx2_{f1} = gpx2 + \frac{\Delta y}{4} \frac{1}{2} \left(\frac{gpy4 + gpy8}{\Delta x} - gpy1 + \frac{gpy7 + gpy10}{\Delta x} - gpy2 \right) \quad (E.8)$$



(a) Convergence dérivée $\partial p/\partial x$

(b) Convergence dérivée seconde $\partial^2 p/\partial x^2$

FIGURE E.3 – Convergence du calcul des dérivées de pression pour la méthode 1

soit

$$gpx2_{f1} = \frac{\partial p}{\partial x} - \frac{\Delta x}{3} \frac{\partial^2 p}{\partial x^2} + \frac{\Delta y^2}{24\Delta x} \frac{\partial^2 p}{\partial y^2} + O(\Delta x^2, \Delta y^2, \Delta x\Delta y) \quad (E.9)$$

Le calcul de gradient est un ordre supérieur par rapport à la méthode précédente. Mais le calcul de la dérivée seconde n'est toujours pas consistant :

$$\frac{gpx5 - gpx2_{f1}}{1/2\Delta x} = \frac{5}{4} \frac{\partial^2 p}{\partial x^2} - \frac{\Delta y^2}{12\Delta x^2} \frac{\partial^2 p}{\partial y^2} - \frac{5\Delta x}{48} \frac{\partial^3 p}{\partial x^3} + \frac{76\Delta y^3}{192\Delta x^2} \frac{\partial^3 p}{\partial y^3} - \frac{5\Delta y}{64} \frac{\partial^3 p}{\partial x^2\partial y} + \frac{\Delta y^2}{16\Delta x} \frac{\partial^3 p}{\partial x\partial y^2} + O(\Delta x^2) \quad (E.10)$$

La figure E.4 confirme ce résultat.

Méthode 3

La dernière méthode permet un calcul qui est pleinement d'ordre 2 :

$$gpx2_{f1} = \frac{gpy2 - gpy1}{30} + \frac{16}{15}gpx2 + \frac{7gpx5 - 8gpx6}{15} \quad (E.11)$$

En effet :

$$gpx2_{f1} = \frac{\partial p}{\partial x} + \frac{5\Delta x^2}{96} \frac{\partial^3 p}{\partial x^3} + \frac{\Delta x\Delta y}{8} \frac{\partial^3 p}{\partial x^2\partial y} + O(\Delta x^3, \Delta y^3, \Delta x^2\Delta y, \Delta x\Delta y^2) \quad (E.12)$$

La dérivée seconde est cette fois-ci bien d'ordre 1 :

$$\frac{gpx5 - gpx2_{f1}}{1/2\Delta x} = \frac{\partial^2 p}{\partial x^2} - \frac{\Delta x}{12} \frac{\partial^3 p}{\partial x^3} - \frac{\Delta y}{4} \frac{\partial^3 p}{\partial x^2\partial y} + O(\Delta x^2) \quad (E.13)$$

comme le démontre la figure E.5

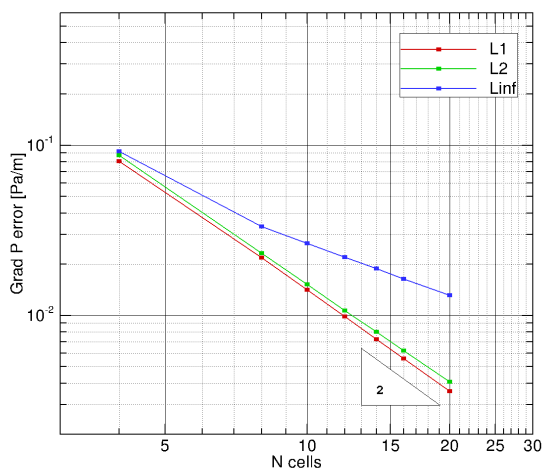
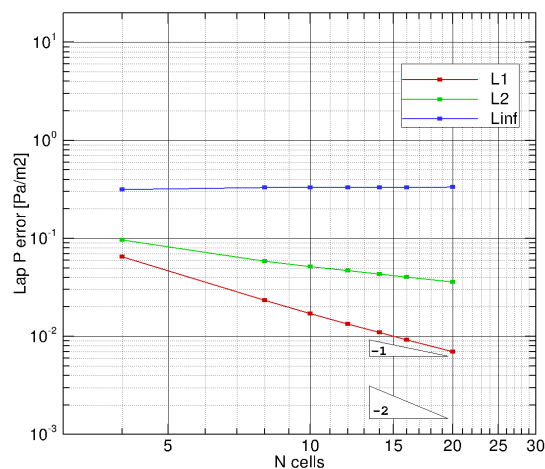
(a) Convergence dérivée $\partial p/\partial x$ (b) Convergence dérivée seconde $\partial^2 p/\partial x^2$

FIGURE E.4 – Convergence du calcul des dérivées de pression pour la méthode 2

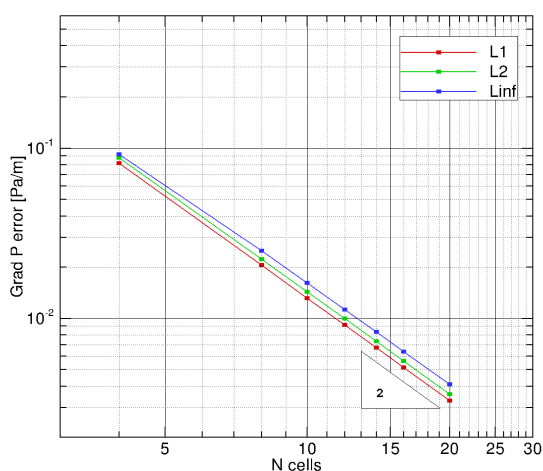
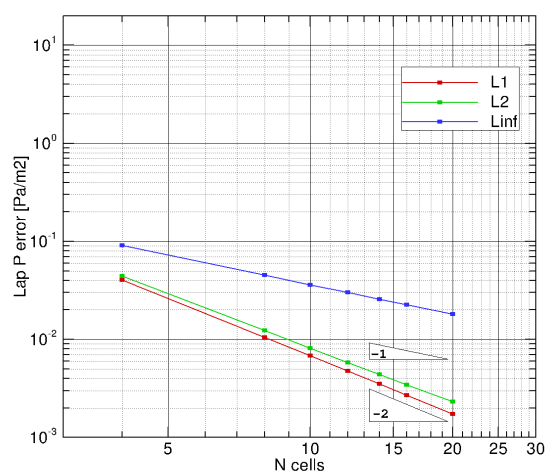
(a) Convergence dérivée $\partial p/\partial x$ (b) Convergence dérivée seconde $\partial^2 p/\partial x^2$

FIGURE E.5 – Convergence du calcul des dérivées de pression pour la méthode 3

Méthode de calcul de dérivées dans la direction tangentielle

Le principal problème est l'évaluation du terme $\partial^2 p/\partial x \partial y$ avec le calcul de $\partial p/\partial y$ sur les interfaces de raffinement de normale x (cf. Fig. E.6). Une formule de calcul à l'ordre 2 est la suivante :

$$gpy2_{f1} = \frac{5gpy1 - gpy2 + 8gpy5}{12} = \frac{\partial p}{\partial y} + \frac{\Delta y^2}{32} \frac{\partial^3 p}{\partial x^2 \partial y} + \frac{\Delta x \Delta y}{8} \frac{\partial^3 p}{\partial x \partial y^2} + O(\Delta x^2) \quad (\text{E.14})$$

ou encore

$$gpy2_{f2} = \frac{-gpy1 + 5gpy2 + 8gpy5}{12} \quad (\text{E.15})$$

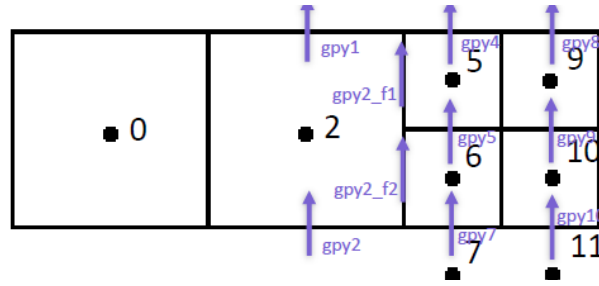


FIGURE E.6 – Convergence du calcul de dérivée seconde $\partial^2 p / \partial x \partial y$

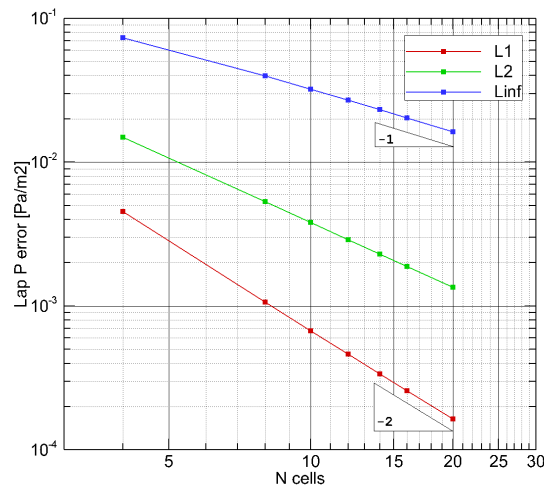


FIGURE E.7 – Convergence du calcul de dérivée seconde $\partial^2 p / \partial x \partial y$

Validation du calcul du critère

Pour le champ de pression initialisé avec le champ de l'écoulement de Taylor-Green, nous pouvons calculer analytiquement le critère attendu :

$$\text{Crit} = \frac{\rho}{4} (4\pi)^2 \sqrt{\cos^2(4\pi x) + \cos^2(4\pi y)} \quad (\text{E.16})$$

Nous effectuons une convergence en maillage du calcul du critère (Fig. E.8).

Remarques

Le calcul rigoureux et exact d'une matrice hessienne aux interfaces de raffinement peut sembler anecdotique dans la mesure où ce calcul ne contribue pas directement à la résolution des équations de Navier-Stokes mais sert juste à déterminer un critère de convergence. De plus, le critère est ensuite moyenné par bloc ce qui a pour effet de limiter la contribution éventuellement imprécise des cellules aux interfaces de raffinement. Cependant l'expérience numérique a montré que les méthodes 1 et 2 donnaient des valeurs de critère aberrantes ayant pour effet de déstabiliser le processus de raffinement avec des poches de raffinement non attendues dans certaines régions de l'écoulement. Un calcul consistant est donc malgré tout nécessaire, avec toutes les contraintes que cela peut avoir sur le coût de calcul pour atteindre une telle précision.

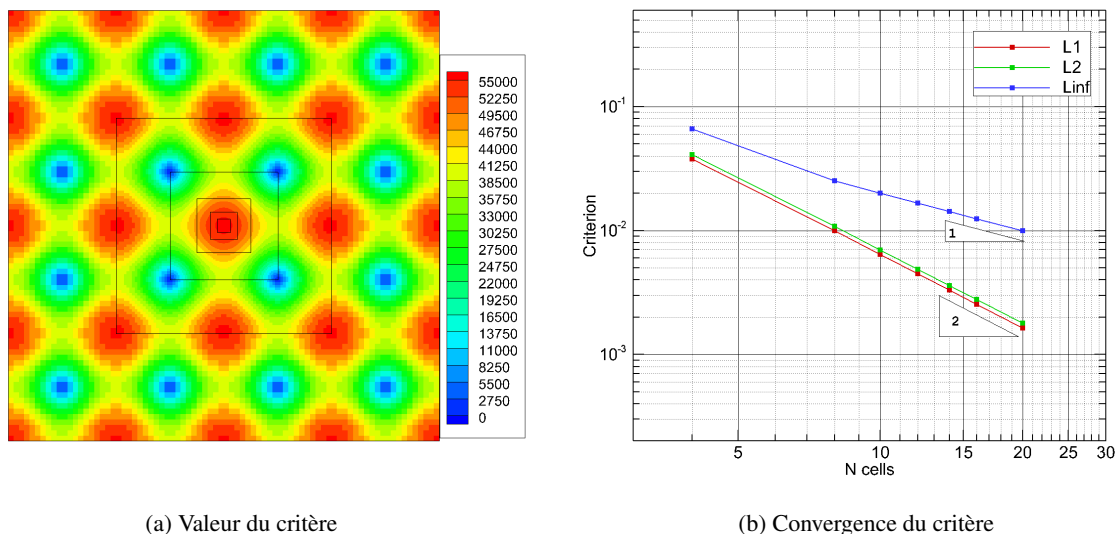


FIGURE E.8 – Convergence du critère sur un champ de pression de l'écoulement de Taylor-Green

Applications numériques

Dans les applications qui suivent, la décision de raffiner ou déraffiner un bloc de maillage s'effectue à partir de la moyenne du critère sur les cellules du bloc. A partir de là un écart-type σ global est calculé. Un bloc est raffiné si :

$$\text{Crit} > \text{RefCoeff} \times \sigma \quad (\text{E.17})$$

A l'inverse il sera déraffiné si :

$$\text{Crit} < \text{DerefCoeff} \times \sigma \quad (\text{E.18})$$

RefCoeff et *DerefCoeff* sont des coefficients qui permettent de fixer les seuils de raffinement et de déraffinement. Ils sont à ajuster en fonction de l'écoulement et de la précision du maillage souhaitée. Par exemple pour l'écoulement de Taylor-Green que nous présentons ci-dessous, ils sont fixés respectivement à 0.8 et 0.3.

Tourbillons de Taylor-Green

D'après la validation effectuée précédemment sur le calcul du critère pour ce cas test, nous nous attendons à ce que le raffinement s'opère principalement aux centres des tourbillons (cf. Fig. E.8). La figure E.9 confirme cela.

Cavité entraînée à Reynolds 1000

L'application du critère de hessien de pression sur cet écoulement conduit à un raffinement dans les coins supérieurs gauche et droit, c'est-à-dire là où la vitesse est singulière et où les variations de pression sont importantes. Sur la figure E.10 nous comparons les champs de pression dans le coin supérieur droit pour un calcul effectué sur une grille cartésienne uniforme de 64×64 cellules et sur une grille AMR dont le niveau le plus grossier correspond à la résolution spatiale d'une grille uniforme de 64×64 et à une grille de 2048×2048 pour le niveau le plus fin. Nous traçons également le profil de vitesse horizontale. Il apparaît que le minimum du profil est mieux prédit avec la grille AMR qu'avec la grille cartésienne uniforme de 64×64

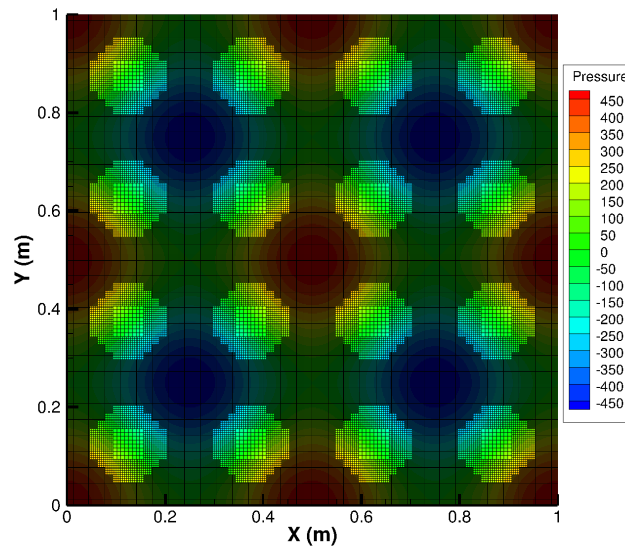


FIGURE E.9 – Maillage adapté selon le hessien de pression pour les tourbillons de Taylor-Green

cellules. Cependant le résultat reste de qualité inférieure à une grille uniforme de 128×128 cellules ; signe que le raffinement des coins supérieurs du domaine n'est pas suffisant pour améliorer la précision globale de l'écoulement. Il faudrait sans doute ajouter d'autres critères de raffinement pour accroître la précision (comme celui sur la vorticit  pour capter les recirculations).

Cylindre   Reynolds 40

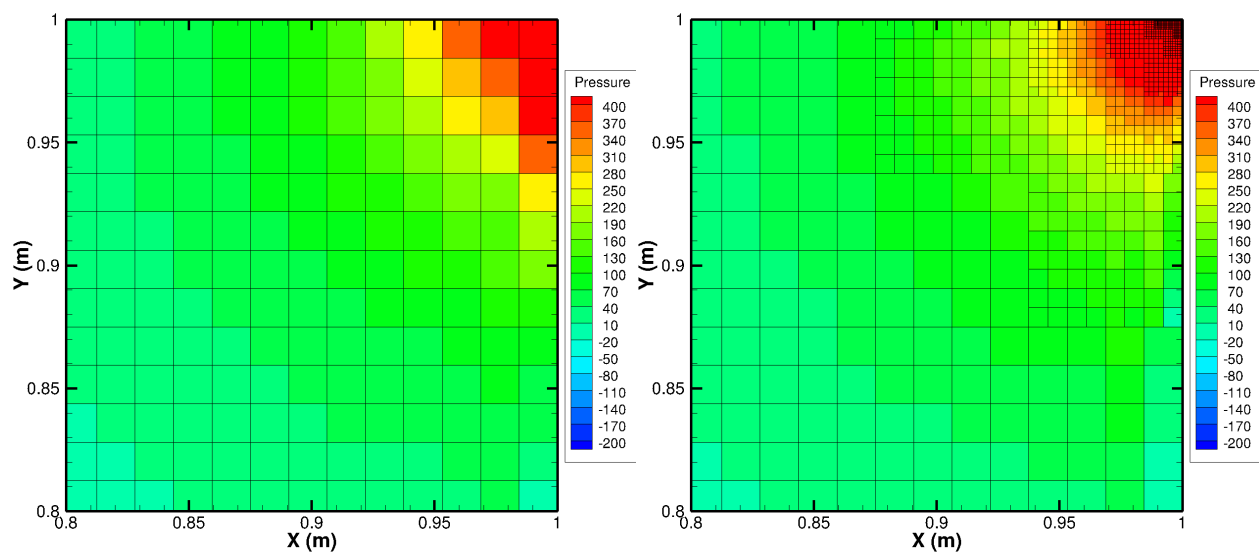
Enfin le dernier cas test porte sur l' coulement stationnaire autour d'un cylindre   $Re = 40$. Nous effectuons deux simulations de l' coulement : une avec une grille statique et une autre avec une grille adaptative selon le hessien de pression. Dans les deux cas, le niveau le plus grossier a une discr tisation spatiale telle que $\frac{D}{\Delta x} = 3.7$ et $\frac{D}{\Delta x} = 60$ pour le niveau le plus fin.

La figure E.11 pr sente les champs de pression, de vitesse et les coefficients de pression et de tra n e.

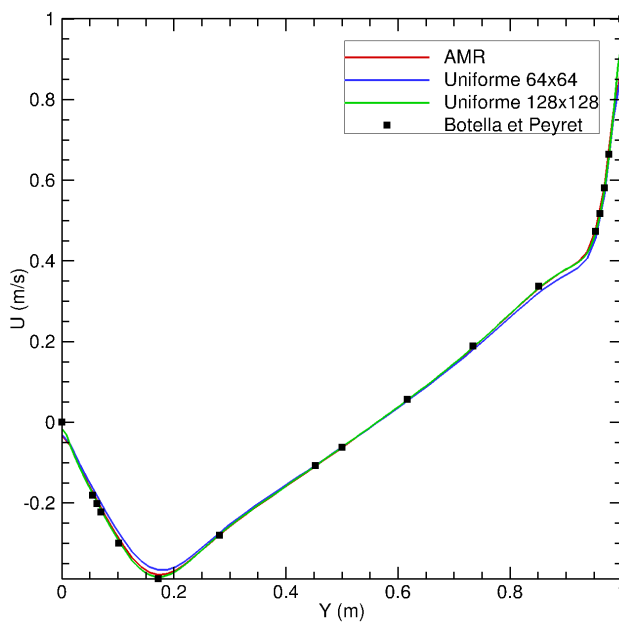
Dans le cas du calcul AMR, le raffinement adaptatif s'est concentr  au voisinage du cylindre, l  o  les variations de pression sont importantes. Le sillage n'est pas d'ailleurs pas raffin .

A noter que dans le cas du calcul avec raffinement adaptatif, la grille comporte 10528 cellules pour un temps de calcul total de 1 heure sur 12 c urs. Le calcul sur grille statique a dur  10 heures sur 24 c urs pour un nombre de cellules de 435712.

En conclusion un crit re bas  sur la matrice hessienne de pression a  t  impl ment  en faiblement-compressible. Le calcul consistant des d riv es secondes de la matrice n cessite la reconstruction du champ de pression   l'ordre 3. Ce calcul a donc un c t  algorithmique important.

(a) Pression pour la grille uniforme 64×64

(b) Pression pour la grille AMR



(c) Profil de vitesse horizontale

FIGURE E.10 – Champ de pression et profil de vitesse pour l'écoulement de cavité entraînée à $Re = 1000$

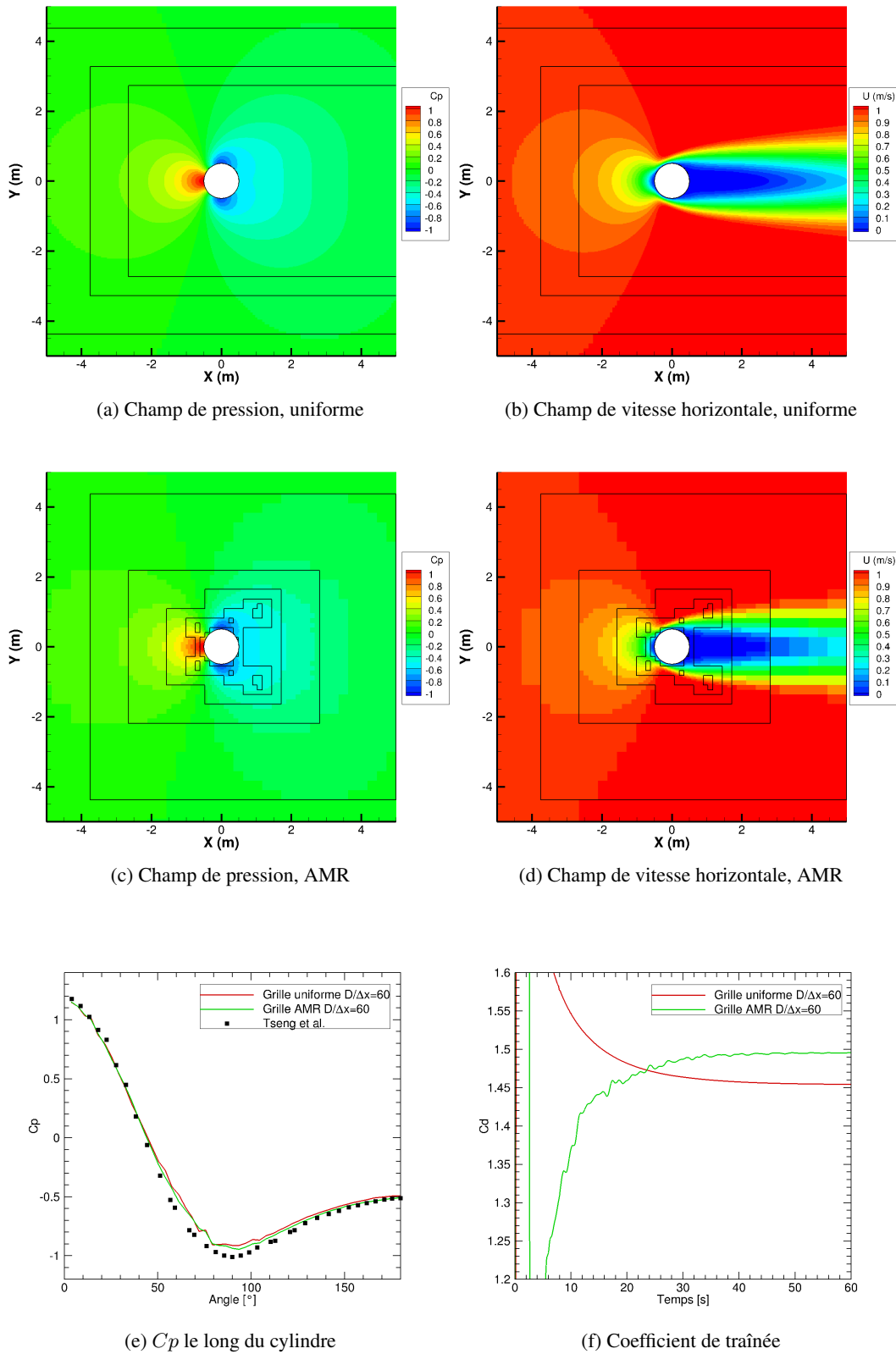


FIGURE E.11 – Écoulement 2D autour d'un cylindre à $Re = 40$

Titre : Contributions au développement d'un solveur volumes finis sur grille cartésienne localement raffinée en vue d'application à l'hydrodynamique navale

Mots clés : Volumes Finis d'ordre élevé, incompressible, maillage cartésiens raffinés localement, méthode de frontière immergée.

Résumé : L'objectif de cette thèse est de répondre au besoin d'accélérer la restitution des résultats de calcul d'un code CFD pour la simulation d'écoulements hydrodynamiques quasi-incompressibles. Ce code présente l'originalité de résoudre explicitement les équations de Navier-Stokes sous l'hypothèse de faible compressibilité avec des schémas numériques d'ordre élevé. Les développements effectués visent à réduire les temps de calcul à précision équivalente.

Une première partie est consacrée à l'implémentation d'une formulation purement incompressible avec une résolution implicite de la pression par un schéma de projection. La formulation incompressible autorise des pas de temps plus grand en s'affranchissant de la vitesse du son, mais au prix d'une algorithmique plus complexe et de la nécessité de résoudre un système linéaire. La comparaison des deux formulations,

faiblement-compressible et incompressible, tend à montrer la pertinence du schéma de projection pour les écoulements laminaires instationnaires.

Un deuxième axe de développement a consisté en la proposition d'une amélioration de la méthode de frontière immergée initialement présente dans le code. Si les résultats obtenus ne sont pas encore pleinement satisfaisants, ils montrent que la montée en ordre d'une méthode de frontière immergée peut être moins contraignante en formulation incompressible.

Enfin la dernière partie traite de l'immersion rapide et robuste de géométries complexes telles qu'elles peuvent être rencontrées dans l'industrie. La localisation géométrique par arbre octal permet d'évaluer rapidement une fonction de distance signée indispensable pour la méthode de frontière immergée.

Title : Development of a numerical solver based on a finite volume method on locally refined grid for hydrodynamic flows

Keywords : Finite volume method, incompressible, locally refined grid, immersed boundary method

Abstract : An original strategy to address hydrodynamic flow was recently proposed through a high-order weakly-compressible Cartesian grid approach. The method is based on a fully-explicit temporal scheme for solving the Navier-Stokes equations. The present thesis aims to reduce the computational time required to obtain the results without deteriorating the accuracy.

A first part is dedicated to the implementation of a truly incompressible formulation with an implicit solution for the pressure field through a projection scheme. The incompressible solver allows larger time step size for time integration since the speed of sound tends to infinity. In return the algorithms are no longer straightforward and a linear system has to be solved through the Pressure Poisson Equation.

Comparisons carried out between both formulations show that the projection scheme can be better adapted to efficiently simulate unsteady viscous flows.

Then an improvement of the immersed boundary method has been proposed. Results are not fully satisfactory for now. However, it seems easier to develop a numerical scheme for the incompressible approach rather than the weakly-compressible one.

Finally, the last part addresses the setup up of complex triangulations in immersed boundary simulations. A fast and robust procedure is developed for distance computation with an octree data structure.