



HAL
open science

Simulation numérique des ballotements d'ergols dans les réservoirs de satellites en microgravité et à faible nombre de Bond

Mathieu Lepilliez

► **To cite this version:**

Mathieu Lepilliez. Simulation numérique des ballotements d'ergols dans les réservoirs de satellites en microgravité et à faible nombre de Bond. Mécanique des fluides [physics.class-ph]. Université Paul Sabatier - Toulouse III, 2015. Français. NNT : 2015TOU30248 . tel-01393960

HAL Id: tel-01393960

<https://theses.hal.science/tel-01393960>

Submitted on 8 Nov 2016

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.



Université
de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le 09/12/2015 par :

MATHIEU LEPILLIEZ

Simulation numérique des ballotements d'ergols dans les réservoirs de satellites en microgravité et à faible nombre de Bond

JURY

ALAIN BERLEMONT	Directeur de recherches
PIERRE BRANCHER	Professeur d'Université
BARBARA BUSSET	Ingénieur ADS
JEAN-LUC ESTIVALEZES	Professeur associé
FRÉDÉRIC GIBOU	Professeur
THIBAUT MÉNARD	Maître de Conférences
JEAN MIGNOT	Docteur - Ingénieur CNES
SÉBASTIEN TANGUY	Maître de Conférences
STÉPHANE VINCENT	Professeur d'Université

École doctorale et spécialité :

MEGEP : Dynamique des fluides

Unité de Recherche :

Institut de Mécanique des Fluides de Toulouse (UMR 5502)

Directeur de Thèse :

Sébastien TANGUY

Rapporteurs :

Alain BERLEMONT et Stéphane VINCENT

Table des matières

Introduction	2
1 Etat de l'art : Ballotements en microgravité	5
1.1 Études théoriques - Analogie au système masse-ressort	8
1.1.1 Ballotement Latéral	8
1.1.2 Expressions empiriques	9
1.2 Études expérimentales	11
1.2.1 Essais en flottaison neutre	11
1.2.2 Drop Towers et Vols paraboliques	13
1.2.3 SloshSat - FLEVO	14
1.2.4 Expériences dans l'ISS - SPHERES-Slosh	15
1.3 Études numériques	17
2 Méthodes Numériques	23
2.1 Méthodes numériques de suivi d'interfaces	24
2.1.1 Introduction au méthodes de suivi d'interface	24
2.1.2 Méthode Level-Set	26
2.2 Navier-Stokes - écoulements Diphasiques	28
2.2.1 Modèle hydrodynamique des écoulements diphasiques	28
2.2.2 Méthode de projection pour les équations de Navier-Stokes	29
2.2.3 La méthode Ghost Fluid	30
2.2.4 Traitement du terme visqueux	34
2.2.5 Schémas de discrétisation et restrictions sur le pas de temps	39
2.3 Résolution du champ de Pression	41
2.3.1 Généralités sur les méthodes itératives	41
2.3.2 Le Solveur BlackBox MultiGrid	45
2.4 Modélisation des sauts à l'interface	54
2.4.1 Principe de la méthode Voronoï	54
2.4.2 Tests comparatifs entre le Ghost Fluid et le Voronoï	54
Solving elliptic problems with discontinuities on irregular domains - the Voronoi In- terface Method	82

3	Modélisation des Frontières Immergées	83
	An efficient Two-Phase Flow Solver in Irregular Domains	83
4	Étude des ballotements dans les réservoirs	129
4.1	Équations de Navier-Stokes	130
4.1.1	Force d'entraînement et Force d'inertie	130
4.1.2	Équations de Navier-Stokes adimensionnalisées	130
4.2	Première approche du ballotement - cas simple	132
4.2.1	Rôle du nombre de Bond impulsional	132
4.2.2	Rôle du nombre de Bond centrifuge	136
4.2.3	Convergence des simulations	147
4.3	Étude de la manœuvre GAP/SUP	148
4.3.1	Description de la manœuvre GAP/SUP	149
4.3.2	Visualisation et Trajectoire	150
4.3.3	Centre de masse	152
4.3.4	Forces et moments	153
4.3.5	Convergence des simulations	153
	Conclusions et Perspectives	157

Introduction

Depuis plusieurs décennies les industriels du secteur aérospatial essaient de comprendre la phénoménologie complexe liée aux ballotements dans les réservoirs de satellite qui interviennent lorsque les fluides dans le réservoir, à savoir un ergol et un gaz pressurant, sont mis en mouvement lors des phases de manœuvre du satellite. La dynamique observée par ces écoulements diphasiques génère des efforts et des couples perturbateurs qui peuvent nuire au contrôle ainsi qu'à la stabilité globale du satellite. La recherche sur ces phénomènes a pris un nouvel essor lorsqu'une nouvelle génération de satellites dits 'agiles' a été mise en place. Cette gamme de satellite, incluant la célèbre constellation Pléiade, exécute constamment des manœuvres afin de se réorienter pour des prises de vues de la terre. Toutefois le contrôle du satellite nécessite d'attendre la stabilisation post-manœuvre, qui peut s'avérer longue lorsque les corrections de trajectoire à effectuer sont importantes et mal dimensionnées. C'est dans ce contexte qu'une étude *R&T* Ballotements d'ergols en très faible poussée, menée conjointement par le Centre National des Études Spatiales (CNES) et Airbus Defence & Space (ADS), a été mise en place afin de définir des modèles analytiques adaptés. Un des objectifs de cette thèse est de modéliser ces phénomènes de ballotement à l'aide d'un code de simulation numérique afin de fournir des données fiables pour établir les modèles analytiques.

Ce manuscrit est constitué de quatre chapitres.

Le chapitre 1 situe la problématique du ballotement dans les réservoirs de satellite et le cadre dans lequel s'inscrit cette thèse, ainsi qu'un état de l'art autour du phénomène de ballotements en microgravité. Des modèles analytiques linéaires proposés par la NASA ainsi que toute une variété d'expériences y sont présentés, et mettent en évidence toute la complexité de la modélisation de cette phénoménologie qui inclut des effets liés à la présence de deux fluides en mouvement dans un domaine fermé.

Le chapitre 2 présente le code de calcul utilisé pour réaliser les simulations numériques. C'est un code basé sur la résolution des équations de Navier-Stokes, ainsi qu'une méthode Level-Set/Ghost Fluid pour capturer le comportement de l'interface. Dans cette thèse des algorithmes d'optimisation de calcul telle que la méthode BlackBox Multigrid ont été implémentés au sein du code pour améliorer ses performances. Une étude est également présentée sur la façon la plus adéquate pour calculer et discrétiser les termes liés aux effets visqueux dans des écoulements diphasiques. Une nouvelle méthode permettant de prendre en compte des sauts à l'interface d'ordre 2 en espace est également proposée dans le cadre de ce chapitre.

Le réservoir contenant le fluide peut avoir différentes géométries selon les cas à modéliser, avec des formes sphérique ou cylindro-sphérique. Pour modéliser ces parois solides, une méthode de

frontière immergée est mise en place. Le chapitre 3 présente la mise en place des méthodes décrites dans le chapitre 2 dans le contexte de ces frontières immergées. De nouvelles notions apparaissent, avec notamment la possibilité de la création d'une ligne triple lorsque l'interface atteint la paroi solide. Plusieurs validations avec des cas analytiques sont proposées.

Le chapitre 4 présente les résultats obtenus au cours de cette thèse sur le phénomène de ballotement. Une manœuvre simple réalisée régulièrement sur les satellites est la manœuvre GAP/SUP, qui consiste à réorienter le satellite en lui faisant faire un virage en lacet. Cette manœuvre est composée d'une phase de forçage durant laquelle le satellite est mis en rotation, puis une seconde phase où la nouvelle position du satellite est atteinte mais les fluides dans le réservoir continuent leurs déplacements et génèrent des couples perturbateurs. La première phase de cette manœuvre fait l'objet principale de ce chapitre.

Chapitre 1

Etat de l'art : Ballotements en microgravité

Sommaire

1.1 Études théoriques - Analogie au système masse-ressort	8
1.1.1 Ballotement Latéral	8
1.1.2 Expressions empiriques	9
1.2 Études expérimentales	11
1.2.1 Essais en flottaison neutre	11
1.2.2 Drop Towers et Vols paraboliques	13
1.2.3 SloshSat - FLEVO	14
1.2.4 Expériences dans l'ISS - SPHERES-Slosh	15
1.3 Études numériques	17

L'objectif de l'étude globale, qui s'inscrit dans le cadre de la *R&T* Ballotements d'ergols en très faible poussée, menée conjointement par le Centre National des Études Spatiales (CNES) et Airbus Defence & Space (ADS), est de définir des modèles analytiques adaptés aux cas de manœuvres d'attitudes en vol et d'évaluer l'impact induit sur le SCAO.

Le SCAO est le Système de Contrôle d'Attitude et d'Orbite embarqué sur le satellite. Il reçoit des commandes provenant du sol pour suivre des ordres donnés pour effectuer une mission, mais est aussi en parti automatisé pour gérer la précision de pointage et la stabilité de la ligne de visée du satellite. En effet de nombreuses perturbations peuvent provenir de l'environnement hostile dans lequel le satellite évolue, avec notamment des vents solaires, des champs magnétiques, et pour les satellites en basse orbite des résidus atmosphériques ainsi que des résidus gravitationnels générés par le champ terrestre. D'autres perturbations provenant du satellite lui-même sont également à prendre en compte avec l'inertie donnée suite à une manœuvre effectuée par de la propulsion, des réactions suite à des mouvements causés par des mécanismes internes (roues à inerties, ouverture de valves), des défauts de construction mécaniques, ou encore des ballotements générés par les matériaux flexibles (panneaux solaires, antennes) et les fluides présents dans les réservoirs. Les ordres de grandeur des amplitudes de ces différentes perturbations sont proposés dans le tableau 1.1.

	Satellite géostationnaire	Satellite en orbite basse (SPOT 1)
Gradient de gravité	négligeable	10^{-3} Nm
Pression atmosphérique	négligeable	10^{-4} Nm
Couples magnétiques	10^{-6} Nm	10^{-3} Nm
Pression de radiation solaire	10^{-6} à 10^{-4} Nm	10^{-4} Nm
Mouvement d'équipements	10^{-2} Nm	10^{-1} Nm
Ballotements des ergols	10^{-3} à 10^{-1} Nm	10^{-3} à 10^{-1} Nm

TABLE 1.1 – Comparaison de l'ordre de grandeur de l'amplitude des différentes perturbations subies par un satellite en orbite géostationnaire (~ 36000 km d'altitude) et un satellite en orbite basse (600 – 2000 km d'altitude)[10].

L'information des perturbations et l'écart par rapport à l'attitude souhaitée est donnée par des capteurs. Ces capteurs utilisent plusieurs sources d'informations externes et internes au satellite. Les informations externes sont basées sur des référentiels observables par le satellite, comme la position des étoiles, du soleil et de la terre, aussi bien dans le spectre du visible que de l'infrarouge. Les informations internes sont mesurées par des gyromètres embarqués.

Ces informations sont transmises à un ordinateur embarqué qui va estimer les corrections à effectuer pour retourner vers un état stable tout en respectant l'ordre de mission donné par l'utilisateur au sol, à l'aide de lois de contrôle. Les corrections sont ensuite effectuées à l'aide d'actionneurs, qui génèrent des couples correcteurs permettant la stabilisation. Ces actionneurs peuvent être magnétiques ou inertiels (roues à réaction, roues cinétiques).

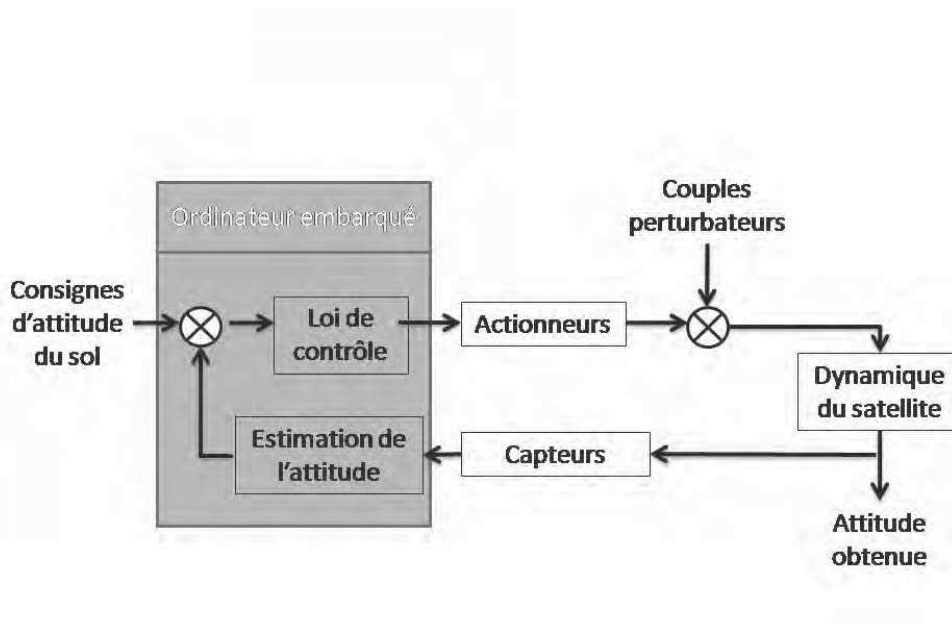


FIGURE 1.1 – Fonctionnement de la boucle SCAO embarquée dans un satellite.

Une fois ces corrections effectuées, l'ordinateur vérifie à nouveau la position et le pointage du satellite. Le principe de fonctionnement est résumé dans la figure 1.1. Un ordre est donné au satellite par le sol afin de respecter un guidage dépendant de la mission et la loi de contrôle est appliquée, les actionneurs corrigent la trajectoire du satellite. Celui-ci se déplace donc vers une nouvelle position et récupère de nouveaux couples perturbateurs. Les capteurs vérifient l'écart de la position et du pointage souhaités, la loi de contrôle est ajustée de nouveau, et ainsi de suite.

Le tableau 1.1 résume les ordres de grandeurs des différentes perturbations subies par le satellite, et nous pouvons constater que le ballotement des ergols fait partie des couples perturbateurs les plus importants, d'où la nécessité de bien appréhender les lois de contrôle qui doivent les corriger. Cette thèse s'inscrit dans le cadre de cette étude afin de modéliser de façon détaillée les phénomènes physiques intervenant durant les phases de manœuvre. Les ballotements sont générés par le fait qu'il y ait deux fluides dans les réservoirs, avec l'ergol (Hydrazine, Mono Methyl Hydrazine (MMH) ou encore du Nitrogen Tetroxide (NTO)) et une bulle de gaz (Hélium, Azote) servant à pressuriser les réservoirs. Comme le satellite consomme de l'ergol pour effectuer sa mise à poste, le gaz est nécessaire afin d'éviter la vaporisation de l'ergol qui serait causée par la dépressurisation du réservoir. Lors de manœuvres les fluides sont mis en mouvement, et des ballotements surviennent.

Les différentes manœuvres sont dans le contexte de cette étude, caractérisées par le nombre de Bond qui compare l'intensité des forces volumiques par rapport aux effets de tension de surface :

$$Bo = \frac{\text{Effets inertiels}}{\text{Effets de Tension de Surface}} \quad \text{et} \quad We = \frac{\text{Énergie cinétique}}{\text{Énergie de Tension de Surface}} \quad (1.0.1)$$

Le nombre de Weber est également décrit, et compare l'énergie cinétique du système par rapport à l'énergie de tension de surface. En fonction de la manœuvre, du remplissage du réservoir, de la forme du réservoir et de l'ergol utilisé, le nombre de Bond peut varier de 0 à 100 dans le contexte spatial.

La figure 1.2 résume les différents régimes en fonction des nombres de Bond et de Weber. Celle-ci illustre la gamme des Bonds dans laquelle s'effectuent la plupart des manœuvres en vol.

Différents outils ont été développés afin de modéliser ces phénomènes. Des modèles analytiques ont été proposés par la NASA avec les travaux d'Abramson [1, 2], qui approche le problème avec un modèle de masse-ressort. Des expériences ont été menées, avec des campagnes de flottaison neutre. L'effet d'apesanteur est également réalisable dans des drop towers et des vols paraboliques permettant ainsi de reproduire les conditions de l'environnement spatial. Des expériences ont été réalisées dans la station spatiale orbitale ISS avec SPHERES-Slosh. Le satellite SloshSat-FLEVO ayant uniquement pour but de faire des mesures in situ sur les ballotements à été envoyé dans l'espace. Enfin des simulations numériques ont été réalisées avec différents codes de calcul modélisant les écoulements diphasiques. Dans cette étude, le code commercial FLOW3D et le code DIVA, développé à l'Institut de Mécanique des Fluides de Toulouse (IMFT) ont été utilisés. Ces différentes études sont détaillées dans la suite de ce chapitre.

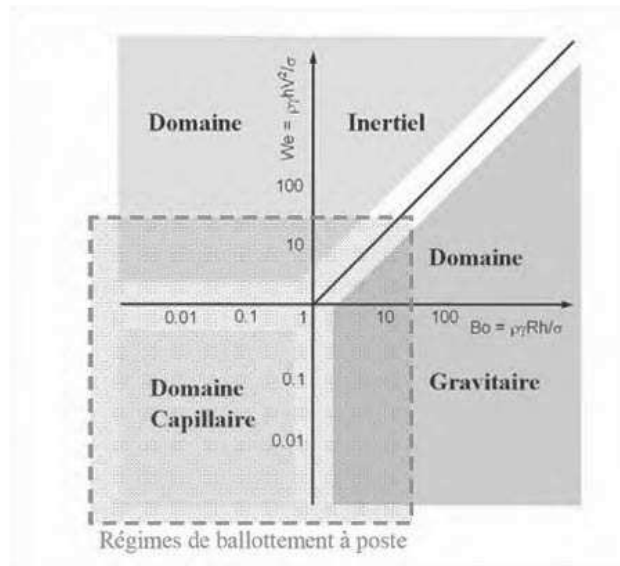


FIGURE 1.2 – Illustration des différents domaines rencontrés lors des régimes de ballotement à poste [4].

1.1 Études théoriques - Analogie au système masse-ressort

Une première approche analytique linéaire pour modéliser les phénomènes oscillants tels que les ballotements sont des modèles de pendule et de masse-ressort. En effet le ballotement peut tout simplement être considéré comme un écart à la position d'équilibre du centre de masse du réservoir, générant ainsi des efforts et couples perturbateurs. Une illustration 1D du problème, extraite de [1, 2], est proposée dans la partie suivante.

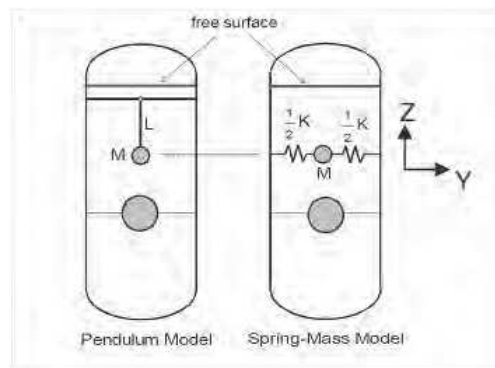


FIGURE 1.3 – Illustration du problème (1.1.1) avec une représentation pendule à gauche et masse-ressort à droite [1, 2].

1.1.1 Ballotement Latéral

La figure 1.3 illustre l'adaptation du problème en système de masse-ressort et en masse-pendule pour la modélisation d'un ballotement latéral. Si la masse est décalée de sa position d'équilibre, elle

va osciller jusqu'à revenir à une position stable. L'équation linéaire du second ordre qui définit ce problème dans la direction y est :

$$\frac{d^2y}{dt^2} + 2\xi\omega_0 \frac{dy}{dt} + \omega_0^2 y = 0 \quad (1.1.1)$$

avec $y(t)$ la position du centre de masse dans la direction latérale, ξ le taux d'amortissement, ω_0 la fréquence du premier mode de ballotement. La solution du système s'écrit :

$$y(t) = Ae^{-\xi\omega_0 t} \sin(\omega t + \varphi) \quad (1.1.2)$$

avec A l'amplitude du signal, $\omega = \sqrt{1 - \xi^2}\omega_0$ et φ le déphasage. A partir de cette solution la force résultante peut être calculée en dérivant la solution de la façon suivante :

$$F_y = m \frac{d^2y}{dt^2} = mA\omega_0^2 e^{-\xi\omega_0 t} \left[(2\xi^2 - 1)\sin(\omega t + \varphi) - 2\xi\sqrt{1 - \xi^2}\cos(\omega t + \varphi) \right] \quad (1.1.3)$$

Le modèle simple sert à illustrer l'idée de la méthode, mais il ne prend pas en compte explicitement les caractéristiques physiques du problème telles que la forme du réservoir, les propriétés physiques du fluide ainsi que le remplissage. De plus cette approche est certes aisée à mettre en place analytiquement mais reste limitée. En effet les phénomènes capturés par cette modélisation seront nécessairement linéaires, ce qui n'est pas toujours le cas.

Dans la suite de cette partie, le système masse-ressort est privilégié au modèle masse-pendule car pour la modélisation de la physique intervenant dans les réservoirs de satellite, les nombres de Bond sont petits. Dans ce cadre là, le système masse-pendule n'est pas adapté car les fréquences propres au ballotement font sortir le pendule du réservoir. Ces systèmes sont généralement employés lorsque la dynamique est plus violente, notamment pour modéliser les ballotements dans les réservoirs de lanceurs.

1.1.2 Expressions empiriques

Dans [1, 2], Abramson propose de nombreuses expressions empiriques basées sur ce modèle simple pour inclure un maximum d'informations de la physique réelle du problème pour différentes géométries simples et classiques. Ces relations ont été déduites de nombreuses expériences, et bien que réalisées en 1967, celles-ci restent une approche très correcte pour une estimation des premières fréquences de ballotements.

1.1.2.1 Fréquence

Pour des cas en microgravité avec des nombres de Bond supérieurs à 10, des valeurs empiriques de la fréquence ω sont données. La fréquence du premier mode de ballotement varie en fonction de l'accélération, de la hauteur du fluide h , de la taille du réservoir R pour les différentes géométries :

$$\omega = f(\gamma, h, R) \quad (1.1.4)$$

Nous pouvons remarquer que dans ce contexte là, la fréquence ne dépend pas explicitement du nombre de Bond, mais des caractéristiques du réservoir et de l'accélération imposée. Pour des Bonds compris entre 0.1 et 10, la loi évolue et prend en compte les effets de tensions de surface, et ω dépend des variables suivantes :

$$\omega = f(\rho, \sigma, Bo, H, l_c) \quad (1.1.5)$$

Dans cette gamme de nombres de Bond, la fréquence dépend des paramètres physiques du fluide, avec H est la dimension hydrostatique et l_c la longueur capillaire. Pour des nombres de Bond inférieurs à 0.1, la fréquence ne dépend plus du nombre de Bond. Afin d'illustrer l'évolution de la fréquence en fonction du nombre de Bond, ces équations ont été appliquées pour un cas de réservoir cylindrique et un remplissage de 17% MMH ou NTO dans [4], et les résultats sont reportés dans la figure 1.4.

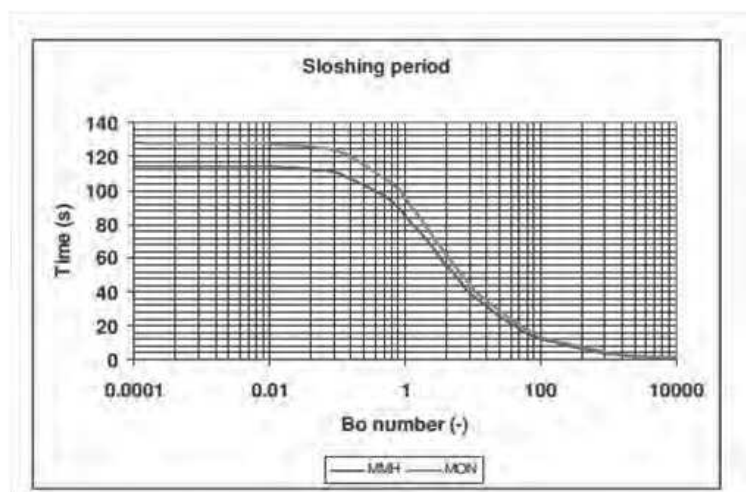


FIGURE 1.4 – Évolution de la première période de ballotement en fonction du Bond dans un réservoir cylindrique avec un remplissage de 17% de NTO et de MMH. [4]

La plus grande variation de la période se situe donc entre les nombres de Bond compris entre 0.1 et 100, ce qui correspond à la gamme étudiée dans cette thèse.

1.1.2.2 Amortissement

L'amortissement est également disponible dans [1, 2], avec une corrélation pour des nombres de Bonds supérieurs à 5 :

$$\xi = f(Bo, \omega, l_c, \nu) \quad (1.1.6)$$

avec ν la viscosité cinématique du fluide. Pour des nombres de Bond plus faibles, la corrélation ne prend plus en compte le Bond explicitement. Abramson propose :

$$\xi = f(\omega, l_c, \nu) \quad (1.1.7)$$

Les corrélations sont appliquées pour le cas précédent, un cas de réservoir cylindrique et un remplissage de 17% MMH ou NTO. Les résultats sont reportés dans la figure 1.6.

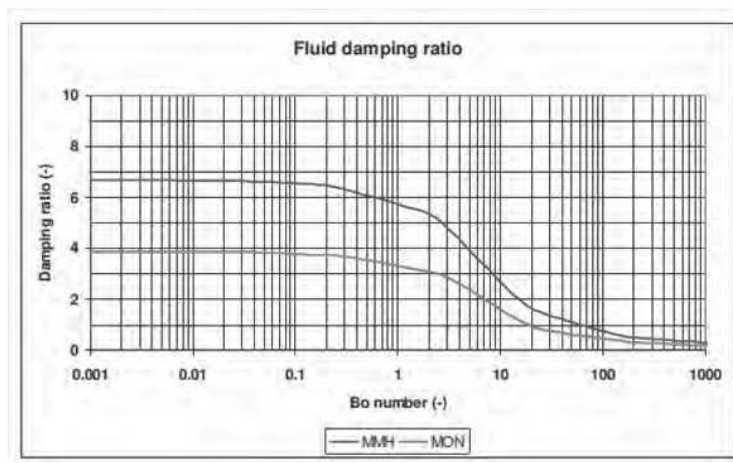


FIGURE 1.5 – Évolution de l’amortissement du ballotement en fonction du Bond dans un réservoir cylindrique avec un remplissage de 17% [4] .

Cette approche théorique donne une première idée de la phénoménologie intervenant dans le cas de réservoirs à géométrie simple. Toutefois la méthode n’est pas complète, car les résultats donnés ne sont valables que pour la première période de ballotement dans le cadre de phénomènes linéaires, or avant d’atteindre l’état d’équilibre plusieurs périodes sont à prendre en compte. De plus, un seul nombre de Bond peut être pris en compte dans cette modélisation, alors que celui-ci évolue dans le temps. Cette approche a toutefois le mérite d’être très accessible et possède le potentiel d’être améliorée en prenant par exemple plusieurs ressorts en compte avec des raideurs différentes pour mieux appréhender le problème.

1.2 Études expérimentales

Afin de valider les modèles théoriques, des données sont nécessaires. Il est toutefois délicat de recréer l’environnement spatial. Aussi diverses alternatives ont été trouvées pour recréer quelques unes des conditions en vol, avec dans le cadre de notre étude les phénomènes de microgravité.

1.2.1 Essais en flottaison neutre

Durant la campagne d’étude précédent cette thèse, des essais en flottaison neutre ont été réalisés dans les laboratoires d’ADS. Le principe de la flottaison neutre est d’utiliser deux fluides non-miscibles, ayant une masse volumique très proches l’un de l’autre. La masse volumique étant fonction de la température, une variation de quelques degrés °C peut affecter le comportement des deux fluides jusqu’à imiter les effets de la microgravité.

Pour réaliser l’analogie entre l’expérience au sol et la manœuvre en vol, le nombre de Bond Bo_{exp} est défini :

$$Bo_{exp} = \frac{(\rho_1 - \rho_2)gRh}{\sigma_{exp}\cos\theta_{exp}} \quad (1.2.1)$$

avec ρ_1 et ρ_2 les masses volumiques des deux fluides, et θ_{exp} l'angle de contact de la ligne triple, R la dimension capillaire et h la hauteur hydrostatique. La simulation des accélérations atteintes en orbite est basée dans le cadre de ces expériences sur la différence des masses volumiques, qui est en partie fonction de la température du milieu. A une température, appelée température de flottaison neutre, les deux fluides atteignent exactement la même masse volumique permettant ainsi de simuler la microgravité ($Bo=0$), tout en conservant des viscosités différentes et des efforts conséquents générés par la capillarité.

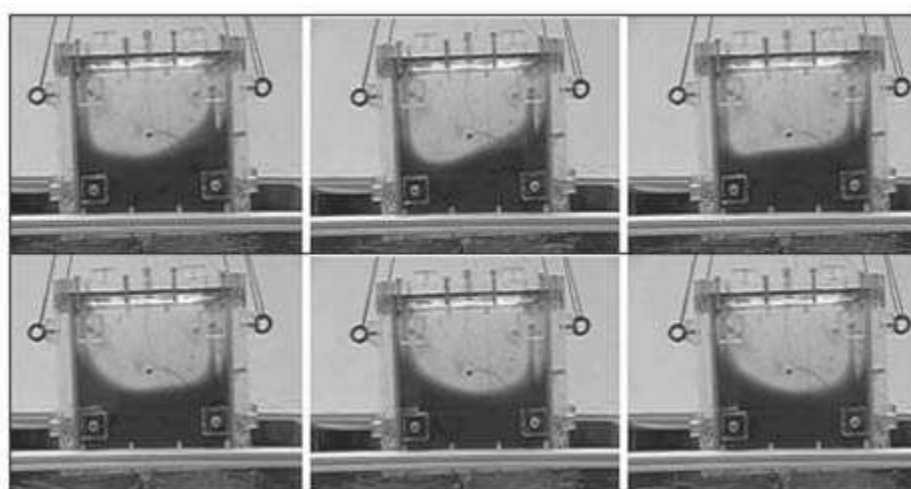


FIGURE 1.6 – Photographie du système expérimental de flottaison neutre mis en place dans les laboratoires d'ADS à Toulouse [5] .

En admettant une équivalence entre le nombre de Bond réel Eq. (1.0.1) et le nombre de Bond expérimental Eq. (1.2.2), on définit :

$$(\rho_1 - \rho_2) = f(T) = \frac{\gamma_r \sigma_{exp}\cos\theta_{exp}}{g \sigma_r\cos\theta_r} \rho_r \quad (1.2.2)$$

avec γ_r l'accélération en vol, et les variables σ_r , θ_r et ρ_r les caractéristiques de l'ergol utilisé en vol. Il est donc possible de générer des accélérations artificielles à l'aide du contrôle de la température. Une fois la température réglée, le réservoir peut être mis en rotation pour mesurer les périodes et taux d'amortissement des ballotements générés par la manœuvre. Un exemple de séquence d'une manœuvre est proposé dans la figure 1.7.

Cette méthode donne de bon résultats pour des manœuvres sur des nombres de Bond compris entre 8 et 60. Toutefois le dispositif expérimental ne permet pas la modélisation d'écoulements pour les Bond inférieurs à 8, qui sont rencontrés lors de certaines manœuvres. De plus, les rapports de masse volumique et de viscosité ne sont pas respectés, et un angle de contact est présent, alors que les ergols sont très mouillants et que l'existence d'un angle de contact au sein du réservoir est hypothétique à l'heure actuelle.

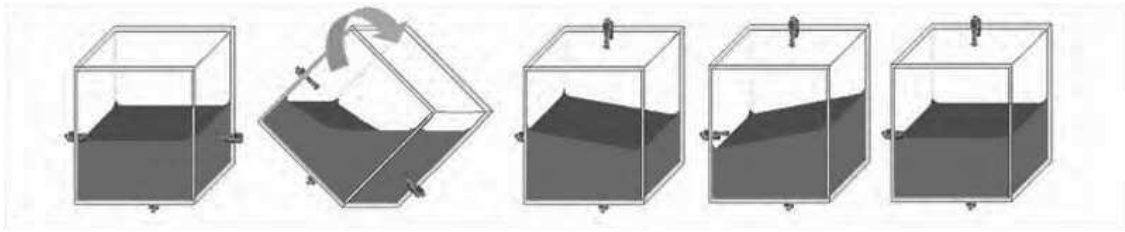


FIGURE 1.7 – Séquence d’une manœuvre effectuée par le réservoir en flottaison neutre (Extrait de [4, 5]) .

1.2.2 Drop Towers et Vols paraboliques

Des expériences nécessitant l’état d’apesanteur peuvent être réalisées autrement que dans l’espace. En effet le phénomène d’apesanteur est le fait de n’être soumis qu’à l’accélération de la gravité terrestre. Aussi le principe de chute libre peut être utilisé pour générer cet état d’apesanteur. Deux procédés sont couramment utilisés pour imiter les conditions du milieu spatial.

Les Drop Towers sont de grandes tours dans lesquelles sont lâchées des expériences en chute libre. Afin de ne limiter le bilan de force qu’à la gravité, le vide est réalisé dans ces tours, notamment pour éliminer des effets de portance et de trainée aérodynamiques. Quelques uns des sites connus pour réaliser ce genre d’expérience sont la tour souterraine de la NASA à Cleveland aux États-Unis, avec une chute libre sur 140m sur une durée de 5.18s. En Europe, le ZARM Institute à Bremen possède une drop tower de 146m, qui permet un état d’apesanteur de 9.3s. La nacelle commence l’expérience au pied de la tour. Une catapulte la projette jusqu’au sommet de la tour, permettant ainsi un aller-retour sur la hauteur de la tour. Une image de l’intérieur de la tour est proposée dans la figure 1.8.

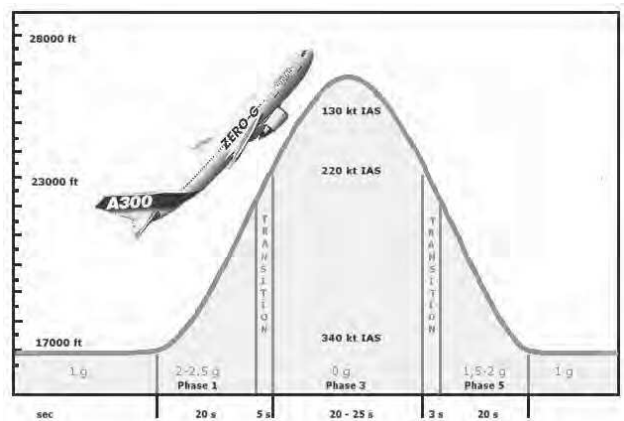


FIGURE 1.8 – A gauche - photo de l’intérieur de la droptower au ZARM Institute à Bremen. La nacelle expérimentale est chargée au sommet de la tour pour une chute libre d’environ 5s [11]. A droite - déroulement d’une parabole à bord de l’A310 AIR ZERO G [12].

Une alternative à ces équipements pour générer l’état d’apesanteur est le vol parabolique. Le principe est le même que celui-ci de la drop tower, à savoir équilibrer le bilan des forces de façon à ne ressentir que l’accélération de la gravité terrestre, et d’annuler les autres efforts. Pour réaliser

cela, un avion se lance dans une parabole en compensant les efforts de traînée et de portance avec la poussée des moteurs. Les paraboles sont réalisées dans un avion Airbus A310 (en Europe) pour un état d'apesanteur d'environ 22s. C'est le CNES qui est responsable de la gestion de ces vols en France. Le déroulement de la parabole est résumé dans la figure 1.8.

Ces méthodes sont régulièrement utilisées dans le cadre d'études sur du matériel destiné à être embarqué sur un satellite. Toutefois les fenêtres de temps en apesanteur sont trop faibles devant les périodes caractéristiques des ballotements. En effet, en fonction du nombre de Bond, la période du premier rebond peut durer entre 40s et 180s, ce qui dépasse largement le temps donné par ces méthodes expérimentales. Celles-ci ne sont donc pas adaptées à la modélisation des phénomènes de ballonnement.

1.2.3 SlosSat - FLEVO

Une expérience menée conjointement par l'ESA (Agence Spatiale Européenne), et la NLR (National Aerospace Laboratory des Pays-Bas) a été également proposée, avec pour but de dédier un satellite expérimental à l'étude des ballotements dans l'environnement spatial. C'est ainsi qu'est né le satellite SlosSat FLEVO (Facility for Liquid Experimentation and Verification in Orbit). Le projet a été lancé en 1989, et le lancement du satellite a été effectué le 12 février 2005. Le satellite est un cube de 90cm de coté, couvert de panneaux solaires et de 12 tuyères, le tout avec une masse de 129kg au décollage.

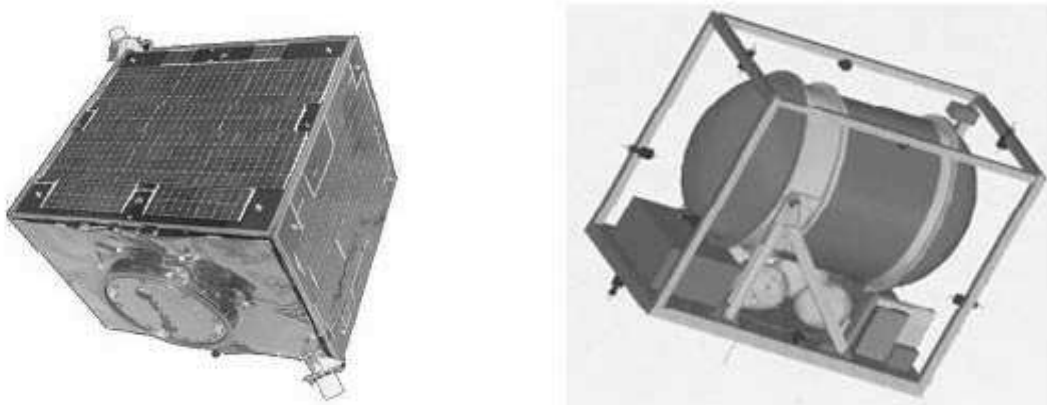


FIGURE 1.9 – Le satellite SlosSat vu de l'extérieur et de l'intérieur avec le réservoir expérimental [7].

L'expérience est constituée d'un réservoir cylindro-Sphérique d'un volume de 87 litres rempli de 33.5 litre d'eau ionisée. Le réservoir comprend 270 capteurs afin de mesurer la répartition de l'eau dans le volume. De plus, 6 accéléromètres et 3 gyroscopes sont présents afin de mesurer les mouvements effectués par le satellite. La figure 1.9 donne un aperçu des vues extérieure et intérieure du satellite.

Un modèle numérique appelé SMS (SlosSat Motion Simulator) [6, 9] a également été réalisé afin de prédire le comportement des fluides lors des manœuvres.

Le satellite lors de sa mission a malheureusement subi des fuites, générant ainsi des moments résiduels et rendant le post-traitement des données délicat. Sur les 8 jours de fonctionnement du satellite, 56h de mission seulement sont disponibles, dont 26h considérées comme exploitables. Des résultats ont donc été publiés pour comparer le code SMS et ces données de vols par Veldman *et al.* dans [9].

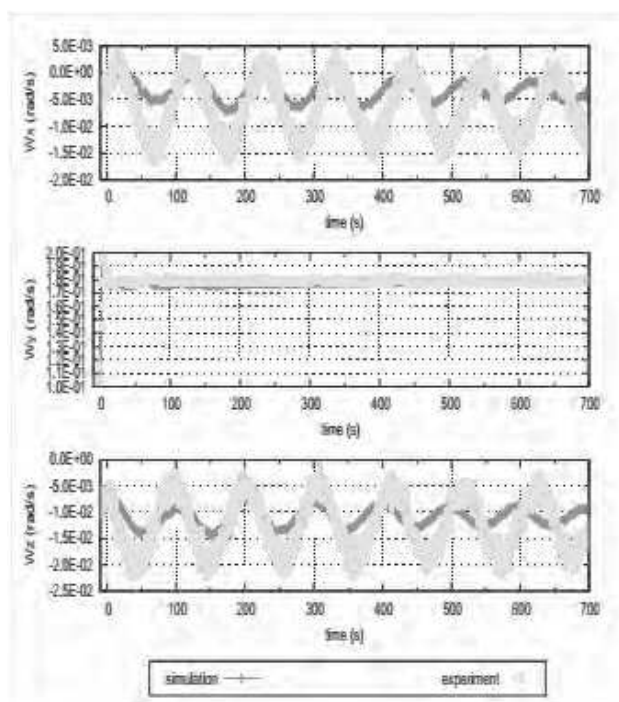


FIGURE 1.10 – Comparaisons entre le code SMS et les résultats expérimentaux de SlosSat [9] .

La figure 1.10 présente un exemple de comparaisons extraites de [9]. Le signal est extrêmement bruité, et la lecture des moments mesurés est délicate. Au cours de cette thèse, j’ai contacté l’ESA et la NLR pour obtenir des informations sur les données, et toutes les mesures effectuées sur la mission m’ont été communiquées. Celles-ci sont malheureusement très denses et le post-traitement n’est pas évident. Malgré les efforts conjoints des services SCAO du CNES et d’ADS, aucune information exploitable dans le cadre de cette thèse n’a pu être effectuée. Les données sont toutefois disponibles et serviront peut être à des travaux ultérieurs.

1.2.4 Expériences dans l’ISS - SPHERES-Slosh

Des expériences sont régulièrement envoyées dans la Station Spatiale Internationale (ISS), et de septembre 2013 à septembre 2015, SPHERES-Slosh, une expérience de la NASA est embarquée à bord de la station pour étudier le ballotement.

Le dispositif expérimental est illustré par le schéma de la figure 1.12. Un réservoir cylindro-sphérique de 12cm de long est placé au centre de l’expérience, dans une boîte sur laquelle sont fixées des LEDs et des caméras pour l’acquisition des données. Trois différents remplissages sont disponibles, avec 20% et 40% de fluide (de l’eau colorée) et un solide de masse équivalente à un remplissage de 40%. Les réservoirs peuvent être changés rapidement à bord de l’ISS. Le tout est

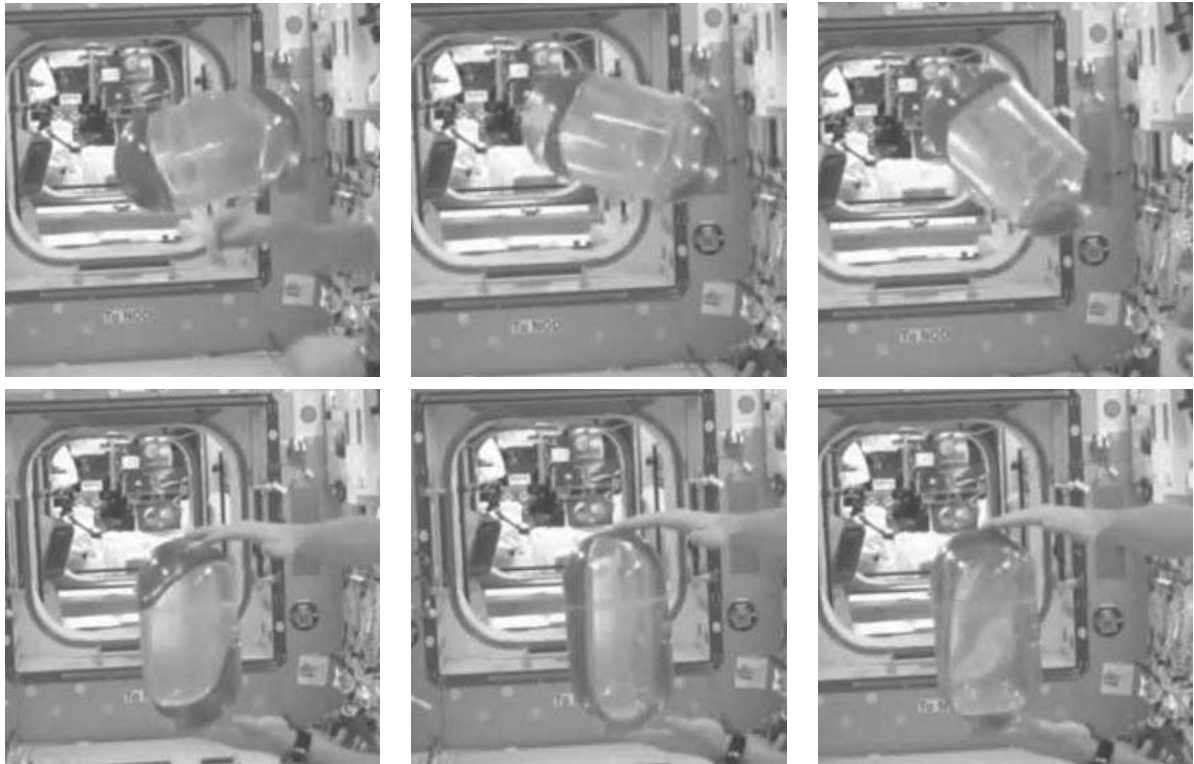


FIGURE 1.11 – Captures d'écran du film proposé par la NASA. Sur la ligne du haut le réservoir est mis en rotation libre par l'astronaute, puis celui-ci bloque le réservoir. Le fluide est alors entrainé par son inertie jusqu'à recouvrir entièrement la paroi sur la ligne du bas [13].

propulsé par le système SPHERES développé par la NASA, qui permet de contrôler les mouvements de l'expérience à l'aide de propulsion de CO₂.

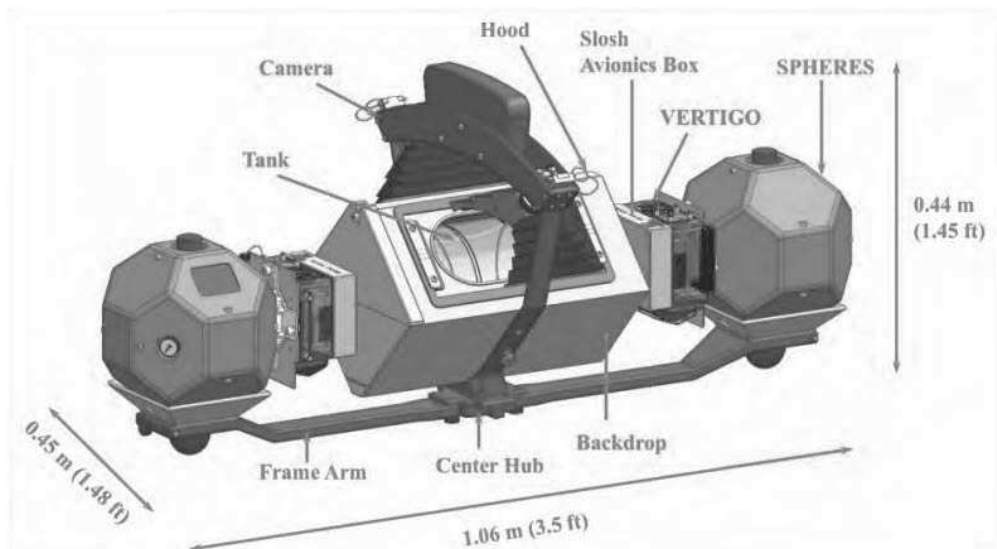


FIGURE 1.12 – Dispositif expérimental de l'expérience SPHERES-Slosh [13].

Les résultats de cette expérience sont classés confidentiels et aucune donnée exploitable n'est

accessible, hormis quelques films ludiques, dont un extrait d'une des séquences vidéo est proposé dans la figure 1.11. Sur ces images, un astronaute donne un moment au réservoir, puis stoppe le mouvement brusquement. Le fluide est entraîné par inertie et se met à recouvrir toute la paroi du réservoir.

Cette expérience a été réalisée pour modéliser des phénomènes à fort nombres de Bond, visant ainsi une application plus orientée vers les lanceurs.

Une autre expérience nommée FLUIDICS, menée conjointement par ADS et le CNES est actuellement en construction, et devrait être embarquée dans l'année 2016. L'objectif de cette expérience est de modéliser des phénomènes pour des faibles nombres de Bond, visant ainsi une application plus orientée vers le comportement des ergols dans les satellites lors des phases de manœuvres pour le maintien à poste. Dans cette expérience un fluide parfaitement mouillant sera embarqué, ce qui se rapproche plus de la réalité puisque les ergols sont majoritairement des fluides parfaitement mouillants. Ces résultats seront d'une grande aide pour valider les écoulements à faible nombre de Bond aussi bien pour les modèles analytiques que pour les modèles numériques. De plus le fluide embarqué dans les réservoirs sera parfaitement mouillant, contrairement à l'expérience de la NASA. Ceci va permettre de vérifier l'apparition d'un angle de contact si celui-ci existe dans certaines configurations.

1.3 Études numériques

Malgré la grande diversité d'expériences actuellement existantes sur la problématique du ballotement en microgravité, peu de données sont accessibles et exploitables pour les ballotements à faible nombre de Bond. La simulation numérique est une ressource appréciable pour ce type de problématiques. En amont de cette thèse, une étude comparative a été menée entre le code commercial FLOW 3D et le code DIVA, développé à l'IMFT.

FLOW 3D est un code numérique permettant de modéliser une grande variété d'écoulements diphasiques. Il résout les équations de Navier-Stokes sur un maillage cartésien en 3D. En présence de plusieurs phases, une méthode dédiée au suivi d'interface est nécessaire. FLOW 3D utilise une méthode VOF (Volume Of Fluid). Un descriptif de cette méthode est donnée dans le chapitre suivant. La version du code utilisée dans cette étude est la version 9.

DIVA est un code de recherche, développé par S. Tanguy et ses doctorants à l'IMFT depuis 2009. C'est un code qui résout également les équations de Navier-Stokes, et qui est capable de modéliser des écoulements diphasiques, ainsi que de prendre en compte des changements de phases tels que l'ébullition et de la vaporisation. Pour modéliser les parois du réservoir, une méthode de frontières immergées est utilisée. La description de la mise en place de cette méthode est l'objet du Chapitre 3.

L'étude menée pour comparer les performances des deux codes porte sur des manœuvres classiquement rencontrées dans le contexte satellite. Un réservoir cylindro-sphérique est rempli à 50%, et soumis au profil d'accélération donné par la figure 1.13.

Les tests sont confidentiels car ceux-ci sont réalisés dans le cadre d'une étude interne qui pourrait révéler des informations sur la capacité de manœuvre et donc sur la performance des satellites. C'est

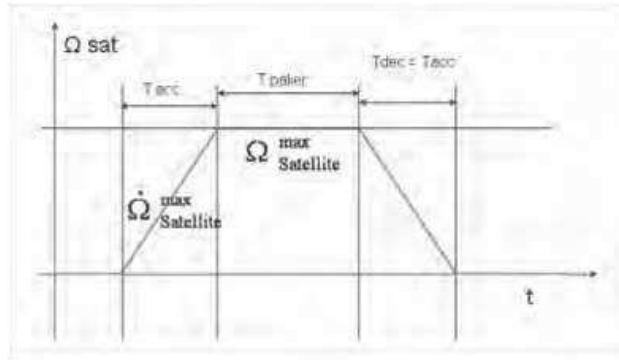


FIGURE 1.13 – Profil de manœuvre utilisé pour le test de comparaison des deux codes [4, 5] .

pourquoi les axes des grandeurs ne sont pas affichés sur les graphiques . L'ordre de grandeur du nombre de Bond sur ce test varie de 0 à 0.35. Les forces et couples mesurés sont donnés dans les figures 1.14 et 1.15. Sur ces images nous pouvons constater que les deux codes donnent des signaux fortement bruités, ce qui indique que les post-traitement des caractéristiques de l'écoulement et de la vitesse sont mal gérés. On peut toutefois remarquer que DIVA semble donner des signaux plus exploitables. C'est en partie pour cette raison que DIVA est choisi pour cette thèse. Le fait que FLOW 3D soit un code commercial ne permet pas d'accéder aux sources pour modifier le code. DIVA étant développé directement dans le laboratoire, les améliorations à apporter sont plus facile à implémenter, et le couplage avec d'autres programmes (comme la boucle SCAO) est également réalisable.

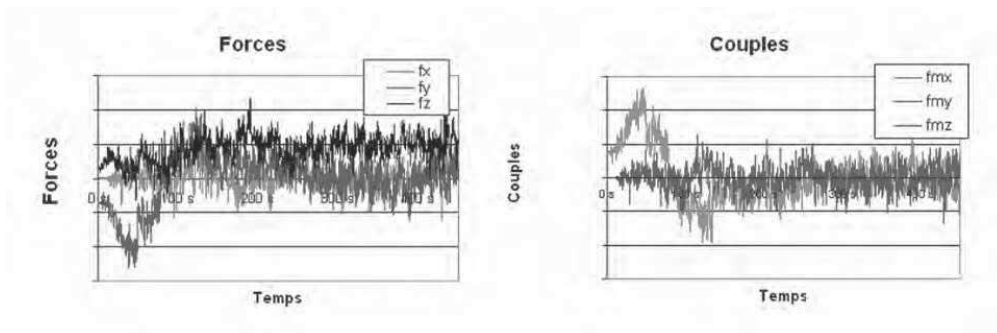


FIGURE 1.14 – Couples et Forces mesurées par la simulation effectuée sur FLOW 3D [4].

De nombreuses améliorations sont toutefois nécessaires afin de garantir que le code prédit bien la dynamique de l'écoulement dans le réservoir pour les faibles nombres de Bond. Ce sont ces améliorations qui ont motivé le suivi de cette étude et que constituent le cœur des travaux réalisés durant cette thèse.

D'autres études numériques sur le ballotement en microgravité ont été menées comme [3] et [8] mais ces études résolvent des écoulements potentiels modélisés par la loi de Bernoulli et négligent par conséquent les effets visqueux. L'amortissement des ballotements ne peut donc pas être modélisé avec ces méthodes.

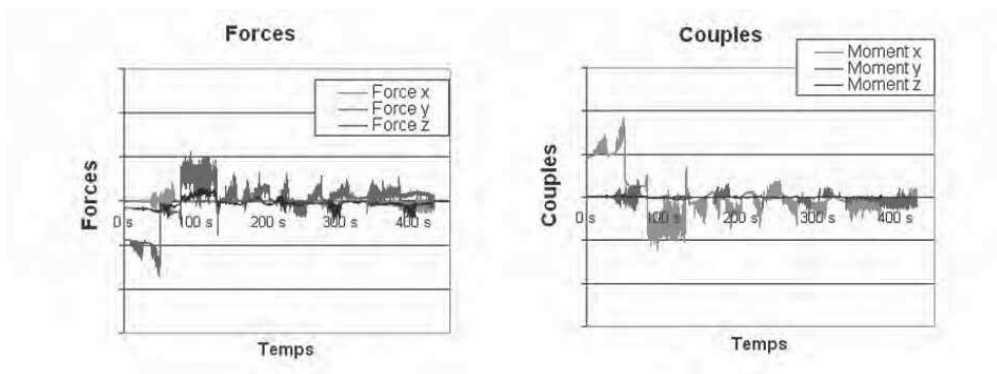


FIGURE 1.15 – Couples et Forces mesurées par la simulation effectuée sur DIVA [4] .

Bibliographie

- [1] N. H. Abramson (1967) *The dynamic behaviour of liquids in moving containers, with applications to space vehicle technology.*
NASA SP 106
- [2] F. T. Dodge (2000) *The new dynamic behaviour of liquids in moving containers*.
Southwest Research Institut, Texas
- [3] G.W. Bao (1994) *Numerical Calculation of Steady Meniscus of Liquid in a Slow Spin Container under a Micro Gravity Field.*
Technische Mechanik **14-2**, 147-154
- [4] B. Busset, H. Bavestrello, L. Dandaleix, C. Figus, C. Berend, D. Theureau (2011) *Ballotements d'ergols en très faible poussée - D1 - Justification et Définition des modèles analytiques.*
Document interne ADS pp 55
- [5] B. Busset, H. Bavestrello, D. Theureau, L. Dandaleix, S. Tanguy, J. Mignot, J. Maureau (2011) *Improvement and validation of low g sloshing modelling for spacecraft applications.*
Space propulsion conference, 2012, 8th may, Bordeaux convention center.
- [6] J. Gerrits, A.E.P. Veldman (2003) *Dynamics of liquid-filled spacecraft*
Journal of Engineering Mathematics**45**, 21-38
- [7] R. Luppès, J.A. Helder, A.E.P. Veldman (2005) *Liquid sloshing in microgravity*
56th Internatinal Astronautical Congress **IAC-05-A2.2.07**
- [8] P. N. Shankar (2003) *A simple method for studying low-gravity sloshing frequencies.*
Proc. R. Soc. Lond. **459**, 3109-3130
- [9] A.E.P. Veldman, J. Gerrits, R. Luppès, J.A. Helder, J.P.B. Vreeburg (2007) *The numerical simulation of liquid sloshing on board spacecraft*
J. of Comp. Phys.textbf224, 82-99
- [10] *Module XII - Techniques et Technologies - La boucle SCAO* (2011)
Cours de technologie spatial - TTVS **Volume 5** , 73-187
- [11] http://www.spaceflight.esa.int/impress/text/education/Glossary/Glossary_D.html
- [12] https://fr.wikipedia.org/wiki/Vol_parabolique
- [13] <https://www.youtube.com/watch?v=fOno4RWAvkg>

Chapitre 2

Méthodes Numériques

Sommaire

2.1	Méthodes numériques de suivi d'interfaces	24
2.1.1	Introduction au méthodes de suivi d'interface	24
2.1.2	Méthode Level-Set	26
2.2	Navier-Stokes - écoulements Diphasiques	28
2.2.1	Modèle hydrodynamique des écoulements diphasiques	28
2.2.2	Méthode de projection pour les équations de Navier-Stokes	29
2.2.3	La méthode Ghost Fluid	30
2.2.4	Traitement du terme visqueux	34
2.2.5	Schémas de discrétisation et restrictions sur le pas de temps	39
2.3	Résolution du champ de Pression	41
2.3.1	Généralités sur les méthodes itératives	41
2.3.2	Le Solveur BlackBox MultiGrid	45
2.4	Modélisation des sauts à l'interface	54
2.4.1	Principe de la méthode Voronoï	54
2.4.2	Tests comparatifs entre le Ghost Fluid et le Voronoï	54
	Solving elliptic problems with discontinuities on irregular domains - the Voronoï Interface Method	82

Ce chapitre traite de la modélisation des écoulements diphasiques. Dans un premier temps, un inventaire de quelques méthodes de suivi d'interfaces est proposé. Puis le modèle physique utilisé est exposé, ainsi que différents schémas de discrétisation spatiale et temporelle utilisés avec une méthode de projection, qui permet de résoudre séparément les champs de vitesses et de pression. Le calcul du champs de pression passe par une résolution d'un système linéaire qui est l'étape la plus couteuse en temps de calcul lors d'une itération temporelle. Différentes méthodes sont comparées pour améliorer les performances du code dans la résolution de ce système linéaire. Enfin un article proposant une nouvelle méthode qui permet d'imposer les conditions de saut au niveau de l'interface avec une précision d'ordre 2 en espace est disponible en annexe de ce chapitre.

2.1 Méthodes numériques de suivi d'interfaces

2.1.1 Introduction aux méthodes de suivi d'interface

Afin de réaliser une simulation d'un écoulement diphasique, il est nécessaire de pouvoir distinguer les différentes phases. La frontière entre ces deux phases est désignée comme l'interface. Il est très important d'être capable de suivre son évolution au cours du temps lors de simulations, bien que cela soit délicat. Plusieurs méthodes existent, classées ici dans deux grandes familles : les méthodes Lagrangiennes et les méthodes Eulériennes.

Les méthodes Lagrangiennes localisent l'interface de façon explicite grâce à des marqueurs qui la suivent, et qui sont transportés par le champ local de vitesse. Les méthodes Eulériennes en revanche transportent un champ scalaire dans tout le domaine, advecté par le champ global de vitesse. La description de l'interface est donc implicite.

2.1.1.1 Méthodes Lagrangiennes

Une des premières méthodes utilisant une formulation Lagrangienne est la méthode MAC (Markers-And-Cells) proposée par Harlow & Welsh (1965) [21]. Des marqueurs sont répartis dans tout le domaine et sont advectés à chaque pas de temps par le champ de vitesse local. La méthode nécessite toutefois un nombre important de marqueurs par rapport au nombre de points dans l'écoulement, et donc un coût de calcul très élevé. Le principe de cette méthode a par la suite été repris par Daly (1968) [12] lors de la modélisation d'instabilités de Rayleigh-Taylor. Celui-ci propose de disposer des marqueurs uniquement sur l'interface. Cette méthode permet donc une résolution inférieure à l'échelle de la maille, le tout à moindre coût. Toutefois la répartition des marqueurs doit rester régulière ou ceux-ci doivent être redistribués, notamment pour la modélisation de problèmes faisant intervenir des ruptures ou reconnections d'interface. Par la suite, Glimm *et al.* (1981) [17] puis Glimm & McBryan (1985) [18] et Chern *et al.* (1986) [10] ont fait évoluer la méthode MAC en méthode de Front Tracking. Le principe de fonctionnement repose sur la base de la méthode MAC en distribuant des marqueurs de façon régulière sur l'interface avec toutefois une connexion entre ces marqueurs afin de gérer les problèmes de rupture et reconnexion d'interface. Ils génèrent ainsi un second maillage de dimension $n - 1$ dans un domaine de dimension n . La forme la plus courante, notamment utilisée par Unverdi & Tryggvason (1992) [54], est une grille triangulaire non-structurée, illustrée par la figure 2.1. Toutefois la méthode ne gère pas automatiquement les ruptures et reconnections d'interface, et sont imposés de manière arbitraire.

Une autre possibilité est de suivre l'évolution de l'interface en adaptant le maillage afin que celui-ci coïncide avec l'interface. Ceci implique un maillage mobile, qui doit être actualisé à chaque pas de temps. Ce sont les méthodes dites de Boundary-Fitted. L'avantage de l'utilisation de telles méthodes est la précision avec laquelle les conditions de saut sont imposées directement à l'interface. Des simulations axisymétriques de l'ascension d'une bulle ont été réalisées par Ryskin & Leal (1984) [43] ou encore par Blanco & Magnaudet (1995) [4] à partir d'une forme fixe. Le maillage utilisé est présenté dans la figure 2.1. Cette méthode présente toutefois des limites lorsque le nombre d'inclusions augmente ou sur des configurations tridimensionnelles impliquant de fortes déformations de l'interface.

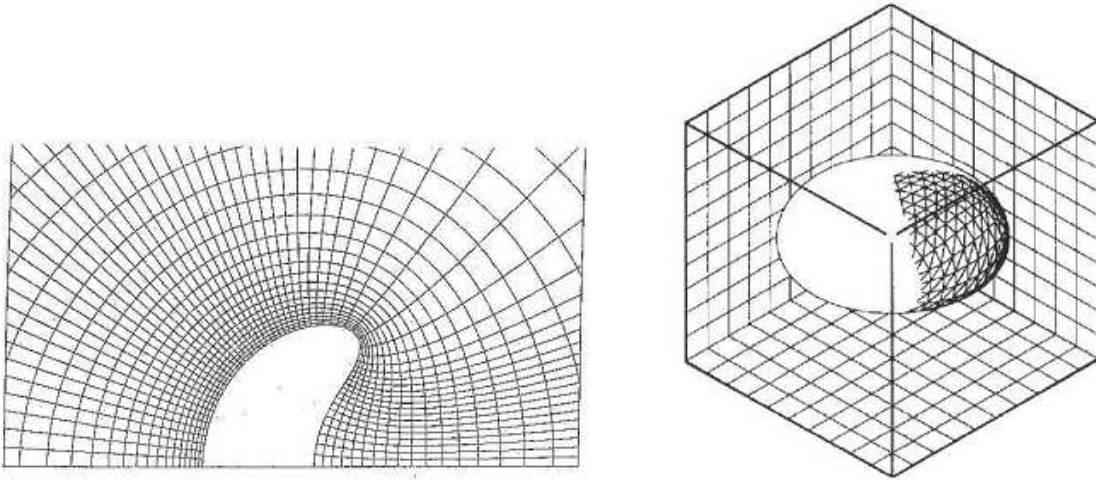


FIGURE 2.1 – A gauche, simulation Boundary Fitted de l’ascension d’une bulle (Blanco & Magnaudet (1995) [4]), à droite : méthode de front tracking avec une grille fixe et des marqueurs connectés sur l’interface (Unverdi & Tryggvason (1992) [54]).

Ceci clôture cette brève introduction non-exhaustive des méthodes Lagrangiennes.

2.1.1.2 Méthodes Eulériennes

Les méthodes Eulériennes sont des méthodes qui transportent des champs scalaires, et décrivent l’interface de manière implicite. La méthode VOF (Volume of Fluid) a été initialement développée par Noh & Woodward (1976) [37] et Hirt & Nichols (1981) [23]. Le champ scalaire défini pour cette méthode est une fonction marqueur C définie dans chaque maille, qui représente la quantité de fluide présente dans celle-ci. Une maille recouverte de fluide donne $C = 1$, alors que $C = 0$ dans une maille sans fluide. Pour les mailles où l’interface est présente, $0 < C < 1$, comme indiqué sur la figure 2.2. A chaque pas de temps, ce scalaire est advecté par la vitesse du fluide grâce à l’équation de transport suivante :

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = 0 \quad (2.1.1)$$

avec \mathbf{u} le champs de vitesse local du fluide. La robustesse de la méthode est dépendante de la précision avec laquelle le scalaire C est advecté. Or le terme $\mathbf{u} \cdot \nabla C$ peut changer brusquement de valeur au niveau de l’interface.

Une fois que la fonction C est transportée, il faut reconstruire l’interface afin de la formuler explicitement. Différentes méthodes ont été développées, avec dans un premier temps la reconstruction SLIC (simple line interface calculation) suggérée par Noh & Woodward (1976) [37], où l’interface est reconstruite avec un segment aligné au maillage dans chaque direction. Cette méthode fut par la suite améliorée par Hirt & Nichols (1981) [23], où un seul segment est reconstruit, parallèle au maillage pour toutes les directions. Bien que cette approche n’ait pas amélioré de façon significative la reconstruction de l’interface, elle a donné le principe à suivre : reconstruire celle-ci à

l'aide non-seulement de la valeur du scalaire C dans la maille, mais aussi en utilisant la valeur des mailles autour. C'est ce raisonnement qui a mené Youngs (1982) [56] à proposer la méthode PLIC (Piecewise Linear Interface Calculation), dans laquelle l'interface est approximée par un segment orienté arbitrairement dans la maille. Les différentes reconstructions sont représentées dans la figure 2.2. L'orientation de ce segment dépend de la normale de l'interface, qui est exprimée de la façon suivante :

$$\mathbf{n} = \frac{\nabla C}{\|\nabla C\|} \quad (2.1.2)$$

Cette normale est donc calculée à l'aide des valeurs de C dans la maille calculée ainsi que dans les mailles adjacentes. La précision de la méthode VOF utilisée dépend de la reconstruction de l'interface car les forces exercées dans le fluide dépendent de la courbure de l'interface.

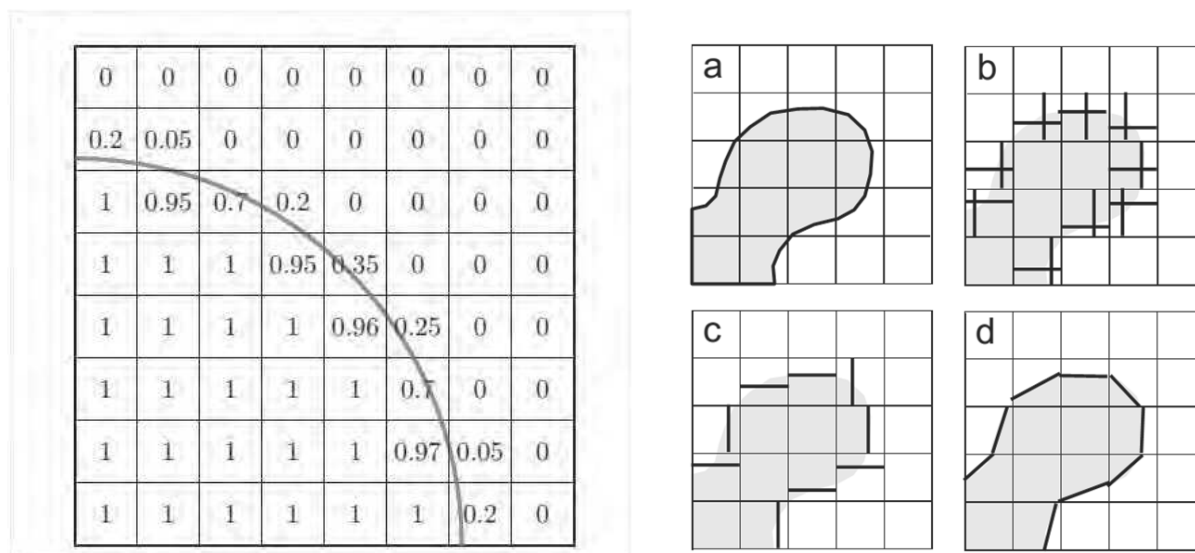


FIGURE 2.2 – A gauche, illustration des valeurs d'un fonction de Couleur C [42]. A droite, illustration des différentes reconstructions, en (a) l'interface physique, en (b) la reconstruction SLIC, en (c) la reconstruction de Hirt & Nichols, en (d) la reconstruction PLIC. Image extraite de [50].

Différentes améliorations pour cette méthode ont été proposées avec entre autres Lafaurie *et al.* [27], Puckett *et al.* [40], Rider & Kothe [41], Gueyffier *et al.* [19], Scardovelli & Zaleski [44], Popinet [39] et Vincent *et al.* [55].

Ceci donne une base non-exhaustive de la littérature disponible sur la méthode VOF. Bien que cette méthode conserve la masse, la reconstruction de l'interface est une étape délicate, qui peut induire des imprécisions sur le calcul des propriétés géométriques de l'interface, et génère donc des courants parasites. Une alternative à la méthode VOF est la méthode Level-Set, qui a été utilisée au cours de cette thèse.

2.1.2 Méthode Level-Set

La méthode Level-Set a été proposée par Osher & Sethian (1988) [38]. C'est une méthode Eulérienne qui utilise une fonction scalaire ϕ définie régulière et continue dans tout le domaine. Elle

doit également répondre à la caractéristique suivante :

$$\begin{cases} \phi < 0 & \text{dans } \Omega^- \\ \phi = 0 & \text{à l'interface} \\ \phi > 0 & \text{dans } \Omega^+ \end{cases} \quad (2.1.3)$$

La fonction distance signée définie positive dans Ω^+ et négative dans Ω^- répond à tous les critères énoncés et sera donc la fonction utilisée dans nos simulations. Son sens physique est par ailleurs intéressant, puisque la valeur absolue de la fonction $\|\phi\|$ correspond à la distance entre le point du maillage et l'interface dans la direction normale de celle-ci.

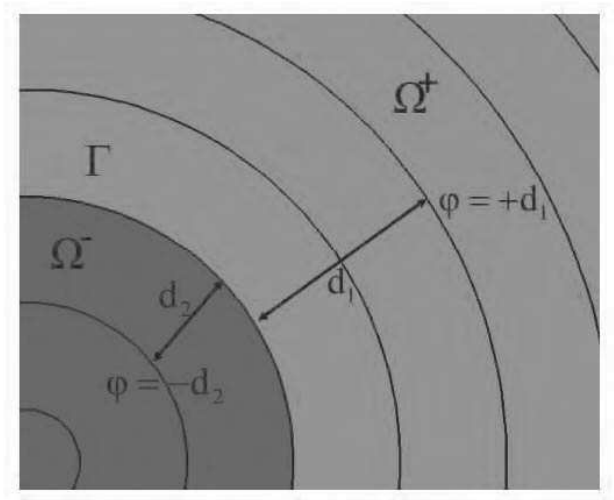


FIGURE 2.3 – Illustration de la fonction level set ϕ .

Tout comme dans la méthode VOF, une étape d'advection est nécessaire par calculer l'évolution temporelle de l'interface. Par définition de la fonction, nous avons à l'interface :

$$\frac{D\phi(\mathbf{x}, t)}{Dt} = 0 \quad (2.1.4)$$

En détaillant la dérivée définie par l'équation 2.1.4, nous obtenons l'équation de transport suivante :

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} + \mathbf{u} \cdot \nabla \phi(\mathbf{x}, t) = 0 \quad (2.1.5)$$

avec \mathbf{u} le champs de vitesse donné par le fluide. Tout comme la méthode VOF, des schémas robustes doivent être employés.

Lorsque le cisaillement à l'interface est important la fonction ϕ peut perdre sa régularité. Il est donc important après l'advection de faire une étape dite de réinitialisation de la fonction redistance, afin de régulariser de nouveau la fonction. Pour cela, Sussman *et al.* (1994) [45] ont proposé l'algorithme suivant :

$$\frac{\partial d}{\partial \tau} = \text{sign}(\phi)[1 - \|\nabla d\|] \quad (2.1.6)$$

L'équation (2.1.6) itère donc sur quelques pas de temps fictif τ pour converger de nouveau vers une fonction régulière d dans tout le domaine avec toutefois la condition initiale donnée dans l'équation (2.1.7) afin d'assurer que la position de l'interface (la ligne de niveau 0) soit conservée.

$$d(\mathbf{x}, t, \tau = 0) = \phi(\mathbf{x}, t) \quad (2.1.7)$$

La fonction $sign(\phi)$ dans l'équation (2.1.6) est une fonction lissée définie comme :

$$sign(\phi) = \begin{cases} -1 & \text{si } \phi < -dx \\ \frac{\phi}{\sqrt{\phi^2 + dx^2}} & \text{si } \|\phi\| \leq dx \\ 1 & \text{si } \phi > dx \end{cases} \quad (2.1.8)$$

En pratique très peu d'itérations suffisent à converger vers une solution stationnaire. L'inconvénient principal de la méthode est toutefois la perte de masse. Celle-ci est fortement liée à l'étape de redistance. La ligne de niveau 0 peut être légèrement déplacée par rapport à sa position initiale. Ceci peut induire un défaut de conservation de la masse d'une des deux phases lorsque le maillage utilisé est trop grossier. Comme la fonction Level-Set est continue dans tout le domaine de calcul, il est possible de calculer avec précision les propriétés géométriques de l'interface en utilisant simplement les relations suivantes :

$$\mathbf{n} = \frac{\nabla\phi}{\|\nabla\phi\|} \quad \kappa(\phi) = -\nabla \cdot \mathbf{n} \quad (2.1.9)$$

Avec κ la courbure locale de l'interface. Les bases de la méthode étant posées, nous allons maintenant nous intéresser à la modélisation d'écoulements diphasiques.

2.2 Equations de Navier-Stokes pour les écoulements diphasiques

Dans cette partie nous aborderons le fonctionnement du code de calcul DIVA pour le traitement d'écoulements diphasiques, tout d'abord d'un point de vue physique, puis nous verrons comment les équations sont discrétisées temporellement. Une attention particulière est portée sur la discrétisation du terme visqueux. Nous verrons également comment sont gérées les discontinuités des champs scalaires à l'interface.

2.2.1 Modèle hydrodynamique des écoulements diphasiques

La modélisation d'écoulements diphasiques incompressibles passe généralement par la résolution des équations de Navier-Stokes incompressibles (4.1.9,4.1.10) ainsi que du bilan des contraintes normales à l'interface (2.2.3) :

$$\nabla \cdot \mathbf{U} = 0 \quad (2.2.1)$$

$$\rho \left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla) \mathbf{U} \right) = -\nabla p + \rho \mathbf{g} + \nabla \cdot (2\mu \mathbf{ID}) \quad (2.2.2)$$

$$[\mathbf{n} \cdot (p\mathbf{I} - 2\mu \mathbf{ID}) \cdot \mathbf{n}]_{\Gamma} = \sigma \kappa \quad (2.2.3)$$

avec \mathbf{U} le vecteur vitesse, p la pression, μ et ρ respectivement la viscosité dynamique et la masse volumique, \mathbf{g} l'accélération de la gravité et \mathbf{ID} le tenseur des taux de déformation tel que $\mathbf{ID} = \frac{1}{2}(\nabla\mathbf{U} + \nabla\mathbf{U}^T)$, \mathbf{n} représente le vecteur normal à l'interface, σ la tension de surface et κ la courbure. $[\]_{\Gamma}$ représente ici l'opérateur de saut à l'interface (Γ).

Dans un écoulement diphasique, les vitesses tangentielles sont toujours continues grâce aux effets visqueux qui assurent le raccordement entre les deux phases. Lorsqu'il n'y a pas de transfert de masse entre les deux phases ni de changement de phase, la vitesse normale à l'interface est également continue. C'est dans ce contexte que se positionne notre étude, ce qui nous permet d'utiliser la relation suivante :

$$[\mathbf{U}]_{\Gamma} = 0 \quad (2.2.4)$$

Enfin, en l'absence de convection Marangoni, les contraintes tangentielles sont continues et s'expriment de la façon suivante :

$$[\mathbf{T} \cdot (2\mu\mathbf{ID}) \cdot \mathbf{T}]_{\Gamma} = 0 \quad (2.2.5)$$

avec \mathbf{T} le vecteur tangent à l'interface. En revanche les contraintes normales sont discontinues de part la définition donnée par l'équation (2.2.3). Celles-ci assurent le bilan des efforts entre la force de tension de surface et les contraintes normales de pression et visqueuses qui existent à l'interface. Il est donc possible de le reformuler comme un saut de pression :

$$[p]_{\Gamma} = \sigma\kappa + [2\mu]_{\Gamma}(\nabla u \cdot \mathbf{n}, \nabla v \cdot \mathbf{n}, \nabla w \cdot \mathbf{n}) \cdot \mathbf{n} \quad (2.2.6)$$

2.2.2 Méthode de projection pour les équations de Navier-Stokes

La résolution numérique des équations de Navier-Stokes est généralement effectuée à l'aide d'une méthode de projection. Le principe de la méthode repose sur le fait que tout champ vectoriel peut être décomposé en une somme d'un champ vectoriel à divergence non-nulle et d'un champ de vecteur dérivant d'un potentiel, ce qui permet de résoudre de façon découplée la vitesse et le champ de pression. Pour illustrer le fonctionnement de la méthode, la viscosité est ici résolue de façon explicite, et sera détaillée plus amplement dans la section sur ce sujet. Les deux inconnues à résoudre dans notre cas sont \mathbf{U}^{n+1} et p^{n+1} :

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + (\mathbf{U}^n \cdot \nabla) \mathbf{U}^n = -\frac{\nabla p^{n+1}}{\rho} + \frac{\nabla \cdot (2\mu\mathbf{ID}^n)}{\rho} + \mathbf{g} \quad (2.2.7)$$

$$\nabla \cdot \mathbf{U}^{n+1} = 0 \quad (2.2.8)$$

Il est donc possible de définir une vitesse intermédiaire \mathbf{U}^* à divergence non-nulle, que l'on peut calculer explicitement :

$$\mathbf{U}^* = \mathbf{U}^n - \Delta t \left((\mathbf{U}^n \cdot \nabla) \mathbf{U}^n - \frac{\nabla \cdot (2\mu\mathbf{ID}^n)}{\rho} - \mathbf{g} \right) \quad (2.2.9)$$

Cette vitesse intermédiaire \mathbf{U}^* peut également s'exprimer d'après les équations (2.2.7) et (2.2.9) comme :

$$\mathbf{U}^* = \mathbf{U}^{n+1} + \Delta t \frac{\nabla p^{n+1}}{\rho} \quad (2.2.10)$$

En utilisant la formulation de \mathbf{U}^* donnée par (2.2.10) et la conservation de la masse (2.2.8), nous obtenons une équation de Poisson sur la pression :

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho} \right) = \frac{\nabla \cdot \mathbf{U}^*}{\Delta t} \quad (2.2.11)$$

Ceci donne donc la démarche à suivre : dans un premier temps la vitesse intermédiaire est calculée explicitement avec les termes au temps t^n , puis le gradient du champ de pression est résolu au temps t^{n+1} . Enfin la vitesse est calculée à son tour au temps t^{n+1} en effectuant une correction du champ \mathbf{U}^* avec la pression. La vitesse \mathbf{U}^{n+1} respecte donc à nouveau la conservation de la masse et de la quantité de mouvement.

2.2.3 La méthode Ghost Fluid

La méthode Ghost Fluid a été développée dans un premier temps par Fedkiw *et al.* [16] afin d'imposer des conditions de saut au niveau d'une interface mobile lorsque l'on résout un système d'équations aux dérivées partielles. En effet les schémas de discrétisation utilisés classiquement sont basés sur des conditions de dérivabilité des variables. Afin d'éviter un épaissement fictif de l'interface en lissant les singularités, la méthode Ghost Fluid permet de prendre en compte les sauts directement dans les schémas de dérivation, à condition de connaître avec précision la position de l'interface. C'est une méthode qu'il est intéressant de coupler avec la méthode Level-Set.

Le principe de la méthode est assez simple. Comme la position de l'interface est connue, il suffit lorsque les dérivées rencontrent l'interface de projeter des points fantômes afin de prolonger par continuité la fonction avant de la dériver (comme l'illustre la figure 2.4). Les extensions des valeurs réelles sont réalisées grâce à la connaissance de la valeur des sauts à l'interface. La prolongation fantôme de la fonction permet une résolution robuste de la dérivée dans les mailles proches de l'interface. Un exemple est donné en une dimension, la méthode peut s'extrapoler en deux et trois dimensions.

Considérons l'application de cette méthode à la résolution d'une équation de Poisson comme cela a été proposé dans [29] :

$$\begin{cases} \nabla \cdot (\beta \nabla u) = f \\ [u]_{\Gamma} = a(x_{\Gamma}) \\ \left[\beta \frac{du}{dx} \right]_{\Gamma} = b(x_{\Gamma}) \\ [\beta]_{\Gamma} \neq 0 \end{cases} \quad (2.2.12)$$

La discrétisation classique du problème (2.2.12) en suivant un schéma centré d'ordre 2 donne :

$$\frac{\beta_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{\Delta x} - \beta_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x}}{\Delta x} = f_i \quad (2.2.13)$$

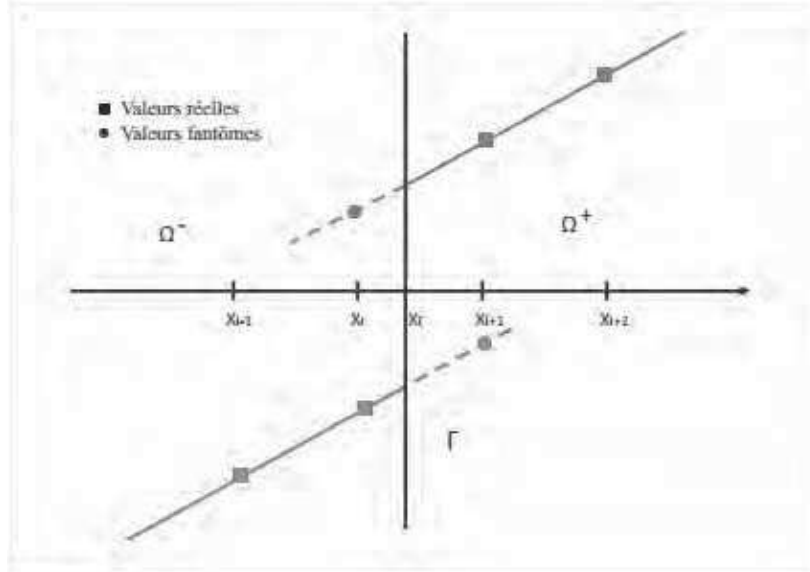


FIGURE 2.4 – Illustration de la méthode Ghost Fluid, avec les points fantômes prolongés par continuité.

Pour plus de lisibilité nous allons traiter trois différents cas, sachant que les trois cas se cumulent lorsqu'un saut supplémentaire est à prendre en compte.

2.2.3.1 Discontinuité sur la variable $[u]_{\Gamma} \neq 0$

Dans le premier cas un saut sur la variable doit être imposé. Cette variable n'est donc pas continue alors que le flux et le coefficient de diffusion le sont :

$$\begin{cases} \nabla \cdot (\beta \nabla u) = f \\ [u]_{\Gamma} = a(x_{\Gamma}) \\ \left[\beta \frac{du}{dx} \right]_{\Gamma} = 0 \\ [\beta]_{\Gamma} = 0 \end{cases} \quad (2.2.14)$$

Regardons le cas où l'interface passe entre les points x_{i+1} et x_i comme illustré sur la figure 2.4. Nous rappelons la discrétisation avec l'indicateur de la phase au-dessus de la variable :

$$\frac{\beta_{i+\frac{1}{2}} \frac{u_{i+1}^+ - u_i^-}{\Delta x} - \beta_{i-\frac{1}{2}} \frac{u_i^- - u_{i-1}^-}{\Delta x}}{\Delta x} = f_i \quad (2.2.15)$$

La dérivée entre x_{i+1} et x_i n'est donc pas continue puisque le point x_{i+1} est du côté + et le point x_i est du côté -. Il est toutefois possible de définir un point fantôme u_i^+ à l'aide de la définition de l'opérateur de saut :

$$a(x_{\Gamma}) = u_i^+ - u_i^- \text{ ou encore } u_i^- = u_i^+ - a(x_{\Gamma}) \quad (2.2.16)$$

En injectant (2.2.16) dans l'équation (2.2.15), nous obtenons :

$$\frac{\beta_{i+\frac{1}{2}} \frac{u_{i+1}^+ - u_i^+ + a(x_\Gamma)}{\Delta x} - \beta_{i-\frac{1}{2}} \frac{u_i^- - u_{i-1}^-}{\Delta x}}{\Delta x} = f_i \quad (2.2.17)$$

qui se réécrit plus classiquement :

$$\frac{\beta_{i+\frac{1}{2}} \frac{u_{i+1}^+ - u_i^+}{\Delta x} - \beta_{i-\frac{1}{2}} \frac{u_i^- - u_{i-1}^-}{\Delta x}}{\Delta x} = f_i - \frac{a(x_\Gamma)}{\Delta x^2} \quad (2.2.18)$$

Ainsi la dérivée est calculée de façon continue de chaque côté de l'interface.

2.2.3.2 Discontinuité sur le flux $[\beta \frac{du}{dx}]_\Gamma \neq 0$

Dans le deuxième cas c'est sur le flux que l'on souhaite imposer un saut, alors que la variable et le coefficient de diffusion sont continus :

$$\begin{cases} \nabla \cdot (\beta \nabla u) = f \\ [u]_\Gamma = 0 \\ \left[\beta \frac{du}{dx} \right]_\Gamma = b(x_\Gamma) \\ [\beta]_\Gamma = 0 \end{cases} \quad (2.2.19)$$

Le raisonnement est très similaire au premier cas. Regardons le cas où l'interface passe entre les points x_{i+1} et x_i comme illustré sur la figure 2.4. Nous rappelons la discrétisation avec l'indicateur de la phase au-dessus du flux :

$$\frac{\left(\beta_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{\Delta x} \right)^+ - \left(\beta_{i-\frac{1}{2}} \frac{u_i^- - u_{i-1}^-}{\Delta x} \right)^-}{\Delta x} = f_i \quad (2.2.20)$$

La dérivée entre $x_{i+\frac{1}{2}}$ et $x_{i-\frac{1}{2}}$ n'est donc pas continue puisque le point x_{i+1} est du côté + et le point x_i est du côté -. Il est toutefois possible de définir une valeur fantôme du flux à l'aide de la définition de l'opérateur saut :

$$b(x_\Gamma) = \beta \frac{du}{dx} \Big|_{i+\frac{1}{2}}^+ - \beta \frac{du}{dx} \Big|_{i+\frac{1}{2}}^- \quad \text{ou encore} \quad \beta \frac{du}{dx} \Big|_{i+\frac{1}{2}}^+ = \beta \frac{du}{dx} \Big|_{i+\frac{1}{2}}^- + b(x_\Gamma) \quad (2.2.21)$$

En injectant (2.2.21) dans l'équation (2.2.20), nous obtenons :

$$\frac{\left(\beta_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{\Delta x} \right)^- + b(x_\Gamma) - \left(\beta_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x} \right)^-}{\Delta x} = f_i \quad (2.2.22)$$

qui se réécrit plus classiquement :

$$\frac{\left(\beta_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{\Delta x} \right)^- - \left(\beta_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x} \right)^-}{\Delta x} = f_i - \frac{b(x_\Gamma)}{\Delta x} \quad (2.2.23)$$

2.2.3.3 Discontinuité du coefficient de diffusion $\beta \neq 0$

Dans le troisième cas seul sur le coefficient de diffusion est discontinu :

$$\begin{cases} \nabla \cdot (\beta \nabla u) = f \\ [u]_{\Gamma} = 0 \\ \left[\beta \frac{du}{dx} \right]_{\Gamma} = 0 \\ [\beta]_{\Gamma} \neq 0 \end{cases} \quad (2.2.24)$$

Regardons le cas où l'interface passe entre les points x_{i+1} et x_i comme illustré sur la figure 2.4. Rappelons la discrétisation du problème :

$$\frac{\beta_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{\Delta x} - \beta_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x}}{\Delta x} = f_i \quad (2.2.25)$$

Dans ce cas-là, il est nécessaire de définir correctement le coefficient $\beta_{i+\frac{1}{2}}$. Pour cela, on s'intéresse à la définition du flux. Dans le cas où le flux est continu à l'interface (il pourrait ne pas l'être sans perte de généralités) :

$$\left[\beta \frac{du}{dx} \right]_{\Gamma} = 0 = \beta^+ \frac{u_{i+1} - u_{\Gamma}}{(1-\theta)\Delta x} - \beta^- \frac{u_{\Gamma} - u_i}{\theta\Delta x} \quad (2.2.26)$$

avec θ défini comme :

$$\theta = \frac{\phi_i}{\phi_i + \phi_{i+1}} \quad (2.2.27)$$

L'équation (2.2.26) nous permet donc d'estimer la valeur de u_{Γ} :

$$u_{\Gamma} = \frac{\theta\beta^+u_{i+1} - (1-\theta)\beta^-u_i}{\beta^+\theta + (1-\theta)\beta^-} \quad (2.2.28)$$

Enfin en remplaçant u_{Γ} dans (2.2.26) nous obtenons $\beta_{i+\frac{1}{2}}$:

$$\beta_{i+\frac{1}{2}} = \frac{\beta^+\beta^-}{\beta^+\theta + (1-\theta)\beta^-} \quad (2.2.29)$$

Nous venons donc de détailler quelques-uns des nombreux cas possibles en 1 dimension de l'application de la méthode Ghost Fluid. Les effets se cumulent lorsque plusieurs discontinuités sont présentes. L'extension en 2D et 3D est assez similaire et les détails peuvent être retrouvés dans [25, 47].

Nous pouvons également remarquer qu'un saut sur la variable ou sur le flux va engendrer une modification du second membre uniquement, alors qu'un saut sur le coefficient de diffusion engendre une modification des coefficients au sein-même de la matrice. Nous reviendrons sur cet aspect plus en détail lors de la section portant sur la modélisation des sauts à l'interface.

2.2.4 Traitement du terme visqueux

Le traitement du terme visqueux est une étape importante de la résolution des écoulements diphasiques et qui doit être analysée avec précision. Il est possible d'effectuer le découpage suivant :

$$\nabla \cdot (2\mu\mathbf{ID}) = \mu\nabla \cdot (2\mathbf{ID}) + 2\mathbf{ID} \cdot \nabla\mu \quad (2.2.30)$$

Si la viscosité est constante des deux côtés de l'interface, alors en utilisant la définition de la dérivée de la fonction de Heaviside, on peut montrer que :

$$\nabla\mu = [\mu]_{\Gamma}\mathbf{n}\delta_{\Gamma} \quad (2.2.31)$$

Ce qui permet d'exprimer l'équation (2.2.30) de la façon suivante :

$$\nabla \cdot (2\mu\mathbf{ID}) = \mu\nabla \cdot (2\mathbf{ID}) + 2[\mu]_{\Gamma}\mathbf{ID} \cdot \mathbf{n}\delta_{\Gamma} \quad (2.2.32)$$

Cette formulation fait apparaître une singularité dans la résolution du terme visqueux. Diverses méthodes ont été mises en place pour résoudre le terme visqueux, et nous allons donc en décrire quelques-unes.

2.2.4.1 La méthode Continuous Surface Force (CSF)

La première approche décrite ici est une formulation continue des équations de Navier-Stokes, où les sauts ne sont pas pris en compte de façon discontinue, et où l'interface est épaissie. Les variables telles que la viscosité dynamique ou la masse volumique sont discontinues au niveau de l'interface, et peuvent donc être exprimées de la façon suivante :

$$\begin{aligned} \rho &= \rho_l + H(\phi)(\rho_g - \rho_l) \\ \mu &= \mu_l + H(\phi)(\mu_g - \mu_l) \end{aligned} \quad (2.2.33)$$

avec $H(\phi)$ la distribution de Heaviside, égale à 1 dans le premier fluide et à 0 dans le second. Regardons maintenant l'écriture des équations de Navier-Stokes pour cette méthode. Les efforts capillaires ne sont pas considérés comme des sauts au niveau de l'interface mais comme une force incorporée dans l'équation de Navier-Stokes [44, 45] :

$$\nabla \cdot \mathbf{U} = 0 \quad (2.2.34)$$

$$\rho \left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla) \mathbf{U} \right) = -\nabla p + \rho \mathbf{g} + \nabla \cdot (2\mu\mathbf{ID}) + \sigma \kappa \mathbf{n} \delta_{\Gamma} \quad (2.2.35)$$

avec δ_{Γ} la distribution de Dirac. Comme mentionné au début de la section, la méthode ne traite pas des sauts comme des discontinuités, et pour pallier à cela, il faut définir de nouvelles fonctions de Dirac δ_e et Heaviside $H_e(\phi)$ qui sont continues, ce qui se traduit numériquement par un épaississement de l'interface. La discrétisation temporelle des équations de Navier-Stokes avec cette méthode donne la méthode de projection suivante :

$$\mathbf{U}_1^* = \mathbf{U}^n - \Delta t \left((\mathbf{U}^n \cdot \nabla) \mathbf{U}^n - \frac{\nabla \cdot (2\mu\mathbf{ID}^n)}{\rho^{n+1}} - \frac{\sigma \kappa \mathbf{n} \delta_e}{\rho^{n+1}} - \mathbf{g} \right) \quad (2.2.36)$$

La pression est donc déterminée à l'aide de la vitesse intermédiaire calculée précédemment :

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{U}_1^*}{\Delta t} \quad (2.2.37)$$

Afin de calculer un champ de vitesse à divergence nulle, l'étape de correction est ensuite effectuée :

$$\mathbf{U}^{n+1} = \mathbf{U}_1^* - \Delta t \frac{\nabla p^{n+1}}{\rho^{n+1}} \quad (2.2.38)$$

Cette méthode fonctionne dans beaucoup de situations mais elle peut être affectée par le développement de courants parasites pour des efforts capillaires importants. D'autres méthodes ont été développées par la suite en prenant en compte les sauts comme des discontinuités. Ce sont ces méthodes-là que nous allons maintenant décrire.

2.2.4.2 La méthode Ghost Fluid Primitive viscous Method (GFPM)

Cette méthode a été proposée par Kang *et al.* [25]. Les auteurs du papier présentent une approche différente, en explicitant le saut inclus dans le terme visqueux (cf équations (2.2.30),(2.2.32)). Le tenseur \mathbf{ID} , qui est composé de la somme du gradient du champ de vitesse et de son transposé peut aussi s'exprimer de la façon suivante :

$$\mathbf{n}^t \cdot \mathbf{ID} \cdot \mathbf{n} = \mathbf{n}^t \cdot \nabla (\mathbf{U} \cdot \mathbf{n}) = \frac{\partial U_n}{\partial n} \quad (2.2.39)$$

Regardons maintenant la méthode de projection adaptée à la GFPM. La vitesse intermédiaire \mathbf{U}_2^* est estimée de la façon suivante :

$$\mathbf{U}_2^* = \mathbf{U}^n - \Delta t \left((\mathbf{U}^n \cdot \nabla) \mathbf{U}^n - \frac{\mu \Delta \mathbf{U}^n}{\rho^{n+1}} - \mathbf{g} \right) \quad (2.2.40)$$

Soulignons le fait que cette formulation paraît similaire à une formulation pour un écoulement monophasique, et que les discontinuités n'apparaissent pas. Il faut donc les prendre en compte lors de la résolution du champ de pression :

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{U}_2^*}{\Delta t} + \nabla \cdot \left(\frac{(\sigma \kappa + 2[\mu]_\Gamma \frac{\partial U_n}{\partial n}) \mathbf{n} \delta_\Gamma}{\rho^{n+1}} \right) \quad (2.2.41)$$

La vitesse au temps t^{n+1} respectant la condition d'incompressibilité s'écrit donc en fonction de la pression mais aussi des termes sources singuliers liés à la présence de l'interface :

$$\mathbf{U}^{n+1} = \mathbf{U}_2^* - \frac{\Delta t}{\rho^{n+1}} \left(\nabla p^{n+1} - \left(\sigma \kappa + 2[\mu]_\Gamma \frac{\partial U_n}{\partial n} \right) \mathbf{n} \delta_\Gamma \right) \quad (2.2.42)$$

2.2.4.3 La méthode Ghost Fluid Conservative viscous Method (GFCM)

Dans cette méthode développée par Sussmann *et al.* dans [46], le terme capillaire est considéré comme une condition de saut dans la résolution de la pression à l'interface. La projection de cette méthode donne une nouvelle vitesse intermédiaire :

$$\mathbf{U}_3^* = \mathbf{U}^n - \Delta t \left((\mathbf{U}^n \cdot \nabla) \mathbf{U}^n - \frac{\nabla \cdot (2\mu \mathbf{ID}^n)}{\rho^{n+1}} - \mathbf{g} \right) \quad (2.2.43)$$

La pression est ensuite calculée en prenant en compte uniquement le saut à l'interface. Le terme $\nabla \cdot (2\mu \mathbf{ID}^n)$ est déterminé à l'aide de la méthode Ghost Fluid afin de prendre en compte la discontinuité évoquée dans l'équation (2.2.32).

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{U}_3^*}{\Delta t} + \nabla \cdot \left(\frac{\sigma \kappa \mathbf{n} \delta_\Gamma}{\rho^{n+1}} \right) \quad (2.2.44)$$

La vitesse au temps t^{n+1} s'écrit donc :

$$\mathbf{U}^{n+1} = \mathbf{U}_3^* - \frac{\Delta t}{\rho^{n+1}} (\nabla p^{n+1} - \sigma \kappa \mathbf{n} \delta_\Gamma) \quad (2.2.45)$$

Cette méthode est généralement utilisée avec un schéma explicite pour résoudre le terme visqueux, car les composantes sont liées les unes aux autres. Par exemple la résolution du champ selon \mathbf{u} fait intervenir \mathbf{v} et \mathbf{w} . Les trois composantes doivent donc toutes être résolues simultanément, rendant l'implémentation de cette méthode délicate si l'on utilise un schéma temporel implicite. La méthode a toutefois été implémentée dans DIVA et fait l'objet du paragraphe suivant.

2.2.4.4 La méthode Ghost Fluid Conservative viscous Method Implicit (GFCMI)

La formulation de la GFCMI ressemble beaucoup à celle de la GFCM, avec toutefois une différence pour le calcul de la vitesse intermédiaire. Il est nécessaire de résoudre un système linéaire, car la formulation implicite fait intervenir une équation de Poisson sur \mathbf{U}_4^* :

$$\mathbf{U}_4^* - \Delta t \frac{\nabla \cdot (2\mu \mathbf{ID}^*)}{\rho^{n+1}} = \mathbf{U}^n - \Delta t ((\mathbf{U}^n \cdot \nabla) \mathbf{U}^n - \mathbf{g}) \quad (2.2.46)$$

Le terme $\nabla \cdot (2\mu \mathbf{ID}^*)$ est également exprimé à l'aide de la méthode Ghost Fluid pour calculer les interpolations de la viscosité sur les bords des cellules dans l'équation (2.2.32). Les étapes d'après sont les mêmes que pour la GFCM avec la résolution du champ de pression :

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{U}_4^*}{\Delta t} + \nabla \cdot \left(\frac{\sigma \kappa \mathbf{n} \delta_\Gamma}{\rho^{n+1}} \right) \quad (2.2.47)$$

Ainsi que l'étape de correction :

$$\mathbf{U}^{n+1} = \mathbf{U}_4^* - \frac{\Delta t}{\rho^{n+1}} (\nabla p^{n+1} - \sigma \kappa \mathbf{n} \delta_\Gamma) \quad (2.2.48)$$

2.2.4.5 La méthode Ghost Fluid Semi-Conservative viscous Method Implicit (GFSCMI)

La méthode GFSCMI a été développée dans le cadre de la modélisation de changements de phases, avec des contraintes de stabilité temporelle non-négligeables sur le pas de temps visqueux [28, 46]. Cette méthode est donc issue de la nécessité d'impliciter de façon plus simple le terme visqueux et de tolérer les champs de vitesses discontinus à l'interface. Le terme visqueux est formulé de la façon suivante :

$$\begin{aligned}
\nabla \cdot (2\mu \mathbf{ID}) &= \nabla \cdot (\mu \nabla \mathbf{U} + \mu \nabla^t \mathbf{U}) \\
&= \nabla \cdot (\mu \nabla \mathbf{U}) + \nabla \cdot (\mu \nabla^t \mathbf{U}) \\
&= \nabla \cdot (\mu \nabla \mathbf{U}) + \nabla^t \mathbf{U} \cdot \nabla \mu + \mu \nabla \cdot (\nabla^t \mathbf{U}) \\
&= \nabla \cdot (\mu \nabla \mathbf{U}) + [\mu]_{\Gamma} \nabla^t \mathbf{U} \cdot \mathbf{n} \delta_{\Gamma}
\end{aligned} \tag{2.2.49}$$

avec $\nabla \cdot (\nabla^t \mathbf{U}) = 0$ dans le cadre d'écoulements incompressibles. Ceci permet donc de reformuler le terme visqueux en une partie continue et un saut, sachant que la partie continue est facilement implicite puisque les composantes de vitesses ne sont pas liées les unes aux autres dans cette formulation. La vitesse intermédiaire est définie de la façon suivante :

$$\mathbf{U}_5^* - \Delta t \frac{\nabla \cdot (\mu \nabla \mathbf{U}_5^*)}{\rho^{n+1}} = \mathbf{U}^n - \Delta t ((\mathbf{U}^n \cdot \nabla) \mathbf{U}^n - \mathbf{g}) \tag{2.2.50}$$

Puis le champ de pression est résolu :

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{U}_5^*}{\Delta t} + \nabla \cdot \left(\frac{(\sigma \kappa + [\mu]_{\Gamma} \frac{\partial U_n^n}{\partial n}) \mathbf{n} \delta_{\Gamma}}{\rho^{n+1}} \right) \tag{2.2.51}$$

Et enfin l'étape de correction :

$$\mathbf{U}^{n+1} = \mathbf{U}_5^* - \frac{\Delta t}{\rho^{n+1}} \left(\nabla p^{n+1} - \left(\sigma \kappa + [\mu]_{\Gamma} \frac{\partial U_n^n}{\partial n} \right) \mathbf{n} \delta_{\Gamma} \right) \tag{2.2.52}$$

Ces méthodes sont testées pour un cas de validation diphasique pour comparer leurs précisions respectives.

2.2.4.6 Validation et convergence des différentes méthodes

Un cas test permettant de comparer la stabilité et la convergence des différentes méthodes est proposé dans cette partie afin d'avoir un premier aperçu de leurs performances. Une bulle d'air sphérique dans de l'eau est soumise à la gravité sans vitesse initiale. C'est un cas test adapté à la comparaison des formulations sur le terme visqueux puisqu'il n'y a pas de trainée de pression générée dans cet écoulement. Le cas est effectué en 2D axi symétrique à plusieurs nombres de Reynolds, ainsi qu'à deux faibles nombres de Bond afin de conserver la forme sphérique de la bulle. Les caractéristiques des différentes simulations sont résumées dans le tableau 2.1.

	$Re_{\infty} = 20$	$Re_{\infty} = 40$	$Re_{\infty} = 60$	$Re_{\infty} = 80$	$Re_{\infty} = 100$
$R_{bulle} [\mu m]$	19.55	44.65	70.45	98.95	127.9
$g_z [m \cdot s^{-2}]$	-9144.4	-1755.2	-705.6	-357.5	-213.9
$\sigma [N \cdot m^{-1}]$	0.56	0.56	0.56	1.12	1.12
$t_f [s]$	0.0004	0.00135	0.002375	0.0048	0.007

TABLE 2.1 – Rayon de la bulle, accélération de la gravité, coefficient de tension de surface et temps final de la simulation de la bulle pour 5 nombres de Reynolds.

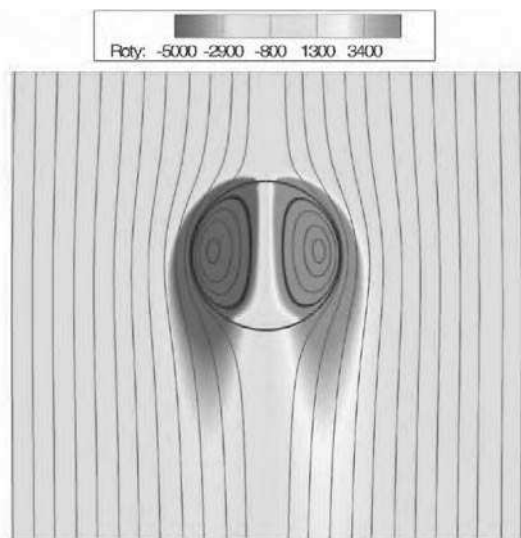


FIGURE 2.5 – Les lignes de courant et le champ de vorticité dans le référentiel mobile de la bulle en ascension pour $Re = 60$.

Le domaine est choisi de façon à ne pas influencer sur la remontée de la bulle, avec $L_r = 8R_{bulle}$ et $L_z = 4L_r$. Les propriétés physiques du gaz et du liquide sont celles de l'air et de l'eau : $\rho_{liq} = 1000 \text{ kg}\cdot\text{m}^{-3}$, $\mu_{liq} = 0.00113 \text{ kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$, $\rho_{gaz} = 1.226 \text{ kg}\cdot\text{m}^{-3}$, $\mu_{gaz} = 1.78\cdot 10^{-5} \text{ kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$. Les conditions des simulations sont présentées dans le tableau 2.1 pour 5 configurations correspondant à $Re = 20, 40, 60$ pour un nombre de Bond $Bo = 0.025$, et à $Re = 80, 100$ pour $Bo = 0.0125$. La figure 2.5 montre le résultat de la simulation d'une bulle d'air pour le cas où le Reynolds final d'ascension est $Re = 60$.

Les simulations sont effectuées avec trois maillages : 32×128 , 64×256 , 128×512 , ce qui correspond respectivement à 8, 16 ou 32 mailles par diamètre.

Maillage	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	- 22.4%	- 32.2%	- 37.2%	- 75.53%	- 43.35%
64×256	+ 2.44%	+ 3.12%	+ 1.07%	- 1.78%	- 1.30%
128×512	+ 2.38%	+ 2.38 %	+ 0.37 %	+ 0.8 %	+ 0.19%

TABLE 2.2 – Erreur sur la valeur du nombre de Reynolds terminal pour des bulles en ascension avec la méthode implicite GFCMI.

Maillage	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	- 22.0%	- 34.0%	- 38.8%	- 63.0%	- 67.4%
64×256	+ 2.5%	+ 1.8%	- 1.3%	- 1.3%	- 3.0%
128×512	+ 0.5%	+ 0.8 %	- 1.0 %	+ 1.9 %	+ 1.9%

TABLE 2.3 – Erreur sur la valeur du nombre de Reynolds terminal pour des bulles en ascension avec la méthode explicite GFCM.

Maillage	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	- 23.5%	- 32.5%	- 9.0%	- 62.5%	- 67.0%
64×256	- 0.5%	+ 2.5%	+ 1.3%	+ 4.3%	+ 3.0%
128×512	- 3.5%	+ 0.6 %	+ 0.7 %	+ 5.4 %	+ 6%

TABLE 2.4 – Erreur sur la valeur du nombre de Reynolds terminal pour des bulles en ascension avec la méthode explicite GFSCM.

Maillage	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	- 18.6%	- 25.0%	- 30.2%	- 52.9%	- 55.1%
64×256	- 1.2%	+ 1.5%	- 0.1%	+ 2.0%	+ 0.9%
128×512	- 3.7%	+ 0.1 %	- 0.2 %	+ 3.2 %	+ 3.9%

TABLE 2.5 – Erreur sur la valeur du nombre de Reynolds terminal pour des bulles en ascension avec la méthode implicite GFSCMI.

Les résultats obtenus avec le GFCMI sont présentés dans le tableau 2.2. De plus, nous montrons aussi les résultats obtenus dans [28] avec la méthode GFCM avec un traitement explicite du terme visqueux, la méthode GFSCM avec les traitements explicite et implicite (GFSCMI) du terme visqueux (Tableau 2.3, 2.4 et 2.5 respectivement). De cette manière une comparaison entre les différentes méthodes pourra être faite.

Dans un premier temps, nous remarquons que le maillage 32×128 n'est pas assez précis pour capturer les effets visqueux de manière précise, la couche limite visqueuse n'étant que très peu résolue. Pour des maillages intermédiaires, les résultats sont corrects, avec une erreur majoritairement inférieure à 3%.

Les méthodes GFCM et GFCMI donnent des résultats convergents. Le maillage plus raffiné amène en effet une diminution sur l'erreur calculée, avec une erreur inférieure à 2% sur tous les cas. Les méthodes GFSCM et GFSCMI en revanche donnent des résultats moins satisfaisants. Ils sont plus éloignés de la solution, et ne convergent pas lorsque le nombre de Reynolds est important. Ceci donne un aperçu de la performance des méthodes sur un cas test diphasique sans frontières immergées. Les deux méthodes implicites seront implémentées dans la partie suivante, afin de comparer leurs performances en présence de domaines irréguliers.

2.2.5 Schémas de discrétisation et restrictions sur le pas de temps

Dans cette partie, quelques informations sont données sur les schémas utilisés, ainsi que les contraintes temporelles liées aux schémas afin de garantir la stabilité du code. Tous les maillages utilisés au cours de cette thèse sont cartésiens. Nous utilisons également le principe du maillage décalé pour la résolution des vitesses. Les fonctions scalaires (telles que la pression et la fonction la Level Set) sont calculées au centre de la maille, alors que les vitesses sont calculées sur les côtés de la maille dans la direction associée. La formulation du code est en volumes finis, avec une discrétisation spatiale d'ordre 2 pour les termes diffusifs et un schéma WENO Z [6] pour les termes advectifs, afin de garantir la stabilité de la solution. Les dérivées temporelles sont calculées par un

schéma Runge-Kutta d'ordre 2, sauf lorsque les simulations prennent en compte un angle de contact. Un schéma Runge-Kutta d'ordre 3 est alors utilisé. Celui-ci est nécessaire afin de garantir la stabilité dans les simulations diphasiques avec des domaines irréguliers. Dans les simulations effectuées, la pression est calculée implicitement, les termes visqueux dépendent de la méthode utilisée, et les efforts capillaires sont calculés explicitement. Aussi des contraintes temporelles sont à respecter pour garantir la stabilité du code.

1. Le terme lié à la convection est contraint par la restriction suivante :

$$\Delta t_{conv} \leq \frac{K_1}{\frac{Max(|u_{i,j,k}|)}{min(\Delta x_i)} + \frac{Max(|v_{i,j,k}|)}{min(\Delta y_j)} + \frac{Max(|w_{i,j,k}|)}{min(\Delta z_k)}} = \frac{1}{C_{cfl}} \quad (2.2.53)$$

C'est également la condition qui est imposée pour la convection de l'interface dans l'équation de transport de la fonction Level-Set ϕ

2. Le terme lié à la viscosité est contraint par la restriction suivante :

$$\Delta t_{visc} \leq \frac{K_2}{Max\left(\frac{\mu^-}{\rho^-}, \frac{\mu^+}{\rho^+}\right) \frac{2}{min(\Delta x_i^2)} + \frac{2}{min(\Delta y_j^2)} + \frac{2}{min(\Delta z_k^2)}} = \frac{1}{V_{cfl}} \quad (2.2.54)$$

3. Pour les écoulements diphasiques, le terme lié à la tension de surface induit la restriction suivante :

$$\Delta t_{tens} \leq \frac{K_4}{4} \sqrt{\frac{Max(\rho^-, \rho^+) min(\Delta x_i, \Delta y_j, \Delta z_k)^3}{\sigma}} = \frac{1}{S_{cfl}} \quad (2.2.55)$$

Les coefficients K_1, K_2, K_3 et K_4 sont choisis entre 0 et 1. Finalement le pas de temps final est contraint par la relation suivante :

$$\frac{1}{\Delta t} \geq \frac{1}{\Delta t_{conv}} + \frac{1}{\Delta t_{visc}} + \frac{1}{\Delta t_{tens}} \quad (2.2.56)$$

Ces contraintes numériques peuvent être approchées d'un point de vue physique. Par exemple le pas de temps restreignant la convection signifie qu'une particule de fluide ne doit pas franchir plus d'une maille au cours d'un pas de temps. Une remarque sur le pas de temps visqueux est toutefois à faire, concernant la formulation GFSCMI. En effet la résolution du terme visqueux à proprement parler est implicite, mais le terme de saut calculé qui intervient dans la résolution du champ de pression est estimé au temps t^n , ce qui induit un pas de temps relié implicitement au terme visqueux. Il faut donc penser à le prendre en compte, bien que le schéma de résolution du terme visqueux soit semi-implicite :

$$\Delta t_{visc} = \frac{\bar{\rho} \Delta x^2}{[\mu]} \quad (2.2.57)$$

où $\bar{\rho}$ est la densité moyenne des deux phases. Cette contrainte est cependant beaucoup moins restrictive que celle utilisée avec un algorithme explicite (eq. (2.2.54)) pour beaucoup des systèmes liquide-gaz, tel que l'eau et l'air par exemple [28]. La résolution des systèmes linéaires sur la pression était effectuée à l'aide d'un algorithme de Gradient Conjugué, mais un solveur multigrille a été implémenté au cours de cette thèse, et plus de détails sont fournis dans la section suivante.

2.3 Résolution du champ de Pression

La phase de résolution de l'équation de Poisson pour la pression fait apparaître un système linéaire de la forme $A\mathbf{u} = \mathbf{b}$. L'inversion de ce système est la partie la plus coûteuse en temps de calcul du code : entre 80% et 99% d'une itération temporelle y est consacré suivant le nombre d'éléments du système linéaire. Il faut donc nécessairement passer par l'amélioration de cette étape si l'on souhaite minimiser le coût d'une itération temporelle. Ce système linéaire peut être résolu de façon directe, ou encore de façon itérative. La matrice A est symétrique, définie positive et creuse. Prenons l'exemple d'une discrétisation de l'opérateur de pression pour un maillage régulier en trois dimensions ayant $n = 100$ points dans chaque direction : la matrice résultante serait donc de dimension $N = 10^6$. L'inversion de ce système avec des méthodes directes comme une décomposition de Cholesky demanderait environ 10^{14} opérations. Ceci serait définitivement trop coûteux, en prenant en compte le fait qu'il y a deux résolutions du champs de pression dans une itération temporelle dans le cadre de l'utilisation d'un schéma RK2, et trois résolutions avec un schéma RK3. C'est pour cette raison que les méthodes itératives sont privilégiées par rapport aux méthodes directes.

2.3.1 Généralités sur les méthodes itératives

Les méthodes itératives sont donc une alternative aux méthodes directes, avec toutefois une différence notable : une méthode itérative converge vers la solution jusqu'à une précision donnée par l'utilisateur, alors qu'une méthode directe donne la solution à l'erreur machine près. Il existe une multitude de méthodes itératives, mais certaines sont privilégiées dans le cadre de la modélisation numérique d'écoulements diphasiques, comme la méthode du gradient conjugué préconditionné ou encore les méthodes multigrilles.

2.3.1.1 Les méthodes de Gradient Conjugué Préconditionné

La méthode du gradient conjugué est une méthode qui converge en un nombre fini d'itérations au plus égal à la dimension du système linéaire. L'idée est de minimiser l'erreur en générant des vecteurs résidus du sous-espace de Krylov de manière récursive en se basant sur l'approximation de la solution. Un lecteur curieux pourra trouver plus d'informations dans [22].

L'utilisation de la méthode est intéressante si elle aboutit rapidement vers une bonne estimation de la solution. Cette vitesse de convergence est dépendante du conditionnement de la matrice A :

$$\|\mathbf{x}^{(L)} - \mathbf{x}\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^L \|\mathbf{x}^0 - \mathbf{x}\|_A \quad (2.3.1)$$

avec \mathbf{x} la solution exacte du problème, $\mathbf{x}^{(L)}$ l'approximation de la solution \mathbf{x} à la L^{ieme} itération du gradient conjugué, \mathbf{x}^0 la valeur initiale et $\kappa(A)$ le conditionnement de la matrice A . On rappelle que le conditionnement d'une matrice est défini comme :

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|. \quad (2.3.2)$$

avec A une matrice inversible, et avec $\kappa(A) \geq 1$. Plus $\kappa(A)$ est grand devant 1, plus le système sera mal conditionné. L'équation 2.3.1 montre la forte relation entre la vitesse de convergence du

Gradient Conjugué et le conditionnement de la matrice : pour une tolérance donnée, le nombre d'itérations va augmenter avec le conditionnement de A . Pour les écoulements multiphasiques, le conditionnement est affecté par les variations de masse volumique entre les différentes phases. Par exemple, entre de l'eau et de l'air, le rapport des masses volumiques est de $\epsilon = 10^{-3}$, ce qui veut donc dire que $\kappa(A) \propto \frac{1}{\epsilon}$ et que $\kappa(A)$ ne dépend plus seulement de la taille du domaine mais aussi de la physique du problème.

C'est par ailleurs pour cette raison que la méthode du Gradient Conjugué est rarement utilisée sous cette forme et qu'un préconditionneur y est appliqué. Le principe de fonctionnement d'un Gradient Conjugué Préconditionné est d'utiliser la matrice $M^{-1}A$ à la place de la matrice A , pour une matrice M symétrique définie positive telle que $M^{-1}\mathbf{r}$ soit aisément calculé pour tout résidu \mathbf{r} . Parmi les préconditionneurs classiques se trouvent les méthodes de Jacobi, ou encore de Cholesky Incomplet.

2.3.1.2 Les méthodes Multigrilles

La famille des méthodes Multigrilles est une alternative aux méthodes Gradient Conjugué. Le principe de base des méthodes multigrilles repose sur une résolution multi-échelle : résoudre le problème à une échelle grossière permet de réduire l'erreur rapidement alors que la résolution à un niveau plus fin permet de mieux décrire les détails de cette solution. Cette méthodologie donne plusieurs possibilités concernant la façon de résoudre le système notamment sur la façon d'établir le problème à résoudre sur l'échelle grossière ainsi que le passage d'une échelle à l'autre.

Plusieurs grandes familles se distinguent, en se basant sur la façon dont sont définies les matrices de *prolongation* et de *restriction*, permettant de passer respectivement de la grille grossière à la grille fine et inversement :

- Les méthodes Multigrilles Géométriques reposent sur un découpage basé sur la discrétisation du domaine et les coefficients sont calculés à l'aide des propriétés géométriques des grilles fines et grossières. Ces méthodes-là sont parfaitement adaptées aux problèmes sur maillages réguliers, avec toutefois une forte dépendance sur le conditionnement de la matrice A . Elles ne sont par conséquent pas adaptées à la modélisation d'écoulements diphasiques.
- Les méthodes Multigrilles Algébriques en revanche reposent uniquement sur la matrice A pour définir les différentes grilles et les coefficients de prolongation et de restriction. Ces méthodes-là sont parfaitement adaptées pour résoudre des systèmes linéaires générés sur des maillages non-structurés. Elles sont toutefois difficiles à mettre en place, et délicates à paralléliser.
- La méthode du Black Box Multigrid (BBMG) fait partie de la famille des méthodes algébriques car les différents coefficients sont basés sur la matrice A , tout en reposant sur les propriétés géométriques du domaine pour définir les différentes grilles. Cette méthode sera vue plus en détail dans la suite de ce chapitre.

Toutes ces familles de méthodes Multigrilles peuvent également être employées comme des préconditionneurs, comme le proposent [8, 26, 31, 49].

2.3.1.3 Comparaison des performances des différents Solveurs

MacLachkan, Tang & Vuik [31] ont mené en 2009 une étude ayant pour objectif de tester la robustesse et l'efficacité de différents solveurs linéaires, incluant notamment les méthodes proposées précédemment. Afin d'effectuer les tests de performance, des études sur des cas diphasiques stationnaires et instationnaires sont présentées, incluant différents paramètres physiques afin d'éprouver les solveurs.

Cette partie s'intéresse principalement à la comparaison du cas test diphasique stationnaire de [31], et une étude instationnaire sera proposée dans la suite de ce chapitre. Dans cette étude, MacLachkan, Tang & Vuik [31] utilisent une méthode de projection similaire à celle présentée dans la section 2.2.2 en trois étapes. Ils négligent néanmoins les termes de tension de surface afin de faciliter la modélisation, ce qui implique que les comparaisons des différents solveurs ne testent que les sauts sur les coefficients de la matrice générés par les différentes masses volumiques, mais le terme de saut sur la variable n'est pas pris en compte, et par conséquent ne modifie pas le second membre du système \mathbf{b} (dans le cadre d'une méthode Ghost Fluid).

Le cas test est un écoulement 3D dans un domaine cubique de côté $L = 1 \text{ cm}^3$ rempli d'un fluide de masse volumique $\rho = 1$ comprenant m bulles sphériques d'un second fluide de masse volumique $\rho = \epsilon$, de rayon s et rangées de façon 'structurée'. Les simulations sont réalisées pour $N = 32^3, 64^3$ et 128^3 , N étant le nombre de degrés de liberté du système.

Les différents paramètres variant entre les différentes simulations sont donc N le nombre de points de calcul, s le rayon des bulles, m le nombre de bulles et enfin ϵ le rapport des masses volumiques entre les deux fluides. Les critères de comparaison entre les différents solveurs sont le nombre d'itérations ou de cycles IT , le temps CPU passé dans le solveur de pression (incluant l'initialisation du solveur ainsi que la résolution), ainsi que la précision RES , mesurée comme la norme relative du résidu : $\|\mathbf{b} - A\mathbf{u}^{(L)}\|_2 / \|\mathbf{b} - A\mathbf{u}^{(0)}\|_2$ où $\mathbf{u}^{(L)}$ représente la L^{me} itération du solveur, avec une initialisation de la solution à zéro ($\mathbf{u}^{(0)} = 0$). Le critère de convergence imposé par les auteurs est tel que $\|\mathbf{r}^{(L)}\|_2 / \|\mathbf{r}^{(0)}\|_2 < \delta = 10^{-8}$.

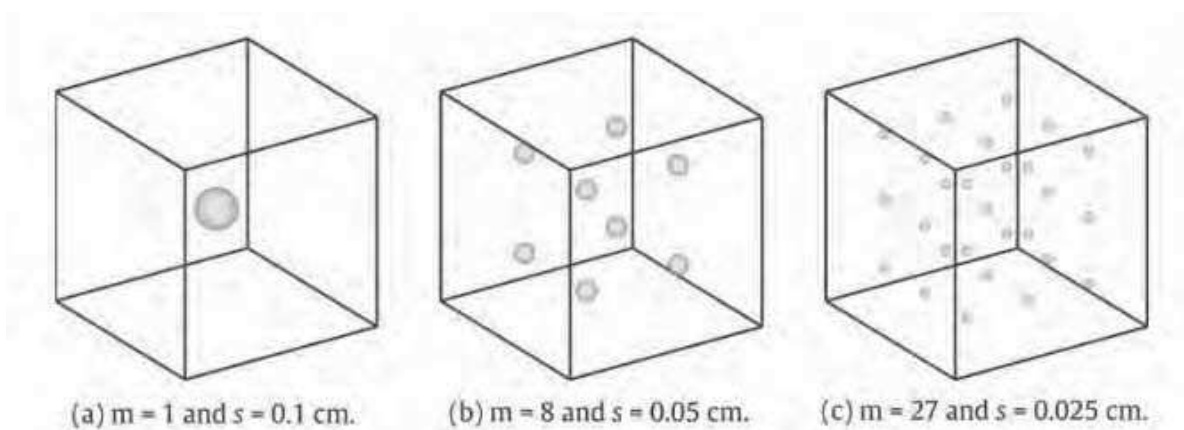


FIGURE 2.6 – Illustration des différents cas proposés dans [31].

Les différentes configurations sont illustrées dans la figure (2.6). Les parties suivantes proposent les résultats obtenus dans [31], en variant respectivement le nombre de points, le rapport des densi-

tés, le rayon et le nombre de bulles. Les méthodes proposées sont les suivantes : le Gradient Conjugué Préconditionné Cholesky Incomplet (IC-CG), avec un MultiGrille Géométrique (GMG-CG), ou encore le BlackBox Multigrid (BBMG-CG). Les méthodes Multigrilles sont aussi proposées en solveur indépendant avec l'Algébrique Multigrid (AMG) ou encore le BlackBox Multigrid (BBMG). D'autres méthodes sont également présentées dans [31] mais leurs performances n'indiquent pas d'intérêt pour la comparaison souhaitée dans le cadre de cette thèse.

2.3.1.3.a Influence du nombre de points

Le tableau 2.6 présente les résultats sur la convergence des méthodes en variant le nombre de points. Un grand nombre de point engendre des coefficients de matrice mal conditionnés. Des différentes méthodes proposées, seulement le BBMG-CG et le BBMG ont une évolution de temps de calcul linéaire en fonction du nombre de point : en doublant le nombre de points dans chaque direction, le temps CPU de ces deux méthodes est également augmenté d'un facteur 8, alors que les autres méthodes nécessitent plus de temps pour converger. Remarquons également que l'AMG demande plus de temps alors que le nombre d'itérations n'augmente pas fortement, ceci étant probablement dû au coût supplémentaire associé à l'utilisation d'un maillage non-structuré. Une discussion plus détaillée est proposée dans [31].

Nombre de points	$N = 32^3$			$N = 64^3$			$N = 128^3$		
	Méthode	CPU	IT	RES	CPU	IT	RES	CPU	IT
IC-CG	0.3	112	0.9E-8	5.8	244	0.1E-8	92.3	444	0.9E-8
GMG-CG	0.2	81	0.9E-8	3.9	132	0.9E-8	36	134	0.8E-8
BBMG-CG	0.1	12	0.2E-8	0.8	12	0.2E-8	7.0	12	0.3E-8
BBMG	0.1	16	0.8E-8	0.8	17	0.3E-8	7.4	17	0.4E-8
AMG	0.9	20	0.1E-7	13.0	40	0.1E-7	129.2	45	0.8E-8

TABLE 2.6 – Résultats sur la convergence des simulations de [31] avec $m = 2^3$, $s = 0.05cm$, $\epsilon = 10^{-3}$ en variant le nombre de points dans le domaine N .

2.3.1.3.b Influence du rapport des densités ϵ

Le tableau 2.7 présente les résultats sur la convergence des méthodes en faisant varier le rapport des densités des deux fluides. Un petit ϵ engendre des coefficients de matrice mal conditionnés (rappelons que $\epsilon = \rho_g/\rho_l$). Le BBMG-CG et le BBMG sont une nouvelle fois les seules méthodes qui convergent indépendamment de la valeur de ϵ , tant en temps CPU qu'en nombre d'itérations. Le GMG-CG montre toutefois la sensibilité de la méthode Multigrille Géométrique à la présence de forts rapports de densité en passant du simple au double en nombre d'itérations ainsi qu'un facteur 10 sur le CPU lorsque ϵ diminue d'un facteur 100. L'IC-CG montre également une forte dépendance au rapport entre les densités.

Rapport des densités	$\epsilon = 10^{-1}$			$\epsilon = 10^{-3}$			$\epsilon = 10^{-5}$		
Méthode	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
IC-CG	3.1	131	0.1E-7	5.8	244	0.1E-8	6.9	289	0.5E-8
GMG-CG	1.0	33	0.1E-7	3.9	132	0.9E-8	8.0	267	0.8E-8
BBMG-CG	0.8	12	0.1E-8	0.8	12	0.2E-8	0.8	12	0.2E-8
BBMG	0.8	17	0.3E-8	0.8	17	0.3E-8	0.8	17	0.3E-8
AMG	8.9	15	0.8E-8	13.0	40	0.1E-7	21.5	92	0.9E-8

TABLE 2.7 – Résultats sur la convergence des simulations de [31] avec $N = 64^3$, $m = 2^3$, $s = 0.05cm$ en variant le rapport des densités des les deux fluides ϵ .

2.3.1.3.c Autres résultats et conclusions

Dans [31], les auteurs présentent également l’influence du nombre de bulles ainsi que l’influence du rayon de celles-ci. Une nouvelle fois, ce sont le BBMG-CG et le BBMG qui ont les meilleures performance en temps CPU, et qui ne sont pas affectés par les modifications sur la matrice dues à la physique du problème.

La méthode BlackBox MultiGrid s’illustre comme un Solveur (aussi bien qu’un préconditionneur) particulièrement adapté à la résolution de l’équation de Poisson portant sur la pression. Ce sont ces études qui ont motivé le développement de cette méthode dans le code DIVA. La section suivante présente plus en détail le fonctionnement de cette méthode, ainsi que des tests sur les performances pour un cas instationnaire entre un IC-CG et un BBMG implémentés dans DIVA.

2.3.2 Le Solveur BlackBox MultiGrid

En 1977, Brandt propose dans son papier [7] l’idée d’une résolution multi-échelle sur différentes grilles pour des problèmes aux conditions aux limites. Il y propose quelques notions de base sur lesquelles repose le BBMG, avec notamment l’idée de passer d’une grille à l’autre à l’aide de coefficients d’interpolation pondérés. Il faudra néanmoins attendre le papier d’Alcouffe [2], où le travail de Brandt est adapté pour résoudre des problèmes de diffusion avec des coefficients fortement discontinus. Un nouveau raisonnement est présenté sur le choix des coefficients d’interpolation, modifiant le stencil de l’opérateur (des diagonales supplémentaires sont ajoutées dans la matrice). Dendy propose son code BBMG, ainsi qu’un mode d’emploi dans [13, 14, 15] dans lequel il résume succinctement les méthodes utilisées et donne quelques détails concernant leurs implémentations. Durant sa thèse [3] sous la direction de Dendy, V. Bandy développe de façon détaillée la manière dont sont établis les coefficients d’interpolation, et présente une étude comparative sur les différentes méthodes de relaxation. Depuis, d’autres publications ont eu lieu sur le sujet, mais dépassent le cadre de notre étude (adaptation du solveur à la modélisation d’un écoulement en milieu poreux par exemple [35]).

2.3.2.1 Fonctionnement général des méthodes Multigrilles

Afin de pouvoir établir le fonctionnement de la méthode BBMG, il est important de définir les notations et de rappeler la philosophie des méthodes multigrilles. Soit la grille fine notée G^M définie par des mailles de taille $h_M = \Delta x$, et une succession de grilles G^1, \dots, G^M correspondant aux maillages $h_1 > \dots > h_M$. Le maillage est supposé carré et structuré par soucis de simplification. Enfin le rapport géométrique entre deux grilles successives est tel que $h_{k+1}/h_k = \frac{1}{2}$. D'autres ratios sont possibles mais ce choix reste le plus judicieux comme indiqué dans [7]. Le système à résoudre est de la forme :

$$A\mathbf{u} = \mathbf{b} \quad (2.3.3)$$

En réécrivant le système (2.3.3) avec les nouvelles notations multigrilles pour la grille fine, nous obtenons :

$$A^M \mathbf{U}^M = \mathbf{b}^M \quad (2.3.4)$$

avec \mathbf{U}^M la solution exacte du problème. Notons \mathbf{u}^M une approximation de la solution exacte et $\mathbf{r}^M = \mathbf{b}^M - A^M \mathbf{u}^M$ le résidu représentant l'écart de cette approximation avec la solution exacte. Dans [7], il a été montré que les méthodes itératives telles que les méthodes Gauss-Seidel ou SOR (Successive Over-Relaxation) convergent rapidement lors des premières itérations, c'est-à-dire tant que l'erreur $\mathbf{V}^M = \mathbf{U}^M - \mathbf{u}^M$ suit de grandes fluctuations, mais ralentissent de façon conséquente lorsque l'erreur devient homogène. Lorsque la convergence ralentit, il est possible d'approximer l'erreur \mathbf{V}^M sur une grille plus grossière G^{M-1} telle que :

$$A^{M-1} \mathbf{v}^{M-1} = \mathbf{b}^{M-1} \equiv I_M^{M-1} (\mathbf{b}^M - A^M \mathbf{u}^M) \quad (2.3.5)$$

où \mathbf{v}^{M-1} représente l'approximation de l'erreur \mathbf{V}^M sur la grille grossière, $\mathbf{v}^M = \mathbf{u}^M$ sur la grille la plus fine, et I_M^{M-1} l'opérateur d'interpolation permettant de passer de la grille fine G^M à la grille grossière G^{M-1} . Une fois le système (2.3.5) suffisamment résolu, il faut corriger la solution sur la grille fine : $\mathbf{v}^M \leftarrow \mathbf{v}^M + I_{M-1}^M \mathbf{v}^{M-1}$, avec I_{M-1}^M l'opérateur d'interpolation allant de G^{M-1} à G^M . Si la convergence du système (2.3.5) ralentit après quelques itérations, il est possible de transférer l'erreur \mathbf{V}^{M-1} sur la grille G^{M-2} , et ainsi de suite. Sur la grille la plus grossière G^1 , le système peut être résolu de façon directe ou de façon itérative jusqu'à une solution suffisamment convergée.

Les méthodes Multigrilles reposent toutes sur cette philosophie. Quelques éléments restent à définir, notamment les coefficients d'interpolation ainsi que le choix de la méthode de relaxation.

2.3.2.2 Définition du BBMG

Les coefficients d'interpolation de la méthode BBMG reposent sur la continuité de l'opérateur initial. Prenons un exemple 2D d'une équation de Poisson avec un coefficient de diffusion :

$$\nabla \cdot (D \nabla \mathbf{u}) = \mathbf{b} \quad (2.3.6)$$

La discrétisation avec un schéma centré classique donne :

$$D_{i+1/2,j} \frac{(u_{i+1,j} - u_{i,j})}{\Delta x^2} - D_{i-1/2,j} \frac{(u_{i,j} - u_{i-1,j})}{\Delta x^2} + D_{i,j+1/2} \frac{(u_{i,j+1} - u_{i,j})}{\Delta y^2} - D_{i,j-1/2} \frac{(u_{i,j} - u_{i,j-1})}{\Delta y^2} = b_{i,j} \quad (2.3.7)$$

que l'on peut réécrire :

$$B_{i+1,j}(u_{i+1,j} - u_{i,j}) - B_{i,j}(u_{i,j} - u_{i-1,j}) + A_{i+1,j}(u_{i,j+1} - u_{i,j}) - A_{i,j}(u_{i,j} - u_{i,j-1}) = b_{i,j} \quad (2.3.8)$$

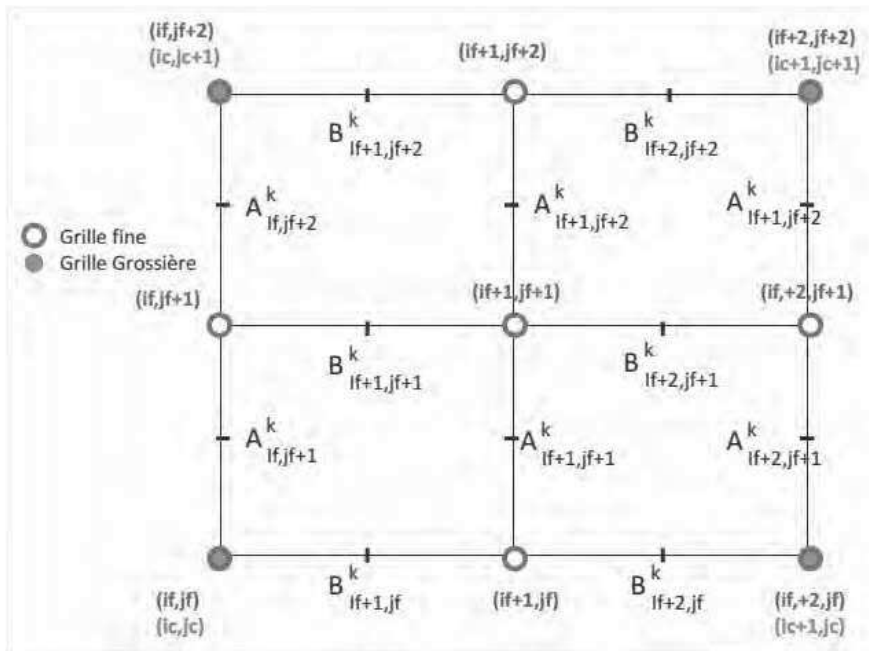


FIGURE 2.7 – Représentation d'une grille grossière ($k-1$) et d'une grille fine (k).

Cette discrétisation peut être simplement extrapolée en 3D. La méthode BBMG repose sur la création de coefficients d'interpolation permettant de conserver la continuité des opérateurs $D\mathbf{u}_x$ et $D\mathbf{u}_y$, sur tous les points où il n'intervient pas de discontinuité. La figure 2.7 permet d'illustrer le raisonnement pour deux grilles k et $k-1$. Soit $\mathbf{v}^k = I_{k-1}^k \mathbf{u}^{k-1}$. Pour les points appartenant à la fois à la grille grossière (que l'on notera k_c) et à la grille fine (que l'on notera k_f), il est logique de considérer $v_{if,jf}^k = u_{ic,jc}^{k-1}$. Pour les points alignés selon une direction, la continuité de l'opérateur $D\mathbf{u}_x$ peut être approximée par $B_{if+2,jf}^k (v_{if+2,jf}^k - v_{if+1,jf}^k) = B_{if+1,jf}^k (v_{if+1,jf}^k - v_{if,jf}^k)$. Avec l'égalité établie pour les points communs aux deux grilles :

$$v_{if+1,jf}^k = \frac{B_{if+1,jf}^k u_{ic,jc}^{k-1} + B_{if+2,jf}^k u_{ic+1,jc}^{k-1}}{B_{if+1,jf}^k + B_{if+2,jf}^k} \quad (2.3.9)$$

Le même raisonnement permet de calculer les points alignés verticalement. Pour les points de la grille fine centrés dans les mailles de la grille grossière on obtient :

$$v_{if+1,jf+1}^k = \frac{B_{if+1,jf+1}^k v_{if,jf+1}^k + B_{if+2,jf+1}^k v_{if+2,jf+1}^k + A_{if+1,jf+1}^k v_{if+1,jf}^k + A_{if+1,jf+2}^k v_{if+1,jf+2}^k}{B_{if+1,jf+1}^k + B_{if+2,jf+1}^k + A_{if+1,jf+1}^k + A_{if+1,jf+2}^k} \quad (2.3.10)$$

Pour de plus amples informations sur les coefficients d'interpolation, Bandy [3] donne dans son manuscrit de thèse les expressions détaillées des coefficients pour des cas 2D et 3D. Maintenant que les coefficients de prolongation I_{k-1}^k sont définis, il reste à établir les coefficients de restriction, ainsi que les opérateurs sur les grilles grossières. Les coefficients de restriction sont tout simplement les opérateurs adjoints des coefficients de prolongation $I_k^{k-1} = (I_{k-1}^k)^*$, correspondant ici au transposé de la matrice I_{k-1}^k . La matrice de l'opérateur sur les grilles grossières est calculée de façon classique dans les méthodes Multigrille :

$$A^{k-1} = I_k^{k-1} A^k I_{k-1}^k \quad (2.3.11)$$

Avec A^{k-1} l'opérateur sur la grille grossière $k-1$. L'opérateur initial étant une matrice diagonale symétrique définie positive dans nos simulations, il y a 5 diagonales non-nulles en 2D et 7 diagonales non-nulles en 3D. Néanmoins les opérateurs générés sur les grilles grossières vont avoir respectivement 9 et 27 diagonales non-nulles, de part la façon dont sont définis les coefficients d'interpolation.

Une fois tous ces éléments calculés, il faut choisir la méthode de relaxation qui sera employée pour résoudre quelques itérations du système à chaque niveau de grille.

Le choix de la méthode de relaxation est importante, car celle-ci contribue au nombre de cycles nécessaires à la résolution et la robustesse du solveur BBMG. Parmi les choix classiques (et non-exhaustifs) se trouvent les méthodes Jacobi et Gauss-Seidel sous plusieurs formes, avec le Point Gauss-Seidel, le Color Gauss-Seidel, ou encore le Alternative Line Gauss-Seidel. Dans cette étude nous avons privilégié le Gauss-Seidel au Jacobi car il converge plus rapidement. Différents tests de performance sont proposés par Bandy [3] et la méthode qui semble la plus robuste est le Color Gauss-Seidel. Afin d'illustrer le fonctionnement de cette méthode, prenons le cas d'un stencil à 5 points (un cas 2D pour la grille fine), où deux couleurs, Rouge et Noir, sont définies telles que :

$$\begin{aligned} \text{Rouge} : i + j &= \text{pair} \\ \text{Noir} : i + j &= \text{impair} \end{aligned} \quad (2.3.12)$$

La méthode est constituée de deux sous-itérations, durant laquelle chaque couleur va être utilisée. Pour les points rouges, la première sous-itération résout :

$$S u_{i,j-1}^n + W u_{i-1,j}^n + C u_{i,j}^{n+\frac{1}{2}} + E u_{i+1,j}^n + N u_{i,j+1}^n = F_{i,j} \quad (2.3.13)$$

avec $i+j$ qui est pair et S, N, W, E et C les coefficients de la matrice A . Vient ensuite la seconde sous-itération sur les points noirs :

$$S u_{i,j-1}^{n+\frac{1}{2}} + W u_{i-1,j}^{n+\frac{1}{2}} + C u_{i,j}^{n+1} + E u_{i+1,j}^{n+\frac{1}{2}} + N u_{i,j+1}^{n+\frac{1}{2}} = F_{i,j} \quad (2.3.14)$$

avec $i + j$ impair. Cette méthode est extrapolable à des matrices avec un stencil à 9 points (2D pour les grilles grossières) avec 4 couleurs, et à un stencil à 27 points avec 8 couleurs (3D sur les grilles grossières).

Tous les éléments nécessaires au bon fonctionnement d'un solveur BlackBox Multigrid étant présentés, il est maintenant possible d'établir l'algorithme que va suivre le solveur au cours d'une résolution.

2.3.2.3 Algorithme du BlackBox MultiGrid

Rappelons que la grille fine est notée G^M pour une discrétisation sur M grilles. Plusieurs cycles de résolution existent, dépendant chacun de l'ordre dans lequel les différentes grilles sont résolues. Dans le cadre de cette thèse le cycle en V est utilisé et présenté par la suite. Dans un premier temps, il y a l'initialisation du code, durant laquelle tous les coefficients d'interpolation et des opérateurs sont calculés pour tous les niveaux de grille. Commence ensuite le cycle en V :

1. Descente - $k = M$

Tant que $k \neq 1$

(a) Résoudre μ itérations de $A^k \mathbf{u}^k = \mathbf{b}^k$ avec le CGS

(b) Estimer le résidu $\mathbf{r}^k = \mathbf{b}^k - A^k \mathbf{u}^k$

(c) Restreindre le résidu sur la grille grossière $k - 1$ de sorte à calculer le 2^{nd} membre de l'équation : $\mathbf{b}^k = I_k^{k-1} \mathbf{r}^k$. Le nouveau problème sur la grille grossière à résoudre s'écrit donc : $A^{k-1} \mathbf{u}^{k-1} = \mathbf{b}^{k-1}$. (\mathbf{u}^{k-1} étant une correction sur l'erreur, celui-ci est initialisé comme étant nul)

(d) Si $(k - 1) = 1$ alors résolution de $A^1 \mathbf{u}^1 = \mathbf{b}^1$ avec un solveur direct

Si non $k = k - 1$

2. Remontée - $k = 2$

Tant que $k \leq M$

(a) Interpoler la solution sur la grille fine k pour corriger la solution :

$$\mathbf{v}^k \leftarrow \mathbf{v}^k + I_{k-1}^k \mathbf{v}^{k-1}$$

(b) Résoudre μ itérations de $A^k \mathbf{u}^k = \mathbf{b}^k$ avec le CGS

3. Si la convergence désirée est atteinte le solveur a résolu le problème.

Si non Recommencer le cycle en V.

Avec μ le nombre d'itérations effectuées dans le CGS (Color Gauss-Seidel). Dans le code que j'ai développé, μ est fixé à 1, avec donc une seule itération de CGS par grille à la descente et une à la remontée du cycle. D'autres alternatives sont possibles, en faisant plus d'itérations à la descente, et en ne faisant pas intervenir le CGS à la remontée. Dans la littérature, il est néanmoins observé que $\mu \in [1; 5]$, puisque le temps passé à faire plus de 5 itérations de CGS sur la grille fine est considérable par rapport au coût des itérations sur une grille plus grossière. L'objectif est donc de faire le moins d'itérations possible de CGS par cycle dans les grilles fines pour optimiser le temps du solveur.

2.3.2.4 Algorithme du BlackBox MultiGrid parallèle

Dans ce paragraphe nous proposons dans cette section une logique d'algorithmique originale, reposant sur un équilibre entre la mémoire utilisée par chaque processeur et le nombre de communications à effectuer pour effectuer les cycles du solveur en instaurant le principe de grille locale - grille globale.

Durant les résolutions sur les grilles fines, chaque processeur va résoudre un système linéaire avec une itération de Gauss-Seidel, qui sera dégradée en solveur Jacobi sur les bords du domaine du fait de la communication à effectuer sur les bords des domaines de chaque processeur. Ceci dégrade donc l'efficacité du solveur, mais reste une contrainte qui ne peut être contournée. C'est la résolution du système en grille locale. Toutefois lorsqu'un certain niveau de grille G^c défini par l'utilisateur est atteint, il est possible de faire une communication de tous les points de chaque processeur à chaque processeur. Ainsi chaque processeur possède l'information sur tout le domaine et peut résoudre les grilles les plus grossière sans communication et sans dégradation du solveur Gauss-Seidel en solveur Jacobi. La remontée s'effectue de façon globale jusqu'au niveau de grille critique G^c et une résolution locale est de nouveau effectuée.

Nous définissons ici la grille critique par l'indice c . L'algorithme est le suivant :

1. Descente locale - $k = M$

Tant que $k \neq c$

- (a) Résoudre μ itérations de $A^k \mathbf{u}^k = \mathbf{b}^k$ avec le CGS avec une communication de la solution aux bords du domaine de chaque processeur.
- (b) Estimer le résidu $\mathbf{r}^k = \mathbf{b}^k - A^k \mathbf{u}^k$
- (c) Restreindre le résidu sur la grille grossière $k - 1$ de sorte à calculer le 2^{nd} membre de l'équation : $\mathbf{b}^k = I_k^{k-1} \mathbf{r}^k$. Le nouveau problème sur la grille grossière à résoudre s'écrit donc : $A^{k-1} \mathbf{u}^{k-1} = \mathbf{b}^{k-1}$. (\mathbf{u}^{k-1} étant une correction sur l'erreur, celui-ci est initialisé comme étant nul).
- (d) Si $(k - 1) = c$ alors la communication globale de la solution de tous les processeurs vers tous les processeurs est effectuée. Sinon $k = k - 1$

2. Descente globale - $k = c$

Tant que $k \neq 1$

- (a) Résoudre μ itérations de $A^k \mathbf{u}^k = \mathbf{b}^k$ avec le CGS
- (b) Estimer le résidu $\mathbf{r}^k = \mathbf{b}^k - A^k \mathbf{u}^k$
- (c) Restreindre le résidu sur la grille grossière $k - 1$ de sorte à calculer le 2^{nd} membre de l'équation : $\mathbf{b}^k = I_k^{k-1} \mathbf{r}^k$. Le nouveau problème sur la grille grossière à résoudre s'écrit donc : $A^{k-1} \mathbf{u}^{k-1} = \mathbf{b}^{k-1}$. (\mathbf{u}^{k-1} étant une correction sur l'erreur, celui-ci est initialisé comme étant nul)
- (d) Si $(k - 1) = 1$ alors résolution de $A^1 \mathbf{u}^1 = \mathbf{b}^1$ avec un solveur direct
Sinon $k = k - 1$

3. Remontée globale - $k = 2$

Tant que $k \leq c$

- (a) Interpoler la solution sur la grille fine k pour corriger la solution :
- $$\mathbf{v}^k \leftarrow \mathbf{v}^k + I_{k-1}^k \mathbf{v}^{k-1}$$
- (b) Résoudre μ itérations de $A^k \mathbf{u}^k = \mathbf{b}^k$ avec le CGS
4. Remontée locale - $k = c$
Tant que $k \leq M$
- (a) Interpoler la solution sur la grille fine k pour corriger la solution :
- $$\mathbf{v}^k \leftarrow \mathbf{v}^k + I_{k-1}^k \mathbf{v}^{k-1}$$
- (b) Résoudre μ itérations de $A^k \mathbf{u}^k = \mathbf{b}^k$ avec le CGS avec une communication de la solution aux bords du domaine de chaque processeur.
5. *Si* la convergence désirée est atteinte le solveur a résolu le problème.
Sinon Recommencer le cycle en V.

2.3.2.5 Validations du BlackBox MultiGrid

Dans cette partie nous présentons deux configurations, respectivement 2D et 3D pour comparer les différentes performances des solveurs implémentés dans DIVA. Ce sont des tests comparatifs portant sur des cas diphasiques, avec une discontinuité prise en compte dans les coefficients de la matrice. Dans notre étude les effets de tension de surface sont pris en compte et modifient également le second membre.

Le premier cas est 2D, avec une goutte d'eau centrée dans un domaine de dimension $[-0.005; 0.005]^2$, de rayon $r = 0.0125$ m soumise à la gravité $(\mathbf{g}, \mathbf{e}_y) = -10$ m.s⁻² dans de l'air. Les caractéristiques physiques utilisées pour le problème sont les suivantes : $\rho_l = 1000$ kg.m⁻³, $\mu_l = 0.001$ Pa.s⁻¹, $\rho_g = 1$ kg.m⁻³, $\mu_g = 0.00001$ Pa.s⁻¹, et $\sigma = 0.07$ N.m⁻¹. Les conditions aux limites sont des parois sauf à droite (dans la direction de la gravité) où une condition libre est imposée. Le critère de convergence pour les solveurs est imposé à 10^{-6} . La résolution est effectuée sur quelques itérations temporelles. Le calcul est réalisé sur un processeur Xeon(R) X587@3.60GHz.

Maillage	IC-CG			BBMG sans sol init			BBMG avec sol init		
	temps (s)	ité	Res	temps	ité	Res	temps	ité	Res
32 ²	0.002	52	7.8.10 ⁻⁷	0.0014	9	5.21.10 ⁻⁷	0.0008	6	6.69.10 ⁻⁷
64 ²	0.012	96	8.8.10 ⁻⁷	0.0045	8	7.61.10 ⁻⁷	0.0024	5	7.20.10 ⁻⁷
128 ²	0.093	185	9.27.10 ⁻⁷	0.018	9	3.22.10 ⁻⁷	0.0093	4	8.5.10 ⁻⁷
256 ²	0.72	367	8.8.10 ⁻⁷	0.071	10	9.86.10 ⁻⁷	0.043	5	3.4.10 ⁻⁷
512 ²	6.50	703	9.8.10 ⁻⁷	0.29	9	6.74.10 ⁻⁷	0.15	4	3.75.10 ⁻⁷
1024 ²	54.69	1386	9.9.10 ⁻⁷	1.29	10	4.26.10 ⁻⁷	0.64	4	8.49.10 ⁻⁷
2048 ²	455.6	2445	9.89.10 ⁻⁷	5.26	10	6.6.10 ⁻⁷	2.33	3	9.2.10 ⁻⁷

TABLE 2.8 – Comparaison des performances 2D des méthodes IC-CG et BBMG avec et sans solution initiale.

Le tableau 2.8 présente les performances des différents solveurs, avec le gradient préconditionné par une méthode de Cholesky incomplète (IC-CG) et le solveur BlackBox MultiGrid (BBMG) dont

les résultats sont présentés avec et sans solution initiale (de l'itération précédente) en entrée. La figure 2.8 présente les temps des différentes méthodes en fonction du nombre de points dans le maillage. Deux droites sont tracées pour comparer la convergence des méthodes. Le BBMG, avec ou sans solution initiale met 4 fois plus de temps lorsque 4 fois plus de points sont à prendre en compte, alors que l'IC-CG nécessite environ 8 fois plus de temps pour 4 fois plus de points. Ceci permet de constater le bon fonctionnement de la méthode BBMG implémentée dans DIVA en 2D. Enfin l'avantage d'utiliser une telle méthode est la possibilité de lui donner la solution calculée à l'itération temporelle d'avant en entrée.

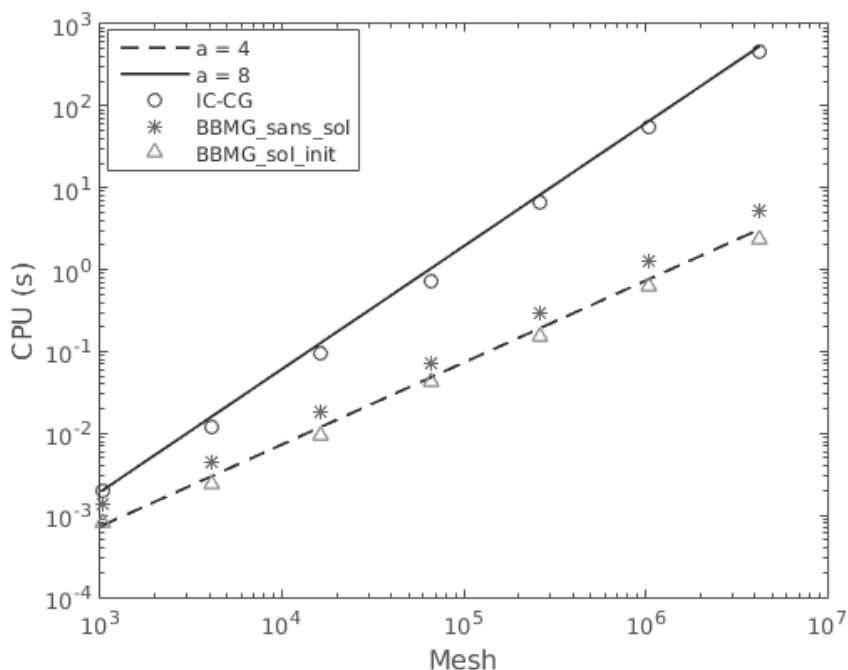


FIGURE 2.8 – Performances des différentes méthodes, IC-CG, BBMG avec et sans solution initiale. La droite en pointillés a un coefficient directeur $a = 4$, et $a = 8$ pour la droite à trait plein.

Le second test est un cas diphasique 3D, avec une discontinuité prise en compte dans les coefficients de la matrice. Une goutte d'eau centrée dans un domaine de dimension $[-0.005; 0.005]^3$, de rayon $r = 0.0125$ m est soumise à la gravité ($\mathbf{g.ez} = -10$ m.s⁻²) dans de l'air. Les caractéristiques physiques utilisées pour le problème sont les suivantes : $\rho_l = 1000$ kg.m⁻³, $\mu_l = 0.001$ Pa.s⁻¹, $\rho_g = 1$ kg.m⁻³, $\mu_g = 0.00001$ Pa.s⁻¹, et $\sigma = 0.07$ N.m⁻¹. Les conditions aux limites sont des parois sauf en-dessous (dans la direction de la gravité) où une condition libre est imposée. Une représentation du cas test est proposée dans la figure 2.9. Le critère de convergence pour les solveurs est imposé à 10^{-6} . La résolution est effectuée sur quelques itérations temporelles. Le calcul est effectué sur le même processeur que pour le 2D, un CPU Xeon(R) X587@3.60GHz.

Le tableau 2.9 présente les performances des différents solveurs, avec le gradient préconditionné par un Cholesky incomplet (IC-CG), et le solveur BlackBox MultiGrille (BBMG) dont les résultats sont présentés avec et sans solution initiale (de l'itération précédente) en entrée. La figure 2.9 présente les temps des différentes méthodes en fonction du nombre de points dans le maillage. Deux

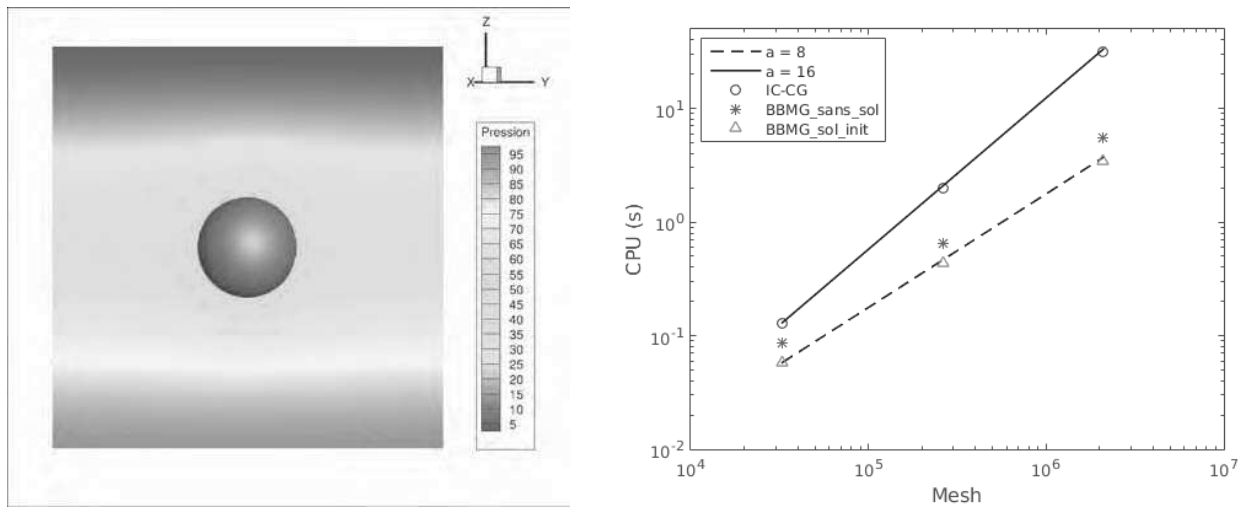


FIGURE 2.9 – A gauche, illustration du cas test 3D avec le champ pression à $t = 0$. A droite, la performances des différentes méthodes, IC-CG, BBMG avec et sans solution initiale. La droite en pointillés a un coefficient directeur $a = 8$, et $a = 16$ pour la droite à trait plein.

Maillage	IC-CG			BBMG sans sol init			BBMG avec sol init		
	temps (s)	ité	Res	temps	ité	Res	temps	ité	Res
32^3	0.13	128	$7.74 \cdot 10^{-7}$	0.087	13	$6.89 \cdot 10^{-7}$	0.058	8	$7.63 \cdot 10^{-7}$
64^3	1.98	220	$9.72 \cdot 10^{-7}$	0.65	12	$7.14 \cdot 10^{-7}$	0.43	6	$8.07 \cdot 10^{-7}$
128^3	31.64	413	$7.85 \cdot 10^{-7}$	5.54	12	$8.12 \cdot 10^{-7}$	3.47	6	$6.14 \cdot 10^{-7}$

TABLE 2.9 – Comparaison des performances 3D des méthodes IC-CG et BBMG avec et sans solution initiale.

droites sont tracées pour vérifier la convergence des méthodes. Le BBMG, avec ou sans solution initiale mets 8 fois plus de temps lorsque 8 fois plus de points sont à prendre en compte, alors que l'IC-CG nécessite 16 fois plus de temps pour 8 fois plus de points. Ceci permet de constater le bon fonctionnement de la méthode BBMG implémentée dans DIVA en 3D.

Le même problème a été résolu avec l'algorithme parallèle sur le supercalculateur EOS du CAL-MIP composé de 12240 processeurs Intel ivybride cadencés à 2.8Ghz répartis sur 612 noeuds de calcul, ce qui a permis de tester le code sur plusieurs centaines de processeurs.

Le tableau 2.10 présente les résultats obtenus. Lors de la résolution parallèle, le solveur IC-CG de PETSc se dégrade en préconditionneur de type Jacobi par blocs, le rendant moins performant qu'un IC-CG classique (qui ne peut pas être utilisé de manière générale dans les résolutions parallèles). Dans un premier temps, nous pouvons constater que le solveur BBMG est plus performant que le solveur de PETSc, avec un rapport de temps entre les temps de calculs qui varie de 10 à 60 sur les simulations effectuées.

Maillage	Nbs de processeurs	BBMG avec sol init		BJ-CG de PETSc	
		ité	temps (s)	ité	temps (s)
256^3	8	10	4.7	400	60
256^3	64	12	1.3	550	14
512^3	64	12	8.7	-	-
512^3	512	13	1.85	1600-1800	100-120
1024^3	512	13	11.15	-	-

TABLE 2.10 – Comparaison des performances 3D parallèles des méthodes BJ-CG de PETSc et BBMG avec solution initiale.

2.4 Modélisation des sauts à l’interface

Comme nous l’avons vu précédemment, la Ghost Fluid Method est une méthode utilisée pour imposer des discontinuités à l’interface. C’est une méthode efficace et robuste notamment pour les sauts de variables. Néanmoins des tests effectués au cours de cette thèse ont mis en évidence que l’ordre de la méthode est fluctuant lorsque des sauts dans les coefficients de diffusions (μ et ρ par exemple) sont à prendre en compte. Pour des sauts de ρ de l’ordre de 1000, généralement rencontrés dans les écoulements diphasiques, la méthode est d’ordre 1,5 environ. Mais lorsque le rapport des masses volumiques est plus faible, l’ordre descend, et peut atteindre un ordre $O(0.8)$. Ceci est dû au fait que lorsque le coefficient de diffusion est modifié, les coefficients de la matrice sont directement affectés.

C’est pour cette raison qu’une étude plus poussée a été menée conjointement à l’IMFT et l’Université de Californie, Santa Barbara (UCSB) dans le département de science computing, lors d’une collaboration effectuée avec A. Guittet et F. Gibou. J’ai effectué un séjour d’un mois en Californie durant lequel nous avons cherché à mettre au point une méthode précise à l’ordre 2 en espace pour les conditions de saut à l’interface. Les travaux sont décrits dans un article, qui fait l’objet de l’annexe. Le principe de la méthode est exposé succinctement, et le papier est disponible à la fin de ce chapitre. Des tests comparatifs entre la nouvelle méthode appelée méthode de Voronoï (VIM) et la Ghost Fluid Method pour un cas analytique sont également présentés.

2.4.1 Principe de la méthode Voronoï

Le principe de cette nouvelle méthode repose sur des diagrammes de Voronoï. Les mailles traversées par l’interface sont modifiées, de façon à ce qu’un des bords de la nouvelle cellule corresponde à l’interface. Ce nouveau maillage permet d’effectuer une discrétisation directe des sauts dans la direction normale de l’interface. La discrétisation spatiale issue de cette méthode permet de préserver un système linéaire symétrique, défini positif. Une fois la solution calculée, celle-ci est reprojctée sur le maillage cartésien avec une méthode qui permet de conserver l’ordre 2 en espace.

2.4.2 Tests comparatifs entre le Ghost Fluid et le Voronoï

Considérons le problème suivant :

$$\begin{cases} \nabla \cdot (\beta \nabla u) = f \\ [u]_{\Gamma} = 0 \\ \left[\beta \frac{du}{dx} \right]_{\Gamma} = 0 \\ [\beta]_{\Gamma} \neq 0 \end{cases} \quad (2.4.1)$$

Avec la solution $u(x, y)$ suivante :

$$u(x, y) \begin{cases} \frac{x(\rho + 1) - x(\rho - 1)r_0^2/r^2}{\rho + 1 + r_0^2(\rho - 1)} & \text{pour } \phi(x, y) < 0 \\ \frac{2x}{\rho + 1 + r_0^2(\rho - 1)} & \text{pour } \phi(x, y) > 0 \end{cases} \quad (2.4.2)$$

avec $r_0 = 0.5$, $r = \sqrt{x^2 + y^2}$, $\phi(x, y) = r_0 - r$ et $\rho = \beta^+/\beta^-$ dans le domaine $[-1, 1]^2$. Ce cas test est extrait d'une étude menée par Berthelsen (2004) [5], et a été repris dans Guittet *et al.* (2015) [20]. Dans cette étude la fonction u est continue, mais le coefficient de diffusion ne l'est pas. La figure 2.10 présente l'allure de la solution.

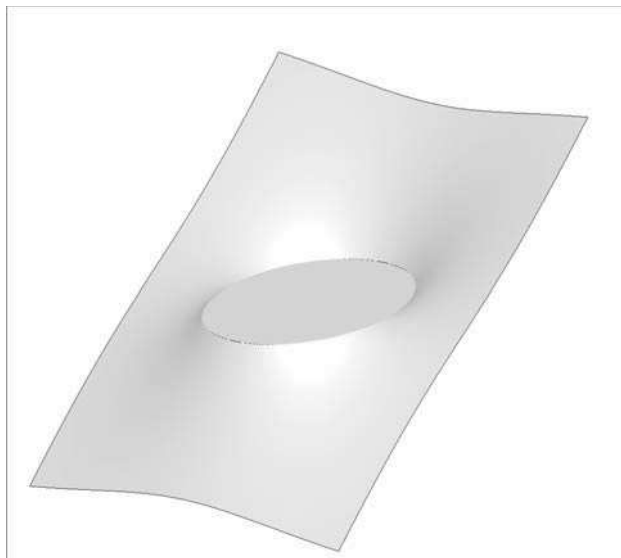


FIGURE 2.10 – Allure de la solution (eq. 2.4.2) pour un saut $\beta = 1000$.

Les tests ont été menés en faisant varier ρ le rapport des coefficients de diffusion pour des valeurs de 10, 100 et 1000. Ces résultats sont reportés respectivement dans les tableaux 2.11, 2.12 et 2.13

Comme nous pouvons le constater dans les tableaux, la méthode Voronoï donne d'excellents résultats, avec une convergence d'ordre $O(2)$ sur presque tous les cas alors que la méthode Ghost Fluid donne une convergence moins régulière, notamment pour les petits rapports de densités. Ceci pourrait s'expliquer par le fait que le vecteur normal est projeté dans les directions cartésiennes du maillage, ce qui a pour conséquence de négliger la composante tangentielle du saut. Ainsi les erreurs générées par l'omission des composantes tangentielles deviennent du même ordre que la composante normale lorsque de faibles rapports du coefficient de diffusion rentrent en jeu, et deviennent négligeables lorsque le rapport augmente.

Maillage	Ghost Fluid Method		Voronoi Method	
	e_{max}	$O(e_{max})$	e_{max}	$O(e_{max})$
8^2	0.0094	-	0.014	-
16^2	0.0041	1.19	0.0028	2.34
32^2	0.0036	0.18	0.00054	2.33
64^2	0.0024	0.58	0.00016	1.77
128^2	0.0013	0.88	0.00005	1.57
256^2	0.00080	0.7	0.00001	2.00
512^2	0.00044	0.86	0.0000038	1.84
1024^2	0.00022	1	0.0000016	1.89

TABLE 2.11 – Comparaison entre la méthode Ghost Fluid et la méthode Voronoï pour un rapport de densités $\beta = 10$

Maillage	Ghost Fluid Method		Voronoi Method	
	e_{max}	$O(e_{max})$	e_{max}	$O(e_{max})$
8^2	0.0116	-	0.017	-
16^2	0.0032	1.85	0.0034	2.31
32^2	0.00087	1.87	0.00069	2.33
64^2	0.00032	1.44	0.00019	1.85
128^2	0.00019	0.75	0.000068	1.49
256^2	0.00011	0.79	0.000016	2.06
512^2	0.000065	0.76	0.0000048	1.77
1024^2	0.000035	0.89	0.0000013	1.44

TABLE 2.12 – Comparaison entre la méthode Ghost Fluid et la méthode Voronoï pour un rapport de densités $\beta = 100$

La méthode VIM n'a pas encore été implémentée dans un code Navier-Stokes, et fait encore l'objet de tests. C'est pour cette raison que la méthode Ghost Fluid reste la méthode utilisée dans le cadre de cette thèse.

Maillage	Ghost Fluid Method		Voronoi Method	
	e_{max}	$O(e_{max})$	e_{max}	$O(e_{max})$
8^2	0.012	-	0.017	-
16^2	0.0028	2.05	0.0035	2.3
32^2	0.00082	1.79	0.00071	2.32
64^2	0.00023	1.87	0.000194	1.86
128^2	0.000075	1.58	0.000069	1.48
256^2	0.000023	1.7	0.000017	2.07
512^2	0.0000072	1.67	0.0000049	1.76
1024^2	0.0000037	0.96	0.0000013	1.88

TABLE 2.13 – Comparaison entre la méthode Ghost Fluid et la méthode Voronoï pour un rapport de densités $\beta = 1000$

Bibliographie

- [1] S. Afkhami, S. Zaleski, M. Bussmann, A mesh-dependent model for applying dynamic contact angles to VOF simulations, *J. Comput. Phys.* 228 (2009) 5370-5389.
- [2] R.E. Alcouffe, A. Brandt, J.E. Dendy, J.W. Painter, The multigrid method for the diffusion equation with strongly discontinuous coefficients, *SIAM J. Sci. Stat. Comput.* 2 (1981) 430–454.
- [3] V. A. Bandy, Black Box Multigrid for convection-diffusion equations on advanced computers, PhD Thesis defended in 1996 in the University of Colorado at Denver, United States.
- [4] A. Blanco, J. Magnaudet, The structure of the axisymmetric high-Reynolds number flow around an ellipsoidal bubble of fixed shape. *Phys. Fluids* 7 (1995) 1265.
- [5] P. A. Berthelsen. A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions. *Journal of Computational Physics*, 197 (2004) 364-386.
- [6] R. Borges, M. Carmona, B. Costa, W.S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws, *J. Comput. Phys.* 227 (2008) 3191-3211.
- [7] A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comp.* 31 (1977), pp.333-390.
- [8] D. Braess, On the combination of the multigrid method and conjugate gradients, *Multigrid Methods, II (Cologne, 1985)*, Lecture Notes in Math., vol. 1228, Springer, Berlin, 1986, pp. 52–64.
- [9] B. Bunner, G. Tryggvason ,Dynamics of homogeneous bubbly flows Part 1. Rise velocity and microstructure of the bubbles, *J. Fluid Mech.*, 446 (2002) 17-52.
- [10] I.-L. Chern, J. Glimm, O. McBryan, B. Plohr, S. Yaniv, Front tracking for gas dynamics. *J. Comput. Phys.*,62 (1986) 83–110.
- [11] A. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* 2 (1967) 12-26.
- [12] B.J. Daly, W.E. Pracht, Numerical study of density-current surges, *Phys. Fluids*,11 (1968) 15-30.
- [13] J.E. Dendy, Black box multigrid, *J. Comput. Phys.* 48 (3) (1982) 366-386.
- [14] J.E. Dendy, Two multigrid methods for three-dimensional equations with highly discontinuous coefficients, *SIAM J. Sci. Stat. Comput.* 8 (1987) 673-685.

- [15] J.E. Dendy, Black box multigrid for periodic and singular problems, *Appl. Math. Comput.* 25 (1, part I) (1988) 1-10
- [16] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (The Ghost Fluid Method), *J. Comput. Phys.* 152 (1999) 457-492.
- [17] J. Glimm, D. Marchesin, O. McBryan, A numerical method for two phase flow with an unstable interface, *J. Comput. Phys.*, 39 (1981) 179–200.
- [18] J. Glimm, O. McBryan, A computational model for interfaces, *Adv. Appl. Math.*, 6 (1985) 422–435.
- [19] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, *J. Comput. Phys.*, 152 (1999) 423–456.
- [20] A. Guittet, M. Lepilliez, S. Tanguy, F. Gibou, Solving elliptic problems with discontinuities on irregular domains - the Voronoi Interface Method, *J. Comput. Phys.* 298 (2015) 747-765.
- [21] F. Harlow and J. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Physics of Fluids*, 8 (1965) :2182–2189.
- [22] M. R. Hestenes, E. Stiefel, Methods of Conjugate Gradients for solving linear Systems, *J. of Research of the National Bureau of Standards*, 49 (1952) 409-436.
- [23] C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries, *J. Comput. Phys.*, 39 (1981) 201–225.
- [24] C. Josserand, L. Lemoyne, R. Troeger, and S. Zaleski. Droplet impact on a dry surface : triggering the splash with a small obstacle. *Journal of Fluid Mechanics*, 524 (2005) 47–56.
- [25] M. Kang, R. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, *J. Sci. Comput.* 15 (2000) 323-360.
- [26] R. Kettler, J. Meijerink, A multigrid method and a combined multigrid-conjugate gradient method for elliptic problems with strongly discontinuous coefficients in general domains, *Tech. Rep.* 604, Shell Oil Company, 1981.
- [27] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, Modelling merging and fragmentation in multiphase flows with surfer, *J. Comput. Phys.* 113 (1994) 134–147.
- [28] B. Lalanne, L. Rueda Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost Fluid Method, *J. Comput. Phys.* 301 (2015) 289-307.
- [29] X.-D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson’s equation on irregular domains, *J. Comput. Phys.* 160 (2000) 151-178.
- [30] H. Liu, S. Krishnan, S. Marella, H.S. Udaykumar, Sharp interface Cartesian grid method II : A technique for simulating droplet interactions with surfaces of arbitrary shape, *J. Comput. Phys.* 210 (2005) 32-54.
- [31] S.P. MacLachlan, J.M. Tang, C. Vuik, Fast and robust solvers for pressure-correction in bubbly flow problems, *J. Comput. Phys.* 227 (2008) 9742-9761.

- [32] S. Marella, S. Krishnan, H. Liu, H.S. Udaykumar, Sharp interface Cartesian grid method I : An easily implemented technique for 3D moving boundary computations, *J. Comput. Phys.* 210 (2005) 1-31.
- [33] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid. Mech.*, (2005) 239-261.
- [34] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, and A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.* 227 (2008) 4825-4852.
- [35] J.D. Moulton, J.E. Dendy, J.M. Hyman, The black box multigrid numerical homogenization algorithm, *J. Comput. Phys.* 141 (1998) 1-29.
- [36] Y.T. Ng, C. Min, F. Gibou, An efficient fluid–solid coupling algorithm for single-phase flows, *J. Comput. Phys.* 228 (2009) 8807-8829.
- [37] W. F. Noh and P. Woodward, Slic (simple line interface calculation). *Lecture Notes in physics, Proceedings of the 5th International Conference on Numerical methods in Fluid Dynamics* 59 (1976) 330–340.
- [38] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed : algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1988) 12-49.
- [39] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, *J. Comput. Phys.* 228 (2009) 5838-5866.
- [40] E. G. Puckett, A. Almgren, J. Bell, D. Marcus, and W. Rider. A high-order projection method for tracking interfaces in variable density incompressible flows, *J. Comput. Phys.* 130 (1997) 269.
- [41] W. J. Rider and D. B. Kothe. Reconstructing volume tracking. *J. Comput. Phys.* 141 (1998) 112–152.
- [42] L. Rueda-Villegas, *Simulation Numérique Direct de l’Effet Leidenfrost*, PhD Thesis defended on 10.12.2013 at IMFT, Toulouse
- [43] G. Ryskin, L.G. Leal, Numerical solution of free boundary problems in fluid mechanics. Part 2. Buoyancy driven motion of a gas bubble through a quiescent liquid, *J. Fluid. Mech.* 148 (1984) 19-35.
- [44] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* 31 (1999) 567-603.
- [45] M. Sussman, P. Smereka, S. Osher, A Level Set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* 114 (1994) 146-159.
- [46] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, *J. Comput. Phys.* 221 (2007) 469-505.
- [47] S. Tanguy, Développement d’une méthode de suivi d’interface. Applications aux écoulements diphasiques. PhD Thesis defend in 2004, at University of Rouen.
- [48] S. Tanguy, M. Sagan, B. Lalanne, F. Couderc, C. Colin, Benchmarks and numerical methods for the simulation of boiling flows, *J. Comput. Phys.* 264 (2014) 1-22.

- [49] O. Tatebe, The multigrid preconditioned conjugate gradient method, in : N.D. Melson, T.A. Manteuffel, S.F. McCormick (Eds.), Sixth Copper Mountain Conference on Multigrid Methods, vol. CP 3224, NASA, Hampton, VA, 1993, pp. 621-634.
- [50] G. Tryggvason, R. Scardovelli, S. Zaleski, Direct Numerical Simulations of Gas-Liquid Multiphase Flows, Cambridge press, 2011.
- [51] H.S. Udaykumar, H.C Kan, W. Shyy, R. Tran-Son-Tay, Multiphase dynamics in arbitrary geometries on fixed Cartesian grids. *J. Comput. Phys.*, 137 (1997) 366–405.
- [52] H.S. Udaykumar, R. Mittal, W. Shyy, Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids. *J. Comput. Phys.*, 153 (1999) 535–574.
- [53] H.S. Udaykumar, R. Mittal, P. Rampungoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, *J. Comput. Phys.*, 174 (2001) 345–380.
- [54] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100 (1992) 25–37.
- [55] J. Delteil, S. Vincent, A. Erriguible, P. Subra-Paternault, Numerical investigations in Rayleigh breakup of round liquid jets with VOF methods, *Computers & Fluids* 50 (2011) 10-23
- [56] D. L. Youngs, Time dependent multi-material flow with large fluid distortion, Morton and Baynes, 1982.



Solving elliptic problems with discontinuities on irregular domains – the Voronoi Interface Method



Arthur Guittet^{a,*}, Mathieu Lepilliez^{c,d,e}, Sebastien Tanguy^c, Frédéric Gibou^{a,b}

^a Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, United States

^b Department of Computer Science, University of California, Santa Barbara, CA 93106-5110, United States

^c Institut de Mécanique des Fluides de Toulouse, 2bis allée du Professeur Camille Soula, 31400 Toulouse, France

^d Centre National d'Etudes Spatiales, 18 Avenue Edouard Belin, 31401 Toulouse Cedex 9, France

^e Airbus Defence & Space, 31 Avenue des Cosmonautes, 31402 Toulouse Cedex 4, France

ARTICLE INFO

Article history:

Received 1 December 2014

Received in revised form 27 April 2015

Accepted 6 June 2015

Available online 2 July 2015

Keywords:

Level-set

Elliptic interface problems

Discontinuous coefficients

Irregular domains

Voronoi

Finite volumes

Quad/octrees

Adaptive mesh refinement

ABSTRACT

We introduce a simple method, dubbed the Voronoi Interface Method, to solve Elliptic problems with discontinuities across the interface of irregular domains. This method produces a linear system that is symmetric positive definite with only its right-hand-side affected by the jump conditions. The solution and the solution's gradients are second-order accurate and first-order accurate, respectively, in the L^∞ norm, even in the case of large ratios in the diffusion coefficient. This approach is also applicable to arbitrary meshes. Additional degrees of freedom are placed close to the interface and a Voronoi partition centered at each of these points is used to discretize the equations in a finite volume approach. Both the locations of the additional degrees of freedom and their Voronoi discretizations are straightforward in two and three spatial dimensions.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

We focus on the class of Elliptic problems that can be written as:

$$\begin{aligned} \nabla \cdot (\beta \nabla u) + ku &= f && \text{in } \Omega^- \cup \Omega^+, \\ [u] &= g && \text{on } \Gamma, \\ [\beta \nabla u \cdot \mathbf{n}_\Gamma] &= h && \text{on } \Gamma, \end{aligned} \quad (1)$$

where the computational domain Ω is composed of two subdomains, Ω^- and Ω^+ , separated by a co-dimension one interface Γ (see Fig. 1), with \mathbf{n}_Γ the outward normal. Here, $\beta = \beta(\mathbf{x})$, with $\mathbf{x} \in \mathbb{R}^n$ ($n \in \mathbb{N}$), is bounded from below by a positive constant and $[q] = q_\Gamma^+ - q_\Gamma^-$ indicates a discontinuity in the quantity q across Γ , f is in L^2 , g , h and k are given. Note that this general formulation includes possible discontinuities in the diffusion coefficient β and in the gradient of the solution ∇u . Dirichlet or Neumann boundary conditions are applied on the boundary of Ω , denoted by $\partial\Omega$. This class of equations, where some or all of the jump conditions are non-zero, is a corner stone in the modeling of the dynamics of

* Corresponding author.

E-mail address: arthur.guittet@gmail.com (A. Guittet).

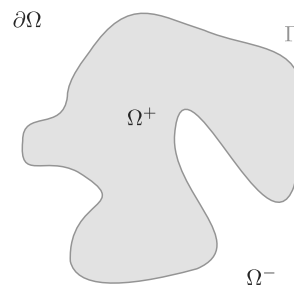


Fig. 1. Geometry of the problem.

important physical and biological phenomena as diverse as multiphase flows with and without phase change, biomolecules' electrostatics, electrokinetics (Poisson–Nernst–Planck) models with source term or electroporation models.

Given the importance of this class of equations, several approaches have been pursued to computationally approximate their solutions, each with their own pros and cons. The finite element method (FEM) is one of the earliest approaches to solve this problem [4,10,12,19,32,35] and has the advantage of providing a simple discretization formalism that guarantees the symmetry and definite positiveness of the corresponding linear system, even in the case of unstructured grids. It also provides a framework where *a priori* error estimates can be used to best adapt the mesh in order to capture small scale details. However, the FEM is based on the generation of meshes that must conform to the irregular domain's boundary and must satisfy some restrictive quality criteria, a task that is difficult, especially in three spatial dimensions. The difficulty is exacerbated when the domain's boundary evolves during the course of a computation, as it is the case for most of the applications modeled by these equations. Mesh generation is the focus of intense research [58], as the creation of unwanted sliver elements can deteriorate the accuracy of the solution.

Methods based on *capturing* the jump conditions do not depend on the generation of a mesh that conforms to the domain's boundary, hence avoiding the mesh generation difficulty altogether. However, they must impose the boundary conditions implicitly, which is a non-trivial task. A popular approach is the Immersed Interface Method (IIM) of Leveque and Li [36], and the more recent development of Immersed Finite Element Method (IFEM) and Immersed Finite Volume Method (IFVM) [40,27,23]. The basis of the IIM is to use Taylor expansions of the solution on each side of the interface and modify the stencils local to the interface in order to impose the jump conditions. As such, solutions can be obtained on simple Cartesian grids and the solution is second-order accurate in the L^∞ norm. The corresponding linear system, however, is asymmetric unless the coefficient β has no jump across the interface. Another difficulty is the need to approximate surface derivatives along Γ as well as the evaluation of high-order jump conditions. These difficulties have been addressed in the Piecewise-polynomial Interface Method of Chen and Strain [11] and several other approaches have improved the efficacy of the IMM [38,39,64,9,1–3]. We note also that the earliest approach on Cartesian grid is that of Mayo [43], who derived an integral equation to solve the Poisson and the bi-harmonic problems with piecewise constant coefficients on irregular domains; the solution is second-order accurate in the L^∞ norm. We also refer the interested researcher to the matched interface and boundary (MIB) method [66,65].

The finite element community has also proposed embedded interface approaches, including discontinuous Galerkin and the eXtended Finite Element Method (XFEM) [37,29,48,17,8,47,33,22,28,62]. The basic idea is to introduce additional degrees of freedom¹ near the interface and augment the standard basis functions on these elements with basis functions that are combined with a Heaviside function in order to help capture the jump conditions.

In [16], the authors introduce a second-order accurate discretization in the case of, possibly adaptive, Cartesian meshes using a cut-cell approach. The jump condition is imposed by determining the fluxes on both side of the interface, which are constructed from a combination of least squares and quadratic approximations. In [51], the authors also use a cut cell approach but impose the jump with the help of a compact 27-point stencil.

The Ghost Fluid Method (GFM), originally introduced to approximate two-phase compressible flows [21], has been applied to the system problem (1) in [41]. The basic idea is to consider fictitious domains and ghost values that capture the jump conditions in the discretization at grid nodes near the interface. An advantage of this approach is that only the right-hand-side of the linear system is affected by the jump conditions. However, in order to propose a dimension-by-dimension approach, the projection of the normal jump conditions must be projected onto the Cartesian directions. As a consequence, the tangential component of the jump is ignored. Nonetheless, the method has been shown to be convergent with first-order accuracy [42]. The GFM was also shown to produce symmetric positive definite second-order accuracy [25] and even fourth-order accuracy [24], but for a different class of problem, namely for solving Elliptic problems on irregular domains with Dirichlet boundary conditions. In fact, symmetric positive definite second-order accurate solutions can also be obtained in the case where Neumann or Robin boundary conditions are imposed on irregular domains [55,50,53,54]. These methods can be trivially extended to the case of adaptive Cartesian grids and we refer the interested readers to the review of Gibou, Min and Fedkiw [26] for more details. In the case of jump conditions, Coco and Russo [14] have also used a fictitious domain

¹ We understand by degrees of freedom the set of locations at which the solution is sampled.

approach, where a relaxation scheme is used to impose the boundary condition; the solution is second-order accurate. The same authors have also introduced a method to consider Dirichlet, Neumann and Robin boundary conditions on an irregular interface [15]. Latige et al. [46] have also presented a method based on fictitious domains using a piecewise polynomial representation of the solution on a dual grid, also obtaining second-order accurate solutions. Finally, [31] have applied the ghost fluid idea in a variational framework.

Related ideas are used in methods combining fictitious domains and variational formulations [49,7], dubbed virtual nodes approaches. Some of these approaches can be considered similar to XFEM methods [30,60,18], while others are different and offer advantages when considering under-resolved, possibly non-smooth, interfaces [49,59,56]. This philosophy has been used in [34], which introduces a virtual node algorithm for solving Elliptic problems on irregular geometries with jump conditions as well as Dirichlet or Neumann boundary conditions imposed on Γ . The solutions are second-order accurate in the L^∞ norm and the approach provides a unifying treatment for Dirichlet, Neumann and jump boundary conditions; however sacrificing simplicity.

Rather recently, Cisternino and Weynans [13] introduced a second-order accurate method that uses additional degrees of freedom on the domain's boundary and use them to discretize the Poisson operator with jump conditions in a dimension-by-dimension framework. The authors also present how to carefully approximate the gradients. The method produces second-order accurate solutions and a nonsymmetric linear system in part because of the need to change the size of the stencil for nodes adjacent to the domain's boundary.

We introduce a capturing computational approach, the Voronoi Interface Method, that produces second-order accurate solutions in the L^∞ norm. This approach is based on building a Voronoi diagram local to the interface, which enables the direct discretization of the jump conditions in the normal direction. The linear system is symmetric positive definite and the jump conditions only influence the right-hand-side. The construction of the local Voronoi mesh is a straightforward and parallelizable process and can be built with existing libraries that are freely available. In the present work, we use the excellent `Voro++` library in three spatial dimensions [57]. This method is different from body-fitted methods in that it relies on the post-processing of an existing background mesh, thus avoiding the standard difficulties associated with body-fitted approaches. We note that previous works have developed solvers for the Poisson equation on Voronoi diagrams (see [63,61] and the references therein); however discontinuities across an irregular interface were not considered.

2. The geometrical tools

2.1. The level-set method

The level-set method [52] is a powerful way of representing irregular interfaces as the zero contour of a continuous function. This representation is convenient in that it can be applied to the case of moving boundaries that can change their topology. It also provides a framework that lends itself to design *sharp* discretizations.

We use the level-set set framework to capture the irregular interface on which the discontinuities are enforced. We define a level-set function ϕ on the domain Ω such that the irregular interface Γ is described by $\Gamma = \{\mathbf{x} \in \mathbb{R}^n \in \Omega \mid \phi(\mathbf{x}) = 0\}$ and ϕ is negative on one side of the interface and positive on the other side, as pictured in Fig. 1. Even though infinitely many functions satisfy this criteria, it is convenient to work with a signed distance function to the irregular interface, i.e. a function that is negative on one side of the interface, positive on the other side and such that its magnitude at every point is the distance from the point to the interface. Constructing a signed distance function from an arbitrary function can be done for example by following the procedure explained in [44]. The normal to the interface is then obtained as

$$\underline{\mathbf{n}} = \frac{\underline{\nabla}\phi}{\|\underline{\nabla}\phi\|},$$

and the curvature as

$$\kappa = \underline{\nabla} \cdot \underline{\mathbf{n}}.$$

Note that if the level-set function is a signed distance function, $\|\underline{\nabla}\phi\| = 1$, and the projection onto Γ of any given point $\underline{\mathbf{x}}$ is easily computed as:

$$\underline{\mathbf{x}}_\Gamma = \underline{\mathbf{x}} - \phi(\underline{\mathbf{x}})\underline{\nabla}\phi(\underline{\mathbf{x}}). \quad (2)$$

2.2. Voronoi diagrams

The solver we present is based on Voronoi diagrams, which can be generated locally with existing procedures and freely available libraries. In this work, we use the excellent `Voro++` library [57]. For the sake of clarity, we introduce the Voronoi diagram: given a set of points, which we call seeds, the Voronoi cell of a given seed consists of all the points of the domain that are closer to that seed than to any other seed. Hence, the collection of all the Voronoi cells of a set of seeds is a tessellation of the domain, i.e. a tiling that fills the domain and does not contain any overlaps.

Given a computational mesh, which we consider to be uniform in this section for clarity, we propose to modify the mesh so that the irregular interface coincides with the edges of the new mesh and the degrees of freedom close to the interface are all located at the same distance from the interface.

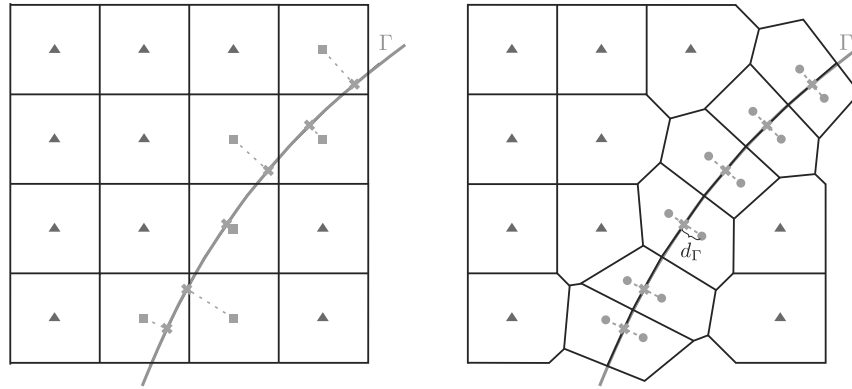


Fig. 2. Illustration of the procedure for generating a Voronoi diagram based computational mesh. The left figure presents the original uniform mesh and the right figure shows the final computational mesh. The purple square degrees of freedom have been removed and the orange dots degrees of freedom have been added close to the interface Γ .

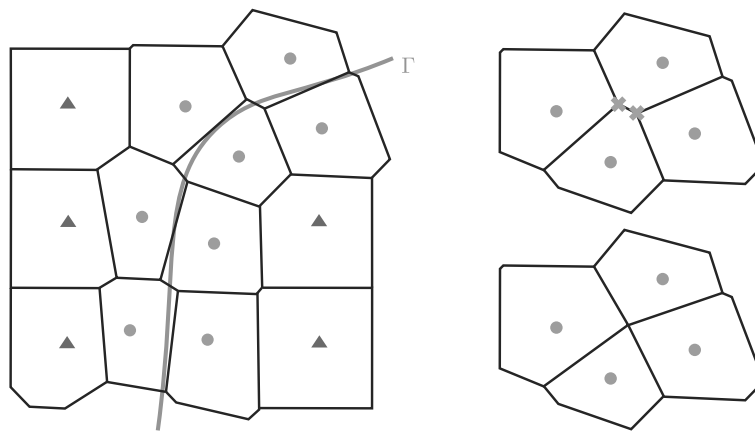


Fig. 3. Example of a Voronoi mesh where a degree of freedom is connected to more than one other degree of freedom located on the other side of the irregular interface. The smoothing procedure is illustrated on the right.

The procedure is illustrated in Fig. 2. Starting from a uniform grid, we find the projection of the degrees of freedom whose control volume is crossed by the irregular interface onto the interface using (2), and we remove those degrees of freedom from the original list of unknowns. If a projected point is within $\text{diag}/5$ of a previously computed projected point, where diag is the length of the diagonal of the smallest grid cell, we skip this point. Otherwise, we add two new degrees of freedom located at a distance d_Γ of the interface in the normal direction on either side of the interface. We repeat this procedure for all projected points. The new set of degrees of freedom is therefore made up of the original degrees of freedom whose control volume is not crossed by the interface and of the new degrees of freedom added next to the interface. This constitutes the set of seeds for the Voronoi diagram computational mesh on which we perform the computations. Each Voronoi cell can then be generated independently based on the local neighborhood of each degree of freedom, making the generation of the Voronoi mesh embarrassingly parallel.

Note that all the new degrees of freedom are placed at the same distance d_Γ from the interface. This is a free parameter of our method, and experimenting with various reasonable values shows little impact on the numerical results. We choose $d_\Gamma = \text{diag}/5$. This simple procedure will be shown in Sections 4.1 and 4.2 to be sufficient to construct second-order accurate solutions in the L^∞ norm.

2.3. Smoothing the mesh

The algorithm described in the previous section can lead to undesirable geometrical configurations in the case when the interface is not sufficiently resolved. Fig. 3 presents one such configuration. The control volumes of some of the degrees of freedom are connected by a face that is not capturing properly the interface.

It is possible to remediate this issue by modifying the Voronoi partition in a post-processing step. The control volume of any degree of freedom that has been added next to the interface should be connected to exactly one control volume associated to a degree of freedom on the other side of the irregular interface. Consequently, if more than one neighbor is found across the interface, we disconnect the undesired ones by removing the connecting edge as shown in Fig. 3. The edge and its two associated vertices are then replaced by a single vertex located in the middle of the removed edge. The control volume of all the degrees of freedom of the resulting mesh are connected to at most one control volume on the other side of the interface and the interface is captured properly.

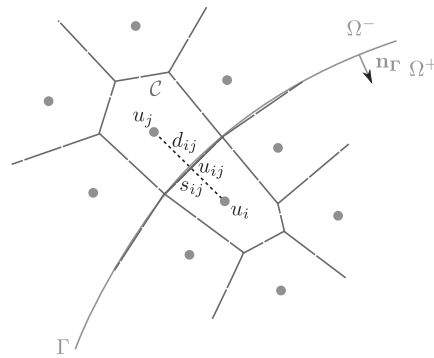


Fig. 4. Nomenclature for the finite volume discretization for the degree of freedom i . For each neighboring degree of freedom j , we call d_{ij} the distance between i and j , s_{ij} the length of the edge (or surface of the polygon in three spatial dimensions) connecting i and j , and u_{ij} the value of u at the middle of the segment $[i, j]$. Note that by construction u_{ij} can be considered to be exactly on the irregular interface Γ , in which case we define u_{ij}^+ and u_{ij}^- .

However, this procedure alters the mesh which is no longer a Voronoi diagram, and the edge between two degrees of freedom is not guaranteed to be orthogonal to the line connecting the two degrees of freedom. The impact of this post-processing algorithm is analyzed in Sections 4.1 and 4.2 and does not seem to improve the method, we therefore recommend not using it.

2.4. Interpolating back to the original mesh

In general, if solving a diffusion equation with discontinuities is part of a larger solver, it is necessary to interpolate the solution from the Voronoi mesh back to the original mesh. This is an easy task given some basic bookkeeping information linking the original degrees of freedom to the ones generated for the Voronoi mesh. With this information, the solution on the Voronoi mesh can be accessed for the same cost than accessing the data on the original mesh. The algorithm to interpolate the solution at a given point (x, y) from the Voronoi mesh is then as follows:

1. locate the cell of the original mesh containing (x, y) ,
2. using the bookkeeping information, identify the Voronoi degree of freedom $v(x, y)$ closest to (x, y) ,
3. find the two neighbors of $v(x, y)$ closest to (x, y) and on the same side of the interface,
4. compute the multilinear interpolation of the solution using those three degrees of freedom and evaluate it at (x, y) .

This simple procedure produces a second-order interpolation at any given point (x, y) .

3. Solving a Poisson equation on Voronoi diagrams

We discretize equation (1) with a finite volume approach on the Voronoi diagram introduced in Section 2.2. We use the notations from Fig. 4. We consider the degree of freedom i with the set of Voronoi neighbors $\{j\}$. Applying a finite volume approach to the problem at i , we can write

$$\int_C \underline{\nabla} \cdot (\beta \underline{\nabla} u) dV = \int_{\partial C} (\beta \underline{\nabla} u) \cdot \underline{n}_C dl \approx \sum_j s_{ij} \beta_i \frac{u_{ij} - u_i}{d_{ij}/2},$$

where \underline{n}_C is the outer normal to the face of C connecting i and j , s_{ij} is the length of that face (or area of the surface in three spatial dimensions) and d_{ij} is the distance between the degrees of freedom i and j . For the case when i and j are on either side of the interface with $\phi_i > 0$, where ϕ_i is the value of the level-set function at the degree of freedom i , we can match the flux at the irregular interface as follows,

$$s_{ij} \beta_i \frac{u_{ij}^+ - u_i}{d_{ij}/2} = s_{ij} \beta_j \frac{u_j - u_{ij}^-}{d_{ij}/2} - s_{ij} [\beta \underline{\nabla} u \cdot \underline{n}_\Gamma].$$

We also know that $u_{ij}^+ = u_{ij}^- + [u]$. Injecting this into the previous expression gives

$$\begin{aligned} s_{ij} \beta_i \frac{u_{ij}^+ - u_i}{d_{ij}/2} &= s_{ij} \beta_j \frac{u_j - u_{ij}^+ + [u]}{d_{ij}/2} - s_{ij} [\beta \underline{\nabla} u \cdot \underline{n}_\Gamma] \\ \Leftrightarrow u_{ij}^+ (\beta_i + \beta_j) &= \beta_j u_j + \beta_i u_i + \beta_j [u] - \frac{d_{ij}}{2} [\beta \underline{\nabla} u \cdot \underline{n}_\Gamma] \\ \Leftrightarrow u_{ij}^+ &= \frac{1}{\beta_i + \beta_j} \left(\beta_j u_j + \beta_i u_i + \beta_j [u] - \frac{d_{ij}}{2} [\beta \underline{\nabla} u \cdot \underline{n}_\Gamma] \right). \end{aligned}$$

In the case when i and j are on the same side of the interface, the derivation is the same but the contributions from the discontinuities vanish. The contribution from the interaction between the degrees of freedom i and j , in the case when $\phi_i > 0$, to the finite volume discretization of (1) can then be written

$$\begin{aligned} \beta_i s_{ij} \frac{u_{ij}^+ - u_i}{d_{ij}/2} &= \frac{2\beta_i}{\beta_i + \beta_j} \frac{s_{ij}}{d_{ij}} \left(\beta_j u_j + \beta_i u_i - (\beta_i + \beta_j) u_i + \beta_j [u] - \frac{d_{ij}}{2} [\beta \nabla u \cdot \mathbf{n}_\Gamma] \right) \\ &= \frac{2\beta_i \beta_j}{\beta_i + \beta_j} \frac{s_{ij}}{d_{ij}} \frac{u_j - u_i}{d_{ij}} - \frac{2\beta_i \beta_j}{\beta_i + \beta_j} \frac{s_{ij}}{d_{ij}} \left(-[u] + \frac{d_{ij}}{2\beta_j} [\beta \nabla u \cdot \mathbf{n}_\Gamma] \right) \\ &= \tilde{\beta}_{ij} s_{ij} \frac{u_j - u_i}{d_{ij}} - \tilde{\beta}_{ij} \frac{s_{ij}}{d_{ij}} \left(-[u] + \frac{d_{ij}}{2\beta_j} [\beta \nabla u \cdot \mathbf{n}_\Gamma] \right), \end{aligned}$$

where $\tilde{\beta}_{ij}$ is the harmonic mean between β_i and β_j , i.e.

$$\tilde{\beta}_{ij} = \frac{|\phi_i| + |\phi_j|}{|\phi_i|/\beta_i + |\phi_j|/\beta_j} = \frac{2\beta_i \beta_j}{\beta_i + \beta_j}.$$

The contribution of the interaction between the degrees of freedom i and j to the linear system is therefore

$$\tilde{\beta}_{ij} s_{ij} \frac{u_j - u_i}{d_{ij}},$$

while the contribution to the right-hand side is

$$\tilde{\beta}_{ij} \frac{s_{ij}}{d_{ij}} \left(-[u] + \frac{d_{ij}}{2\beta_j} [\beta \nabla u \cdot \mathbf{n}_\Gamma] \right).$$

Note that we made use of the fact that ϕ is a distance function and u_{ij} is midway between i and j to simplify the expression. Similarly, we can derive the contribution of the interaction between i and j for the case when $\phi_i < 0$ and obtain the general expression for the interaction between any degrees of freedom i and j

$$\tilde{\beta}_{ij} s_{ij} \frac{u_j - u_i}{d_{ij}} + \text{Vol}(\mathcal{C}) \cdot k_i \cdot u_i = \tilde{\beta}_{ij} \frac{s_{ij}}{d_{ij}} \left(-\text{sign}(\phi_i)[u] + \frac{d_{ij}}{2\beta_j} [\beta \nabla u \cdot \mathbf{n}_\Gamma] \right) + \text{Vol}(\mathcal{C}) \cdot f_i,$$

where $\text{Vol}(\mathcal{C})$ is the volume of the Voronoi cell associated to the degree of freedom i and k_i and f_i are the respective values of k and f at the degree of freedom i . This discretization is entirely implicit and leads to a symmetric positive definite matrix. The discontinuities contribute only to the right-hand side of the linear system. Note that this formulation is identical to the Ghost Fluid Method of [41] in the case where the irregular interface is orthogonal to the flux between the two degrees of freedom and located midway. In fact, the Voronoi Interface Method can be interpreted as a Ghost Fluid Method where the flux between two degrees of freedom is guaranteed to be orthogonal to the face connecting their respective control volumes. We solve the linear system with the Conjugate Gradient iterative solver provided by the Petsc libraries [5,6] and preconditioned with the Hypre multigrid [20]. We enforce Dirichlet boundary conditions on $\partial\Omega$.

4. Numerical validation on uniform meshes

In this section, we validate the addition of the degrees of freedom along the irregular interface and analyze the convergence of our method on various examples. In order to demonstrate that our solver captures the discontinuities properly, all the results in this section are presented on meshes that are uniform away from the interface. Doing so, we make sure that the error on the interface dominates the overall error. Since the degrees of freedom are the seeds of the Voronoi cells, it is convenient to compute the gradient of the solution at every point located in the middle of two degrees of freedom, i.e. $\nabla u_{ij} \cdot \mathbf{n}_{ij} = (u_j - u_i)/d_{ij}$ where \mathbf{n}_{ij} is the normal to the edge of the Voronoi cell connecting the degrees of freedom i and j . The errors presented are normalized.

4.1. Validation of the construction of the Voronoi diagrams close to the interface

In this first example, we are interested in the influence of the quality of the mesh close to the interface. We consider three different possibilities, represented in Fig. 5 for a circular irregular interface described by $\phi(x, y) = -\sqrt{x^2 + y^2} + r_0$, with $r_0 = 0.5$, in a domain $\Omega = [-1, 1]^2$.

For the first case, we place the new degrees of freedom at regular intervals on the irregular interface, making use of the explicit parametric expression available for a circle. We choose to place $N = 1.5 \lfloor \frac{2\pi r_0}{\min(x_{\min}, y_{\min})} \rfloor$ new degrees of freedom on either side of the interface, at a distance $\text{diag}/5$ from the interface with $\text{diag} = \sqrt{x_{\min}^2 + y_{\min}^2}$. For the second case, we place the new degrees of freedom according to the procedure described in Section 2.2, at a distance $\frac{\text{diag}}{5}$ from the

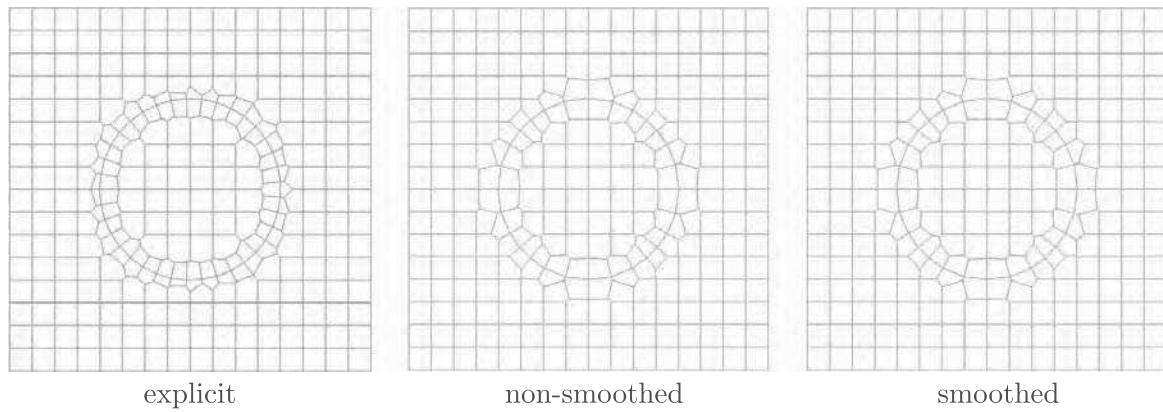


Fig. 5. Visualization of the three different meshes on a resolution $2^4 \times 2^4$ for Section 4.1.

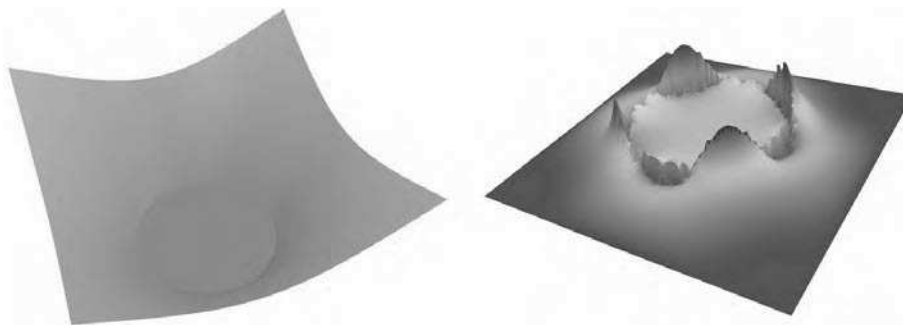


Fig. 6. Left: representation of the solution for Example 4.1. Right: visualization of the localization of the error on a non-smoothed mesh (case 2) of resolution $2^7 \times 2^7$.

Table 1

Convergence of the error on the solution in the L^∞ norm for Example 4.1. The first case corresponds to the degrees of freedom placed along the interface using the explicit parametrization, the second case corresponds to the mesh obtained following the method described in Section 2.2, and the third case is the smoothed version of case 2.

Resolution	Explicit		Non-smoothed		Smoothed	
	Solution	Order	Solution	Order	Solution	Order
2^3	$3.66 \cdot 10^{-3}$	–	$1.27 \cdot 10^{-2}$	–	$1.20 \cdot 10^{-2}$	–
2^4	$1.79 \cdot 10^{-3}$	1.03	$2.34 \cdot 10^{-3}$	2.44	$2.20 \cdot 10^{-3}$	2.45
2^5	$5.77 \cdot 10^{-4}$	1.63	$6.17 \cdot 10^{-4}$	1.92	$6.02 \cdot 10^{-4}$	1.87
2^6	$1.56 \cdot 10^{-4}$	1.89	$1.62 \cdot 10^{-4}$	1.93	$1.61 \cdot 10^{-4}$	1.91
2^7	$4.32 \cdot 10^{-5}$	1.86	$4.45 \cdot 10^{-5}$	1.87	$4.23 \cdot 10^{-5}$	1.86
2^8	$1.13 \cdot 10^{-5}$	1.94	$1.14 \cdot 10^{-5}$	1.97	$1.14 \cdot 10^{-5}$	1.96
2^9	$2.83 \cdot 10^{-6}$	1.99	$2.96 \cdot 10^{-6}$	1.95	$2.95 \cdot 10^{-6}$	1.94
2^{10}	$7.19 \cdot 10^{-7}$	1.98	$7.46 \cdot 10^{-7}$	1.99	$7.45 \cdot 10^{-7}$	1.98

interface. Finally, for the third case, we start from the partition obtained with the second case and modify it according to the procedure described in Section 2.3 to obtain a smoothed mesh.

We monitor the convergence of our method on these three meshes for the following solution taken from [64],

$$u(x, y) = \begin{cases} 1 + \log(2\sqrt{x^2 + y^2}) & \text{if } \phi(x, y) < 0, \\ 1 & \text{if } \phi(x, y) > 0, \end{cases}$$

and $\beta(x, y) = 1$. Note that for this case we have $[u] = 0$ and $[\nabla u \cdot \mathbf{n}] = 2$, with continuous β and a discontinuity in the flux across the interface. A representation of the solution is given in Fig. 6 together with a visualization of the localization of the error. We report the convergence of the solver on this example for the three different meshes in Tables 1 and 2. We observe second-order convergence for the solution and first-order convergence for the gradient of the solution, and very similar errors for all three meshes. We conclude that smoothing the mesh obtained with the procedure explained in Section 2.3 does not seem to improve the accuracy of the solver.

4.2. Influence of the smoothing of the mesh

We further consider the influence of the smoothing procedure described in Section 2.3. This time, we consider an interface described by $\phi(x, y) = -\sqrt{x^2 + y^2} + r_0 + r_1 \cos(5\theta)$, with $r_0 = 0.5$, $r_1 = 0.15$ and θ the angle between (x, y) and $(1, 0)$,

Table 2

Convergence of the error on the gradient of the solution in the L^∞ norm for Example 4.1. The first case corresponds to the degrees of freedom placed along the interface using the explicit parametrization, the second case corresponds to the mesh obtained following the method described in Section 2.2, and the third case corresponds to its smoothed version.

Resolution	Explicit		Non-smoothed		Smoothed	
	Gradient	Order	Gradient	Order	Gradient	Order
2^3	$4.60 \cdot 10^{-2}$	–	$4.71 \cdot 10^{-2}$	–	$4.24 \cdot 10^{-2}$	–
2^4	$3.28 \cdot 10^{-2}$	0.49	$2.40 \cdot 10^{-2}$	0.98	$2.36 \cdot 10^{-2}$	0.84
2^5	$1.79 \cdot 10^{-2}$	0.87	$1.23 \cdot 10^{-2}$	0.96	$1.23 \cdot 10^{-2}$	0.94
2^6	$9.56 \cdot 10^{-3}$	0.91	$8.34 \cdot 10^{-3}$	0.56	$8.35 \cdot 10^{-3}$	0.56
2^7	$5.15 \cdot 10^{-3}$	0.89	$3.94 \cdot 10^{-3}$	1.08	$3.94 \cdot 10^{-3}$	1.08
2^8	$2.56 \cdot 10^{-3}$	1.01	$2.12 \cdot 10^{-3}$	0.90	$2.12 \cdot 10^{-3}$	0.90
2^9	$1.30 \cdot 10^{-3}$	0.98	$1.12 \cdot 10^{-3}$	0.92	$1.12 \cdot 10^{-3}$	0.92
2^{10}	$6.61 \cdot 10^{-4}$	0.98	$5.61 \cdot 10^{-4}$	1.00	$5.61 \cdot 10^{-4}$	1.00

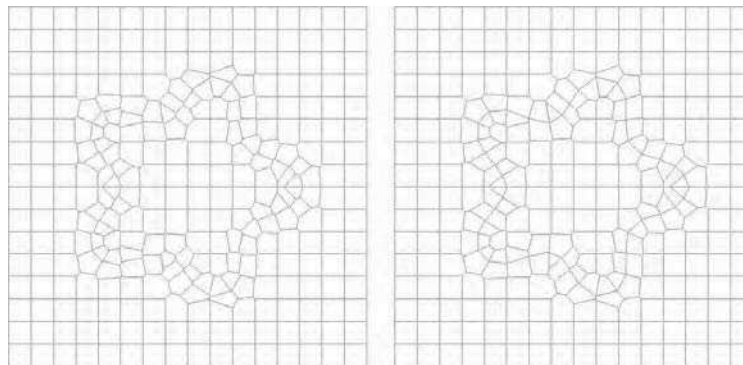


Fig. 7. Visualization of the non-smoothed (left) and smoothed (right) meshes on a resolution $2^4 \times 2^4$ for Section 4.2.

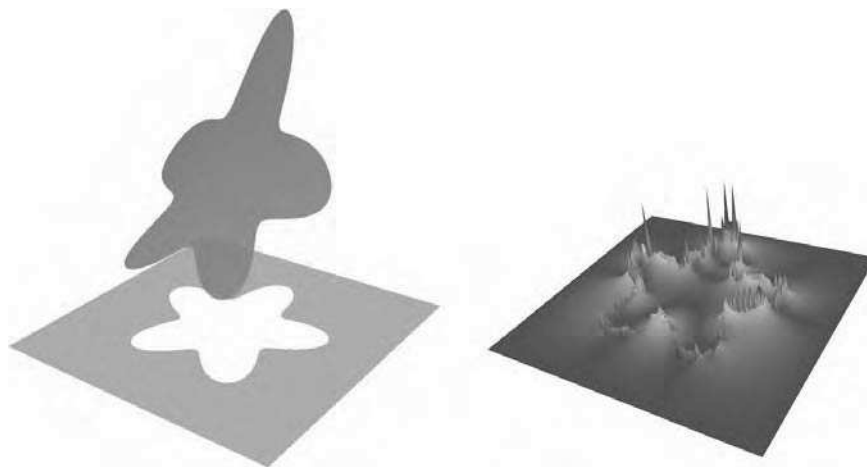


Fig. 8. Left: representation of the solution for Example 4.2. Right: visualization of the error on a non-smoothed mesh and for a resolution $2^7 \times 2^7$.

in a domain $\Omega = [-1, 1]^2$. Since we do not have an explicit parametrization of the interface that would enable to place the degrees of freedom at regular intervals, we only consider the mesh generated from the procedure described in Section 2.2 and its smoothed version obtained by applying the procedure described in Section 2.3. Fig. 7 gives a visualization of the meshes obtained.

For this section, we choose to work with the exact solution taken from [9]

$$u(x, y) = \begin{cases} 0 & \text{if } \phi(x, y) < 0, \\ e^x \cos(y) & \text{if } \phi(x, y) > 0, \end{cases}$$

with $\beta^- = \beta^+ = 1$. The solution is represented in Fig. 8. We monitor the convergence of the solver in Table 3 and observe second-order convergence for the solution and first-order convergence for the gradient of the solution in both cases. Given that the smoothing algorithm requires additional processing and does not seem to improve the accuracy (in fact, we notice for this particular example that the non-smoothed results are more accurate), we choose to work with the non-smoothed mesh constructed as described in Section 2.2 for the remaining of this article.

Table 3
Convergence of the error on the solution and its gradient in the L^∞ norm for Example 4.2.

Resolution	Non-smoothed				Smoothed			
	Solution	Order	Gradient	Order	Solution	Order	Gradient	Order
2^3	$2.39 \cdot 10^{-3}$	–	$3.01 \cdot 10^{-2}$	–	$2.00 \cdot 10^{-2}$	–	$1.06 \cdot 10^{-1}$	–
2^4	$1.06 \cdot 10^{-3}$	1.17	$5.08 \cdot 10^{-1}$	–4.07	$1.44 \cdot 10^{-2}$	0.47	$5.27 \cdot 10^{-1}$	–2.32
2^5	$3.43 \cdot 10^{-4}$	1.63	$8.42 \cdot 10^{-3}$	5.91	$4.21 \cdot 10^{-3}$	1.78	$2.84 \cdot 10^{-2}$	4.21
2^6	$6.82 \cdot 10^{-5}$	2.33	$3.95 \cdot 10^{-3}$	1.09	$1.62 \cdot 10^{-3}$	1.38	$1.66 \cdot 10^{-2}$	0.78
2^7	$2.79 \cdot 10^{-5}$	1.29	$2.93 \cdot 10^{-3}$	0.43	$4.55 \cdot 10^{-4}$	1.83	$8.03 \cdot 10^{-3}$	1.04
2^8	$7.35 \cdot 10^{-6}$	1.92	$1.40 \cdot 10^{-3}$	1.06	$1.11 \cdot 10^{-4}$	2.03	$2.66 \cdot 10^{-3}$	1.59
2^9	$1.89 \cdot 10^{-6}$	1.96	$6.31 \cdot 10^{-4}$	1.15	$2.76 \cdot 10^{-5}$	2.01	$9.62 \cdot 10^{-4}$	1.47
2^{10}	$4.75 \cdot 10^{-7}$	1.99	$3.28 \cdot 10^{-4}$	0.94	$6.86 \cdot 10^{-6}$	2.01	$3.34 \cdot 10^{-4}$	1.53

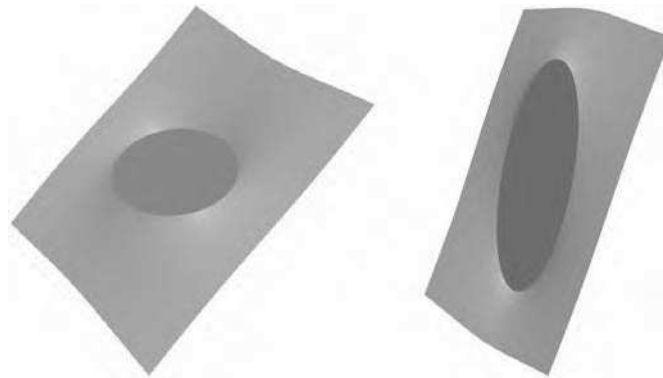


Fig. 9. Two examples of solutions for Example 4.3. Left: $\beta^- = 1$ and $\beta^+ = 10$. Right: $\beta^- = 10$ and $\beta^+ = 1$.

4.3. Example with a discontinuity in the diffusion coefficient

We now consider the exact solution

$$u(x, y) = \begin{cases} \frac{x(\rho+1) - x(\rho-1)r_0^2/r^2}{\rho+1+r_0^2(\rho-1)} & \text{if } \phi(x, y) < 0, \\ \frac{2x}{\rho+1+r_0^2(\rho-1)} & \text{if } \phi(x, y) > 0, \end{cases}$$

with $r_0 = .5$, $r = \sqrt{x^2 + y^2}$, $\phi(x, y) = -r^2 + r_0^2$ and $\rho = \beta^+/\beta^-$ in the domain $\Omega = [-1, 1]^2$. This corresponds to Example 7.3 from [64]. In this case, u is continuous, but the diffusion coefficient β experiences a large jump across the irregular interface Γ . We also have $[\underline{\nabla}u \cdot \underline{n}] \neq 0$. A visualization of the solution is given in Fig. 9. The gradient of the solution is given by

$$\underline{\nabla}u(x, y) = \begin{cases} \frac{1}{\rho+1+r_0^2(\rho-1)} \begin{pmatrix} \rho + 1 - r_0^2(\rho - 1) \frac{y^2 - x^2}{(x^2 + y^2)^2} \\ r_0^2(\rho - 1) \frac{2xy}{(x^2 + y^2)^2} \end{pmatrix} & \text{if } \phi(x, y) < 0, \\ \frac{1}{\rho+1+r_0^2(\rho-1)} \begin{pmatrix} 2 \\ 0 \end{pmatrix} & \text{if } \phi(x, y) > 0. \end{cases}$$

The errors on the solution and its gradient are monitored in Table 4 which shows second-order convergence on the solution and first-order convergence on its gradient. Fig. 10 provides a visualization of the localization of the error. We also monitor the evolution of the 1-norm condition number of the matrix of the linear system as the mesh is refined in and present the results in Table 5. The condition number depends on the mesh resolution and on the diffusion coefficient. When the diffusion coefficient is large, the condition number gets large rapidly. However, the discontinuities at the interface are entirely captured by the right hand side and therefore do not affect the conditioning of the matrix.

4.4. A complete example

This example is meant to test our method to its full capacity, with discontinuities in all four quantities (the solution, its gradient, the diffusion coefficient and the flux across the interface), and with a complex irregular interface. We choose the exact solution

$$u(x, y) = \begin{cases} e^x & \text{if } \phi(x, y) < 0, \\ \cos(x) \sin(y) & \text{if } \phi(x, y) > 0, \end{cases}$$

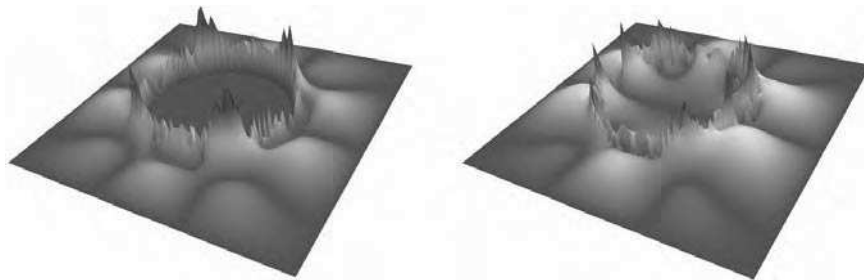
Table 4

Convergence on the solution and its gradient for Example 4.3, for two different combinations of diffusion coefficients.

Resolution	$\beta^- = 1, \beta^+ = 10^5$				$\beta^- = 10^5, \beta^+ = 1$			
	Solution	Order	Gradient	Order	Solution	Order	Gradient	Order
2^3	$1.10 \cdot 10^{-2}$	–	$3.84 \cdot 10^{-2}$	–	$1.74 \cdot 10^{-2}$	–	$1.14 \cdot 10^{-1}$	–
2^4	$3.42 \cdot 10^{-3}$	1.68	$1.93 \cdot 10^{-2}$	1.00	$4.70 \cdot 10^{-3}$	1.89	$5.65 \cdot 10^{-2}$	1.01
2^5	$1.39 \cdot 10^{-3}$	1.30	$1.17 \cdot 10^{-2}$	0.72	$1.26 \cdot 10^{-3}$	1.90	$2.54 \cdot 10^{-2}$	1.15
2^6	$3.82 \cdot 10^{-4}$	1.86	$6.19 \cdot 10^{-3}$	0.92	$3.39 \cdot 10^{-4}$	1.90	$1.77 \cdot 10^{-2}$	0.53
2^7	$1.34 \cdot 10^{-4}$	1.51	$3.59 \cdot 10^{-3}$	0.79	$1.05 \cdot 10^{-4}$	1.69	$8.06 \cdot 10^{-3}$	1.13
2^8	$3.43 \cdot 10^{-5}$	1.97	$1.84 \cdot 10^{-3}$	0.96	$2.67 \cdot 10^{-5}$	1.98	$4.87 \cdot 10^{-3}$	0.73
2^9	$9.76 \cdot 10^{-6}$	1.81	$1.05 \cdot 10^{-3}$	0.82	$7.55 \cdot 10^{-6}$	1.82	$2.40 \cdot 10^{-3}$	1.02
2^{10}	$2.58 \cdot 10^{-6}$	1.92	$5.41 \cdot 10^{-4}$	0.95	$1.96 \cdot 10^{-6}$	1.95	$1.25 \cdot 10^{-3}$	0.94

Table 5Evolution of the condition number as the mesh is refined for Example 4.3. The condition number depends solely on the diffusion coefficient β and on the resolution on the mesh since the discontinuities are captured by the right hand side of the linear system.

Resolution	$\beta^- = 1, \beta^+ = 10^5$	$\beta^- = 10^5, \beta^+ = 1$	$\beta^- = \beta^+ = 10^5$	$\beta^- = \beta^+ = 1$
2^3	$2.38 \cdot 10^6$	$2.02 \cdot 10^6$	$5.59 \cdot 10^6$	$1.72 \cdot 10^2$
2^4	$1.17 \cdot 10^7$	$5.09 \cdot 10^6$	$1.23 \cdot 10^7$	$8.11 \cdot 10^2$
2^5	$4.86 \cdot 10^7$	$1.57 \cdot 10^7$	$2.69 \cdot 10^7$	$3.66 \cdot 10^3$
2^6	$2.09 \cdot 10^8$	$7.18 \cdot 10^7$	$5.72 \cdot 10^7$	$1.58 \cdot 10^4$
2^7	$8.52 \cdot 10^8$	$3.14 \cdot 10^8$	$1.19 \cdot 10^8$	$6.59 \cdot 10^4$
2^8	$3.86 \cdot 10^9$	$1.32 \cdot 10^9$	$2.42 \cdot 10^8$	$2.70 \cdot 10^5$
2^9	$1.48 \cdot 10^{10}$	$5.43 \cdot 10^9$	$4.89 \cdot 10^8$	$1.09 \cdot 10^6$
2^{10}	$6.59 \cdot 10^{10}$	$2.20 \cdot 10^{10}$	$9.84 \cdot 10^8$	$4.39 \cdot 10^6$

**Fig. 10.** Visualization of the localization of the error in the L^∞ norm on a grid of resolution $2^7 \times 2^7$ for Example 4.3. Left: $\beta^- = 1$ and $\beta^+ = 10^5$. Right: $\beta^- = 10^5$ and $\beta^+ = 1$.

in a domain $\Omega = [-1, 1]^2$ with $\phi(x, y) = -\sqrt{x^2 + y^2} + r_0 + r_1 \cos(n\theta)$, where $r_0 = 0.5$, $r_1 = 0.15$ and $n = 5$, and we define the diffusion coefficient as

$$\beta(x, y) = \begin{cases} y^2 \ln(x+2) + 4 & \text{if } \phi(x, y) < 0, \\ e^{-y} & \text{if } \phi(x, y) > 0. \end{cases}$$

Note that with our method, each degree of freedom has a control volume that is entirely on one side of the irregular interface, and therefore we can easily define a forcing term for any analytical solution. The exact solution and the diffusion coefficient are represented in Fig. 11. The convergence is summarized in Table 6 and once again indicates second-order convergence for the solution and first-order convergence for the gradient of the solution. A visualization of the localization of the error on the solution is given in Fig. 12.

Fig. 13 presents the percentage of the runtime consumed by the four principal components of the algorithm, i.e. constructing the Voronoi mesh, assembling the matrix, computing the right hand side and solving the linear system. For coarse grids, the bottleneck of the computation is the construction of the Voronoi mesh, but for high resolution we observe that inverting the linear system is the costliest. These results correspond to our implementation in the absence of parallelization. The finest resolution of 1024×1024 takes 18 s on a single core of an Intel i7-2600 3.40 GHz cpu.

4.5. Adding subdomains

We now propose an example with multiple subdomains. Note that the method we propose leads naturally to a linear system with N rows, the number of degrees of freedom. This number increases slightly as the number of subdomains increases and additional degrees of freedom are added next to the irregular interfaces.

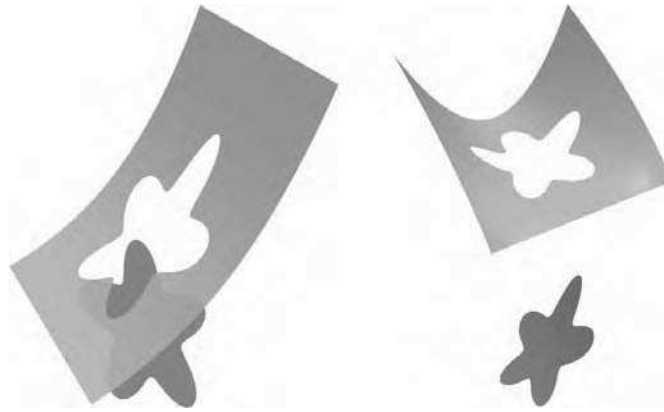


Fig. 11. Left: visualization of the solution u for Example 4.4. Right: visualization of the diffusion coefficient β .

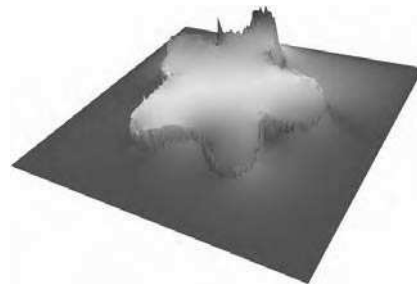


Fig. 12. Visualization of the localization of the error for Example 4.4 on a resolution of $2^7 \times 2^7$.

Table 6
Convergence on the solution and its gradient for Example 4.4.

Resolution	Solution	Order	Gradient	Order
2^3	$3.97 \cdot 10^{-3}$	–	$4.37 \cdot 10^{-1}$	–
2^4	$9.98 \cdot 10^{-4}$	1.99	$5.01 \cdot 10^{-1}$	–0.20
2^5	$2.90 \cdot 10^{-4}$	1.78	$3.67 \cdot 10^{-3}$	7.09
2^6	$8.84 \cdot 10^{-5}$	1.71	$1.73 \cdot 10^{-3}$	1.09
2^7	$2.06 \cdot 10^{-5}$	2.10	$9.84 \cdot 10^{-4}$	0.81
2^8	$5.22 \cdot 10^{-6}$	1.98	$4.77 \cdot 10^{-4}$	1.05
2^9	$1.33 \cdot 10^{-6}$	1.97	$2.45 \cdot 10^{-4}$	0.96
2^{10}	$3.39 \cdot 10^{-7}$	1.97	$1.23 \cdot 10^{-4}$	0.99

For simplicity, we choose to work with non-intersecting irregular interfaces, but our method is suited for any general configuration. We divide the computational domain $\Omega = [-1, 1]^2$ in 4 subdomains represented in Fig. 14 and separated by the three contours defined by

$$\Gamma^0 = \left\{ (x, y), \phi^0(x, y) = \sqrt{x^2 + y^2} - 0.2 \right\},$$

$$\Gamma^1 = \left\{ (x, y), \phi^1(x, y) = \sqrt{x^2 + y^2} - 0.5 + 0.1 \cos(5\theta) \right\},$$

$$\Gamma^2 = \left\{ (x, y), \phi^2(x, y) = \sqrt{x^2 + y^2} - 0.8 \right\},$$

where θ is the angle between (x, y) and the x -axis. We choose the exact solution

$$u(x, y) = \begin{cases} e^x + 1.3 & \text{if } (x, y) \in \Omega^0, \\ \cos(y) + 1.8 & \text{if } (x, y) \in \Omega^1, \\ \sin(x) + 0.5 & \text{if } (x, y) \in \Omega^2, \\ -x + \ln(y + 2) & \text{if } (x, y) \in \Omega^3, \end{cases}$$

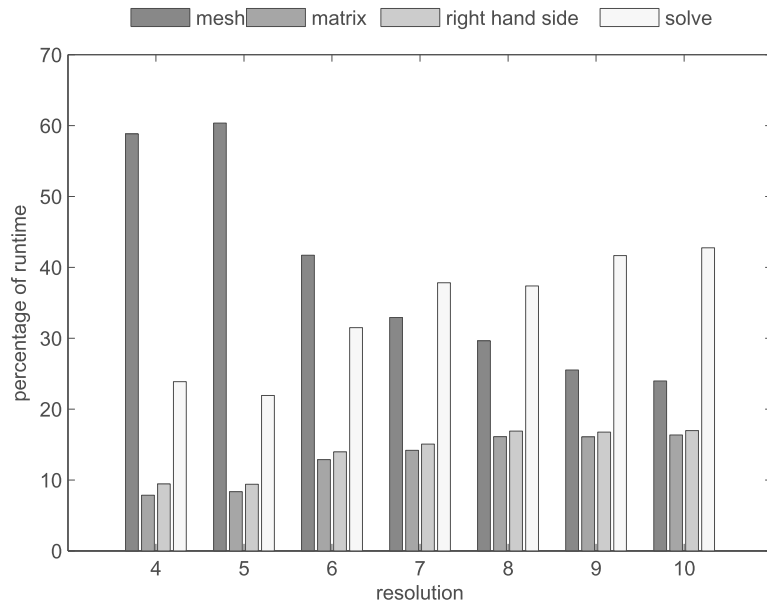


Fig. 13. Representation of the computation time consumed by the four main sections of our implementation, constructing the mesh, assembling the matrix, computing the right hand side and solving the linear system for Example 4.4. For coarse grids, building the Voronoi partition takes the most time, but as the resolution of the grid increases, the inversion of the linear system becomes the costliest.

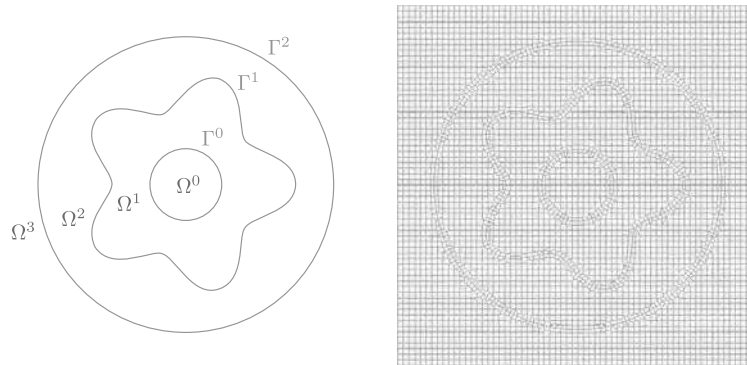


Fig. 14. Illustration of the division of the computational domain into four subdomains, together with the Voronoi mesh generated, for Example 4.5.

and the diffusion coefficient

$$\beta(x, y) = \begin{cases} y^2 + 1 & \text{if } (x, y) \in \Omega^0, \\ e^x & \text{if } (x, y) \in \Omega^1, \\ y + 1 & \text{if } (x, y) \in \Omega^2, \\ x^2 + 1 & \text{if } (x, y) \in \Omega^3. \end{cases}$$

The solution and the diffusion coefficient are represented in Fig. 15.

The convergence of the solver is presented in Table 7. We observe second order convergence for the solution and first order convergence for its gradient.

4.6. Application to three spatial dimensions

We now present an example in three spatial dimensions and with a spherical interface of radius 0.5 in a domain $\Omega = [-1, 1]^3$. We work with the exact solution

$$u(x, y, z) = \begin{cases} e^z & \text{if } \phi(x, y, z) < 0, \\ \cos(x) \sin(y) & \text{if } \phi(x, y, z) > 0, \end{cases}$$

and the diffusion coefficient

$$\beta(x, y, z) = \begin{cases} y^2 \ln(x + 2) + 4 & \text{if } \phi(x, y, z) < 0, \\ e^{-z} & \text{if } \phi(x, y, z) > 0. \end{cases}$$

The geometry, together with a slice of the solution and of the diffusion coefficient, is represented in Fig. 16. Table 8 presents the numerical results and indicates second-order convergence for the solution and first-order convergence for its gradient.

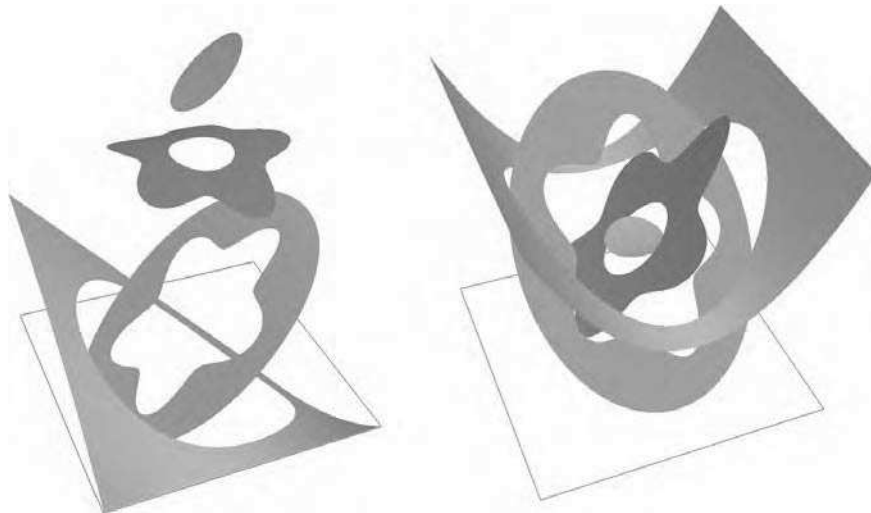


Fig. 15. Visualization of the solution (left) and the diffusion coefficient (right) for Example 4.5.

Table 7

Convergence on the solution and its gradient in the L^∞ norm for Example 4.5.

Resolution	Solution	Order	Gradient	Order
2^4	$1.00 \cdot 10^{-3}$	–	$3.33 \cdot 10^{-3}$	–
2^5	$2.33 \cdot 10^{-4}$	2.11	$1.01 \cdot 10^{-3}$	1.72
2^6	$6.23 \cdot 10^{-5}$	1.90	$3.46 \cdot 10^{-4}$	1.54
2^7	$1.56 \cdot 10^{-5}$	2.00	$1.59 \cdot 10^{-4}$	1.12
2^8	$4.00 \cdot 10^{-6}$	1.96	$6.82 \cdot 10^{-5}$	1.22
2^9	$1.01 \cdot 10^{-6}$	1.99	$4.00 \cdot 10^{-5}$	0.77
2^{10}	$2.55 \cdot 10^{-7}$	1.99	$2.24 \cdot 10^{-5}$	0.84

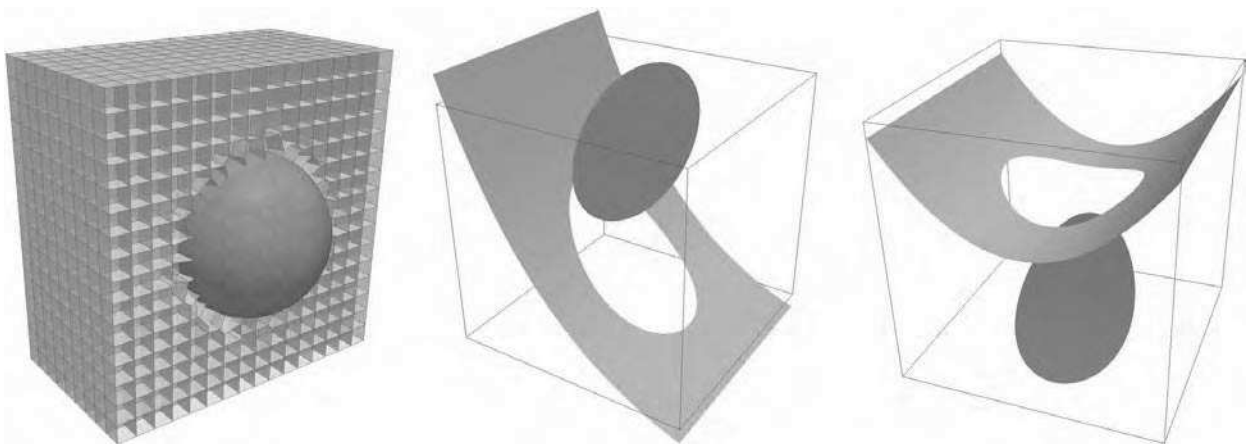


Fig. 16. Left: representation of the irregular interface and the associated Voronoi mesh on a resolution $2^4 \times 2^4 \times 2^4$ for Example 4.6. Center: visualization of the solution on the slice $x = 0$. Right: visualization of the diffusion coefficient on the slice $x = 0$. Note that the surfaces have been translated to facilitate the visualization.

Table 8

Convergence on the solution and its gradient in the L^∞ norm on a sphere (Example 4.6).

Resolution	Solution	Order	Gradient	Order
2^3	$3.61 \cdot 10^{-3}$	–	$1.13 \cdot 10^{-2}$	–
2^4	$1.21 \cdot 10^{-3}$	1.58	$7.69 \cdot 10^{-3}$	0.56
2^5	$3.04 \cdot 10^{-4}$	1.99	$3.83 \cdot 10^{-3}$	1.01
2^6	$7.74 \cdot 10^{-5}$	1.98	$2.43 \cdot 10^{-3}$	0.66
2^7	$1.97 \cdot 10^{-5}$	1.98	$1.24 \cdot 10^{-3}$	0.98

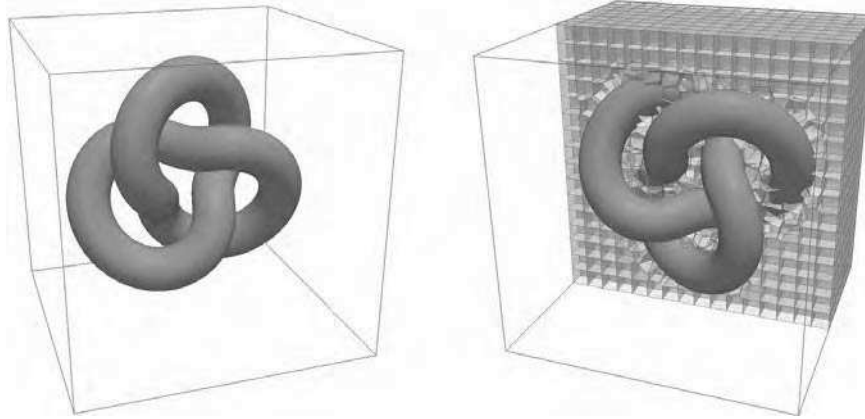


Fig. 17. Representation of the irregular interface and the associated Voronoi mesh on a resolution $2^4 \times 2^4 \times 2^4$ for Example 4.7.

4.7. A complex geometry in three spatial dimensions

For a more complicated geometry, we select the intricate contour borrowed from [34] and parametrized by

$$\Gamma_{\text{trefoil}} = \left\{ \frac{R}{3} \begin{pmatrix} (2 + \cos(3t)) \cos(2t) \\ (2 + \cos(3t)) \sin(2t) \\ \sin(3t) \end{pmatrix}, t \in [0, 2\pi] \right\},$$

where $R = 0.7$ is the major radius of the trefoil. We then define

$$\Omega^+ = \left\{ \mathbf{x} \in \mathbb{R}^3, \min_{\mathbf{y} \in \Gamma_{\text{trefoil}}} \|\mathbf{x} - \mathbf{y}\|_2 < r \right\},$$

with $r = 0.15$ the minor radius of the trefoil. The contour is represented in Fig. 17.

For this example, we use the exact solution

$$u(x, y, z) = \begin{cases} yz \sin(x) & \text{if } \phi(x, y, z) < 0, \\ xy^2 + z^3 & \text{if } \phi(x, y, z) > 0, \end{cases}$$

and the diffusion coefficient

$$\beta(x, y, z) = \begin{cases} y^2 + 1 & \text{if } \phi(x, y, z) < 0, \\ e^{x+z} & \text{if } \phi(x, y, z) > 0. \end{cases}$$

Slices of the solution and of the diffusion coefficient are displayed in Fig. 18. The convergence of our method on this complex irregular interface is reported in Table 9 and once more indicates second order convergence for the solution and first order convergence for its gradient.

4.8. The screened Poisson equation

This example and the following one add a non-zero k to the previous Example 4.7. In this example, we choose $k < 0$ as

$$k(x, y, z) = \begin{cases} -e^x & \text{if } \phi(x, y, z) < 0, \\ -\cos(y) \sin(z) - 2 & \text{if } \phi(x, y, z) > 0. \end{cases}$$

The convergence results are presented in Table 10 and show second order convergence with errors very similar to the ones obtained when $k = 0$.

4.9. The Helmholtz equation case

Our last example on a uniform base mesh is exactly the same than the one from the previous section but for the Helmholtz equation case, i.e. $k(x, y, z) > 0$. We set

$$k(x, y, z) = \begin{cases} e^y & \text{if } \phi(x, y, z) < 0, \\ \cos(x) \sin(z) + 2 & \text{if } \phi(x, y, z) > 0. \end{cases}$$

In this case, the linear system obtained is more complicated to solve because the matrix is no longer diagonally dominant, meaning that the problem is not convex and iterative solvers such as the Conjugate Gradient used so far are not guaranteed to converge. Instead, we solve the linear directly with an LU decomposition. The numerical results are presented in Table 11 and are almost identical to the results from the previous section, illustrating the second order convergence of our method for the Helmholtz equation.

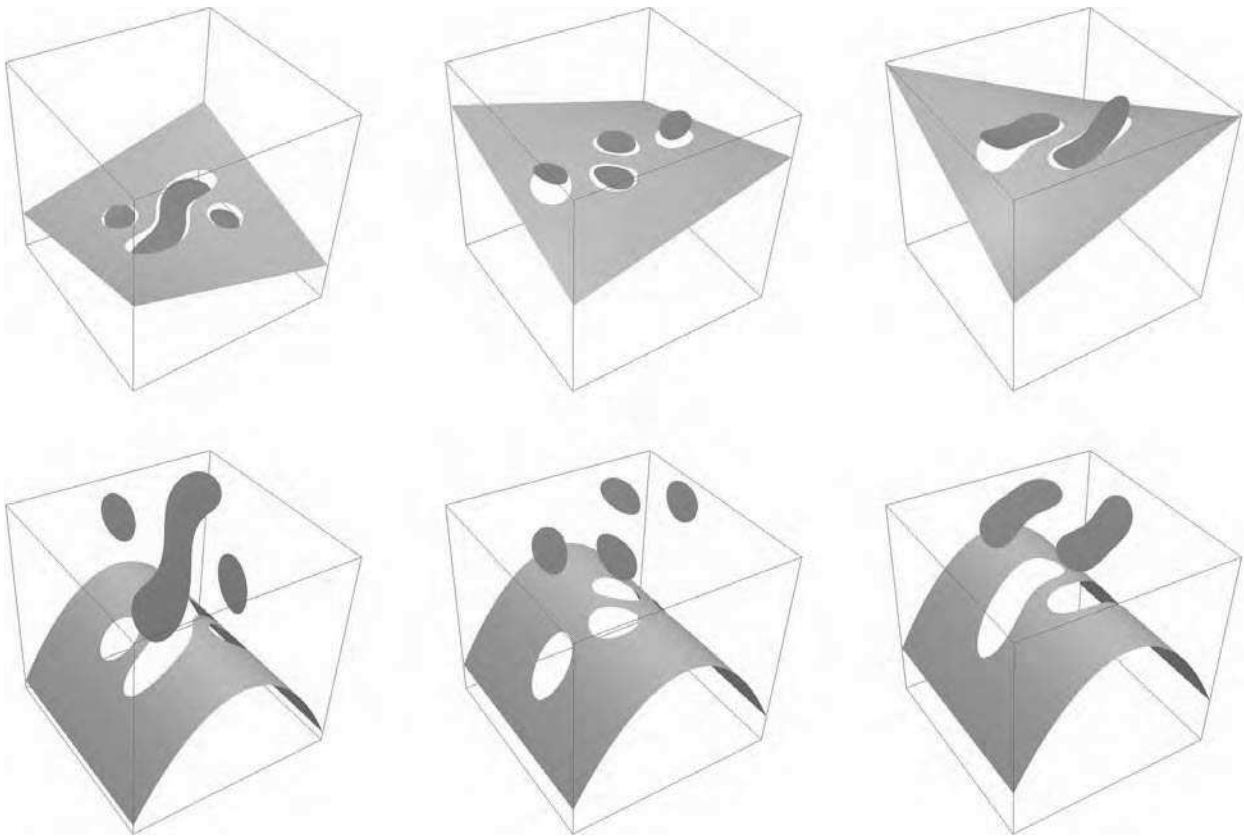


Fig. 18. Visualization of the solution (top row) and the diffusion coefficient (bottom row) on three slices for Example 4.7. The slices are taken, from left to right, at $x = 0.3$, $x = -0.3$ and $x = -0.5$. Note that the surfaces of the diffusion coefficient have been translated to facilitate the visualization.

Table 9

Convergence on the solution and its gradient in the L^∞ norm on a complex three-dimensional contour (Example 4.7).

Resolution	Solution	Order	Gradient	Order
2^4	$3.89 \cdot 10^{-3}$	–	$1.44 \cdot 10^{-1}$	–
2^5	$1.34 \cdot 10^{-3}$	1.54	$2.56 \cdot 10^{-2}$	2.67
2^6	$3.45 \cdot 10^{-4}$	1.96	$9.75 \cdot 10^{-3}$	1.21
2^7	$8.25 \cdot 10^{-5}$	2.07	$5.04 \cdot 10^{-3}$	0.95

Table 10

Convergence on the solution and its gradient in the L^∞ norm on a complex three-dimensional contour for the screened Poisson equation (Example 4.8).

Resolution	Solution	Order	Gradient	Order
2^4	$3.87 \cdot 10^{-3}$	–	$1.44 \cdot 10^{-1}$	–
2^5	$1.34 \cdot 10^{-3}$	1.53	$2.25 \cdot 10^{-2}$	2.67
2^6	$3.44 \cdot 10^{-4}$	1.96	$9.74 \cdot 10^{-3}$	1.21
2^7	$8.22 \cdot 10^{-5}$	2.06	$5.04 \cdot 10^{-3}$	0.95

Table 11

Convergence on the solution and its gradient in the L^∞ norm on a complex three-dimensional contour for the Helmholtz equation (Example 4.9).

Resolution	Solution	Order	Gradient	Order
2^4	$3.91 \cdot 10^{-3}$	–	$1.44 \cdot 10^{-1}$	–
2^5	$1.35 \cdot 10^{-3}$	1.54	$2.26 \cdot 10^{-2}$	2.67
2^6	$3.46 \cdot 10^{-4}$	1.96	$9.75 \cdot 10^{-3}$	1.21
2^7	$8.27 \cdot 10^{-5}$	2.06	$5.05 \cdot 10^{-3}$	0.95

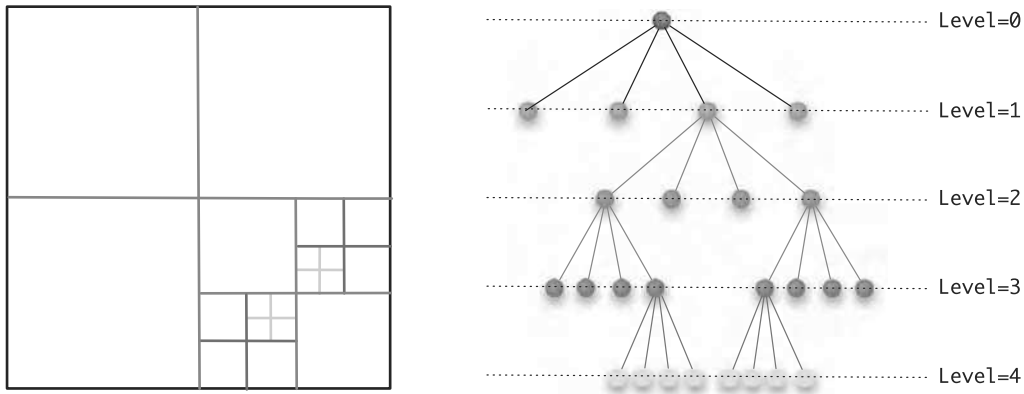


Fig. 19. Example of a Quadtree grid.

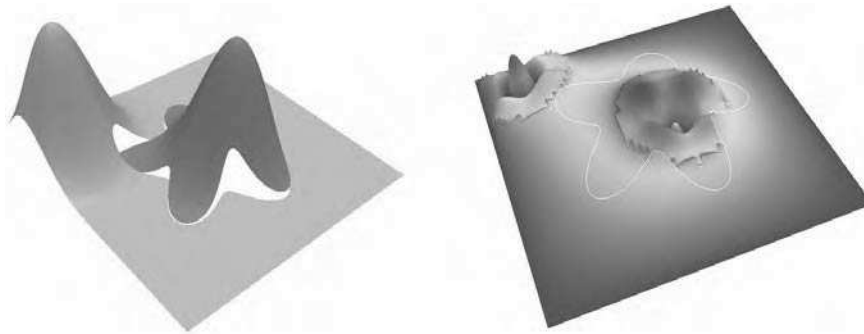


Fig. 20. Left: visualization of the solution for Example 5.2 with $\alpha = 10$. Higher values of α narrow the peaks. Right: representation of the error interpolated on the base Quadtree mesh of level 8/10 for $\alpha = 50$.

5. Extension to adaptive meshes

In the previous section we demonstrated the efficiency of our method based on uniform meshes in both two and three spatial dimensions. However, it can be applied straightforwardly to any mesh and in this section we propose an implementation on Quad/Oc-trees.

5.1. Introduction to the Quad/Oc-tree data structure

A Quad/Oc-tree grid refers to a Cartesian grid that uses the Quad/Oc-tree data structure for its storage in two/three spatial dimensions. Starting from a root cell corresponding to the entire domain, four (eight in three spatial dimensions) children are created if the cell satisfies a given splitting criterion. The process is iterated recursively until the maximum allowed level is reached. The root cell has level 0 and the finest cells have the maximum level allowed. We denote a tree with coarsest level n and finest level m by level n/m . The process is illustrated in Fig. 19. This data structure provides an $O(\ln(n))$ access to the data stored at the leaves. We refer the reader to [45] for further details on the Quad/Oc-tree data structure and the associated discretization techniques.

Since this article is a proof of concept and we know the exact solution in all the numerical example we propose, we make use of this knowledge in the construction of the tree. We will use solutions of the form

$$u(\mathbf{x}) = e^{-\alpha \|\mathbf{x} - \mathbf{x}_0\|_2^2},$$

and any given leaf \mathcal{L} of the tree with center coordinates \mathbf{x}_c is split if $\|\mathbf{x}_0 - \mathbf{x}_c\|_2 < \lambda \cdot \text{diag}$, where diag is the length of the diagonal of \mathcal{L} and λ controls the spread of the mesh around the peaks of the solution.

5.2. Solution on a Quadtree mesh

For this example, we consider the exact solution, represented in Fig. 20,

$$u(x, y) = \begin{cases} e^{-50((x+0.7)^2 + (y-0.7)^2)} & \text{if } \phi(x, y) < 0, \\ e^{-50((x-0.1)^2 + (y+0.1)^2)} & \text{if } \phi(x, y) > 0, \end{cases}$$

with $\beta^- = \beta^+ = 1$ and on the domain $\Omega = [-1, 1]^2$. We use the same geometry as in Section 4.4, i.e. $\phi(x, y) = -\sqrt{x^2 + y^2} + 0.5 + 0.15 \cos(5\theta)$.

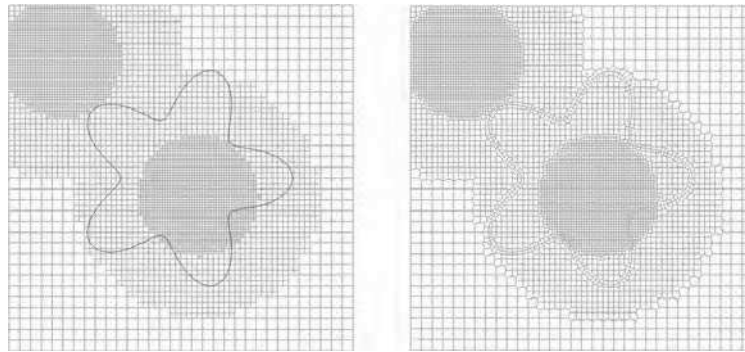


Fig. 21. Left: the initial level 5/7 Quadtree mesh generated for Example 5.2. The irregular interface Γ is represented in red. Right: the corresponding Voronoi mesh. Note that the mesh is not uniform along Γ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 12

Convergence on the solution and its gradient in the L^∞ norm on a Quadtree-based Voronoi mesh (Example 5.2).

Resolution	Solution	Order	Gradient	Order
5/7	$2.56 \cdot 10^{-3}$	–	$8.89 \cdot 10^{-3}$	–
6/8	$6.48 \cdot 10^{-4}$	1.98	$5.14 \cdot 10^{-3}$	0.79
7/9	$1.69 \cdot 10^{-4}$	1.94	$2.52 \cdot 10^{-3}$	1.03
8/10	$4.30 \cdot 10^{-5}$	1.97	$1.33 \cdot 10^{-3}$	0.92
9/11	$1.09 \cdot 10^{-5}$	1.99	$6.84 \cdot 10^{-4}$	0.96

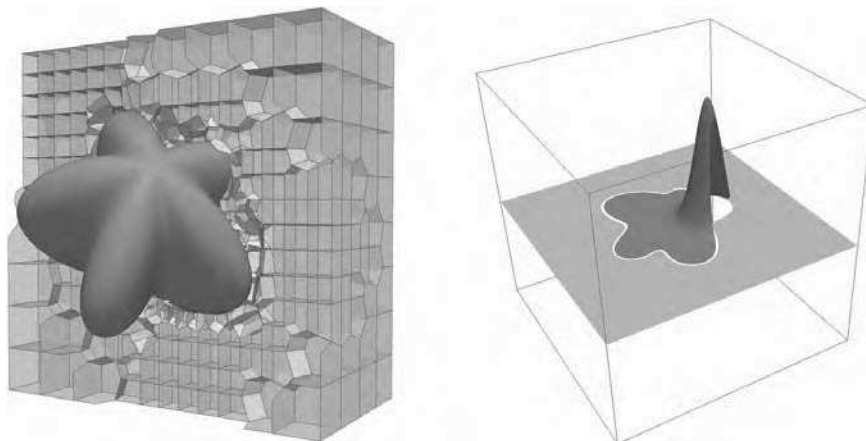


Fig. 22. Left: representation of the geometry together with the Voronoi mesh on the initial mesh of resolution 3/5 for Example 5.3. Note that the mesh is not uniform along Γ . Right: visualization of the solution on the $y = -0.1$ slice and after applying three successive refinement operations to the original mesh.

We start by constructing a mesh of level 5/7 with the criteria described in the previous section and with $\lambda = 8$. A visualization of this initial mesh is given in Fig. 21. Note that the mesh is not uniform close to the interface. We then split every cell of the mesh to monitor the convergence of the solver. The results are presented in Table 12 and show second-order convergence for the solution and first-order convergence for its gradient.

5.3. Solution on an Octree mesh

For this last example, we work with the exact solution

$$u(x, y, z) = \begin{cases} e^{-50((x+0.7)^2+(y-0.7)^2+(z-.5)^2)} & \text{if } \phi(x, y, z) < 0, \\ e^{-50((x-0.1)^2+(y+0.1)^2+(z+.3)^2)} & \text{if } \phi(x, y, z) > 0, \end{cases}$$

and on the irregular interface described by

$$\phi(x, y, z) = -\sqrt{x^2 + y^2 + z^2} + r_0 + r_1 \cos(5\theta) \cos\left(\frac{2\pi}{r_0}z\right),$$

with $r_0 = 0.25$ and $r_1 = r_0/3$, and rotated around the z -axis, y -axis and x -axis by respectively 0.6 radians, 0.2 radians and 0.9 radians. We set $\beta^- = \beta^+ = 1$. The geometry and the Voronoi mesh generated for the initial level 3/5 are represented

Table 13

Convergence on the solution and its gradient in the L^∞ norm on star shaped irregular interface and on an Octree base mesh (Example 5.3).

Resolution	Solution	Order	Gradient	Order
3/5	$6.76 \cdot 10^{-2}$	–	$6.54 \cdot 10^{-2}$	–
4/6	$1.47 \cdot 10^{-2}$	2.20	$1.97 \cdot 10^{-2}$	1.73
5/7	$3.58 \cdot 10^{-3}$	2.04	$7.04 \cdot 10^{-3}$	1.49
6/8	$8.96 \cdot 10^{-4}$	2.00	$4.24 \cdot 10^{-3}$	0.73

in Fig. 22. The convergence is presented in Table 13 and shows second-order convergence for the solution and first-order convergence for its gradient.

6. Summary

We have presented a novel fully implicit approach based on Voronoi diagrams for solving an Elliptic equation with discontinuities in the solution, its gradient, the diffusion coefficient and the flux across an irregular interface. The interface was captured through a level-set framework, and the equation was discretized with a finite volumes approach on the local Voronoi mesh. The contributions from the discontinuities were included naturally in the right-hand-side of the linear system, preserving its positive symmetric definiteness. We demonstrated second-order convergence of the solution and first-order convergence of its gradient, in the L^∞ norm, in both two and three spatial dimensions and on both uniform and adaptive Quad/Oc-tree base meshes. Large ratios in the diffusion coefficients are readily considered. We believe this approach could be utilized for numerous physical applications including those mentioned in the introduction.

Acknowledgement

The research of A. Guittet and F. Gibou was supported in part by ONR N00014-11-1-0027 and by the W.M. Keck Foundation. Part of this work has been supported by a CNES and a MeGEP scholarship, allowing M. Lepilliez to visit the authors at UCSB.

References

- [1] L. Adams, T. Chartier, New geometric immersed interface multigrid solvers, *SIAM J. Sci. Comput.* 25 (2004) 1516–1533.
- [2] L. Adams, T. Chartier, A comparison of algebraic multigrid and geometric immersed interface multigrid methods for interface problems, *SIAM J. Sci. Comput.* 26 (2005) 762–784.
- [3] L. Adams, Z. Li, The immersed interface/multigrid methods for interface problems, *SIAM J. Sci. Comput.* 24 (2002) 463–479.
- [4] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, *Computing* 5 (1970) 207–213.
- [5] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, *PETSc Users Manual*, Argonne National Laboratory, 2012.
- [6] S. Balay, J. Brown, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, *PETSc web page*, 2012.
- [7] Z. Bao, J.-M. Hong, J. Teran, R. Fedkiw, Fracturing rigid materials, *IEEE Trans. Vis. Comput. Graph.* 13 (2007) 370–378.
- [8] T. Belytschko, N. Moës, S. Usui, C. Parimi, Arbitrary discontinuities in finite elements, *Int. J. Numer. Methods Eng.* 50 (2001) 993–1013.
- [9] P.A. Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, *J. Comput. Phys.* 197 (2004) 364–386.
- [10] J. Bramble, J. King, A finite element method for interface problems in domains with smooth boundaries and interfaces, *Adv. Comput. Math.* 6 (1996) 109–138.
- [11] T. Chen, J. Strain, Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems, *J. Comput. Phys.* 227 (16) (2008) 7503–7542.
- [12] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, *Numer. Math.* 79 (1996) 175–202.
- [13] M. Cisternino, L. Weynans, A parallel second order Cartesian method for elliptic interface problems, *Commun. Comput. Phys.* 12 (2012) 1562–1587.
- [14] A. Coco, G. Russo, Second order multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface, I: one dimensional problems, *Numer. Math., Theory Methods Appl.* 5 (19) (2012).
- [15] A. Coco, G. Russo, Finite-difference ghost-point multigrid methods on cartesian grids for elliptic problems in arbitrary domains, *J. Comput. Phys.* 241 (0) (2013) 464–501.
- [16] R. Crockett, P. Colella, D. Graves, A cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions, *J. Comput. Phys.* 230 (7) (2011) 2451–2469.
- [17] C. Daux, N. Moës, J. Dolbow, N. Sukumar, T. Belytschko, Arbitrary branched and intersecting cracks with the extended finite element method, *Int. J. Numer. Methods Eng.* 48 (2000) 1741–1760.
- [18] J. Dolbow, I. Harari, An efficient finite element method for embedded interface problems, *Int. J. Numer. Methods Eng.* 78 (2009) 229–252.
- [19] M. Dryja, A Neumann–Neumann algorithm for mortar discretization of elliptic problems with discontinuous coefficients, *Numer. Math.* 99 (2005) 645–656.
- [20] R.D. Falgout, U.M. Yang, Hypre: a library of high performance preconditioners, in: P.M. Sloot, A.G. Hoekstra, C.K. Tan, J.J. Dongarra (Eds.), *Computational Science – ICCS 2002*, in: *Lecture Notes in Computer Science*, vol. 2331, Springer, Berlin, Heidelberg, 2002, pp. 632–641.
- [21] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *J. Comput. Phys.* 152 (1999) 457–492.
- [22] T. Fries, T. Belytschko, The intrinsic XFEM: a method for arbitrary discontinuities without additional unknowns, *Int. J. Numer. Methods Eng.* 68 (2006) 1358–1385.
- [23] E. Ewing, Z. Li, T. Lin, Y. Lin, The immersed finite volume element methods for the elliptic interface problems, *Math. Comput. Simul.* 50 (1999) 43–61.

- [24] F. Gibou, R. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, *J. Comput. Phys.* 202 (2005) 577–601.
- [25] F. Gibou, R. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, *J. Comput. Phys.* 176 (2002) 205–227.
- [26] F. Gibou, C. Min, R. Fedkiw, High resolution sharp computational methods for elliptic and parabolic problems in complex geometries, *J. Sci. Comput.* 54 (2013) 369–413.
- [27] Y. Gong, B. Li, Z. Li, Immersed-interface finite-element methods for elliptic interface problems with non-homogeneous jump conditions, *SIAM J. Numer. Anal.* 46 (2008) 472–495.
- [28] S. Grof, A. Reusken, An extended pressure finite element space for two-phase incompressible flows with surface tension, *J. Comput. Phys.* 224 (2007) 40–58.
- [29] G. Guyomarch, C.-O. Lee, K. Jeon, A discontinuous Galerkin method for elliptic interface problems with application to electroporation, *Commun. Numer. Methods Eng.* 25 (2009) 991–1008.
- [30] A. Hansbo, P. Hansbo, A finite element method for the simulation of strong and weak discontinuities in solid mechanics, *Comput. Methods Appl. Mech. Eng.* 193 (2004) 3523–3540.
- [31] S. Hou, W. Wang, L. Wang, Numerical method for solving matrix coefficient elliptic equation with sharp-edged interfaces, *J. Comput. Phys.* 229 (19) (2010) 7162–7179.
- [32] J. Huang, J. Zou, A mortar element method for elliptic problems with discontinuous coefficients, *IMA J. Numer. Anal.* 22 (2001) 549–576.
- [33] H. Ji, J. Dolbow, On strategies for enforcing interfacial constraints and evaluating jump conditions with extended finite element method, *Int. J. Numer. Methods Eng.* 61 (2004) 2508–2535.
- [34] J.L.H. Jr., L. Wang, E. Sifakis, J.M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions, *J. Comput. Phys.* 231 (4) (2012) 2015–2048.
- [35] B. Lamichhane, B. Wohlmuth, Mortar finite elements for interface problems, *Computing* 72 (2004) 333–348.
- [36] R. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* 31 (1994) 1019–1044.
- [37] A.J. Lew, G.C. Buscaglia, A discontinuous-Galerkin-based immersed boundary method, *Int. J. Numer. Methods Eng.* 76 (2008) 427–454.
- [38] Z. Li, A fast iterative algorithm for elliptic interface problems, *SIAM J. Numer. Anal.* 35 (1998) 230–254.
- [39] Z. Li, K. Ito, *The Immersed Interface Method – Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, SIAM Frontiers in Applied Mathematics, vol. 33, 2006.
- [40] Z. Li, T. Lin, X. Wu, New cartesian grid methods for interface problems using the finite element formulation, *Numer. Math.* 96 (2003) 61–98.
- [41] X.D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, *J. Comput. Phys.* 154 (2000) 151.
- [42] X.-D. Liu, T. Sideris, Convergence of the ghost-fluid method for elliptic equations with interfaces, *Math. Comput.* 72 (2003) 1731–1746.
- [43] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, *SIAM J. Numer. Anal.* 21 (1984) 285–299.
- [44] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids, *J. Comput. Phys.* 219 (2) (Dec. 2006) 912–929.
- [45] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive Cartesian grids, *J. Comput. Phys.* 225 (1) (2007) 300–321.
- [46] M. Latige, T. Colin, G. Gallice, A second order cartesian finite volume method for elliptic interface and embedded Dirichlet problems, *Comput. Fluids* 83 (2013) 70–76.
- [47] N. Moës, M. Cloirec, P. Cartraud, J. Remacle, A computational approach to handle complex microstructure geometries, *Comput. Methods Appl. Mech. Eng.* 192 (2003) 3162–3177.
- [48] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, *Int. J. Numer. Methods Eng.* 46 (1999) 131–150.
- [49] N. Molino, J. Bao, R. Fedkiw, A virtual node algorithm for changing mesh topology during simulation, *ACM Trans. Graph.: SIGGRAPH Proc.* 23 (2004) 385–392.
- [50] Y.T. Ng, C. Min, F. Gibou, An efficient fluid–solid coupling algorithm for single-phase flows, *J. Comput. Phys.* 228 (23) (Dec. 2009) 8807–8829.
- [51] M. Oevermann, C. Scharfenberg, R. Klein, A sharp interface finite volume method for elliptic equations on Cartesian grids, *J. Comput. Phys.* 228 (2009) 5184–5206.
- [52] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1) (1988) 12–49.
- [53] J. Papac, F. Gibou, C. Ratsch, Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions, *J. Comput. Phys.* 229 (3) (Feb. 2010) 875–889.
- [54] J. Papac, A. Helgadottir, C. Ratsch, F. Gibou, A level set approach for diffusion and Stefan-type problems with Robin boundary conditions on Quadtree/Octree adaptive cartesian grids, *J. Comput. Phys.* (2012).
- [55] J.W. Purvis, J.E. Burkhalter, Prediction of critical Mach number for store configurations, *AIAA J.* 17 (1979) 1170–1177.
- [56] C. Richardson, J. Hegemann, E. Sifakis, J. Hellrung, J. Teran, An XFEM method for modeling geometrically elaborate crack propagation in brittle materials, *Int. J. Numer. Methods Eng.* 88 (2011) 1042–1065.
- [57] C.H. Rycroft, Voronoi++: a three-dimensional Voronoi cell library in C++, *Chaos* 19 (2009).
- [58] Sandia National Laboratory, International Meshing Roundtable, Thistle Marble Arch, London, United Kingdom, 2014.
- [59] E. Sifakis, K. Der, R. Fedkiw, Arbitrary cutting of deformable tetrahedralized objects, in: *Proceedings of SIGGRAPH 2007*, 2007, pp. 73–80.
- [60] J.-H. Song, P. Areias, T. Belytschko, A method for dynamic crack and shear band propagation with phantom nodes, *Int. J. Numer. Methods Eng.* 67 (2006) 868–893.
- [61] N. Sukumar, Voronoi cell finite difference method for the diffusion operator on arbitrary unstructured grids, *Int. J. Numer. Methods Eng.* 57 (2003) 1–34.
- [62] F. van der Bos, V. Gravemeier, Numerical simulation of premixed combustion using an enriched finite element method, *J. Comput. Phys.* 228 (2009) 3605–3624.
- [63] R. Vanselow, Relations between FEM and FVM applied to the Poisson equation, *Computing* 57 (1996) 93–104.
- [64] A. Wiegmann, K. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, *SIAM J. Numer. Anal.* 37 (2000) 827–862.
- [65] S. Yu, Y. Zhou, G. Wei, Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces, *J. Comput. Phys.* 224 (2) (2007) 729–756.
- [66] S. Zhao, G. Wei, High-order {FDTD} methods via derivative matching for Maxwell's equations with material interfaces, *J. Comput. Phys.* 200 (1) (2004) 60–103.

Chapitre 3

Modélisation des Frontières Immergées

Ce chapitre est consacré à la modélisation d'écoulements en présence de frontières immergées. Nous y reprenons les travaux de Ng, Min et Gibou sur les écoulements monophasiques, et proposons une extension à la modélisation d'écoulements diphasiques. Dans cette étude, plusieurs discrétisations des termes visqueux sont comparées. Une discrétisation spatiale des termes capillaires est également proposée, ainsi qu'une méthode pour décrire la ligne triple lorsque l'interface rencontre le domaine irrégulier. Des cas analytiques de validation sont présentés et l'intégralité de ces méthodes fait l'objet d'une publication soumise au *Journal of Computational Physics*, disponible à la suite de ce chapitre.

An Efficient Two-Phase Flow Solver in Irregular Domains

Mathieu Lepilliez^{a,b,c}, Elena Roxana Popescu^a, Frederic Gibou^{d,e}, Sébastien Tanguy^a

^a*Institut de Mécanique des Fluides de Toulouse, 2bis allée du Professeur Camille Soula, 31400 Toulouse, France*

^b*Centre National d'Etudes Spatiales, 18 Avenue Edouard Belin, 31401 Toulouse Cedex 9, France*

^c*Airbus Défense & Space, 31 Avenue des Cosmonautes, 31402 Toulouse Cedex 4, France*

^d*Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, United States*

^e*Department of Computer Science, University of California, Santa Barbara, CA 93106-5110, United States*

Abstract

We present numerical methods that enable the direct numerical simulation of two-phase flows in irregular domains. A method is presented to account for surface tension effects in a mesh cell containing a triple line between the liquid, gas and solid phases. Our numerical method is based on the level-set method to capture the liquid-gas interface and on the single-phase Navier-Stokes solver in irregular domain proposed in [32] to impose the solid boundary in an Eulerian framework. We also present a strategy for the implicit treatment of the viscous term and how to impose both a Neumann boundary condition and a jump condition when solving for the pressure field. Special care is given on how to take into account the contact angle, the no-slip boundary condition for the velocity field and the volume forces. Finally, we present numerical results in two and three spatial dimensions evaluating our simulations with several benchmarks.

Keywords: Sharp Interface Methods, Irregular domains, Implicit Viscosity, Contact lines, Level Set Method, Ghost-Fluid Method

1. Introduction

Multiphase flows are ubiquitous in most modern processing technology and are critical to the understanding of a wide range of physical and biological phenomena. Consequently, the design of numerical methods enabling the simulation of multiphase flows is the subject of intense research. When considering solid boundaries, explicit descriptions of the geometry have demonstrated results of superior accuracy in the case of one-phase flows, when compared to implicit representations of the geometry. Indeed, body-fitted grids can be constructed to enable the accurate treatment of boundary conditions and to capture the rapid variations of the solution near walls, e.g. the turbulent boundary layer in high speed flows. In the case of multiphase flows, however, body-fitted approaches are less practical since conforming mesh must be reconstructed several time during the course of a simulation. This is, for example, the case of the simulation of bubbly flows or sprays, where the fluid interface can be strongly deformed and stretched. The automatic meshing necessary in such cases is a difficult task and in turn, the resulting mesh could lack the quality needed to guarantee accurate numerical results.

Email address: mathieu.lepilliez@imft.fr (Mathieu Lepilliez)

Since the pioneer work of Peskin [35], introducing the immersed boundary method, an important research effort has been devoted to the development of Eulerian fluid solvers in irregular domains. These methods have the advantage that no remeshing is necessary during the course of a simulation in the case of uniform grids. If adaptive grids are employed, then the remeshing is straightforward since the free boundary is embedded in the grid. Two classes of methods have emerged in the Eulerian framework: ‘delta’ formulations, where the Dirac distribution representing surface forces is approximated by a regularized function [49] (and the references therein) and ‘sharp’ approaches, where jump conditions at the free boundary are enforced at the discrete level [14, 17, 37, 47, 48, 58, 60] (and the references therein). Formulations based on regularized Dirac distributions provide a pathway to easily extend single-phase to multiphase solvers [7, 29]. On the other hand, these methods suffer from drawbacks that limit their use in some applications due to the inherent smearing of the solution near interfaces. Although efficient adaptive mesh refinement techniques [2, 18] can help reduce the extent of the smearing by imposing increased resolution where needed, ‘sharp’ numerical methods are still desired in some applications. These methods, however, are difficult to design due to the difficulty of imposing boundary conditions implicitly.

While several works have been presented in the last decade on immersed boundary methods for incompressible one-phase flows [4, 22, 28, 30, 31, 32, 51, 52, 54, 59], only a few [26, 33, 53] have been dedicated to incompressible two-phase flows, especially in the case where a contact line of the liquid-gas interface is formed on the embedded solid boundary. In such a situation, two different interfaces must be managed in a computational cell, and the numerical solver for the pressure must impose both a jump condition between the liquid and the gas to account for surface tension effects and a Neumann boundary condition between the fluid and the solid phase to impose the no-slip boundary condition.

An attractive immersed boundary method for one-phase flows, based on a finite volume discretization of the Laplace operator, has been proposed in [32]. It enables to impose solid boundary conditions on the pressure and on the velocity with a second-order spatial accuracy and to maintain the symmetric definite positiveness resulting linear system. Consequently, standard methods such as the preconditioned conjugate gradient can be used to invert efficiently the linear system. In the present paper, we extend the method of [32] to the more complex situation of two-phase flows. In particular, specific details are given on how to implicitly discretize the viscous term while imposing the jump condition on viscosity, how to enforce the no-slip condition when computing the viscous terms, how to carry out the computation of the surface tension on the contact line, how to impose the appropriate contact angle by extrapolating the liquid-gas level-set function inside the solid domain and to deal with volume forces inside the mesh cells which are crossed by the contact line.

Several benchmarks are proposed in section 4. In the first part of section 4, one-phase flow simulations are presented in order to compare two possible approaches to compute the viscous terms in two and three spatial dimensions. Next, test-cases involving two-phase flows in a complex geometry (droplet deposited on a slanted wall, half-filled rotating spherical tank) are presented and compared to theoretical solutions in order to highlight the behavior of the proposed numerical methods for the computation of the surface tension and the volume forces in the grid cells that are both crossed by the liquid-gas interface and the solid boundary. Since orders of accuracy are difficult to determine for such simulations, grid sensitivity studies are presented for all the benchmarks in order to ascertain that all the computations converge to the correct solution.

2. Equations and Standard Projection Methods

The Navier-Stokes equations describe the motion of fluids at the continuum level. However, their formulations and approximations depend on how surface forces are represented. Nevertheless, standard state-of-the-art numerical approximations are based on the projection method for single-phase flows, introduced by Chorin [10].

2.1. Single-Phase Flows

Consider a domain $\Omega = \Omega_l \cup \Omega_s$ with boundary $\partial\Omega$. The regions Ω_l and Ω_s represent the fluid and solid regions, respectively. The boundary between the fluid and the solid is denoted by Γ_s . The incompressible Navier-Stokes equations for one-phase Newtonian flows are written as:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega_l, \\ \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) &= -\nabla p + \mu \Delta \mathbf{u} + \rho \mathbf{g} \quad \text{in } \Omega_l, \end{aligned}$$

where t is the time, ρ the fluid density, $\mathbf{u} = (u, v, w)$ the velocity field, μ the viscosity assumed constant, p the pressure and \mathbf{g} the acceleration due to gravity.

In the case of single-phase flows, Chorin used the Hodge decomposition of vector fields, to design a three-stage projection method to solve the Navier-Stokes equations [10]: first, given a velocity \mathbf{u}^n at time $t^n = n\Delta t$, an intermediate velocity \mathbf{u}^* can be computed for a time step Δt without considering the pressure component:

$$\mathbf{u}^* = \mathbf{u}^n - \Delta t \left((\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \frac{\mu \Delta \mathbf{u}^n}{\rho} - \mathbf{g} \right).$$

Second, the pressure field p^{n+1} serves as the scalar function of the Hodge decomposition, satisfying:

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

with homogeneous Neumann boundary conditions on $\partial\Omega$ and non-homogeneous Neumann boundary condition on Γ_s :

$$\mathbf{n}_s \cdot \frac{\nabla p}{\rho} \Big|_{\Gamma_s} = \mathbf{n}_s \cdot (\mathbf{u}^* - \mathbf{u}_s) \Big|_{\Gamma_s}.$$

where \mathbf{u}_s is the specified velocity field on the solid's boundary. Finally, the fluid velocity \mathbf{u}^{n+1} is defined at the new time step t^{n+1} as the projection of \mathbf{u}^* onto the divergence-free space:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \frac{\nabla p^{n+1}}{\rho}.$$

2.2. Two-Phase Flows

Consider a computational domain, Ω , that contains solid, liquid and gas regions denoted by Ω_s , Ω_l and Ω_g , respectively¹ (see figure 1). We call Γ the liquid's boundary and \mathbf{n} its outward

¹We consider a liquid region and a gas region for the sake of presentation, but this work is equally applicable to two liquids or two gases.

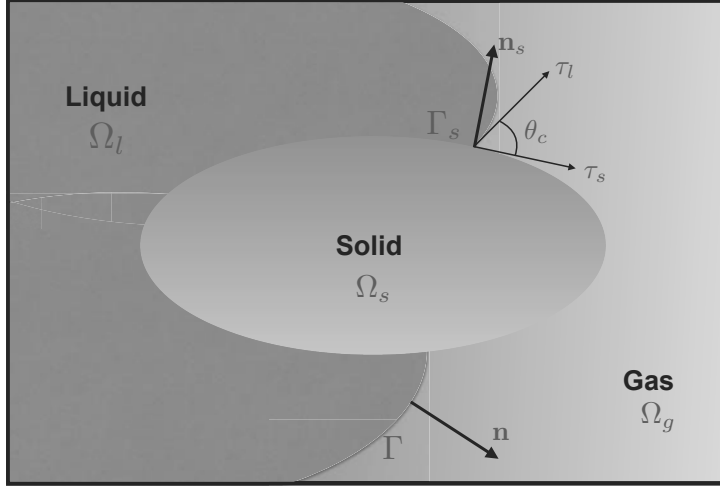


Figure 1: The computational domain in the case of two-phase flows in irregular domains.

normal. Likewise, we call Γ_s the solid's boundary and \mathbf{n}_s its outward normal. Finally, $\partial\Omega$ denotes the boundary of Ω . The incompressible Navier-Stokes equations for Newtonian two-phase flows can be described differently, whether the jump conditions are considered as functions in the entire computational domain, or as sharp jump conditions locally applied to the density, viscosity and pressure field. In what follows, we briefly present the main formulations that are used.

2.2.1. The “Delta” Formulation

The “delta” formulation [39, 44] writes the Navier-Stokes equations for two-phase flows as:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, \\ \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) &= -\nabla p + \rho \mathbf{g} + \nabla \cdot (2\mu \underline{\underline{\mathbf{D}}}) + \sigma \kappa \mathbf{n} \delta_\Gamma, \end{aligned}$$

where σ is the surface tension, κ is the interface local mean curvature, δ_Γ a multidimensional Dirac distribution localized at the interface, and $\underline{\underline{\mathbf{D}}}$ the rate-of-deformation tensor defined as:

$$\underline{\underline{\mathbf{D}}} = \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2}.$$

The viscous-stress tensor can be split into two parts:

$$\nabla \cdot (2\mu \underline{\underline{\mathbf{D}}}) = 2\mu \nabla \cdot (\underline{\underline{\mathbf{D}}}) + 2\underline{\underline{\mathbf{D}}} \cdot \nabla \mu.$$

Also, in the case where the viscosity is piecewise constant, it can be proved by using the definition of the Heaviside derivative that:

$$\nabla \mu = [\mu] \mathbf{n} \delta_\Gamma,$$

where $[\mu] = \mu_g - \mu_l$ denotes the jump in μ across Γ . Therefore, one can write:

$$\nabla \cdot (2\mu \underline{\underline{\mathbf{D}}}) = 2\mu \nabla \cdot (\underline{\underline{\mathbf{D}}}) + 2[\mu] \underline{\underline{\mathbf{D}}} \cdot \mathbf{n} \delta_\Gamma.$$

and it is clear that a jump condition in the viscous term has to be imposed. If no phase change occurs, the density and the viscosity are piecewise constant and only vary across the interface. They can then be defined as:

$$\begin{aligned}\rho &= \rho_l + H_\Gamma(\rho_g - \rho_l), \\ \mu &= \mu_l + H_\Gamma(\mu_g - \mu_l),\end{aligned}$$

with H_Γ denoting the Heaviside distribution, equal to 1 in the liquid and 0 in the gas phase, with the following definition for density and viscosity jump conditions:

$$\begin{aligned}[\rho] &= \rho_l - \rho_g, \\ [\mu] &= \mu_l - \mu_g.\end{aligned}$$

When applying a projection method to the “delta” formulation, one obtains the following explicit discretization: first, solve for an intermediate velocity field \mathbf{u}_1^* with

$$\mathbf{u}_1^* = \mathbf{u}^n - \Delta t \left((\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \frac{\nabla \cdot (2\mu \underline{\underline{\mathbf{D}}}^n)}{\rho^{n+1}} - \frac{\sigma \kappa}{\rho^{n+1}} \mathbf{n} \delta_\Gamma - \mathbf{g} \right).$$

Second, solve a Poisson equation to determine the Hodge field:

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{u}_1^*}{\Delta t}.$$

And finally project \mathbf{u}_1^* onto the divergence-free velocity field:

$$\mathbf{u}^{n+1} = \mathbf{u}_1^* - \Delta t \frac{\nabla p^{n+1}}{\rho^{n+1}}.$$

This numerical scheme is usually used in the framework of the Continuum Surface Force (CSF) model, which requires the definition of smoothed functions δ_ϵ and H_ϵ to approximate the Dirac and Heaviside distributions. The ease of implementation of this method comes with drawback of artificially thickening the interface. Unlike the CSF model, the “sharp” interface approach [21, 36, 45] build a sharp discretization of singular source terms, avoiding the introduction of a fictitious interface thickness. Some details on the theoretical equivalence between the jump condition formulation and the “delta” formulation is detailed in [23, 24].

2.2.2. The Ghost-Fluid Primitive Viscous Method

In [21], the authors proposed a sharp interface method for incompressible two-phase flows based on the principles of the Ghost-Fluid Method [14, 25]. This approach is widely used in the literature, even though it is only first-order accurate for solving the Poisson equation with jump conditions. In [19], a second-order approach developed, but it has not been applied to Navier-Stokes equations yet. The Ghost-Fluid Method introduced in [21] can be described briefly as proposed in [23]: first, the intermediate velocity \mathbf{u}_2^* is updated with

$$\mathbf{u}_2^* = \mathbf{u}^n - \Delta t \left((\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \frac{\mu \Delta \mathbf{u}^*}{\rho^{n+1}} - \mathbf{g} \right).$$

Next, the Poisson equation for the pressure is solved with the appropriate jump condition:

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{u}_2^*}{\Delta t} + \nabla \cdot \left(\frac{\sigma \kappa + 2[\mu] (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n}{\rho^{n+1}} \delta_\Gamma \mathbf{n} \right).$$

Finally the correction step gives the divergence free velocity at t^{n+1} :

$$\mathbf{u}^{n+1} = \mathbf{u}_2^* - \frac{\Delta t}{\rho^{n+1}} \left(\nabla p^{n+1} - \left(\sigma \kappa + 2[\mu] (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n \right) \mathbf{n} \delta_\Gamma \right).$$

This method has been dubbed the Ghost-Fluid Primitive viscous Method (GFPM) in [23]. In this framework, the computation of the viscous terms requires to determine explicitly the jump condition on the projected viscous stresses. As these jump conditions depend on the numerical solution, achieving a fully implicit temporal discretization of the viscous terms is not a straightforward task in the case of the GFPM. Depending on whether the CSF formulation or the GFPM formulation is used, the intermediate velocity in the prediction step is different. Even though these two methods are formally equivalent, they lead to different definitions of the predicted velocity field in the projection step.

2.2.3. The Ghost-Fluid Conservative Viscous Method

In [45], Sussman introduces another formulation for the intermediate velocity field, which [23] refers to as the Ghost-Fluid Conservative viscous Method (GFCM). In this case, the intermediate velocity \mathbf{u}_3^* is obtained as:

$$\mathbf{u}_3^* = \mathbf{u}^n - \Delta t \left((\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \frac{1}{\rho^{n+1}} \nabla \cdot (2\mu \underline{\underline{\mathbf{D}}}^n) - \mathbf{g} \right).$$

Next, the Poisson equation is solved to determine the pressure field:

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{u}_3^*}{\Delta t} + \nabla \cdot \left(\frac{\sigma \kappa \mathbf{n} \delta_\Gamma}{\rho^{n+1}} \right).$$

Finally in the correction step, the physical velocity field can be determined with the pressure field previously computed:

$$\mathbf{u}^{n+1} = \mathbf{u}_3^* - \frac{\Delta t}{\rho^{n+1}} (\nabla p^{n+1} - \sigma \kappa \mathbf{n} \delta_\Gamma).$$

In order to remove the $O(\Delta x^2)$ time step restriction incurred by the viscosity term, the following implicit temporal discretization is used, referred to in this paper as the Ghost-Fluid Conservative Method with an Implicit scheme (GFCMI): the intermediate velocity \mathbf{u}_4^* is updated as:

$$\rho^{n+1} \mathbf{u}_4^* - \Delta t \nabla \cdot (2\mu \underline{\underline{\mathbf{D}}}^*) = \rho^{n+1} (\mathbf{u}^n - \Delta t ((\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \mathbf{g})), \quad (2)$$

which leads to a large linear system where the three velocity components are coupled. The subsequent steps are similar to the one used in GFCM, with the computation of the pressure field:

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{u}_4^*}{\Delta t} + \nabla \cdot \left(\frac{\sigma \kappa \mathbf{n} \delta_\Gamma}{\rho^{n+1}} \right),$$

followed by the correction step:

$$\mathbf{u}^{n+1} = \mathbf{u}_4^* - \frac{\Delta t}{\rho^{n+1}} (\nabla p^{n+1} - \sigma \kappa \mathbf{n} \delta_\Gamma).$$

2.2.4. The Ghost-Fluid Semi-Conservative Method

Finally in [23] the authors proposed a semi-conservative formulation for discretizing the viscous term, referred to in this paper as the Ghost-Fluid Semi-Conservative Method with an Implicit scheme (GFSCMI). Starting with the splitting:

$$\begin{aligned}
\nabla \cdot (2\mu \underline{\mathbf{D}}) &= \nabla \cdot (\mu \nabla \mathbf{u} + \mu \nabla^T \mathbf{u}) \\
&= \nabla \cdot (\mu \nabla \mathbf{u}) + \nabla \cdot (\mu \nabla^T \mathbf{u}) \\
&= \nabla \cdot (\mu \nabla \mathbf{u}) + \nabla^T \mathbf{u} \cdot \nabla \mu + \mu \nabla \cdot (\nabla^T \mathbf{u}) \\
&= \nabla \cdot (\mu \nabla \mathbf{u}) + [\mu] \nabla^T \mathbf{u} \cdot \mathbf{n} \delta_\Gamma,
\end{aligned}$$

where $\nabla \cdot (\mu \nabla \mathbf{u})$ is a formal equivalent of the one-phase flow formulation of viscous effects. Therefore each component is solved independently from the others, as the cross-terms of the viscous term in $[\mu] \nabla^T \mathbf{u} \cdot \mathbf{n} \delta_\Gamma$ are enforced explicitly as a jump condition. The prediction step is:

$$\rho^{n+1} \mathbf{u}_5^* - \Delta t \nabla \cdot (\mu \nabla \mathbf{u}_5^*) = \rho^{n+1} \left(\mathbf{u}^n - \Delta t \left((\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \mathbf{g} \right) \right).$$

Then the computation of the pressure field takes into account part of the viscous effects in the pressure jump:

$$\nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = \frac{\nabla \cdot \mathbf{u}_5^*}{\Delta t} + \nabla \cdot \left(\frac{(\sigma \kappa + [\mu] (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n)}{\rho^{n+1}} \mathbf{n} \delta_\Gamma \right). \quad (3)$$

Finally, the correction step to compute the divergence free velocity is:

$$\mathbf{u}^{n+1} = \mathbf{u}_5^* - \frac{\Delta t}{\rho^{n+1}} \left(\nabla p^{n+1} - \left(\sigma \kappa + [\mu] (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n \right) \mathbf{n} \delta_\Gamma \right). \quad (4)$$

Remark: The implementation of an implicit temporal discretization of the viscous terms is easier with this method (GFSCMI) than GFCMI. Indeed if GFSCMI is used, each velocity component can be computed by solving a simple linear system (symmetric definite positive) that does not treat cross-derivatives implicitly. However, even though GFCMI treats cross-derivatives implicitly, a few iterations of Gauss-Seidel suffice for convergence and the results are superior to GFSCMI. These methods are compared in section 3.4 to offer some insight on how velocity boundary conditions should be enforced on irregular domains.

2.2.5. A Note on the Time Step Constraint

Let us remark that, unlike GFCMI, GFSCMI does not fully remove the stability constraint on the time step due to viscosity, since an explicit part depending on the viscosity jump condition remains in equation (4). Therefore, as it has been discussed in [23], a stability constraint depending on the viscosity jump condition must be imposed to ensure stability of the method. Specifically, the following standard time step constraints for the convection and the surface tension effects, must be imposed to ensure the temporal stability of the computation:

$$\Delta t_{\text{conv}} = \frac{\Delta x}{\max |\mathbf{u}|}, \quad \Delta t_{\text{surf.tens}} = \frac{1}{2} \sqrt{\frac{\rho_l}{\sigma}} \Delta x^{3/2},$$

for a global time step restriction of:

$$\frac{1}{\Delta t} = \frac{1}{\Delta t_{\text{conv}}} + \frac{1}{\Delta t_{\text{surf.tens}}}. \quad (5)$$

If GFSCMI is used, the following constraint must be added to account for the jump condition on viscosity in equation (4):

$$\Delta t_{\text{visc}} = \frac{\bar{\rho}}{[\mu]} \Delta x^2,$$

where $\bar{\rho}$ is the average density of the two phases. This leads to a new global time step restriction given by:

$$\frac{1}{\Delta t} = \frac{1}{\Delta t_{\text{conv}}} + \frac{1}{\Delta t_{\text{surf.tens}}} + \frac{1}{\Delta t_{\text{visc}}}.$$

2.3. A Note on Solid's Boundary Conditions for Multiphase Flows

As stated earlier, when irregular domains are handled, Neumann boundary conditions on the domain boundaries are enforced on Γ_s the fluid-solid interface for the pressure field. It leads to the following relation for GFCMI:

$$\mathbf{n}_s \cdot \frac{\nabla p}{\rho} \Big|_{\Gamma_s} = \mathbf{n}_s \cdot (\mathbf{u}^* - \mathbf{u}_s + \sigma \kappa \mathbf{n} \delta_\Gamma) \Big|_{\Gamma_s},$$

and for GFSCMI as:

$$\mathbf{n}_s \cdot \frac{\nabla p}{\rho} \Big|_{\Gamma_s} = \mathbf{n}_s \cdot \left(\mathbf{u}^* - \mathbf{u}_s + (\sigma \kappa + [\mu] (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n) \mathbf{n} \delta_\Gamma \right) \Big|_{\Gamma_s}.$$

One can notice that jump conditions have to be taken into account on the irregular domain as well. Section 3 will discuss the numerical discretization of these boundary conditions.

3. Numerical Methods

In this section, we provide the details for the spatial discretizations of the three different projection methods.

3.1. Capturing the Moving Interface

In this work, we use the level-set method of Osher and Sethian [34] to represent the interface between the two fluids. A signed and continuous level-set function ϕ is defined in a domain Ω such as $\Omega^- = \{\mathbf{x} : \phi(\mathbf{x}) < 0\}$, $\Omega^+ = \{\mathbf{x} : \phi(\mathbf{x}) > 0\}$ and $\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\}$ where Γ represents the interface captured by the following transport equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \quad (6)$$

One of the benefits for using the level-set function is its regular property in the whole domain. Sussman et al. [44] developed an algorithm allowing to maintain this property by re-initializing the level-set function with the following equation:

$$\frac{\partial d}{\partial \tau} = \text{Sign}(\phi) (1 - |\nabla d|), \quad (7)$$

where $\text{Sign}(\phi)$ is a regularized signed function defined by Sussman *et al.* in [44] and d the update of the function ϕ . Equation (7) iterates for a few fictitious time step τ to finally converge to a continuous signed distance function in the whole domain. Thus, geometric properties such as \mathbf{n} the outward unit normal vector to the interface and κ the local mean curvature can be accurately computed:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \quad \text{and} \quad \kappa(\phi) = \nabla \cdot \mathbf{n}.$$

Spatial derivatives in equations (6) and (7) are computed with the WENO-Z scheme [5], and the temporal derivative with a third order TVD Runge-Kutta scheme. Let us notice that the temporal integration is fully coupled with the Navier-Stokes solver [21].

3.2. Discretization of the Projection Method in Irregular Domains

The method developed in this study to solve the Poisson equation (3) on irregular domains is based on Ng, Min and Gibou's algorithm [32]. This numerical method presents several attractive features. Indeed the spatial discretization for the pressure Poisson equation is second order, for all computational nodes, including the irregular domain nodes crossed by the solid-fluid interface where the embedded Neumann Boundary condition is enforced. Moreover, the resulting matrix of the linear system is still symmetric definite positive which allows using a classical Preconditioned Conjugate Gradient or a Black Box MultiGrid solver [12].

Consider a vector field \mathbf{u}^* in the domain Ω , separated by the interface Γ_s into two distinct subdomains Ω_f and Ω_s , such as $\Omega = \Omega_f \cup \Omega_s$. As Ω_s would correspond to a solid media, we only solve the Navier-Stokes equations in Ω_f . Let ϕ_s be another level-set function such as $\Omega_s = \{\mathbf{x} : \phi_s(\mathbf{x}) > 0\}$, $\Omega_f = \{\mathbf{x} : \phi_s(\mathbf{x}) < 0\}$ and $\Gamma_s = \{\mathbf{x} : \phi_s(\mathbf{x}) = 0\}$. We consider a MAC grid configuration and a cell $C_{i,j} = [i - 1/2, i + 1/2] \times [j - 1/2, j + 1/2]$, partially covered with Ω_f . For more clarity we only consider a 2D example to illustrate the method which can be extrapolated to 3D. The configuration is illustrated in Fig 2.

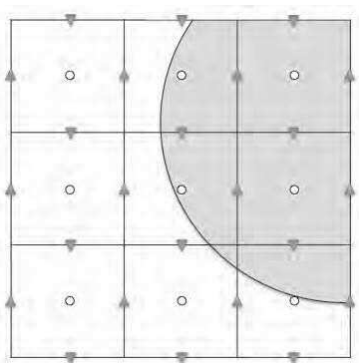


Figure 2: Standard MAC grid configuration: the scalar variables are sampled at the cells' centers (circles), the x-component of the velocity field is sampled on the vertical faces (blue triangles), and the y-component of the velocity field is sampled on the horizontal faces (red triangles). The irregular domain is represented by the shaded area Ω_s .

Considering a finite volume approach, we integrate the left hand side of equation (3) over $C_{i,j}$ and apply the divergence theorem:

$$\int_{C_{i,j} \cap \Omega_f} \nabla \cdot \left(\frac{\nabla p}{\rho} \right) dA = \int_{\partial(C_{i,j} \cap \Omega_f)} \mathbf{n} \cdot \left(\frac{\nabla p}{\rho} \right) dl,$$

with dA and dl differential area and length. Similarly for the right hand side:

$$\int_{C_{i,j} \cap \Omega_f} \nabla \cdot \mathbf{u}^* dA + \int_{C_{i,j} \cap \Omega_f} \nabla \cdot \left(\frac{\mathbf{F}_{ST}}{\rho} \right) dA = \int_{\partial(C_{i,j} \cap \Omega_f)} \mathbf{n} \cdot \mathbf{u}^* dl + \int_{\partial(C_{i,j} \cap \Omega_f)} \mathbf{n} \cdot \left(\frac{\mathbf{F}_{ST}}{\rho} \right) dl,$$

where \mathbf{F}_{ST} can represent the surface tension effects and the jump conditions due to viscosity depending on the choice of the intermediate velocity:

- For GFCMI, $\mathbf{F}_{ST} = \sigma \kappa \mathbf{n}_f \delta_f$.
- For GFSCMI, $\mathbf{F}_{ST} = (\sigma \kappa + [\mu]_\Gamma (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n) \mathbf{n} \delta_f$.

We now only consider the contribution of all the components of $\partial(C_{i,j} \cap \Omega_f)$, and define the length fraction $L_{i,j}$ of the face covered by the irregular domain $\{x | \phi_s(x) \leq 0\}$ which can be linearly approximated here for $[i - \frac{1}{2}] \times [j - \frac{1}{2}, j + \frac{1}{2}]$ by:

$$L_{i-\frac{1}{2},j} = \begin{cases} \Delta y \frac{\phi_{i-\frac{1}{2},j-\frac{1}{2}}}{\phi_{i-\frac{1}{2},j-\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}}} & \text{for } \phi_{i-\frac{1}{2},j-\frac{1}{2}} < 0 \quad \text{and } \phi_{i-\frac{1}{2},j+\frac{1}{2}} > 0, \\ \Delta y \frac{\phi_{i-\frac{1}{2},j+\frac{1}{2}}}{\phi_{i-\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j-\frac{1}{2}}} & \text{for } \phi_{i-\frac{1}{2},j-\frac{1}{2}} > 0 \quad \text{and } \phi_{i-\frac{1}{2},j+\frac{1}{2}} < 0, \\ \Delta y & \text{for } \phi_{i-\frac{1}{2},j-\frac{1}{2}} < 0 \quad \text{and } \phi_{i-\frac{1}{2},j+\frac{1}{2}} < 0, \\ 0 & \text{for } \phi_{i-\frac{1}{2},j-\frac{1}{2}} > 0 \quad \text{and } \phi_{i-\frac{1}{2},j+\frac{1}{2}} > 0. \end{cases} \quad (8)$$

For more clarity the subscript "s" is omitted, but it is the function ϕ_s that is used to estimate the length fractions. Using the length fraction established in equation (8), and sampled values at the center of the cell, we obtain:

$$\begin{aligned} - \int_{\partial(C_{i,j} \cap \Omega_f)} \mathbf{n} \cdot \left(\frac{\nabla p}{\rho} \right) dl &\simeq \frac{L_{i-\frac{1}{2},j}}{\rho_{i-\frac{1}{2},j}} \left(\frac{p_{i,j} - p_{i-1,j}}{\Delta x} \right) + \frac{L_{i+\frac{1}{2},j}}{\rho_{i+\frac{1}{2},j}} \left(\frac{p_{i,j} - p_{i+1,j}}{\Delta x} \right) \\ &+ \frac{L_{i,j-\frac{1}{2}}}{\rho_{i,j-\frac{1}{2}}} \left(\frac{p_{i,j} - p_{i,j-1}}{\Delta y} \right) + \frac{L_{i,j+\frac{1}{2}}}{\rho_{i,j+\frac{1}{2}}} \left(\frac{p_{i,j} - p_{i,j+1}}{\Delta y} \right) \\ &- \int_{C_{i,j} \cap \Gamma} \mathbf{n} \cdot \left(\frac{\nabla p}{\rho} \right) dl, \end{aligned}$$

where $\int_{C_{i,j} \cap \Gamma}$ is the integral over the interface with the irregular external boundary. Similarly, we obtain:

$$\begin{aligned} - \int_{\partial(C_{i,j} \cap \Omega_f)} \mathbf{n} \cdot \mathbf{u}^* dl &\simeq L_{i-\frac{1}{2},j} u_{i-\frac{1}{2},j}^* - L_{i+\frac{1}{2},j} u_{i+\frac{1}{2},j}^* + L_{i,j-\frac{1}{2}} v_{i,j-\frac{1}{2}}^* - L_{i,j+\frac{1}{2}} v_{i,j+\frac{1}{2}}^* \\ &- \int_{C_{i,j} \cap \Gamma} \mathbf{n} \cdot \mathbf{u}^* dl. \end{aligned}$$

For cells where both Γ and Γ_s are present, respectively the fluid-fluid interface and the fluid-solid interface, the Ghost-Fluid method [14, 25] is applied on each part of Γ : let $\beta_{i,j} = \frac{1}{\rho_{i,j}}$ be a diffusion coefficient in the cell, computed with a harmonic average of the values β^+ in the region where ϕ is positive and β^- for the negative region, and $a(x_\Gamma) = \sigma\kappa$ the corresponding jump function for GFCMI. For the interface crossing a cell border between $x_{i,j}$ and $x_{i+1,j}$:

$$\beta_{i+1/2,j} = \frac{\beta^+\beta^-}{\beta^+\theta + \beta^-(1-\theta)},$$

and

$$a(x_\Gamma)_{i+1/2,j} = \sigma\kappa_{i,j}\theta + \sigma\kappa_{i+1,j}(1-\theta),$$

with

$$\theta = \frac{|\phi_{i+1,j}|}{|\phi_{i,j}| + |\phi_{i+1,j}|}.$$

Then one can define

$$\begin{aligned} - \int_{\partial(C_{i,j} \cap \Omega^-)} \mathbf{n} \cdot \left(\frac{\sigma\kappa \mathbf{n}_f \delta_f}{\rho} \right) dl &\simeq L_{i-\frac{1}{2},j} \beta_{i-\frac{1}{2},j} a(x_\Gamma)_{i-\frac{1}{2},j} + L_{i+\frac{1}{2},j} \beta_{i+\frac{1}{2},j} a(x_\Gamma)_{i+\frac{1}{2},j} \\ &+ L_{i,j-\frac{1}{2}} \beta_{i,j-\frac{1}{2}} a(x_\Gamma)_{i,j-\frac{1}{2}} + L_{i,j+\frac{1}{2}} \beta_{i,j+\frac{1}{2}} a(x_\Gamma)_{i,j+\frac{1}{2}} \\ &- \int_{C_{i,j} \cap \Gamma} \mathbf{n} \cdot \left(\frac{\sigma\kappa \mathbf{n}_f \delta_f}{\rho} \right) dl. \end{aligned}$$

If GFSCMI is used, one can notice that the formulation is almost the same, except that the function $a(x_\Gamma)$ will be completed with the jump due to the viscous term:

$$a(x_\Gamma)_{i+1/2,j} = \theta (\sigma\kappa + [\mu]_\Gamma (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n)_{i,j} + (1-\theta) (\sigma\kappa + [\mu]_\Gamma (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n)_{i+1,j},$$

and

$$\begin{aligned} - \int_{\partial(C_{i,j} \cap \Omega^-)} \mathbf{n} \cdot \left(\frac{(\sigma\kappa + [\mu]_\Gamma (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n) \mathbf{n}_f \delta_f}{\rho} \right) dl &\simeq L_{i-\frac{1}{2},j} \beta_{i-\frac{1}{2},j} a(x_\Gamma)_{i-\frac{1}{2},j} + L_{i+\frac{1}{2},j} \beta_{i+\frac{1}{2},j} a(x_\Gamma)_{i+\frac{1}{2},j} \\ &+ L_{i,j-\frac{1}{2}} \beta_{i,j-\frac{1}{2}} a(x_\Gamma)_{i,j-\frac{1}{2}} + L_{i,j+\frac{1}{2}} \beta_{i,j+\frac{1}{2}} a(x_\Gamma)_{i,j+\frac{1}{2}} \\ &- \int_{C_{i,j} \cap \Gamma} \mathbf{n} \cdot \left(\frac{(\sigma\kappa + [\mu]_\Gamma (\mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n})^n) \mathbf{n}_f \delta_f}{\rho} \right) dl. \end{aligned}$$

Finally, one obtains the following problem with Neumann boundary conditions on the embedded solid boundary:

$$\begin{aligned} &\frac{L_{i-\frac{1}{2},j}}{\rho_{i-\frac{1}{2},j}} \left(\frac{p_{i,j} - p_{i-1,j}}{\Delta x} \right) + \frac{L_{i+\frac{1}{2},j}}{\rho_{i+\frac{1}{2},j}} \left(\frac{p_{i,j} - p_{i+1,j}}{\Delta x} \right) + \frac{L_{i,j-\frac{1}{2}}}{\rho_{i,j-\frac{1}{2}}} \left(\frac{p_{i,j} - p_{i,j-1}}{\Delta y} \right) + \frac{L_{i,j+\frac{1}{2}}}{\rho_{i,j+\frac{1}{2}}} \left(\frac{p_{i,j} - p_{i,j+1}}{\Delta y} \right) \\ &= L_{i-\frac{1}{2},j} u_{i-\frac{1}{2},j}^* - L_{i+\frac{1}{2},j} u_{i+\frac{1}{2},j}^* + L_{i,j-\frac{1}{2}} v_{i,j-\frac{1}{2}}^* - L_{i,j+\frac{1}{2}} v_{i,j+\frac{1}{2}}^* \\ &+ L_{i-\frac{1}{2},j} \beta_{i-\frac{1}{2},j} a(x_\Gamma)_{i-\frac{1}{2},j} + L_{i+\frac{1}{2},j} \beta_{i+\frac{1}{2},j} a(x_\Gamma)_{i+\frac{1}{2},j} \\ &+ L_{i,j-\frac{1}{2}} \beta_{i,j-\frac{1}{2}} a(x_\Gamma)_{i,j-\frac{1}{2}} + L_{i,j+\frac{1}{2}} \beta_{i,j+\frac{1}{2}} a(x_\Gamma)_{i,j+\frac{1}{2}}. \end{aligned}$$

As Ng, Min and Gibou demonstrated in [32], the above discretization forms a symmetric positive definite linear system for the pressure field.

3.3. Viscosity Modeling

The implementation of the viscosity terms depends on the formulation of the intermediate velocity.

3.3.1. Viscosity Modeling - GFCMI Equation

We have seen in section 2 that the equation of the intermediate velocity for GFCMI is:

$$\rho^{n+1}\mathbf{U}^* - \Delta t(\nabla \cdot (2\mu\underline{\underline{\mathbf{D}}}^*)) = \mathbf{F}_{\text{rhs}}, \quad (9)$$

with \mathbf{F}_{rhs} the right-hand side of equation (9):

$$\mathbf{F}_{\text{rhs}} = \rho^{n+1}(\mathbf{u}^n - \Delta t((\mathbf{U}^n \cdot \nabla) \mathbf{U}^n - \mathbf{g})). \quad (10)$$

In order to compute the intermediate velocity field, one must compute all the terms of the right-hand side of equation (10). The advection term is solved with a fifth order WENO-Z scheme [5]. Let us remind the stress tensor formulation for the discretization of the viscous term:

$$\underline{\underline{\mathbf{D}}} = \frac{1}{2}(\nabla \mathbf{u} + \nabla^T \mathbf{u}) \quad \text{with} \quad \nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}.$$

The discretization is presented as a 2D discretization, but can be easily extended to a 3D formulation. By applying the divergence operator:

$$\nabla \cdot (2\mu\underline{\underline{\mathbf{D}}}) = \begin{pmatrix} \frac{\partial}{\partial x} \left(2\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) \\ \frac{\partial}{\partial y} \left(2\mu \frac{\partial v}{\partial y} \right) + \frac{\partial}{\partial x} \left(\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) \end{pmatrix}.$$

By projecting on the \mathbf{e}_x direction:

$$\nabla \cdot (2\mu\underline{\underline{\mathbf{D}}}) \cdot \mathbf{e}_x \Big|_{i+\frac{1}{2},j} = \frac{\partial}{\partial x} \left(2\mu \frac{\partial u}{\partial x} \right) \Big|_{i+\frac{1}{2},j} + \frac{\partial}{\partial y} \left(\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) \Big|_{i+\frac{1}{2},j}, \quad (11)$$

with:

$$\frac{\partial}{\partial x} \left(2\mu \frac{\partial u}{\partial x} \right) \Big|_{i+\frac{1}{2},j} \simeq \frac{2\mu_{i+1,j} \left(\frac{u_{i+\frac{3}{2},j} - u_{i+\frac{1}{2},j}}{\Delta x} \right) - 2\mu_{i,j} \left(\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} \right)}{\Delta x},$$

$$\frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) \Big|_{i+\frac{1}{2},j} \simeq \frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{u_{i+\frac{1}{2},j+\frac{1}{2}} - u_{i+\frac{1}{2},j}}{\Delta y} \right) - \mu_{i+\frac{1}{2},j-\frac{1}{2}} \left(\frac{u_{i+\frac{1}{2},j} - u_{i+\frac{1}{2},j-1}}{\Delta y} \right)}{\Delta y},$$

and

$$\frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial x} \right) \Big|_{i+\frac{1}{2},j} \simeq \frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{v_{i+1,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}}{\Delta x} \right) - \mu_{i+\frac{1}{2},j-\frac{1}{2}} \left(\frac{v_{i+1,j-\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta x} \right)}{\Delta y}.$$

Likewise, by projecting on the $\mathbf{e}_y|_{i,j+\frac{1}{2}}$ direction:

$$\nabla \cdot (2\mu \underline{\mathbf{D}}) \cdot \mathbf{e}_y|_{i,j+\frac{1}{2}} = \frac{\partial}{\partial x} \left(\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) \Big|_{i,j+\frac{1}{2}} + \frac{\partial}{\partial y} \left(2\mu \frac{\partial v}{\partial y} \right) \Big|_{i,j+\frac{1}{2}}, \quad (12)$$

with:

$$\frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial y} \right) \Big|_{i,j+\frac{1}{2}} \simeq \frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j}}{\Delta y} \right) - \mu_{i-\frac{1}{2},j+\frac{1}{2}} \left(\frac{u_{i-\frac{1}{2},j+1} - u_{i-\frac{1}{2},j}}{\Delta y} \right)}{\Delta x},$$

$$\frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial x} \right) \Big|_{i,j+\frac{1}{2}} \simeq \frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{v_{i+1,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}}{\Delta x} \right) - \mu_{i-\frac{1}{2},j+\frac{1}{2}} \left(\frac{v_{i,j+\frac{1}{2}} - v_{i-1,j+\frac{1}{2}}}{\Delta x} \right)}{\Delta x},$$

and

$$\frac{\partial}{\partial y} \left(2\mu \frac{\partial v}{\partial y} \right) \Big|_{i,j+\frac{1}{2}} \simeq \frac{2\mu_{i,j+1} \left(\frac{v_{i,j+\frac{3}{2}} - v_{i,j+\frac{1}{2}}}{\Delta y} \right) - 2\mu_{i,j} \left(\frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y} \right)}{\Delta y}.$$

In order to compute rightfully the viscous term at the intermediate time step, one must solve a coupled linear system of two matrices with 9 diagonals per velocity components as:

$$\begin{aligned} & a_{i+1,j} u_{i+\frac{3}{2},j}^* + a_{i,j} u_{i-\frac{1}{2},j}^* + a_{i+\frac{1}{2},j+\frac{1}{2}} u_{i+\frac{1}{2},j+1}^* + a_{i+\frac{1}{2},j-\frac{1}{2}} u_{i+\frac{1}{2},j-1}^* + \alpha u_{i+\frac{1}{2},j}^* \\ & - b_{i+\frac{1}{2},j+\frac{1}{2}} v_{i+1,j+\frac{1}{2}}^* + b_{i+\frac{1}{2},j+\frac{1}{2}} v_{i,j+\frac{1}{2}}^* - b_{i+\frac{1}{2},j-\frac{1}{2}} v_{i+1,j-\frac{1}{2}}^* + b_{i-\frac{1}{2},j-\frac{1}{2}} v_{i,j-\frac{1}{2}}^* = F_{x\text{-rhs}} \end{aligned}$$

and

$$\begin{aligned} & c_{i+\frac{1}{2},j+\frac{1}{2}} v_{i+1,j+\frac{1}{2}}^* + c_{i-\frac{1}{2},j+\frac{1}{2}} v_{i-1,j+\frac{1}{2}}^* + c_{i,j+1} v_{i,j+\frac{3}{2}}^* + c_{i,j} v_{i,j-\frac{1}{2}}^* + \beta v_{i,j+\frac{1}{2}}^* \\ & - b_{i+\frac{1}{2},j+\frac{1}{2}} u_{i+\frac{1}{2},j+1}^* - b_{i-\frac{1}{2},j+\frac{1}{2}} u_{i+\frac{1}{2},j}^* + b_{i+\frac{1}{2},j+\frac{1}{2}} u_{i-\frac{1}{2},j+1}^* + b_{i-\frac{1}{2},j+\frac{1}{2}} u_{i-\frac{1}{2},j}^* = F_{y\text{-rhs}} \end{aligned}$$

with $F_{x\text{-rhs}}$ and $F_{y\text{-rhs}}$ the right-hand side terms from GFCMI projection method in equation (10), respectively projected onto the x - and y - directions, respectively. The matrix coefficients are deduced from equations (9), (11) and (12) as:

$$\begin{aligned} a_{i+1,j} &= -\frac{2\mu_{i+1,j}}{\Delta x^2} \Delta t, & b_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x \Delta y} \Delta t, & c_{i+\frac{1}{2},j+\frac{1}{2}} &= -\frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x^2} \Delta t, \\ a_{i,j} &= -\frac{2\mu_{i,j}}{\Delta x^2} \Delta t, & b_{i+\frac{1}{2},j-\frac{1}{2}} &= \frac{\mu_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta x \Delta y} \Delta t, & c_{i-\frac{1}{2},j+\frac{1}{2}} &= -\frac{\mu_{i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x^2} \Delta t, \\ a_{i+\frac{1}{2},j+\frac{1}{2}} &= -\frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}}}{\Delta y^2} \Delta t, & b_{i-\frac{1}{2},j+\frac{1}{2}} &= \frac{\mu_{i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x \Delta y} \Delta t, & c_{i,j+1} &= -\frac{2\mu_{i,j+1}}{\Delta y^2} \Delta t, \\ a_{i+\frac{1}{2},j-\frac{1}{2}} &= -\frac{\mu_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y^2} \Delta t, & b_{i-\frac{1}{2},j-\frac{1}{2}} &= \frac{\mu_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x \Delta y} \Delta t, & c_{i,j} &= -\frac{2\mu_{i,j}}{\Delta y^2} \Delta t, \end{aligned} \quad (13)$$

$$\alpha = \rho_{i+\frac{1}{2},j}^{n+1} - a_{i+1,j} - a_{i,j} - a_{i+\frac{1}{2},j+\frac{1}{2}} - a_{i+\frac{1}{2},j-\frac{1}{2}}, \quad (14)$$

and

$$\beta = \rho_{i,j+\frac{1}{2}}^{n+1} - c_{i+\frac{1}{2},j+\frac{1}{2}} - c_{i-\frac{1}{2},j+\frac{1}{2}} - c_{i,j+1} - c_{i,j}. \quad (15)$$

As the linear system is diagonally dominant (depending on the time step constraint, if $\rho \gg \Sigma a_{ij}$ then the system is diagonally dominant, which is always the case in our configurations), it can be solved efficiently with a few iterations of the Gauss-Seidel algorithm. For a 3D system, a third coupled matrix has to be accounted for, resulting in a 15-diagonal matrix per velocity components.

3.3.2. Viscosity Modeling - GFSCMI Equation

GFSCMI projection proposed in [23] enables an easier implementation for an implicit temporal discretization of the viscous terms, albeit semi-implicit:

$$\rho^{n+1} \mathbf{u}_5^* - \Delta t \nabla \cdot (\mu \nabla \mathbf{u}_5^*) = \mathbf{F}_{\text{rhs}} = \rho^{n+1} \left(\mathbf{u}^n - \Delta t (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \mathbf{g} \right). \quad (16)$$

The viscous term can be computed as:

$$\nabla \cdot (\mu \nabla \mathbf{u}) = \begin{pmatrix} \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) \\ \frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial y} \right) \end{pmatrix}.$$

By projecting in the x -direction:

$$\nabla \cdot (\mu \nabla \mathbf{u}) \cdot \mathbf{e}_x \Big|_{i+\frac{1}{2},j} = \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) \Big|_{i+\frac{1}{2},j} + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) \Big|_{i+\frac{1}{2},j}, \quad (17)$$

with:

$$\frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) \Big|_{i+\frac{1}{2},j} \simeq \frac{\mu_{i+1,j} \left(\frac{u_{i+\frac{3}{2},j} - u_{i+\frac{1}{2},j}}{\Delta x} \right) - \mu_{i,j} \left(\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} \right)}{\Delta x},$$

and

$$\frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) \Big|_{i+\frac{1}{2},j} \simeq \frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j}}{\Delta y} \right) - \mu_{i+\frac{1}{2},j-\frac{1}{2}} \left(\frac{u_{i+\frac{1}{2},j} - u_{i+\frac{1}{2},j-1}}{\Delta y} \right)}{\Delta y}.$$

Likewise, by projecting in the y -direction:

$$\nabla \cdot (\mu \nabla \mathbf{u}) \cdot \mathbf{e}_y \Big|_{i,j+\frac{1}{2}} = \frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial x} \right) \Big|_{i,j+\frac{1}{2}} + \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial y} \right) \Big|_{i,j+\frac{1}{2}}, \quad (18)$$

with:

$$\frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial x} \right) \Big|_{i,j+\frac{1}{2}} \simeq \frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{v_{i+1,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}}{\Delta x} \right) - \mu_{i-\frac{1}{2},j+\frac{1}{2}} \left(\frac{v_{i,j+\frac{1}{2}} - v_{i-1,j+\frac{1}{2}}}{\Delta x} \right)}{\Delta x},$$

and

$$\frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial y} \right) \Big|_{i,j+\frac{1}{2}} \simeq \frac{\mu_{i,j+1} \left(\frac{v_{i,j+\frac{3}{2}} - v_{i,j+\frac{1}{2}}}{\Delta y} \right) - \mu_{i,j} \left(\frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y} \right)}{\Delta y}.$$

In order to solve for the viscous term in the intermediate time step, one must solve a linear system in each direction:

$$a_{i+1,j} u_{i+\frac{3}{2},j}^* + a_{i,j} u_{i-\frac{1}{2},j}^* + a_{i+\frac{1}{2},j+\frac{1}{2}} u_{i+\frac{1}{2},j+1}^* + a_{i+\frac{1}{2},j-\frac{1}{2}} u_{i+\frac{1}{2},j-1}^* + \alpha u_{i+\frac{1}{2},j}^* = F_{x\text{-rhs}},$$

and

$$c_{i+\frac{1}{2},j+\frac{1}{2}} v_{i+1,j+\frac{1}{2}}^* + c_{i-\frac{1}{2},j+\frac{1}{2}} v_{i-1,j+\frac{1}{2}}^* + c_{i,j+1} v_{i,j+\frac{3}{2}}^* + c_{i,j} v_{i,j-\frac{1}{2}}^* + \beta v_{i,j+\frac{1}{2}}^* = F_{y\text{-rhs}},$$

with $F_{x\text{-rhs}}$ and $F_{y\text{-rhs}}$ the right-hand side terms from GFSCMI projection method in Eq. (16), respectively projected onto the x - and y - direction. The coefficients above are given by equations (13), (14) and (15).

As explained before, this formulation is quite straightforward as each component is decoupled from the others. This section is concluded by some comments on how to define the density $\rho_{i,j+\frac{1}{2}}^{n+1}$ and $\rho_{i+\frac{1}{2},j}^{n+1}$, as well as all the viscosity constants $\mu_{i\pm\frac{1}{2},j\pm\frac{1}{2}}$. The definitions must be compatible with the computation of the correction step, which itself relies on the way the density is computed when the Poisson equation for the pressure is solved. We use the accurate computation of the harmonic average when grid cells are crossed by the fluid-fluid interface, as proposed in [21, 25]:

$$\frac{1}{\rho^{n+1}} = \frac{\frac{1}{\rho^+ \rho^-}}{\frac{1}{\rho^+} \theta + \frac{1}{\rho^-} (1 - \theta)} = \frac{1}{\rho^- \theta + \rho^+ (1 - \theta)}.$$

3.4. Velocity Boundary Conditions

On the solid-fluid interface, we impose the following Dirichlet boundary condition to enforce the solid's velocity \mathbf{u}_s on the fluid to ensure the no-slip condition:

$$\mathbf{u}|_{\Gamma_s} = \mathbf{u}_s.$$

In [16], the authors introduced a second-order accurate numerical scheme to enforce a Dirichlet boundary condition for the Poisson equation in irregular domains. We follow this approach to implicitly impose the no-slip condition when considering the viscous term, i.e. in the case of the linear systems for both GFCMI and GFSCMI given in section 3.3.2.

The projection on the x -direction of the stress tensor in the case of GFCMI gives:

$$\nabla \cdot (2\mu \underline{\underline{\mathbf{D}}}) \cdot \mathbf{e}_x \Big|_{i+\frac{1}{2},j} = \frac{\partial}{\partial x} \left(2\mu \frac{\partial u}{\partial x} \right) \Big|_{i+\frac{1}{2},j} + \frac{\partial}{\partial y} \left(\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) \Big|_{i+\frac{1}{2},j}. \quad (19)$$

The discretization of equation (19) gives:

$$\begin{aligned} \nabla \cdot (2\mu \underline{\mathbf{D}}) \cdot \mathbf{e}_x \Big|_{i+\frac{1}{2},j} &\simeq \frac{2\mu_{i+1,j} \left(\frac{\partial u}{\partial x} \right)_{i+1,j} - 2\mu_{i,j} \left(\frac{\partial u}{\partial x} \right)_{i,j}}{\Delta x_i} \\ &+ \frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{\partial u}{\partial y} \right)_{i+\frac{1}{2},j+\frac{1}{2}} - \mu_{i+\frac{1}{2},j-\frac{1}{2}} \left(\frac{\partial u}{\partial y} \right)_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{j-\frac{1}{2}}} \\ &+ \frac{\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{\partial v}{\partial x} \right)_{i+\frac{1}{2},j+\frac{1}{2}} - \mu_{i+\frac{1}{2},j-\frac{1}{2}} \left(\frac{\partial v}{\partial x} \right)_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{j-\frac{1}{2}}}. \end{aligned}$$

Each term is discretized as follows:

- Discretization of $\frac{\partial u}{\partial x} \Big|_{i+1,j}$:

If $\phi_{i+\frac{1}{2},j} \phi_{i+\frac{3}{2},j} > 0$, the cell $[i + \frac{1}{2}, i + \frac{3}{2}] \times [j]$ is entirely covered with fluid. In this case, the approximation is:

$$\frac{\partial u}{\partial x} \Big|_{i+1,j} = \frac{u_{i+\frac{3}{2},j} - u_{i+\frac{1}{2},j}}{\Delta x_{i+1}}.$$

If $\phi_{i+\frac{1}{2},j} \phi_{i+\frac{3}{2},j} \leq 0$, there are two possible situations: either the cell point $(i + \frac{1}{2}, j)$ is in the solid domain, which means that $u_{i+\frac{1}{2},j} = 0$, or the cell point $(i + \frac{3}{2}, j)$ is in the solid domain. In this scenario, the method of [16] is applied, giving:

$$\frac{\partial u}{\partial x} \Big|_{i+1,j} = \frac{u_\Gamma - u_{i+\frac{1}{2},j}}{\theta \Delta x_{i+1}},$$

with u_Γ the boundary velocity condition and $\theta \Delta x_{i+1}$ the length fraction of the cell covered by the fluid, hence:

$$\theta = \frac{|\phi_{i+\frac{1}{2},j}|}{|\phi_{i+\frac{1}{2},j}| + |\phi_{i+\frac{3}{2},j}|}.$$

The interpolated length fraction is computed on a staggered cell point in order to keep second-order accuracy with:

$$\phi_{i+\frac{1}{2},j} = \frac{\phi_{i+1,j} + \phi_{i,j}}{2} \quad \text{and} \quad \phi_{i+\frac{3}{2},j} = \frac{\phi_{i+2,j} + \phi_{i+1,j}}{2}.$$

- Discretization of $\frac{\partial u}{\partial x} \Big|_{i,j}$:

If $\phi_{i+\frac{1}{2},j} \phi_{i-\frac{1}{2},j} > 0$ the cell $[i - \frac{1}{2}, i + \frac{1}{2}] \times [j]$ is entirely covered with fluid. In this case, the approximation is:

$$\frac{\partial u}{\partial x} \Big|_{i,j} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x_i}.$$

If $\phi_{i+\frac{1}{2},j}\phi_{i-\frac{1}{2},j} \leq 0$, there are two possible situations: either the cell point $(i + \frac{1}{2}, j)$ is in the solid domain, which means that $u_{i+\frac{1}{2},j} = 0$, or the cell point $(i - \frac{1}{2}, j)$ is in the solid domain. In this case, the method of [16] is applied, giving:

$$\frac{\partial u}{\partial x} \Big|_{i,j} = \frac{u_{i+\frac{1}{2},j} - u_{\Gamma}}{\theta \Delta x_i}, \quad \text{with } \theta = \frac{|\phi_{i+\frac{1}{2},j}|}{|\phi_{i+\frac{1}{2},j}| + |\phi_{i-\frac{1}{2},j}|}.$$

- Discretization of $\frac{\partial u}{\partial y} \Big|_{i+\frac{1}{2},j+\frac{1}{2}}$:

If $\phi_{i+\frac{1}{2},j+1}\phi_{i+\frac{1}{2},j} > 0$, the cell $[i + \frac{1}{2}] \times [j, j + 1]$ is entirely covered with fluid. In this case, the approximation is:

$$\frac{\partial u}{\partial y} \Big|_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j}}{\Delta y_{j+\frac{1}{2}}}.$$

If $\phi_{i+\frac{1}{2},j+1}\phi_{i+\frac{1}{2},j} \leq 0$, there are two possible situations: either the cell point $(i + \frac{1}{2}, j)$ is in the solid domain, which means that $u_{i+\frac{1}{2},j} = 0$, or the cell point $(i + \frac{1}{2}, j + 1)$ is in the solid domain. In this case, the method of [16] is applied, giving:

$$\frac{\partial u}{\partial y} \Big|_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{u_{\Gamma} - u_{i+\frac{1}{2},j}}{\theta \Delta y_{j+\frac{1}{2}}}, \quad \text{with } \theta = \frac{|\phi_{i+\frac{1}{2},j}|}{|\phi_{i+\frac{1}{2},j}| + |\phi_{i+\frac{1}{2},j+1}|}.$$

- Discretization of $\frac{\partial u}{\partial y} \Big|_{i+\frac{1}{2},j-\frac{1}{2}}$:

If $\phi_{i+\frac{1}{2},j-1}\phi_{i+\frac{1}{2},j} > 0$, the cell $[i + \frac{1}{2}] \times [j, j - 1]$ is entirely covered with fluid. In this case, the approximation is:

$$\frac{\partial u}{\partial y} \Big|_{i+\frac{1}{2},j-\frac{1}{2}} = \frac{u_{i+\frac{1}{2},j} - u_{i+\frac{1}{2},j-1}}{\Delta y_{j-\frac{1}{2}}}.$$

If $\phi_{i+\frac{1}{2},j-1}\phi_{i+\frac{1}{2},j} \leq 0$, there are two possible situations: either the cell point $(i + \frac{1}{2}, j)$ is in the solid domain, which means that $u_{i+\frac{1}{2},j} = 0$, or the cell point $(i + \frac{1}{2}, j - 1)$ is in the solid domain. In this case, the method of [16] is applied, giving:

$$\frac{\partial u}{\partial y} \Big|_{i+\frac{1}{2},j-\frac{1}{2}} = \frac{u_{i+\frac{1}{2},j} - u_{\Gamma}}{\theta \Delta y_{j-\frac{1}{2}}}, \quad \text{with } \theta = \frac{|\phi_{i+\frac{1}{2},j}|}{|\phi_{i+\frac{1}{2},j}| + |\phi_{i+\frac{1}{2},j-1}|}.$$

Let us now focus on the cross derivatives:

- Discretization of $\frac{\partial v}{\partial x} \Big|_{i+\frac{1}{2},j+\frac{1}{2}}$:

If $\phi_{i+1,j+\frac{1}{2}}\phi_{i,j+\frac{1}{2}} > 0$, the cell $[i, i + 1] \times [j + \frac{1}{2}]$ is entirely covered with fluid. In this case, the approximation is:

$$\frac{\partial v}{\partial x} \Big|_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{v_{i+1,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}.$$

If $\phi_{i+1,j+\frac{1}{2}}\phi_{i,j+\frac{1}{2}} \leq 0$, there are two possible situations: either the cell point $(i+1, j+\frac{1}{2})$ is in the solid domain. In this case, the method of [16] is applied, giving:

$$\left. \frac{\partial v}{\partial x} \right|_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{v_\Gamma - v_{i,j+\frac{1}{2}}}{\theta \Delta x_{i+\frac{1}{2}}}, \quad \text{with} \quad \theta = \frac{|\phi_{i,j+\frac{1}{2}}|}{|\phi_{i+1,j+\frac{1}{2}}| + |\phi_{i,j+\frac{1}{2}}|}.$$

or the cell point $(i, j+\frac{1}{2})$ is in the solid domain and:

$$\left. \frac{\partial v}{\partial x} \right|_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{v_{i+1,j+\frac{1}{2}} - v_\Gamma}{\theta \Delta x_{i+\frac{1}{2}}}, \quad \text{with} \quad \theta = \frac{|\phi_{i+1,j+\frac{1}{2}}|}{|\phi_{i+1,j+\frac{1}{2}}| + |\phi_{i,j+\frac{1}{2}}|}.$$

- Discretization of $\left. \frac{\partial v}{\partial x} \right|_{i+\frac{1}{2},j-\frac{1}{2}}$:

If $\phi_{i+1,j-\frac{1}{2}}\phi_{i,j-\frac{1}{2}} > 0$, the cell $[i, i+1] \times [j-\frac{1}{2}]$ is entirely covered with fluid. In this case, the approximation is:

$$\left. \frac{\partial v}{\partial x} \right|_{i+\frac{1}{2},j-\frac{1}{2}} = \frac{v_{i+1,j-\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}.$$

If $\phi_{i+1,j-\frac{1}{2}}\phi_{i,j-\frac{1}{2}} \leq 0$, there are two possible situations: either the cell point $(i+1, j-\frac{1}{2})$ is in the solid domain. In this case, the method of [16] is applied, giving:

$$\left. \frac{\partial v}{\partial x} \right|_{i+\frac{1}{2},j-\frac{1}{2}} = \frac{v_\Gamma - v_{i,j-\frac{1}{2}}}{\theta \Delta x_{i+\frac{1}{2}}}, \quad \text{with} \quad \theta = \frac{|\phi_{i,j-\frac{1}{2}}|}{|\phi_{i+1,j-\frac{1}{2}}| + |\phi_{i,j-\frac{1}{2}}|}.$$

or the cell point $(i, j-\frac{1}{2})$ is in the solid domain and:

$$\left. \frac{\partial v}{\partial x} \right|_{i+\frac{1}{2},j-\frac{1}{2}} = \frac{v_{i+1,j-\frac{1}{2}} - v_\Gamma}{\theta \Delta x_{i+\frac{1}{2}}}, \quad \text{with} \quad \theta = \frac{|\phi_{i+1,j-\frac{1}{2}}|}{|\phi_{i+1,j-\frac{1}{2}}| + |\phi_{i,j-\frac{1}{2}}|}.$$

The procedure is easier for GFSCMI, as the cross derivative terms are considered explicitly as a jump condition in the pressure resolution instead of implicitly.

3.5. Imposing the Contact Angle

When both the liquid-gas interface and the fluid-solid interfaces are located in the same cell, one needs to compute the surface tension effect on the irregular domain boundaries. Modeling accurately the contact line on a wall is challenging because the surface physics is not currently completely understood. In this study we only focus on static contact angles based on thermodynamic properties. Modeling dynamic contact angles is out of the scope of this paper for the reasons that are provided in section 3.6.

When the equilibrium state is reached, the static contact angle θ_s (see figure 1) can be used to obtain a Neumann boundary condition on the fluid interface level-set function:

$$\mathbf{n}_s \cdot \nabla \phi = \cos \theta_s \quad \text{with} \quad \mathbf{n}_s = \frac{\nabla \phi_s}{|\nabla \phi_s|}.$$

As θ_s is known from the properties of the flow, one needs to enforce this condition on ϕ when a contact line appears. Our method is based on the multidimensional extrapolation suggested by [3, 14] to define the fluid level-set function's extrapolation in the irregular domain:

$$\frac{\partial \phi}{\partial \tau} + H(\phi_s) \mathbf{n}_s \cdot \nabla \phi = \cos \theta_s, \quad (20)$$

where τ is a fictitious time step, and $H(\phi_s)$ a Heaviside function. Equation (20) is only applied on the cells in solid domains. A second order upwind scheme was used for the discretization of $\mathbf{n}_s \cdot \nabla \phi$.

The curvature also needs to be extrapolated to properly handle the pressure jump condition in cells where the three phases are present. The extrapolation process is the same that the one used for the level-set function in equation (21), i.e.:

$$\frac{\partial \kappa}{\partial \tau} + H(\phi_s) \mathbf{n}_s \cdot \nabla \kappa = 0, \quad (21)$$

where we recall that κ is the local mean curvature.

We note that this procedure is similar but different from that of [26], in which the authors consider a moving contact line with a reconstruction of the level-set function up to $2.5\Delta x$ in the fluid domain in order to keep a smooth computation of the local curvature. This procedure thus alters the flows near the solid-fluid interface. In our computation of the static contact angle, the extension of the local curvature is in the solid media only and does not affect the flow.

3.6. Advancing and Receding Contact Line

It is well known that the contact angle of a droplet or a bubble can vary when they move on a solid surface. In particular, it has been observed in several experimental studies that the value of the contact angle depends on the interface velocity and on its sign, i.e. the contact angle will take different value depending on whether the interface is advancing or receding, all other physical parameters being equal. Moreover, the dynamics of the interface will not only depend on the physicochemical properties of the fluid and of the solid, but also on the surface quality, which can be characterized for example by its roughness. De Gennes showed that roughness can induce a hysteresis phenomenon on the contact angle [15]. In [42], the author presents simulations involving contact angle hysteresis enforced on a solid boundary fitted to the computational domain.

Although many studies have been performed on this difficult topic [1, 40, 41, 42, 43], there is a lack of consensus on how to handle a moving contact angle in numerical simulations in a way that does not depend on the grid cell size. For this reason, we do not propose in this paper any numerical models to account for a moving contact line. Indeed, this topic is out of the scope of this paper, since this problem should be solved accurately on simplest configurations without irregular domains (horizontal wall fitted to the computational domain) before performing an implementation with irregular domains. Consequently, the validations of our numerical simulations will involve configurations with a steady shape of the interface. We note, however, that considering a static contact line enables us to illustrate the extend to which the method limits spurious parasitic currents and thus illustrate its numerical stability.

4. Numerical Examples and Validations

In this section, we first provide numerical examples that test the two different viscosity solvers (GFCMI and GFSCMI). Because of the lack of analytical solutions for two-phase flows, we consider

one-phase flows. We then turn our attention to two-phase flows and report numerical results in two and three spatial dimensions.

4.1. Flow Past a Cylinder

We consider a flow past a cylinder in order to show that the different discretizations are able to compute both steady and unsteady regimes. Coutanceau and Bouard [11] showed in an experiment that a transition regime exists between Reynolds numbers 40 and 50. Consider a domain $\Omega = [-16; 16] \times [-8; 8]$ and a cylinder with a radius of $r = 0.5$ and center at $[0, 0]$. We impose a Dirichlet condition $U_\infty = 1 \text{ m} \cdot \text{s}^{-1}$ on the left, symmetric boundary conditions on the top and bottom, and an outflow boundary condition on the right of the domain, with a no-slip condition on the cylinder boundary. In these simulations we define the viscosity coefficient as $\mu = 2rU_\infty/Re$, with $Re = 20, 40$ and 100 , corresponding respectively to the case A, B and C. Experimental and numerical experiments from others allow us to compare the total drag, C_D , and lift, C_L , forces acting on the cylinder:

$$\mathbf{F}_{f \rightarrow s} = \int_{\theta=0}^{2\pi} \underline{\underline{\sigma}} \mathbf{n} R d\theta \quad \text{with} \quad \underline{\underline{\sigma}} = -p\mathbf{II} + 2\mu\underline{\underline{D}},$$

giving the drag and lift coefficients:

$$C_D = \frac{F_x}{\frac{1}{2}\rho U_\infty^2 D} \quad \text{and} \quad C_L = \frac{F_y}{\frac{1}{2}\rho U_\infty^2 D}.$$

For steady flows, one can also measure some characteristics of the vortices in the wake of the cylinder, such as their centers position (x_c, y_c) and the length, x_s , of the recirculation as illustrated in Fig 3.

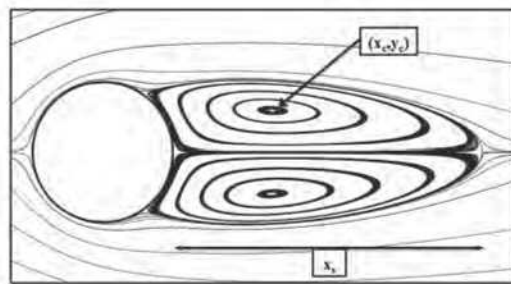


Figure 3: Streamlines for a flow past a cylinder at $Re = 40$. Characteristics such as the location of the vortices center (x_c, y_c) and the length, x_s , of the recirculation are represented.

We will present here some comparisons for a one-phase flow between the Implicit Conservative Viscous Method (ICVM) and the Implicit Primitive Viscous Method (IPVM) which corresponds respectively to GFCMI and to GFSCMI for a two-phase flow. Indeed, most studies involving one-phase flows use a numerical scheme similar to the IPVM [32] to deal with viscous terms. However as it has been discussed previously, in the framework of two-phase flows, a numerical scheme based on a conservative formulation of the viscous terms can be preferred for temporal stability and accuracy (see Appendix for more details).

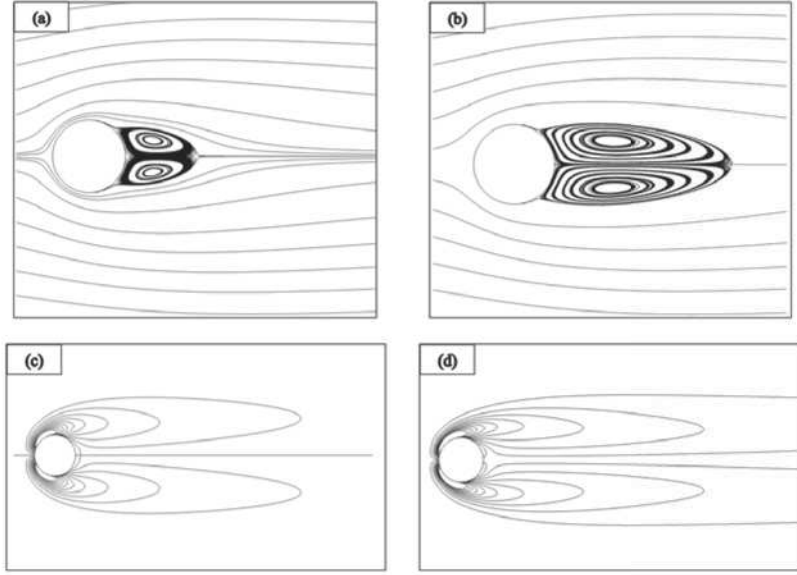


Figure 4: Streamline representation and vorticity contours (s^{-1}) $[-.5; .5; 5]$ for Case A (a,c) and B (b,d).

Therefore, this test-case will allow us to check that both methods will provide similar results in order to validate the ICVM for a one-phase flow before performing computations of GFCMI in irregular domains. The simulations have been performed with three different grids, 64×128 , 128×256 and 256×512 , which correspond respectively to $\Delta x = D/8$, $\Delta x = D/16$ and $\Delta x = D/32$.

	Re = 20				Re = 40			
	C_D	x_s	x_c	y_c	C_D	x_s	x_c	y_c
Coutanceau & Bouard [11]	-	0.93	0.33	0.23	-	2.13	0.76	0.29
Xu [56]	2.23	0.93	0.36	0.21	1.66	2.24	0.72	0.3
Ye <i>et al.</i> [58]	2.03	0.92	-	-	1.52	2.27	-	-
Xie & Desjardin [55]	2.19	1	-	-	1.60	2.26	-	-
Tritton [50]	2.22	-	-	-	1.48	-	-	-
Present study with ICVM								
$\Delta x = D/8$	2.25	0.9	0.318	0.217	1.64	2.24	0.714	0.297
$\Delta x = D/16$	2.214	0.942	0.344	0.223	1.628	2.251	0.709	0.299
$\Delta x = D/32$	2.224	0.937	0.351	0.218	1.614	2.227	0.706	0.297
Present study with IPVM								
$\Delta x = D/8$	2.318	0.916	0.317	0.221	1.687	2.255	0.713	0.3
$\Delta x = D/16$	2.239	0.946	0.344	0.224	1.639	2.257	0.709	0.3
$\Delta x = D/32$	2.247	0.94	0.351	0.218	1.625	2.228	0.706	0.298

Table 1: Drag coefficient and recirculations characteristics for $Re = 20$ and $Re = 40$.

Streamlines and vorticity contours have been plotted in figure 4 for cases *A* and *B* corresponding

to $Re = 20$ and $Re = 40$, respectively. Steady vortices are attached to the rear of the cylinder. These vortices are growing if the Reynolds number is increased. In table 1, some comparisons between our numerical simulations and several numerical and experimental studies are presented. Our two schemes (ICVM and IPVM) provide similar results and converge to an expected value if the size of the grid cell is at least equal to $\Delta x = D/16$.

In the case $Re = 100$, the flow in the wake of the cylinder is unsteady due to the development of a Von-Kármán instability. The vorticity contours have been plotted in figure 5 in order to visualize the periodic detachment of vortices in the wake of the cylinder. The temporal evolution of the drag and the lift coefficients are also presented in figure 5. Quantitative comparisons between our numerical simulations and other data from the literature are reported in table 2. These comparisons show a satisfactory agreement between all results.

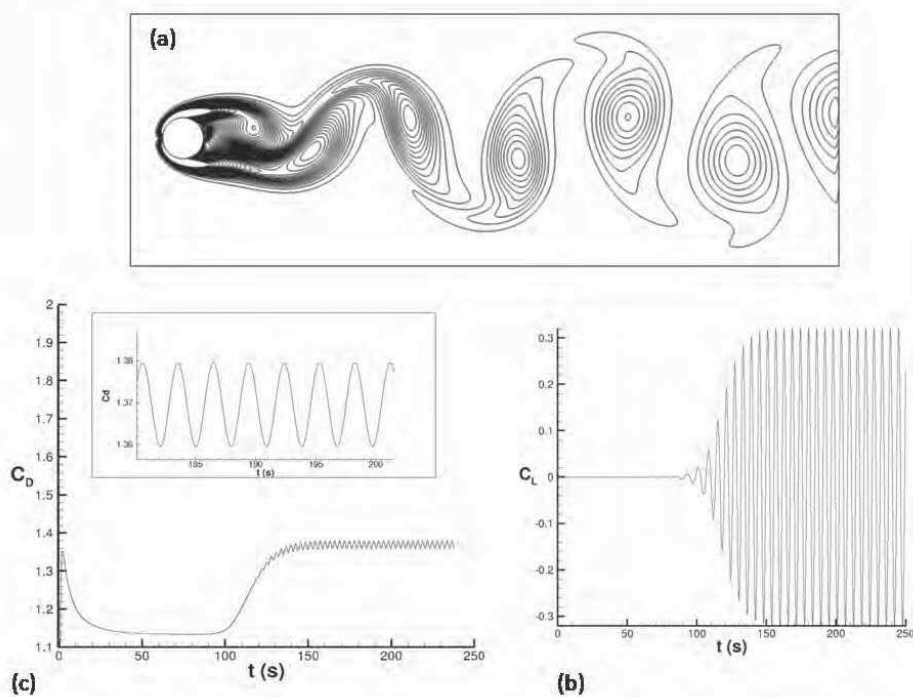


Figure 5: (a) Vorticity contours $[-4 : .15 : 4]$ for Case C. (b) and (c) represent the evolution of the drag and lift coefficients through time.

4.2. Flow Past a Sphere

We now consider the simulation of the flow past a sphere. This study will allow us to show that our numerical simulations are able to reproduce 3D steady axisymmetric and 3D non-axisymmetric steady flows. Consider a domain $\Omega = [-8, 8] \times [-4, 4] \times [-4, 4]$ and a sphere with radius $r = 0.5$ and center located at $(0, 0)$. We impose a Dirichlet condition $U_\infty = 1 \text{ m} \cdot \text{s}^{-1}$ on the left, symmetric conditions on the top, bottom, front and back walls, and an outflow boundary condition on the

	$Re = 100$	
	C_L	C_D
Braza [6] <i>et al.</i>	± 0.25	1.346 ± 0.015
Liu [26] <i>et al.</i>	± 0.339	1.35 ± 0.012
Calhoun [8]	± 0.298	1.330 ± 0.014
Ng <i>et al.</i> [32]	± 0.36	1.368 ± 0.016
Guittet <i>et al.</i> [18]	± 0.331	1.401 ± 0.016
Xu [56]	± 0.353	1.42 ± 0.01
Choi [9] <i>et al.</i>	± 0.315	1.4 ± 0.011
Present study with ICVM		
$\Delta x = D/8$	± 0.261	1.404 ± 0.008
$\Delta x = D/16$	± 0.334	1.419 ± 0.011
$\Delta x = D/32$	± 0.321	1.369 ± 0.010
Present study with IPVM		
$\Delta x = D/8$	± 0.262	1.430 ± 0.008
$\Delta x = D/16$	± 0.336	1.424 ± 0.011
$\Delta x = D/32$	± 0.322	1.376 ± 0.010

Table 2: Drag and lift coefficients for $Re = 100$ for a flow past a cylinder.

right side of the computational field. A no-slip boundary condition is imposed on the sphere's boundary. In these simulations, we define the viscosity coefficient as $\mu = 2rU_\infty/Re$, with $Re = 25, 50, 100, 150$ and 200 for steady axisymmetric flows as depicted by figure 6, and 250 for the steady non-axisymmetric flow.

The drag and lift coefficients are defined as:

$$\mathbf{F}_{f \rightarrow s} = \int_{\theta=0}^{\pi} \int_{\Psi=0}^{2\pi} R^2 \sin \theta \underline{\underline{\sigma}} \mathbf{n} d\theta d\Psi,$$

$$C_D = \frac{F_x}{\frac{1}{2}\rho U_\infty^2 \pi D^2/4} \quad \text{and} \quad C_L = \frac{F_y}{\frac{1}{2}\rho U_\infty^2 \pi D^2/4}.$$

	$Re = 25$	$Re = 50$	$Re = 100$	$Re = 150$	$Re = 200$
Schiller-Naumann	2.27	1.53	1.09	0.91	0.80
ICVM Method					
$\Delta x = D/8$	2.27	1.49	1.051	0.88	0.78
$\Delta x = D/16$	2.23	1.464	1.017	0.84	0.74
IPVM Method					
$\Delta x = D/8$	2.31	1.50	1.052	0.87	0.77
$\Delta x = D/16$	2.268	1.48	1.024	0.84	0.74

Table 3: Convergence of drag coefficient for a flow past a sphere, for ICVM and IPVM for different Reynolds numbers in the steady regime.

Figure 7 gives some comparisons on the drag coefficient obtained from several published works, including the experimental studies of [20, 38], numerical simulations, and the semi-empiric correla-

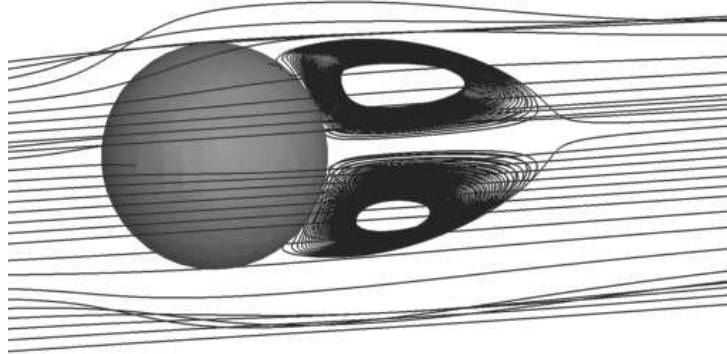


Figure 6: Illustration of streamlines for a flow past a sphere at $Re = 150$.

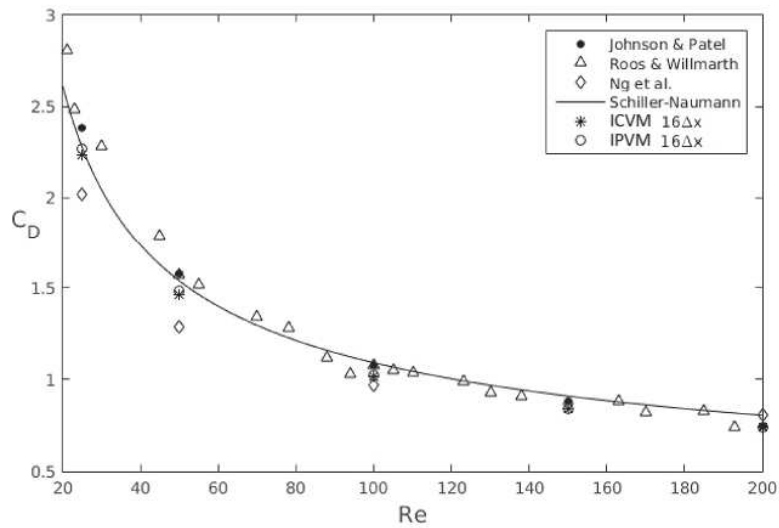


Figure 7: Drag coefficient evolution versus the Reynolds number.

tion of Schiller-Naumann. This correlation is given by the following relation:

$$C_D = \begin{cases} \frac{24(1 + 0.15Re^{0.687})}{Re} & \text{for } Re \leq 1000, \\ 0.44 & \text{for } Re > 1000. \end{cases}$$

Simulations for both IPVM and ICVM are presented for $\Delta x = D/16$ which corresponds to a grid with 128^3 cells. We can observe a good agreement of our numerical simulations with the results reported in the literature. In table 3, we present some comparisons of the drag coefficient between our simulations for two grid resolutions and the Schiller-Naumann law. These comparisons show

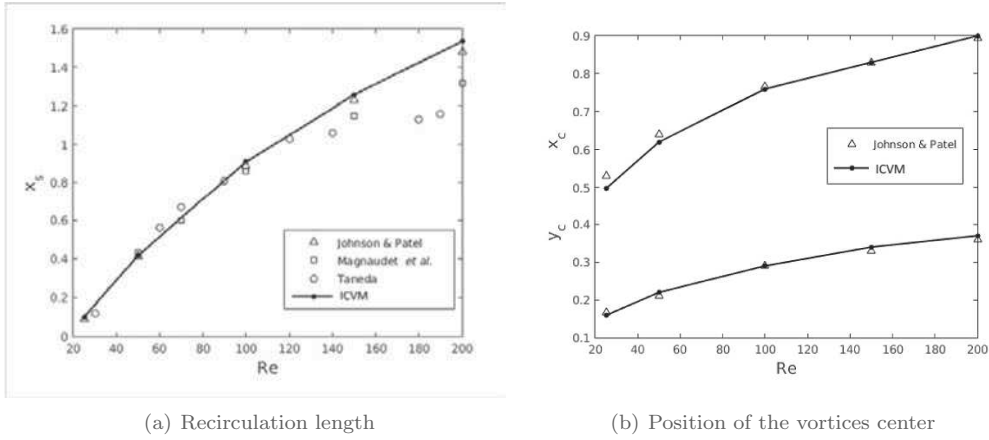


Figure 8: Geometric characteristics of the flow past a sphere for different Reynolds number

that the two numerical methods converge towards the same solutions with a slight discrepancy ($< 10\%$) with the Schiller-Naumann correlation.

Figure 8 presents a comparison of the recirculation length as well as the position of the center of the vortices between our simulation and the experimental results from Johnson & Patel [20], Taneda [46], Magnaudet [27]. In this case, we use the ICVM method with $\Delta x = D/16$. One can observe good agreement. Figure 9 compares the streamlines and vorticity contours from Johnson and Patel [20] and our simulations for $Re = 150$.

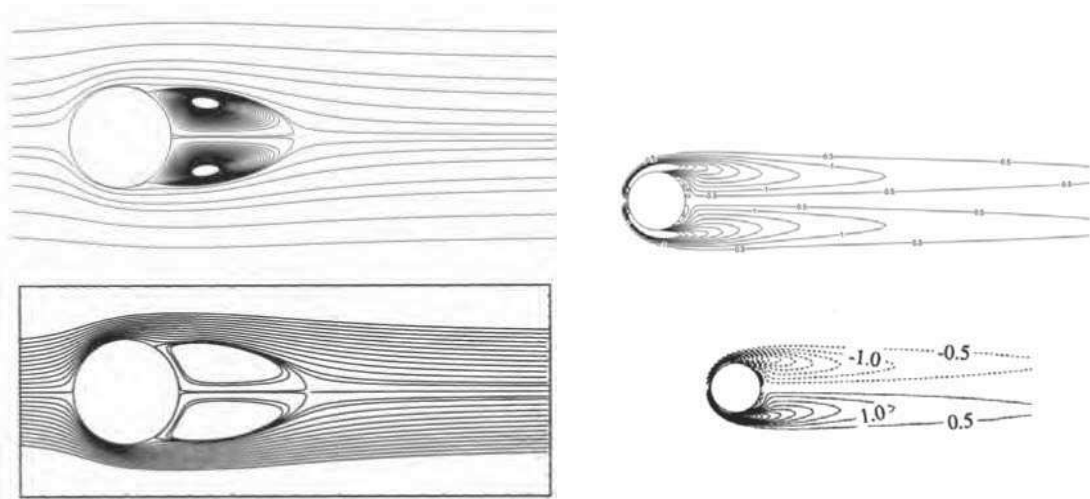


Figure 9: Top: streamlines (left) and vorticity (right) contours $[-5:5:5]$ in the $x-y$ plane for $Re = 150$. Bottom: corresponding results from Johnson & Patel [20].

Figure 10 compares the streamlines results obtained with our simulations and those of Johnson

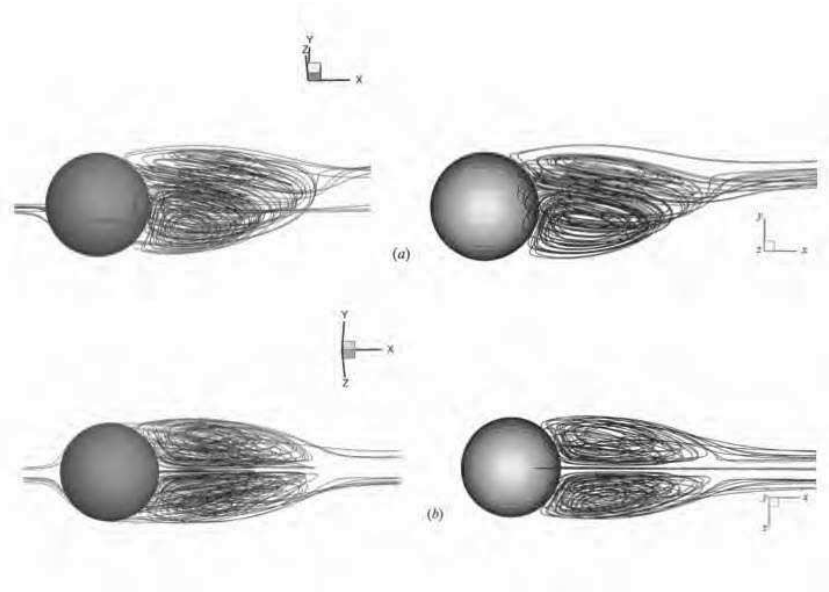


Figure 10: Illustration of streamlines for a flow past a sphere at $Re = 250$. On the left present results and on the right corresponding results from [20].

and Patel [20] for $Re = 250$. We can observe the expected instability and that the streamlines are in good agreement. Figure 11 presents some projected streamlines on different planes. The qualitative comparison for this non-axisymmetric case are in good agreement with [20] for both IPVM and ICVM. In conclusion, one can observe that IPVM and ICVM behave similarly in terms of numerical accuracy.

4.3. Two Dimensional Two-Phase Flows in Irregular Domains without Volume Forces

In this section, we present some numerical simulations of a hemispherical droplet deposited on a wall without initial velocity. These simulations are performed with a horizontal solid boundary (aligned to mesh axes) but also with a slanted plane, see figure 12, in order to demonstrate the ability of the proposed numerical method to obtain the expected shape of the droplet deposited on arbitrary solid boundaries. This configuration is instigated by the previous work of Dupont [13] which has been performed with a horizontal solid boundary located on the boundary of the computational domain. An additional difficulty suggested in [33] is to consider embedded solid boundary instead of using the computational domain as solid boundaries. If the static contact angle is not 90° the contact line will move until equilibrium is reached. If the gravity is zero, the exact final shape of the drop can be obtained due to geometrical considerations and mass conservation.

The numerical domain is $\Omega = [-3R_0, 3R_0] \times [-3R_0, 3R_0]$. The simulations are carried out with both GFSCMI and GFCMI, with a horizontal wall, and with a slanted wall (for $\alpha = 40^\circ$), with $\theta_s \in [30, 120]$, and the following parameters: $R_0 = 6.4 \times 10^{-4}$ m, $\sigma = 7.2 \times 10^{-2}$ N · m⁻¹,

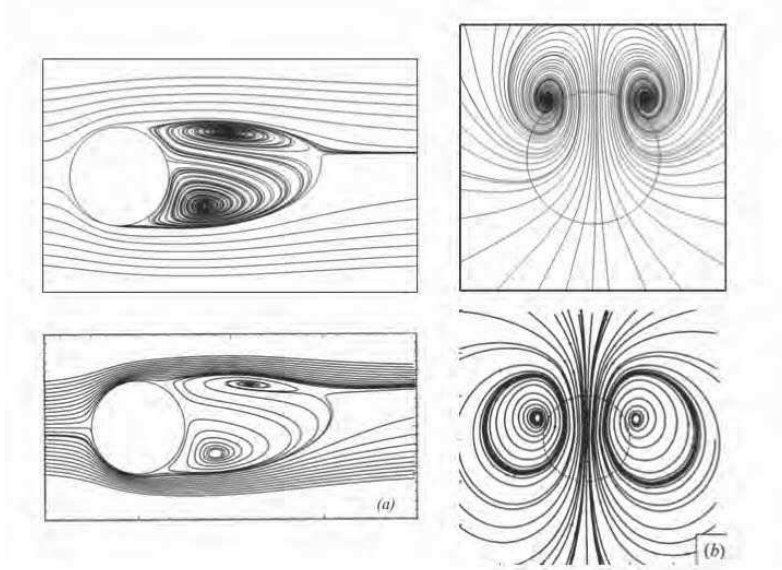


Figure 11: Illustration of streamlines for a flow past a sphere at $Re = 250$, on top our simulation and bottom corresponding results from [20]. (a) Streamlines projected on the on the x-y plane for [20], and (b) streamlines projected on the y-z plane 8 diameters downstream from the sphere center.

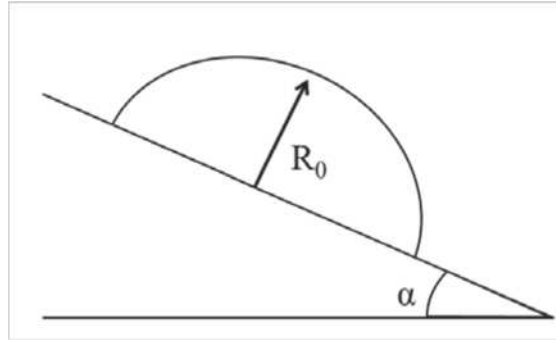


Figure 12: Initial shape of the drop released on an slanted plan.

$\rho_L = 1000 \text{ kg} \cdot \text{m}^{-3}$ and $\mu_L = 10^{-2} \text{ kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$ for the liquid and $\rho_G = 1 \text{ kg} \cdot \text{m}^{-3}$ and $\mu_G = 10^{-5} \text{ kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$ for the gas.

For 2D cases without volume forces, mass conservation gives:

$$R = R_0 \sqrt{\frac{\pi}{2(\theta_s - \sin \theta_s \cos \theta_s)}}, \quad L = 2R \sin \theta_s, \quad e = R(1 - \cos \theta_s),$$

with R the equilibrium radius of the circle, L the spreading length, e the height of the drop, as illustrated in figure 13.

Figure 14 illustrates the shapes of two drops at equilibrium on a slanted wall for two static

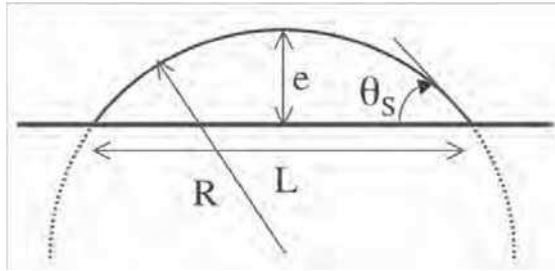


Figure 13: Characteristics of the final drop shape (from [13]).

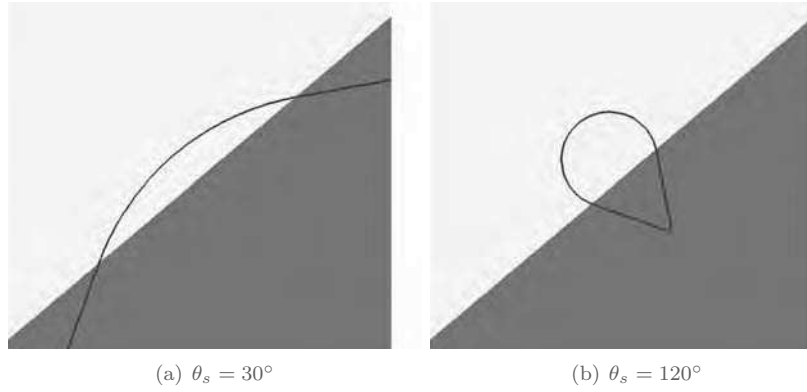


Figure 14: Shapes of 2D drops on a slanted wall for different values of the contact angle.

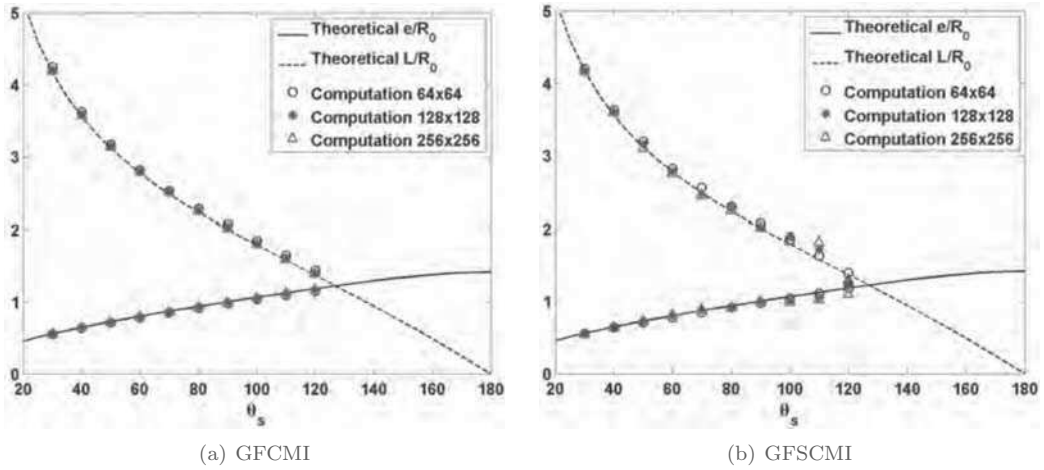


Figure 15: Results on a flat wall.

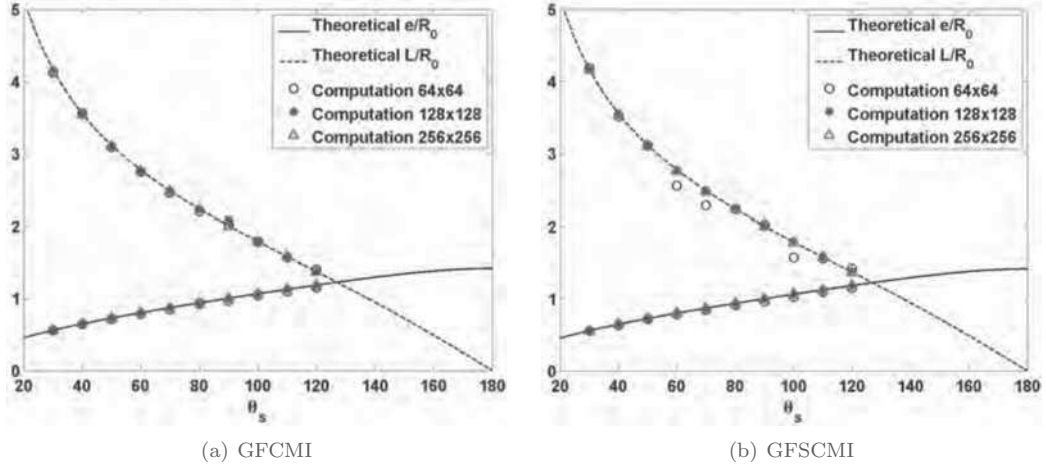


Figure 16: Results on an inclined wall ($\alpha = 40^\circ$).

contact angles, $\theta_s = 30^\circ$ and $\theta_s = 120^\circ$. In figure 15 and figure 16, one can observe the convergence to expected theoretical values. Figure 15 gives the results for a wall aligned to the mesh, while figure 16 shows non-aligned wall to the mesh. Both methods converge towards the correct solution, with a gain in accuracy is observed with GFCMI. In order to compare more accurately the two methods, a study on spurious currents and mass conservation for different static contact angles is presented in table 4.

		GFCMI				GFSCMI			
		e	L	$\ V_{max}\ $	%Vol	e	L	$\ V_{max}\ $	%Vol
$\theta = 30^\circ$	64^2	0.557	4.122	0.00224	0.975	0.553	4.181	0.0065	1
	128^2	0.555	4.158	0.0015	0.998	0.559	4.16	0.015	1.002
	256^2	0.556	4.136	0.00059	1	0.561	4.153	0.011	1.005
	theoretical	0.558	4.164	-	-	0.558	4.164	-	-
$\theta = 60^\circ$	64^2	0.784	2.748	0.003	0.977	0.784	2.55	0.0055	0.97
	128^2	0.794	2.753	0.0071	0.990	0.794	2.766	0.0125	0.992
	256^2	0.799	2.752	0.0015	0.999	0.799	2.755	0.0121	0.995
	theoretical	0.799	2.76	-	-	0.799	2.76	-	-
$\theta = 90^\circ$	64^2	0.961	2.014	0.0083	0.962	0.965	2.026	0.009	0.97
	128^2	0.992	2.08	0.0034	0.991	0.99	1.999	0.012	0.987
	256^2	0.999	1.998	0.0012	1	0.997	2.002	0.007	0.995
	theoretical	1	2	-	-	1	2	-	-
$\theta = 120^\circ$	64^2	1.151	1.397	0.012	0.975	1.147	1.418	0.028	0.976
	128^2	1.164	1.369	0.0059	0.98	1.167	1.380	0.0105	0.985
	256^2	1.181	1.363	0.0061	0.996	1.185	1.363	0.0061	0.996
	theoretical	1.18	1.365	-	-	1.18	1.365	-	-

Table 4: Comparison of characteristics of GFCMI and GFSCMI for a slanted wall

In order to quantify the accuracy of the numerical methods, we have performed a mesh sensitivity study with three different grids for different contact angles in the case of a slanted wall. The results are presented in table 4, where the value of e , L , $|V_{max}|$ and the volume loss are reported in the case of GFCMI and GFSCMI. The maximum velocity is reported when the drop reaches its equilibrium shape, hence enabling the quantification of spurious currents. Ghost-Fluid methods and more generally sharp-interface methods [21, 45] are well known to produce low amplitude spurious currents in comparison to continuum-surface-force methods. However in the framework of a coupling between a two-phase flow solver with an embedded solid boundary, the problem of spurious currents can be much more stringent on the contact line. Consequently, this parameter is of great importance to assess the relevance of such numerical methods. In figure 17, the temporal evolution of the maximum velocity is given, indicating that the spurious currents decrease with the mesh refinement.

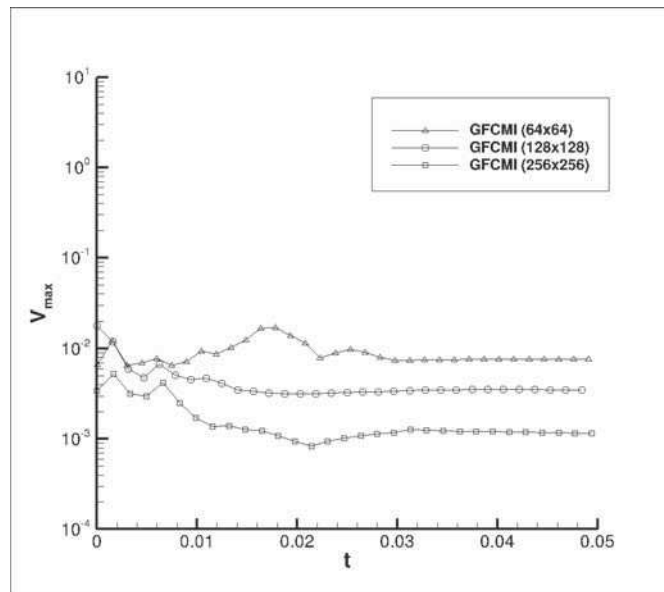


Figure 17: An example of the evolution of $|V_{max}|$ throughout time on an inclined wall ($\alpha = 40^\circ$) and $\theta_s = 90^\circ$ with GFCMI .

The results presented in the table 4 demonstrate that the two methods (GFSCMI and GFCMI) converge toward the expected solution. In some situations, when considering a specific parameter, the errors can increase at one point when the grid is refined. In our opinion, this singular behavior should not be alarming since it can be assuredly explained by the compensation of errors acting in an opposite way. This phenomenon occurs frequently when numerical simulations involving complex and strongly coupled mathematical models, see [48] for example. On the one hand GFCMI and GFSCMI provide results of equivalent accuracy if geometrical properties are considered (e , L , volume loss). On the other hand GFCMI provides an important gain in accuracy on the amplitude of parasitic currents. Indeed, if the latter is used, the amplitude of the maximum velocity is 5 to 20 times smaller on the finest grid (except if $\theta = 120^\circ$ where the two methods provide similar results).

4.4. Two Dimensional Two-Phase Flows in Irregular Domains with Volume Forces

In this section, the previous study is completed by applying different gravity fields, hence changing the final equilibrium drop's shape. The Bond number ($Bo = \rho_L g R_0^2 / \sigma$) characterizes the gravitational stress applied to the drop, with $Bo \in [0, 10]$. Results are presented for $\theta_s = 60^\circ$ and $\theta_s = 120^\circ$. Two asymptotic solutions are provided, with one being dominated by capillarity effects and the other by gravity. If the shape of the drop is driven by the surface tension effects, its thickness, e_0 , is deduced from:

$$e_0 = R_0(1 - \cos \theta_s) \sqrt{\frac{\pi}{2(\theta_s - \sin \theta_s \cos \theta_s)}}. \quad (22)$$

For $Bo \gg 1$, the drop spreads under the effect of gravity and forms a small puddle with a thickness, e_∞ , directly proportional to the capillary length:

$$e_\infty = 2 \sqrt{\frac{\sigma}{\rho_L g}} \sin\left(\frac{\theta_s}{2}\right). \quad (23)$$

Figure 18 illustrates the shape of a drop at equilibrium on a slanted wall for a static contact angle of $\theta_s = 60^\circ$ and for three different Bond numbers. The simulations are found to be in good agreement with the two asymptotic solutions (22) and (23). One can observe that the transition between these two regimes is approximately for $Bo = 1$, when gravity and surface tension effects are of the same order of magnitude. GFCMI converges faster, as illustrated in figures 19 and 21. In order to compare more accurately the two methods, details on the simulations are provided in tables 5 and 6. An illustration of the sharp pressure field is also available in figure 20.

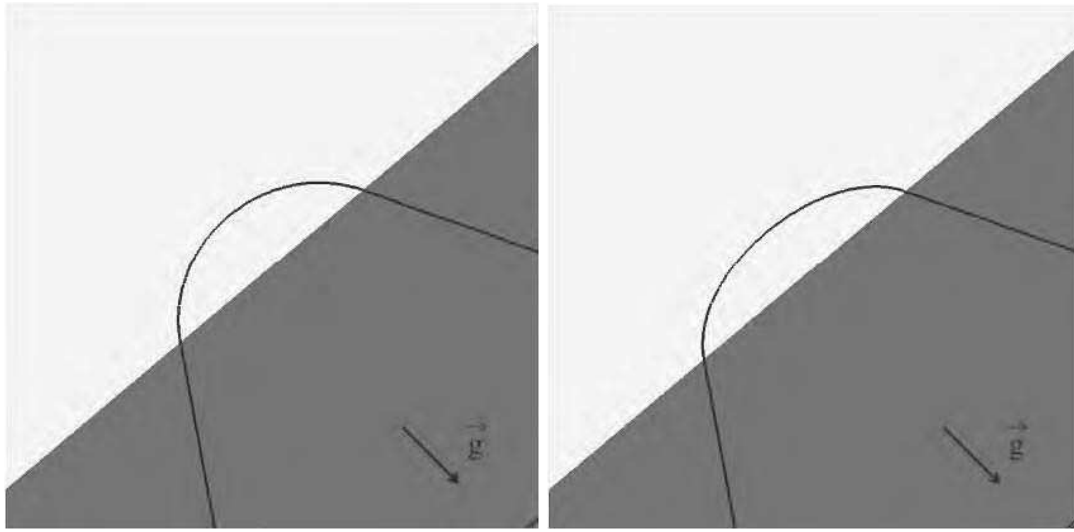
		GFCMI			GFSCMI		
		e	$\ V_{max}\ $	%Vol	e	$\ V_{max}\ $	%Vol
$E_0 = 0.1$	64^2	0.966	0.024	0.976	0.967	0.017	0.978
	128^2	0.977	0.0082	0.991	0.979	0.0105	0.990
	256^2	0.986	0.0011	0.998	0.985	0.010	0.995
$E_0 = 1$	64^2	0.848	0.02	0.969	0.839	0.06	0.963
	128^2	0.860	0.015	0.987	0.854	0.06	0.980
	256^2	0.869	0.011	0.996	0.862	0.05	0.986
$E_0 = 10$	64^2	0.401	0.18	0.962	0.365	0.15	0.970
	128^2	0.376	0.14	0.968	0.383	0.17	0.995
	256^2	0.395	0.0102	0.982	0.390	0.2	0.952

Table 5: Comparison of characteristics of GFCMI and GFSCMI for a slanted wall ($\alpha = 40^\circ$) and for $\theta_s = 60^\circ$

Both methods converge to the same values when the mesh is refined, but as it is the case for the previous example, GFCMI gives better results for lower resolutions. Also, the spurious currents are reduced in the case of GFCMI when the mesh is refined, which is not the case of GFSCMI, hence suggesting a lack of stability of the latter.

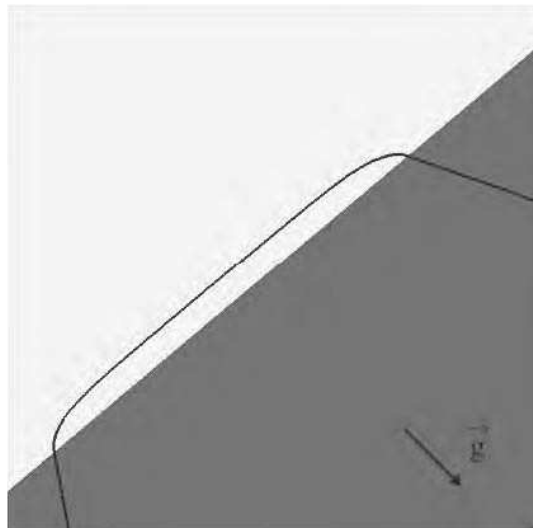
4.5. Three Dimensional Two-Phase Flows in Irregular Domains Without Volume Forces

This study is the same configuration that the one used in section 4.3 except that they represent three dimensional numerical simulations. The slanted wall is chosen in a way as to enforce the wall



(a) $Bo = 0.1$

(b) $Bo = 1$



(c) $Bo = 10$

Figure 18: Illustration of the shape of a drop on a slanted wall for a static contact angle of $\theta_s = 60^\circ$ and for different Bond numbers.

not to align to the grid in any direction (we take $z = 0.85(x + y)$ for $\alpha = 40^\circ$). The contact angle used in this study is $\theta_s = 120^\circ$. The analytical radius, R , thickness, e and length, L , of the drop

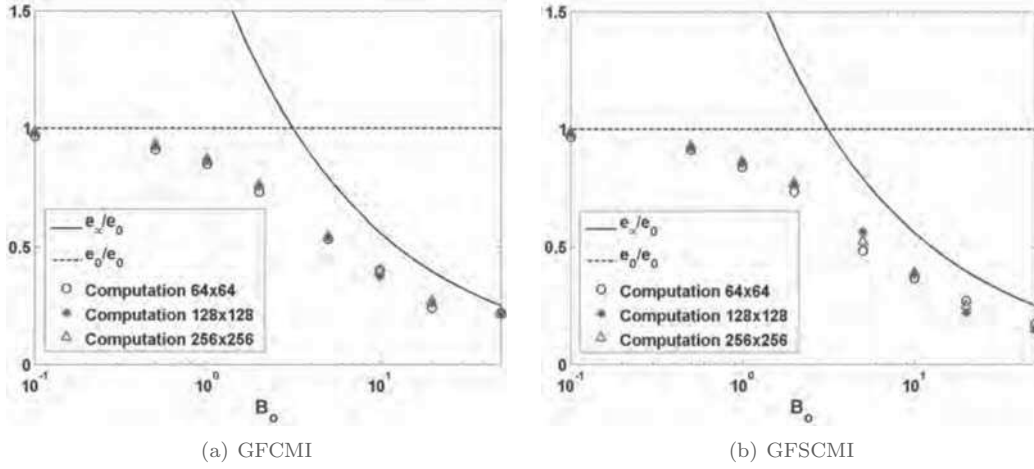


Figure 19: Results on a slanted wall as a function of B_0 for $\theta_s = 60^\circ$.

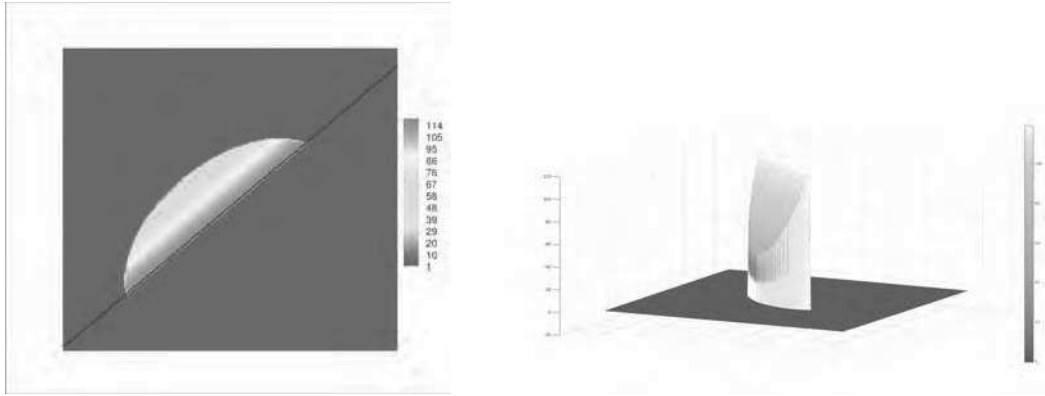


Figure 20: Resulting pressure field (Pa) for $B_0 = 1$ and $\theta_s = 60^\circ$.

are obtained as before, using mass conservation arguments and geometrical properties:

$$R = R_0 \left(\frac{2}{(2 + \cos \theta_s)(1 - \cos \theta_s)^2} \right)^{\frac{1}{3}}, \quad L = 2R \sin \theta_s, \quad e = R(1 - \cos \theta_s). \quad (24)$$

Figure 22 illustrates the final shape of the drop on an slanted wall for $\theta_s = 120^\circ$. In table 7, some comparisons between GFSCMI and GFCMI are presented. Even though the mesh with 64^3 cells is not fine enough to model the behavior of the drop, the error is still less than 9% for the analytical values of the thickness of the drop between the two methods. The error is reduced when the mesh is 128^3 , with an error of 3.1% for GFCMI and 4.7% for GFSCMI.

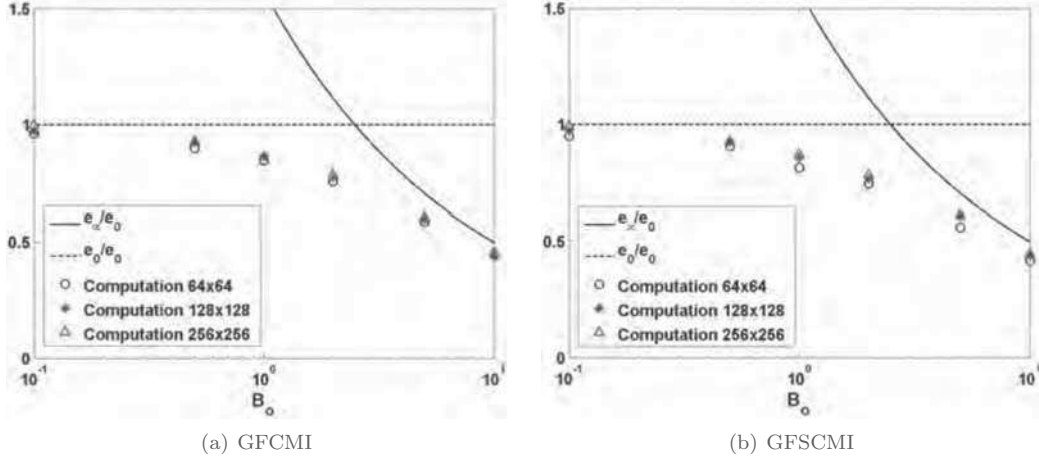


Figure 21: Results on a slanted wall as a function of B_0 for $\theta_s = 120^\circ$.

		GFCMI			GFSCMI		
		e	$\ V_{max}\ $	%Vol	e	$\ V_{max}\ $	%Vol
$E_0 = 0.1$	64^2	0.961	0.0105	0.975	0.950	0.034	0.97
	128^2	0.972	0.006	0.987	0.974	0.0105	0.992
	256^2	0.998	0.0044	0.996	0.994	0.011	0.98
$E_0 = 1$	64^2	0.846	0.02	0.932	0.813	0.062	0.895
	128^2	0.864	0.015	0.964	0.857	0.04	0.97
	256^2	0.846	0.010	0.995	0.870	0.025	0.988
$E_0 = 10$	64^2	0.444	0.18	0.83	0.416	0.2	0.85
	128^2	0.442	0.15	0.905	0.440	0.15	0.905
	256^2	0.456	0.11	0.970	0.442	0.15	0.905

Table 6: Comparison of characteristics of GFCMI and GFSCMI for a slanted wall ($\alpha = 40^\circ$) and for $\theta_s = 120^\circ$

	e/R_0	L/R_0
Analytical results	1.26	1.45
GFCMI		
$\Delta_x = R_0/10$	1.16	1.47
$\Delta_x = R_0/20$	1.22	1.45
GFSCMI		
$\Delta_x = R_0/10$	1.17	1.48
$\Delta_x = R_0/20$	1.20	1.46

Table 7: Geometrical characteristics of the final shape of the drop on a slanted wall ($\alpha = 40^\circ$) in 3D for $\theta_s = 120^\circ$.

4.6. Half-Filled Rotating Spherical Tank

In order to highlight the applicability of the numerical methods presented in this paper, we now describe the benchmark of a rotating spherical tank half-filled with water, with the other half filled

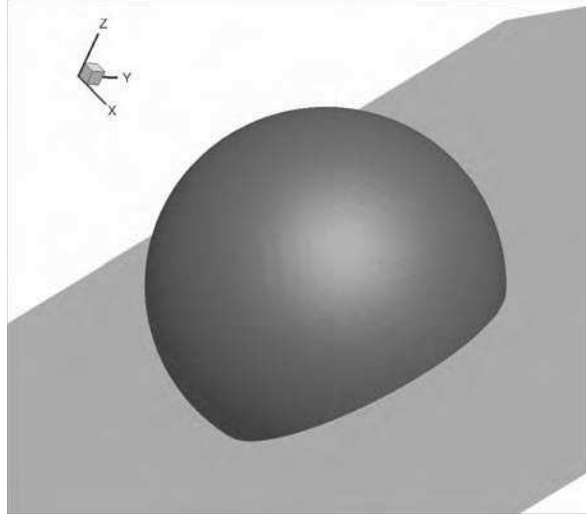


Figure 22: Final shape of the drop on an slanted wall for $\theta_s = 120^\circ$.

with a gas. The center of the spherical tank is located at a distance $x = L$, from a rotational axis along the z -axis. Consequently, the spherical tank is rotating in the xy -plane. As our simulations will be performed in the tank reference frame R' , it is necessary to account for the inertial forces, which can be decomposed into two terms in the Navier-Stokes equations:

$$\mathbf{F}_v = -\rho\mathbf{a}_1 - \rho\mathbf{a}_2,$$

with

$$\mathbf{a}_1 = \boldsymbol{\Omega}_{R'/R} \times (\boldsymbol{\Omega}_{R'/R} \times \mathbf{OM}) \quad \text{and} \quad \mathbf{a}_2 = 2\boldsymbol{\Omega}_{R'/R} \times \mathbf{V}_{R'}(M), \quad (25)$$

where \mathbf{a}_1 represents the centrifugal acceleration and \mathbf{a}_2 denotes the Coriolis acceleration. We also denote by $\boldsymbol{\Omega}_{R'/R} = \omega \mathbf{e}_z$ the rotation vector and $\mathbf{OM} = (L+x)\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z$ the position vector of a fluid particle located at M in the tank's reference frame. Since the gravity force acts along the z -axis, we obtain the following expression for the three components of the volume force in the Navier-Stokes equations:

$$F_x = \rho(-2\omega v + \omega^2(L+x)), \quad F_y = \rho(2\omega u + \omega^2 y), \quad \text{and} \quad F_z = -\rho g_z.$$

In this simulation example, we are interested in computing the forces acting on the wall of the tank due to the fluid motion. In the simple configuration, which is presented in this section, a steady theoretical solution of the interface shape and of the pressure force can be obtained. This enables us to assess the accuracy of our numerical methods in a situation where the fluid motion induced by non-inertial forces is coupled to a three-dimensional simulation in an irregular geometry. We have imposed a contact angle between the liquid-gas interface and the solid tank boundary equal to $\theta_s = 90^\circ$.

In the general case, the liquid-gas interface will evolve towards a parabolic-like shape. However, it can be demonstrated that in our configuration, if $R/L \ll 1$, the liquid-gas interface will be a flat surface forming an angle equal to $\theta_{eq} = \tan^{-1} \left(\frac{\omega^2 L}{g_z} \right)$ with the xy -plane.

The theoretical expression of the resulting force is obtained from the integration of the pressure field on the surface of the tank. Let's start with the equilibrium of the forces :

$$\nabla p = -\rho \mathbf{a}_e + \rho \mathbf{g} \quad \text{with} \quad \mathbf{a}_e = -\omega^2 ((L+x) \mathbf{e}_x + y \mathbf{e}_y).$$

Therefore the decomposition of the pressure gradient in each direction is :

$$\frac{\partial p}{\partial x} = \rho \omega^2 (L+x), \quad \frac{\partial p}{\partial y} = \rho \omega^2 y \quad \text{and} \quad \frac{\partial p}{\partial z} = -\rho g_z.$$

Thus with an integration of the pressure field one obtains :

$$p(x, y, z) = \rho \omega^2 \left(Lx + \frac{x^2 + y^2}{2} \right) - \rho g_z z + p_0.$$

In order to compute the force, one must integrate this pressure on the wall of the spherical tank. The integration per direction gives :

$$P_x = \frac{4\pi}{3} \rho \omega^2 L R^3, \quad P_y = 0, \quad P_z = -\frac{4\pi}{3} \rho g_z R^3.$$

The resulting force is expressed here with ρ_l and divided by 2, as the tank is half filled with water :

$$\mathbf{F}_{\mathbf{v}_s \rightarrow \mathbf{f}} = \frac{2}{3} \pi R^3 \rho_l (\omega^2 L \mathbf{e}_x - g_z \mathbf{e}_z).$$

The fluid properties are close to that of water for the liquid and that of air for the gas, i.e. we take $\rho_L = 1000 \text{ kg} \cdot \text{m}^{-3}$, $\mu_L = 10^{-3} \text{ kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$, $\rho_G = 1 \text{ kg} \cdot \text{m}^{-3}$, $\mu_G = 10^{-5} \text{ kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$ and $\sigma = 7 \times 10^{-2} \text{ N} \cdot \text{m}^{-1}$. The radius of the spherical tank is $R = 2.5 \times 10^{-2} \text{ m}$ and the acceleration due to gravity is equal to $g_z = 9.81 \text{ m} \cdot \text{s}^{-2}$. We have performed simulations for the four configurations given in table 8.

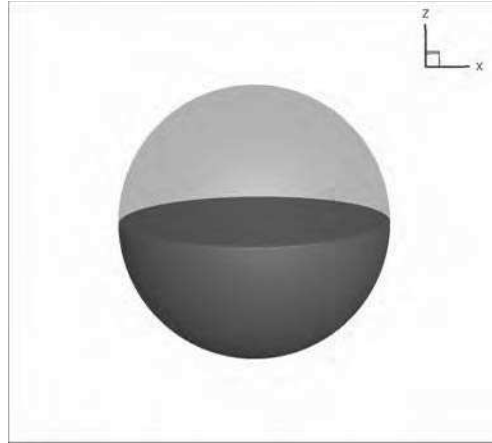


Figure 23: Initial position of the interface for the four simulations of the rotating spherical tank.

	Case A	Case B	Case C	Case D
$\frac{\omega^2 L}{g_z}$	0.176	1	5.67	1
$\frac{R}{L}$	0.05	0.05	0.05	1

Table 8: Configuration of the four cases

For large time, a flat interface is expected in cases *A*, *B* and *C* with $\theta_{eq} = 10^\circ$, 45° and 80° . For case *D*, considering that the ratio R/L is not negligible anymore, a parabolic-like shape of the interface is observed. This can be demonstrated by the former equation depicting the pressure field as :

$$p(x, y, z) = \rho\omega^2 \left(Lx + \frac{x^2 + y^2}{2} \right) - \rho g_z z + p_0.$$

Indeed the interface is an isobar $p(x, y, z) = p_0$, so the shape of the resulting interface in the following plan equation would be :

$$z = f(x, y) = \frac{\omega^2}{g_z} \left(Lx + \frac{x^2 + y^2}{2} \right).$$

If $L \gg R$ then the linear term Lx of the equation will dominate the flow, giving a plan interface, but when $L \simeq R$ then the quadratic term is not negligible anymore and gives the shape of a parabola to the interface.

	Case A		Case B		Case C	
	$F_x(N)$	$F_z(N)$	$F_x(N)$	$F_z(N)$	$F_x(N)$	$F_z(N)$
64^3	0.05574	-0.3166	0.3189	-0.3518	1.746	-0.302
128^3	0.05618	-0.3234	0.3358	-0.3330	1.974	-0.337
Theory	0.0566	-0.321	0.321	-0.321	1.82	-0.32

Table 9: Comparisons between theoretical and the computational results (for two different grids) of the pressure force acting on the wall for cases *A*, *B*, and *C*.

The initial interface (in blue), depicted in figure 24, is simply a horizontal plane in all four cases. The boundary of the spherical tank and the axis orientation of the computational field is also depicted in figure 24. All the simulations are carried out until the final time $t = 5$ s. In figure 24, the final steady shape of the interface obtained with a 128^3 grid is plotted for the four cases. As expected, a flat sloped interface is observed in the simulations of cases *A*, *B* and *C*. In case *D*, we clearly observe a parabolic-like shaped interface, in agreement with theory. Quantitative comparisons between theoretical and computational results are presented in table 9. In figure 25, the interface location and the pressure field are plotted in the xz -plane for case *B* and *D*, where we used a grid with size 128^3 . This gives a more accurate visualization of the interface shape and of the pressure isocontours.

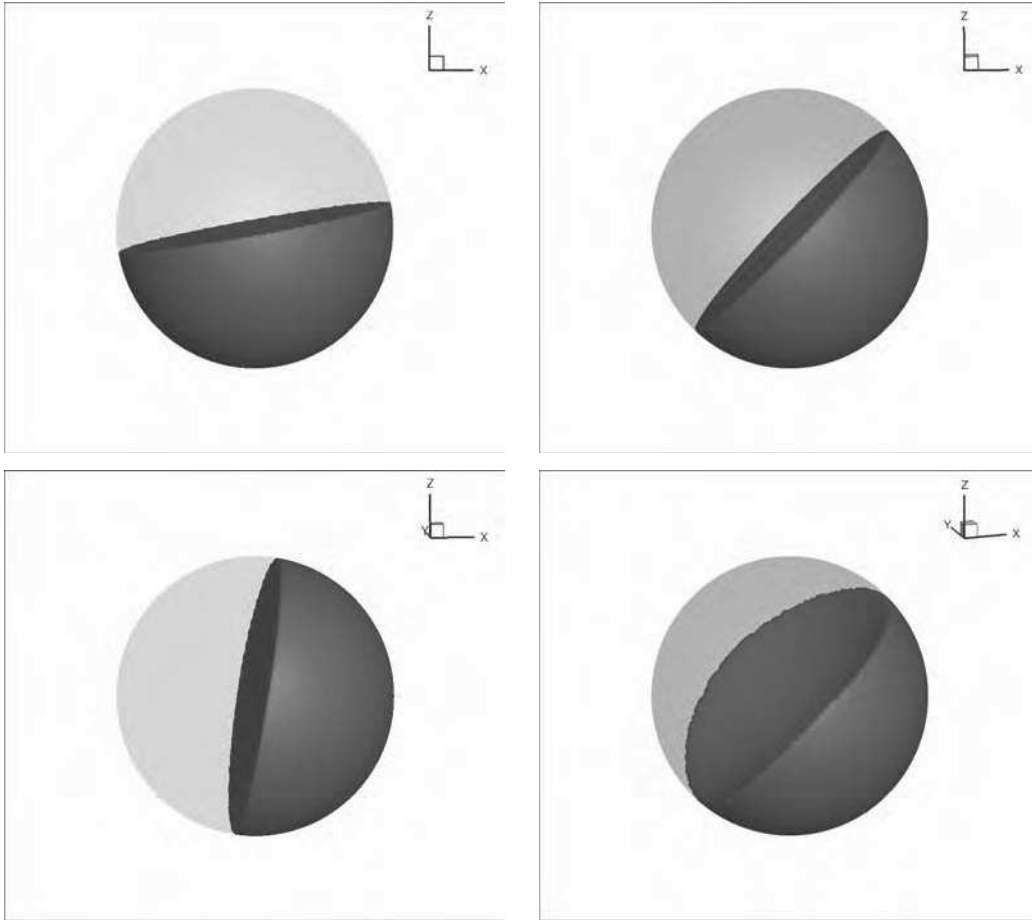


Figure 24: Final steady position of the interface for the four simulations (cases A, B, C and D) of the rotating spherical tank (grid 128^3).

5. Conclusion

In this paper, several numerical methods are proposed to compute numerical simulations of liquid-gas flows with an embedded solid boundary inside a Cartesian grid. Our numerical method is presented as an extension of the previous work in [32], on a coupling between a one-phase flow and a solid, to two-phase flows simulations where contact lines can be formed at the boundary between the three phases. Several challenges must be faced to compute accurately the surface tension effects, the viscous effects, the contact angle and the volume forces in this more complex framework. We propose a detailed analysis and suitable solutions to overcome these difficulties. Next, accurate benchmarks are defined and numerical simulations are performed in order to assess the suitable behavior of our numerical method in various situations. As an implicit temporal discretization is required to compute the viscous terms in the grid cells which are crossed by the solid-fluid

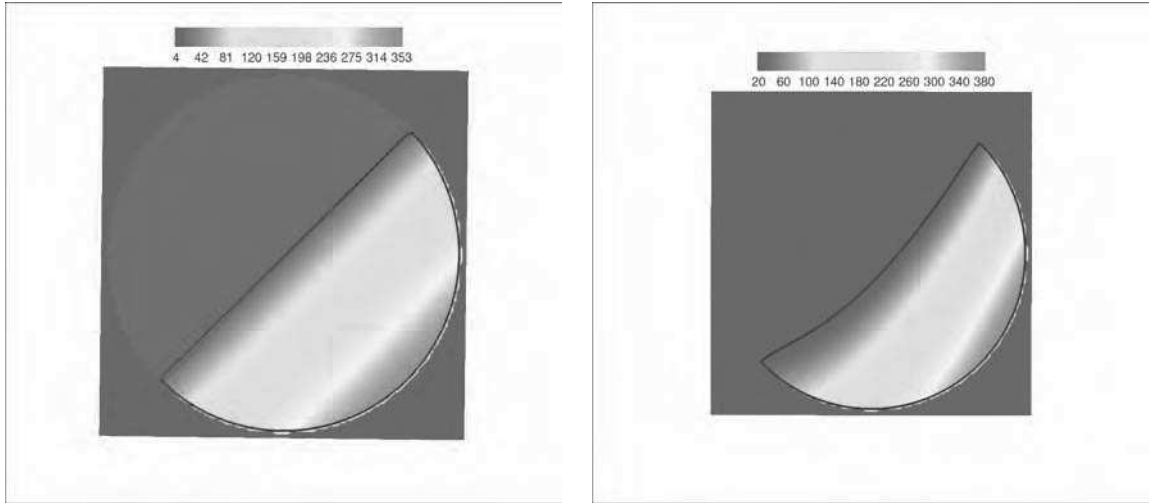


Figure 25: Final steady position of the interface and pressure field (Pa) for the four simulations (cases *A*, *B*, *C* and *D*) of the rotating spherical tank for cases *B* and *D* (grid 128^3).

interface, we compare two possible implementations of this implicit temporal discretization. This comparison brings out that the Conservative formulation (GFCMI) of the viscous terms should be favored instead of the Semi-Conservative formulation (GFSCMI), since an important decrease of the spurious currents on the contact line is observed with the conservative one.

Acknowledgements

Mathieu Lepilliez and Sébastien Tanguy gratefully acknowledge funding by Airbus Defence & Space and the CNES (the French national space agency) in the context of the “sloshing in satellite tank under micro-gravity influence” program. Frederic Gibou gratefully acknowledges funding from ONR N00014-11-1-0027.

Appendix A. Rising spherical bubbles at low Reynolds numbers

In this appendix, a study is carried out to compare the different projection methods on regular domains. This study was initially proposed in [23] and is completed in this paper with GFCMI.

A spherical air bubble is placed in water under gravity stress. This study is adequate to compare how the different projection methods affect the viscous terms as no drag is generated by the pressure in this flow. These are 2D axisymmetric numerical simulations with a variation of the Reynolds number, and two different Bond numbers, in order to keep the spherical shape. The characteristics of the numerical simulations are reported in table A.10.

	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
$R_{\text{bubble}}[\mu\text{m}]$	19.55	44.65	70.45	98.95	127.9
$g_z[\text{m} \cdot \text{s}^{-2}]$	-9144.4	-1755.2	-705.6	-357.5	-213.9
$\sigma[\text{N} \cdot \text{m}^{-1}]$	0.56	0.56	0.56	1.12	1.12
$t_f[\text{s}]$	0.0004	0.00135	0.002375	0.0048	0.007

Table A.10: Parameters used in the numerical experiment of this appendix.

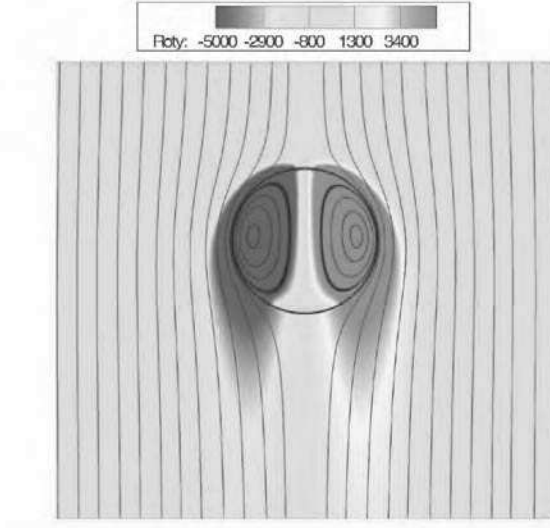


Figure A.26: Streamlines and vorticity contours (s^{-1}) in the moving referential of the rising bubble for $Re = 60$.

The numerical domain is chosen in order not to disturb the rise of the bubble, with $L_r = 8R_{\text{bubble}}$ and $L_z = 4L_r$. The physical properties of the two fluids are those of air and water, i.e., $\rho_L = 1000 \text{ kg} \cdot \text{m}^{-3}$, $\mu_{liq} = 0.00113 \text{ kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$, $\rho_{gas} = 1.226 \text{ kg} \cdot \text{m}^{-3}$, $\mu_{gas} = 1.78 \text{ kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$. Table A.10 gives the other physical parameters for five different configurations corresponding to $Re = 20, 40$ and 60 for the Bond number $Bo = 0.025$ and to $Re = 80, 100$ for $Bo = 0.0125$. Figure A.26 illustrates the results for the simulation of a rising air bubble where the final Reynolds number is $Re = 60$. The numerical simulations are conducted for three different meshes, 32×128 , 64×256 and 128×512 corresponding respectively to 8, 16 and 32 cells per diameter.

Mesh	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	- 22.4%	- 32.2%	- 37.2%	- 75.53%	- 43.35%
64×256	+ 2.44%	+ 3.12%	+ 1.07%	- 1.78%	- 1.30%
128×512	+ 2.38%	+ 2.38 %	+ 0.37 %	+ 0.8 %	+ 0.19%

Table A.11: Error on the final Reynolds number value for rising air bubbles with GFCMI.

Mesh	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	- 22.0%	- 34.0%	- 38.8%	- 63.0%	- 67.4%
64×256	+ 2.5%	+ 1.8%	- 1.3%	- 1.3%	- 3.0%
128×512	+ 0.5%	+ 0.8 %	- 1.0 %	+ 1.9 %	+ 1.9%

Table A.12: Error on the final Reynolds number value for rising air bubbles with the GFCM.

Mesh	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	- 18.6%	- 25.0%	- 30.2%	- 52.9%	- 55.1%
64×256	- 1.2%	+ 1.5%	- 0.1%	+ 2.0%	+ 0.9%
128×512	- 3.7%	+ 0.1 %	- 0.2 %	+ 3.2 %	+ 3.9%

Table A.13: Error on the final Reynolds number value for rising air bubbles with the GFSCMI.

Mesh	$Re_\infty = 20$	$Re_\infty = 40$	$Re_\infty = 60$	$Re_\infty = 80$	$Re_\infty = 100$
32×128	- 23.5%	- 32.5%	- 9.0%	- 62.5%	- 67.0%
64×256	- 0.5%	+ 2.5%	+ 1.3%	+ 4.3%	+ 3.0%
128×512	- 3.5%	+ 0.6 %	+ 0.7 %	+ 5.4 %	+ 6%

Table A.14: Error on the final Reynolds number value for rising air bubbles with the GFSCM with an explicit scheme used to compute the viscous terms.

Results obtained with GFCMI are presented in table A.11. For completeness, we also include the results presented in [23], namely the results obtained for GFCM (table A.12), in which the viscous terms were computed using an explicit scheme, the results obtained for GFSCMI (table A.13), and those obtained for GFSCM (table A.14), where the viscous terms were computed with an explicit scheme.

First of all, one can observe that the mesh 32×128 cannot describe the viscous effects accurately, as the viscous boundary layer is not thick enough to be fully resolved. For the other meshes, the results are satisfactory with an error under 3% in most cases.

The results obtained with GFCM and GFCMI indicate convergence with less than 3% on every cases. GFSCM and GFSCMI do not give results that are as accurate. This study thus indicates that GFCM and GFCMI are better suited for modeling viscous effects in multiphase flows, and confirms that GFCMI is a reliable method to compute the viscous terms in the case of irregular domains.

References

- [1] S. Afkhami, S. Zaleski, M. Bussmann, A mesh-dependent model for applying dynamic contact angles to VOF simulations, *J. Comput. Phys.* 228 (2009) 5370-5389.
- [2] A.S. Almgren, J.B. Bell, P. Colella, L. H. Howell, M. L. Welcome, A Conservative Adaptive Projection Method for the Variable Density Incompressible NavierStokes Equations , *J. Comput. Phys.* 1 (1998) 1-46.
- [3] T. Aslam, A partial differential equation approach to multidimensional extrapolation, *J. Comput. Phys.* 193 (2004) 349-355.
- [4] P. A. Berthelsen and O. M. Faltinsen. A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries, *J. Comput. Phys* 227 (2008) 4354-4397.
- [5] R. Borges, M. Carmona, B. Costa, W.S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws, *J. Comput. Phys.* 227 (2008) 3191-3211.
- [6] M. Braza, P. Chassaing, H. Ha Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, *Journal of Fluids Mechanics* 165 (1986) 79-130.
- [7] B. Bunner, G. Tryggvason ,Dynamics of homogeneous bubbly flows Part 1. Rise velocity and microstructure of the bubbles, *J. Fluid Mech.*, 446 (2002) 17-52.
- [8] D. Calhoun, A cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions, *J. Comput. Phys.* 176 (2002) 231-275.
- [9] J. Choi, R.C. Oberoi, J. R. Edwards, and J. A. Rosati, An immersed boundary method for complex incompressible flows, *J. Comput. Phys.* 224 (2007) 757-784.
- [10] A. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* 2 (1967) 12-26.
- [11] R. Coutanceau, M. et Bouard. Experimental determination of main features of the viscous flow in the wake of a circular cylinder in uniform translation. part 1 steady flow. *Journal of Fluids Mechanics*, 79 (1977) 4018-4037.
- [12] J.E. Dendy, Black box multigrid, *J. Comput. Phys.* 48 (3) (1982) 366386.
- [13] J-B. Dupont and D. Legendre. Numerical simulation of static and sliding drop with contact angle hysteresis, *J. Comput. Phys.* 229(2010) 2453-2478.
- [14] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (The Ghost-Fluid Method), *J. Comput. Phys.* 152 (1999) 457-492.
- [15] P.G de Gennes, Wetting: statics and dynamics, *Rev. Mod. Phys.* 57 (1985) 827.
- [16] F. Gibou, R. Fedkiw, L.-T. Cheng, and M. Kang. A second-order-accurate symmetric discretization of the poisson equation on irregular domains, *J. Comput. Phys.* 176 (2002) 205-227.

- [17] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level-set based sharp interface method for the multiphase incompressible Navier-Stokes equations with phase change, *J. Comput. Phys.* 222 (2007) 536-555.
- [18] A. Guittet, M. Theillard, F. Gibou, A stable projection method for the incompressible Navier-Stokes equations on arbitrary geometries and adaptive Quad/Octrees, *J. Comput. Phys.* 292 (2015) 215-238.
- [19] A. Guittet, M. Lepilliez, S. Tanguy, F. Gibou, Solving elliptic problems with discontinuities on irregular domains - the Voronoi Interface Method, *J. Comput. Phys.* 298 (2015) 747-765.
- [20] T.A. Johnson and V.C. Patel, Flow past a sphere up to a Reynolds number of 300, *Journal of Fluids Mechanics*, 378 (1999) 19-70.
- [21] M. Kang, R. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, *J. Sci. Comput.* 15 (2000) 323-360.
- [22] S. Kang, G. Iaccarino, F. Ham, P. Moin, Prediction of wall-pressure fluctuation in turbulent flows with an immersed boundary method, *J. Comput. Phys.* 228 (2009) 6753-6772.
- [23] B. Lalanne, L. Rueda Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost-Fluid Method, *J. Comput. Phys.* 301 (2015) 289-307.
- [24] R.J. Leveque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, *SIAM J. Sci. Comput.* 18 (1997) 709-735.
- [25] X.-D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, *J. Comput. Phys.* 160 (2000) 151-178.
- [26] H. Liu, S. Krishnan, S. Marella, H.S. Udaykumar, Sharp interface Cartesian grid method II: A technique for simulating droplet interactions with surfaces of arbitrary shape, *J. Comput. Phys.* 210 (2005) 32-54.
- [27] J. Magnaudet, M. Rivero, J. Fabre, Accelerated flows past a rigid sphere or a spherical bubble. Part 1. Steady straining flow. *J. Fluid Mech.* 284 (1995) 97-135.
- [28] S. Marella, S. Krishnan, H. Liu, H.S. Udaykumar, Sharp interface Cartesian grid method I: An easily implemented technique for 3D moving boundary computations, *J. Comput. Phys.* 210 (2005) 1-31.
- [29] T. Menard, S. Tanguy, A. Berlemont, Coupling level set/VOF/Ghost-Fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet, *Int. J. of Multiphase Flow* 33 (2007) 510-524.
- [30] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid. Mech.* (2005) 239-261.
- [31] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, and A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.* 227 (2008) 4825-4852.

- [32] Y.T. Ng, C. Min, F. Gibou, An efficient fluid-solid coupling algorithm for single-phase flows, *J. Comput. Phys.* 228 (2009) 8807-8829.
- [33] E. Noel, A. Berlemont, J. Cousin, T. Menard, Application of the immersed boundary method to simulate flows inside and outside the nozzles, ICLASS 2012, 12th Triennial International Conference on Liquid Atomization and Spray Systems, Heidelberg, Germany, September 2-6, 2012
- [34] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.* 79 (1988) 12-49.
- [35] C.S. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.* 10 (1977) 252-271.
- [36] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, *Int. J. Numer. Methods Fluids* 30 (1999) 775-793.
- [37] N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Geiger, S. Hoon and R. Fedkiw, Directible Photorealistic Liquids, ACM SIGGRAPH/ Eurographics Symposium on Computer Animation (SCA), edited by R. Boulic and D. Pai (2004) pp 193-202.
- [38] F.W. Roos and W.W. Willmarth, Some experimental results on sphere and disk drag, *AIAA J.*, 9 (1971) 285-291.
- [39] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* 31 (1999) 567-603.
- [40] S. Sikalo, H.-D. Wilhelm, I.V. Roisman, S. Jakirlic, C. Tropea, Dynamic contact angle of spreading droplets: Experiments and simulations, *Physics of Fluids*, 17 (2005) 62103-62116.
- [41] J.H. Snoeijer, B. Andreotti, Moving Contact Lines: Scales, Regimes, and Dynamical Transitions, *Annual Review of Fluid Mechanics* 45 (2013) 269-292.
- [42] P. D.M. Spelt, A level-set approach for simulations of flows with multiple moving contact lines with hysteresis, *J. Comput. Phys.* 207 (2005) 389-404.
- [43] Y Sui, H Ding, P.D.M. Spelt, Numerical simulations of flows with moving contact lines, *Annual Review of Fluid Mechanics* 46 (2014) 97-119.
- [44] M. Sussman, P. Smereka, S. Osher, A Level Set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* 114 (1994) 146-159.
- [45] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, *J. Comput. Phys.* 221 (2007) 469-505.
- [46] S. Taneda, Experimental investigation of the wake behind a sphere at low Reynolds numbers, *J. Phys. Soc. Japan* 11 (1956) 1104-1108.
- [47] S. Tanguy, T. Menard, A. Berlemont, A Level-Set Method for vaporizing two-phase flows, *J. Comput. Phys.* 221 (2007) 837-853.

- [48] S. Tanguy, M. Sagan, B. Lalanne, F. Couderc, C. Colin, Benchmarks and numerical methods for the simulation of boiling flows, *J. Comput. Phys.* 264 (2014) 1-22.
- [49] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A Front-Tracking Method for the Computation of Multiphase Flow, *J. Comput. Phys.* 169 (2001) 708-759.
- [50] D.J. Tritton, *Physical Fluid Dynamics*, Oxford Science Publications, 1988.
- [51] Y. Tseng, J. H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* 192 (2003) 593-623.
- [52] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2005) 448-476.
- [53] S. Vincent, A. Sarthou, J.P. Caltagirone, F. Sonilhac, P. Fevrier, C. Mignot, G. Pianet, Augmented Lagrangian and penalty methods for the simulation of two-phase flows interacting with moving solids. Application to hydroplaning flows interacting with real tire tread patterns, *J. Comput. Phys.* 230 (2011) 956-983.
- [54] S. Vincent, J.C. Brandle de Motta, A. Sarthou, J.L. Estivalezes, O. Simonin, E. Climent, A lagrangian VOF tensorial penalty method for the DNS of resolved particle-laden flows, *J. Comput. Phys.* 256 (2014) 582-614.
- [55] W. Xie and P.E. Desjardin, A level set embedded interface method for conjugate heat transfer simulations of low speed 2D flow, *Computers & Fluids*, 37 (2008) 1262-1275.
- [56] S. Xu, The immersed interface method for simulating prescribed motion of rigid objects in an incompressible viscous flow, *J. Comput. Phys.* 227 (2008) 5045-5071.
- [57] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, *J. Comput. Phys.* 215 (2006) 12-40.
- [58] T. Ye, R. Mittal, H.S. Udaykumar, and W. Shyy, An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* 156 (1999) 209-240.
- [59] K. Yokoi, F. Xiao, H. Liu, K. Fukasaku, Three-dimensional numerical simulation of flows with complex geometries in a regular Cartesian grid and its application to blood flow in cerebral artery with multiple aneurysms, *J. Comput. Phys.* 202 (2005) 1-19.
- [60] L. Zhilin, An overview of the immersed interface method and its applications, *Taiwanese Journal of Mathematics*, 7 (2003) 1-49.

Chapitre 4

Étude de ballotements dans les réservoirs de satellites

Sommaire

4.1	Équations de Navier-Stokes	130
4.1.1	Force d'entraînement et Force d'inertie	130
4.1.2	Équations de Navier-Stokes adimensionnalisées	130
4.2	Première approche du ballotement - cas simple	132
4.2.1	Rôle du nombre de Bond impulsional	132
4.2.2	Rôle du nombre de Bond centrifuge	136
4.2.3	Convergence des simulations	147
4.3	Étude de la manœuvre GAP/SUP	148
4.3.1	Description de la manœuvre GAP/SUP	149
4.3.2	Visualisation et Trajectoire	150
4.3.3	Centre de masse	152
4.3.4	Forces et moments	153
4.3.5	Convergence des simulations	153

Dans ce chapitre nous présentons des simulations numériques de ballotements dans les réservoirs de satellites réalisées avec le code DIVA et les méthodes numériques qui ont été décrites au préalable dans ce manuscrit. Dans un premier temps le système physique à résoudre, à savoir l'équation de Navier-Stokes est adimensionnalisée, et fait apparaître les effets importants lors des phases de manœuvre dans les réservoirs. Différents instants tel que le forçage, son intensité et le temps durant lequel le système y est soumis, et le régime de ballotement à proprement parler sont étudiés. Les deux aspects sont abordés avec des cas simplifiés pour comprendre une partie des phénomènes intervenants, puis une étude plus complexe de manœuvre réelle, appelée la manœuvre GAP/SUP est proposée.

4.1 Équations de Navier-Stokes adaptées à l'étude des ballotements en micro-gravité

Pour comprendre quels sont les phénomènes physiques qui dominant l'écoulement, il est nécessaire de partir des équations de Navier-Stokes et de les adimensionnaliser.

$$\nabla \cdot \mathbf{U} = 0 \quad (4.1.1)$$

$$\rho \left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla) \mathbf{U} \right) = -\nabla p + \rho \mathbf{g} + \nabla \cdot (2\mu \mathbf{ID}) \quad (4.1.2)$$

avec \mathbf{U} le vecteur vitesse, p la pression, μ et ρ respectivement la viscosité dynamique et la masse volumique, \mathbf{g} l'accélération de la gravité et \mathbf{ID} le tenseur des taux de déformation tel que $\mathbf{ID} = \frac{1}{2}(\nabla \mathbf{U} + \nabla \mathbf{U}^T)$ représente le vecteur normal à l'interface, σ la tension de surface et κ la courbure. Dans ce contexte, la gravité est très faible de part la distance entre le satellite et la Terre et sera donc négligée par la suite.

L'objectif de cette modélisation est de comprendre l'impact des phénomènes de ballotement sur le satellite, ainsi celui-ci sera considéré comme le référentiel initial du problème. Toutefois dans le cadre de notre modélisation il est plus aisé de résoudre les équations de Navier-Stokes dans le référentiel non galiléen du réservoir.

4.1.1 Force d'entraînement et Force d'inertie

Comme l'illustre la figure 4.1, le réservoir est excentré par rapport au centre de masse du satellite. Ce changement de repère va donc engendrer des accélérations de Coriolis et d'entraînement. L'accélération de Coriolis s'exprime comme :

$$\mathbf{a}_{\text{Cor}} = \mathbf{a}_1 = 2\boldsymbol{\Omega} \times \mathbf{u} \quad (4.1.3)$$

avec $\boldsymbol{\Omega}$ la vitesse de rotation et \mathbf{u} le champs de vitesse local, et l'accélération d'entraînement :

$$\mathbf{a}_{\text{Ent}} = \mathbf{a}_2 + \mathbf{a}_3 \quad \text{avec} \quad \mathbf{a}_2 = \dot{\boldsymbol{\Omega}} \times \mathbf{OM} + \mathbf{a}_R \quad \text{avec} \quad \mathbf{a}_3 = \boldsymbol{\Omega} \times \boldsymbol{\Omega} \times \mathbf{OM} \quad (4.1.4)$$

avec $\dot{\boldsymbol{\Omega}}$ l'accélération de rotation, \mathbf{OM} le vecteur position de la particule défini par $\mathbf{OM} = x\mathbf{e}_x + (L + y)\mathbf{e}_y + z\mathbf{e}_z$ avec L est le bras de levier (distance entre le centre du réservoir et l'axe de rotation) et \mathbf{a}_R l'accélération entre le centre de masse du satellite et du réservoir, qui sera nulle dans le cadre de cette étude puisque le réservoir est fixé au satellite.

4.1.2 Équations de Navier-Stokes adimensionnalisées

Maintenant que les efforts auxquels sont soumis le réservoir par rapport au changement de repère sont exprimés, nous pouvons les intégrer aux équations de Navier-Stokes.

$$\nabla \cdot \mathbf{U} = 0 \quad (4.1.5)$$

$$\rho \left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla) \mathbf{U} \right) = -\nabla p + \nabla \cdot (2\mu \mathbf{ID}) + \rho(\mathbf{a}_{\text{Cor}} + \mathbf{a}_{\text{ent}}) \quad (4.1.6)$$

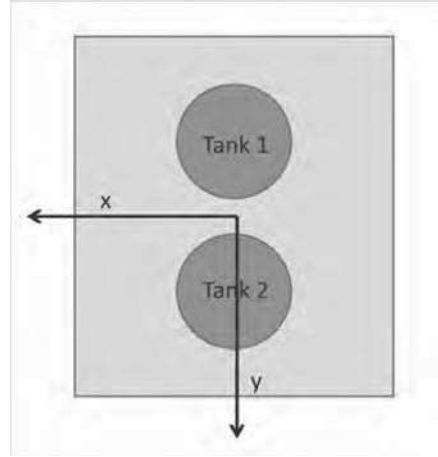


FIGURE 4.1 – Schéma de la position des réservoirs d'ergols dans un satellite.

Afin d'adimensionnaliser la masse volumique et la viscosité, rappelons que dans le cadre de la modélisation d'écoulements diphasiques, ces variables dépendent du fluide dans lequel elles sont exprimées. Aussi en utilisant la fonction Level Set proposée dans les chapitres précédents, il est possible de définir dans tout le domaine : $\rho(\phi) = \rho_l \left(\frac{\rho_g}{\rho_l} + \left(1 - \frac{\rho_g}{\rho_l}\right) H(\phi) \right)$ et $\mu(\phi) = \mu_l \left(\frac{\mu_g}{\mu_l} + \left(1 - \frac{\mu_g}{\mu_l}\right) H(\phi) \right)$. Les variables adimensionnées que nous choisissons sont les suivantes :

$$\rho^* = \frac{\rho}{\rho_l} \quad \mu^* = \frac{\mu}{\mu_l} \quad p^* = \frac{pD_0}{\sigma} \quad \mathbf{U}^* = \mathbf{U} \sqrt{\frac{\rho_l D_0}{\sigma}} \quad t^* = t \sqrt{\frac{\sigma}{\rho_l D_0^3}} \quad (4.1.7)$$

$$\nabla^* = D_0 \nabla \quad \mathbf{a}_1^* = \frac{\mathbf{a}_1}{\gamma} \quad \mathbf{a}_2^* = \frac{\mathbf{a}_2}{\gamma_a} \quad \mathbf{a}_3^* = \frac{\mathbf{a}_3}{\gamma} \quad (4.1.8)$$

avec D_0 le diamètre initial de la bulle, et γ et γ_a des accélérations caractéristiques de l'écoulement. Les équations de Navier-Stokes deviennent donc :

$$\nabla^* \cdot \mathbf{U}^* = 0 \quad (4.1.9)$$

$$\rho^* \left(\frac{\partial \mathbf{U}^*}{\partial t^*} + (\mathbf{U}^* \cdot \nabla^*) \mathbf{U}^* \right) = -\nabla^* p^* + \text{Oh} \nabla^* \cdot (2\mu^* \mathbf{ID}^*) + \text{Bo}_{imp} \rho^* (\mathbf{a}_2^*) + \text{Bo}_{cent} \rho^* (\mathbf{a}_1^* + \mathbf{a}_3^*) \quad (4.1.10)$$

avec Oh le nombre d'Ohnesorge et deux nombres de Bo différents définis comme :

$$\text{Oh} = \left(\frac{\mu_l^2}{\rho_l \sigma D_0} \right)^{\frac{1}{2}} \quad \text{Bo}_{cent} = \frac{\rho_l \gamma D_0^2}{\sigma} \quad \text{Bo}_{imp} = \frac{\rho_l \gamma_a D_0^2}{\sigma} \quad (4.1.11)$$

Comme expliqué dans le Chapitre 1, le nombre de Bond compare les effets inertiels liés aux forces volumiques avec effets de tension de surface, et le nombre d'Ohnesorge compare les effets visqueux aux effets de tension de surface. Revenons maintenant sur le choix des variables γ et γ_a .

Dans le cadre de nos modélisations, il faut mettre en avant les efforts auxquelles le réservoir est soumis. Il est possible de définir une accélération caractéristique γ en fonction de la vitesse de rotation Ω telle que $\gamma = \Omega^2 L$, et une seconde accélération dépendant de l'accélération de rotation $\gamma_a = \dot{\Omega} L$. Ces deux accélérations caractéristiques permettent de prendre en compte différents effets.

Lors des manœuvres, le terme $\dot{\Omega}$ est important car il caractérise l'énergie transmise au système par la mise en mouvement du réservoir. Lorsque la manœuvre est en cours et que le satellite tourne sur lui-même, l'accélération est stoppée afin que le satellite tourne à vitesse de rotation constante. C'est à ce moment là que le terme Ω prend toute son importance. Les définitions des nombres de Bond que nous allons utiliser dans la suite de cette thèse sont donc :

$$\text{Bo}_{cent} = \frac{\rho\Omega^2LD_0^2}{\sigma} \quad \text{Bo}_{imp} = \frac{\rho\dot{\Omega}LD_0^2}{\sigma} \quad (4.1.12)$$

4.2 Première approche du ballotement - cas simple

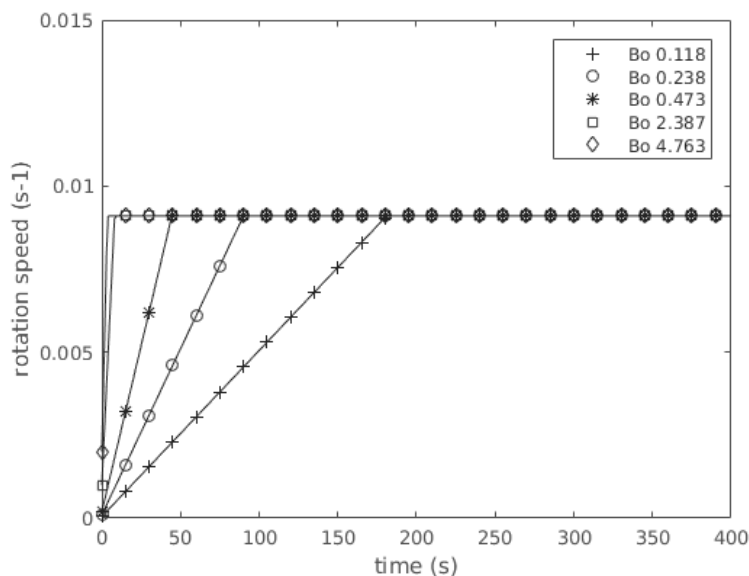
Étant donnée la complexité d'une manœuvre complète, une première approche qui comprend la première phase de la manœuvre est proposée. Les modélisations suivantes portent sur un réservoir sphérique, de rayon $R_{tank} = 0.2925m$, avec un remplissage de 50% en volume. Le domaine modélisé est de dimension $[-0.4-0.4] \times [-0.4-0.4] \times [-0.4-0.4]$. Le liquide utilisé possède les caractéristiques suivantes avec $\rho_l = 1004kg.m^{-3}$, $\mu_l = 9.13 \times 10^{-4}kg.m^{-1}.s^{-1}$, et pour le gaz $\rho_g = 2.41kg.m^{-3}$, $\mu_g = 1.99 \times 10^{-5}kg.m^{-1}.s^{-1}$, avec le coefficient de tension de surface $\sigma = 0.03325N.m^{-1}$. Le centre de la bulle de gaz, de rayon $R_{gaz} = 0.229m$, est initialement confondu avec le centre géométrique du réservoir. Le liquide est parfaitement mouillant donc $\theta_s = 0^\circ$ ce qui implique qu'il n'y a pas de formation de ligne triple. La vitesse et l'accélération de rotation sont définies selon l'axe \mathbf{e}_z telles que $\dot{\Omega} = \dot{\Omega}\mathbf{e}_z$ et $\Omega = \Omega\mathbf{e}_z$. L'étude suivante comporte deux parties. Dans un premier temps le nombre de Bond impulsif varie, pour un nombre de Bond centrifuge constant, ce qui revient à atteindre une vitesse de rotation finale identique pour tous les cas. Ceci permet de caractériser le rôle de l'impulsion donnée au système. Dans un second temps le nombre de Bond impulsif est constant, c'est à dire que l'accélération à laquelle le réservoir est soumis reste identique pour toutes les simulations, mais le temps durant lequel le réservoir est soumis à cette accélération varie. Notons que l'on s'intéresse à un état stationnaire par rapport au référentiel tournant, la solution ne doit a priori pas dépendre du nombre d'Ohnesorge qui caractérise les effets visqueux.

4.2.1 Rôle du nombre de Bond impulsif

Dans cette partie, nous proposons une étude sur le rôle du nombre de Bond impulsif. Le nombre de Bond centrifuge est fixé tel que $\text{Bo}_{cent} = 0.194$. Durant la première phase de la manœuvre, le réservoir est soumis à différentes accélérations. Le tableau 4.1 résume les cinq accélérations choisies, ainsi que les nombres de Bond impulsifs résultants et les temps durant lesquels le système est soumis au forçage. Les simulations présentées dans cette partie sont réalisées avec un maillage 64^3 . La figure 4.7 illustre le profil des manœuvres proposées dans cette étude. Toutes les simulations présentées sont donc effectuées avec le même nombre de Bond centrifuge terminal.

4.2.1.1 Trajectoires

Une première approche pour étudier le rôle du nombre de Bond impulsif sur la manœuvre et le comportement des fluides dans le réservoir est de tracer les différentes trajectoires suivies par

FIGURE 4.2 – Profil d'accélération de rotation $\dot{\Omega}$ en fonction du nombre de Bond (Bo_{imp}).

Bo_{imp}	0.118	0.238	0.473	2.387	4.763
$\dot{\Omega}$ ($rad.s^{-2}$)	0.00005	0.0001	0.0002	0.001	0.002
Δt (s)	182	91	45.5	9.1	4.55

TABLE 4.1 – Récapitulatif des simulations effectuées sur l'étude de Bo_{imp} .

le centre de masse des fluides, afin de déterminer si ceux-ci ont un impact sur les efforts générés sur le réservoir. La figure 4.3 présente les différentes trajectoires en fonction des nombres de Bond proposés dans le tableau 4.1.

Nous pouvons nous apercevoir que ces trajectoires se superposent, alors que le rapport entre les extrémités de la gamme de nombres de Bond choisie est de rapport 40. Une étude plus complète des différentes étapes de ces trajectoires est proposée dans la partie portant sur le nombre de Bond d'entraînement.

4.2.1.2 Centre de masse

L'évolution des centres de masse du réservoir en fonction du nombre de Bond est tracée sur les figures 4.4(a) et 4.4(b). Les différences observées sont un déphasage temporel dû aux différents temps auxquels le réservoir est soumis aux accélérations.

4.2.1.3 Forces et Moments

L'évolution du centre de masse que nous observons semble indiquer que la seule différence entre ces différentes configurations est le déphasage généré par la façon dont le système est forcé. Il est possible de valider cette assertion en nous appuyant sur les courbes de l'évolution des forces et des moments.

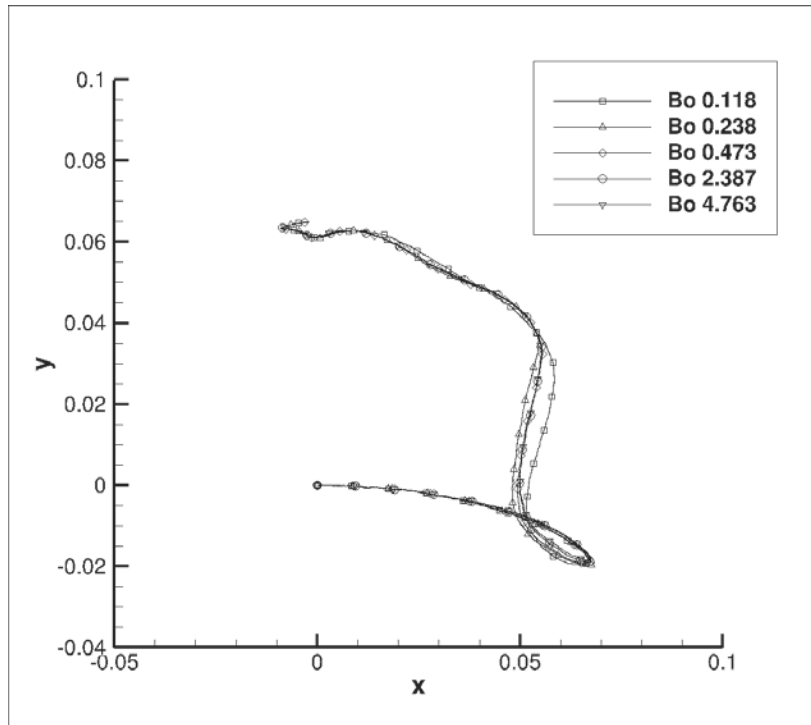


FIGURE 4.3 – Trajectoires de la masse volumique du réservoir en fonction du nombre de Bond (Bo_{imp}).

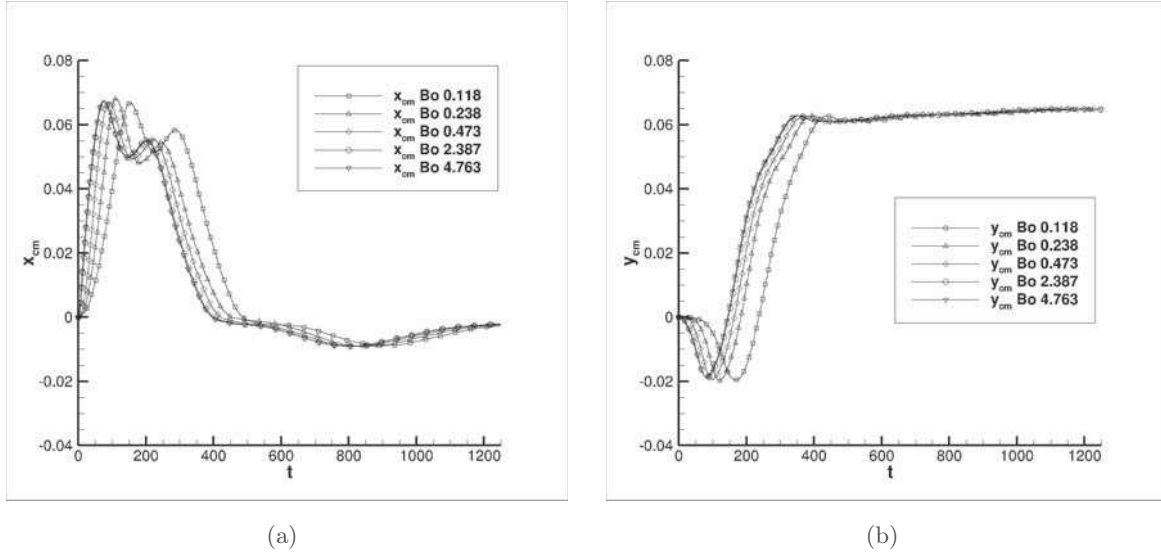


FIGURE 4.4 – Position du centre de masse (m) - (a) dans la direction \mathbf{e}_x et (b) dans la direction \mathbf{e}_y au cours de la manœuvre.

Les figures 4.5(a) et 4.5(b) présentent les forces générées par les fluides sur le réservoir, respectivement dans les directions \mathbf{e}_x et \mathbf{e}_y . Nous pouvons constater que la force dans la direction \mathbf{e}_x tend bien vers 0 lorsque l'équilibre est atteint, et que la force dans la direction \mathbf{e}_y tend vers la même valeur à l'équilibre pour tous les nombres de Bond testés. La valeur de cette force ($F_y \simeq 2 \times 10^{-3} N$) est proche de la valeur théorique, qui s'exprime de la façon suivante $F_{theo} = \rho_l V \Omega^2 L \simeq 1.7 \times 10^{-3} N$,

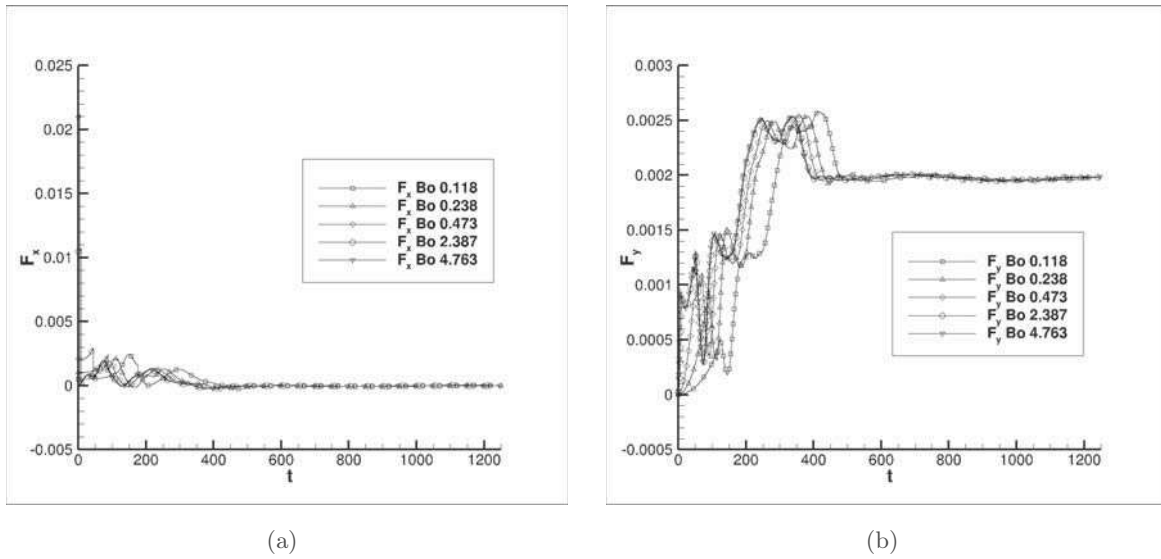


FIGURE 4.5 – Évolution des forces générées par les fluides dans le réservoir en fonction des différents nombres de Bond Bo_{imp} .

avec V le volume de liquide présent dans le réservoir.

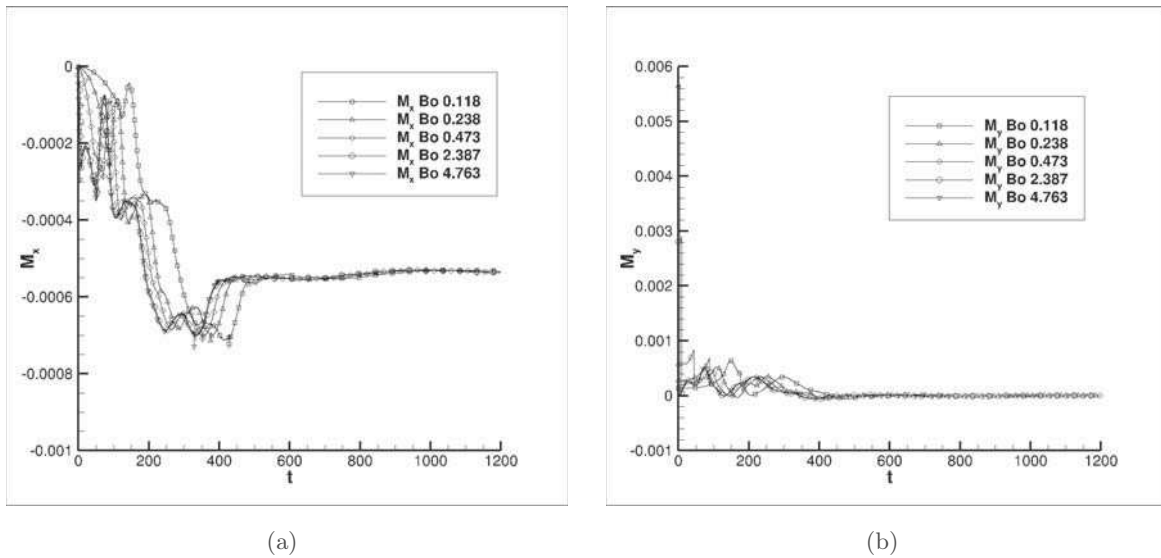


FIGURE 4.6 – Évolution des moments générés par les fluides dans le réservoir en fonction des différents nombres de Bond Bo_{imp} .

Nous retrouvons également les mêmes tendances avec les moments sur les figures 4.6(a) et 4.6(b) avec le courbe perturbateur d'une amplitude de $0.00053N \cdot m$ dans la direction \mathbf{e}_x , et un couple nul à l'équilibre dans la direction \mathbf{e}_y .

Nous pouvons donc conclure ici que dans le cadre de cette étude, le nombre de Bond impulsif n'est pas décisif sur la prédiction de la dynamique des fluides au sein du réservoir dans la gamme de Bond étudiée. Il génère tout au plus un déphasage des signaux, mais étant donné que la quantité d'énergie injectée au système reste la même, la même position d'équilibre est retrouvée à la fin

de la manœuvre. Nous allons donc maintenant regarder l'effet de la variation du nombre de Bond d'entraînement.

4.2.2 Rôle du nombre de Bond centrifuge

Dans cette partie, nous proposons une étude sur le rôle du nombre de Bond centrifuge. Le nombre de Bond impulsional est fixé tel que $Bo_{imp} = 0.473$. Durant la première phase de la manœuvre, le réservoir est soumis à l'accélération $\dot{\Omega} = 0.0002 rad.s^{-1}$. Le tableau 4.2 résume les six temps d'accélération choisis, ainsi que les vitesses de rotation et les nombres de Bond centrifuges résultants. Les simulations présentées dans cette partie sont réalisées avec un maillage 128^3 . La figure 4.7 illustre le profil des manœuvres proposées dans cette étude.

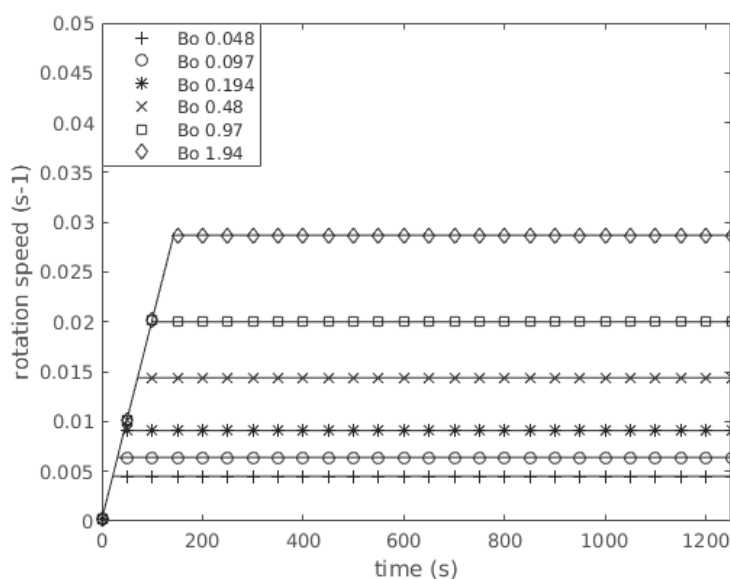


FIGURE 4.7 – Profil d'accélération de rotation $\dot{\Omega}$ en fonction du nombre de Bond (Bo_{cent}).

Bo_{cent}	0.048	0.097	0.194	0.48	0.97	1.94
Δt_{acc} (s)	22.5	32.5	45.3	72	101.3	143.5
Ω ($rad.s^{-1}$)	0.0045	0.0064	0.0091	0.0144	0.020	0.287

TABLE 4.2 – Récapitulatif des simulations effectuées sur l'étude du Bo_{cent} .

4.2.2.1 Visualisations et Trajectoires

Des visualisations des écoulements lors des différentes manœuvres sont proposées dans cette partie, avec la représentation de la trajectoire du centre de masse dans le plan xy.

Trajectoire pour le cas $Bo_{cent} = 0.048$.

La figure 4.8 montre le comportement de la bulle de gaz à différents instants pour le cas $Bo_{cent} = 0.048$.

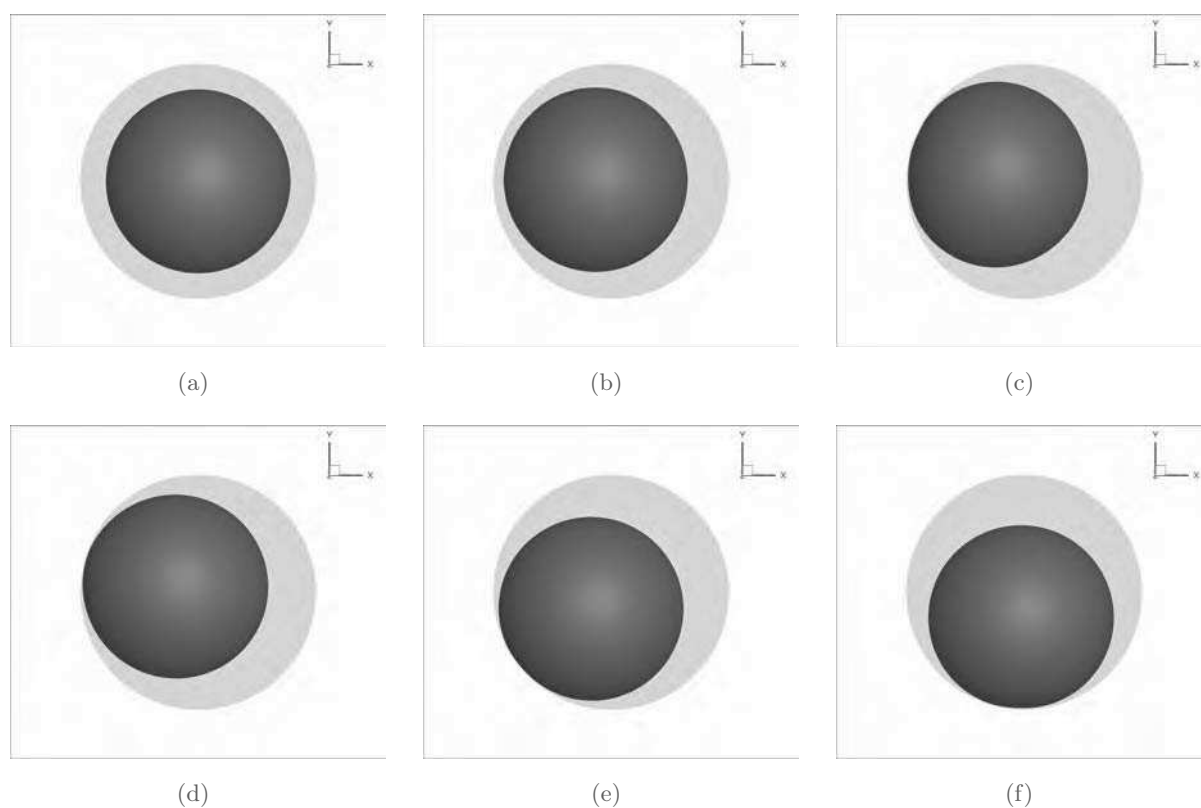


FIGURE 4.8 – Visualisation dans le plan xy de la bulle de gaz durant la manœuvre pour $Bo_{cent} = 0.048$.

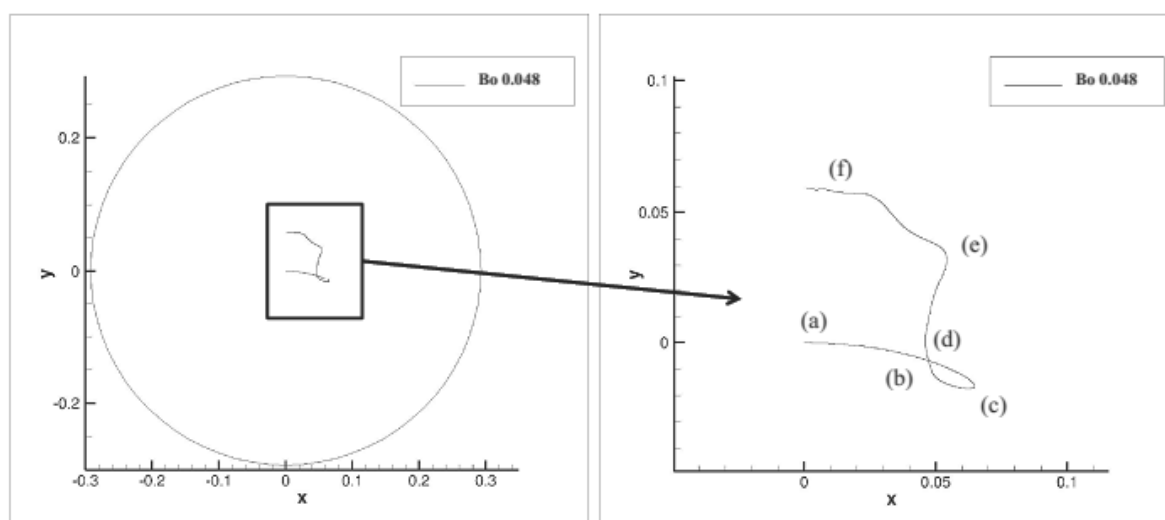


FIGURE 4.9 – Trajectoire du centre de masse dans le plan xy pour $Bo_{cent} = 0.048$.

Ces différents instants sont également reportés sur le suivi de la trajectoire présenté dans la figure 4.9. Nous pouvons y distinguer plusieurs étapes :

- a) Position initiale de la bulle
- b) Début du mouvement sous l'influence du forçage en direction de la paroi du réservoir

- c) Rebond et légère déformation de la bulle sur la paroi
- d) Retour à la forme sphérique de la bulle par les effets de capillarité
- e) Advection de la bulle le long de la paroi du réservoir (2^{nd} rebond)
- f) Position d'équilibre finale de la bulle plaquée légèrement contre la paroi du réservoir

Dans cette visualisation, deux grandes étapes ressortent de la manœuvre, à savoir le premier rebond, et la mise à l'équilibre de la bulle vers sa position finale, qui paraît directe dans cette manœuvre.

Trajectoire pour le cas $Bo_{cent} = 1.94$.

La figure 4.10 montre le comportement de la bulle de gaz à différents instants pour le cas $Bo_{cent} = 1.94$.

Ces différents instants sont également reportés sur le suivi de trajectoire présenté dans la figure 4.11. Nous pouvons y distinguer plusieurs étapes :

- a) Position initiale de la bulle
- b) Début du mouvement sous l'influence du forçage en direction de la paroi du réservoir
- c) Rebond et grande déformation de la bulle sur la paroi
- d) Advection de la bulle déformée le long de la paroi du réservoir
- e) Advection et oscillation de la bulle déformée
- f) Advection de la bulle et ballotements
- g) Position d'équilibre finale de la bulle plaquée contre la paroi du réservoir

Dans cette manœuvre, une étape supplémentaire se révèle importante par rapport à la manœuvre avec un nombre de Bond plus faible, à savoir l'étape de ballotement vers la position d'équilibre entre les points (e) et (g), qui est mise en évidence notamment sur la figure 4.11. L'analyse du comportement des bulles dans les différentes configurations est proposée en détail dans les parties suivantes, avec une première approche basée sur les forces d'inertie :

$$\mathbf{a}_{Ent} = \dot{\Omega} \times \mathbf{OM} + \Omega \times \Omega \times \mathbf{OM} + \mathbf{a}_R \quad (4.2.1)$$

avec $\dot{\Omega} = \dot{\Omega} \mathbf{e}_z$, et $\mathbf{OM} = x\mathbf{e}_x + (L + y)\mathbf{e}_y + z\mathbf{e}_z$ ce qui peut se réécrire comme :

$$\mathbf{a}_{Ent} = - \left((L + y)\dot{\Omega} + x\Omega^2 \right) \mathbf{e}_x + \left(x\dot{\Omega} - (L + y)\Omega^2 \right) \mathbf{e}_y \quad (4.2.2)$$

Dans un premier temps, le terme selon la direction $-\mathbf{e}_x$ génère le mouvement des fluides avec $(L + y)\dot{\Omega}$. Le liquide va donc être plaqué dans la direction opposée à cette force, dans la direction \mathbf{e}_x , et la bulle va être déplacée dans le sens de la force, dans la direction $-\mathbf{e}_x$ comme observé dans les figures 4.11(c) et 4.9(c). Lorsque le forçage est terminé et que la vitesse maximale de rotation est atteinte, $\dot{\Omega} = 0$, et $-(L + y)\Omega^2 \mathbf{e}_y$ va dominer l'écoulement et plaquer la bulle dans la même direction.

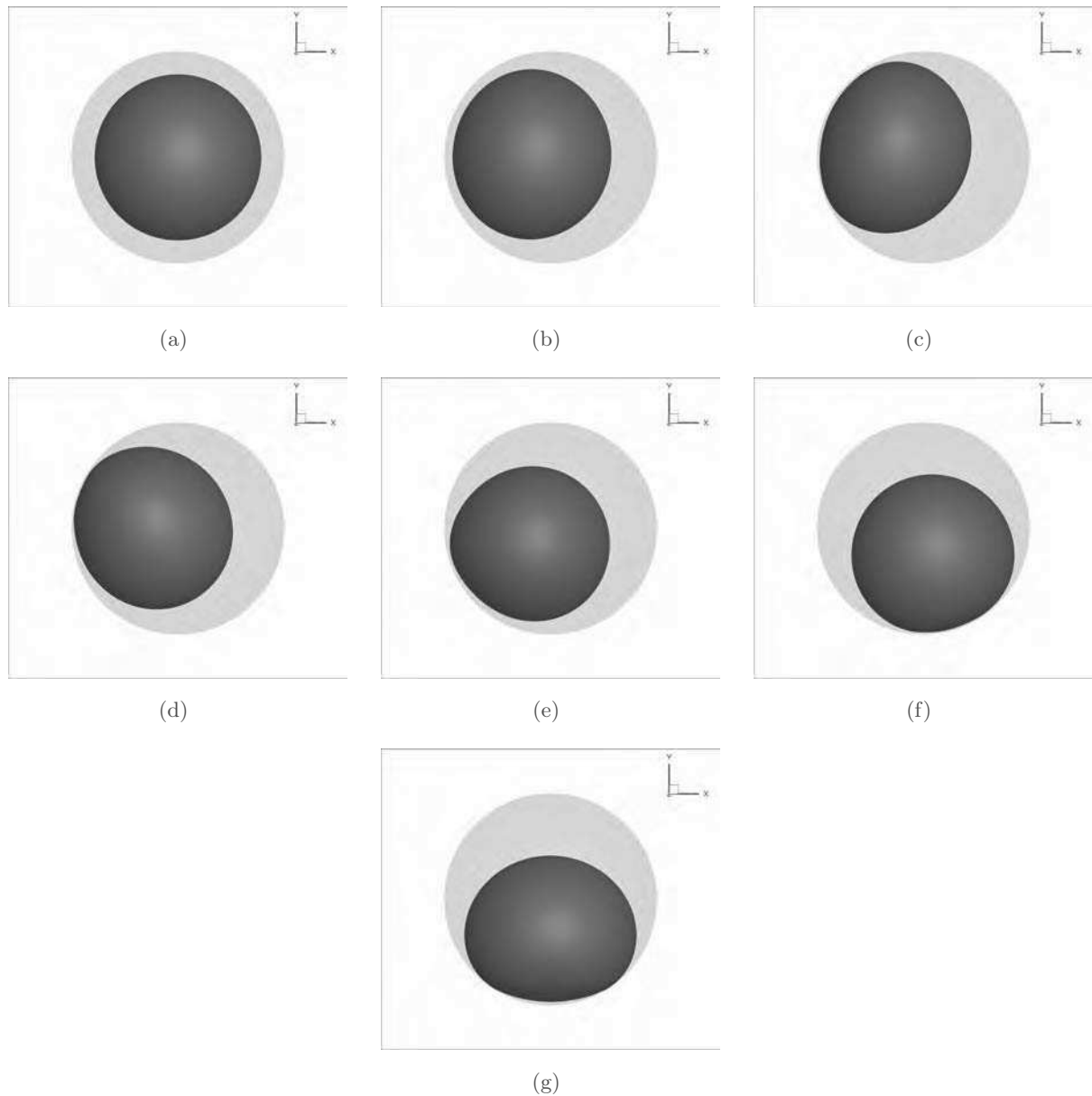


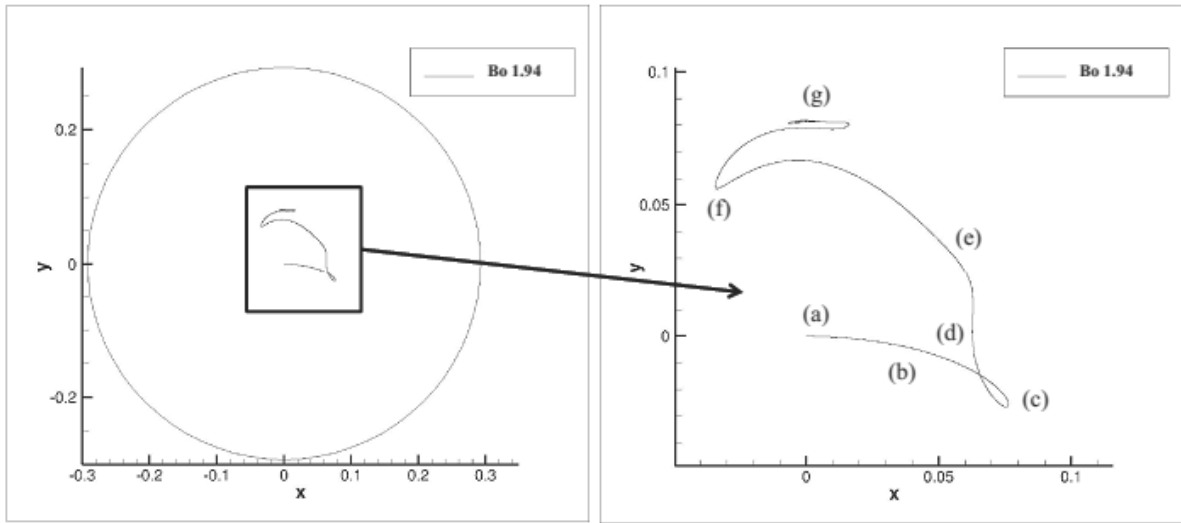
FIGURE 4.10 – Visualisation dans le plan xy de la bulle de gaz durant la manœuvre pour $Bo_{cent} = 1.94$.

4.2.2.2 Centre de Masse

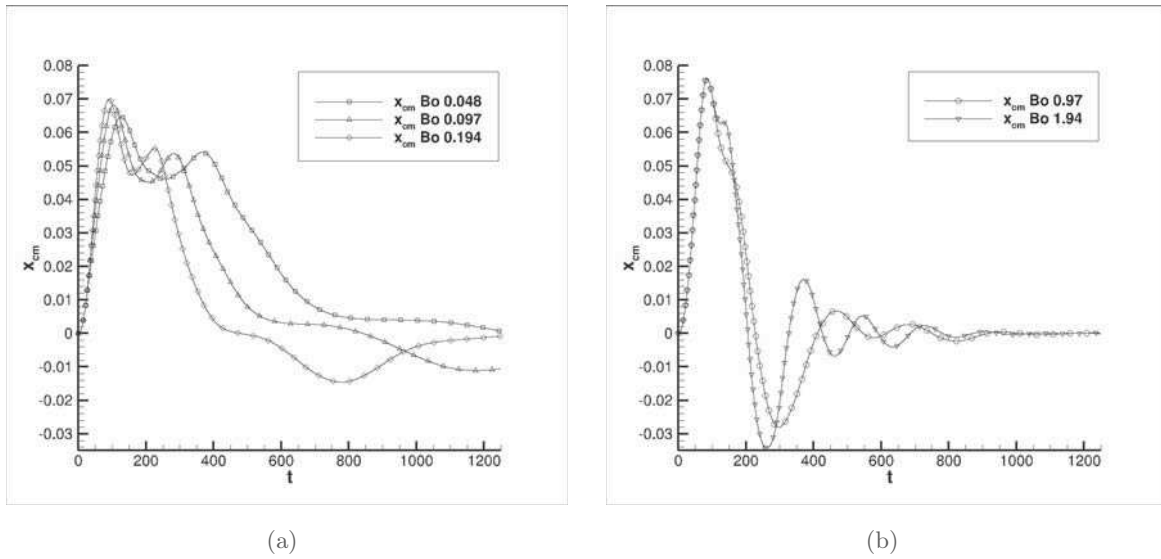
Centre de Masse dans la direction e_x .

Les figures 4.12(a) et 4.12(b) représentent l'évolution de la position du centre de masse dans la direction e_x au cours de la manœuvre pour différents nombres de Bond. Dans la figure 4.12(a) les nombres de Bond petits devant 1 présentent une certaine similitude. Cette similitude est mise en évidence dans la figure 4.13(a), où les variables sont adimensionnées de la façon suivante :

$$x_{cm}^* = \frac{x_{cm}}{D_{tank}} \quad \text{et} \quad t^* = \frac{t\Omega}{2\pi} \quad (4.2.3)$$

FIGURE 4.11 – Trajectoire du centre de masse dans le plan xy pour $Bo_{cent} = 1.94$.

où D_{tank} est le diamètre du réservoir. En effet les courbes se superposent, ce qui met en évidence une dépendance linéaire des temps caractéristiques avec le nombre de Bond. La figure 4.13(b) qui présente des nombres de Bond supérieurs à ceux proposés précédemment montre une transition, car les courbes ne superposent plus.

FIGURE 4.12 – Position du centre de masse (m) dans la direction e_x au cours de la manœuvre.

Un raisonnement sur les échelles de temps peut expliquer certaines choses. Un temps caractéristique de l'écoulement illustrant les effets inertiels peut être assimilé à Ω . Le temps classiquement choisi pour illustrer les phénomènes liés aux effets de capillarité est la fréquence de Rayleigh, définie comme $\omega_{Rayleigh} = \sqrt{\frac{\sigma}{\rho_l D_0^3}}$ avec D_0 le diamètre initial de la bulle de gaz. L'application numérique donne $\omega_{Rayleigh} = 0.0186s^{-1}$. Les valeurs de Ω sont proposées dans le tableau 4.2. Si nous regardons le comportement de la bulle de gaz durant la manœuvre, celle-ci est propulsée dans un premier temps

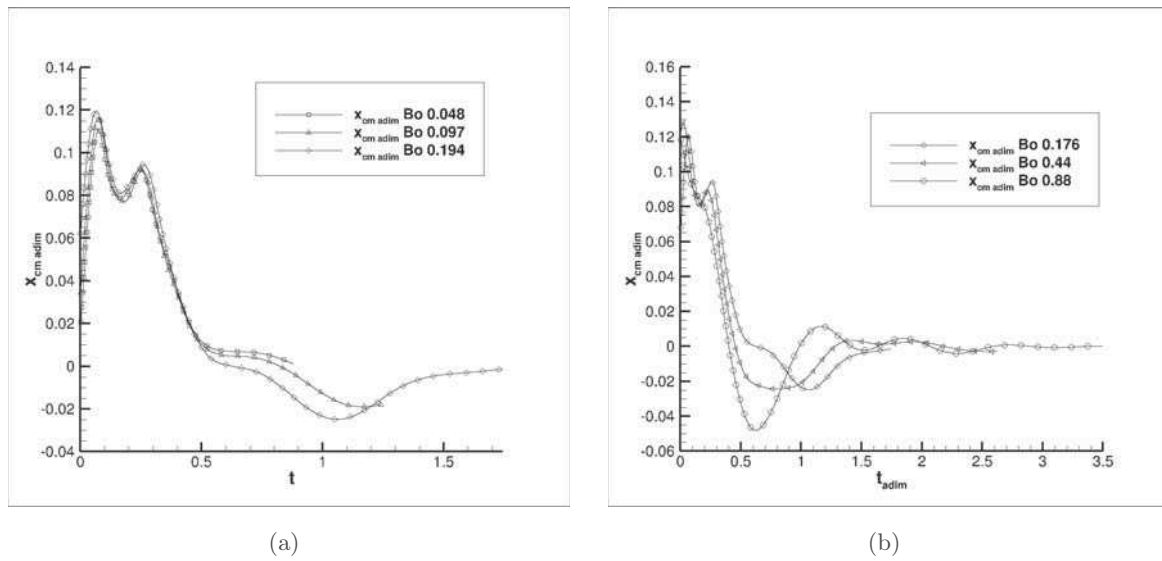


FIGURE 4.13 – Position du centre de masse (m) dans la direction \mathbf{e}_x au cours de la manœuvre.

sur la paroi, puis elle rebondit. Les différents nombres de Bond proposés permettent de comparer des cas où les effets de capillarité dominent comme dans la figure 4.12(a), et d'autres où l'inertie domine (figure 4.12(b)). Lorsque la capillarité domine, la bulle n'est que très peu déformée, alors qu'elle l'est d'avantage lorsque les effets inertiels dominent. De plus en comparant visuellement le comportement de la bulle, comme dans la figure 4.14, nous pouvons constater que pour de faibles nombres de Bond, la bulle se déforme peu, et revient à sa forme initiale avant de continuer son mouvement. Dans le cas des nombres de Bond plus élevés, le temps caractéristique des effets inertiels est plus faible que celui des effets de capillarité. La bulle n'a donc pas le temps de retourner vers une forme sphérique avant d'être advectée. Ceci va induire un couplage entre le mouvement de la bulle et les oscillations liées à cette déformation. Dans ce cas, les déplacements de la bulle ne dépendront plus linéairement de Ω comme cela a été observé dans le cas des nombres de Bond inférieurs.

La figure 4.12(b) présente des nombres de Bond plus proches de l'unité. Dans celle-ci, une phase de ballotement après le forçage apparaît très nettement, avec une amplitude décroissante et une période qui évolue dans le temps. Les ballotements s'effectuent dans la direction \mathbf{e}_x puisque le bras de levier L est dans la direction \mathbf{e}_y . Le tableau 4.3 présente les périodes de ballotements ainsi que leurs caractéristiques.

Nous pouvons observer dans un premier temps une dissipation d'énergie puisque l'amplitude des deux signaux présentés dans la figure 4.12(b) diminue vers une position d'équilibre où la bulle est plaquée à la paroi dans la direction $-\mathbf{e}_y$. De plus, les périodes de ballotement ne sont pas régulières. Ceci met en évidence la présence de déformations suivant plusieurs modes d'oscillations. Une décomposition en harmonique sphérique de la forme de l'interface permettrait de mieux comprendre ce transfert d'énergie, et sera effectuée dans de futurs travaux.

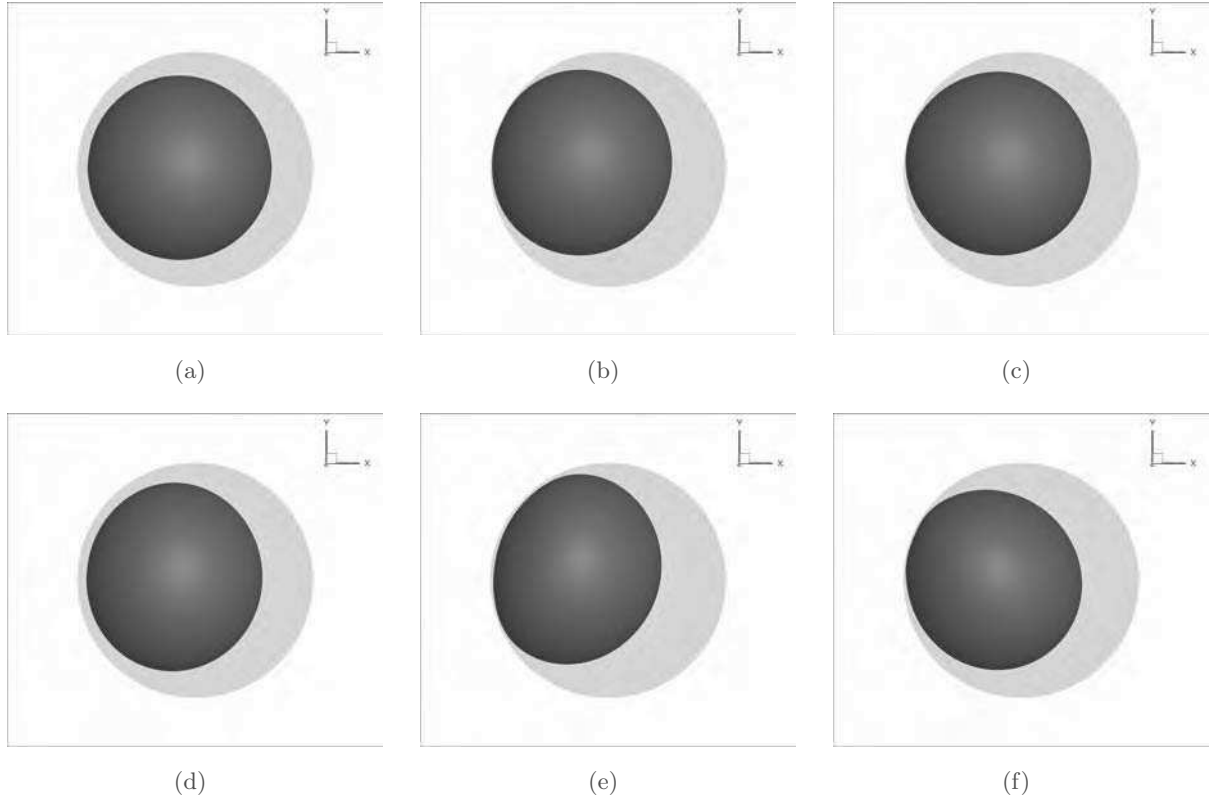


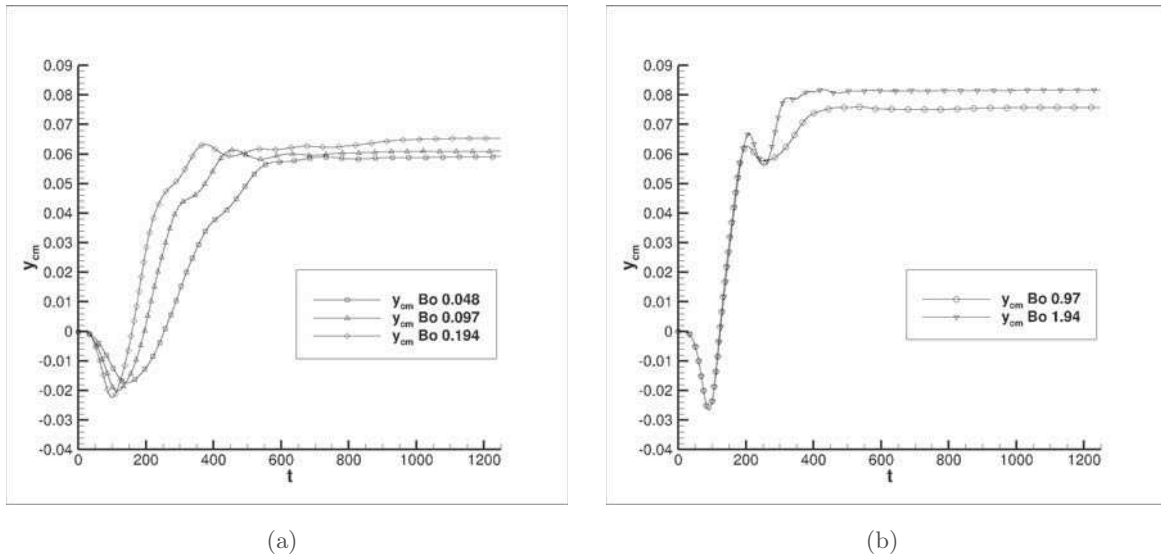
FIGURE 4.14 – Visualisation dans le plan xy de la bulle de gaz durant le premier rebond pour des nombres de Bond de $Bo_{cent} = 0.048$ (a,b,c) et $Bo_{cent} = 1.94$ (d,e,f).

demi-période	$Bo_{cent} = 0.97$		$Bo_{cent} = 1.94$	
	Δ Amplitude	Demi-Période (s)	Δ Amplitude	Demi-Période (s)
1	0.078	154	0.096	121
2	0.0347	168	0.04	107
3	0.008	109	0.0227	95
4	0.004	106	0.017	84
5	0.0047	138	0.009	86
6	0.0025	137	0.006	95
7	-	-	0.003	90
8	-	-	0.00015	93

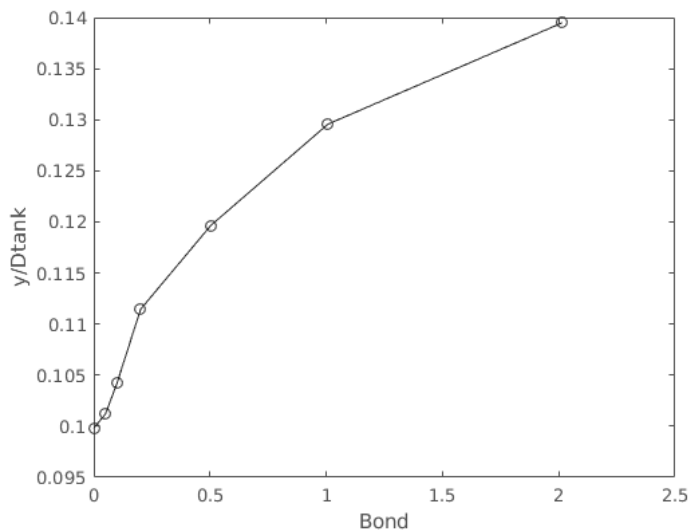
TABLE 4.3 – Caractéristiques des différentes périodes de ballotement post-forçage pour les nombres de Bond $Bo_{cent} = 0.97$ et $Bo_{cent} = 1.94$.

Centre de Masse dans la direction e_y .

Les figures 4.15(a) et 4.15(b) représentent l'évolution de la position du centre de masse dans la direction e_y au cours de la manœuvre pour différents nombres de Bond. La figure 4.15(a) présente à nouveau une similitude entre les différentes courbes. Nous pouvons également constater que toutes les simulations convergent vers un état d'équilibre, la bulle positionnée contre la paroi dans la

FIGURE 4.15 – Position du centre de masse (m) dans la direction \mathbf{e}_y au cours de la manœuvre.

direction $-\mathbf{e}_y$, avec toutefois une valeur finale qui diffère en fonction des nombres de Bond. Ceci se traduit physiquement par un aplatissement plus ou moins important de la bulle sur la paroi. Les valeurs de la position du centre de masse à l'équilibre dans la direction \mathbf{e}_y sont tracées sur la figure 4.16. Un point asymptotique a été rajouté, et correspond au cas où la bulle est posée contre la paroi tout en étant parfaitement sphérique (ce qui correspondrait à un cas où $\Omega = \text{Bo}_{cent} = 0$). Un autre régime asymptotique est celui où la bulle ne pourrait plus être écrasée d'avantage sur la paroi du réservoir. Ceci fera l'objet d'une prochaine étude ayant pour but de caractériser à l'aide d'une loi analytique la position du centre de masse dans la direction \mathbf{e}_y à l'équilibre.

FIGURE 4.16 – Position du centre de masse dans la direction \mathbf{e}_y adimensionnée par le diamètre du réservoir D_{tank} en fonction du nombre de Bond (Bo_{cent}).

4.2.2.3 Diamètre d'étalement de la bulle

Maintenant que la position d'équilibre de la bulle est définie dans la direction \mathbf{e}_y , nous mesurons le diamètre d'étalement de la bulle dans le plan yz . Cette mesure est effectuée à l'aide des visualisations proposées par la figure 4.17

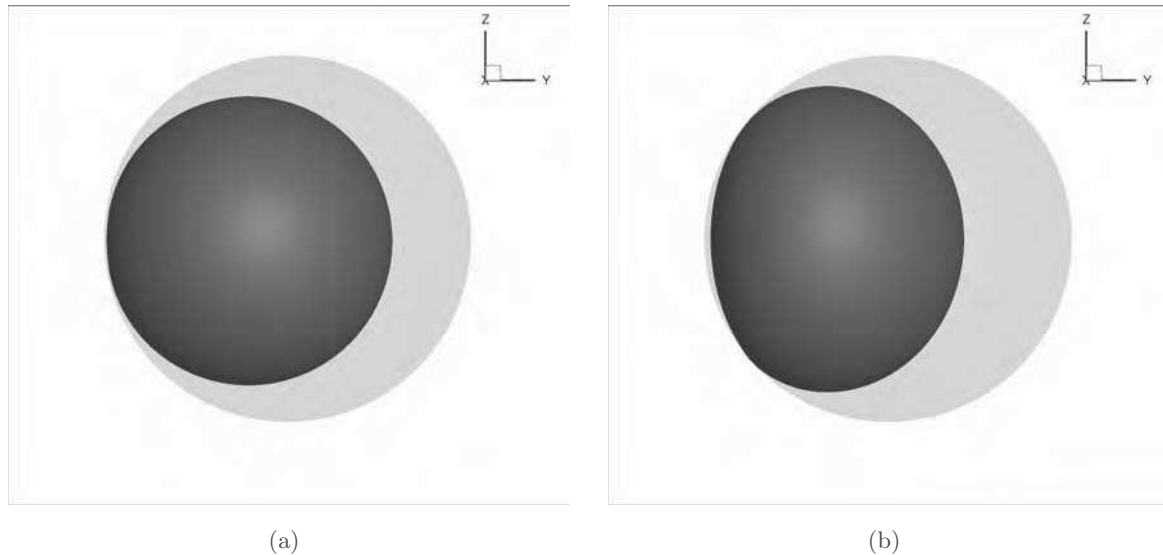


FIGURE 4.17 – Étalement de la bulle lorsque l'équilibre est atteint pour les nombres de Bond $Bo_{cent} = 0.048$ et $Bo_{cent} = 1.94$.

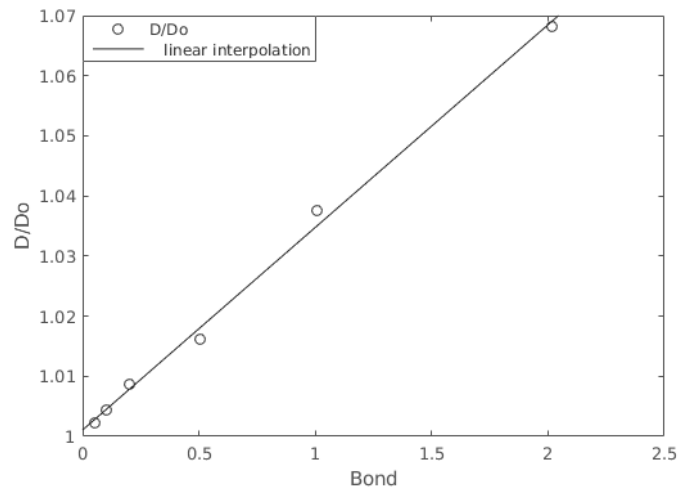


FIGURE 4.18 – Diamètre d'étalement de la bulle dans le plan xz adimensionnée par le diamètre du réservoir D_{tank} en fonction du nombre de Bond (Bo_{cent}).

La mesure de ce diamètre d'étalement, adimensionnée par le diamètre initial de la bulle D_0 , est proposée dans la figure 4.18, ainsi qu'une interpolation linéaire des données. Ce que nous observons ici est que le diamètre d'étalement semble dépendre linéairement du nombre de Bond avec :

$$\frac{D}{D_0} = 0.0407\text{Bo}_{cent} + 1 \quad (4.2.4)$$

Toutefois, cette équation ne caractérise pas pleinement le problème car elle ne tient pas compte du remplissage qui pourrait intervenir dans une loi de la forme suivante :

$$\frac{D}{D_0} = \alpha f\left(\frac{D_0}{D_{tank}}\right) \text{Bo}_{cent} + 1 \quad (4.2.5)$$

avec α une constante et f une fonction à déterminer. Des simulations supplémentaires avec différents taux de remplissages sont nécessaires pour caractériser ce phénomène.

4.2.2.4 Forces et Moments

Les principales préoccupations pour les industriels du domaine spatial sont les répercussions de la dynamique engendrée par les mouvements du centre de masse liquide dans le réservoir, avec les forces et les moments résultants des manœuvres effectuées.

Forces

La figure 4.19 présente les forces décomposées dans chaque direction, résultantes des efforts générés par le fluide dans le réservoir en réponse à la manœuvre effectuée.

La figure 4.19(a) ne présente que les courbes de force dans la direction \mathbf{e}_x pour des nombres de Bond $\text{Bo}_{cent} = 0.044$ et $\text{Bo}_{cent} = 1.94$ afin de montrer le contraste entre les amplitudes des efforts subis par le réservoir. Les valeurs maximales et minimales de ces efforts sont reportées pour une plus large gamme de nombres de Bond dans les tableaux 4.4 et 4.5. La figure 4.19(b) montre la linéarité de la force résultante du forçage externe dans la direction \mathbf{e}_y par rapport à l'évolution du nombre de Bond, comme le démontre le tableau 4.5. Enfin la courbe 4.19(c) confirme que les efforts selon la direction \mathbf{e}_z sont négligeables, puisque le mouvement s'effectue dans le plan xy.

	$Fx_{max}(mN)$	$Fx_{min}(mN)$	$Fx_{fin}(mN)$
$\text{Bo}_{cent} = 0.048$	0.737	0	$\simeq 0$
$\text{Bo}_{cent} = 0.097$	1.5	-0.06	$\simeq 0$
$\text{Bo}_{cent} = 0.194$	2.2	-0.23	$\simeq 0$
$\text{Bo}_{cent} = 0.48$	4.2	-0.48	$\simeq 0$
$\text{Bo}_{cent} = 0.97$	4.7	-1	$\simeq 0$
$\text{Bo}_{cent} = 1.94$	8.78	-3.8	$\simeq 0$

TABLE 4.4 – Caractéristiques des forces résultantes de la dynamique du fluide appliquées sur le réservoir dans la direction \mathbf{e}_x post-forçage.

Moments

De même que pour les forces, la figure 4.20 présente les moments décomposés dans les directions \mathbf{e}_x et \mathbf{e}_y résultants des efforts générés par le fluide dans le réservoir en réponse à la manœuvre effectuée.

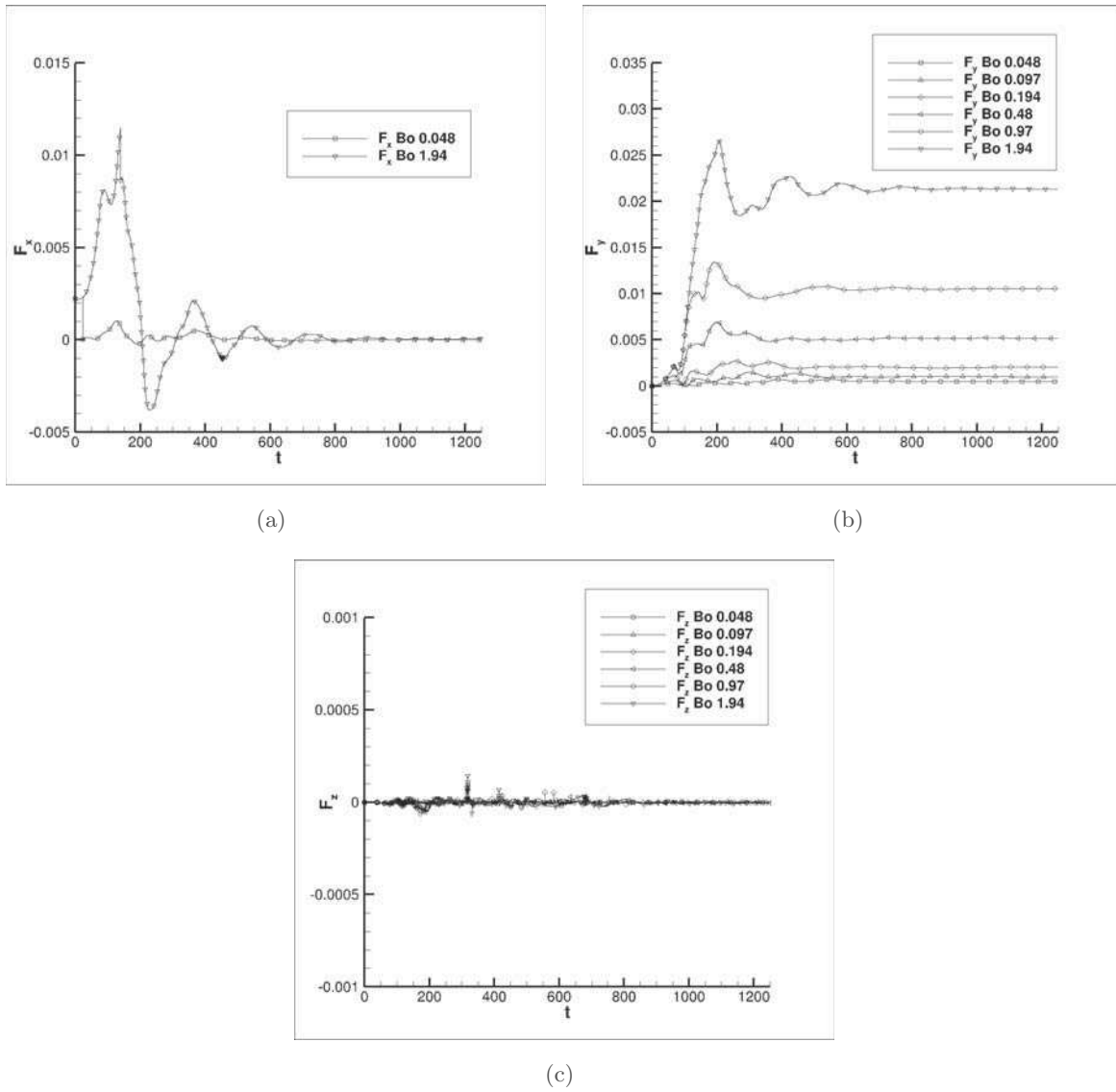


FIGURE 4.19 – Forces (N) résultantes de la dynamique de l'écoulement engendré par la manœuvre imposée sur le réservoir.

	$F_{y_{max}}(mN)$	$F_{y_{min}}(mN)$	$F_{y_{fin}}(mN)$
$Bo_{cent} = 0.048$	0.7	0	0.5
$Bo_{cent} = 0.097$	1.4	0	1
$Bo_{cent} = 0.194$	2.7	0	2
$Bo_{cent} = 0.48$	6.9	0	5.17
$Bo_{cent} = 0.97$	13.3	0	10.55
$Bo_{cent} = 1.94$	26.5	0	21

TABLE 4.5 – Caractéristiques des forces résultantes de la dynamique du fluide appliquées sur le réservoir dans la direction \mathbf{e}_y post-forçage.

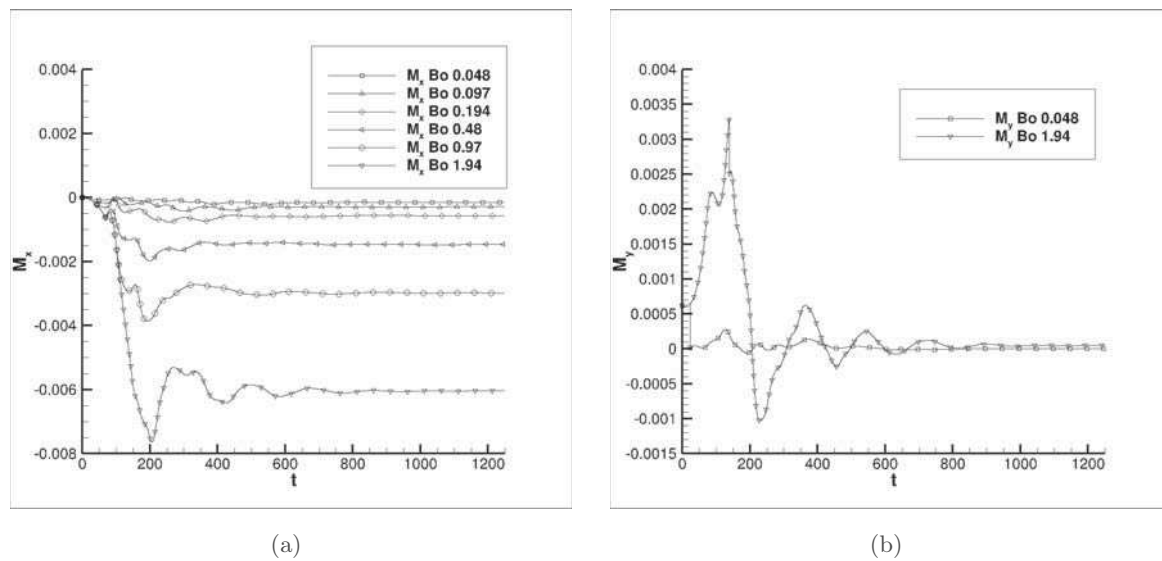


FIGURE 4.20 – Moments ($N \cdot m$) résultants de la dynamique de l'écoulement engendré par la manœuvre imposée sur le réservoir.

La figure 4.20(b) ne présente que les courbes de moment dans la direction \mathbf{e}_y pour des nombres de Bond $Bo_{cent} = 0.048$ et $Bo_{cent} = 1.94$ afin de montrer le contraste entre les amplitudes des efforts subis par le réservoir. Les valeurs maximales et minimales de ces moments sont reportées pour une plus large gamme de nombres de Bond dans le tableau 4.6. La figure 4.20(a) montre également la linéarité du moment résultant du forçage externe dans la direction \mathbf{e}_y par rapport à l'évolution du nombre de Bond, comme le démontre le tableau 4.6.

	$Mx_{max}(mN \cdot m)$	$Mx_{min}(mN \cdot m)$	$Mx_{fin}(mN \cdot m)$
$Bo_{cent} = 0.048$	0	-0.2	-0.14
$Bo_{cent} = 0.097$	0	-0.42	-0.28
$Bo_{cent} = 0.194$	0	-0.77	-0.57
$Bo_{cent} = 0.48$	0	-1.9	-1.46
$Bo_{cent} = 0.97$	0	-3.8	-2.98
$Bo_{cent} = 1.94$	0	-7.6	-6.03

TABLE 4.6 – Caractéristiques des moments résultants de la dynamique du fluide appliquées sur le réservoir dans la direction \mathbf{e}_x post-forçage.

4.2.3 Convergence des simulations

Dans cette partie, la convergence des simulations et la validité des résultats sont discutées. Les figures 4.21 et 4.22 présentent des simulations pour des nombres de Bond $Bo_{imp} = 0.47$ et $Bo_{cent} = 0.194$ avec les maillages 64^3 et 128^3 . Bien que les courbes ne soient pas parfaitement superposées, celles-ci permettent de montrer une bonne convergence des simulations et d'être confiants dans nos résultats.

Nous venons de voir la première étape d'une manœuvre qu'un satellite peut réaliser dans l'espace,

	$My_{max}(mN \cdot m)$	$My_{min}(mN \cdot m)$	$My_{fin}(mN \cdot m)$
$Bo_{cent} = 0.048$	0.28	-0.057	0.002
$Bo_{cent} = 0.097$	0.46	-0.069	-0.006
$Bo_{cent} = 0.194$	0.6	-0.062	0.004
$Bo_{cent} = 0.48$	1.15	-0.1	0.007
$Bo_{cent} = 0.97$	1.2	-0.33	0.00256
$Bo_{cent} = 1.94$	2.62	-1	0.05

TABLE 4.7 – Caractéristiques des moments résultants de la dynamique du fluide appliquées sur le réservoir dans la direction \mathbf{e}_y post-forçage.

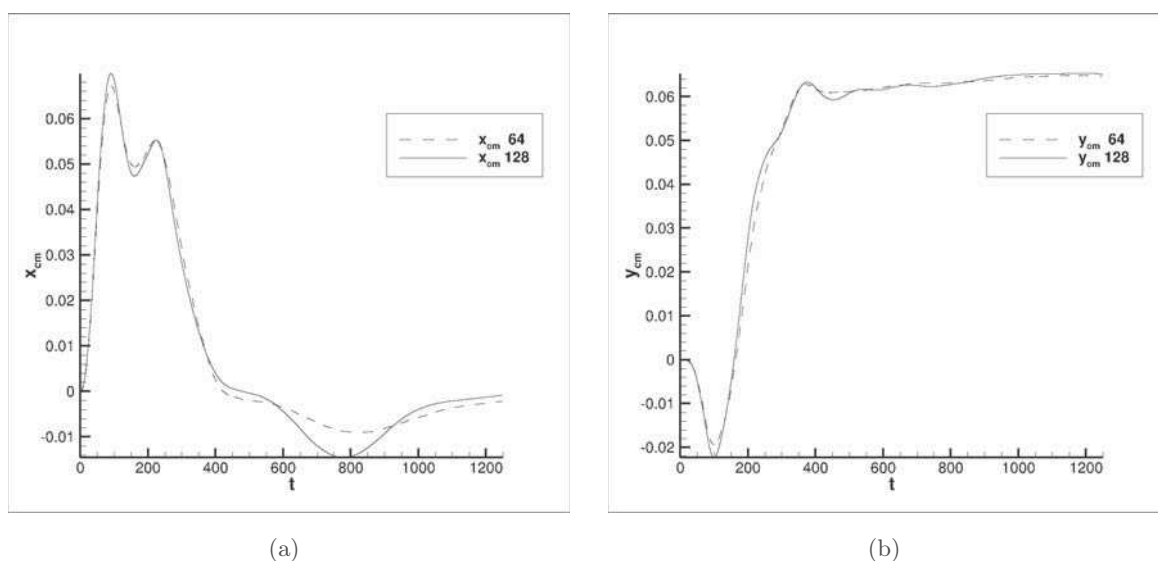


FIGURE 4.21 – Convergence sur les positions des centres de masse (a) dans la direction \mathbf{e}_x et (b) dans la direction \mathbf{e}_y pour des maillages 64^3 et 128^3 .

et toute la complexité des phénomènes pouvant apparaître. Toutefois après cette étape de forçage vient une étape de relaxation, engendrant une nouvelle dynamique de l'écoulement, qui n'est plus soumise à aucune force extérieure. Nous présentons succinctement dans la suite de ce chapitre une manœuvre faisant apparaître une phase de forçage ainsi qu'une phase de relaxation.

4.3 Étude de la manœuvre GAP/SUP

La manœuvre GAP/SUP est une des manœuvres les plus classiques en terme de contrôle et de maintien à poste du satellite. En effet cette manœuvre permet au satellite d'effectuer une rotation de 180° sur lui-même pour s'orienter soit vers la Terre pour effectuer des mesures ou retransmettre des informations au segment sol, soit se réorienter vers le soleil de manière à charger ses panneaux solaires, comme l'illustre la figure 4.23.

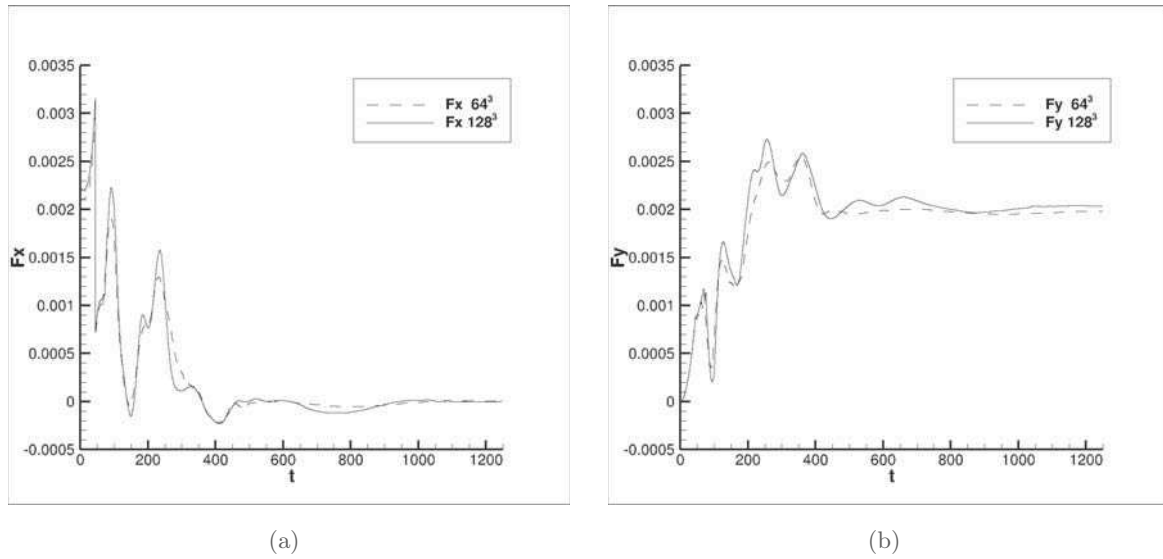


FIGURE 4.22 – Convergence sur les forces (a) dans la direction \mathbf{e}_x et (b) dans la direction \mathbf{e}_y pour des maillages 64^3 et 128^3 .

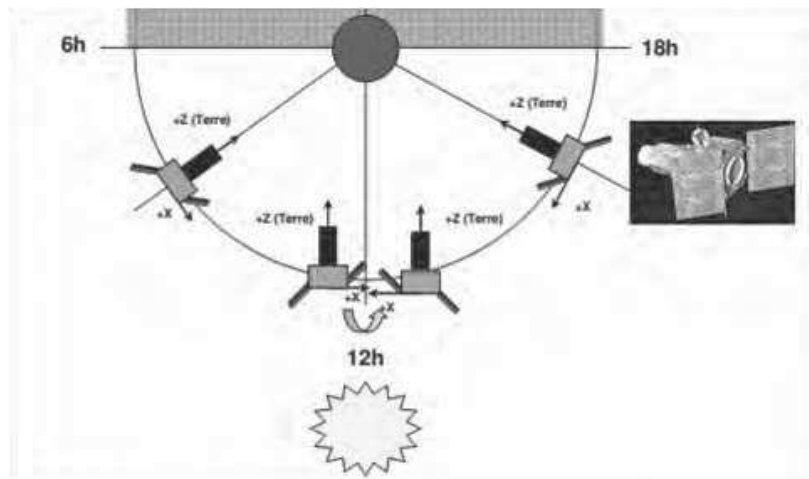


FIGURE 4.23 – Schématisation de la manœuvre GAP/SUP.

4.3.1 Description de la manœuvre GAP/SUP

La modélisation suivante porte sur un réservoir sphérique, de rayon $R_{tank} = 0.2925m$, avec un remplissage de 50%. Le domaine modélisé est de dimension $[-0.3 : 0.3] \times [-0.3 : 0.3] \times [-0.3 : 0.3]$ avec un maillage 128^3 . Le liquide utilisé est de l'hydrazine et possède les caractéristiques suivantes avec $\rho_l = 1004kg.m^{-3}$, $\mu_l = 9.13 \times 10^{-4}kg.m^{-1}.s^{-1}$, et le gaz est l'Hélium pressurisé avec $\rho_g = 2.41kg.m^{-3}$, $\mu_g = 1.99 \times 10^{-5}kg.m^{-1}.s^{-1}$, avec le coefficient de tension de surface $\sigma = 0.0667N.m^{-1}$. L'hydrazine est parfaitement mouillant, ce qui implique que $\theta_s = 0^\circ$. Le centre de la bulle de gaz, de rayon $R_{gaz} = 0.229m$, est initialement confondu avec le centre géométrique du réservoir. Le profil d'accélération est présenté dans la figure 4.24, avec l'accélération et la vitesse de rotation selon l'axe \mathbf{e}_z . Nous y retrouvons plusieurs phases : l'accélération jusqu'à atteindre la vitesse de rotation ($Bo_{imp} = 3.8$), puis une rotation à vitesse constante ($Bo_{cent} = 0.75$ et $Bo_{imp} = 0$), une décélération

($Bo_{imp} = 3.8$), puis une phase de relaxation ($Bo_{cent} = 0$ et $Bo_{imp} = 0$). Remarquons que le nombre d'Ohnesorge est faible ($Oh \simeq 2.3 \times 10^{-4}$), mais qu'il devient le seul nombre adimensionnel non-nul lors de la phase de relaxation. Le mécanisme de relaxation dépendra donc fortement des effets de dissipation visqueuse.

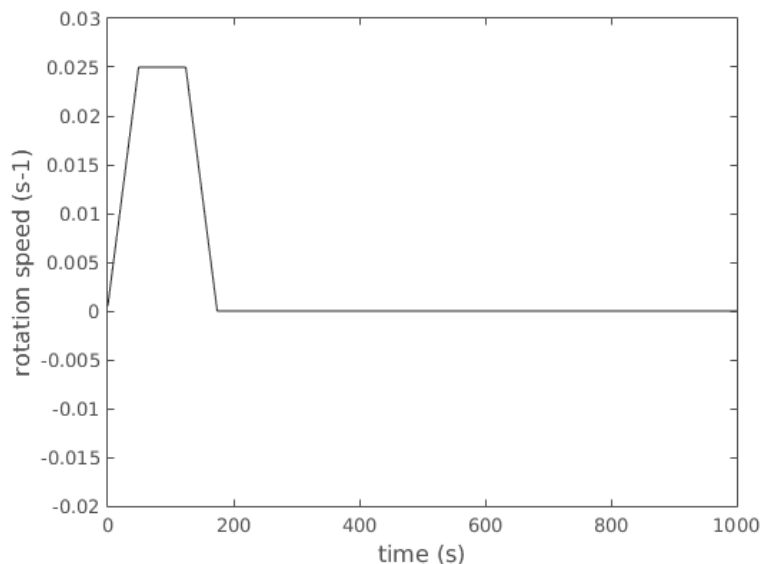


FIGURE 4.24 – Schématisation de la manœuvre GAP/SUP.

4.3.2 Visualisation et Trajectoire

La trajectoire du centre de masse des fluides dans le réservoir est proposée dans la figure 4.25. Celle-ci est également illustrée par les visualisations durant les phases de forçage et de relaxation dans les figures 4.26 et 4.27.

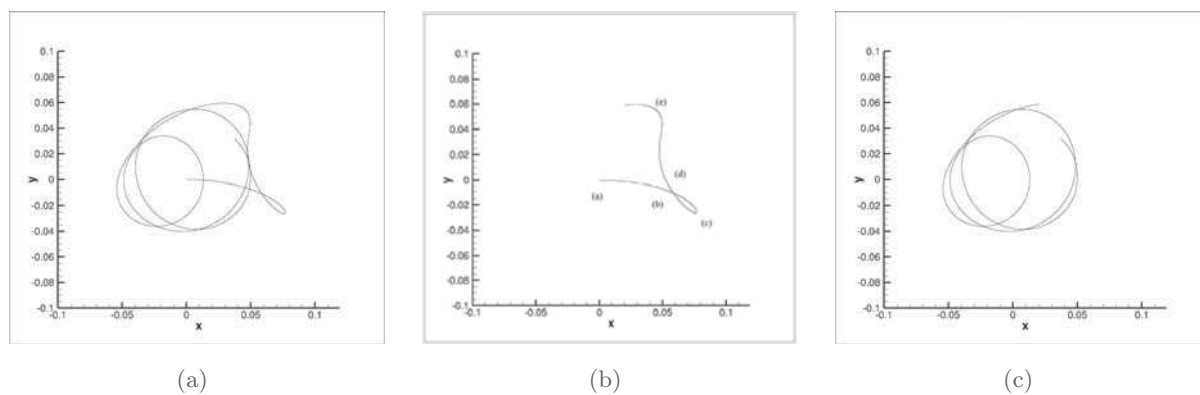


FIGURE 4.25 – Trajectoire du centre de masse du réservoir dans le plan xy , (a) global, (b) durant la phase de forçage, et (c) durant la phase de relaxation.

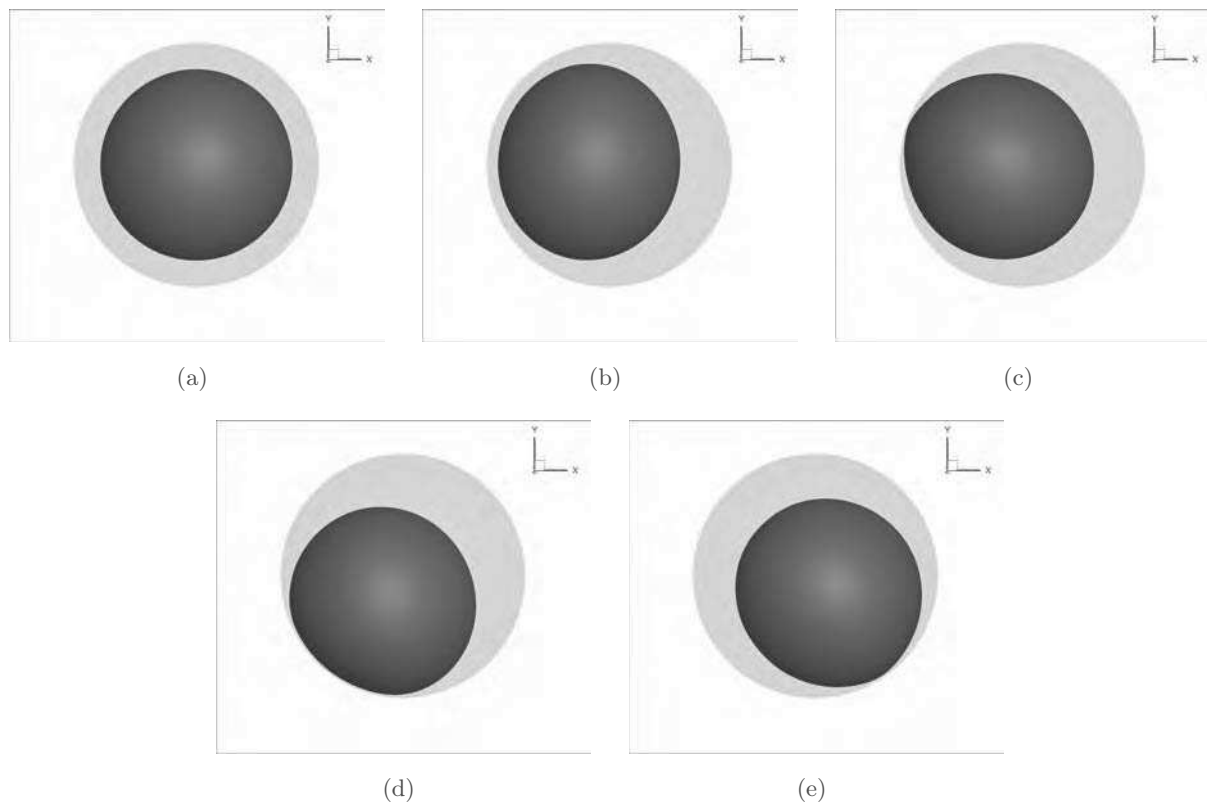


FIGURE 4.26 – Visualisation de la dynamique de la bulle durant la phase de forçage.

Phase de forçage

Les différents instants de la trajectoire durant la phase de forçage sont illustrés par les figures 4.25(b) et 4.26 avec les étapes suivantes :

- a) Position initiale de la bulle
- b) Début du mouvement sous l'influence du forçage en direction de la paroi du réservoir
- c) Rebond et grande déformation de la bulle sur la paroi
- d) Advection de la bulle déformée le long de la paroi du réservoir
- e) Advection et oscillation de la bulle déformée

Nous pouvons ici constater les similitudes de comportement des fluides dans cette étude avec l'étude faite précédemment sur la phase de forçage.

Phase de relaxation

Les différents instants de la trajectoire durant la phase de relaxation sont proposées dans les figures 4.25(c) et 4.27. Durant la relaxation, la bulle parcourt la paroi du réservoir, en se rapprochant de sa forme sphérique initiale au fur et à mesure de la stabilisation de l'écoulement. Enfin la vitesse de déplacement de la bulle est très fortement ralentie. C'est que nous verrons sur les courbes du centre de masse, avec la période du signal qui s'allonge au cours du temps.

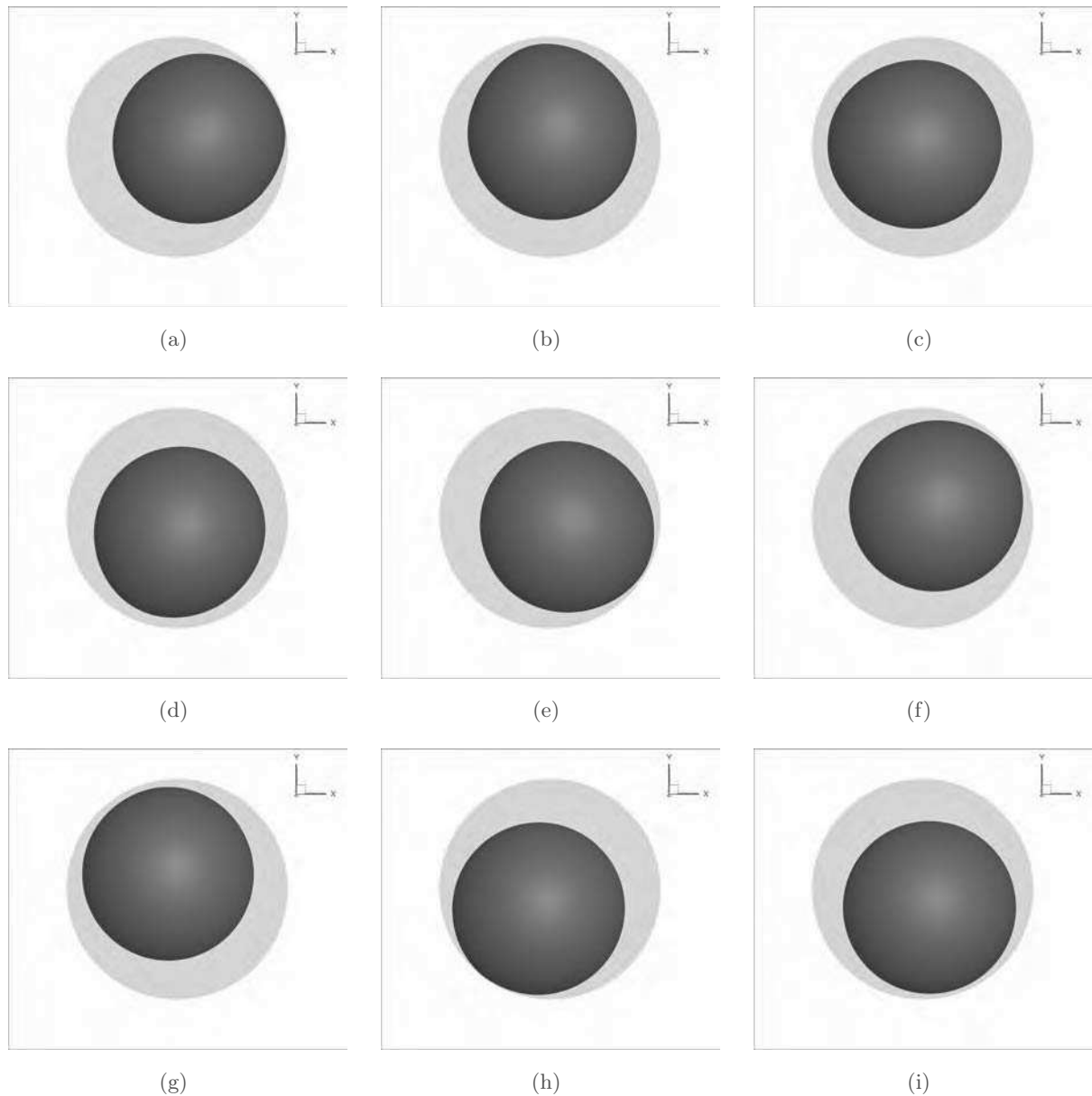


FIGURE 4.27 – Visualisation de la dynamique de la bulle durant la phase de relaxation.

4.3.3 Centre de masse

L'évolution de la position du centre de masse au cours du temps dans les directions \mathbf{e}_x et \mathbf{e}_y est proposée dans les figures 4.28(a) et 4.28(b). Tout comme pour la trajectoire, nous constatons une certaine ressemblance du signal durant les premiers instants de la modélisation à l'étude faite précédemment sur forçage. La décélération ramène toutefois la bulle vers le centre du réservoir avec une certaine inertie de rotation, ce qui la fait tourner autour du centre de masse géométrique du réservoir.

La première boucle observée entre 200s et 300s a un rayon plus petit que les deux suivantes, ce qui se traduit sur les figures 4.28(a) et 4.28(b) par une amplitude du signal plus faible. Ensuite, l'amplitude est plus régulière, avec néanmoins la période qui varie. Encore une fois la décomposition

en harmonique sphérique semble être un bon moyen pour analyser le phénomène, et sera appliquée dans une prochaine étude.

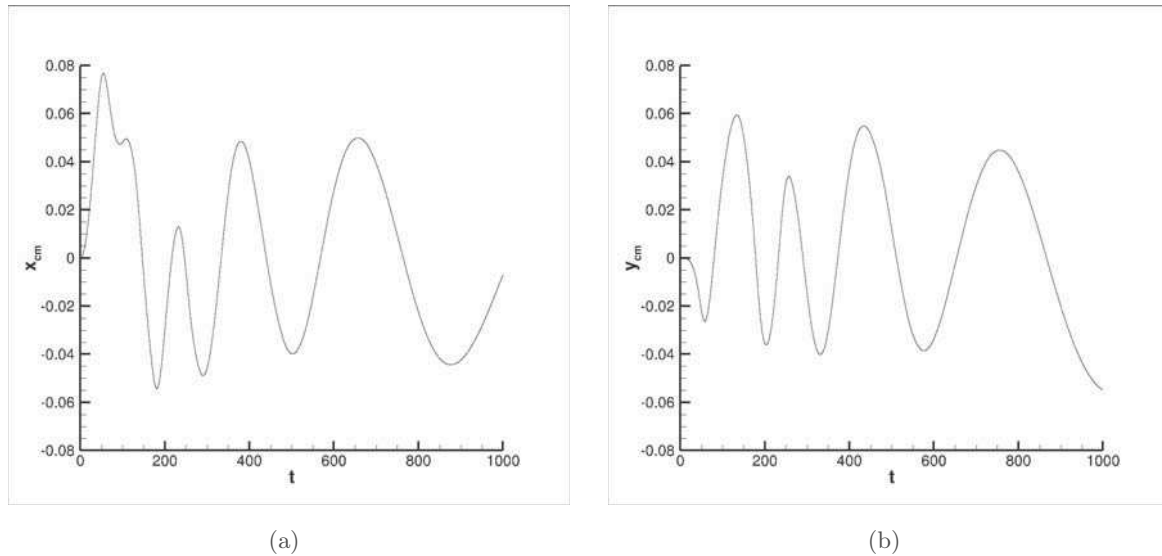


FIGURE 4.28 – Position du centre de masse (a) dans la direction \mathbf{e}_x et (b) dans la direction \mathbf{e}_y .

4.3.4 Forces et moments

Les figures 4.29(a), 4.29(b) et 4.29(c) présentent respectivement les efforts dans les directions \mathbf{e}_x , \mathbf{e}_y et \mathbf{e}_z . Dans un premier temps, nous pouvons constater que les efforts diminuent après la période de forçage, les maximums des forces étant atteints à la fin des phases d'accélération et de décélération. De plus, les efforts tendent bien vers 0 à la fin de la phase de relaxation. Enfin les efforts dans la direction \mathbf{e}_z sont nuls tout au long de la manœuvre, montrant que la symétrie du mouvement est préservée.

Les figures 4.30(a) et 4.30(b) présentent respectivement les moments résultants des efforts dans les directions \mathbf{e}_x , et \mathbf{e}_y . Tout comme pour les forces, nous pouvons observer que les couples de plus grandes amplitudes apparaissent à la fin de la phase de décélération.

Durant la phase de relaxation, de petites oscillations apparaissent sur les forces courbes et les moments. Ces oscillations ont des périodes comprises entre $35s$ et $42s$ ce qui correspond à des fréquences entre $0.0285s^{-1}$ à $0.0238s^{-1}$. Dans le cadre de cette simulation, la fréquence de Rayleigh définie précédemment est d'environ $0.0263s^{-1}$.

4.3.5 Convergence des simulations

Dans cette partie, la convergence des simulations est de nouveau discutée. Les figures 4.31 et 4.32 présentent respectivement la position du centre de masse et les forces générées par la dynamique des fluides dans le réservoir avec les maillages 64^3 et 128^3 . Nous pouvons constater que les résultats sont convergés pour la phase de forçage, aussi bien sur le centre de masse que sur les efforts. Toutefois un déphasage des courbes apparaît lors de la relaxation. Ceci montre la nécessité de valider les résultats obtenus dans cette partie à l'aide de simulations plus résolues avec un maillage 256^3 . Les

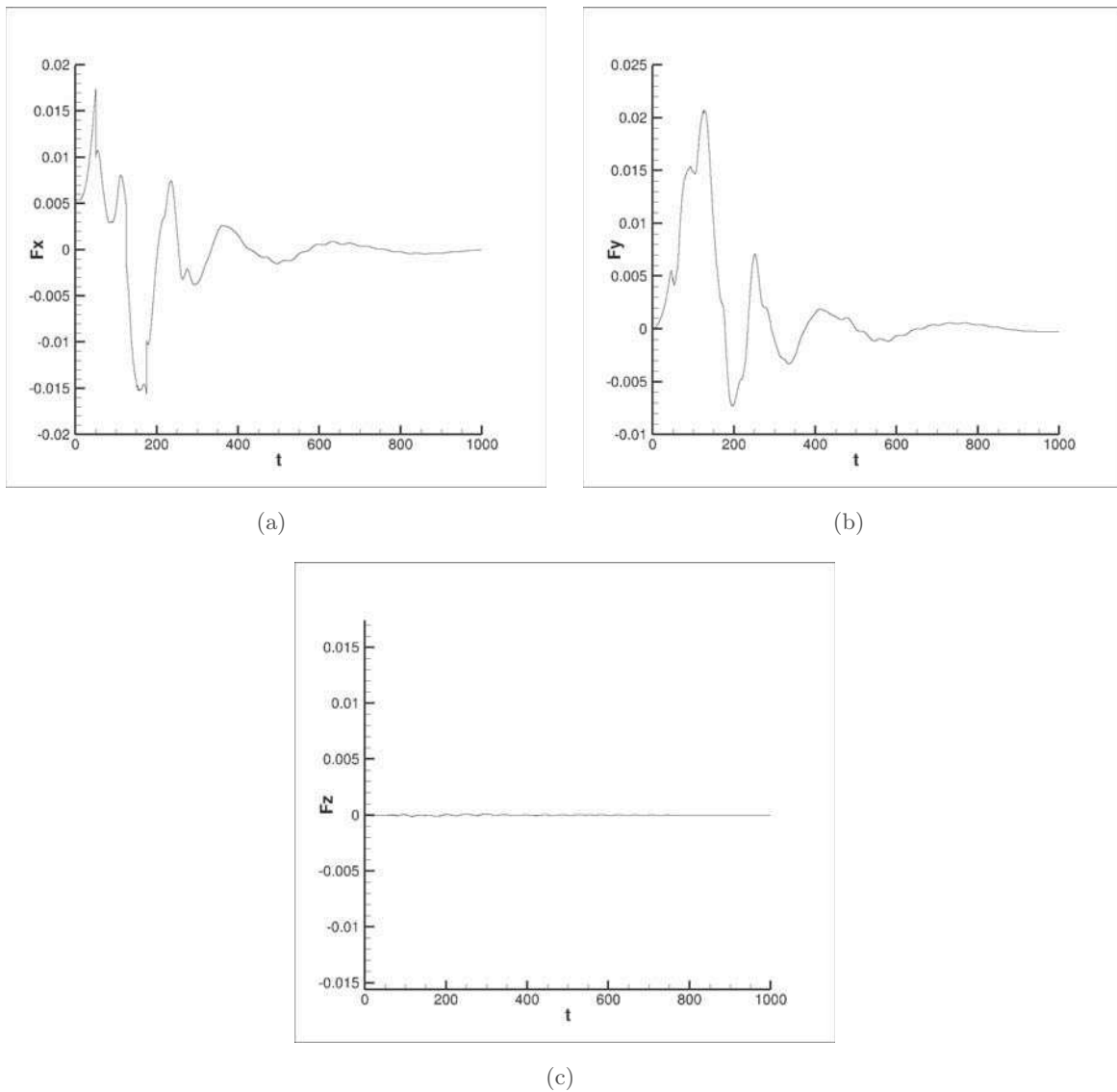


FIGURE 4.29 – Force résultante de la dynamique des fluides sur le réservoir (a) dans la direction \mathbf{e}_x , (b) dans la direction \mathbf{e}_y et (c) dans la direction \mathbf{e}_z .

résultats de ces simulations seront prochainement disponibles car elles sont actuellement en cours sur le centre de calcul régional EOS du CALMIP.

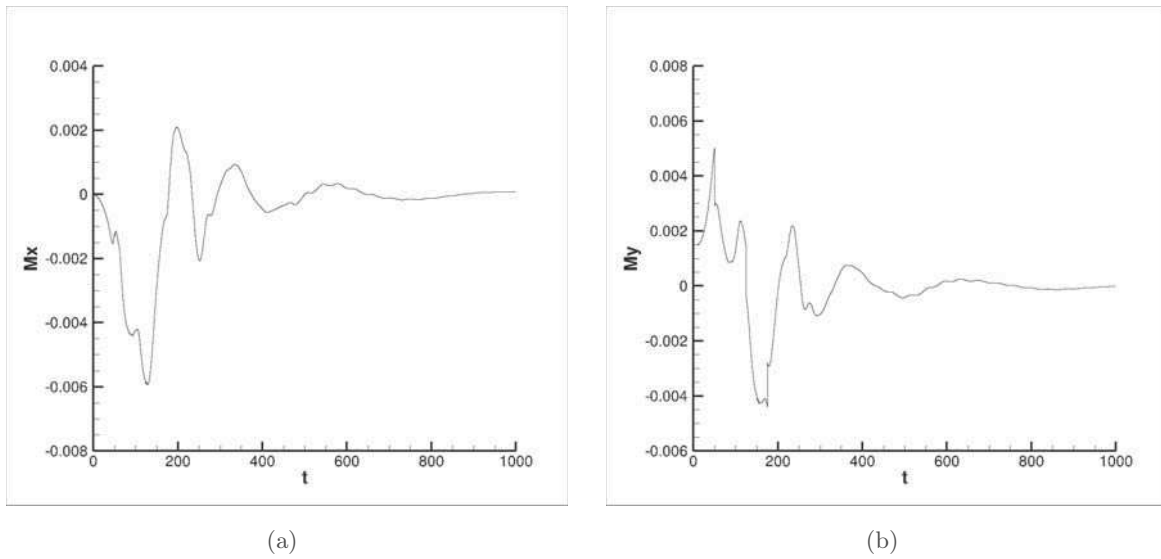


FIGURE 4.30 – Moment résultant de la dynamique des fluides sur le réservoir (a) dans la direction \mathbf{e}_x , (b) dans la direction \mathbf{e}_y et (c) dans la direction \mathbf{e}_z .

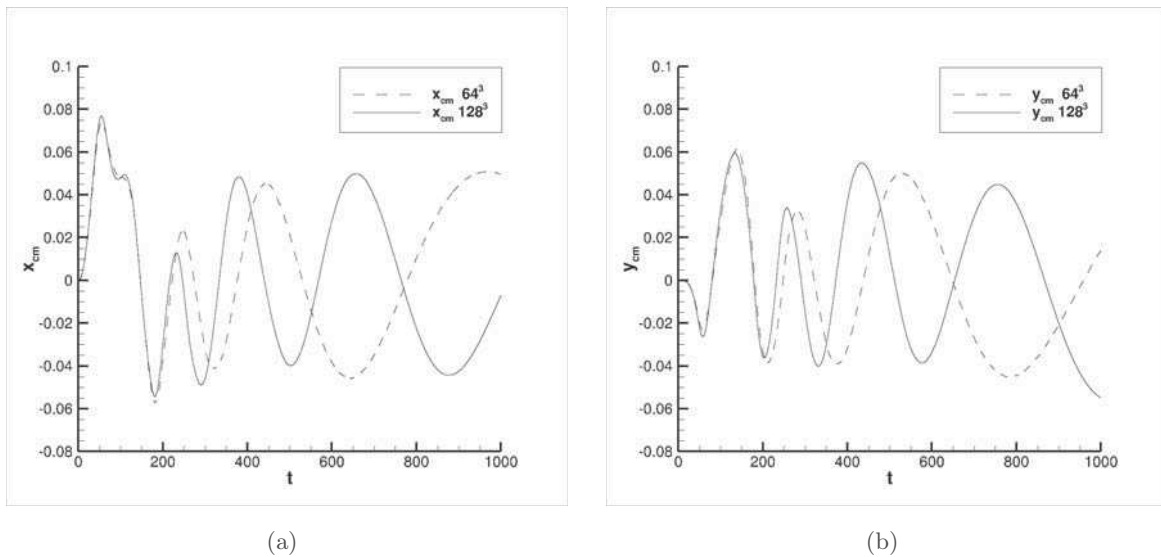


FIGURE 4.31 – Convergence sur les positions des centres de masse (a) dans la direction \mathbf{e}_x et (b) dans la direction \mathbf{e}_y pour des maillages 64^3 et 128^3 .

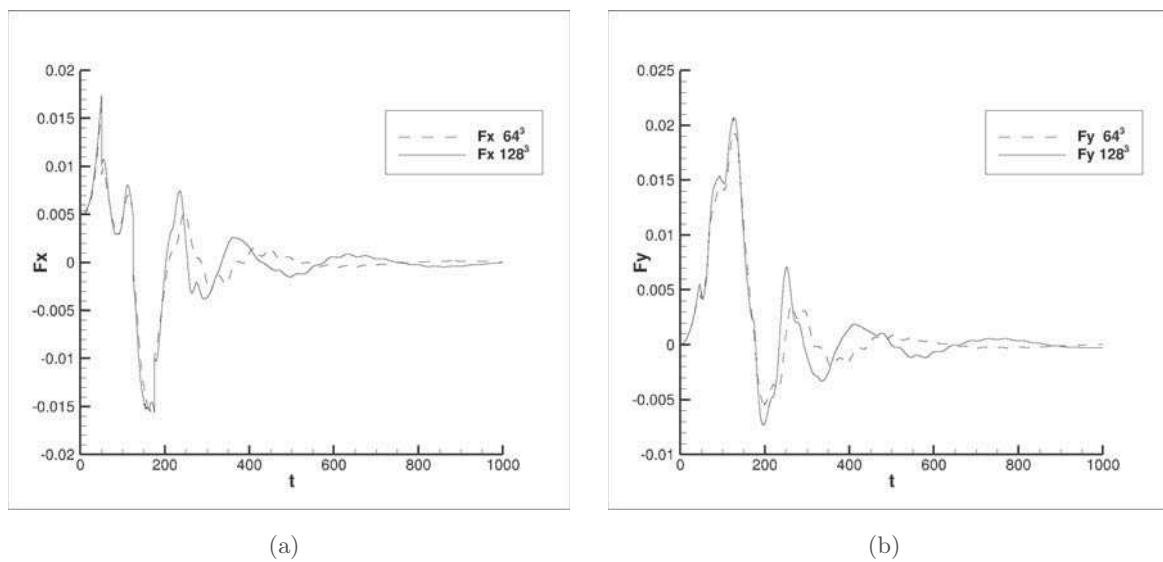


FIGURE 4.32 – Convergence sur les forces résultantes de la dynamique des fluides sur le réservoir (a) dans la direction \mathbf{e}_x et (b) dans la direction \mathbf{e}_y pour des maillages 64^3 et 128^3 .

Conclusions et Perspectives

Ce travail a porté sur la modélisation de ballotements dans les réservoirs de satellite. Il s'inscrit dans le cadre de la *R&T* Ballotements d'ergols en très faible poussée, menée conjointement par le Centre National des Études Spatiales (CNES) et Airbus Defence & Space (ADS), qui a été mise en place afin de définir des modèles analytiques adaptés pour le contrôle d'attitude du satellite lors de phases de manœuvres. Le chapitre 1 a mis en évidence le manque de données et d'outils complets pour appréhender la problématique puisque les modèles analytiques existants ne sont pas suffisamment adaptés pour prédire correctement tous les phénomènes intervenant, et que les données expérimentales sont encore trop peu nombreuses et difficilement accessibles. C'est la raison pour laquelle ce travail numérique est une étape importante.

Durant cette thèse, plusieurs objectifs ont été accomplis, avec notamment l'amélioration du code de calcul DIVA pour obtenir des simulations fiables rapidement. Dans un premier temps l'algorithme BlackBox Multigrid qui a été implémenté et présenté au cours du chapitre 2 présente des performances remarquables, avec des résolutions du système linéaire pour la résolution du champ de pression jusqu'à soixante fois plus rapide que l'algorithme du gradient conjugué préconditionné utilisé jusque là. Ceci va permettre d'accéder à des simulations sur des maillages plus importants tels que le 256^3 pour valider la convergence de nos résultats. La Voronoï Interface Method qui a été présentée dans le chapitre 2 révèle une précision spatiale d'ordre 2 sur le calcul des sauts à l'interface sur des cas analytiques, ce qui n'est pas le cas de la Ghost Fluid Method, jusqu'à maintenant utilisée dans nos simulations. Une implémentation de la méthode VIM dans un code de calcul Navier-Stokes est prévue pour améliorer la précision des solveurs utilisés.

Le chapitre 3 porte sur l'implémentation de la méthode IBM proposée par Ng, Min et Gibou sur les écoulements monophasiques, et son extension à la modélisation d'écoulements diphasiques dans notre code de calcul. Dans cette étude, plusieurs discrétisations des termes visqueux ont été comparées. Une discrétisation spatiale des termes capillaires est également proposée, ainsi qu'une méthode pour décrire la ligne triple lorsque l'interface rencontre le domaine irrégulier. Des cas analytiques de validation ont été présentés et l'intégralité de ces méthodes fait l'objet d'une publication soumise au *Journal of Computational Physics*.

Dans le chapitre 4, nous avons présenté des simulations permettant de comprendre quelques aspects de la phénoménologie liée aux ballotements. En effet les manœuvres classiquement exécutées dans le cadre du maintien à poste du satellite se composent de différentes phases avec notamment la phase de forçage, durant laquelle le satellite est soumis à une vitesse et une accélération de rotation, et la phase de relaxation, où les fluides excités dans le réservoir reviennent à une position d'équilibre.

La caractérisation de la première phase a été l'objet principal du chapitre 4, avec une étude sur les rôles des différents termes liés dans un premier temps à l'accélération de rotation, puis à la vitesse de rotation. Les résultats révèlent que l'accélération de rotation ne joue pas un rôle décisif sur la dynamique de la bulle dans le réservoir dans la gamme de nombre de Bond que nous avons étudiés. En revanche la vitesse de rotation, qui conditionne le nombre de Bond centrifuge, est directement reliée aux efforts exercés par les fluides sur le réservoir lorsque l'équilibre des forces est atteint. Les résultats ont permis de retrouver certaines similitudes et caractéristiques des écoulements, mais d'autres paramètres doivent être étudiés pour compléter les lois proposées dans ce chapitre, avec notamment le rôle du remplissage du réservoir.

La dernière partie du chapitre 4 porte sur la modélisation d'une manœuvre complète de réorientation du satellite en effectuant une rotation sur lui-même de 180° . La complexité de la dynamique résultante de la manœuvre ne permet pas une caractérisation détaillée des phénomènes, mais quelques observations peuvent être effectuées. La phase de forçage de la manœuvre est très similaire aux manœuvres effectuées dans la première partie du chapitre. Durant la phase de relaxation, la bulle est entraînée le long de la paroi du réservoir et en fait plusieurs fois le tour, en ralentissant progressivement.

Toutefois ces études ont besoin d'une validation par l'expérience, et l'expérience FLUIDICS prévue pour début 2016 permettra de valider définitivement les algorithmes implémentés dans le code de calcul pour des configurations en microgravité. Une suite de cette thèse est également prévue et portera notamment sur l'ajout d'une membrane élastique dans le réservoir, entre les deux fluides, et son impact sur les ballotements.

