



Schéma de transport de l'interface d'un écoulement diphasique visqueux non miscible par la méthode des caractéristiques

Mireille El-Haddad

► To cite this version:

Mireille El-Haddad. Schéma de transport de l'interface d'un écoulement diphasique visqueux non miscible par la méthode des caractéristiques. Dynamique des Fluides [physics.flu-dyn]. Université Pierre et Marie Curie - Paris VI; Université Saint-Joseph (Beyrouth). Faculté des Sciences, 2016. Français. NNT : 2016PA066475 . tel-01508500

HAL Id: tel-01508500

<https://theses.hal.science/tel-01508500>

Submitted on 14 Apr 2017

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

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

Schéma de transport de l'interface d'un écoulement diphasique visqueux non miscible par la méthode des caractéristiques.

THÈSE

présentée et soutenue publiquement le 18 Novembre 2016

pour l'obtention du grade du

Docteur de l'Université Pierre et Marie Curie - Paris 6
et
Docteur de l'Université Saint-Joseph - Liban
(Spécialité Mathématiques)

par

Mireille El-Haddad

Composition du jury

Rapporteurs : REMACLE Jean François
MOHAMMADI Bijan

Examineurs : HECHT Frédéric (Directeur de thèse)
SAYAH Toni (Directeur de thèse)
PIRONNEAU Olivier
MANSOUR Gihane
ISRAWI Samer
NATAF Frédéric

Remerciements

J'aimerais tout d'abord exprimer mes profonds remerciements à mon directeur de thèse du côté Libanais, M.Toni Sayah, pour m'avoir fait confiance et donné envie de réaliser une thèse. J'aimerais également lui dire à quel point j'ai apprécié toutes les heures qu'il a consacrées à diriger ce travail, pour ses qualités pédagogiques et scientifiques et pour les nombreux encouragements qu'il m'a prodigués. J'ai beaucoup appris à ses côtés et je lui adresse ma gratitude pour tout cela. Enfin, je suis extrêmement sensible à ses qualités humaines de gentillesse, de patience et de compréhension qu'il a manifestées à mon égard durant mes travaux.

Je voudrais également remercier mon directeur de thèse du côté Français, M. Frédéric Hecht, qui m'a accueilli au sein du Laboratoire Jacques Louis-Lions et qui m'a dirigé tout au long de cette thèse. Il a toujours été disponible pour me guider et répondre à mes questions, et s'est toujours intéressé à l'avancée du projet. Les nombreuses discussions que nous avons eues ainsi que ses conseils sont pour beaucoup dans le résultat final de ce travail. Son esprit scientifique, sa capacité d'analyse et son enthousiasme m'ont montré que le monde de la recherche pouvait être un univers passionnant.

Je suis tellement reconnaissante envers l'ensemble des membres du jury, M.Jean François Remacle et M. Bijan Mohammadi, Rapporteurs, Mme. Gihane Mansour, M. Olivier Pironneau, M. Frédéric Nataf, M. Samer Israwi, Examineurs. J'éprouve un profond respect pour leurs travaux et leur parcours et je suis fier d'avoir pu partager avec eux mon travail.

Je tiens à remercier le programme de financement conjoint du Conseil national libanais de la Recherche Scientifique CNRS-L et l'Université Saint-Joseph USJ (Liban) et le financement fourni par le Laboratoire Jacques-Louis Lions-Université Pierre et Marie-Curie (France) qui ont soutenu ce travail.

Mes remerciements vont aussi à tous mes collègues qu'ils soient au Liban ou à Paris qui m'ont soutenu par leur encouragement. En particulier, un grand merci à Pierre-Henri Tournier et Ryadh Haferssas qui m'ont aidé beaucoup dans la partie technique du calcul parallèle.

Je n'oublie pas de remercier ma famille qui tiennent une place immense dans mon cœur. Papa, tu es une vraie école de la vie, je ne cesse d'apprendre tous les jours avec toi. Maman, une femme aussi adorable que toi je n'en connais pas, tu as toujours été là pour moi, et à aucun moment tu n'as cessé de me couvrir de ta tendresse. Mes Soeurs Christine, Sally et Simona, je pense aux moments que l'on a partagé durant ces trois années, à nos fous rires et nos pleurs... La meilleure amie du monde, Rémie, merci pour être là toujours.....

une petite chose pour toi mais crois moi tu m'as énormément aidé et j'apprécie de tout mon coeur ton soutien.

Enfin, Mes plus grands remerciements vont bien évidemment à mon mari Dani, tu as été toujours là pour moi. Tu m'écoutes quand j'ai des soucis, tu me remontes le moral quand je suis triste, tu m'encourages quand j'hésite et tu me consoles quand je subis un échec. Pour tout cela et pour ces années passionnantes, du fond du coeur : MERCI MON AMOUR!

Table des matières

Introduction	1
1 Motivation	2
1.1 La phase de remplissage	3
1.2 La phase de solidification	4
1.3 Importance de la simulation numérique dans l'étude du remplissage en fonderie.	4
2 Hypothèses physiques de l'étude	6
3 Etat de l'art sur les schémas de transport de l'interface libre	6
3.1 Méthodes Lagrangiennes	7
3.2 Méthodes Eulériennes	8
3.2.1 Les méthodes de suivi d'interface	8
3.2.2 Les méthodes de capture d'interface	10
3.3 Méthodes mixtes	13
3.3.1 Volume-of-Fluid Sub-Mesh (VOF-SM) method	13
3.3.2 Hybrid particle Level-Set method	13
3.4 Conclusion	13
4 Plan du manuscrit	14

Chapitre 1	
Modélisation d'un fluide biphasique	17

1.1 Le problème de remplissage d'un moule	18
---	----

1.2	Etat de l'art sur la résolution numérique du problème de Navier-Stokes . .	21
1.3	Discrétisation du problème du fluide diphasique	23
1.3.1	Formulation variationnelle	23
1.3.2	Discrétisation	24
1.3.2.1	Discrétisation en temps : Méthode des caractéristiques. . .	25
1.3.2.2	Discrétisation en espace : Méthode des éléments finis . . .	27
1.3.2.3	Formulation variationnelle discrète en temps et en espace .	27
1.3.2.4	Le problème algébrique	28
1.3.3	Résultats numériques	29
1.4	Solution régularisée et réinitialisation	34
1.4.1	Solution régularisée	34
1.4.2	Algorithme de réinitialisation	35
1.4.2.1	Equation de réinitialisation	36
1.4.2.2	Discrétisation de l'équation de réinitialisation	37
1.4.3	Résultats numériques	38
1.5	Conditions aux bords de Navier	41
1.5.1	Etat de l'art sur les conditions aux bords de Navier	41
1.5.2	Remplissage d'un réservoir avec les conditions aux bords de Navier	43
1.5.3	Résultats numériques	45
1.6	Conditions aux bords de Neumann sous effet gravitationnel	47
1.6.1	Résultats numériques	50
1.7	Conservation de masse	52
1.7.1	Résultats numériques	53

Chapitre 2	
Méthode de projection	56

2.1	Motivation	57
2.2	Méthode de projection avec les conditions aux bords de Dirichlet	57
2.2.1	Présentation de la méthode	57
2.2.2	Formulation variationnelle et discrétisation	59
2.3	Méthode de projection avec les conditions aux bords de Navier	60
2.4	Résultats numériques	61
2.4.1	Dimension deux	61
2.4.2	Problème trois dimensionnel : exemple du cube	64
2.4.3	Cas industriel (3D)	68

Chapitre 3	
Validations pratiques	72

3.1	Validations numériques : Simulation eau-air	73
3.1.1	Expérience de Martin et Moyce en 2D	74
3.1.2	Expérience de Martin et Moyce en 3D	77
3.2	Validations numériques : Simulation métal liquide-air (Campbell benchmark)	79

Chapitre 4	
Calcul Parallèle Itérative	84

4.1	Motivation	85
4.2	Méthodes de décomposition de domaine	86
4.2.1	Etat de l'art sur les algorithmes itératives de Schwarz	86
4.2.2	Présentation des algorithmes de Schwarz	87
4.2.3	Connection avec l'algorithme de Block Jacobi	90
4.2.4	Les problèmes Multi-dimensionnels avec plusieurs sous-domaines	92

4.2.5	Algorithme de décomposition de domaine utilisé avec le logiciel Freefem++	93
4.3	Méthode de décomposition de domaine avec la méthode de caractéristiques.	94
4.4	Validation des résultats numériques : Comparaison des résultats séquentiels et parallèles	96
4.4.1	Premier cas industriel	96
4.5	Deuxième cas industriel	100
Chapitre 5		
Conclusions et perspectives		108
5.1	Conclusions	109
5.2	Perspectives	109
Bibliographie		110
Bibliographie		111

Introduction

1 Motivation

La fonderie produit des pièces métalliques finies ou semi-finies à l'ensemble de l'industrie. La fabrication des pièces se fait en fondant un métal ou un alliage liquide, puis en les versant dans un moule et à la fin en laissant le métal liquide se refroidir jusqu'à solidification. Les industries se recourent à la fonderie soit pour produire des pièces difficilement usinables qu'il serait dur ou impossible de les fabriquer par d'autres méthodes, soit pour la réalisation à des prix moins chers de pièces simples, soit encore pour profiter de l'avantage des propriétés physiques ou d'utilisation (dilatation, corrosion, frottement, usure, basses et hautes températures) et des propriétés mécaniques des divers métaux et alliages coulés, soit enfin pour couler des alliages non corroyables.



FIGURE 1 – Fusion des métaux dans le four et versement du métal liquide dans le moule.

La fonderie a pu développer continuellement ses productions à cause des avancements à tous les niveaux techniques, tant en précision qu'en crédibilité, lui autorisant de concurrencer, et parfois lui permettant de remplacer, dans des usages très sévères ou de sécurité, des pièces habituellement fabriquées par d'autres techniques (forgeage par exemple) comme des vilebrequins de moteurs ou des bras de suspension pour les véhicules automobiles. Tous ces progrès et avantages motivent la très grande propagation des pièces de fonderie dans tous les domaines de l'industrie, et les capacités de développement de cette technique permettent de prévoir de plus larges applications dans le futur.

En générale, la fabrication des pièces dans les fonderies peut être réalisée en plusieurs étapes. Tout d'abord, on commence par la conception des plans à produire et la fabrication des moules. Ensuite, on prépare et on fusionne le métal ou l'alliage dans des fours. Après, on verse le métal liquide dans les moules et on refroidit le liquide jusqu'à solidification. Enfin, on démoule la pièce solide, on la nettoie, on la réusine pour retirer la partie oxydée et on fait l'inspection finale de la qualité du produit.

La qualité des pièces obtenues dépend des étapes suivies qui comporte le remplissage

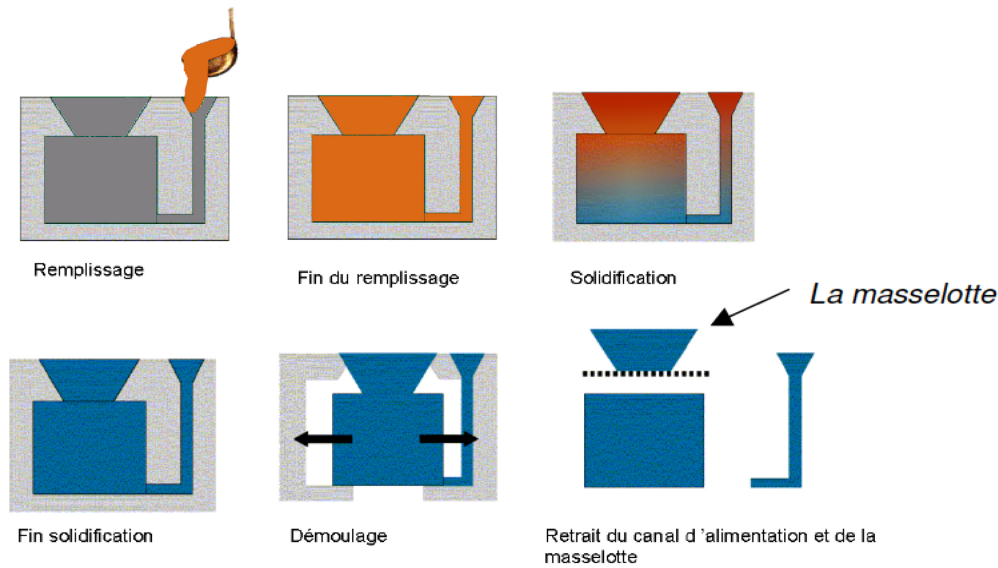


FIGURE 2 – Description schématisée d’un remplissage en source sous gravité, suivie de la phase de solidification .

de la moule et la solidification du métal liquide. Dans cette partie, on décrit ces deux phases et on signale les défauts causés lorsqu’elles ne sont pas bien maîtrisées. Les moyens de contrôle dont les fondeurs appliquent sur la coulée, sont le contrôle de la vitesse de coulée, le contrôle des propriétés thermiques des moules (par l’application de peintures appelées poteyage à l’intérieur des moules, ou encore l’ajout de poudres exothermiques), et enfin pour la coulée par gravité, ils est nécessaire de se servir d’un système de masselotage dimensionné. Les masselottes sont des parties de la pièce qui seront retirées et qui permettent de maintenir l’excès du métal chaud jusqu’au refroidissement et le retrait thermique progressif du métal coulé.

1.1 La phase de remplissage

Pendant le remplissage des moules, le métal liquide fondu a une température d’environ 700°C ; tandis que le moule et l’air ont la température ambiante. Un échange thermique se fait entre eux et le métal liquide se refroidit et commence la solidification près des parois de moule et se termine normalement dans les masselottes. Il faut bien noter que la vitesse de remplissage doit dépasser celle de solidification en particulier dans les portions de pièces de faible épaisseur. Le remplissage peut s’arrêter complètement ou partiellement dans des pièces de petites épaisseurs à cause de la solidification du métal dans ces zones ce qui rend la pièce fondue inutilisable.

A ce niveau, le transport de l’interface peut causer la formation des inclusions de ma-

tériaux ou d'air dans le moule. De plus, les changements topologiques comme le replis de cette surface libre induit à l'oxydation du métal. Ces phénomènes peuvent causer des hétérogénéités et des défauts qui abiment la qualité de la pièce. D'autre part, l'échange thermique lors du remplissage déterminent la température du métal à la fin du remplissage, ce qui conduit à un point de départ pour l'étape de solidification.

1.2 La phase de solidification

Il est assez essentiel de bien modéliser durant le procédé de la fonderie les contractions thermomécaniques des métaux pour bien prédire la forme finale de la pièce après refroidissement. Des masselottes sont ajoutées pour contrôler les retraits dues à la solidification du métal. De plus, l'étude de la solidification permet de détecter des défauts tels que des craquelures (craques) qui apparaissent dans des zones fragiles et qui sont difficiles à percevoir dans le métal au cours du procédé.

1.3 Importance de la simulation numérique dans l'étude du remplissage en fonderie.

Un des grands enjeux pour la fonderie est de fabriquer des pièces de plus en plus complexes et bonnes en effectuant le minimum possible d'essais et par suite en économisant au maximum possible d'énergies et de temps. Le bon déroulement du remplissage et de la solidification exige beaucoup de métier à cause de la complexité des phénomènes mécaniques et thermomécaniques mis en jeu. Les méthodes qui s'appuient sur l'expérience et qui ne sont pas fondées sur une étude scientifique nécessitent de nombreux essais avant d'obtenir un résultat satisfaisant, et par conséquent causent des coûts.

La simulation numérique des procédés de fonderie devient un outil incontournable, de nos jours. Elle nous permet de comprendre ces procédés, obtenir des produits conformes avec le nombre d'essais minimal. Elle permet de diminuer les coûts, améliorer la qualité des pièces obtenues, comprendre les défauts éventuels. Elle est aujourd'hui un élément essentiel pour la compétitivité et le développement de l'industrie de la fonderie.

Dans ce travail, on désire de modéliser des pièces uniques de grandes tailles de géométries compliquées de moule en sable. La fabrication de ce genre des pièces nécessite parfois une semaine de travail et les moules sont détruites lors du démoulage. Refaire une pièce exige des coûts significatifs. La simulation nous permet de prédire le temps pour arrêter le procédés de remplissage. De plus, on vise à avoir une description assez précise de l'interface pour étudier dans une autre étape le transport sur cette surface d'un phénomène d'oxydation.



FIGURE 3 – Produit d'une pièce semi-finie.



FIGURE 4 – Produit dans l'industrie aéronautique.

2 Hypothèses physiques de l'étude

Pour formuler le modèle mathématique approprié à notre problème, on commence par introduire les hypothèses physiques. On considère un écoulement laminaire et instationnaire des deux fluides non miscibles l'air et le métal liquide. Dans cet écoulement, les deux fluides sont considérés visqueux, newtoniens (le tenseur des contraintes dépend linéairement du taux de déformation) et de rapport de densité assez important. En outre, les fluides sont considérés comme incompressibles et isothermes, négligeant ainsi les variations de la densité et de la viscosité due aux changements dans la pression ou la température (on suppose que le remplissage des pièces étudiées est suffisamment rapide pour que les variations de température soient négligeables du point de vue couplage thermomécanique. En d'autre terme, la solidification ne commence pas durant le remplissage de moule). De plus, on suppose que les deux fluides sont homogènes et par suite la viscosité et la densité sont des constantes pour chaque fluide.

On suppose aussi que les deux fluides sont non miscibles et la zone de séparation entre les fluides est une interface d'épaisseur nulle dans laquelle les propriétés physiques des deux fluides changent brusquement. Pour traiter les variations des propriétés physiques à travers l'interface, il est nécessaire de mettre en oeuvre des conditions de saut. Dans notre étude, on néglige la tension de surface entre les deux fluides et on suppose que l'interface est imperméable, donc le transfert de masse à travers l'interface est négligé (voir [9] et [94]). Enfin, on note que le seul moteur de ce procédé de remplissage est la force de gravité.

3 Etat de l'art sur les schémas de transport de l'interface libre

L'étude de la phase de remplissage en fonderie nécessite d'être capable de simuler des écoulements trois dimensionnels, instationnaires, et à surface libre animée des changements géométriques et topologiques. Le développement d'un algorithme de calcul fiable pour le transport de l'interface libre dans un écoulement diphasique nécessite la discrétisation et le suivi méticuleux de la surface libre.

Il existe dans la littérature plusieurs approches pour l'étude du transport de l'interface libre. Le choix de la méthode numérique pour la construction du schéma de transport de l'interface est délicat. Les méthodes présentées jusqu'à notre jour présentent des inconvénients qui nécessitent d'être contournés. De nombreux compromis sont conçus pour la résolution des problèmes d'écoulement diphasique, ceci dépend des objectifs de la simulation à effectuer.

La suite de cette section est consacrée à une étude bibliographique dans laquelle on donnera un aperçu des différentes méthodes numériques existantes dans la littérature pour

le transport de l'interface.

Depuis les travaux fondateurs de Harlow et Welch [72], plusieurs méthodes ont été développées pour résoudre le problème du transport de l'interface, on cite les études [6],[35],[136] et les références qui y sont inclus comme exemples. La façon la plus populaire est de diviser ces méthodes en deux grandes catégories selon le maillage :

- a) Les méthodes Lagrangiennes qui sont caractérisées par un système de coordonnées se déplaçant avec le fluide.
- b) Les méthodes Euleriennes qui sont caractérisés par un maillage fixe et le fluide se déplace d'une cellule à l'autre.

Dans les sections suivantes, on présente en détails Les méthodes Lagrangiennes et Euleriennes les plus connues pour motiver notre choix des méthodes Euleriennes qui présentent des algorithmes robustes dans beaucoup d'applications.

3.1 Méthodes Lagrangiennes

Les méthodes Lagrangiennes semblent être le mieux adapté pour les problèmes traitant des surfaces libres. Ces méthodes ont les avantages suivantes : elles permettent le suivi précis des interfaces, l'application des conditions aux bords de l'interface faciles et l'absence du terme de convection non linéaire. Le principal problème avec les méthodes de suivi de l'interface de Lagrange est l'imprécision numérique due aux maillages irréguliers obtenues (voir la figure 1, à gauche). Ainsi, dans leur forme originale ces méthodes ne conviennent que pour la simulation des petites déformations interfaciales et elles ne permettent pas de prendre en compte les changements topologiques. Pour une description de l'écoulement purement Lagrangienne avec l'approche de volume finie, on cite l'article de Hirt et al. [79]. Pour plus des détails sur ces méthodes avec l'approche des éléments finis, on se réfère à Kawahara et al. ([104], [77]).

Pour se débarrasser des graves distorsions des maillages dans le cadre Lagrangien, la méthode a été améliorée en ajoutant un algorithme de remaillage et de modification de zone. Cet algorithme repose sur l'introduction d'un nouveau maillage et d'un transfert ultérieur de l'information entre l'ancien maillage et le nouveau. Cette méthode d'interpolation entre les maillages peut être tout à fait arbitraire et par suite les maillages obtenus peuvent être arbitraires autant que le nombre de cellules, leur géométrie et topologie sont concernées. Cette méthode de remaillage est appelée Free Lagrangian methods (figure 1, à droite), elle a été utilisée en combinaison avec les méthodes de suivi d'interface (Voir [34], [51], [52]). Cette méthode a deux inconvénients importants : elle ne permet pas de gérer les changements topologiques de l'interface (sauf si des algorithmes très sophistiqués sont ajoutés) et elle exige d'effectuer des remaillages fréquentes, ce qui la rend chère et peu fiable, en particulier pour les problèmes trois dimensionnels.

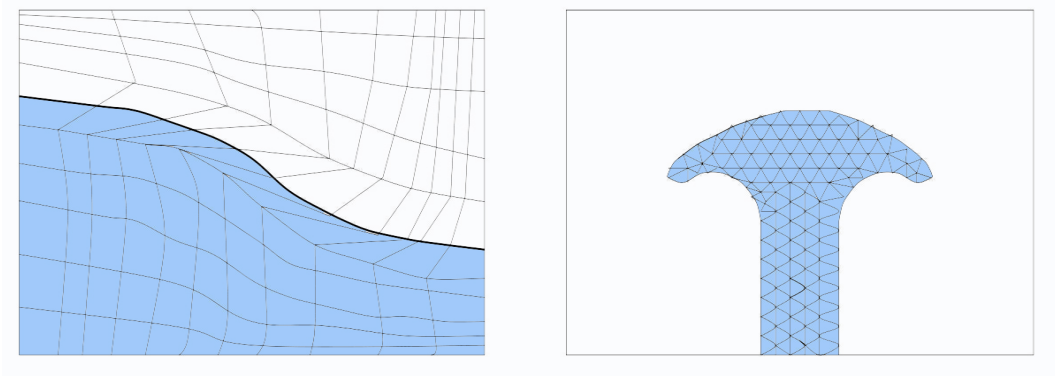


FIGURE 5 – A gauche : Suivi d'interface Lagrangienne ; A droite : Free Lagrangian method

3.2 Méthodes Eulériennes

Dans les méthodes Eulériennes, les équations qui décrivent le mouvement du fluide et de l'interface sont résolues sur un maillage fixe. Les méthodes de suivi d'interface repèrent précisément la position de l'interface et en déduisent la répartition spatiale des phases, alors que les méthodes de capture d'interface repèrent la position des phases et en déduisent celle de l'interface. Le repérage de l'interface est maintenant implicite contrairement à la classe des méthodes de suivi d'interface. Dans les méthodes de capture d'interface, la localisation de l'interface est déterminée par l'advection par la vitesse du fluide soit d'une fonction caractéristique ou d'un isocontour-zéro d'une fonction continue. L'équation de transport obtenue décrit l'interface comme étant une ligne matérielle qui se propage avec le fluide. L'avantage immédiat de ces méthodes est la gestion naturelle des phénomènes de rupture et de coalescence.

3.2.1 Les méthodes de suivi d'interface

Les méthodes de suivi d'interface sont basées sur l'utilisation d'un ensemble de points appelés marqueurs qui marque le lieu de l'interface (Voir figure (6)). Chaque cellule de calcul contient toujours les mêmes éléments fluides et le maillage est mobile avec le fluide. Ainsi, les méthodes Lagrangiennes sont caractérisées par un système de coordonnées qui se déplace avec le fluide. Comme le temps évolue, les points marqueurs doivent être relocalisés le long de la nouvelle interface. En outre, cette méthode exige le transfert des informations entre l'interface et le maillage fixe une fois l'interface est déplacé. De plus, dans cette approche, il est difficile de traiter l'évolution des marqueurs d'interface lorsque l'interface devient fortement retiré ou déformé [43], [26], [27] et [16].

Dans la suite, on présente les méthodes de suivi d'interface les plus connues et on montre les avantages et les inconvénients de chaque méthode.

Surface tracking method

La méthode Surface tracking a été introduite par Hyman [76], elle a été développée plus tard (Voir [58], [59] et [45], [46]). Cette méthode représente l'interface comme une série de courbes interpolées à travers un ensemble discret de points sur l'interface. A chaque pas de temps, les informations sur la localisation des points et la séquence dans laquelle elles sont reliées sont sauvgardées. Les points sont ensuite déplacés selon une évolution de l'équation de l'interface. Les informations concernant la position ainsi que l'orientation et la courbure de l'interface sont explicitement disponibles pendant le processus de calcul.

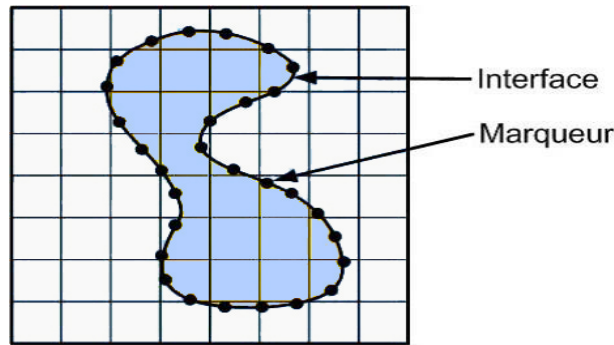


FIGURE 6 – Front tracking method

L'avantage de cette méthode est qu'elle repère précisément et à chaque instant la position de l'interface. tandis qu'au cours de la simulation, des difficultés apparaissent avec la gestion des changements de la topologie de l'interface comme la fusion et le pliage de l'interface, ce qui nécessite de renuméroter et repositionner les marqueurs de l'interface continuellement et exige des algorithmes très compliqués, chers et dur à valider surtout pour les problèmes trois dimensionnelles.

Volume tracking method

Le premier algorithme des méthodes Volume tracking Euleriennes pour les écoulements à surface libre a été introduit par Harlow et Welch [72]. L'approche a été développée plus tard par de nombreux chercheurs (on cite [74], [103] et [120]). Cette approche utilise un maillage uniforme fixe. La surface libre est donnée par les cellules qui contiennent les deux fluides, c'est à dire les marqueurs, et qui sont adjacentes à une cellule vide.

Les principaux avantages de ces méthodes sont les suivants :

- a) Elles peuvent traiter plus qu'un seul fluide.
- b) Elles peuvent traiter les interfaces soumis à de grandes déformations.

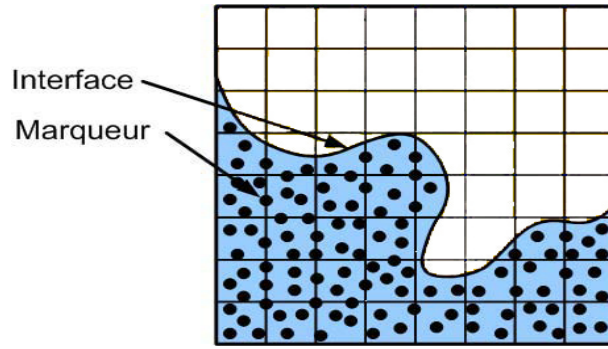


FIGURE 7 – MAC

c) On peut simuler l'interaction de l'interface.

Les inconvénients sont les suivants :

- a) Elles n'ont aucune donnée concernant la localisation exacte, l'orientation et la courbure de l'interface.
- b) Les particules peuvent s'accumuler dans certaines parties de la grille en laissant d'autres parties pas bien résolues ce qui nécessite la réorganisation des points de l'interface
- c) La méthode de calcul est chère car elle exige effectivement un système de grille double (Euler et marqueurs),
- d) Il est difficile d'imposer des conditions aux limites sur l'interface.

3.2.2 Les méthodes de capture d'interface

Volume of Fluid (VOF) method

Le premier algorithme de cette méthode a été proposé par Hirt et Nichols [75]. Cette méthode a été développée pour la simulation d'écoulements diphasiques. Le principe de la méthode VOF est de détecter les fluides par une fonction unité C qui varie entre 0 et 1, et qui constitue le volume occupé par le fluide dans une maille. Ainsi, une cellule de calcul occupée par l'un des fluides possède la valeur 1, tandis que celle qui est occupée par l'autre fluide possède la valeur 0. L'interface se situe dans les mailles où la fonction VOF partagée entre les valeurs 0 et 1 (voir figure 8). La fonction couleur C est advectée par la vitesse du fluide qui est décrit par l'équation suivante :

$$(1) \quad \frac{\partial C}{\partial t} + (u \cdot \nabla) C = 0.$$

Un algorithme de type VOF se fait en deux étapes : une étape de propagation et une autre de reconstruction. La première étape devrait être effectuée avec un grand soin, puisque

l'advection d'une fonction discontinue nécessite un traitement numérique particulier. De même l'étape de la reconstruction exige une attention particulière, puisque la position, l'orientation et la courbure de l'interface affecte directement la précision de l'algorithme. Dans l'algorithme VOF initial, la méthode Simple Line Interface Calculation (SLIC) a été utilisée (Voir [103]). Plus tard, la méthode Piecewise Linear Interface Construction (PLIC) a été développée pour une description et une localisation de l'interface plus précise (Voir [7], [115], [122] et [123]).

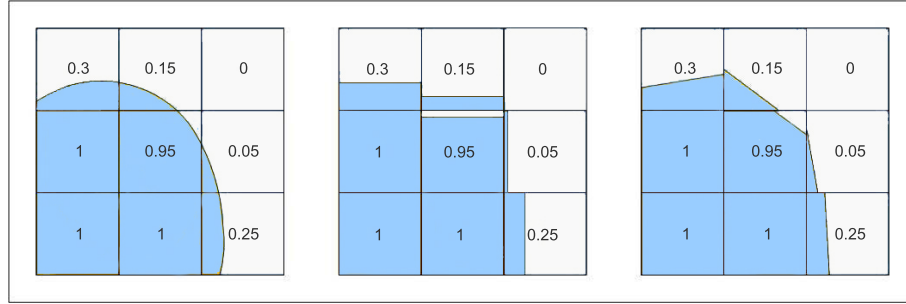


FIGURE 8 – Principe de la méthode VOF ; (à gauche) Interface réelle et fonction couleur associée ; (au milieu) Classe de méthodes SLIC pour reconstruire l'interface ; (à droite) Classe de méthodes PLIC pour reconstruire l'interface.

Les méthodes de capture d'interface VOF sont des outils numériques couramment utilisés dans la simulation des problèmes hydrodynamiques à surface libre pour les raisons principales suivantes :

- a) Traitement naturelle des changements topologiques.
- b) Préservation de la masse.
- c) Extension facile pour des problèmes de trois dimensions.

Les principales désavantages de ces méthodes sont les suivants :

- a) Advection d'une fonction discontinue.
- b) Difficultés à déterminer la position exacte de l'interface, ainsi que la normale et la courbure à l'interface,
- c) Diffusion numérique des details de l'interface et des conditions aux bords interfaciales.
- d) Les algorithmes de reconstruction qui sont efficaces, sont complexes et coûteux à mettre en place en 3D.

Level-Set method

Le premier algorithme de capture d'interface qui se base sur le transport d'un champ scalaire continu a été proposé par Dervieux et Thomasset [38]. Plus tard, cette approche

a été développé en utilisant la fonction distance signée. La méthode Level-Set a été introduite par Osher et Sethian [107] en 1988 et a été étudiée dans [137], [26], [152], [153] et [138]. Dans cette méthode, l'interface est constituée par la courbe de niveau zéro d'une fonction distance signée régulière lisse ϕ qui a pour expression :

$$(2) \quad |\phi(x, t) = |x - x_\Gamma|$$

où x_Γ correspond au point de l'interface le plus proche du point x . Avec cette définition de la fonction distance, l'interface est décrite par le lieu des points x tels que $\phi(x, t) = 0$. Par ailleurs, la fonction Level Set est caractérisée par $\phi(x, t) > 0$ d'un côté de l'interface et $\phi(x, t) < 0$ de l'autre côté.

L'advection de la fonction Level-Set par la vitesse du fluide permet de prédire le mouvement de l'interface. Ceci se fait en résolvant l'équation de transport suivante :

$$(3) \quad \frac{\partial \phi}{\partial t} + (u \cdot \nabla) \phi = 0.$$

Les avantages de cette méthode sont qu'elle permet de prendre en compte les changements topologiques naturellement, simplifie remarquablement le problème de convection, permet de calculer facilement les caractéristiques géométriques à partir de la fonction distance, et est facile à étendre aux problèmes en 3D. Un certain nombre d'inconvénients peut toutefois affecter la fiabilité de la méthode. Pendant le processus de calcul et après plusieurs itérations, la propriété de la distance algébrique ne reste pas conserver. Un algorithme de réinitialisation est habituellement utilisé pour corriger ce défaut. De plus, cette méthode a l'inconvénient qu'elle ne conserve pas la masse, ce qui signifie des gains ou des pertes de masse non-physiques peuvent se produire à chaque pas de temps. Comme le temps évolue, même les petites erreurs négligeables peuvent être accumulées et aboutissent à des erreurs inacceptables par la suite. Pour la méthode Level-Set classique, ce type de gain/perte de masse provient principalement de deux aspects : D'une part, la discrétisation de l'équation de transport de la fonction Level-Set peut conduire à une dissipation numérique et causer une perte/gain de masse. D'autre part, le processus de réinitialisation mentionné avant (pour garder la propriété d'une distance signée à la fonction Level-Set) entraîne un mouvement considérable de la ligne de niveau zéro, ce qui conduit à une diffusion numérique supplémentaire et par suite le gain/perte de la masse.

Il y a eu de nombreuses études pour améliorer la conservation de la masse de la méthode Level-Set. Une des approches consiste à résoudre l'équation de transport de la fonction Level-Set par des schémas d'ordre élevés accompagnée par une technique d'adaptation de maillage [142]. Une autre approche consiste à coupler la méthode Level-Set avec d'autres méthodes, telles que Particle Level-Set method [73] ou Coupled Level-Set and Volume of Fluid (CLSVOF) [15] .

Bien que les méthodes couplées présentent des avantages dans certains aspects sur la méthode originale, le couplage nous introduit d'autres problèmes. Leur coût est beaucoup plus élevé que celui de la méthode simple. Par exemple, dans l'approche Particle Level-Set method, plusieurs particules sont requises par cellule pour une solution exacte. De même,

l'efficacité de la méthode CLSVOF est limitée par des restrictions sur la taille de pas de temps pour le transport géométrique du scalaire VOF. Ainsi, dans ces approches, la simplicité de la méthode Level-Set est perdue partiellement.

3.3 Méthodes mixtes

Les méthodes mixtes sont certainement les plus précises pour la modélisation d'une interface dans un écoulement diphasique, puisqu'elles profitent de la finesse de résolution des méthodes Lagrangiennes avec la robustesse des méthodes Eulériennes. Il y a de nombreux schémas dans la littérature. Dans la suite, on présente quelques méthodes et on discute leurs avantages et leurs inconvénients.

3.3.1 Volume-of-Fluid Sub-Mesh (VOF-SM) method

La méthode Volume-of-Fluid Sub-Mesh (VOF-SM) a été proposée par Vincent et al. [149]. Cette méthode peut modéliser des géométries très complexes avec une précision élevée lors des fortes changements topologiques de l'interface en utilisant les techniques Marker And Cell [118], Particle In Cell [32] et la méthode Moving Particle Semi-implicit [98]. Vincent et al. [149] ont montré que la description de l'interface avec la méthode VOF-SM est une à deux fois meilleure que les autres méthodes VOF classiques. L'avantage de cette méthode est qu'elle décrit l'écoulement multiphasique à plus petite échelle que les méthodes basées sur l'utilisation d'une approche totalement Eulérienne ce qui réduit la diffusion numérique. Cependant, cette méthode peut exiger un coût important en mémoire et en temps de calcul pour les problèmes trois dimensionnelles.

3.3.2 Hybrid particle Level-Set method

La méthode "Hybrid particle Level-Set" a été proposée par Enright et al. [44]. Elle consiste à mettre des marqueurs près de la ligne de niveau zéro et de les transporter par la vitesse du fluide. Si les marqueurs qui se trouvaient au départ dans l'un des côtés traversent l'interface à l'autre côté, la position de l'interface n'est pas précise. Une correction de l'algorithme se fait par la reconstruction locale de l'interface en bénéficiant des marqueurs. Ceci réduit l'erreur et les diffusions numériques provenant de la méthode Level-Set a une précision d'échelle inférieure à la taille de la maille et donc assure la conservation de masse. Malheureusement, la méthode "Particle Level-Set" est difficile à implémenter (surtout en 3D) puisqu'il devient difficile d'optimiser la distribution des marqueurs.

3.4 Conclusion

En se basant sur cette bibliographie, on peut tirer comme conclusions que les méthodes Lagrangiennes représentent sans doute l'interface d'une manière très précise ; tandis qu'elles

sont moins efficaces pour les simulations avec des grandes déformations ce qui est notre cas dans le remplissage d'un moule. En outre, les algorithmes d'optimisation de la distribution des marqueurs rajoutent des contraintes supplémentaires qui sont la durée de simulation et la mémoire requise pour les calculs ce qui nous empêche la simulation des moules avec des dizaines des millions de degrés de liberté. De plus, le calcul des propriétés géométriques comme la normale et la courbure exige un traitement spécial. Enfin, l'implémentation numérique des algorithmes Lagrangiens, déjà complexes en 2D, devient plus difficiles en 3D. Toutes ces raisons nous ont motivé à utiliser les méthodes Eulériennes qui sont mieux adaptées au suivi de l'interface dans le cas des déformations fortes en trois dimensions. Pour ce type de méthodes, on a vu l'approche discontinue (VOF) et l'approche continue (Level-Set). Suite aux désavantages de la méthode VOF déjà cités, nous avons opté à l'utilisation de la méthode Level-Set dans notre travail de recherche. On analysera en détail les avantages et inconvénients de cette méthode qui ont motivé notre choix.

4 Plan du manuscrit

Dans le chapitre 1, nous nous intéressons à la construction d'un algorithme deux dimensionnel pour la modélisation du remplissage d'un moule. En premier lieu, nous couplons les équations de Navier-Stokes pour la description de l'écoulement des fluides avec l'équation de transport en advection la fonction Level-Set pour la capture de l'interface. La densité et la viscosité des fluides seront mis à jour en fonction de la Level-Set en évoluant en temps. En second lieu, pour la résolution numérique de ces équations, nous découplons les parties non-convectives du reste des équations de Navier-Stokes et de l'équation de transport et nous utilisons la méthode des caractéristiques d'ordre "1" pour la discrétisation en temps et la méthode des éléments finis pour la discrétisation en espace avec la paire P1b-P1 pour la vitesse et la pression qui satisfait la condition inf-sup et l'élément finis P1 pour la Level-Set. Après avoir établi la simulation numérique, nous avons remarqué un problème au niveau de l'interface près du bords, l'interface ne peut pas toucher le bord d'une manière physique et il reste dans la dernière maille du domaine près du bord. Ceci nous a obligé à utiliser une solution régularisée qui corrige le phénomène physique près du bord d'une part, mais qui introduit une diffusion numérique supplémentaire d'autre part. De plus, il est connu que la fonction Level-Set ne conserve pas la propriété de distance algébrique, ceci cause la rupture de l'interface et augmente l'erreur en volume. A chaque pas de temps, nous avons utilisé un algorithme de reinitialisation jusqu'à atteindre un état d'équilibre. Cet algorithme corrige la fonction distance en moins de six itérations. En troisième lieu, nous avons remarqué que les conditions aux bords de Dirichlet introduisent un phénomène de couche limite près du bord qui n'est pas réaliste. Nous avons remplacé la condition au bord collante par les conditions aux bords de Navier qui permettent aux fluides de glisser d'une manière plus naturelle près du bord et qui diminuent l'erreur en volume causée par la diffusion numérique. En quatrième lieu, nous avons introduit une nouvelle condition aux bords pour l'équation de transport qui représente l'angle que fait

l'interface avec le bord d'un point de vue physique sous l'effet gravitationnelle et qui réduit considérablement l'erreur en volume. Enfin, pour préserver les propriétés physiques des fluides, nous ajoutons un algorithme de conservation de masse.

Dans le chapitre 2, afin de réduire le temps de calcul des équations de Navier-Stokes, nous avons utilisé la méthode de projection de Chorin/Temam pour la résolution des équations de Navier-Stokes. Ceci nous permet d'étendre l'algorithme aux problèmes trois dimensionnels industriels. L'algorithme de la méthode de projection se décompose en trois étapes : à chaque pas de temps, la première étape consiste à calculer une vitesse intermédiaire qui ne satisfait pas la contrainte d'incompressibilité ; dans la deuxième étape, on utilise la vitesse intermédiaire déjà calculée qu'on projète dans l'ensemble des champs à divergence nulle pour obtenir la valeur de la pression solution du problème ; la troisième étape consiste à construire la vitesse finale à partir des résultats obtenues. L'algorithme qui se base sur la méthode classique et celui sur la méthode de projection donnent des résultats très similaires ; alors que le second réduit considérablement le temps de calcul et le coût de mémoire. Dans les résultats numériques, nous avons étendu l'algorithme avec la méthode de projection pour résoudre le remplissage des moules industrielles de moyenne taille.

Le chapitre 3, est consacré aux validations de l'algorithme que nous avons construit dans les deux premiers chapitres. En premier lieu, nous validons la capacité de l'algorithme proposé à modéliser des fluides comme l'eau et l'air de rapport de densité important (1000 :1) et de rapport de viscosité (10 :1). Nous modélisons l'expérience de l'écroulement du barrage de J.C.Martin et W.J.Moyce en 2D et 3D et nous comparons avec les résultats expérimentaux. Ce cas test est souvent utilisé pour valider les modèles d'étude des écoulements instationnaires à surface libre. Il consiste à considérer une colonne d'eau rectangulaire, initialement au repos sur un plan horizontal rigide. L'eau commence à s'écrouler sous l'effet de la gravité. Le liquide s'étale sur le plan horizontal et la hauteur de la colonne se baisse sur le mur verticale. Nous comparons l'évolution de la colonne d'eau sur le plan horizontale et sa hauteur résiduelle sur le mur vertical avec deux simulations de maillages différents pour montrer la convergence de l'algorithme. En deuxième lieu, nous validons la capacité de l'algorithme proposé à modéliser des fluides comme le métal liquide (en particulier l'aluminium) et l'air de rapport de densité important (2385 :1) et de rapport de viscosité (18.18 :1) lors du processus du remplissage des moules par une coulée (métal liquide). Un benchmark du processus de remplissage conçu par le professeur Campbell a été établi par l'algorithme de remplissage. Les résultats numériques montrent une bonne correspondance entre l'expérience et le calcul approché.

Dans le chapitre 4, nous parallélisons le code de calcul avec la méthode de décomposition de domaine de Schwarz. En premier lieu, nous comparons dans le cas d'un moule de moyenne taille les résultats des codes séquentielles et parallèles. Nous avons remarqué une diffusion numérique de la solution en évoluant en temps dans les équations qui contiennent

les termes des courbes caractéristiques. Nous avons introduit une nouvelle méthode de calcul des courbes caractéristiques en étendant les sous domaines. Nous avons obtenus de très bons résultats en comparant les codes séquentielles et parallèles modifiés. En deuxième lieu, nous avons modélisé le remplissage d'un moule industrielle de plus grand taille qui exige des résolutions des systèmes algébriques de l'ordre de 10 millions inconnues avec le code de calcul parallèle avec 128 coeurs et nous avons présenté une étude de la forte scalabilité de la méthode.

Dans le chapitre 5, on donne la conclusion finale de la thèse et on présente les perspectives du travail.

Chapitre 1

Modélisation d'un fluide biphasique

Sommaire

1.1	Le problème de remplissage d'un moule	18
1.2	Etat de l'art sur la résolution numérique du problème de Navier-Stokes	21
1.3	Discrétisation du problème du fluide diphasique	23
1.3.1	Formulation variationnelle	23
1.3.2	Discrétisation	24
1.3.3	Résultats numériques	29
1.4	Solution régularisée et réinitialisation	34
1.4.1	Solution régularisée	34
1.4.2	Algorithme de réinitialisation	35
1.4.3	Résultats numériques	38
1.5	Conditions aux bords de Navier	41
1.5.1	Etat de l'art sur les conditions aux bords de Navier	41
1.5.2	Remplissage d'un réservoir avec les conditions aux bords de Navier	43
1.5.3	Résultats numériques	45
1.6	Conditions aux bords de Neumann sous effet gravitationnel .	47
1.6.1	Résultats numériques	50
1.7	Conservation de masse	52
1.7.1	Résultats numériques	53

Dans ce chapitre nous présentons l'algorithme que nous avons construit pour le schéma de transport de l'interface lors d'un écoulement diphasique incompressible non-miscible. De plus, nous rappelons les méthodes numériques existantes dans la littérature pour la résolution numérique du problème de Navier-Stokes. Nous discutons le choix des conditions aux bords appropriées pour les équations des écoulements et du transport d'interface et l'effet des conditions limites sur la forme de l'interface et sur la dissipation de masse. Nous présentons également des résultats numériques accompagnés de discussions montrant les avantages et les inconvénients correspondantes.

1.1 Le problème de remplissage d'un moule

Nous considérons un intervalle $[0, T] \subset \mathbb{R}$, où T est un réel positif, et un instant quelconque $t \in [0, T]$. Soit Ω un domaine borné, ouvert, simplement connexe dans $\mathbb{R}^d$, $d = 2$ ou 3, de frontière Lipschitzienne $\partial\Omega$. Nous notons par $\mathbf{n}$ la normale unitaire sortante sur $\partial\Omega$.

Soit $(\mathbf{e}_1, \mathbf{e}_2)$ la base canonique de $\mathbb{R}^2$ (respectivement $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ la base canonique de $\mathbb{R}^3$).

Nous supposons que Ω représente un moule qui contient deux fluides, ainsi, elle est divisée en deux sous-domaines Ω_1 et Ω_2 qui varient avec le temps et qui sont séparées par une interface Γ tels que $\overline{\Omega} = \overline{\Omega}_1(t) \cup \overline{\Omega}_2(t)$ et $\Omega_1(t) \cap \Omega_2(t) = \emptyset$. (Voir figure 1.1).

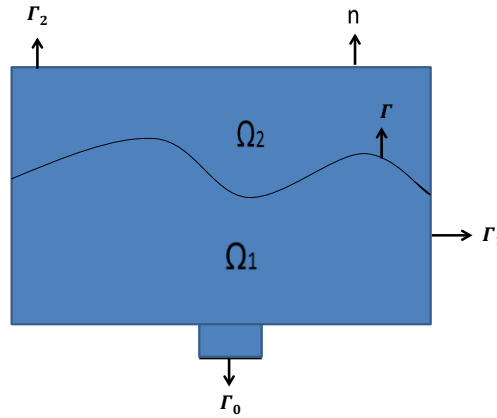


FIGURE 1.1 – Un domaine quelconque Ω

Nous notons par $\partial\Omega_i$ le bord de Ω_i , $i = 1, 2$, qui est constitué de : $\partial\Omega_1 \cap \partial\Omega_2 = \Gamma(t)$ où Γ est l'interface qui sépare les deux fluides et qui correspond au bord entre les deux fluides, $\partial\Omega = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$ où Γ_0 est le fond du bord qui correspond à l'entrée comme indiqué dans la figure 1.3, Γ_2 est le bord d'en haut qui correspond au bord de sortie et $\Gamma_1 = \partial\Omega \setminus \Gamma_0 \cup \Gamma_2$ est le reste du bord qui correspond au mur.

L'objectif de cette section est d'étudier le transport de l'interface entre les deux fluides lors du remplissage d'un moule.

Nous notons par $\mathbf{u}_i$ et p_i respectivement la vitesse et la pression dans le domaine $\Omega_i, i = 1, 2$, et par μ_i et ρ_i respectivement la viscosité et la densité dynamique constante du fluide dans $\Omega_i, i = 1, 2$.

Le mouvement du fluide biphasique est décrit dans chaque sous-domaine et à chaque instant $t \in [0, T]$ par les équations de Navier-Stokes suivantes :

$$(1.1) \quad \begin{cases} \rho_i \left(\frac{\partial \mathbf{u}_i}{\partial t} + (\mathbf{u}_i \cdot \nabla) \mathbf{u}_i \right) - \operatorname{div}(2\mu_i \mathbb{D}\mathbf{u}_i) + \nabla p_i = \mathbf{f}_i & \text{dans } \Omega_i \ (i = 1, 2), \\ \operatorname{div} \mathbf{u}_i = 0 & \text{dans } \Omega_i \ (i = 1, 2), \end{cases}$$

où $\mathbb{D}\mathbf{u}_i = \frac{1}{2}(\nabla \mathbf{u}_i + {}^t\nabla \mathbf{u}_i)$ est le tenseur de déformation et $\mathbf{f}_i$ représente les forces extérieures sur $\Omega_i, i = 1, 2$.

La densité (respectivement la viscosité) du fluide peut être écrite dans Ω sous la forme suivante :

$$\rho(\mathbf{x}, t) = \rho_1 1_{\mathbf{x} \in \Omega_1(t)} + \rho_2 1_{\mathbf{x} \in \Omega_2(t)} \quad (\text{respectivement } \mu(\mathbf{x}, t) = \mu_1 1_{\mathbf{x} \in \Omega_1(t)} + \mu_2 1_{\mathbf{x} \in \Omega_2(t)}),$$

où $1_{\mathbf{x} \in \Omega_i}$ est la fonction caractéristique du sous-domaine $\Omega_i, i = 1, 2$.

Nous notons par $\mathbf{f}$ la densité des forces extérieures, $\mathbf{u}$ la vitesse et p la pression du fluide dans Ω tel que $\mathbf{f} = \mathbf{f}_i, \mathbf{u} = \mathbf{u}_i$ et $p = p_i$ dans $\Omega_i, i = 1, 2$.

Le problème (1.7) peut être réécrit sous la forme suivante :

A chaque instant $t \in]0, T[$:

$$(1.2) \quad \begin{cases} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) - \operatorname{div}(2\mu \mathbb{D}\mathbf{u}) + \nabla p = \mathbf{f} & \text{dans } \Omega_1 \cup \Omega_2, \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } \Omega_1 \cup \Omega_2. \end{cases}$$

Ce système doit être muni des conditions aux bords et initiales correspondantes, ainsi, nous considérons sur l'entrée Γ_0 une condition aux bords de type Dirichlet non-homogène $\mathbf{u} = U_{in}$, sur le mur Γ_1 une condition aux bords de type Dirichlet homogène $\mathbf{u} = 0$ et sur Γ_2 une condition de surface libre $(2\mu \mathbb{D}\mathbf{u} - p\mathbf{I}) \cdot \mathbf{n} = 0$.

En outre, le système (1.2) est également complété par une condition de saut sur l'interface $\Gamma(t)$ qui impose la continuité de la vitesse et l'équilibre de la contrainte normale par la tension superficielle sur l'interface.

$$[\mathbf{u}]|_{\Gamma} = 0 \quad \text{et} \quad [2\mu \mathbb{D}\mathbf{u} - p\mathbf{I}]|_{\Gamma} \cdot \mathbf{n} = 0,$$

où $[\cdot]|_{\Gamma}$ est le saut à travers Γ dans la direction de la normale de Ω_1 , i.e. $[\cdot]|_{\Gamma} = \cdot|_{\Omega_1} - \cdot|_{\Omega_2}$.

Remarque 1.1.1. Dans ce travail, nous négligeons la force de tension sur l'interface.

Le problème à étudier sera le suivant :

A chaque temps $t \in]0, T[$

$$(1.3) \quad \left\{ \begin{array}{ll} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) - \operatorname{div}(2\mu \mathbb{D} \mathbf{u}) + \nabla p = \mathbf{f} & \text{dans } \Omega_1 \cup \Omega_2, \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } \Omega_1 \cup \Omega_2, \\ \mathbf{u} = U_{in} & \text{sur } \Gamma_0, \\ \mathbf{u} = 0 & \text{sur } \Gamma_1, \\ (2\mu \mathbb{D} \mathbf{u} - pI) \cdot \mathbf{n} = 0 & \text{sur } \Gamma_2, \\ [2\mu \mathbb{D} \mathbf{u} - pI] \cdot \mathbf{n} = 0 & \text{sur } \Gamma, \\ [\mathbf{u}] = 0 & \text{sur } \Gamma, \\ \mathbf{u}(\mathbf{x}, 0) = 0 & \text{dans } \Omega, \end{array} \right.$$

où $\mathbf{f} = -\rho g \mathbf{e}_2$ en dimension 2 ($\mathbf{f} = -\rho g \mathbf{e}_3$ en dimension 3 respectivement) est le vecteur de la force gravitationnelle, I est la matrice identité et U_{in} désigne la vitesse du flux entrant sur Γ_0 .

Pour le transport de l'interface, la difficulté principale est de gérer les changements géométriques et topologiques. Ainsi, nous résolvons le problème en utilisant la fonction Level-Set sur un maillage uniforme fixe. En particulier, nous suivons l'approche d'Osher et Sethian (Voir [107]), nous introduisons la fonction distance signée à l'interface $\Gamma(t)$:

$$(1.4) \quad \phi(x, t) = \pm \min_{y \in \Gamma(t)} |x - y|,$$

où la fonction ϕ est considérée négative dans le domaine $\Omega_1(t)$ et positive dans le domaine $\Omega_2(t)$. Par conséquent, à chaque temps t , l'interface des fluides correspond à l'isocontour nulle de la fonction continue ϕ :

$$\Gamma = \{\mathbf{x} \in \mathbb{R}^d / \phi(\mathbf{x}, t) = 0, \forall t \geq 0\}.$$

La densité ρ et la viscosité μ peuvent être réécrites sous la forme suivante :

$$\rho(\mathbf{x}, t) = \rho_1(\phi \leq 0) + \rho_2(\phi > 0) \text{ et } \mu(\mathbf{x}, t) = \mu_1(\phi \leq 0) + \mu_2(\phi > 0).$$

L'interface est alors capturée, à chaque instant t , par l'advection de la fonction Level-Set par la vitesse du fluide. Il peut être décrit par l'équation de transport suivante :

$$(1.5) \quad \frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi = 0.$$

Cette équation est munie de la condition aux bords de type Neumann homogène $\frac{\partial \phi}{\partial \mathbf{n}} = 0$ et d'une condition initiale qui exprime la position de l'interface à l'instant initial $\phi(\mathbf{x}, 0) = \phi_0$.

Le problème de transport de l'interface sera le suivant :

$$(1.6) \quad \begin{cases} \frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi = 0 & \text{dans } \Omega, \\ \frac{\partial \phi}{\partial \mathbf{n}} = 0 & \text{sur } \partial\Omega, \\ \phi(\mathbf{x}, 0) = \phi_0 & \text{dans } \Omega. \end{cases}$$

Les systèmes d'équations (1.3) et (1.6) constituent alors notre problème complet à résoudre.

1.2 Etat de l'art sur la résolution numérique du problème de Navier-Stokes

Dans cette section, nous allons décrire diverses méthodes de résolution numériques utilisées pour approcher la solution des équations de Navier-Stokes.

Le problème de Navier-Stokes est construit à partir des équations de l'équilibre dynamique et de la continuité (conservation de la masse). Nous considérons un domaine de calcul Ω' ouvert borné qui désigne un fluide, ces équations peuvent s'écrire sous la forme : Trouver la vitesse $\mathbf{u}(x, t)$ et la pression $p(x, t)$ tels que $\forall (x, t) \in \Omega' \times \mathbb{R}^+$,

$$(1.7) \quad \begin{cases} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) - \operatorname{div}(2\mu \mathbb{D}\mathbf{u}) + \nabla p = \mathbf{f}, \\ \operatorname{div} \mathbf{u} = 0 \\ \text{conditions limites et initiales,} \end{cases}$$

où ρ est la densité, μ est la viscosité et $\mathbf{f}$ représente les forces extérieures sur Ω' .

La méthode de discrétisation en espace choisie est la méthode des éléments finis. Dans ce cadre, il existe deux approches générales pour le calcul de la vitesse et la pression : soit simultanément ou soit étape par étape. La première approche consiste en la recherche de la vitesse et de la pression en même temps, ce qui se traduit par de grands systèmes algébriques surtout pour les simulations tridimensionnelles (voir le livre de Cuvelier et al. [36] pour la discussion de ce type des méthodes). Dans la deuxième approche, la vitesse et la pression sont découplées, ceci induit la résolution des systèmes algébriques plus petits que le cas d'avant et, par conséquent, on obtient une réduction significative du coût

de calcul. Parmi ces méthodes, on peut citer celle basée sur l'équation de Poisson de la pression (voir le livre de Gresho et Sani [66]), la méthode de compressibilité artificielle (voir le livre de Temam [145]) et la méthode d'Uzawa pour la formulation lagrangienne augmentée des équations de Navier-Stokes (voir, par exemple, le livre de Glowinski et de Tallec [60]), la méthode de pénalisation (voir, par exemple, le livre de Gunzburger [70]), la méthode de projection (voir [29] et [143]), et etc.

A part le problème de couplage vitesse-pression, il y a un autre problème très important lié au traitement de la non-linéarité convective du problème de Navier-Stokes. Il existe deux grandes catégories pour le traitement, soit en résolvant directement en une seule étape, soit en découplant la partie purement convective du reste des équations de Navier-Stokes. Les méthodes implicites sont difficiles puisqu'elles nécessitent la résolution d'un problème de Navier-Stokes stationnaires qui n'a pas forcément une solution unique en utilisant souvent des schémas de type point fixe ou de type Newton. La non-unicité de la solution rend la convergence de ces méthodes incertaine à haute nombre de Reynolds, par contre ces méthodes ont des bonnes propriétés de stabilité. Pour les problèmes transitoires, les méthodes qui découplent la partie purement convective du reste des équations de Navier-Stokes semblent être supérieures puisqu'elle découple le terme de convection du reste de l'équation de Navier-Stokes et on n'a pas le problème de l'unicité. Parmi ces méthodes on cite le schéma à pas fractionnaire de Bristeau et al. [18] (voir aussi Glowinski et Pironneau [61].....) et l'algorithme de transport-diffusion de Pironneau [109].

Il y a encore un problème très important lié à l'approximation des équations de Navier-Stokes, à savoir le choix des discrétisations pour la vitesse et la pression. Il est connu que le système de Navier-Stokes est un problème de type point selle où les espaces d'approximation de la vitesse et la pression doivent satisfaire la condition de compatibilité inf-sup qui est aussi appelée la condition de Ladyzhenskaya - Babuska - Brezzi (LBB). Si les espaces de dimension fini ne satisfont pas cette condition, la solution numérique n'est pas précise. Cette condition est assez restrictive, puisqu'elle empêche l'utilisation de certaines interpolations d'ordre faible convenables pour la couple vitesse-pression. Pour satisfaire cette condition, nous pouvons soit choisir des paires d'espaces de dimension fini convenables pour la vitesse et la pression, voir, par exemple, les livres de Pironneau [111] et de Quarteroni et Valli [117] pour la collection des approximations par élément fini de la vitesse et la pression, soit en utilisant des paramètres numériques artificielles qui devraient être réglés correctement au schéma ce qui n'est pas toujours faciles (voir [117] pour une étude de ces techniques).

Conclusion

Nous allons résoudre le problème de Navier-Stokes en couplant la vitesse et la pression. Pour le traitement de la non-linéarité, nous découplons la partie purement convective du reste des équations de Navier-Stokes en utilisant l'algorithme de transport-diffusion de Pironneau, appelée encore méthode des caractéristiques pour la discrétisation en temps.

Cette méthode à l'avantage qu'elle n'exige pas la condition de stabilité restrictive (CFL) pour le pas de temps et à chaque itération, nous avons besoin de calculer le second membre seulement. Pour la discrétisation en espace, nous utilisons le paire P1bulle-P1 des éléments finis, qui est le moins coûteux qui satisfait la condition de stabilité inf-sup.

Le code de Navier-Stokes utilisé pour l'écoulement du fluide a été vérifié avec le logiciel Freefem++ avec Abboud [1] et a été validé avec le Benchmark de Turek dans les exemples de Freefem++ [128].

Le code de l'équation de transport utilisé pour le transport de l'interface a été validé et vérifié dans Freefem++ avec Pironneau [113].

1.3 Discrétisation du problème du fluide diphasique

Dans cette section, nous introduisons la formulation variationnelle correspondante au problème de remplissage d'un moule, nous discrétisons le système obtenu en utilisant la méthode des caractéristiques en temps et la méthode des éléments finis en espace et nous terminons par présenter des résultats numériques correspondantes.

1.3.1 Formulation variationnelle

Dans le but d'introduire la formulation variationnelle correspondante au problème (1.3) et (1.6), nous introduisons l'espace $L^2(\Omega)$ des fonctions de carré intégrable au sens de Lebesgues sur Ω . De même, nous introduisons les espaces de Sobolev suivants (pour des entiers m et p) :

$$W^{m,p}(\Omega) = \{\mathbf{v} \in L^p(\Omega), \partial^\alpha \mathbf{v} \in L^p(\Omega), \forall |\alpha| \leq m\},$$

$$H^m(\Omega) = W^{m,2}(\Omega),$$

menés de la semi-norme et la norme suivantes :

$$|\mathbf{v}|_{m,p,\Omega} = \left\{ \sum_{|\alpha|=m} \int_{\Omega} |\partial^\alpha \mathbf{v}(x)|^p dx \right\}^{\frac{1}{p}}$$

et

$$\|\mathbf{v}\|_{m,p,\Omega} = \left\{ \sum_{K \leq m} |\mathbf{v}|_{K,p,\Omega}^p \right\}^{\frac{1}{p}}.$$

Nous rappelons que les conditions aux bords du domaine sont :

Sur l'entrée Γ_0 une condition aux bords de type Dirichlet non-homogène $\mathbf{u} = U_{in}$, sur le mur Γ_1 une condition aux bords de type Dirichlet homogène $\mathbf{u} = 0$ et sur Γ_2 une condition de surface libre $(2\mu \mathbb{D}\mathbf{u} - pI) \cdot \mathbf{n} = 0$.

Nous notons X_u le sous-espace de H^1 défini par :

$$X_u = \{\mathbf{u} \in H^1(\Omega) / \mathbf{u} = U_{in} \text{ sur } \Gamma_0 \text{ et } \mathbf{u} = 0 \text{ sur } \Gamma_1\}$$

et X_v le sous-espace de H^1 défini par :

$$X_v = \{\mathbf{v} \in H^1(\Omega) / \mathbf{v} = 0 \text{ sur } \Gamma_0 \cup \Gamma_1\}.$$

Nous introduisons aussi l'espace

$$H_0^1(\Omega) = \{\mathbf{v} \in H^1(\Omega) / \mathbf{v} = 0 \text{ sur } \partial\Omega\}$$

de dual $H^{-1}(\Omega)$ muni de la norme duale suivante :

$$\|\mathbf{u}\|_{-1,\Omega} = \sup_{\mathbf{v} \in H_0^1(\Omega), \mathbf{v} \neq 0} \frac{\int_{\Omega} \nabla \mathbf{u} \nabla \mathbf{v}}{\|\mathbf{v}\|_{1,\Omega}},$$

Nous notons $M = L^2(\Omega)$, $Y = L^2(\Omega)$ et nous supposons que $\mathbf{f}(\cdot, \mathbf{x}) \in L^2(\Omega)$. La formulation faible du problème de Navier-Stokes couplé avec l'équation de transport est donnée par : chercher $(\mathbf{u}(\cdot, \mathbf{x}), p(\cdot, \mathbf{x})) \in X_u \times M$, $\phi \in Y$ tel que

$$(1.8) \quad \begin{cases} \left(\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi, r \right) = 0 & \forall r \in Y, \\ \left(\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right), \mathbf{v} \right) + (2\mu \mathbb{D} \mathbf{u}, \mathbb{D} \mathbf{v}) - (p, \operatorname{div} \mathbf{v}) = (\mathbf{f}, \mathbf{v}) & \forall \mathbf{v} \in X_v, \\ (q, \operatorname{div} \mathbf{u}) = 0 & \forall q \in M. \end{cases}$$

Les termes aux bords $\int_{\Gamma_0} (2\mu \mathbb{D} \mathbf{u} - pI) \cdot \mathbf{n} \mathbf{v}$ et $\int_{\Gamma_1} (2\mu \mathbb{D} \mathbf{u} - pI) \cdot \mathbf{n} \mathbf{v}$ s'annulent à cause des conditions aux bords de Dirichlet qui imposent une fonction test nulle sur les bords correspondants. Le terme aux bords $\int_{\Gamma_2} (2\mu \mathbb{D} \mathbf{u} - pI) \cdot \mathbf{n} \mathbf{v}$ est nulle puisque sur Γ_2 on a la condition $(2\mu \mathbb{D} \mathbf{u} - pI) \cdot \mathbf{n} = 0$.

Pour une discussion sur l'existence et de l'unicité de la solution de ce problème qui est au-delà du cadre de notre recherche, nous pouvons se référer à ([140], section 2.3).

1.3.2 Discrétisation

Dans cette section, nous présentons la stratégie numérique que nous avons conçu pour résoudre le système couplé continu obtenu dans la section précédente. Notre approche est basée sur la méthode des caractéristiques combinée avec une méthode d'éléments finis.

1.3.2.1 Discrétisation en temps : Méthode des caractéristiques.

La méthode des caractéristiques, encore appelée méthode de Lagrange-Garlekin a été introduite par Benqué [13] et analysée par Pironneau [109]. Le but de cette méthode est la discrétisation en temps du terme de convection non-linéaire tout en évitant les conditions de stabilité contraignantes comme la condition de stabilité Courant-Friedrichs-Lewy (CFL) des schémas de discrétisation classiques explicites.

Cette méthode consiste à réduire le problème de convection non-linéaire à un problème de recherche des courbes caractéristiques, c'est à dire de position de la particule à l'instant précédent considéré.

En effet, l'opérateur $\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi$ peut être considéré comme une dérivée totale qui transforme les coordonnées Eulériennes en coordonnées Lagrangiennes. Grâce à cette formulation, il est théoriquement possible de suivre les particules au cours du temps le long de leurs trajectoires en résolvant, pour chaque particule, une équation différentielle dite équation de caractéristiques :

$$(1.9) \quad \begin{cases} \frac{\partial X}{\partial t}(x, s; t) = \mathbf{u}(X(x, s; t), t); \\ X(x, s; s) = x, \end{cases}$$

où $X(x, s; t)$ définit la position d'une particule à l'instant t qui était en x à l'instant s . Cette approche nous permet théoriquement d'éviter les contraintes de stabilité sur le pas de temps (par exemple un bon choix du pas de temps $\Delta t \approx 1.5h/|\mathbf{u}|$ où $|\mathbf{u}|$ est la vitesse moyenne, ce choix a été proposé dans [112]).

De plus, il a été démontré dans [109] que si la courbe caractéristique est calculée exactement et le second membre est également intégré exactement, le système résultant est inconditionnellement stable.

L'équation différentielle ordinaire définie par (1.9) a pour solution une courbe caractéristique $X(x, s; t)$.

En posant $\phi(t) = \phi(X(t); t)$, on obtient :

$$\frac{D\phi}{Dt} = \left(\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi \right) (X(x, s; t), t) = 0.$$

Poursuite le long des courbes caractéristiques, la formulation variationnelle devient :

$$(1.10) \quad \begin{cases} \left(\frac{D\phi}{Dt}, r \right) = 0 & \forall r \in Y, \\ \left(\rho \frac{D\mathbf{u}}{Dt}, \mathbf{v} \right) + (2\mu \mathbb{D}\mathbf{u}, \mathbb{D}\mathbf{v}) - (p, \text{div}\mathbf{v}) = (f, \mathbf{v}) & \forall \mathbf{v} \in X_v, \\ (q, \text{div}\mathbf{u}) = 0 & \forall q \in M. \end{cases}$$

Nous découpons l'intervalle $[0, T]$ en N segments d'épaisseur $\Delta t = \frac{T}{N}$, nous introduisons les points $t_n = n\Delta t$, pour $n = 0, \dots, N$, et nous notons $\mathbf{u}^n(x) = \mathbf{u}(x, t^n)$, $p^n(x) = p(x, t^n)$, $\rho^n(x) = \rho(x, t^n)$, $\mu^n(x) = \mu(x, t^n)$ et $\phi^n(x) = \phi(x, t^n)$.

Le terme $\frac{D}{Dt}\phi$ sera approché par

$$(1.11) \quad \begin{cases} \frac{D}{Dt}\phi(x, t^{n+1}) \approx \frac{\phi(x, t^{n+1}) - \phi(\bar{x}, t^n)}{\Delta t} \\ \text{avec } \bar{x} \equiv X_{(\mathbf{u}^n, \Delta t)}(x) = x - \mathbf{u}^n(x) \Delta t, \end{cases}$$

et le terme $\frac{D}{Dt}\mathbf{u}$ par

$$(1.12) \quad \begin{cases} \frac{D}{Dt}\mathbf{u}(x, t^{n+1}) \approx \frac{\mathbf{u}(x, t^{n+1}) - \mathbf{u}(\bar{x}, t^n)}{\Delta t} \\ \text{avec } \bar{x} \equiv X_{(\mathbf{u}^n, \Delta t)}(x) = x - \mathbf{u}^n(x) \Delta t. \end{cases}$$

Le système discrétisé en temps associé à (1.10) sera alors :

$$(1.13) \quad \begin{cases} \frac{1}{\Delta t}(\phi^{n+1} - \phi^n \circ X_{(\mathbf{u}^n, \Delta t)}, r) = 0 & \forall r \in Y, \\ \left(\frac{\rho^{n+1}}{\Delta t}(\mathbf{u}^{n+1} - \mathbf{u}^n \circ X_{(\mathbf{u}^n, \Delta t)}), \mathbf{v} \right) + (2\mu^{n+1} \mathbb{D}\mathbf{u}^{n+1}, \mathbb{D}\mathbf{v}) \\ \quad - (p^{n+1}, \text{div}\mathbf{v}) = (f^n, \mathbf{v}) & \forall \mathbf{v} \in X_v, \\ (q, \text{div}\mathbf{u}^{n+1}) = 0 & \forall q \in M, \end{cases}$$

où les termes $\phi^n \circ X_{(\mathbf{u}^n, \Delta t)}(x) = \phi^n(x - \mathbf{u}^n(x)\Delta t)$ et $\mathbf{u}^n \circ X_{(\mathbf{u}^n, \Delta t)}(x) = \mathbf{u}^n(x - \mathbf{u}^n(x)\Delta t)$.

Le principal avantage de la discrétisation des équations de Navier-Stokes par la méthode des caractéristiques (1.13) réside d'une part du fait qu'on peut effectuer l'assemblage des matrices une fois pour toute, et d'autre part qu'on peut choisir un pas de temps grand ce qui apporte un gain considérable en temps de calcul.

- a) L'avantage de la discrétisation de l'équation de transport et l'équation de reinitiation qu'on va voir dans la suite par la méthode des caractéristiques réside dans le fait qu'à chaque itération, on n'a besoin de mettre à jour que le second membre, c'est à dire le premier membre reste constant au cours de la résolution. Alors que pour les équations de Navier-Stokes, puisque la densité et la viscosité varient, le second membre doit être calculé ainsi que la matrice doit être inversée à chaque itération.
- b) Concernant le problème de la discrétisation des courbes caractéristiques, ou plus précisément l'approximation de $X^n(x)$, les problèmes attendus sont :

- La courbe caractéristique peut traverser certains éléments du domaine ou même sortir du domaine de calcul, il est donc nécessaire de localiser l'élément qui contient le dernier point de l'advection ou de trouver les cas où celui-ci touche un bord limite.
- De plus, dans la pratique, le calcul exact des courbes caractéristiques ou le second membre est inabordable, il doit être calculé en utilisant des formules d'intégration numérique. Malheureusement, les approches numériques empêchent la stabilité inconditionnelle du schéma de Lagrange-Galerkin, soit d'ordre 1 ou d'ordre 2. Pour des études plus exhaustives au sujet de la stabilité de la méthode de Lagrange-Galerkin, on se réfère à [53].

1.3.2.2 Discrétisation en espace : Méthode des éléments finis

A partir de maintenant nous supposons que Ω est polygonal. Soit T_h une famille régulière de triangulation de Ω par des triangles (ou tétraèdres) k , de paramètre h . De plus, nous notons $\mathbf{u}_h$, p_h , ρ_h , μ_h et ϕ_h les approximations par éléments finis de $\mathbf{u}$, p , ρ , μ et ϕ . Nous décrivons dans cette section l'approximation en espace des courbes caractéristiques $X(x, t^{n+1}, t^n)$ dans l'espace. Nous remarquons que dans la résolution numérique, seulement une approximation u_h de u est connue à chaque degré de liberté du maillage T_h du domaine, et nous avons besoin de calculer une approximation de $X_h(x, t^{n+1}; t^n)$ où $X_h(x, t^{n+1}; t^n)$ est la solution au temps t^n de la courbe caractéristique approchée :

$$(1.14) \quad \begin{cases} \frac{\partial X_h}{\partial t}(x, t^{n+1}; t) = \mathbf{u}_h(X_h(x, t^{n+1}; t), t) \\ X_h(x, t^{n+1}; t^{n+1}) = x. \end{cases}$$

On obtient

$$(1.15) \quad X_h(x, t^{n+1}; t) = x - \int_t^{t^{n+1}} \mathbf{u}_h(X_h(x, t^{n+1}; t), t) dt.$$

Pour approcher $X_h(x, t^{n+1}; t)$, l'algorithme de calcul des caractéristiques est obtenue directement de l'équation (1.15) par :

$$(1.16) \quad X_h(x, t^{n+1}; t^n) = x - \Delta t u_h^n(x).$$

1.3.2.3 Formulation variationnelle discrète en temps et en espace

Nous introduisons les espaces discrets $X_{u,h} \subset X_u$, $X_{v,h} \subset X_v$, $M_h \subset M$, $Y_h \subset Y$.

De plus, nous notons $\mathbf{u}_h^{n+1}$, p_h^{n+1} , ρ_h^{n+1} , μ_h^{n+1} et ϕ_h^{n+1} les approximations par éléments finis de $\mathbf{u}^{n+1}$, p^{n+1} , ρ^{n+1} , μ^{n+1} et ϕ^{n+1} solutions de (1.17) à l'instant t^{n+1} .

La vitesse est discrétisée par des " mini-éléments " :

$$X_{v,h} = \{\mathbf{v}_h \in X_v \cap C^0(\bar{\Omega}); \forall k \in T_h; \mathbf{v}_{h|k} \in P_b(k)^d\},$$

où $P_b(k)$ est défini par la somme des fonctions $P_1(k)$ et des fonctions bulles sur k (pour tout élément k , la fonction bulle est égale au produit des coordonnées barycentriques associées aux sommets de k).

La pression est discrétisée par les éléments finis continus de degré 1 :

$$M_h = \{q_h \in M \cap C^0(\bar{\Omega}); \forall k \in T_h, q_{h|k} \in P_1(k)\}.$$

La fonction Level-Set est discrétisée par les éléments finis continus de degré 1 :

$$Y_h = \{r_h \in Y \cap C^0(\bar{\Omega}); \forall k \in T_h, r_{h|k} \in P_1(k)\}.$$

Le système discret correspondant à la formulation variationnelle (1.8) est de la forme suivante : chercher $(\mathbf{u}_h^{n+1}, p_h^{n+1}) \in X_{u,h} \times M_h$ et $\phi_h^{n+1} \in Y_h$ tels que

$$(1.17) \quad \begin{cases} \left(\frac{\phi_h^{n+1} - \phi_h^n \circ X_h^n}{\Delta t}, r_h \right) = 0 & \forall r_h \in Y_h, \\ \left(\rho_h^n \frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n \circ X_h^n}{\Delta t}, \mathbf{v}_h \right) + (2\mu_h^n \mathbb{D}\mathbf{u}_h^{n+1}, \mathbb{D}\mathbf{v}_h) - (p_h^{n+1}, \text{div}\mathbf{v}_h) = (f_h^n, \mathbf{v}_h) & \forall \mathbf{v}_h \in X_{v,h}, \\ (q_h, \text{div}\mathbf{u}_h^{n+1}) = 0 & \forall q_h \in M_h, \end{cases}$$

où ρ_h^n et μ_h^n sont les densités et viscosités discrètes correspondantes.

1.3.2.4 Le problème algébrique

Soient $\{\Phi_i\}_{i=1,\dots,I}$ et $\{\Psi_i\}_{i=1,\dots,J}$ les bases de X_h et M_h respectivement, avec $I = \dim(X_h)$ et $J = \dim(M_h)$. On note

$$A = (A_{ij}), \quad A_{ij} = \int_{\Omega} \Phi_i \Phi_j + \int_{\Omega} \nabla \Phi_i \cdot \nabla \Phi_j,$$

$$B = (Bx_{ij}, By_{ij}), \quad Bx_{ij} = - \int_{\Omega} \Psi_i \partial_x \Phi_j, \quad By_{ij} = - \int_{\Omega} \Psi_i \partial_y \Phi_j.$$

Alors on peut écrire (1.13) sous la forme :

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} U_h \\ p_h \end{pmatrix} = \begin{pmatrix} G_h \\ 0 \end{pmatrix},$$

où G_h est une fonction qui rassemble les forces exercées sur le fluide et les termes de discrétisation de l'opérateur de transport.

Il existe dans la littérature [133] plusieurs méthodes numériques simples pour approcher ce problème, parmi lesquelles on utilise la méthode de pénalisation qui modifie l'équation en ajoutant un terme supplémentaire qui n'a pas une grande influence sur la solution discrète mais qui assure la coercivité de la matrice de masse. Alors on introduit une forme bilinéaire, notée s et définie par $s(p_h^{n+1}, q_h) = -\varepsilon(p_h^{n+1}, q_h)$, où ε est une constante réelle très petite. On considère alors la formulation variationnelle pénalisée suivante :

$$(1.18) \quad \begin{cases} (\frac{\rho_h^n}{\Delta t} \mathbf{u}_h^{n+1}, \mathbf{v}_h) + (2\mu_h^n \mathbb{D} \mathbf{u}_h^{n+1}, \mathbb{D} \mathbf{v}_h) - (p_h^{n+1}, \text{div} \mathbf{v}_h) = (g, \mathbf{v}_h), \\ (-q_h, \text{div} \mathbf{u}_h^{n+1}) - s(p_h^{n+1}, q_h) = 0. \end{cases}$$

Le problème algébrique correspondant s'écrit :

$$\begin{pmatrix} A & B^T \\ B & -\varepsilon I \end{pmatrix} \begin{pmatrix} U_h^\varepsilon \\ p_h^\varepsilon \end{pmatrix} = \begin{pmatrix} G_h^\varepsilon \\ 0 \end{pmatrix}.$$

Dans [134], il a été démontré que pour ε suffisamment petit, il existe une constante $C > 0$ telle que

$$\|U_h - U_h^\varepsilon\| \leq \varepsilon C.$$

On élimine $p_h^\varepsilon = (1/\varepsilon)BU_h^\varepsilon$ pour obtenir la relation suivante :

$$(A + (1/\varepsilon)B^TB)U_h^\varepsilon = G_h^\varepsilon.$$

Comme la matrice $A + (1/\varepsilon)B^TB$ est symétrique, définie positive et creuse, on peut résoudre ce problème par des techniques connues par exemple sparse solver, MUMPS, CG,

1.3.3 Résultats numériques

Le but est de visualiser l'évolution de l'interface lors du remplissage d'un réservoir contenant initialement deux fluides incompressibles, visqueux et non-miscibles. Les résultats numériques sont établies avec le logiciel FreeFem++ [78].

On considère le domaine $\Omega =]0, a[\times]0, b[\cup]c, c + d[\times] - e, 0]$ qui est composé d'un réservoir rectangulaire bidimensionnel $]0, a[\times]0, b[$ où $a = 2 \text{ dm}$, $b = 1 \text{ dm}$ et d'un tuyau centré au milieu du fond de longueur $d = 0.4 \text{ dm}$ et de largeur $e = 0.2 \text{ dm}$.

Le trou de couleur jaune représente le bord Γ_0 , la couleur verte représente le bord Γ_1 et la couleur bleue représente le bord Γ_2 , (comme le montre la figure (1.3) ci-dessous).

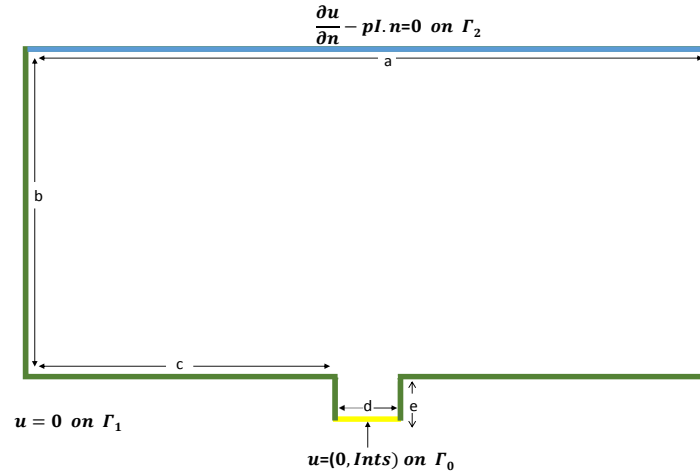


FIGURE 1.2 – Le réservoir du problème deux dimensionnels.

A l'instant initiale, l'interface Γ est représentée par la fonction Level-Set d'équation

$$\phi_0 = y - 0.1.$$

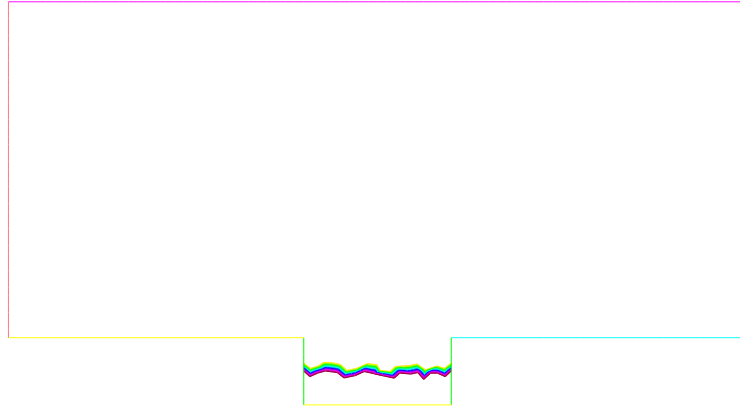


FIGURE 1.3 – L'interface du problème deux dimensionnels à l'instant initial

Le maillage considéré est représenté par la figure (1.4). Les bords Γ_0 , Γ_1 et Γ_2 sont découpés en N segments par unité de longueur (où $N = 50$).

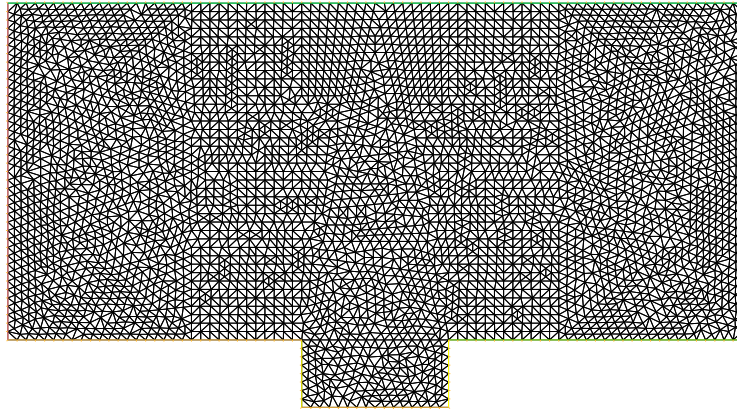


FIGURE 1.4 – Le maillage du problème deux dimensionnels.

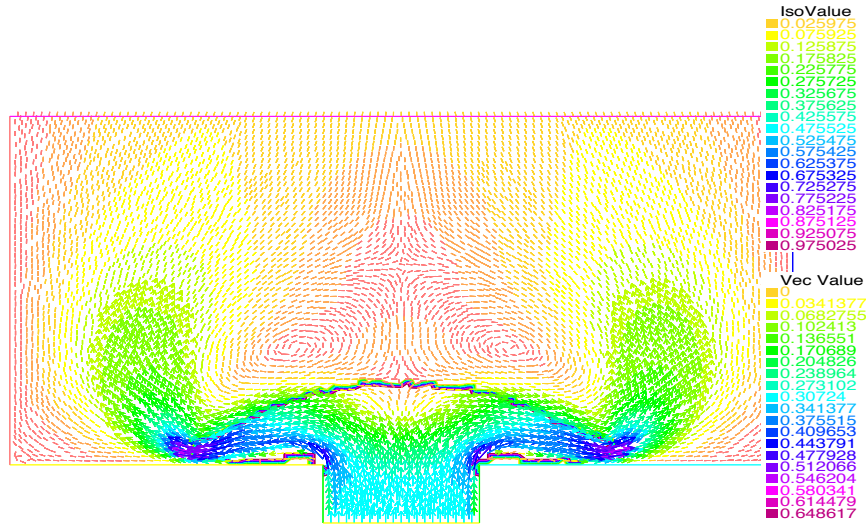


FIGURE 1.5 – L'interface du problème deux dimensionnel approchée par l'algorithme classique de Navier-Stokes à $t = 0.168 s$

Les figures (1.5), (1.6) et (1.7) montrent l'évolution de l'interface à l'instant $t = 0.168 s$, $t = 0.5 s$ et $t = 1.3 s$ respectivement.

On remarque que l'interface ne peut pas toucher le bord d'une manière naturelle et physique. De plus, on voit des discontinuités de l'interface dans le domaine.

Pour mieux voir le phénomène près du bord, on a fait la simulation avec un maillage plus grossier. La figure (1.8) montre l'interface qui ne touche pas le bord et qui reste toujours dans la dernière maille adjacente au bord.

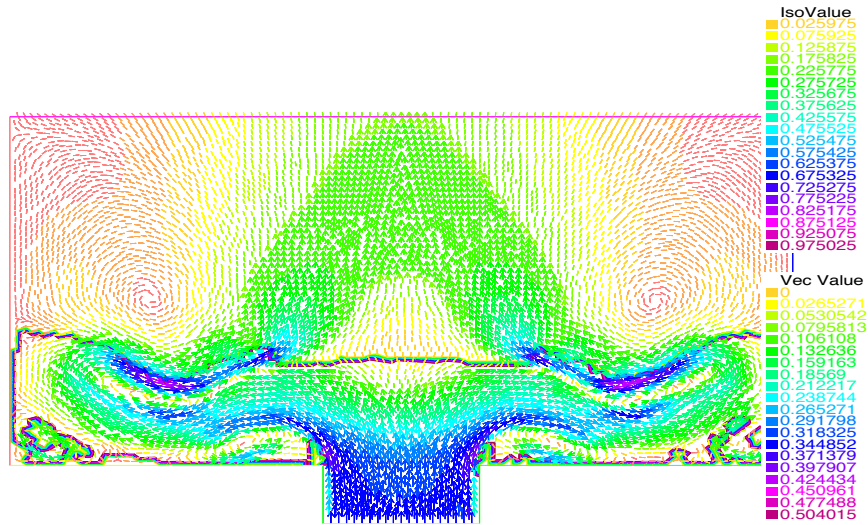


FIGURE 1.6 – L'interface du problème deux dimensionnel approchée par l'algorithme classique de Navier-Stokes à $t = 0.5$ s

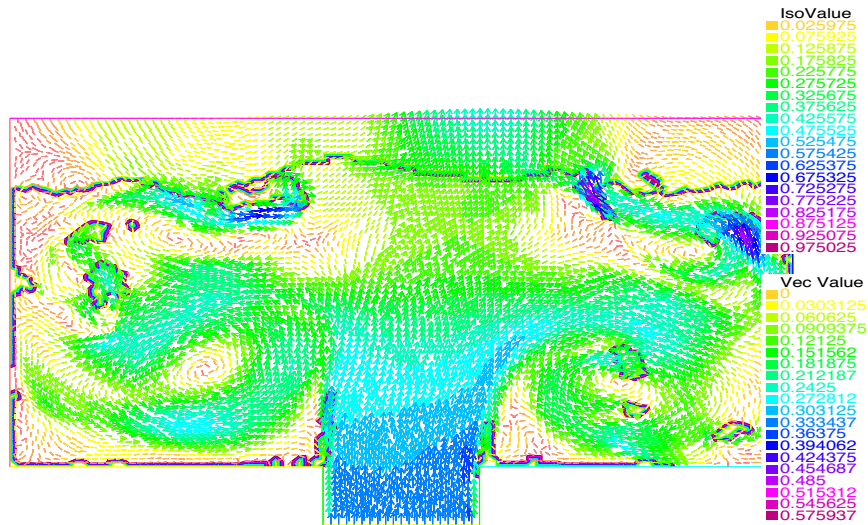


FIGURE 1.7 – L'interface du problème deux dimensionnel approchée par l'algorithme classique de Navier-Stokes à $t = 1.3$ s

Ces problèmes proviennent de l'équation de transport et ils sont inhérents à la méthode de caractéristiques. En fait, l'équation de transport de la fonction Level-Set est une équation non-linéaire hyperbolique. Une solution unique de cette équation peut être trouvée par la méthode de caractéristiques pour un temps suffisamment petit. Des discontinuités se

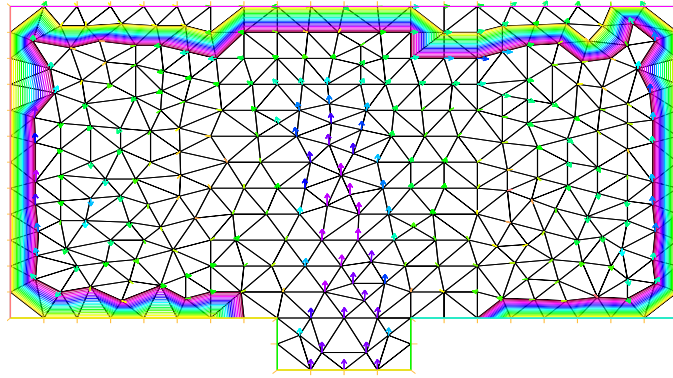


FIGURE 1.8 – L’interface du problème deux dimensionnel approchée par l’algorithme classique de Navier-Stokes avec un maillage grossier.

développent sur l’interface après un certain temps fini (Voir [127]). En effet, après un temps fini, des régions du domaine peuvent être traversées par plusieurs courbes caractéristiques qui se croisent (par suite on peut avoir plusieurs solutions) ou peuvent être non représentées par des courbes caractéristiques (par suite pas de solution).

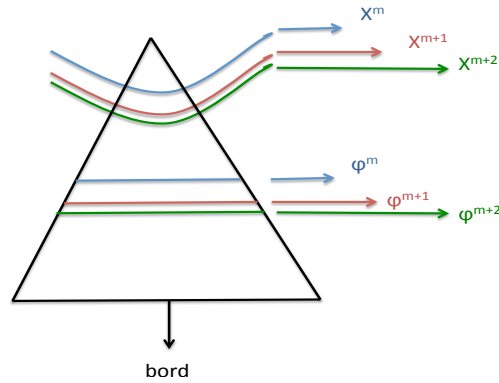


FIGURE 1.9 – Evolution des courbes caractéristiques et des isovaleurs de la fonction Level-Set dans le triangle adjacent au bord.

D’autre part, quand l’interface arrive au dernier triangle qui est adjacent au bord, la vitesse qui transporte le fluide est nulle sur les sommets du bord. La figure (1.9) illustre l’évolution des courbes caractéristiques et des isovaleurs correspondants aux instants t^m , t^{m+1} et t^{m+2} . On voit que les courbes caractéristiques ainsi que les isovaleurs de la Level-Set se rétrécissent de plus en plus en évoluant en temps et continuent à s’approcher proportionnellement, comme l’indique la figure (1.9) sans toucher le bord. Ce phénomène est dû aux interpolations des vitesses aux 3 sommets (avec deux sommets de vitesse nulle toujours). Ces interpolations ralentissent de plus en plus la vitesse de l’interface en s’approchant du bord. Ceci empêche l’interface de toucher le bord et lui exige de rester toujours dans la

dernière maille ce qui n'est pas physique.

Pour résoudre ce problème, on utilise dans la section suivante des solutions régularisées en modifiant l'équation de transport en lui ajoutant un terme de viscosité artificielle.

1.4 Solution régularisée et réinitialisation

L'équation de transport de la fonction Level-Set est un cas particulier des équations de type Hamilton-Jacobi (Voir [39]) qui s'écrit sous la forme suivante :

$$(1.19) \quad \frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \mathbf{n} |\nabla \phi| = 0, \quad \text{dans } [0, T] \times \Omega$$

où $|\cdot|$ est la norme euclidienne et $\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$ la normale à l'interface. Dans ce cas particulier, seule la composante normale de la vitesse du fluide décrit le mouvement de l'interface [130].

1.4.1 Solution régularisée

L'existence de la solution de l'équation (1.19) a été démontrée par plusieurs auteurs, parmi lesquels on cite A.Douglis [42], S.N.Kruzkov [89], [87], [88], W.H. Fleming [48], [49], [47], A.Friedman [50], S.H.Benton [12], et P.-L.Lions [95].

L'étude de l'unicité de la solution de l'équation (1.19) pour des hypothèses assez générales (pour un temps assez large,...) a exigé une nouvelle notion des solutions plus générale que les solutions faibles, les solutions au sens de viscosité. La méthode de viscosité évanescante a été introduite par P.-L.Lions et M.G. Crandall [33], elle consiste à ajouter un terme de viscosité artificielle $-\epsilon \Delta \phi_\epsilon$ qui disparaît par passage à la limite [8].

Des conditions aux bords appropriées doivent être également attribuées à ϕ . Dans cette section, on considère les conditions aux bords de Neumann homogènes pour ϕ (Voir [105]).

Ce terme ajouté $-\epsilon \Delta \phi_\epsilon$ enrichit la régularité de la solution ϕ pour devenir dans $H^1(\Omega)$. L'équation (1.19) complétée par des conditions aux bords de type Neumann sera exprimée de la manière suivante pour ϵ petit

$$(1.20) \quad \begin{aligned} \partial_t \phi_\epsilon + (\mathbf{u} \cdot \nabla) \phi_\epsilon - \epsilon \Delta \phi_\epsilon &= 0. \\ \frac{\partial \phi_\epsilon}{\partial \mathbf{n}} &= 0. \end{aligned}$$

La formulation variationnelle de l'équation de transport devient :

Trouver $\phi \in H^1(\Omega)$ tel que $\forall r \in H^1(\Omega)$, on a :

$$(1.21) \quad \int_{\Omega} \left(\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi \right) r + \epsilon \int_{\Omega} \nabla \phi \nabla r = 0.$$

Le choix de ϵ est assez délicat. Dans la littérature (voir par exemple [105]) la valeur de ϵ est proportionnelle au pas du maillage telle que $\epsilon = ch^{1-d}$ où c est une constante et $0 \leq d \leq 1$. On a effectué des simulations numériques pour diverses valeurs de ϵ et on a remarqué qu'une amélioration de la représentation de l'interface commence à partir de $\epsilon \geq h/50$. De plus, On a remarqué que l'erreur en volume du remplissage diffuse de plus en plus lorsque avec la valeur de ϵ augmente. Tandis que pour $h/50 \leq \epsilon < h/10$ on voit des oscillations qui causent la rupture de l'interface. Pour $\epsilon \geq h/10$, il y a une disparition des discontinuités et des oscillations (comme on va voir dans les résultats numériques). Ces phénomènes sont dues à l'irrégularité qui affecte la fonction Level-Set ϕ avec le temps. Le problème de cette méthode est la perte de la propriété distance $\|\nabla \phi\| = 1$, qui ne se conserve pas durant les itérations. Dans ce cadre, on a remarqué que la valeur de $\|\nabla \phi\|$ diverge plus rapidement avec la diminution du paramètre ϵ , ce qui explique la corruption de l'interface pour les petites valeurs de ϵ (voir Fig. 1.10).

Numériquement, on va voir dans le chapitre 3 en faisant la validation avec l'expérience de Martin et Moyce que la convergence de l'algorithme s'obtient pour le choix de $\epsilon = h/5$ et cette valeur a encore été utilisée dans le chapitre 3 pour la validation du Benchmark de Campbell. Pour ça on a adopté cette valeur pour les résultats numériques le long de toute la thèse.

1.4.2 Algorithme de réinitialisation

Dans cette partie, on se concentre à l'amélioration des résultats déjà obtenus. Comme on a noté, l'un des principaux inconvénients de la simulation des écoulements diphasiques incompressibles avec la méthode Level-Set est la perte de la propriété distance de la fonction ϕ . D'où l'introduction d'un algorithme de réinitialisation pour y remédier.

La fonction Level-Set est une fonction régulière vérifiant la propriété

$$(1.22) \quad \|\nabla \phi\| = 1$$

qui est une conséquence directe de la propriété distance. Cette propriété est très intéressante d'un point de vue numérique.

En fait, lorsque la fonction Level-Set possède une forte courbure locale, la propriété de régularité (1.22) peut être perdue. En effet, la fonction Level-set devient très inclinée ou très plate dans des régions du domaine en particulier près de l'interface ce qui détruit la forme de l'interface (Fig. 1.10).

A l'état initial, toutes les lignes de niveaux sont bien présentées selon la définition de la méthode Level-Set (Fig.1.11, à gauche). Au bout d'un temps t , l'advection par un champ

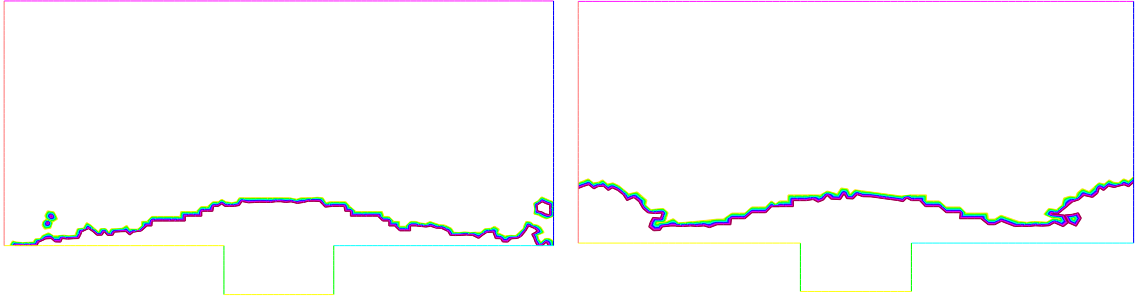


FIGURE 1.10 – Distortions de l'interface avant réinitialisation.

de vitesse a provoqué le rétrécissement des lignes de niveaux dans certaines régions et l'écartement dans d'autres (Fig.1.11, à droite).

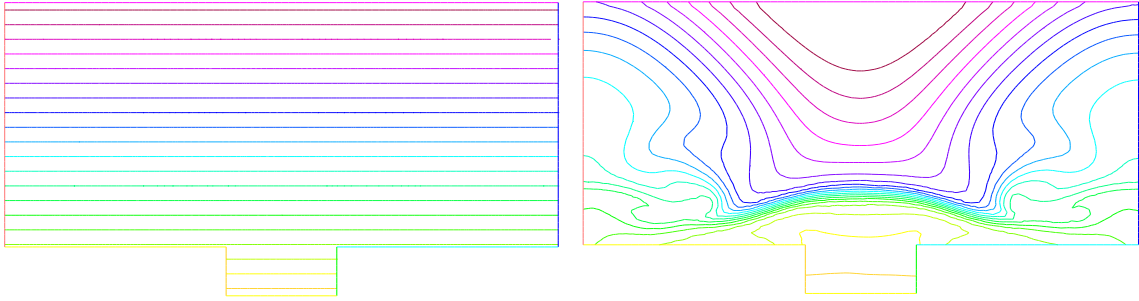


FIGURE 1.11 – Les isocontours de ϕ . A gauche : à $t = 0$, à droite : à $t = 5$

Ces oscillations n'affectent pas seulement la forme de l'interface, mais aussi la qualité de la normale, la courbure, le terme de viscosité artificielle et la conservation de la masse.

Il est alors nécessaire de corriger la position des lignes de niveau par rapport à la ligne de niveau 0, de manière à respecter la propriété de distance algébrique (1.22), ceci se fait en utilisant des algorithmes de réinitialisation de la fonction Level-Set.

1.4.2.1 Equation de réinitialisation

La réinitialisation est tout simplement le processus d'amélioration de la fonction Level set ϕ en la remplaçant par une autre fonction Φ qui a le même contour zéro mais qui preserve la propriété de distance pendant l'évolution du temps.

La nécessité de la réinitialisation dans les problèmes d'hydrodynamique a été mentionnée par Sussman et al. [137]. Le principe de cet algorithme est de corriger de façon itérative la position des lignes de niveau (iso-contours) à partir de la seule ligne de niveau valable qui est l'interface. Cette correction doit être faite de manière à respecter la propriété de distance algébrique. L'algorithme de réinitialisation se présente sous la forme d'une équation aux dérivées partielles instationnaires que l'on doit résoudre à la fin de chaque pas

de temps physique. La réinitialisation de la fonction Level-Set sur l'ensemble du domaine correspond à la solution stationnaire de l'équation d'Hamilton-Jacobi ci-dessous :

$$(1.23) \quad \frac{\partial \Phi}{\partial \tau} = \text{sign}(\phi)(1 - \|\nabla \Phi\|),$$

où τ est un temps fictif de réinitialisation permettant l'atteinte de l'état d'équilibre. Cette équation se complète avec la condition initiale suivante :

$$(1.24) \quad \Phi(x, t, \tau = 0) = \phi(x, t).$$

En théorie, l'équation (1.23) ne modifie pas la position de l'interface et est donc utilisée pour recalculer la position des autres lignes de niveau de manière indépendante, en respectant la propriété de distance algébrique. En pratique, les erreurs importantes de discrétisation introduites par les schémas numériques peuvent causer un changement de la position de l'interface.

Pour la discrétisation de l'équation (1.23), on l'écrit de la forme suivante :

$$(1.25) \quad \frac{\partial \Phi}{\partial \tau} + w \nabla \Phi = \text{sign}(\phi) \quad \text{avec } w = \text{sign}(\phi) \frac{\nabla \Phi}{\|\nabla \Phi\|}.$$

1.4.2.2 Discrétisation de l'équation de réinitialisation

L'équation (1.23) est discrétisée en temps en utilisant la méthode des caractéristiques et la formulation variationnelle discrète correspondante sera alors : Trouver $\Phi^{n+1} \in Y$ tel que $\forall r \in Y$, on a :

$$(1.26) \quad \left(\frac{\Phi^{n+1} - \Phi^n \circ X_{(\mathbf{w}^n, \Delta\tau)}^n}{\Delta\tau}, r \right) - (\text{sign}(\phi)^n, r) = 0.$$

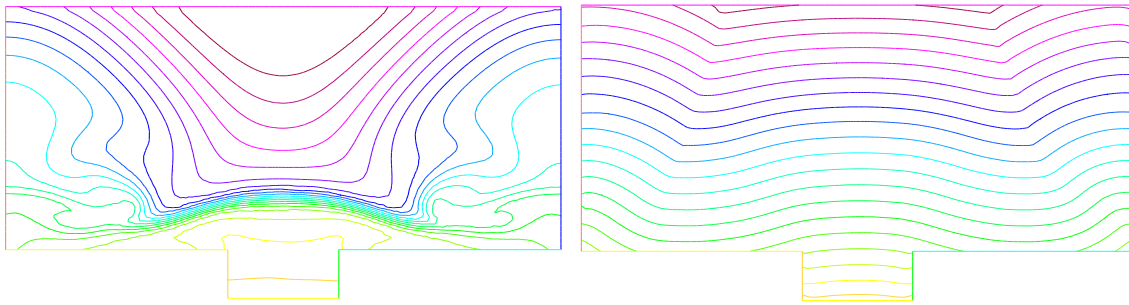
La discrétisation en espace est similaire à celle utilisée pour la fonction Level Set.

En ce qui concerne la fonction $\text{sign}(\phi)^n$, elle est approchée numériquement par une fonction lissée. Ce lissage est nécessaire pour obtenir de meilleurs propriétés de conservation et pour assurer la stabilité (Voir Peng et al.[108]) :

$$(1.27) \quad \text{sign}(\phi)_h^n = \frac{\phi_h^n}{\sqrt{\phi_h^{n2} + \|\nabla \phi_h^n\|^2 h^2}}.$$

La formulation variationnelle discrète s'écrit alors : Trouver $\Phi_h^{n+1} \in Y_h$ tel que $\forall r_h \in Y_h$, on a :

$$(1.28) \quad \left(\frac{\Phi_h^{n+1} - \Phi_h^n \circ X_{(\mathbf{w}_h^n, \Delta\tau)}^n}{\Delta\tau}, r_h \right) - (\text{sign}(\phi)_h^n, r_h) = 0.$$

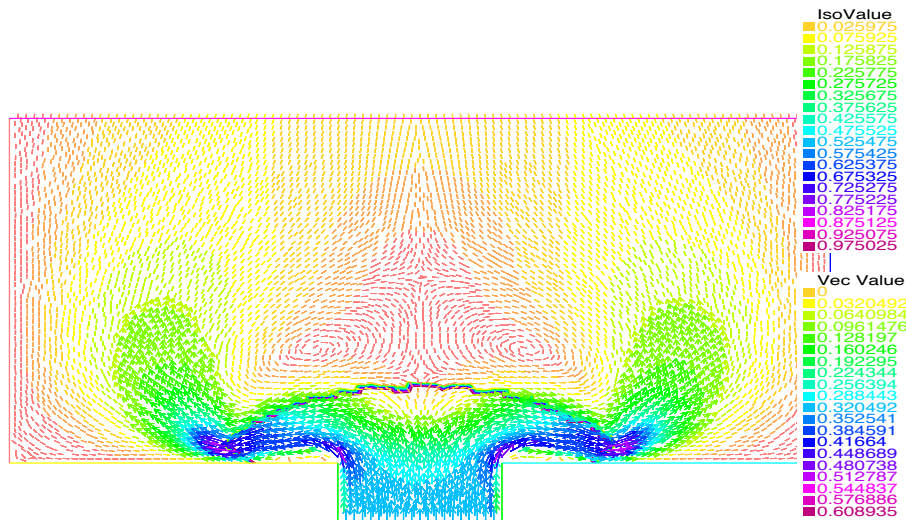
FIGURE 1.12 – Les isocontours de ϕ sans et avec réinitialisation à $t = 5$

La figure (1.12) montre bien les avantages de l'algorithme de réinitialisation où les lignes de niveaux après réinitialisation sont à distances égales et de plus la propriété de distance algébrique se préserve.

On a remarqué durant les essais numériques que cette propriété de distance algébrique est fondamentale pour l'erreur en volume et pour appliquer l'algorithme de conservation de masse qu'on va voir dans la section (1.7).

1.4.3 Résultats numériques

Le but de cette section est de visualiser la solution régularisée du problème (1.3) de l'évolution de l'interface lors du remplissage d'un réservoir. On considère le même exemple et les mêmes données de la section (1.3.3). Les figures (1.13), (1.14) et (1.15) montrent l'évolution de l'interface à l'instant $t = 0.168 s$, $t = 0.5 s$ et $t = 1.3 s$.

FIGURE 1.13 – L'interface du problème deux dimensionnels approchée par l'algorithme classique de Navier-Stokes et la solution régularisée pour l'équation de transport à $t = 0.168 s$

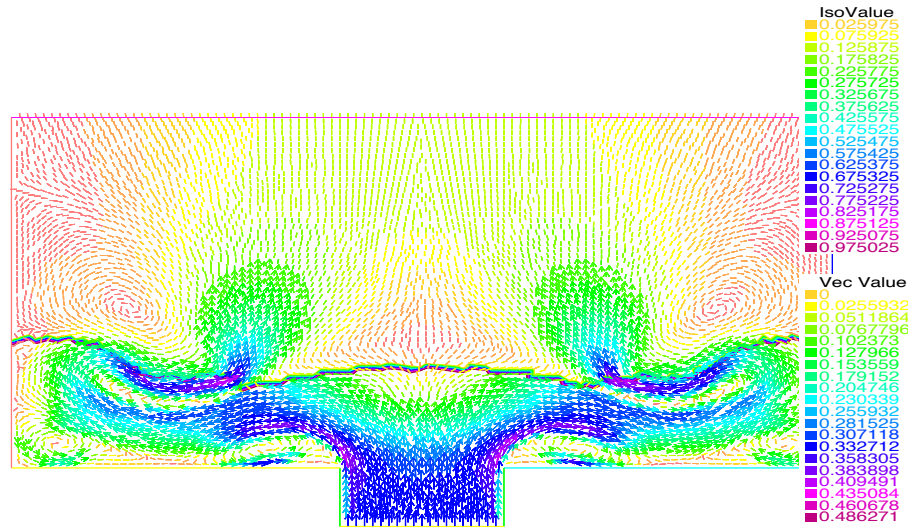


FIGURE 1.14 – L’interface du problème deux dimensionnels approchée par l’algorithme classique de Navier-Stokes et la solution régularisée pour l’équation de transport à $t = 0.5 \text{ s}$

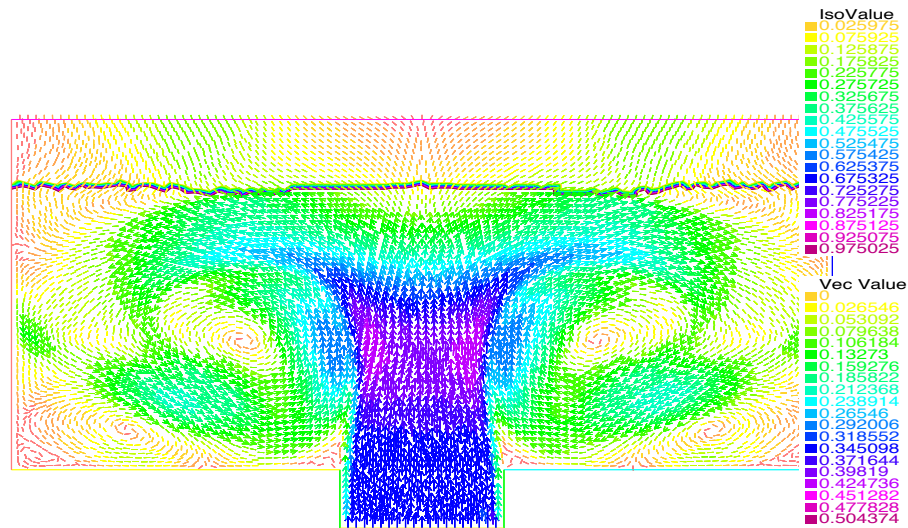


FIGURE 1.15 – L’interface du problème deux dimensionnels approchée par l’algorithme classique de Navier-Stokes et la solution régularisée pour l’équation de transport à $t = 1.3 \text{ s}$

Quelques constatations :

- On remarque qu’avec la solution régularisée, l’interface touche le bord. De plus, il n’y a pas des discontinuités qui se développent pour un temps assez large (le temps nécessaire pour finir le remplissage du réservoir).
- On remarque un problème de couche limite ([131]) près du bord sur l’interface fluide-fluide-solide causé par les conditions aux bords du type Dirichlet. En effet,

lorsqu'un fluide s'écoule le long d'un bord fixe, les vitesses sur le bord sont nulles (conditions aux bords de type Dirichlet) alors qu'à l'infini (c'est-à-dire loin du bord) elles sont égales à la vitesse de l'écoulement du fluide. La vitesse à l'infini se maintient jusqu'à une courte distance du bord et il y a une variation plus forte des vitesses dans une zone d'épaisseur très mince qui s'appelle la couche limite. Pour capter ces grandes variations de vitesses à la paroi, on a besoin des maillages très fin dans cette zone ce qui est très coûteux. Pour remédier à ce défaut, on propose de considérer dans la section suivante les conditions aux bords de type Navier qui permettent le glissement du fluide sur le bord.

Malgré cette amélioration au niveau de l'interface, des résultats numériques effectués ont montré que le volume diffuse avec le temps. Pour illustrer les idées, on considère l'erreur relative du volume

$$err_V = \frac{|V_e - V_n|}{|V_n|},$$

où V_e est le volume exacte et V_n est le volume numérique.

Pour $h = \frac{1}{50}$, on trace err_V en fonction du temps pour des valeurs de $\epsilon = h/5$.

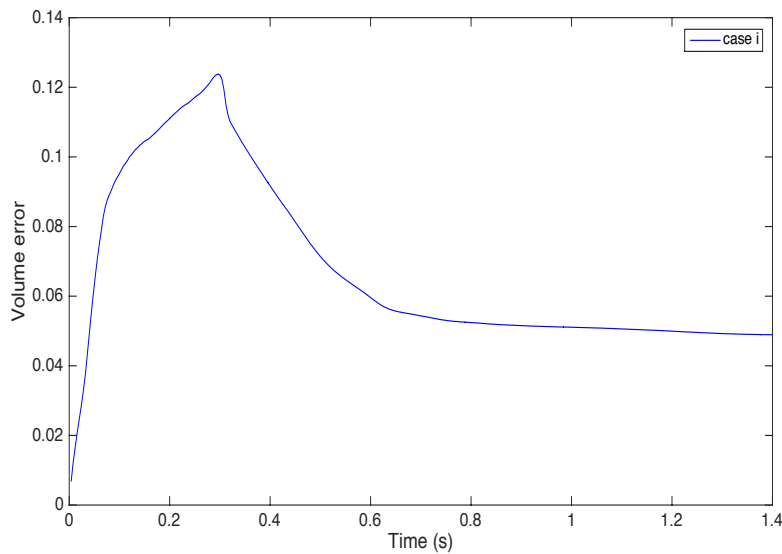


FIGURE 1.16 – Graphe de l'erreur de volume relative pour les conditions aux bords de Dirichlet.

On remarque que l'erreur relative du volume fluctue avec le temps où elle passe à un maximum de presque 12 pour cent et se stabilise à 5 pour cent pour un temps large. Or, l'écoulement considéré est incompressible et ceci doit se traduire par une conservation du volume. Dans les sections suivantes, on utilisera de nouveaux type de condition aux bords

pour éviter d'un part le problème de couche limite et pour diminuer d'autre part l'erreur de volume.

1.5 Conditions aux bords de Navier

1.5.1 Etat de l'art sur les conditions aux bords de Navier

La plupart des travaux jusqu'à nos jours considèrent les conditions aux bords de type Dirichlet. Cependant, comme l'a noté Serrin [131], celles-ci ne sont pas toujours réalistes et conduisent en général à des phénomènes de couches limites près des parois (Comme on a vu dans l'exemple précédent).

En fait, dans la première section, les conditions limites imposées sont des conditions de contact collant (une vitesse nulle aux parois). Physiquement dans le cas d'un fluide très visqueux comme les polymères, ces conditions sont conformes à la réalité, le fluide adhère aux parois du moule. Toutefois, dans le type d'écoulement que nous étudions, les viscosités mises en jeu sont faibles, et les vitesses de remplissage relativement importantes. De ce fait, en considérant une vitesse nulle à la paroi, les gradients de vitesse sont très importants (Figure 1.17). Si on considère des conditions de contact collant, il est très difficile de capter

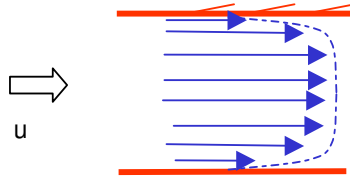


FIGURE 1.17 – Condition aux bords de type Dirichlet

les grandes variations de vitesse qui ont lieu à la paroi. Ces variations, et la forme des écoulements qui en découlent, peuvent toutefois être observées en utilisant un maillage extrêmement fin dans cette zone. En général, cette solution est abandonnée car elle est trop coûteuse. On préférera négliger la couche collante à la paroi de manière à avoir des formes d'écoulement réalistes. Dans le cadre de l'étude de fluide très peu visqueux, nous introduisons l'hypothèse du fluide glissant sur la paroi. Figure (1.18) nous montre

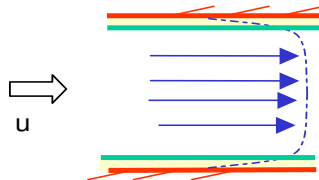


FIGURE 1.18 – Condition aux bords de type Navier

la couche limite collante négligée à l'aproximité de la paroi, ce qui permet de réduire l'ampleur des gradients de vitesse observés.

En 1827, Navier [102] a proposé une condition dite de glissement avec frottement à la paroi, qui permet de prendre en compte le glissement du fluide près du bord et de mesurer l'effet de frottement en considérant que la composante tangentielle du tenseur des contraintes est proportionnelle à la composante tangentielle, désigné par l'indice $\boldsymbol{\tau}$, du champ de vitesse :

$$(1.29) \quad \begin{cases} \mathbf{u} \cdot \mathbf{n} = 0, \\ \alpha \mathbf{u} \cdot \boldsymbol{\tau} + {}^t \mathbf{n} (2\mu \mathbb{D} \mathbf{u} - pI) \cdot \boldsymbol{\tau} = h. \end{cases}$$

Ces conditions peuvent être implémentées numériquement par pénalisation. Dans ce but, on les réécrit sous la forme suivante :

$$(1.30) \quad \begin{cases} \frac{1}{\beta} \mathbf{u} \cdot \mathbf{n} + {}^t \mathbf{n} (2\mu \mathbb{D} \mathbf{u} - pI) \cdot \mathbf{n} = g, \\ \alpha \mathbf{u} \cdot \boldsymbol{\tau} + {}^t \mathbf{n} (2\mu \mathbb{D} \mathbf{u} - pI) \cdot \boldsymbol{\tau} = h, \end{cases}$$

où α est le coefficient de frottement et $\mathbb{D} \mathbf{u}$ est le tenseur de déformations défini par :

$$\mathbb{D} \mathbf{u} = \frac{\nabla \mathbf{u} + {}^t \nabla \mathbf{u}}{2}.$$

Le vecteur tangentiel $\boldsymbol{\tau}$ est choisi tel que $(\mathbf{n}, \boldsymbol{\tau})$ soit un système orthonormal de vecteurs. Les fonctions α et β sont choisies pour être constantes positives par morceaux sur Γ_1 . Le choix $\beta \rightarrow 0$ donne la condition de non-pénétration (bord imperméable). Si $\beta \rightarrow 0$, par suite le paramètre $\alpha \rightarrow 0$ décrit la condition de glissement libre sur Γ_1 alors que $\beta \rightarrow \infty$ implique la condition de non-glissement (conditions aux limites de Dirichlet homogènes si h et g sont nulles). Un des cas pratiques et intéressants consiste à considérer α nul qui traduit une condition de glissement de Navier, mais sans frottement. Ces conditions sont aussi utilisées dans les simulations numériques d'écoulements en présence de parois rugueuses [5], [21], [20], [22], [80], [81] comme en aérodynamique, ou bien de parois perforées [10]. Dans le cas d'un bord plat, la deuxième condition ci-dessus peut être remplacée si $\alpha = 0$, par une condition portant sur la composante tangentielle du tourbillon (voir [151]) :

$$\mathbf{u} \cdot \mathbf{n} = g \quad \text{et} \quad (\nabla \wedge \mathbf{u}) \wedge \mathbf{n} = h \wedge \mathbf{n}.$$

Rappelons que ces conditions aux limites sont le plus souvent utilisées pour traiter des problèmes réalistes.

1.5.2 Remplissage d'un réservoir avec les conditions aux bords de Navier

Dans cette section, on désire remplacer les conditions aux bords de Dirichlet par les conditions aux bords de Navier. On considère le cas d'un glissement sans frottement $\alpha = 0$ et on choisit le paramètre de pénalisation β tel que $\beta \rightarrow 0$. De plus, on considère $h = 0$ et $g = 0$.

Il faut bien noter que d'un point de vue mathématique, on a un problème de comptabilité lors du changement du type des conditions aux bords. Ce problème vient des points d'intersection entre les conditions limites de glissement et les conditions aux bords artificielles de type Dirichlet. Durant la modélisation des conditions limites, le traitement des singularités due à l'incompatibilité des conditions limites ne modifie pas réellement l'écoulement globale. Si c'est le cas, on change les conditions limites.

On considère le problème de Navier-Stokes diphasique suivant : A chaque pas de temps $t \in]0, T[$

$$(1.31) \quad \left\{ \begin{array}{ll} \frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi - \varepsilon \Delta \phi = 0 & \text{dans } \Omega, \\ \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) - \operatorname{div}(2\mu \mathbb{D} \mathbf{u}) + \nabla p = \mathbf{f} & \text{dans } \Omega_1 \cup \Omega_2, \\ \operatorname{div} \mathbf{u} = 0 & \text{dans } \Omega_1 \cup \Omega_2, \\ \mathbf{u} = U_{in} & \text{sur } \Gamma_0, \\ \frac{1}{\beta} \mathbf{u} \cdot \mathbf{n} + {}^t \mathbf{n} (2\mu \mathbb{D} \mathbf{u} - pI) \cdot \mathbf{n} = 0 & \text{sur } \Gamma_1, \\ \alpha \mathbf{u} \cdot \boldsymbol{\tau} + {}^t \mathbf{n} (2\mu \mathbb{D} \mathbf{u} - pI) \cdot \boldsymbol{\tau} = 0 & \text{sur } \Gamma_1, \\ (2\mu \mathbb{D} - pI) \cdot \mathbf{n} = 0 & \text{sur } \Gamma_2, \\ [2\mu \mathbb{D} \mathbf{u} - pI] \cdot \mathbf{n} = 0 & \text{sur } \Gamma, \\ \varepsilon \frac{\partial \phi}{\partial \mathbf{n}} = 0 & \text{sur } \Gamma_0 \cup \Gamma_1 \cup \Gamma_2, \\ \phi(\mathbf{x}, 0) = \phi_0 & \text{dans } \Omega, \\ \mathbf{u}(\mathbf{x}, 0) = 0 & \text{dans } \Omega_1 \cup \Omega_2. \end{array} \right.$$

On introduit les espaces fonctionnels suivants

$$X_u = \{ \mathbf{u} \in H^1(\Omega) \mid \mathbf{u} = U_{in} \text{ sur } \Gamma_0 \},$$

$$X_v = \{ \mathbf{v} \in H^1(\Omega) \mid \mathbf{v} = 0 \text{ sur } \Gamma_0 \},$$

$$M = L^2(\Omega)$$

et

$$Y = H^1(\Omega).$$

La formulation variationnelle du problème (1.31) s'écrit sous la forme :

Trouver $(\mathbf{u}, p) \in X_u \times M$, $\phi \in Y$ tel que pour tout $(\mathbf{v}, q) \in X_v \times M$ et $\forall r \in Y$

$$(1.32) \quad \left\{ \begin{array}{l} \left(\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi, r \right) + \varepsilon (\nabla \phi, \nabla r) = 0. \\ \left(\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right), \mathbf{v} \right) + (2\mu \mathbb{D} \mathbf{u}, \mathbb{D} \mathbf{v}) - (p, \operatorname{div} \mathbf{v}) - \int_{\Gamma_0} (2\mu \mathbb{D} \mathbf{u} - pI) \mathbf{n} \cdot \mathbf{v} \, ds \\ \quad - \int_{\Gamma_1} (2\mu \mathbb{D} \mathbf{u} - pI) \mathbf{n} \cdot \mathbf{v} \, ds - \int_{\Gamma_2} (2\mu \mathbb{D} \mathbf{u} - pI) \mathbf{n} \cdot \mathbf{v} \, ds = (-\rho g \mathbf{e}_2, \mathbf{v}). \\ (\operatorname{div} \mathbf{u}, q) = 0. \end{array} \right.$$

Le terme sur Γ_0 s'annule à cause de la fonction test nulle, le terme sur Γ_2 s'annule à cause de la condition aux bords $(2\mu \mathbb{D} \mathbf{u} - pI) \mathbf{n} \cdot \mathbf{v}$. Pour Γ_1 , on implémente les conditions aux bords de Navier de la manière suivante :

En utilisant la décomposition de la fonction test

$$\mathbf{v} = (\mathbf{v} \cdot \mathbf{n}) \mathbf{n} + (\mathbf{v} \cdot \boldsymbol{\tau}) \boldsymbol{\tau},$$

l'intégrale sur Γ_1 sera exprimée de la manière suivante :

$$(1.33) \quad \begin{aligned} & \int_{\Gamma_1} (2\mu \mathbb{D} \mathbf{u} - pI) \mathbf{n} \cdot \mathbf{v} \, ds \\ &= \int_{\Gamma_1} {}^t \mathbf{n} (2\mu \mathbb{D} \mathbf{u} - pI) \mathbf{n} \, \mathbf{v} \cdot \mathbf{n} \, ds + \int_{\Gamma_1} {}^t \mathbf{n} (2\mu \mathbb{D} \mathbf{u} - pI) \boldsymbol{\tau} \, \mathbf{v} \cdot \boldsymbol{\tau} \, ds \\ &= \int_{\Gamma_1} \beta^{-1} (\mathbf{u} \cdot \mathbf{n}) (\mathbf{v} \cdot \mathbf{n}) \, ds + \int_{\Gamma_1} \alpha (\mathbf{u} \cdot \boldsymbol{\tau}) (\mathbf{v} \cdot \boldsymbol{\tau}) \, ds \\ &= \int_{\Gamma_1} \beta^{-1} {}^t \mathbf{u} (\mathbf{n} \, {}^t \mathbf{n}) \mathbf{v} \, ds + \int_{\Gamma_1} \alpha {}^t \mathbf{u} (\boldsymbol{\tau} \, {}^t \boldsymbol{\tau}) \mathbf{v} \, ds. \end{aligned}$$

De plus, la deuxième équation de la formulation variationnelle peut être réécrite sous la forme suivante :

$$(1.34) \quad \begin{aligned} & \left(\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right), \mathbf{v} \right) + (2\mu \mathbb{D} \mathbf{u}, \mathbb{D} \mathbf{v}) - (p, \operatorname{div} \mathbf{v}) - (q, \operatorname{div} \mathbf{u}) \\ & \quad - \int_{\Gamma_1} \beta^{-1} {}^t \mathbf{u} (\mathbf{n} \, {}^t \mathbf{n}) \mathbf{v} \, ds - \int_{\Gamma_1} \alpha {}^t \mathbf{u} (\boldsymbol{\tau} \, {}^t \boldsymbol{\tau}) \mathbf{v} \, ds = (-\rho g \mathbf{e}_d, \mathbf{v}). \end{aligned}$$

En ce qui concerne la discrétisation, on utilise les mêmes types de discrétisation en espace et en temps effectués dans la section (1.3.2). Le système discret associé au problème (1.32)

s'écrit sous la forme suivante :

Trouver $(\mathbf{u}_h^{n+1}, p_h^{n+1}) \in X_{u,h} \times M_h$, $\phi_h^{n+1} \in Y_h$ tel que pour tout $(\mathbf{v}_h, q_h) \in X_{v,h} \times M_h$ et $r_h \in Y_h$,

$$(1.35) \quad \left\{ \begin{array}{l} \left(\frac{\phi_h^{n+1}}{\Delta t}, r_h \right) + \varepsilon (\nabla \phi_h^{n+1}, \nabla r_h) = \left(\frac{\phi_h^n o X^n}{\Delta t}, r_h \right). \\ \left(\rho_h^n \frac{\mathbf{u}_h^{n+1}}{\Delta t}, \mathbf{v}_h \right) + (2\mu_h^n \mathbb{D} \mathbf{u}_h^{n+1}, \mathbb{D} \mathbf{v}_h) - (p_h^{n+1}, \text{div} \mathbf{v}_h) - (q_h, \text{div} \mathbf{u}_h^{n+1}) \\ \quad - \int_{\Gamma_1} \beta^{-1} {}^t \mathbf{u}_h^{n+1} ({}^n \mathbf{t} \mathbf{n}) \mathbf{v}_h ds - \int_{\Gamma_1} \alpha {}^t \mathbf{u}_h^{n+1} ({}^t \boldsymbol{\tau} \mathbf{t}) \mathbf{v}_h ds \\ \quad = (-\rho_h^n g \mathbf{e}_d, \mathbf{v}_h) + \left(\rho_h^n \frac{\mathbf{u}_h^n o X^n}{\Delta t}, \mathbf{v}_h \right). \end{array} \right.$$

1.5.3 Résultats numériques

Le but est de visualiser la solution avec les conditions aux bords de Navier du problème de l'évolution de l'interface lors du remplissage d'un réservoir. On considère le même exemple et les mêmes données de la section 1.4.3 et on choisit $\beta = 10^{-6}$ et $\alpha = 0$.

Les figures (1.19), (1.20) et (1.21) montrent l'évolution en fonction du temps de la solution approchée du transport de l'interface avec un écoulement muni des conditions aux bords de type Navier aux instants $t = 0.168$ s, $t = 0.5$ s et $t = 1.3$ s.

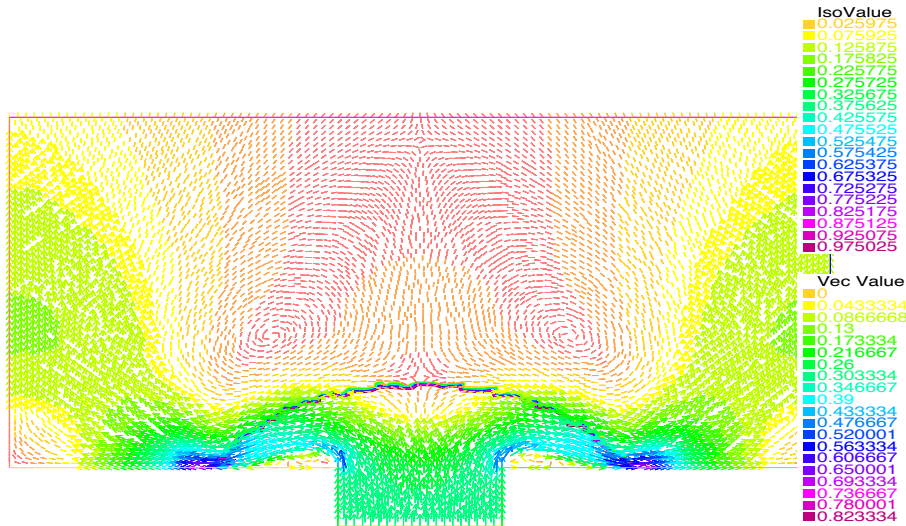


FIGURE 1.19 – L'interface approchée avec les conditions aux bords de Navier à $t = 0.168$ s.

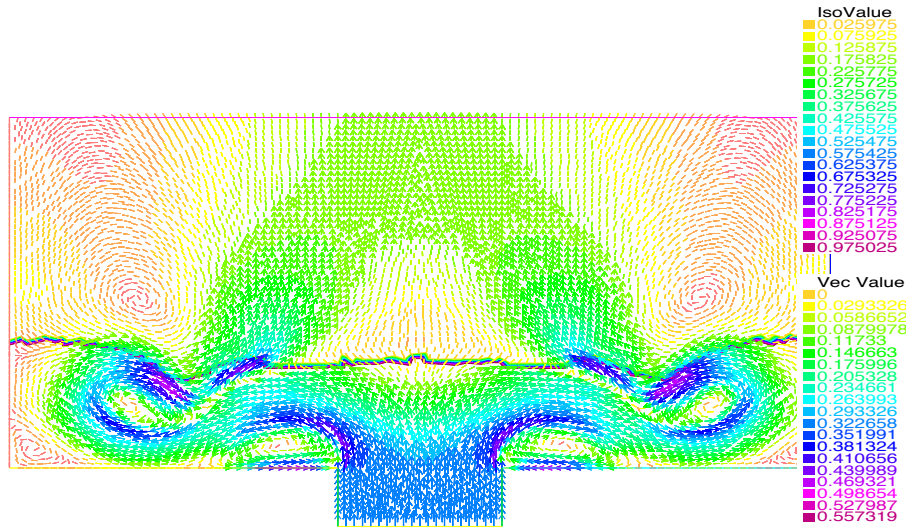


FIGURE 1.20 – L’interface approchée avec les conditions aux bords de Navier à $t = 0.5$ s.

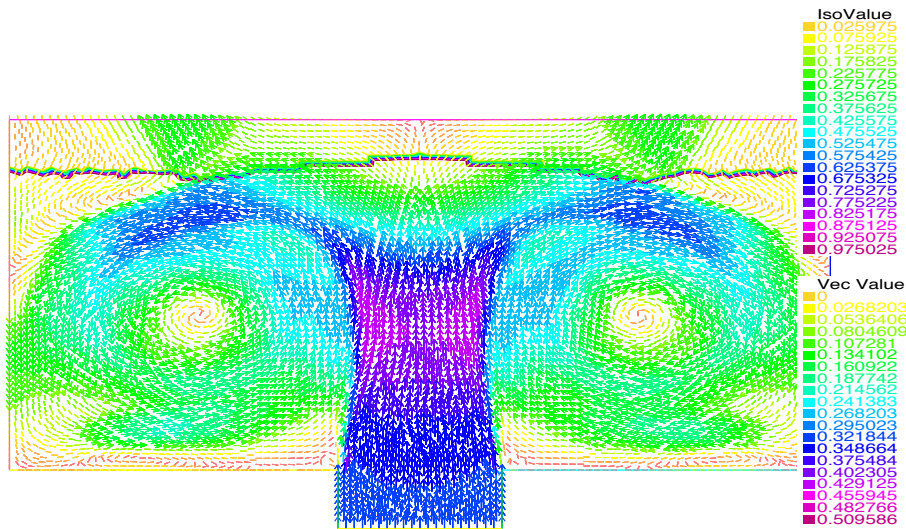


FIGURE 1.21 – L’interface approchée avec les conditions aux bords de Navier à $t = 1.3$ s.

Quelques constatations :

- L’interface fluide-fluide-solide se déplace près du bord imperméable d’une manière plus réaliste en utilisant la condition de Navier, par suite le choix ce type de condition aux bords est plus convenable que celle de Dirichlet.
- Avec les conditions du type Navier, l’interface se déplace en faisant un angle droit

avec le bord en particulier pour des grandes valeurs de la viscosité. Ceci est dû aux conditions aux bords de Neumann homogènes de l'équation de transport, ce qui n'est pas physique. Une idée est de trouver les conditions aux bords de Neumann sous l'effet gravitationnel de la pression qui sera l'objet d'étude de la section suivante.

La figure (1.22) montre l'erreur err_V en fonction du temps. On remarque que l'erreur de

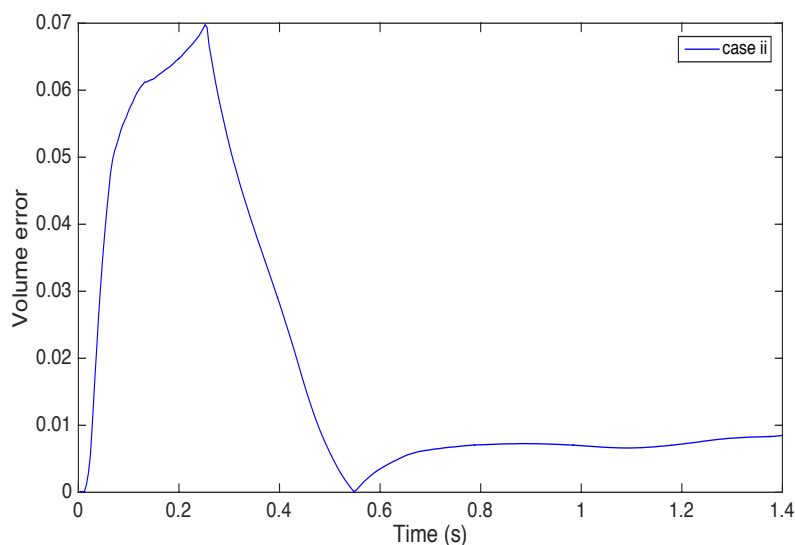


FIGURE 1.22 – Graphe de l'erreur de volume relative pour les conditions aux bords de Navier.

volume pour le problème avec les conditions de Navier a diminué par rapport à celle avec Dirichlet (Figure 1.16). On constate que cette erreur dépend fortement des conditions aux bords utilisées. D'où l'idée d'introduire pour la phase des conditions aux bords de Neumann sous l'effet gravitationnel de la pression qui va diminuer remarquablement l'erreur de volume comme on va voir dans la section suivante.

1.6 Conditions aux bords de Neumann sous effet gravitationnel

Dans Les sections précédentes, on a considéré les conditions aux bords de Neumann homogène qui sont utilisées dans de nombreuses applications (voir par exemple [105]). Pour améliorer l'angle que fait l'interface avec le bord et l'erreur en volume, on va proposer une nouvelle condition limite naturelle qui sera appelée condition aux bords de Neumann

non-homogène sous l'effet gravitationnel. Ce dernier type de conditions aux bords traduit la pente de l'angle entre la surface libre et le bord d'une manière naturelle et physique. La condition aux limites non homogène sous effet gravitationnel est une condition limite de la pression : on néglige le terme de convection au voisinage du bord Γ_1 et on considère la composante normale de la première équation de (1.3). On obtient en multipliant la première équation de (1.3) par $\mathbf{n}$:

$$(1.36) \quad \rho \left(\underbrace{\partial_t \mathbf{u} \cdot \mathbf{n}}_{(1)}|_{\Gamma_1} + \underbrace{\mathbf{u} \cdot \nabla \mathbf{u} \cdot \mathbf{n}}_{(2)}|_{\Gamma_1} \right) - \operatorname{div}(2\mu \mathbb{D} \mathbf{u}) \cdot \mathbf{n}|_{\Gamma_1} + \nabla p \cdot \mathbf{n}|_{\Gamma_1} = -\rho g \mathbf{e}_d \cdot \mathbf{n}|_{\Gamma_1}$$

Le premier terme (1) $(\partial_t \mathbf{u}) \cdot \mathbf{n} = \partial_t (\mathbf{u} \cdot \mathbf{n}) = 0$ puisque $\mathbf{n}$ ne dépend pas du temps et $\mathbf{u} \cdot \mathbf{n} = 0$.

On obtient alors la condition aux bords sur Γ_1 suivante :

$$\partial_{\mathbf{n}} p|_{\Gamma_1} = -\rho g \mathbf{e}_d \cdot \mathbf{n}|_{\Gamma_1} - \rho \left((\mathbf{u} \cdot \nabla) \mathbf{u} \right) \cdot \mathbf{n}|_{\Gamma_1} - 2\mu \mathbb{D} \mathbf{u} \cdot \mathbf{n}|_{\Gamma_1}.$$

Dans notre application, la viscosité est relativement faible. Ainsi, on néglige le second terme et la condition aux limites de la pression sur Γ_1 peut être écrite sous la forme suivante :

$$\partial_{\mathbf{n}} p|_{\Gamma_1} = -\rho g \mathbf{e}_d \cdot \mathbf{n}|_{\Gamma_1} - \rho \left((\mathbf{u} \cdot \nabla) \mathbf{u} \right) \cdot \mathbf{n}|_{\Gamma_1}.$$

Puisque le domaine Ω_2 contient un fluide de très faible densité, on néglige la pression correspondante. De même, dans notre application on néglige la force de tension superficielle. De plus, on impose sur Γ_0 une vitesse moyenne U_{in} constante tout au long du remplissage et sur l'interface Γ une condition de surface libre $[2\mu \mathbb{D} \mathbf{u} - pI] \cdot \mathbf{n} = 0$. D'autre part, puisque dans notre application on considère des fluides de faible viscosité, on suppose que les forces normales qui agissent sur l'interface sont négligeables, (Voir figure (1.27) et (2.11) dans les résultats numériques pour une comparaison entre les forces exercées sur l'interface et celles sur l'entrée qui valident l'hypothèse pour les problèmes en 2D et 3D). Par conséquent, la pression totale est négligeable sur l'interface Γ qui pourra être considérée comme un isocontour nul. D'où la pression et la fonction Level-Set ont les mêmes isovaleurs $\phi = p = 0$ sur la surface libre et par suite, la fonction Level-Set ϕ peut être considérée égale à la pression p à une fonction multiplicative près notée c ($\phi = -cp$) dans un voisinage du bord Γ .

La condition aux bords de Neumann non-homogène peut être écrite sur Γ_1 sous la forme suivante :

$$(1.37) \quad \partial_{\mathbf{n}} \phi = -c \partial_{\mathbf{n}} p - p \partial_{\mathbf{n}} c.$$

Mais $p = 0$ sur l'interface et par suite on aura

$$(1.38) \quad \partial_{\mathbf{n}} \phi = -c \partial_{\mathbf{n}} p.$$

Puisque $\|\nabla\phi\| = 1$, on normalise la condition limite et on obtient :

$$(1.39) \quad \partial_{\mathbf{n}}\phi = \frac{\left(\rho g \mathbf{e}_d + \rho((\mathbf{u} \cdot \nabla) \mathbf{u})\right) \cdot \mathbf{n}}{\left\|\left(\rho g \mathbf{e}_d + \rho((\mathbf{u} \cdot \nabla) \mathbf{u})\right) \cdot \mathbf{n}\right\|}.$$

On note par $G(u)$ le second membre de (1.39).

Dans cette section on impose sur l'équation de transport sur Γ_1 la condition aux bords non-homogène qu'on va appeler condition aux bords sous effet gravitationnel :

$$(1.40) \quad \partial_{\mathbf{n}}\phi = G(\mathbf{u}).$$

Nous remarquerons d'après les résultats numériques que les conditions aux limites sous l'effet de la pression donnent une pente physique de l'angle entre la surface libre et l'interface. De plus, elles diminuent remarquablement l'erreur relative de volume.

l'équation (1.5) complétée avec les conditions aux bords sera écrite de la manière suivante :

$$(1.41) \quad \begin{cases} \partial_t \phi + \mathbf{u} \cdot \nabla \phi - \epsilon \Delta \phi = 0 & \text{in } \Omega, \\ \partial_{\mathbf{n}} \phi = G(\mathbf{u}) & \text{sur } \Gamma_1, \\ \partial_{\mathbf{n}} \phi = 0 & \text{sur } \Gamma_0 \cup \Gamma_2, \\ \phi(\mathbf{x}, 0) = \phi_0 & \text{dans } \Omega, \end{cases}$$

où ϕ_0 est la position initiale de l'interface.

La formulation variationnelle discrète devient :

Trouver $(\mathbf{u}_h^{n+1}, p_h^{n+1}) \in X_{u,h} \times M_h$, $\phi_h^{n+1} \in Y_h$ tel que pour tout $(\mathbf{v}_h, q_h) \in X_{v,h} \times M_h$ et $r_h \in Y_h$,

$$(1.42) \quad \begin{cases} \left(\frac{\phi_h^{n+1}}{\Delta t}, r_h\right) + \varepsilon (\nabla \phi_h^{n+1}, \nabla r_h) - \epsilon \int_{\Gamma_1} G(\mathbf{u}_h^n) r_h = \left(\frac{\phi_h^n \circ X^n}{\Delta t}, r_h\right). \\ \left(\rho_h^n \frac{\mathbf{u}_h^{n+1}}{\Delta t}, \mathbf{v}_h\right) + (2\mu_h^n \mathbb{D} \mathbf{u}_h^{n+1}, \mathbb{D} \mathbf{v}_h) - (p_h^{n+1}, \text{div} \mathbf{v}_h) - (q_h, \text{div} \mathbf{u}_h^{n+1}) \\ \quad - \int_{\Gamma_1} \beta^{-1} {}^t \mathbf{u}_h^{n+1} (\mathbf{n} \cdot {}^t \mathbf{n}) \mathbf{v}_h ds - \int_{\Gamma_1} \alpha {}^t \mathbf{u}_h^{n+1} (\boldsymbol{\tau} \cdot {}^t \boldsymbol{\tau}) \mathbf{v}_h ds \\ \quad = (-\rho_h^n g \mathbf{e}_d, \mathbf{v}_h) + \left(\rho_h^n \frac{\mathbf{u}_h^n \circ X^n}{\Delta t}, \mathbf{v}_h\right). \end{cases}$$

1.6.1 Résultats numériques

On considère le même exemple et les mêmes données de la section (1.5.3). Le but est de visualiser la solution du problème d'évolution de l'interface lors du remplissage d'un réservoir avec les conditions aux bords de Navier pour l'écoulement et les conditions aux bords sous effet gravitationnel pour l'équation de transport. Les figures (1.23), (1.24) et (1.25) montrent l'évolution de l'interface dans ce cas à l'instant $t = 0.168\text{ s}$, $t = 0.5\text{ s}$ et $t = 1.3\text{ s}$. On remarque que la condition aux bords de Neumann non-homogène sous l'effet gravitationnel sur Γ_1 améliore l'angle entre l'interface Γ et le bord Γ_1 .

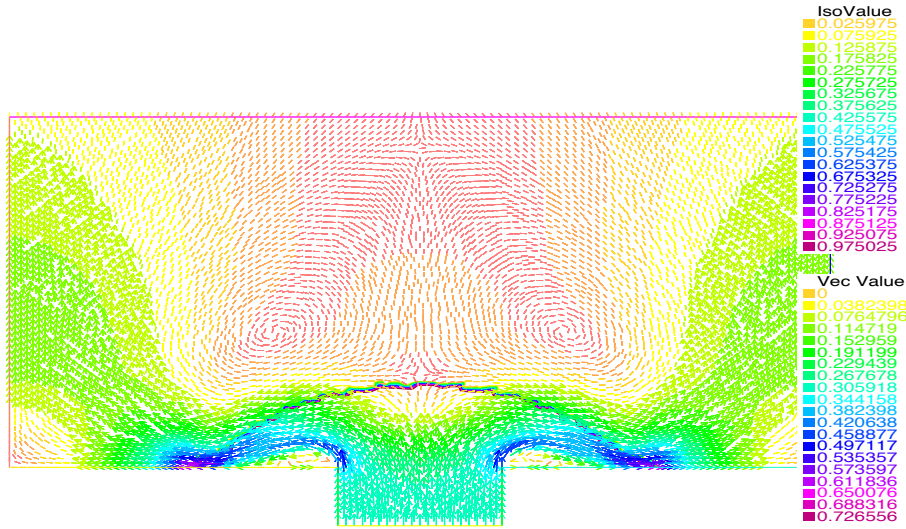


FIGURE 1.23 – L'interface approchée avec les conditions aux bords sous effet gravitationnel pour l'équation de transport à $t = 0.168\text{ s}$.

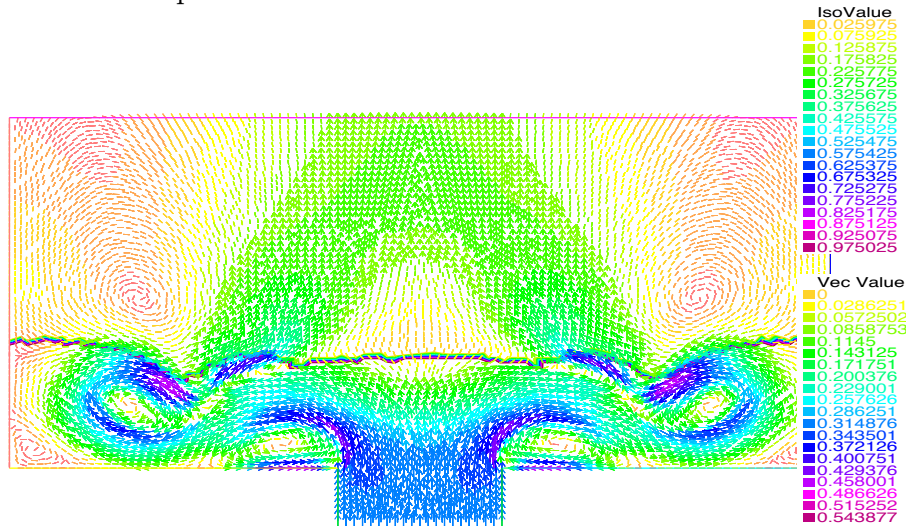


FIGURE 1.24 – L'interface approchée avec les conditions aux bords sous effet gravitationnel pour l'équation de transport à $t = 0.5\text{ s}$.

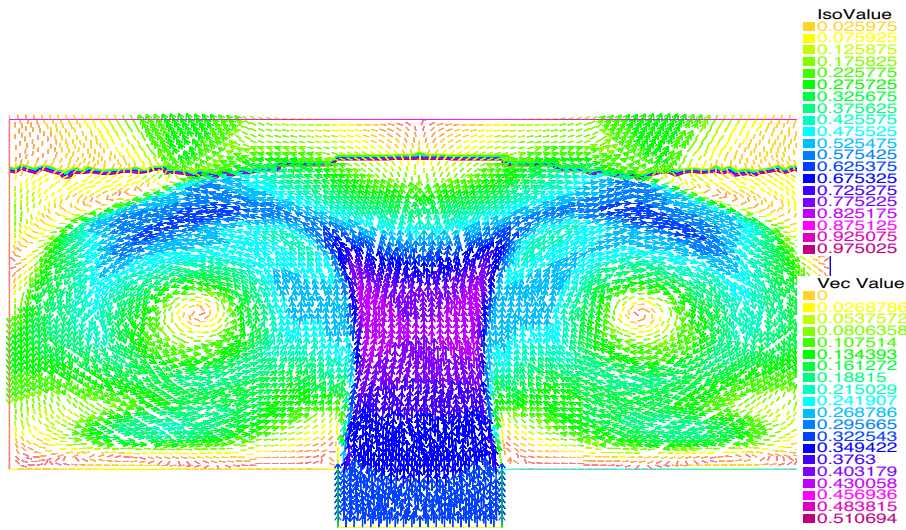


FIGURE 1.25 – L’interface approchée avec les conditions aux bords sous effet gravitationnel pour l’équation de transport à $t = 1.3$ s.

La figure (1.26) représente l’erreur de volume err_V en fonction du temps pour $\epsilon = h/5$ en utilisant les conditions aux bords de Navier et celles sous effet gravitationnel. On remarque que l’erreur en volume encore diminue avec cette nouvelle condition aux bords.

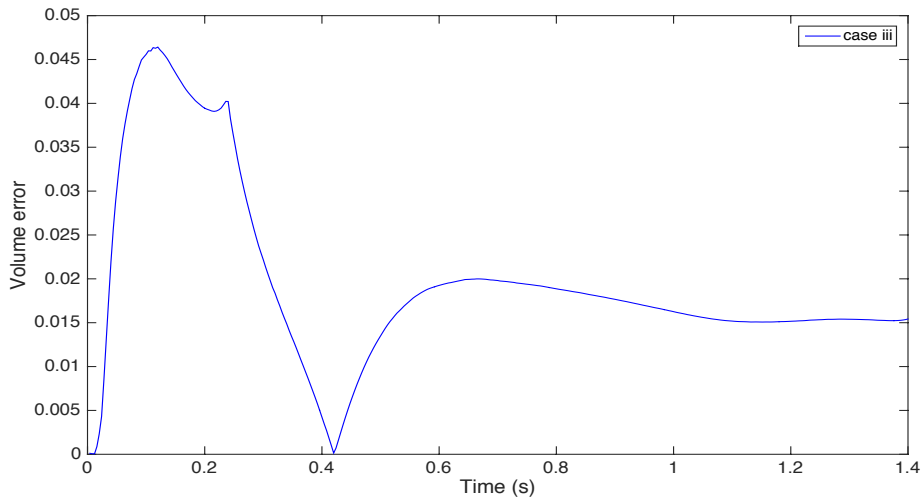


FIGURE 1.26 – Erreur relative du volume pour les conditions aux bords de Navier et celles sous effet gravitationnel.

De plus, pour valider l’hypothèse que la pression totale est nulle sur l’interface qu’on a supposé pour la condition aux bords sous effet gravitationnel. On compare les forces

exercée sur l'interface $\left| 2\mu_1 \int_{\Gamma} \mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n} \right|$ et celle exercée sur l'entrée $\left| 2\mu_1 \int_{\Gamma_0} \mathbf{n}^T \cdot \nabla \mathbf{u} \cdot \mathbf{n} - p \right|$.

Figure (1.27) montre une comparaison entre ces forces durant le temps. On remarque que celles agissantes sur l'interface sont négligeable (la courbe touche à peu près l'axe horizontale) par rapport à celle sur l'entrée. Par suite, la pression sur l'interface est largement plus petite, ce qui valide les hypothèses considérées dans le cas des conditions aux bords sous effet gravitationnel.

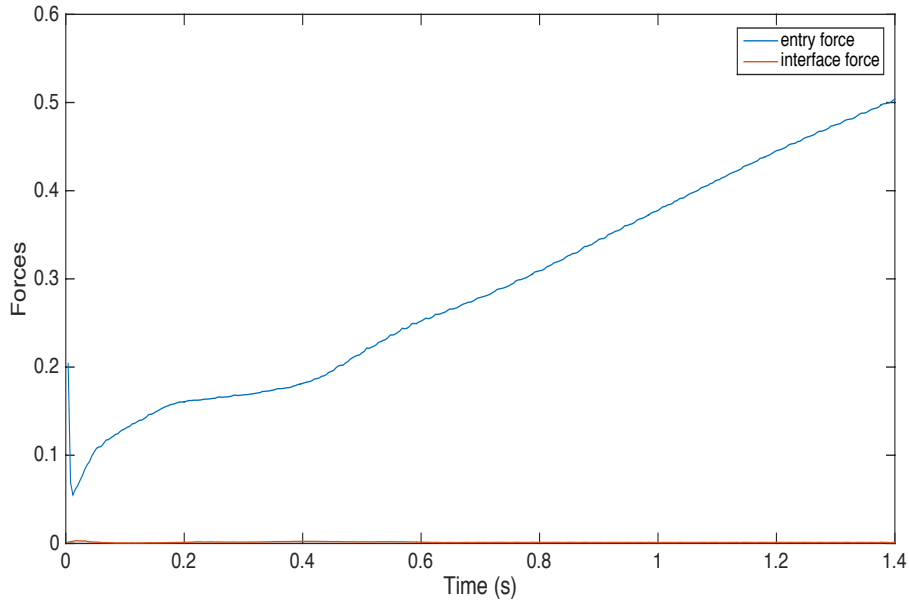


FIGURE 1.27 – Variation des forces exercées sur l'interface et les forces sur l'entrée avec le temps.

1.7 Conservation de masse

La résolution numérique de l'équation de transport de la fonction Level-Set induit une diffusion numérique qui pollue la quantité de masse durant l'évolution du temps. Cette diffusion introduite se traduit par une augmentation ou diminution de l'erreur du volume qui dépend des changements topologiques de l'interface et qui s'accumule avec l'évolution du temps. D'autre part, l'écoulement qu'on a considéré est incompressible et le volume occupé par chacun des fluides doit être conservée. Il existe dans la littérature de nombreuses approches qui peuvent être utilisées afin de préserver la masse, on cite par exemple Chang et al. [26], Sussman and Fatemi [141].

Dans ce travail, on adopte la méthode proposée par Smolianski [140] qui semble simple, pas chère et très efficace dans notre cas. La simplicité de la méthode provient du fait que la conservation de la masse peut être imposée en ajoutant un algorithme simple et facile à implémenter. Mais cet algorithme semble efficace dans le cas où l'erreur du volume induite est faible à chaque pas de temps, pour cela il faut que la méthode utilisée soit avec une précision acceptable même avant l'application de la conservation de la masse. Dans notre cas, on a pu réduire remarquablement l'erreur de volume en utilisant la condition aux bords sous l'effet de la pression, et par suite l'algorithme de la conservation de la masse a montré une efficacité remarquable dans nos applications.

Le concept de la méthode consiste à faire déplacer le zéro-isocontour à chaque pas de temps en déplaçant la fonction level-set Φ correspondante en lui ajoutant à une constante signée c_Φ qui n'est autre que la distance entre l'ancienne et la nouvelle ligne de niveau zéro telle que la nouvelle fonction Level-set Φ_{new} réduit l'erreur de la masse correspondante. On définit alors un nouveau domaine Ω_2^{new} donné par

$$\Omega_2^{new} = \{\mathbf{x} \in \Omega : \Phi_{new} > 0\}.$$

L'expression de c_Φ est donnée par la formule :

$$\begin{aligned} S_{exact} - S(\Omega_2) &= \int_{\Omega_2^{new}} dx - \int_{\Omega_2} dx \\ (1.43) \quad &= \int_{\Gamma} (c_\Phi \mathbf{n}) \cdot \mathbf{n} d\Gamma + O(c_\Phi^2) \\ &= c_\Phi \int_{\Gamma} d\Gamma + O(c_\Phi^2), \end{aligned}$$

où S_{exact} est l'aire exacte (ou volume en 3D) de la region occupée par le second fluide, $S(\Omega_2)$ est l'aire numérique de Ω_2 . En notant $L = \int_{\Gamma} d\Gamma$, on approche c_Φ par

$$(1.44) \quad c_\Phi \simeq \frac{S_{exact} - S(\Omega_2)}{L(\Gamma)}.$$

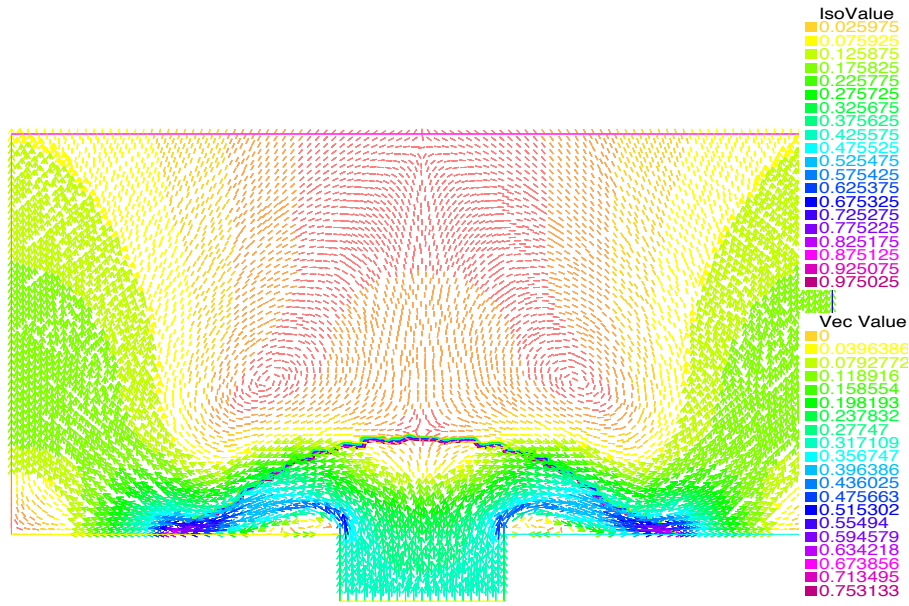
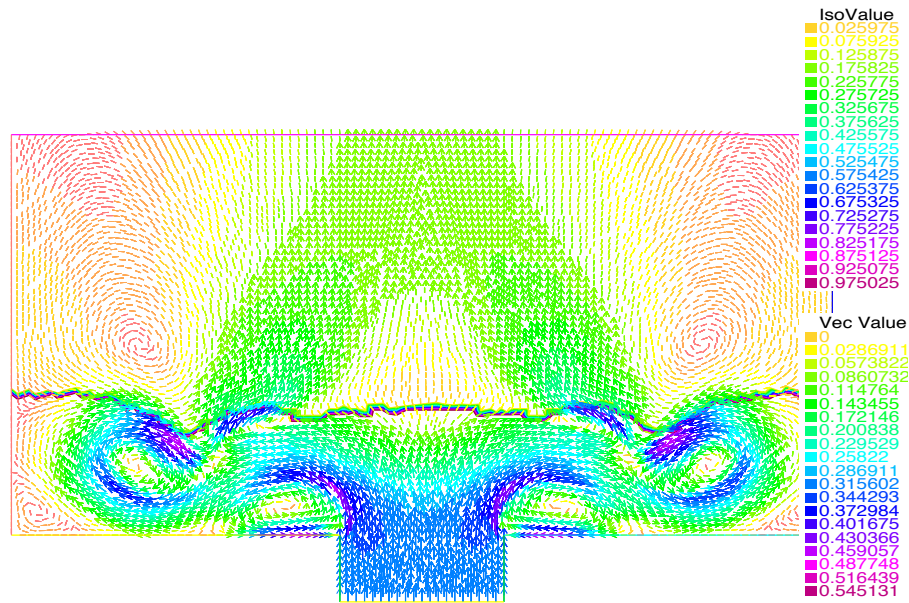
Alors la fonction Level-Set corrigée devient

$$(1.45) \quad \Phi_{new} \simeq \Phi + c_\Phi ||\nabla \Phi||.$$

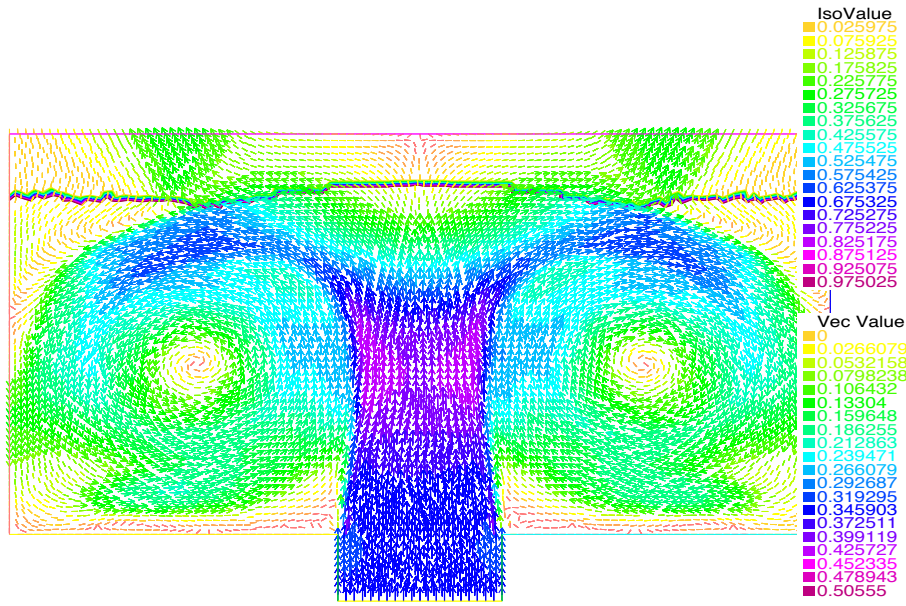
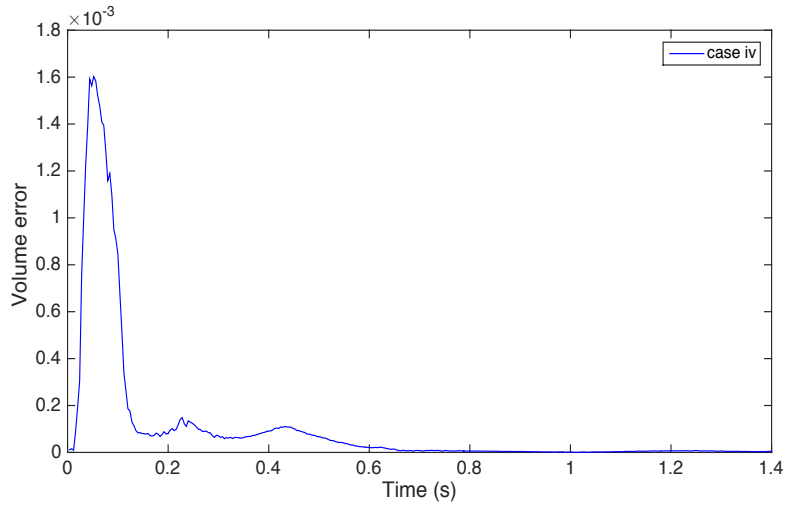
La précision de la formule (1.44) est de l'ordre $O(c_\Phi^2)$. Tout d'abord, il est à noter que si $S(\Omega_2) > S_{exact}$, on a $c_\Phi < 0$ et la fonction Level-Set Φ sera déplacée vers le bas. Si $S(\Omega_2) < S_{exact}$, on a $c_\Phi > 0$ et la fonction Level-Set Φ sera déplacée vers le haut.

1.7.1 Résultats numériques

On considère le même exemple et les mêmes données et les mêmes conditions aux bords de la section (1.6.1). Les figures (1.28), (1.29) et (1.30) montrent l'évolution de l'interface à l'instant $t = 0.168$ s, $t = 0.5$ s et $t = 1.3$ s avec l'algorithme de conservation de masse.

FIGURE 1.28 – L'interface approchée à $t = 0.168$ s avec la conservation de masseFIGURE 1.29 – L'interface approchée à $t = 0.5$ s avec la conservation de masse

La figure (1.31) montre l'erreur err_V en fonction du temps pour $\epsilon = h/5$ avec les conditions aux bords de Navier et celles sous effet gravitationnel après la conservation de masse. On remarque une nette diminution de l'erreur relative du volume avec la conservation de la masse par rapport aux cas d'avant, d'où l'avantage de cette technique.

FIGURE 1.30 – L'interface approchée à $t = 1.3$ s avec la conservation de masseFIGURE 1.31 – Erreur relative du volume pour $\epsilon = h/5$ avec les conditions aux bords de Navier et de Neumann non-homogène après la conservation de masse.

Conclusion On a construit dans ce chapitre un algorithme pour modéliser le remplissage d'un moule en 2D. On a essayé d'étendre le problème en 3D pour des moules de moyennes tailles, la mémoire est insuffisante pour résoudre le problème de Navier-Stokes. Pour réduire ce coût de calcul, on va utiliser la méthode de projection dans le chapitre suivant.

Chapitre 2

Méthode de projection

Sommaire

2.1	Motivation	57
2.2	Méthode de projection avec les conditions aux bords de Dirichlet	57
2.2.1	Présentation de la méthode	57
2.2.2	Formulation variationnelle et discrétisation	59
2.3	Méthode de projection avec les conditions aux bords de Navier	60
2.4	Résultats numériques	61
2.4.1	Dimension deux	61
2.4.2	Problème trois dimensionnel : exemple du cube	64
2.4.3	Cas industriel (3D)	68

2.1 Motivation

La méthode de projection est connue comme étant une des plus économiques tant d'un point de vue calcul que mémoire, c'est pour cela que nous allons l'utiliser dans la suite. Cette méthode a été introduite par Chorin [29], [30] et elle a été développée par Temam [143], [144]. Son avantage est qu'on résoud, à chaque pas de temps, une suite d'équations elliptiques pour la vitesse et la pression. On aboutit ainsi à des matrices découplées et donc plus petites que celles du chapitre précédant, ce qui rend les simulations numériques beaucoup plus rapides (surtout en trois dimensions).

On effectue la discrétisation en temps en utilisant toujours la méthode des caractéristiques (Achdou et Guermond [2]). Pour la discrétisation en espace, on peut utiliser soit les éléments finis $P_b - P_1$ pour la vitesse et la pression pour la discrétisation de l'équation de Navier-Stokes. On peut également utiliser des éléments finis de même ordre pour la vitesse et la pression (par exemple $P_1 - P_1$) qui réduit significativement le nombre de degré de liberté et qui rend le problème moins coûteux surtout en dimension 3 ; mais ce choix ne satisfait pas la condition de stabilité inf-sup et cause quelques difficultés numériques. Dans la littérature on trouve plusieurs approches de stabilisation qui compense la conditions inf-sup dont on peut citer par exemple [19], [17], [82] et [134]).

La mise en oeuvre du schéma est plus difficile dans le cas d'un fluide biphasique en raison de la discontinuité du saut des coefficients de densité et de viscosité.

Comme on l'a noté dans le chapitre précédent, l'utilisation de la méthode de caractéristiques pour la discrétisation des équations de Navier-Stokes se résume en deux étapes : la première est la résolution d'une équation de type advection pour déterminer la trajectoire des particules du fluide puis la résolution d'équations de type Stokes généralisées. Alors on écrit la première équation du problème en coordonnées Lagrangiennes sous la forme suivante :

$$(2.1) \quad \rho \frac{D\mathbf{u}}{Dt} - \operatorname{div}(2\mu\mathbb{D}\mathbf{u}) + \nabla p = \rho\mathbf{g}.$$

2.2 Méthode de projection avec les conditions aux bords de Dirichlet

2.2.1 Présentation de la méthode

L'algorithme de la méthode de projection de Chorin/Temam est basé sur la décomposition du champ de vecteur en un vecteur à divergence nulle et un autre irrotationnel. De plus, cet algorithme suppose que $\frac{\partial p}{\partial \mathbf{n}} = 0$. Typiquement, l'algorithme se décompose

en trois étapes : à chaque pas de temps, la première étape consiste à calculer une vitesse intermédiaire qui ne satisfait pas la contrainte d'incompressibilité ; dans la deuxième étape, on utilise la vitesse intermédiaire déjà calculée qu'on projète dans l'ensemble des champs à divergence nulle pour obtenir la valeur de la pression solution du problème ; la troisième étape consiste à construire la vitesse finale à partir des résultats obtenues. L'algorithme se résume de la manière suivante : Ayant $\mathbf{u}^n$, on calcule $(\mathbf{u}^{n+1}, p^{n+1})$ tel que :

Etape1- Calcul de la vitesse intermédiaire $\mathbf{u}^*$:

$$(2.2) \quad \left\{ \begin{array}{ll} \rho^n \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} - \operatorname{div}(2\mu^n \mathbb{D}\mathbf{u}^*) + \nabla p^n = \rho^n \mathbf{g} & \text{dans } \Omega, \\ - [2\mu^n \mathbb{D}\mathbf{u}^* \cdot \mathbf{n}] = 0 & \text{sur } \Gamma, \\ \mathbf{u}^* = U_{in} & \text{sur } \Gamma_0, \\ \mathbf{u}^* = 0 & \text{sur } \Gamma_1, \\ -\mu^n \mathbb{D}\mathbf{u}^* \cdot \mathbf{n} = 0 & \text{sur } \Gamma_2. \end{array} \right.$$

Etape 2- Calcul de la pression p^{n+1} :

$$(2.3) \quad \left\{ \begin{array}{ll} \operatorname{div} \mathbf{u}^* - \operatorname{div} \left(\frac{\Delta t}{\rho^n} \nabla (p^{n+1} - p^n) \right) = 0 & \text{in } \Omega, \\ [p^{n+1} - p^n] = 0 & \text{on } \Gamma, \\ \frac{\partial}{\partial \mathbf{n}} (p^{n+1} - p^n) = 0 & \text{on } \Gamma_0 \cup \Gamma_1, \\ p^{n+1} - p^n = 0 & \text{on } \Gamma_2. \end{array} \right.$$

Etape 3- Calcul de la vitesse finale $\mathbf{u}^{n+1}$:

$$(2.4) \quad \mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^n} \nabla (p^{n+1} - p^n) \text{ dans } \Omega.$$

En fait, La décomposition de l'opérateur implique une décomposition des conditions sur la frontière et des conditions de saut sur l'interface.

La nature de projection du système apparaît dans le deuxième sous-problème : la vitesse intermédiaire $\mathbf{u}^*$ n'est pas à divergence nulle, et la deuxième étape représente une projection L^2 de $\mathbf{u}^*$ sur l'ensemble des fonctions à divergence nulle (Voir Bell et Marcus [11]). La pression résultante doit être considérée dans $M = \{q \in H^1(\Omega) / q|_{\Gamma_2} = 0\}$ et non pas dans $L_0^2(\Omega)$, et la vitesse finale à chercher est à divergence nulle dont la composante normale est nulle, elle appartient à l'espace $H_{\operatorname{div}}(\Omega) = \{\mathbf{v} \in L^2(\Omega) / \operatorname{div} \mathbf{v} = 0 \text{ dans } \Omega, \mathbf{v} \cdot \mathbf{n} = 0 \text{ sur } \Gamma_1 \cup \Gamma_0\}$ et non pas à $H^1(\Omega)$.

Le problème (2.3) s'obtient à partir de

$$(2.5) \quad \frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} + \frac{1}{\rho^n} \nabla(p^{n+1} - p^n) = 0 \quad \text{dans } \Omega.$$

en appliquant l'opérateur de divergence et en utilisant la condition d'incompressibilité. Alors que la condition au bord de Neumann homogène s'obtient en prenant sa composante normale et en utilisant le fait que $\mathbf{u}^{n+1} \cdot \mathbf{n} = \mathbf{u}^* \cdot \mathbf{n}$ sur Γ_0 et Γ_1 . Cette condition aux bords de Neumann homogène n'est pas physique et la pression exacte ne la satisfait pas mais elle peut être une solution approchée raisonnable pour beaucoup des situations pratiques (voir Gresho et Sani [63], Orszag et al. [106] et Shen et Temam [132]).

Dans cette méthode, on perd la composante tangentielle de la vitesse sur Γ_0 et Γ_1 , pour la récupérer, la troisième étape peut être réécrite sous la forme suivante :

$$(2.6) \quad \begin{cases} \mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^n} \nabla(p^{n+1} - p^n) & \text{dans } \Omega. \\ \mathbf{u}^{n+1} = U_{in} & \text{sur } \Gamma_0, \\ \mathbf{u}^{n+1} = 0 & \text{sur } \Gamma_1. \end{cases}$$

2.2.2 Formulation variationnelle et discrétisation

La méthode de projection introduite dans la section précédente, peut se mettre sous une formulation variationnelle. Pour cela, on introduit les espaces fonctionnels suivants :

$$X_{U_{in},0} = \{\mathbf{v} \in H^1(\Omega) / \mathbf{u} = U_{in} \text{ sur } \Gamma_0 \text{ et } \mathbf{u} = 0 \text{ sur } \Gamma_1\},$$

$$X = H_0^1(\Omega) = \{\mathbf{v} \in H^1(\Omega) / \mathbf{v} = 0 \text{ sur } \Gamma_0 \cup \Gamma_1\}$$

et

$$M = \{q \in H^1(\Omega) / q|_{\Gamma_2} = 0\}.$$

Pour la discrétisation en espace, on choisit les éléments finis continus pour la vitesse et la pression et on introduit les espaces discrets suivants :

$$X_{h,U_{in},0} = \{\mathbf{v}_h \in X_{U_{in},0}; \forall k \in T_h; \mathbf{v}_{h|k} \in P_b(k)^2\},$$

$$X_h = \{\mathbf{v}_h \in X; \forall k \in T_h; \mathbf{v}_{h|k} \in P_b(k)^2\},$$

$$M_h = \{q_h \in M \cap C^0(\Omega); \forall k \in T_h, q_{h|k} \in P_1(k)\}$$

et

$$L_h = \{\mathbf{u}_h \in L^2(\Omega); \forall k \in T_h; \mathbf{u}_{h|k} \in P_1(k)^2\}.$$

La formulation variationnelle associée à la méthode de projection s'écrit :

1- Chercher $\mathbf{u}^* \in X_{U_{in},0}$ tel que pour tout $\mathbf{v} \in X$, on a :

$$(2.7) \quad \left(\rho^n \frac{\mathbf{u}^* - \mathbf{u}^n o X^n}{\Delta t}, \mathbf{v} \right) + (2\mu^n \mathbb{D}\mathbf{u}^*, \mathbb{D}\mathbf{v}) + (\nabla p^n, \mathbf{v}) = (\rho^n \mathbf{g}, \mathbf{v}).$$

2- Chercher $p^{n+1} \in M$ tel que pour tout $q \in M$, on a :

$$(2.8) \quad (\operatorname{div} \mathbf{u}^*, q) + \left(\frac{\Delta t}{\rho^n} \nabla(p^{n+1} - p^n), \nabla q \right) = 0.$$

3- Chercher $\mathbf{u}^{n+1} \in L^2(\Omega)$ tel que pour tout $\mathbf{v} \in L^2(\Omega)$, on a :

$$(2.9) \quad (\mathbf{u}^{n+1}, \mathbf{v}) = (\mathbf{u}^*, \mathbf{v}) - \left(\frac{\Delta t}{\rho^n} \nabla(p^{n+1} - p^n), \mathbf{v} \right).$$

La formulation variationnelle discrète associée s'écrit :

1- Chercher $\mathbf{u}_h^* \in X_{h,U_{in},0}$ tel que pour tout $\mathbf{v}_h \in X_h$, on a :

$$(2.10) \quad \left(\rho^n \frac{\mathbf{u}_h^* - \mathbf{u}_h^n o X_h^n}{\Delta t}, \mathbf{v}_h \right) + (2\mu^n \mathbb{D}\mathbf{u}_h^*, \mathbb{D}\mathbf{v}_h) + (\nabla p_h^n, \mathbf{v}_h) = (\rho^n \mathbf{g}, \mathbf{v}_h).$$

2- Chercher $p_h^{n+1} \in M_h$ tel que pour tout $q_h \in M_h$, on a :

$$(2.11) \quad (\operatorname{div} \mathbf{u}_h^*, q_h) + \left(\frac{\Delta t}{\rho^n} \nabla(p_h^{n+1} - p_h^n), \nabla q_h \right) = 0.$$

3- Chercher $\mathbf{u}_h^{n+1} \in L_h$ tel que pour tout $\mathbf{v}_h \in L_h$, on a :

$$(2.12) \quad (\mathbf{u}_h^{n+1}, \mathbf{v}_h) = (\mathbf{u}_h^*, \mathbf{v}_h) - \left(\frac{\Delta t}{\rho^n} \nabla(p_h^{n+1} - p_h^n), \mathbf{v}_h \right).$$

Les systèmes algébriques linéaires résultants de la discrétisation par élément finis des deux premiers sous- problèmes sont des matrices symétriques définies positives qui incorporent le coefficient de diffusion discontinue $\frac{1}{\rho^n}$. La méthode des gradients conjugués préconditionnés par la décomposition de Cholesky incomplète est utilisée pour la résolution des systèmes de la vitesse et la pression.

Il existe dans la littérature plusieurs variantes de la méthode de projection et on peut se référer à [69] pour plus de détails.

2.3 Méthode de projection avec les conditions aux bords de Navier

Dans cette section, on remplace les conditions aux bords de Dirichlet par les conditions aux bords de Navier suivant :

$$(2.13) \quad \begin{cases} \frac{1}{\beta} \mathbf{u} \cdot \mathbf{n} + {}^t \mathbf{n} (2\mu \mathbb{D}\mathbf{u}) \cdot \mathbf{n} = 0 \\ \alpha \mathbf{u} \cdot \boldsymbol{\tau} + {}^t \mathbf{n} (2\mu \mathbb{D}\mathbf{u}) \cdot \boldsymbol{\tau} = 0. \end{cases}$$

Au niveau du problème (2.2), la différence de ces conditions de Navier avec celles introduites dans le premier chapitre (1.30), réside dans l'absence du terme de la pression qui ne figure pas dans (2.2).

Dans ce cas, on couple les équations des deux composantes de la vitesse et on effectue l'implémentation numérique du premier sous problème (2.2).

Le deuxième sous-problème (2.3) se traite exactement comme dans la section précédente et le troisième sous problème (2.5) sera remplacé par le suivant :

$$(2.14) \quad \left\{ \begin{array}{ll} -\varepsilon \Delta \mathbf{u}^{n+1} + \mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho^n} \nabla (p^{n+1} - p^n) & \text{dans } \Omega, \\ \mathbf{u}^{n+1} = U_{in} & \text{sur } \Gamma_0, \\ \frac{1}{\beta} \mathbf{u}^{n+1} \cdot \mathbf{n} + {}^t \mathbf{n} (2\mu^n \mathbb{D} \mathbf{u}^{n+1}) \cdot \mathbf{n} = 0 & \text{sur } \Gamma_1, \\ \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} = 0 & \text{sur } \Gamma_2, \end{array} \right.$$

où ε est un paramètre de pénalisation assez petit.

2.4 Résultats numériques

2.4.1 Dimension deux

On reprend l'exemple du premier chapitre, on fait les simulations cette fois avec la méthode de projection pour les équations de Navier-Stokes.

La densité de l'eau est 1 Kg.dm^{-3} , sa viscosité cinématique est $10^{-4} \text{ dm}^2.\text{s}^{-1}$ et sa viscosité dynamique est $10^{-4} \text{ Kg.dm}^{-1}.\text{s}^{-1}$; tandis que la densité de l'air est $10^{-3} \text{ Kg.dm}^{-3}$, sa viscosité cinématique est $10^{-3} \text{ dm}^2.\text{s}^{-1}$ et sa viscosité dynamique est $10^{-6} \text{ Kg.dm}^{-1}.\text{s}^{-1}$. L'accélération gravitationnelle est $g = 0.98 \text{ dm.s}^{-2}$. De plus, on choisit $U_{in} = 0.4 \text{ dm.s}^{-1}$, $h = 0.02$, $\Delta t = 2h$.

On présente les résultats graphiques pour chacun des cas :

- Cas *i* : Solution avec les conditions aux bords de type Dirichlet pour la vitesse et Neumann homogène pour la fonction Level-Set avant la conservation de masse.
- Cas *ii* : Solution avec les conditions aux bords de type Navier pour la vitesse et Neumann homogène pour la fonction Level-Set avant la conservation de masse.
- Cas *iii* : Solution avec les conditions aux bords de type Navier pour la vitesse et Neumann non-homogène pour la fonction Level-Set avant la conservation de masse.
- Cas *iv* : Solution avec les conditions aux bords de type Navier pour la vitesse et Neumann non-homogène pour la fonction Level-Set après la conservation de masse.

Les figures (2.1), (2.2) et (2.3) montrent les résultats numériques pour les 4 cas considérés à $t = 0.168 s$, $t = 0.5 s$ et $t = 1.3 s$. On remarque que les résultats sont graphiquement similaires à ceux obtenus par la méthode classique.

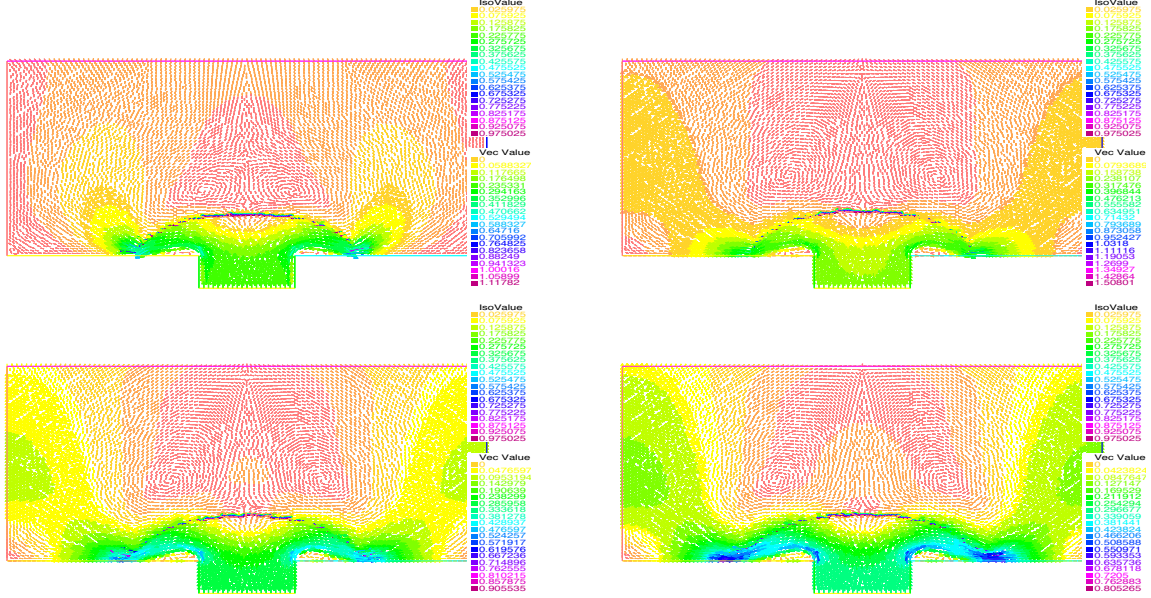


FIGURE 2.1 – A $t = 0.168 s$. En haut : De gauche à droite : cas i, cas ii.
En bas : De gauche à droite : cas iii, cas iv.

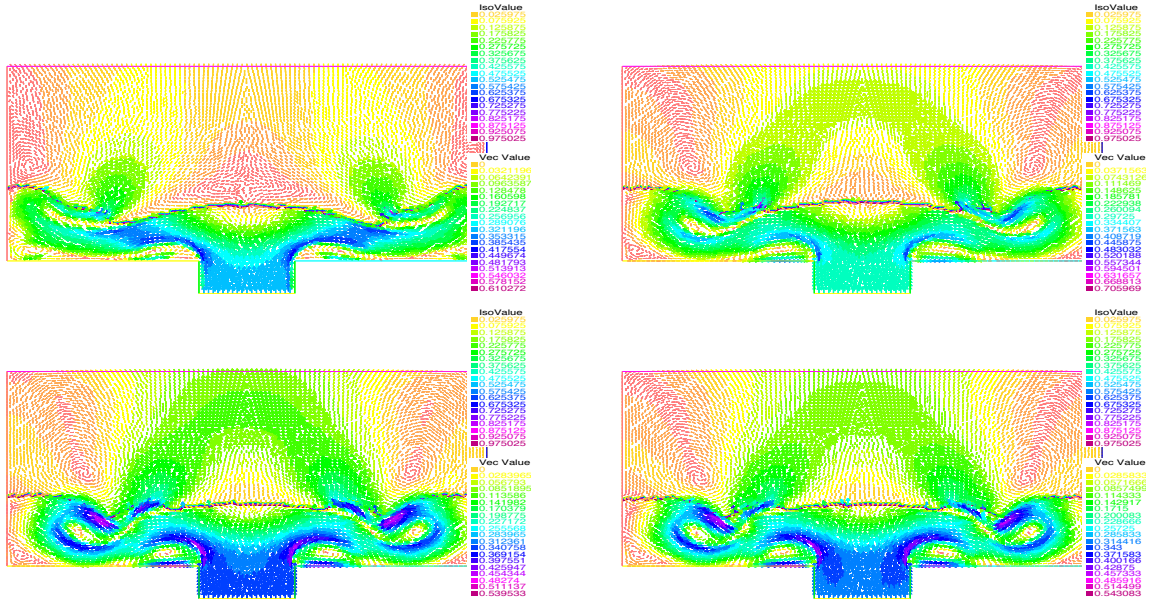


FIGURE 2.2 – A $t = 0.5 s$. En haut : De gauche à droite : cas i, cas ii.
En bas : De gauche à droite : cas iii, cas iv.

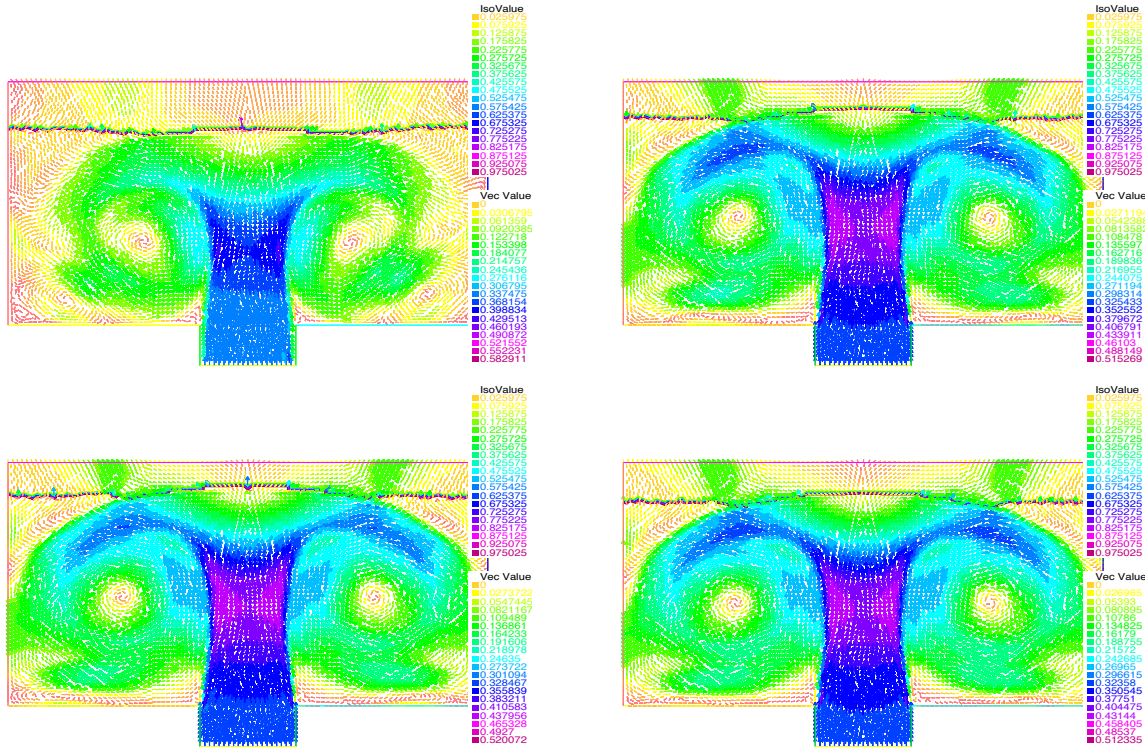


FIGURE 2.3 – A $t = 1.3$ s. En haut : De gauche à droite : cas i, cas ii.
En bas : De gauche à droite : cas iii, cas iv.

On trace l'erreur err_V en fonction du temps pour dans chacun de ces cas.

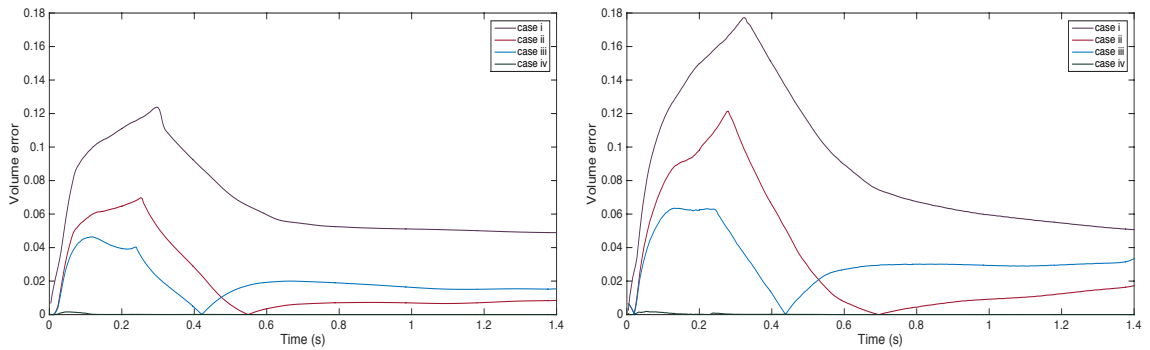


FIGURE 2.4 – Comparaison de l'erreur en volume pour les cas i , ii , iii and iv . A gauche : méthode classique ; A droite : méthode de projection

Figure (2.4) montre une comparaison entre les erreurs en volume pour la méthode classique et celle de projection. L'erreur en volume obtenu en utilisant la méthode classique est

légèrement plus petit que celui obtenu par la méthode de projection pour les trois premier cas. Cette petite différence est dû à la condition aux bords non-physique sur la pression. Après la conservation de masse, on obtient à peu près le même résultat entre les deux méthodes. De plus, le temps de calcul et la taille en mémoire utilisée dans la méthode de projection sont largement plus petits.

Remarque 2.4.1. *Pour les simulations séquentiel, on a utilisé la machine hpc2 "seq" du laboratoire Jacques Louis-Lions avec 40Gb de mémoire et de temps maximale 100 heures.*

Dans la figure (2.5), on compare le temps de calcul pour résoudre l'algorithme avec la méthode classique et la méthode de projection.

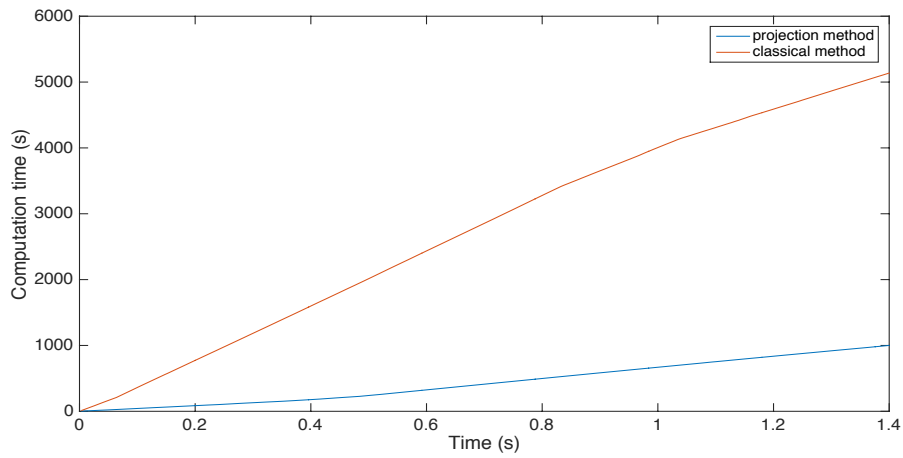


FIGURE 2.5 – Graphe de comparaison du temps calcul entre les méthodes classique et projection.

On peut remarquer un gain significatif pour la méthode de projection en deux dimensions ce qui va augmenter remarquablement dans le cas trois dimensionnel.

2.4.2 Problème trois dimensionnel : exemple du cube

Dans cette section, le domaine Ω est un parallélépipède, avec une base rectangulaire de dimension $a = 1 \text{ dm}$, $b = 1 \text{ dm}$ et de hauteur $z = 1 \text{ dm}$ centrée avec un petit trou au fond de dimensions $a_1 = b_1 = 0.4 \text{ dm}$, $e = .1 \text{ dm}$ (Voir figure 2.6).

La densité de l'eau est 1 Kg.dm^{-3} , sa viscosité cinématique est $10^{-4} \text{ dm}^2.\text{s}^{-1}$ et sa viscosité dynamique est $10^{-4} \text{ Kg.dm}^{-1}.\text{s}^{-1}$; tandis que la densité du gaz est $10^{-3} \text{ Kg.dm}^{-3}$, sa viscosité cinématique est $10^{-3} \text{ dm}^2.\text{s}^{-1}$ et sa viscosité dynamique est $10^{-6} \text{ Kg.dm}^{-1}.\text{s}^{-1}$. L'accélération gravitationnelle est $g = 0.98 \text{ dm.s}^{-2}$. De plus, on choisit $U_{in} = 0.5 \text{ dm.s}^{-1}$, $h = 0.02$, $\Delta t = 2h$ et $\epsilon = h/5$.

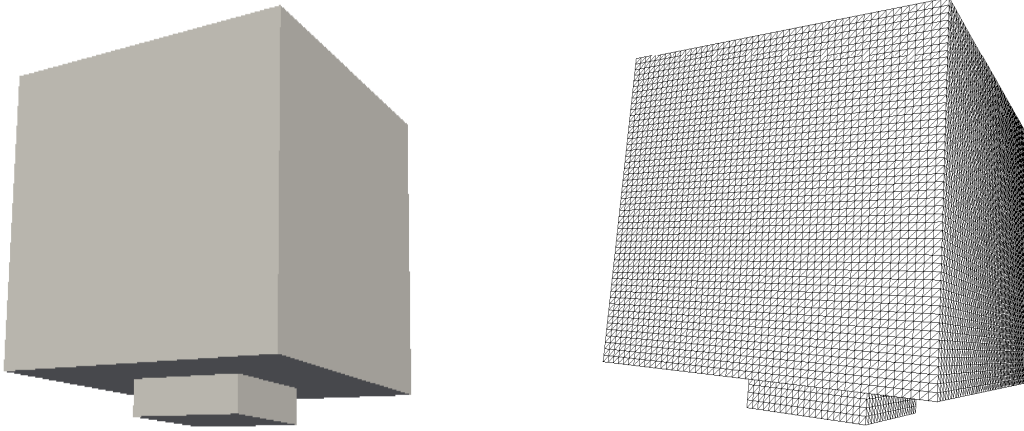


FIGURE 2.6 – A gauche : La moule cubique. A droite : Le maillage du cube.

A l'instant initiale, l'interface Γ est représentée par la fonction Level-Set d'équation $\phi_0 = z - 0.06$.

Les figures (2.7)-(2.8) et (2.9) nous montrent l'évolution de l'interface aux instants $t = 0$ s, $t = 0.12$ s, $t = 0.2$ s, $t = 0.4$ s, $t = 0.6$ s, et $t = 0.8$ s.

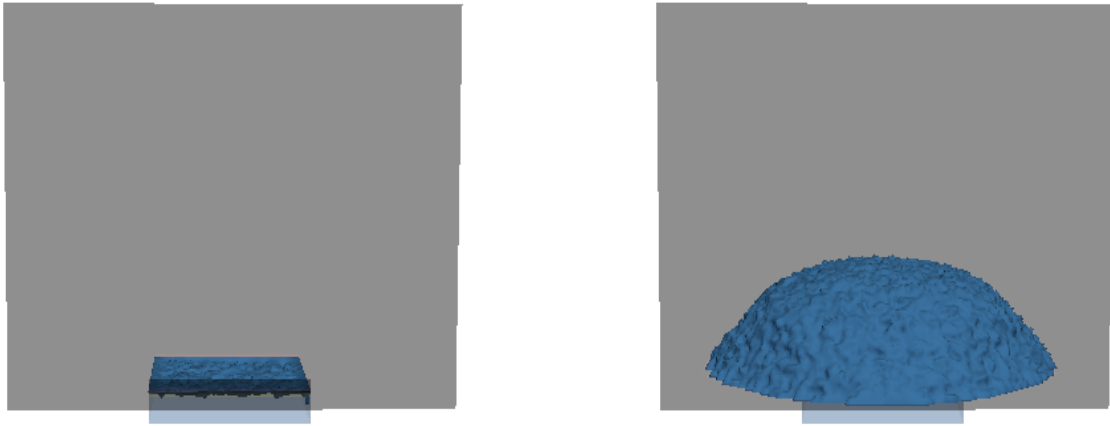


FIGURE 2.7 – L'interface approchée dans le problème trois dimensionnel à $t = 0$, $t = 0.12$ s.

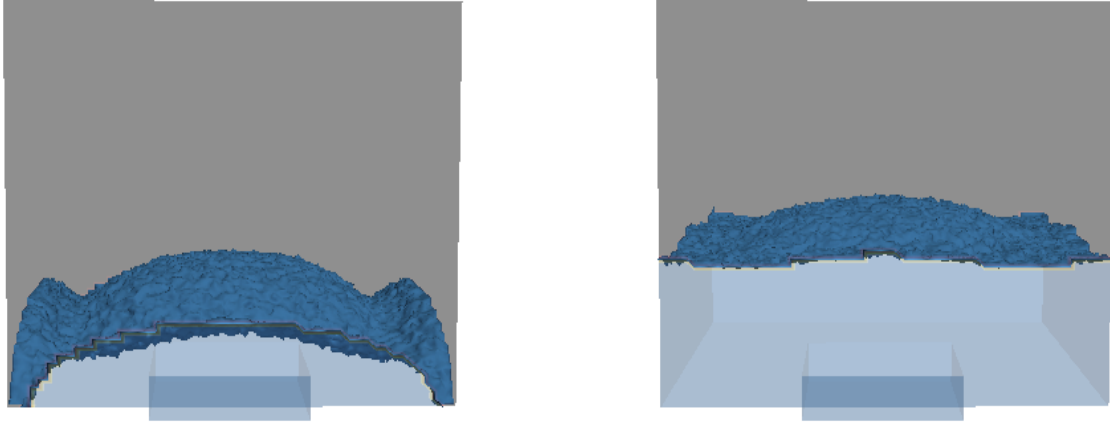


FIGURE 2.8 – L’interface approchée dans le problème trois dimensionnel à $t = 0.2 s$, $t = 0.4 s$.

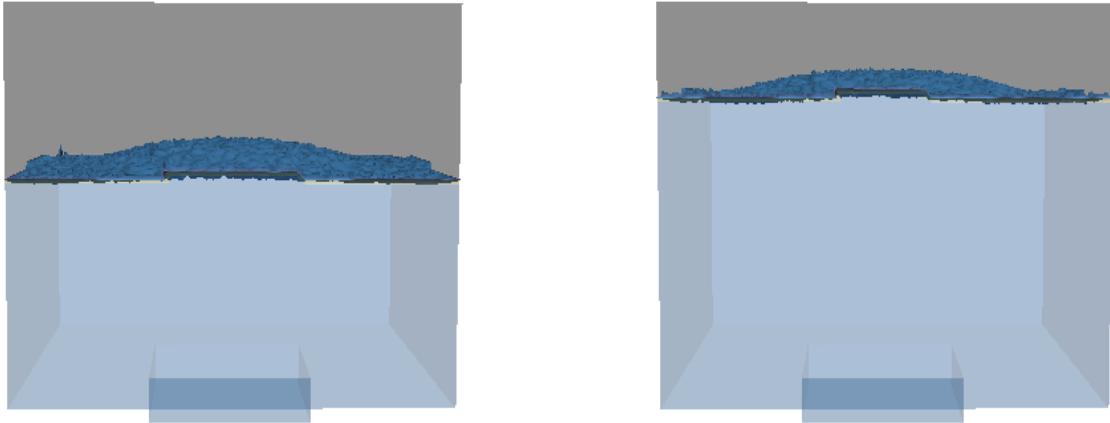


FIGURE 2.9 – L’interface approchée dans le problème trois dimensionnel à $t = 0.6 s$ and $t = 0.8 s$.

La figure (2.10) nous montre l’évolution d’erreur en volume en fonction du temps pour les cas i , ii , iii et iv . On peut voir que l’erreur en volume change de la même manière comme dans le cas 2D mais elle atteint un maximum de 27% pour le cas i , elle diminue dans le cas ii à un maximum de 14% et avec le cas iii à 9%. Après la conservation de masse, l’erreur en volume est moins que 0.1% durant toute la simulation correspondante.

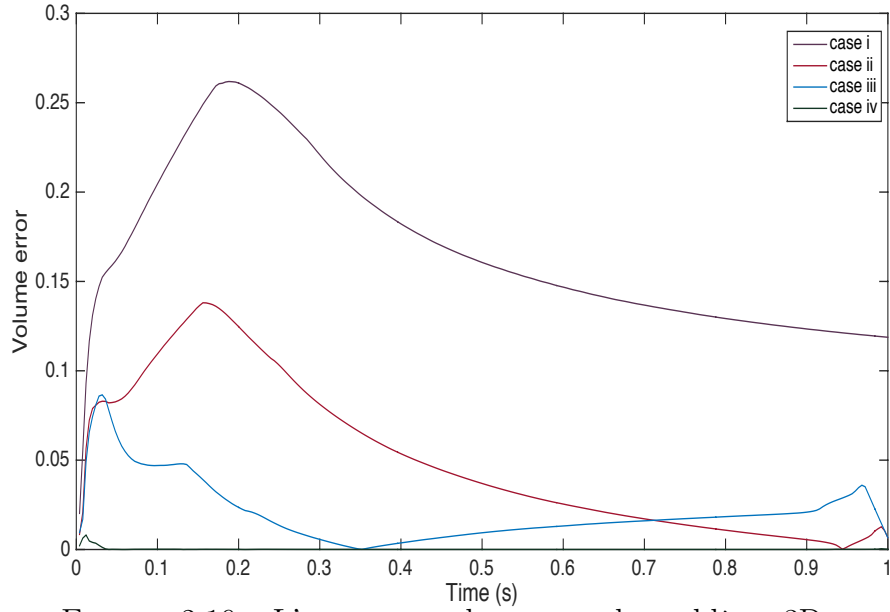


FIGURE 2.10 – L'erreur en volume pour le problème 3D.

La figure (2.11) montre que les forces exercées sur l'interface sont négligeables par rapport à celles agissant sur l'entrée dans les problèmes trois dimensionnels, par suite la pression disparaît sur l'interface.

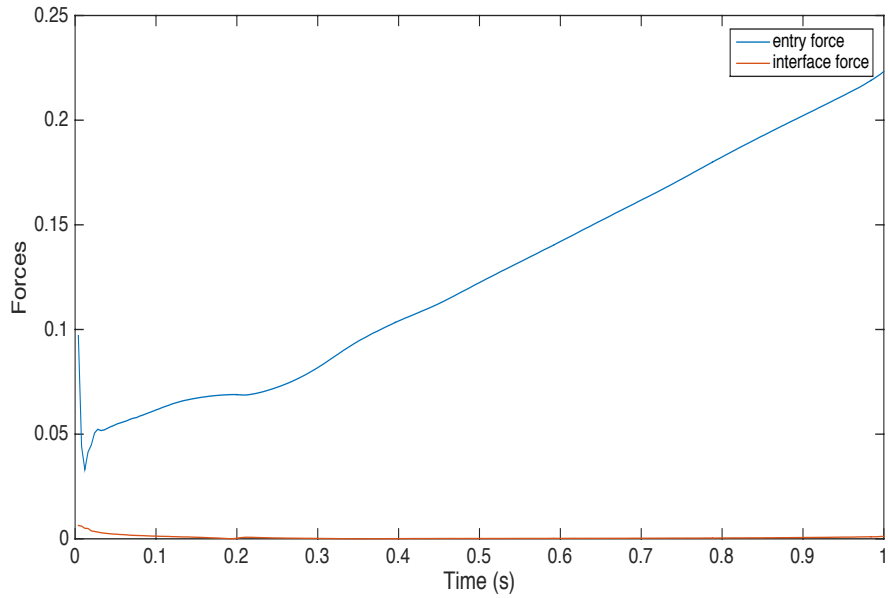


FIGURE 2.11 – Variation de la force exercée sur l'interface et la force sur l'entrée.

2.4.3 Cas industriel (3D)

Dans cette section, on modélise le remplissage d'un moule. Le domaine Ω est un moule industriel indiqué par la figure (2.12).

La densité de l'air considérée est $1Kg.m^{-3}$, la viscosité cinématique est $1.818e^{-3}m^2.s^{-1}$ et la viscosité dynamique est $1.818e^{-3}Kg.m^{-1}.s^{-1}$; tandis que la densité de l'aluminium liquide est $2385Kg.m^{-3}$, la viscosité cinématique est $1e^{-4}m^2.s^{-1}$ et la viscosité dynamique est $2.385e^{-5}Kg.m^{-1}.s^{-1}$.

La hauteur maximale de ce moule est $1.6 m$, la longueur maximale est de $1 m$ et la largeur maximale est de $0.8 m$. De plus, on considère les données suivante : $U_{in} = 0.5 m/s$, $h = 0.02$, $\Delta t = 2h$ et $\epsilon = h/5$.

Le maillage considéré contient 59364 sommets et 294544 tétraèdres.

Le nombre des degrés de liberté est de 353,908 pour la vitesse, 59,364 pour la pression et la fonction Level-Set.

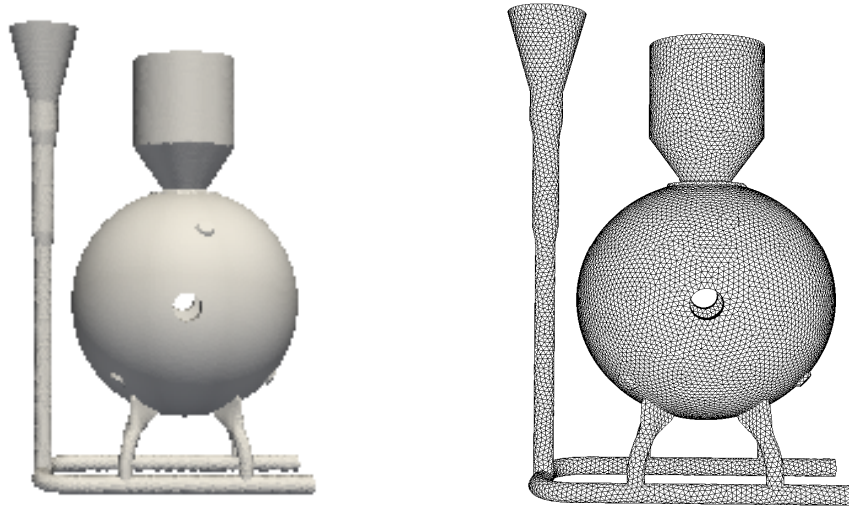


FIGURE 2.12 – A gauche : La moule de l'industrie; A droite : Le maillage correspondant.

A l'instant initiale, l'interface Γ est représentée par la fonction Level-Set d'équation $\phi_0 = -z + 0.9$.

Les figures (2.13), (2.14) et (2.15) nous montrent l'évolution de l'interface pour le problème industriel aux instants $t = 0$, $t = 0.4 s$, $t = 0.8 s$, $t = 1.04 s$, $t = 2 s$, $t = 3 s$, $t = 4 s$, $t = 7.2 s$ et $t = 8.4 s$.

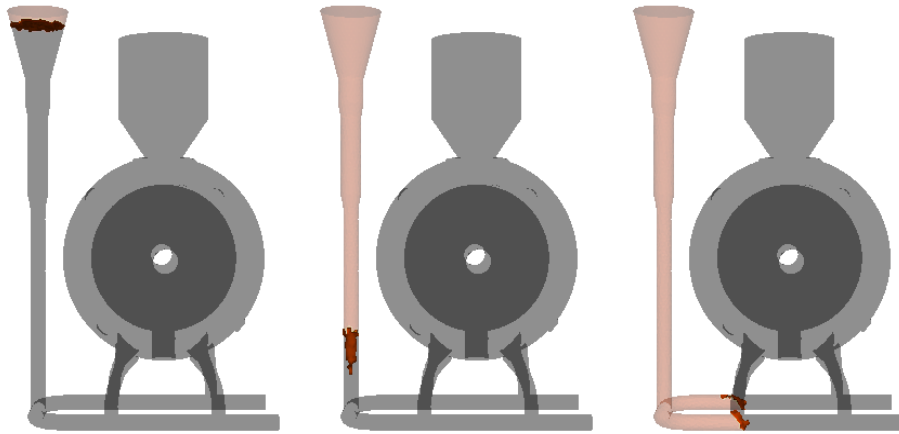


FIGURE 2.13 – L'interface approchée à l'instant initial, $t = 0.4$ s et $t = 0.8$ s.

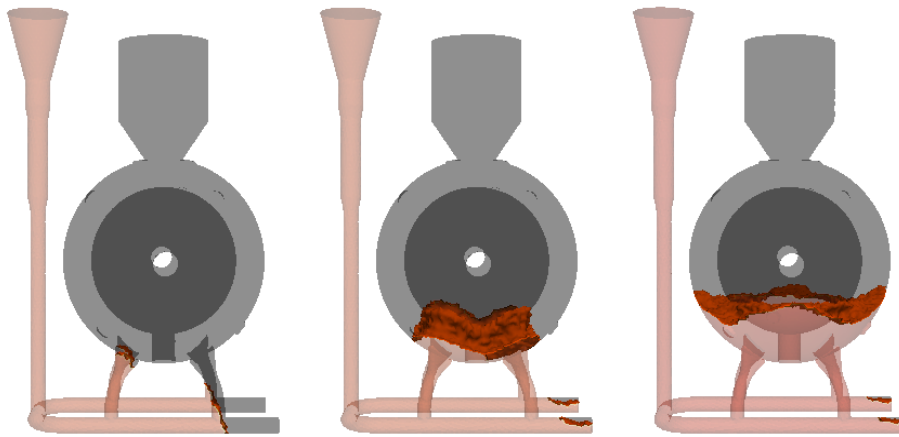


FIGURE 2.14 – L'interface approchée à $t = 1.04$ s, $t = 2$ s et $t = 3$ s.

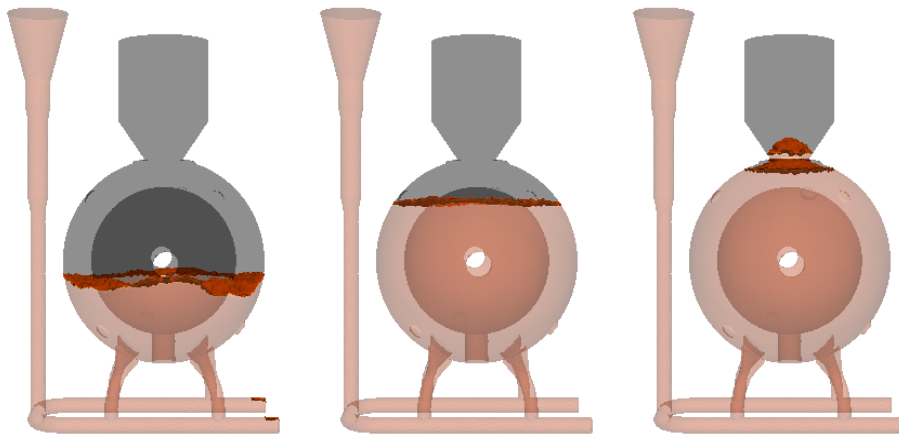


FIGURE 2.15 – L'interface approchée à $t = 4$ s, $t = 7.2$ s et $t = 8.4$ s.

La figure (2.16) nous montre l'évolution de l'erreur pendant le temps pour le cas *iv*. On remarque que cette erreur atteint au maximum la valeur 0.0035, ce qui montre la puissance de l'algorithme proposé.

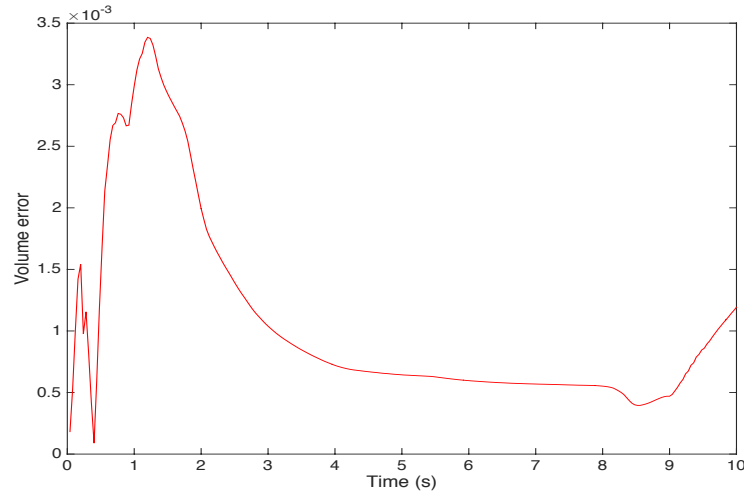


FIGURE 2.16 – Variation de l'erreur en volume pour le cas *iv* pour le cas industriel.

La figure (2.17) nous montre le temps de calcul de la simulation du premier cas industriel.

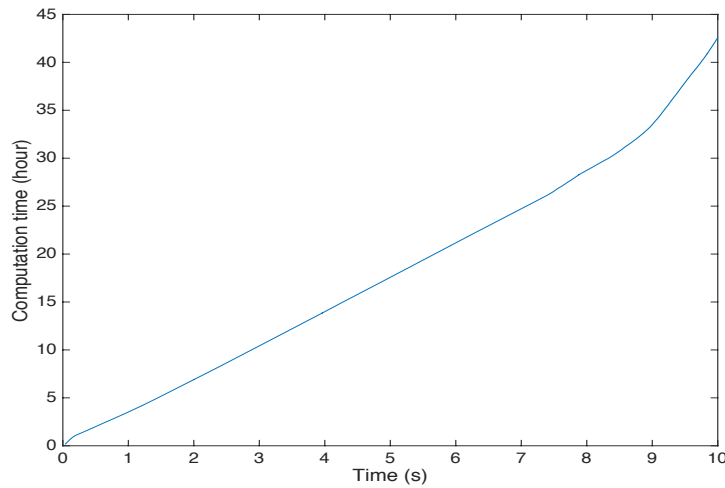


FIGURE 2.17 – Temps de calcul de la simulation du cas industriel.

Conclusion On a construit dans ce chapitre un algorithme pour modéliser le remplissage d'un moule en 2D et en 3D avec la méthode de projection pour les équations de Navier-Stokes. Dans le chapitre suivant, on va valider cet algorithme avec deux expériences :

L'expérience de Martin et Moyce et le Benchmark de Campbell.

De plus, on a essayé d'utiliser l'algorithme de calcul pour des moules de plus grandes tailles, la mémoire est insuffisante pour résoudre le problème. Pour améliorer la performance de l'algorithme, on va faire dans le chapitre 4, un calcul parallèle en utilisant la méthode de décomposition de domaine en particulier les algorithmes itératives de Schwarz.

Chapitre 3

Validations pratiques

Sommaire

3.1	Validations numériques : Simulation eau-air	73
3.1.1	Expérience de Martin et Moyce en 2D	74
3.1.2	Expérience de Martin et Moyce en 3D	77
3.2	Validations numériques : Simulation métal liquide-air (Campbell benchmark)	79

3.1 Validations numériques : Simulation eau-air

Dans cette section, le but est de valider l'algorithme proposé dans les chapitres précédents pour simuler le transport de l'interface dans un écoulement diphasique. En particulier, on valide la capacité de l'algorithme proposé de modéliser des fluides de rapport de densité important (1000 :1) et de rapport de viscosité (10 :1). On considère l'expérience de l'écroutement du barrage de J.C.Martin et W.J.Moyce, où on peut se référer à [99] pour les résultats expérimentaux. Ces résultats sont souvent utilisés pour valider les modèles d'étude des écoulements instationnaires à surface libre. Un schéma du barrage utilisé dans l'expérience est illustré dans la figure (3.1).

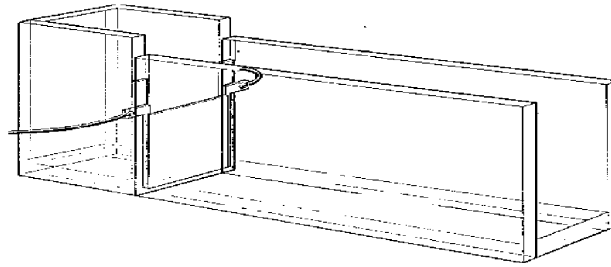


FIGURE 3.1 – Schéma du barrage de l'expérience de J.C.Martin et W.J.Moyce.

Dans cette expérience, on considère une colonne d'eau rectangulaire, initialement au repos sur un plan horizontal rigide retenue par une membrane de papier ciré. Cette membrane est maintenue par un film de cire à une bande métallique du réservoir. Elle est libérée au moyen d'un courant électrique de haute intensité : lorsque le courant circule, la cire fond à l'instant $t = 0$ et l'eau commence à s'écrouler sous l'effet de la gravité. Le liquide s'étale sur le plan horizontal et la hauteur de la colonne se baisse sur le mur verticale. Le mouvement du fluide est enregistré à l'aide d'une caméra prenant 300 images par seconde. On observe la vitesse de chute d'une colonne d'eau dans le sens de la hauteur et dans le sens de l'écroutement.

La hauteur initiale H de la colonne d'eau est 0.5715 dm . La densité de l'eau est 1 Kg.dm^{-3} , sa viscosité cinématique est $10^{-4} \text{ dm}^2.\text{s}^{-1}$ et sa viscosité dynamique est $10^{-4} \text{ Kg.dm}^{-1}.\text{s}^{-1}$; tandis que la densité de l'air est $10^{-3} \text{ Kg.dm}^{-3}$, sa viscosité cinématique est $10^{-3} \text{ dm}^2.\text{s}^{-1}$ et sa viscosité dynamique est $10^{-6} \text{ Kg.dm}^{-1}.\text{s}^{-1}$, l'accélération gravitationnelle est $g = 0.98 \text{ dm.s}^{-2}$.

3.1.1 Expérience de Martin et Moyce en 2D

Le domaine de calcul est de longueur $14H$ et de hauteur $1.4H$.

On fait deux simulations pour modéliser l'expérience en 2D avec deux pas de maillage en espace $h = 1/50$ et $h = 1/100$ respectivement. Les pas de temps sont $\Delta t = 2/50$ et $\Delta t = 2/100$ respectivement.

Dans le problème 2D, la convergence a été obtenue pour le paramètre de régularité $\epsilon = h/5$. Durant la simulation, on impose sur la surface supérieure du domaine de calcul les conditions aux bords de sortie libre. La position dimensionnelle approchée du front de l'interface sur le plan horizontale $l^*(l^* = l/H)$ et la hauteur résiduelle de la colonne d'eau $h^*(h^* = h/H)$ variant en fonction du temps adimensionnel $t^* = t\sqrt{g/H}$ sont comparées avec les résultats expérimentaux présentés dans [99].

Les figures (3.2), (3.3), (3.4) et (3.5) illustrent une comparaison entre les expériences et les résultats numériques pour différents instants. On a pris soin de mettre les figures dans l'échelle dimensionnelle réelle pour comparer avec l'expérience.

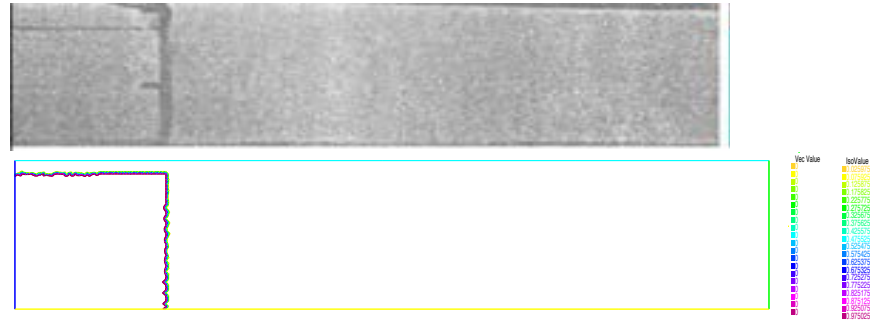


FIGURE 3.2 – En haut : Experience ; En bas : Interface approchée à $t = 0s$.

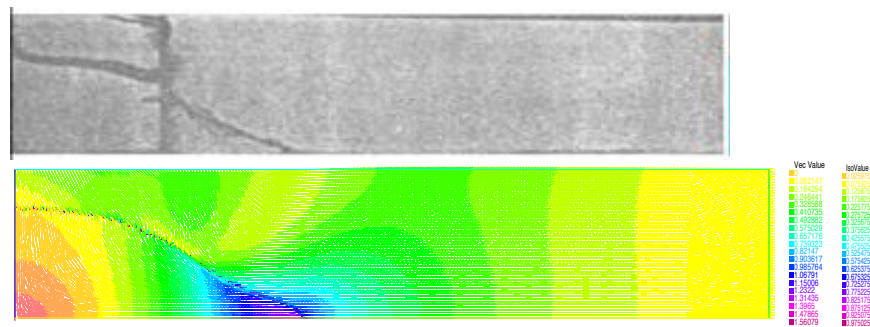


FIGURE 3.3 – En haut : Experience ; En bas : Interface approchée à $t = 0.07s$.

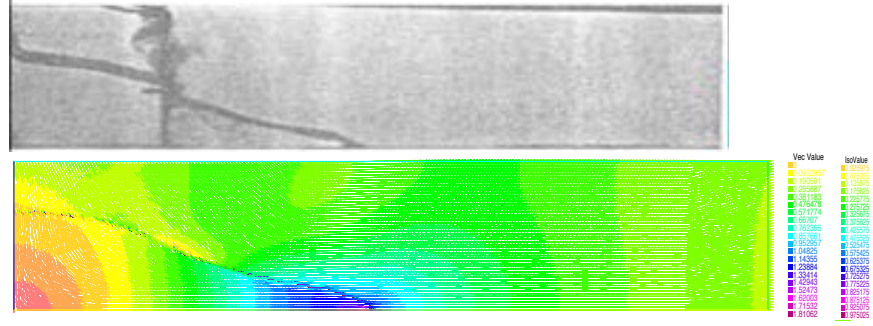


FIGURE 3.4 – En haut : Experience ; En bas : Interface approchée à $t = 0.1s$.

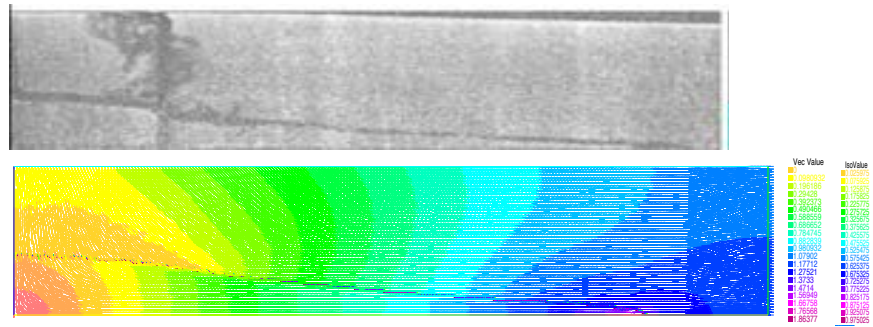


FIGURE 3.5 – En haut : Experience ; En bas : Interface approchée à $t = 0.2s$.

En outre, on peut remarquer que la forme de l'interface approchée et son mouvement sont très similaires aux résultats expérimentaux, ce qui montrent que la méthode numérique désignée dans les chapitres précédents est efficace pour modéliser le transport de l'interface pour des écoulements diphasiques des fluides incompressible non-miscible de rapport de densité important.

La figure (3.6) nous montre une comparaison de la position adimensionnelle de l'interface sur le plan horizontal entre l'expérience et les résultats des deux simulations. De la figure (3.6), on peut remarquer que la position approchée de l'interface sur le plan horizontal est très proche à celle de l'expérience durant toute la simulation ce qui valide l'algorithme utilisé.

Figure (3.7) nous montre la hauteur de la colonne d'eau résiduelle adimensionnelle qui varie avec le temps. Encore une fois, les résultats expérimentaux et numériques sont très proches, ce qui confirme la validité de la méthode proposée dans ce travail.

On peut voir que le maillage le plus fin $h = 1/100$ donne des résultats plus proche de l'expérience que le maillage $h = 1/50$ ce qui est attendu dans le cadre du calcul approché et la convergence des schémas numériques.

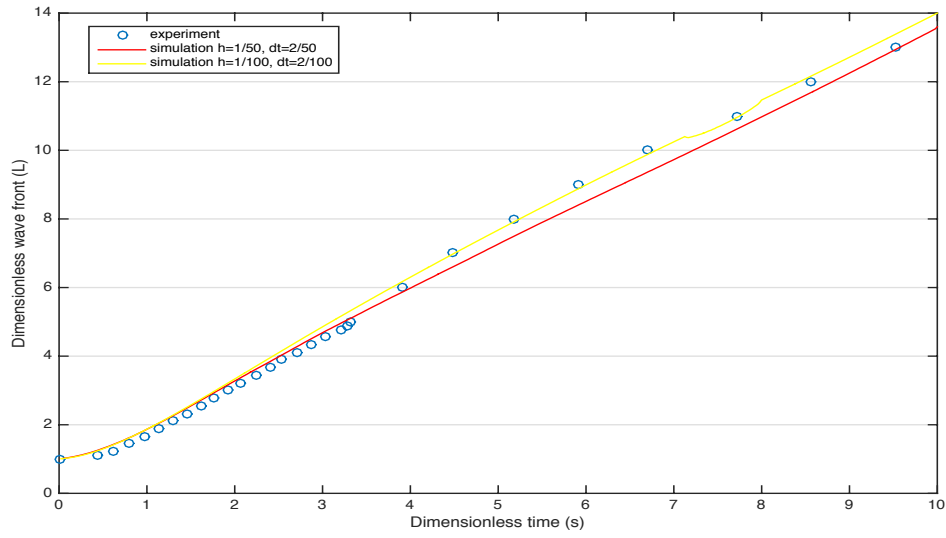


FIGURE 3.6 – Comparaison des résultats expérimentaux et approchés de la position de l'interface adimensionnelle en fonction du temps adimensionnel.

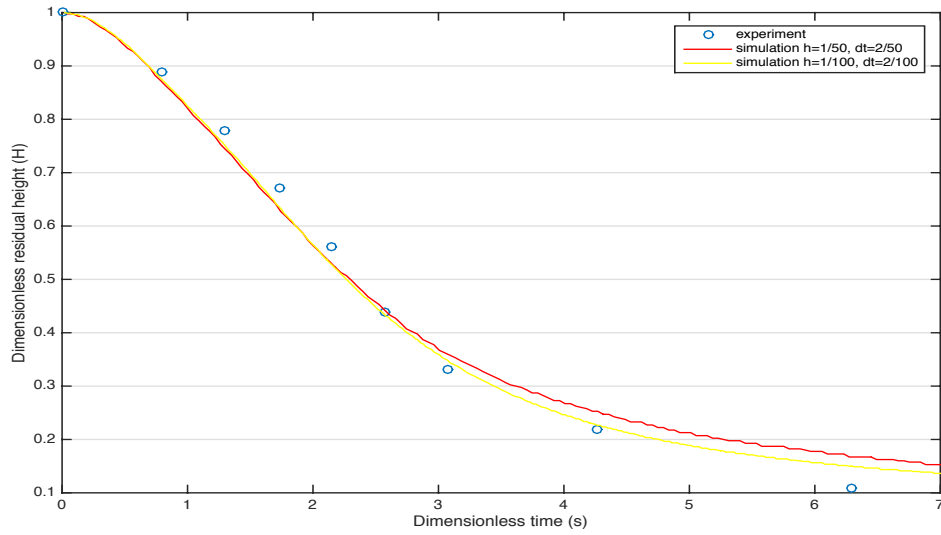


FIGURE 3.7 – Comparaison des résultats expérimentaux et approchés de la hauteur résiduelle adimensionnelle de la colonne d'eau en fonction du temps adimensionnel.

3.1.2 Experience de Martin et Moyce en 3D

Dans cette section, on reprend les mêmes données de l'expérience de Martin et Moyce en 2D. On étend le problème en 3D avec le code parallèle. Le domaine de calcul choisi est de longueur $8H$, de largeur $1H$ et de hauteur $1.4H$.

On effectue deux simulations pour l'éroulement du barrage en 3D, les pas de maillage en espace sont $h = 1/50$ et $h = 1/60$ respectivement. (Les limitations du mémoire et du temps cpu nous limitent à $h = 1/60$ pour le maillage fin).

Les pas de temps pour les deux simulations sont $\Delta t = 2/50$ et $\Delta t = 2/60$ respectivement. Dans les problèmes trois dimensionnels, la convergence a été obtenue pour le paramètre de régularisation $\epsilon = h/5$.

La position approchée de l'interface sur le plan horizontale L et la hauteur résiduelle approchée de la colonne d'eau H variant avec le temps t sont comparées avec les résultats expérimentaux en 3D de Martin et Moyce qui sont présentés dans [126].

Les figures (3.8), (3.9) et (3.10) illustrent la forme de l'interface entre les phases eau-air pour des instants différents. De plus, la forme de l'interface approchée et son mouvement sont très similaires aux résultats présentés dans [126] ce qui valide notre algorithme désigné pour la modélisation du transport de l'interface pour les problèmes trois dimensions.

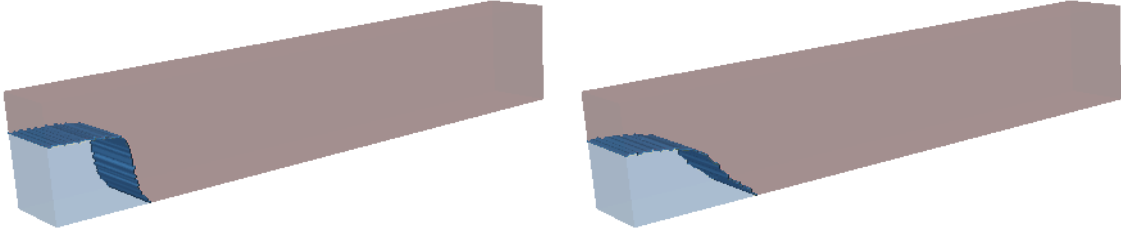


FIGURE 3.8 – Interface approchée en 3 D ; A gauche : $t = 0.0317029s$; A droite : $t = 0.0634057s$.

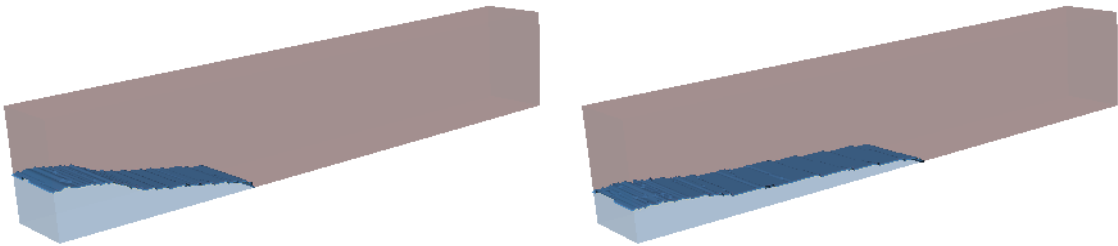


FIGURE 3.9 – Interface approchée en 3 D ; A gauche : $t = 0.13134s$; A droite : $t = 0.203804s$.

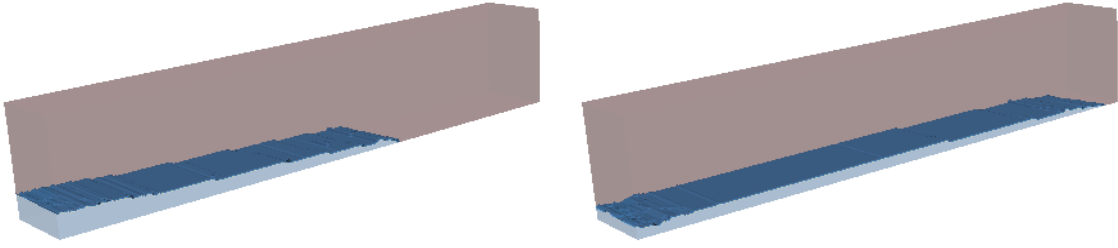


FIGURE 3.10 – Interface approchée en 3 D; A gauche : $t = 0.253623s$; A droite : $t = 0.375905s$.

La figure (3.11) montre une comparaison de la position de l'interface sur le plan horizontale entre l'expérience et les résultats des deux simulations en 3D. De plus, la figure (3.12) montre la hauteur de la colonne d'eau résiduelle adimensionnelle qui varie avec le temps en 3D. On peut remarquer que les résultats numériques sont très proches des résultats de l'expérience durant toute la simulation, ce qui valide l'algorithme utilisé en 3D.

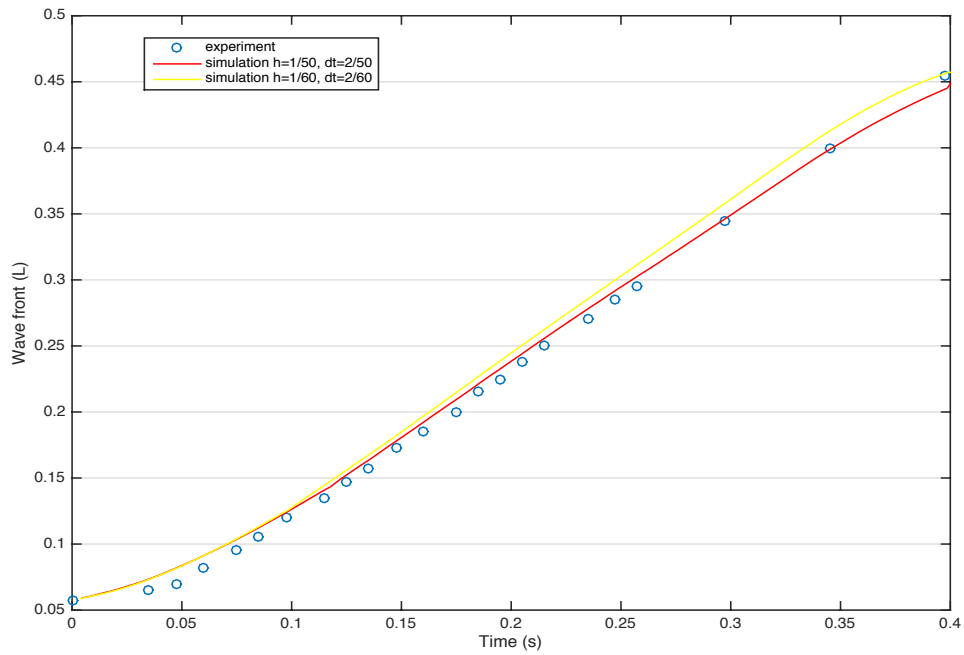


FIGURE 3.11 – Comparaison des résultats expérimentaux et approchés de la position de l'interface en fonction du temps en 3D.

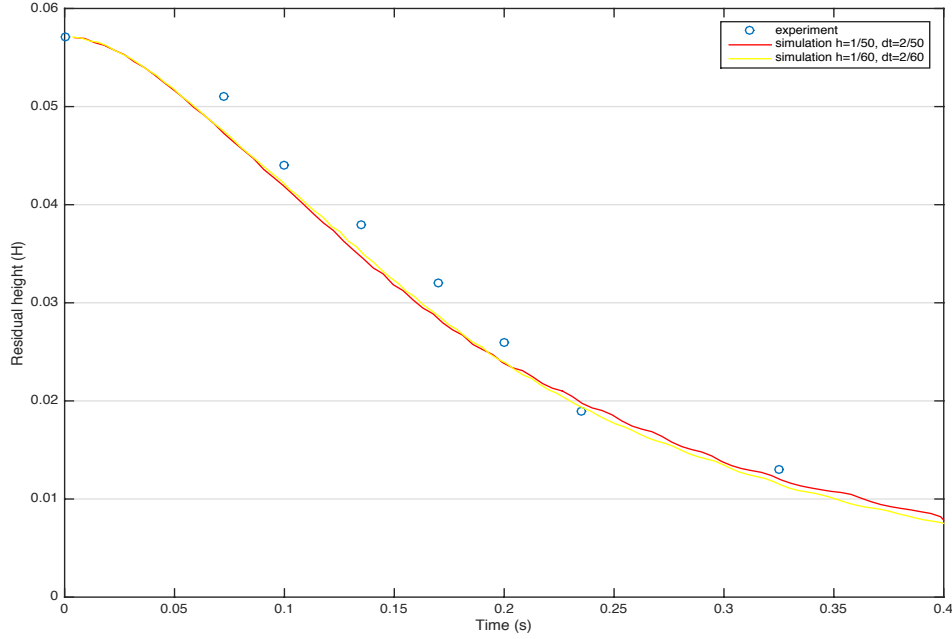


FIGURE 3.12 – Comparaison des résultats expérimentaux et approchés de la hauteur résiduelle de la colonne d’eau en fonction du temps en 3D.

3.2 Validations numériques : Simulation métal liquide-air (Campbell benchmark)

Dans cette section, on désire valider l’algorithme construit dans les chapitres précédents pour modéliser le transport de l’interface d’un écoulement diphasique pour des fluides de rapport de densité important (2385 :1) et de rapport de viscosité (18.18 :1) lors du processus du remplissage des moules par une coulée (métal liquide). Un benchmark du processus de remplissage conçu par le Campbell [135] a été simulé par l’algorithme de remplissage.

Le moule dans le benchmark est une plaque de taille $200mm \times 100mm \times 10mm$ dans les directions X, Y et Z respectivement. La cavité du moule est réalisée en résine de sable collée. Le liquide de la coulée est de 99.999% d’aluminium pure. Le processus de remplissage est enregistré avec la technologie d’imagerie à rayons X. L’instant où le bouchon a été enlevé est désigné comme l’instant initial. L’expérience a été répétée plusieurs fois. Les figures (3.14)-(3.19) montrent deux séquences différentes qui sont comparées aux simulations numériques. Les figures présentent des extraits vidéos à intervalle de 0.25 s pendant le remplissage de deux moules séparés pour vérifier la reproductibilité du phénomène. Les chercheurs ont pris soin de veiller à ce que les conditions expérimentales dans chaque cas ont été reproduites aussi exactement que possible.

La forme et la taille de la cavité du moule, ainsi que le canal d’alimentation sont présentés

dans la figure (3.13). Dans l'expérience, la densité de l'air est de $1Kg.m^{-3}$, la viscosité cinématique est $1e^{-5}m^2.s^{-1}$ et la viscosité dynamique est $1e^{-5}Kg.m^{-1}.s^{-1}$. La densité de l'aluminium liquide est $2385Kg.m^{-3}$, la viscosité cinématique est $0.55e^{-6}m^2.s^{-1}$ et la viscosité dynamique est $1e^{-3}Kg.m^{-1}.s^{-1}$. La température de coulée est $720^{\circ}C$.

Il convient de noter que de légères différences dans le comportement peuvent être vu dans les expériences, même si elles ont été conçus pour être identiques. Ceci est prévu et se trouve dans la nature des phénomènes turbulents. Il est intéressant de noter que les illustrations de la modélisation de l'expérience dans trois cas extrêmes : écoulement laminaire très visqueux, écoulement laminaire peu visqueux, et écoulement turbulent, sont présentés dans [135]. L'approximation numérique qui est la plus proche des résultats observés expérimentalement était pour un grand nombre de Reynolds (40,000 et plus). La plupart des modèles prédisent le temps de remplissage correctement à environ 2,0 secondes.

Dans notre simulation, notre algorithme n'a pas été construit pour simuler un écoulement turbulent, cependant des viscosités numériques artificielles sont utilisées pour le métal liquide et l'air tout en gardant le même rapport de viscosité (18.18 :1), puisque on a remarqué que ce rapport a une grande influence sur la forme de l'interface. La densité de l'air est $1Kg.m^{-3}$, la viscosité cinématique est $1.818e^{-3}m^2.s^{-1}$ et la viscosité dynamique est $1.818e^{-3}Kg.m^{-1}.s^{-1}$; tandis que la densité de l'aluminium est $2385Kg.m^{-3}$, la viscosité cinématique est $1e^{-4}m^2.s^{-1}$ et la viscosité dynamique est $2.385e^{-5}Kg.m^{-1}.s^{-1}$. Ce qui induit un un nombre de Reynolds $Re = 10000$. Le temps de remplissage simulé est 1.98 s ce qui est très proche au temps de remplissage durant l'expérience.

Il est admis que, puisque le liquide d'aluminium est directement versé dans la cavité pendant l'expérience, par conséquent, la vitesse d'entrée devrait être variable avec le temps. Dans l'étude actuelle, une vitesse d'entrée moyenne de $0.5 m.s^{-1}$ a été utilisée au même titre que celle utilisée dans [126]. La vitesse moyenne est estimée à partir du canal d'alimentation et des expériences numériques. Les pas de maillage et de temps considérés sont $h = 0.002$, $\Delta t = 2h$. Pour faciliter la génération du maillage, le canal d'alimentation est légèrement modifié, comme illustré dans la figure (3.13). On verra que l'influence de cette petite modification du système d'alimentation est négligeable et l'interface simulée est toujours proche des résultats expérimentaux enregistrés durant l'expérience à des différents instants.

Les figures (3.14)-(3.15)-(3.16)-(3.17)-(3.18) et (3.19) montrent que le lieu des interfaces simulées est en accord avec les résultats expérimentaux à $t = 0.74 s$, $t = 1 s$, $t = 1.24 s$, $t = 1.5 s$, $t = 1.74 s$ et $t = 2 s$.

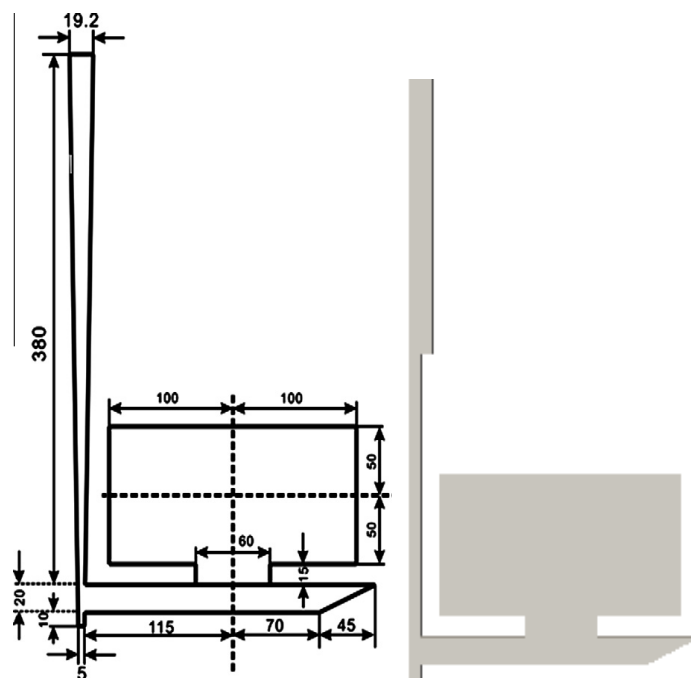


FIGURE 3.13 – A gauche : le moule de Campbell, A droite : le moule de la simulation.

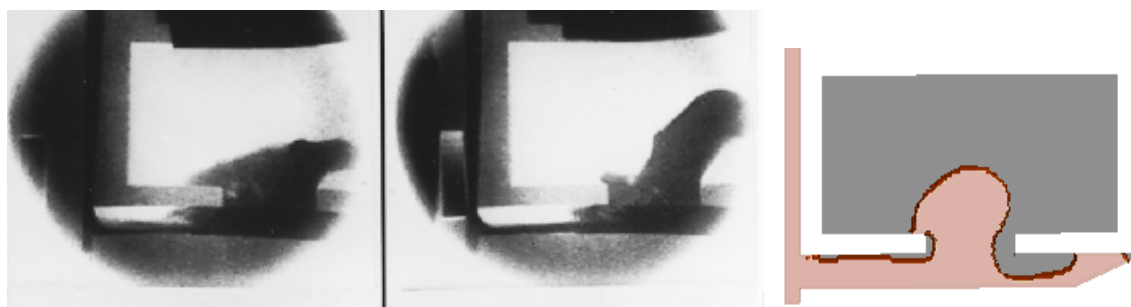


FIGURE 3.14 – Interface à $t = 0.74 s$. à gauche et au milieu : expériences ; à droite : interface simulée.

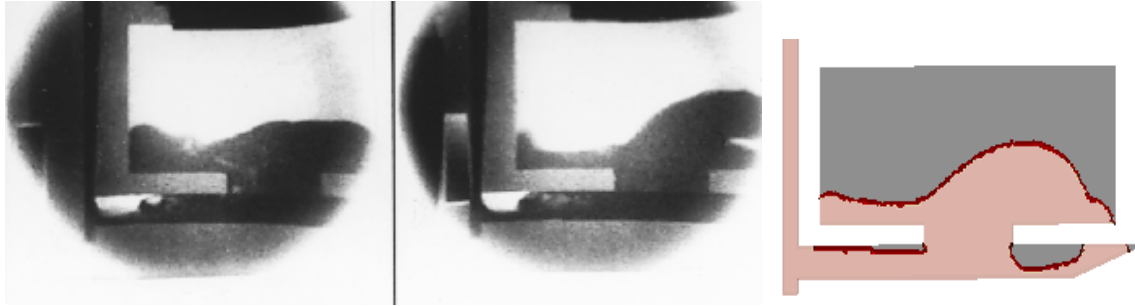


FIGURE 3.15 – Interface à $t = 1$ s. à gauche et au milieu : expériences ; à droite : interface simulée.

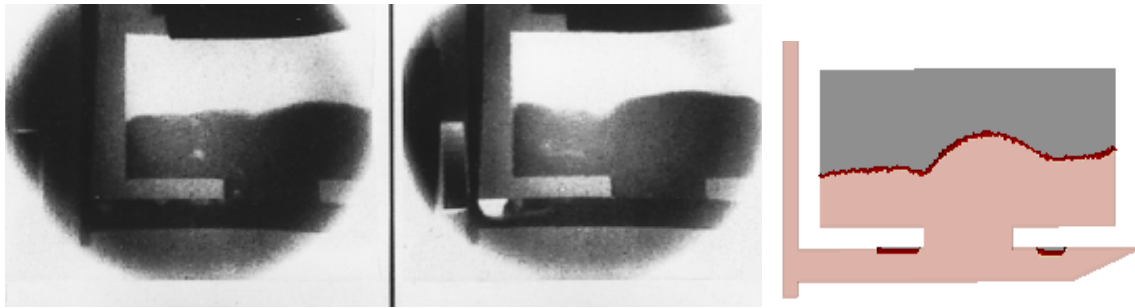


FIGURE 3.16 – Interface à $t = 1.24$ s. à gauche et au milieu : expériences ; à droite : interface simulée.

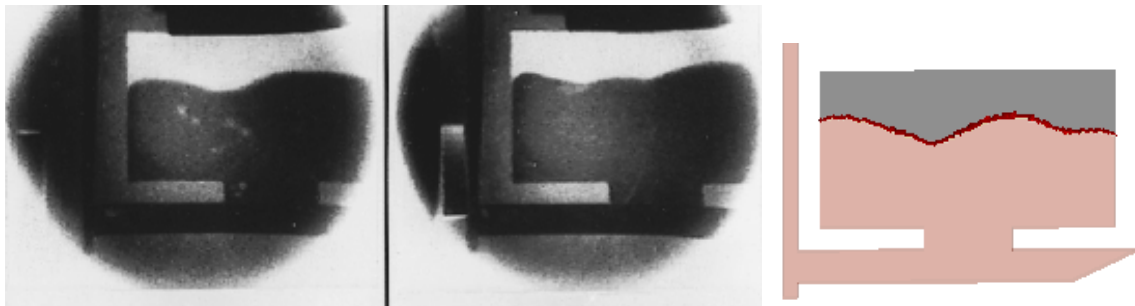


FIGURE 3.17 – Interface à $t = 1.5$ s. à gauche et au milieu : expériences ; à droite : interface simulée.

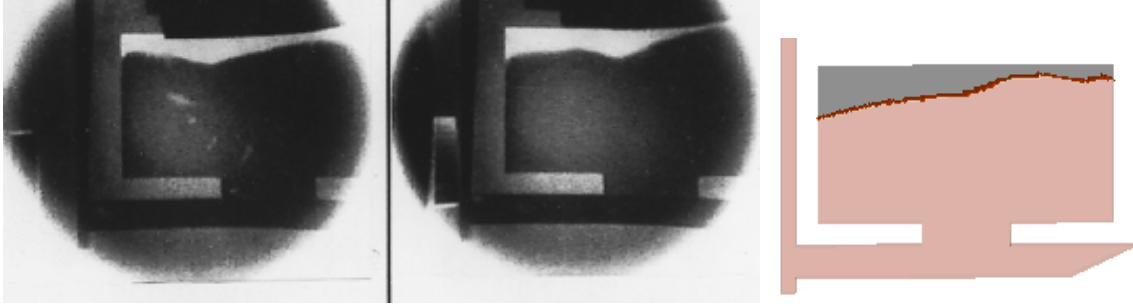


FIGURE 3.18 – Interface à $t = 1.74$ s. à gauche et au milieu : expériences ; à droite : interface simulée.

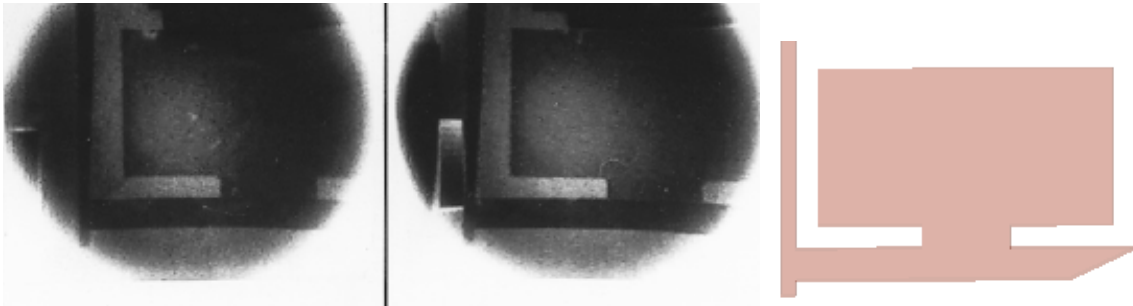


FIGURE 3.19 – Interface à $t = 2$ s. à gauche et au milieu : expériences ; à droite : interface simulée.

Chapitre 4

Calcul Parallèle Itérative

Sommaire

4.1	Motivation	85
4.2	Méthodes de décomposition de domaine	86
4.2.1	Etat de l'art sur les algorithmes itératives de Schwarz	86
4.2.2	Présentation des algorithmes de Schwarz	87
4.2.3	Connection avec l'algorithme de Block Jacobi	90
4.2.4	Les problèmes Multi-dimensionnels avec plusieurs sous-domaines	92
4.2.5	Algorithme de décomposition de domaine utilisé avec le logiciel Freefem++	93
4.3	Méthode de décomposition de domaine avec la méthode de caractéristiques.	94
4.4	Validation des résultats numériques : Comparaison des ré- sultats séquentiels et parallèles	96
4.4.1	Premier cas industriel	96
4.5	Deuxième cas industriel	100

4.1 Motivation

Lors de la modélisation du remplissage de moule trois dimensionnel industriel en particuliers pour ceux de grandes tailles, on a rencontré des limitations en termes de temps de calcul et des insuffisances des ressources de mémoire. La validation de l'algorithme a été faite dans le chapitre précédent. On a testé des simulations numériques en 3D pour des moules de moyennes tailles, le premier exemple industriel est un cas où le maillage contient 294,544 tétraèdres et 59,364 sommets (Voir figure 2.12). Le temps de calcul du code séquentiel est à peu près deux jours. De plus, dans notre étude, il y a des moules de taille plus grand qui exige des maillages plus gros qui ne seront pas faisables avec notre code séquentiel. Le deuxième exemple industriel de ce chapitre un maillage de 2,646,497 tétraèdres et 468,362 sommets, ce qui nous force à paralléliser l'algorithme.

Autrefois, une façon pour améliorer la performance d'un programme (soit en vitesse ou en quantité de donnée traitée) était seulement une question d'attendre les prochaines générations de processeurs. Tous les dix-huit mois, la performance de l'ordinateur doublée. Mais depuis environ 2005 cette vitesse stagne à 2-3 GHz. L'augmentation des performances est presque entièrement due à l'augmentation du nombre de coeurs par processeur. La majorité des fournisseurs de processeurs produisent des machines multicoeurs parallèles. Donc il ne suffit pas d'attendre une prochaine génération de la machine pour garantir une meilleure performance d'un logiciel. Le seul moyen pour accéder à la performance est d'utiliser le calcul parallèle, ce qui implique de gros effort dans le développement algorithmique. De plus, tout la chaîne de calcul doit être parallèle : Visualisation, stockage de données, génération de maillage.

Le but de ce chapitre est d'améliorer la performance de notre algorithme au niveau du temps de calcul et place mémoire afin de résoudre notre problème. Il s'agit de résoudre ce problème en parallèle sur des machines multicoeurs qui contiennent moins de 250 coeurs et de mémoire moins de 1000 GB. (Voir figure(4.1) pour plus des détails sur la capacité de la machine de calcul qu'on a utilisé pour les simulations). Pour nos simulations en parallèle, on a utilisé la machine "hpc2 big-para" avec 128 coeurs et une mémoire de 880 GB.

En particulier, on se concentre dans ce travail sur les solveurs itératifs linéaires parallèles. On va utiliser les méthodes de Schwarz dans la version parallèle de Freefem ++ qui est actuellement liée à HPDDM, un cadre C ++ des méthodes de décomposition de domaine de haute performance, disponible à l'adresse suivante : <https://github.com/hpddm/hpddm>.

Queue	Max Memory	Min- Max CPU	Max Wall time
Interactif -----	80gb	1 - 20	01:00:00
seq ---	500gb	1	100:00:00
small_para -----	1000gb	2 - 50	100:00:00
big_para -----	2000gb	51 - 250	100:00:00

FIGURE 4.1 – Machine de calcul utilisée pour les simulations numériques.

4.2 Méthodes de décomposition de domaine

4.2.1 Etat de l'art sur les algorithmes itératives de Schwarz

L'une des principales classes d'algorithmes de décomposition de domaine est l'algorithme de Schwarz. Hermann Schwarz était un analyste allemand du 19^{ème} siècle. Il était intéressé à démontrer l'existence et l'unicité du problème de Poisson. A son époque, il n'y avait pas les espaces de Sobolev, ni le théorème de Lax-Milgram. Le seul outil disponible était la transformée de Fourier, limitée par sa nature même et avec les géométries simples. Afin d'examiner des situations plus générales, H. Schwarz a mis au point un algorithme itératif pour résoudre le problème de Poisson sur une union de géométries simples, voir [129] (Pour une présentation historique de ce type de méthodes voir [57]). H. Schwarz a démontré la convergence de l'algorithme et a déduit que le problème de Poisson est bien posé dans des géométries complexes. Avec l'apparition des ordinateurs, cette méthode a également acquis un intérêt pratique comme un solveur linéaire itératif. Par la suite, les ordinateurs parallèles sont devenus disponibles et une petite modification de l'algorithme faite par P.L. Lions dans [97] le rend adapté à ces architectures. Sa convergence peut être prouvée en utilisant le principe du maximum [96].

Il existe des différentes versions de l'algorithme de Schwarz au niveau continue, ayant comme point de départ la méthode de H. Schwarz (voir [129]) : Jacobi Schwarz Method (JSM) [40], Additif Schwarz Method (ASM) [146] et Restricted Additif Schwarz (RAS) [23], [24]. La première caractéristique naturelle de ces algorithmes est qu'ils sont équivalentes à une méthode Block-Jacobi lorsque le recouvrement est minimal.

4.2.2 Présentation des algorithmes de Schwarz

Le domaine Ω considéré est l'union d'un disque et d'un rectangle, voir figure (4.2). On considère le problème de Poisson qui consiste à trouver $\mathbf{u} : \Omega \rightarrow \mathbb{R}$ telle que :

$$(4.1) \quad \begin{cases} -\Delta \mathbf{u} = f & \text{dans } \Omega, \\ \mathbf{u} = 0 & \text{sur } \partial\Omega. \end{cases}$$

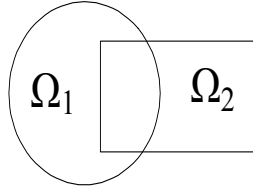


FIGURE 4.2 – Domaine de Schwarz.

L'algorithme de schwarz originale

L'algorithme de Schwarz est une méthode itérative basée sur la résolution alternative des sous-problèmes dans des domaines Ω_1 et Ω_2 . Il calcule $(\mathbf{u}_1^{n+1}, \mathbf{u}_2^{n+1})$ à partir de $(\mathbf{u}_1^n, \mathbf{u}_2^n)$ de la manière suivante : On calcule $\mathbf{u}_1^{n+1}$ en résolvant

$$(4.2) \quad \begin{cases} -\Delta \mathbf{u}_1^{n+1} = f & \text{dans } \Omega_1 \\ \mathbf{u}_1^{n+1} = 0 & \text{sur } \partial\Omega_1 \cup \partial\Omega \\ \mathbf{u}_1^{n+1} = \mathbf{u}_2^n & \text{sur } \partial\Omega_1 \cup \overline{\Omega_2}. \end{cases}$$

et ensuite, on calcule $\mathbf{u}_2^{n+1}$ ayant $\mathbf{u}_1^{n+1}$ en résolvant

$$(4.3) \quad \begin{cases} -\Delta \mathbf{u}_2^{n+1} = f & \text{dans } \Omega_2 \\ \mathbf{u}_2^{n+1} = 0 & \text{sur } \partial\Omega_2 \cup \partial\Omega \\ \mathbf{u}_2^{n+1} = \mathbf{u}_1^{n+1} & \text{sur } \partial\Omega_2 \cup \overline{\Omega_1}. \end{cases}$$

l'algorithme de schwarz parallèle

C'est une méthode itérative qui résout simultanément dans tous les sous-domaines, $i = 1, 2$

le problème :

$$(4.4) \quad \begin{cases} -\Delta \mathbf{u}_i^{n+1} = f & \text{dans } \Omega_i \\ \mathbf{u}_i^{n+1} = 0 & \text{sur } \partial\Omega_i \cup \partial\Omega \\ \mathbf{u}_i^{n+1} = \mathbf{u}_{3-i}^n & \text{sur } \partial\Omega_i \cup \overline{\Omega}_{3-i}. \end{cases}$$

Les algorithmes (4.1), (4.3) et (4.4) calculent des fonctions locales $\mathbf{u}_i$ dans $\Omega_i, i = 1, 2$. Afin d'écrire les algorithmes qui résolvent le problème globale et qui calculent des fonctions globales dans $H^1(\Omega)$, l'espace dans laquelle le problème (4.1) est naturellement posé, on a besoin des opérateurs d'extensions et des partitions de l'unité.

Opérateurs d'extensions et partition de l'unité

Soit l'opérateur d'extension $E_i, i = 1, 2$ tel que $E_i(w_i) : \Omega \rightarrow \mathbb{R}$ est l'extension d'une fonction $w_i : \Omega_i \rightarrow \mathbb{R}$, par 0 à l'extérieur de Ω_i . On définit également la fonction de partition de l'unité continue $\mathcal{X}_i : \Omega_i \rightarrow \mathbb{R}, \mathcal{X}_i \geq 0$ et $\mathcal{X}_i(x) = 0$ pour $x \in \partial\Omega_i \setminus \partial\Omega$ et tel que :

$$w = \sum_{i=1}^2 E_i(\mathcal{X}_i w|_{\Omega_i}).$$

pour toute fonction $w : \Omega \rightarrow \mathbb{R}$.

Il y a deux façons d'écrire des algorithmes liés qui agissent sur les fonctions $\mathbf{u}^n \in H^1(\Omega)$. Soit $\mathbf{u}^n$ une approximation de la solution du problème de Poisson (4.1), $\mathbf{u}^{n+1}$ est calculée en résolvant les sous-problèmes locaux suivants :

$$(4.5) \quad \begin{cases} -\Delta \mathbf{w}_i^{n+1} = f & \text{dans } \Omega_i, \\ \mathbf{w}_i^{n+1} = 0 & \text{sur } \partial\Omega_i \cup \partial\Omega, \\ \mathbf{w}_i^{n+1} = \mathbf{u}^n & \text{sur } \partial\Omega_i \cup \overline{\Omega}_{3-i}, \end{cases}$$

puis on les colle ensemble en utilisant les fonctions de partition de l'unité :

$$(4.6) \quad \mathbf{u}^{n+1} = \sum_{i=1}^2 E_i(\mathcal{X}_i \mathbf{u}_i^n).$$

L'algorithme (4.5) qui itère sur $\mathbf{u}^n$ et l'algorithme (4.4) qui itère sur $(\mathbf{u}_1^n, \mathbf{u}_2^n)$ sont équivalents. La démonstration de ce résultat se trouve dans [40].

Une autre formulation de l'algorithme (4.5) a été introduite dans [40] en terme du résidu continu $\mathbf{r}^n = f + \Delta \mathbf{u}^n$. De cette façon, on se rapproche de la définition algébrique des méthodes de décomposition de domaine.

Par la suite, on décrit quelques algorithmes correspondants :

L'algorithme RAS au niveau continue

- a) Calculer le résidu $\mathbf{r}^n : \Omega \rightarrow \mathbb{R}$ avec :

$$(4.7) \quad \mathbf{r}^n = f + \Delta \mathbf{u}^n.$$

- b) Pour $i = 1, 2$ effectuer une correction locale $\mathbf{v}_i^n$:

$$(4.8) \quad -\Delta \mathbf{v}_i^n = \mathbf{r}^n \text{ dans } \Omega_i, \quad \mathbf{v}_i^n = 0 \text{ sur } \partial\Omega_i.$$

- c) Effectuer un moyen des corrections locales et mettre à jour les valeurs de $\mathbf{u}^n$ en calculant $\mathbf{u}^{n+1}$:

$$(4.9) \quad \mathbf{u}^{n+1} = \mathbf{u}^n + E_1(\mathcal{X}_1 \mathbf{v}_1^n) + E_2(\mathcal{X}_2 \mathbf{v}_2^n),$$

où $(\mathcal{X}_i)_{i=1,2}$ constitue la partition de l'unité déjà définie.

L'algorithme ASM au niveau continue

- a) Calculer le résidu $\mathbf{r}^n : \Omega \rightarrow \mathbb{R}$ avec :

$$(4.10) \quad \mathbf{r}^n = f + \Delta \mathbf{u}^n.$$

- b) Pour $i = 1, 2$ effectuer une correction locale $\mathbf{v}_i^n$:

$$(4.11) \quad -\Delta \mathbf{v}_i^n = \mathbf{r}^n \text{ dans } \Omega_i, \quad \mathbf{v}_i^n = 0 \text{ sur } \partial\Omega_i.$$

- c) Mettre à jour $\mathbf{u}^n$:

$$(4.12) \quad \mathbf{u}^{n+1} = \mathbf{u}^n + E_1(\mathbf{v}_1^n) + E_2(\mathbf{v}_2^n),$$

En résumé, à partir de l'algorithme de Schwarz original (4.2) qui est séquentiel, on a ainsi trois algorithmes continus qui sont essentiellement parallèles :

- Algorithme (4.4) Jacobi Schwarz Method (JSM)
- Algorithme (4.7)- (4.8)-(4.9) Restricted Additive Schwarz Method (RAS)
- Algorithme (4.10)- (4.11)-(4.12) Additive Schwarz Method (ASM)

4.2.3 Connection avec l'algorithme de Block Jacobi

Dans la section précédente, on a remarqué que les trois méthodes illustrent différents points de vue de l'algorithme de Schwarz. L'aspect continu a souligné l'intérêt du recouvrement qui peut être souvent caché par la discrétisation. Au niveau discret, on voit que l'algorithme de Schwarz est, d'un point de vue algébrique, une variation d'un algorithme de block-Jacobi.

On rappelle d'abord la définition de l'algorithme du block Jacobi puis on établit une connexion avec les algorithmes de Schwarz. On considère le système linéaire :

$$(4.13) \quad A\mathbf{U} = \mathbf{F}$$

où A est une matrice de dimension $m \times m$, $\mathbf{F} \in \mathbb{R}^m$ est le second membre de dimension m et $\mathbf{U} \in \mathbb{R}^m$ est une solution de dimension m où m est un entier. l'ensemble des indices $\{1, \dots, m\}$ est partitionné en deux ensembles $\mathcal{N}_1 = \{1, \dots, m_s\}$ et $\mathcal{N}_2 = \{m_s + 1, \dots, m\}$. Soit $\mathbf{U}_1 = (\mathbf{U}_k)_{k \in \mathcal{N}_1} = \mathbf{U}_{\mathcal{N}_1}$, $\mathbf{U}_2 = (\mathbf{U}_k)_{k \in \mathcal{N}_2} = \mathbf{U}_{\mathcal{N}_2}$ et de même $\mathbf{F}_1 = \mathbf{F}|_{\mathcal{N}_1}$, $\mathbf{F}_2 = \mathbf{F}|_{\mathcal{N}_2}$. Le système linéaire a la forme du bloc suivant :

$$(4.14) \quad \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{pmatrix}$$

où $A_{ij} = A|_{\mathcal{N}_i \times \mathcal{N}_j}$, $1 \leq i, j \leq 2$.

L'algorithme de Jacobi se résume de la manière suivante :

Soit D la matrice diagonale de A , l'algorithme de Jacobi s'écrit :

$$D\mathbf{U}^{n+1} = D\mathbf{U}^n + (\mathbf{F} - A\mathbf{U}^n)$$

ou d'une façon équivalente,

$$\mathbf{U}^{n+1} = \mathbf{U}^n + D^{-1}(\mathbf{F} - A\mathbf{U}^n) = \mathbf{U}^n + D^{-1}\mathbf{r}^n$$

où $\mathbf{r}^n$ est le résidu de l'équation.

On définit maintenant l'algorithme de Block Jacobi qui s'écrit comme suivant :

$$(4.15) \quad \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} \begin{pmatrix} \mathbf{U}_1^{n+1} \\ \mathbf{U}_2^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1 - A_{12}\mathbf{U}_2^n \\ \mathbf{F}_2 - A_{21}\mathbf{U}_1^n \end{pmatrix}.$$

Afin d'avoir une forme plus compacte de l'algorithme précédent, on introduit $R_i, i = 1, 2$ l'opérateur de restriction de $\mathcal{N}$ dans $\mathcal{N}_i$. Les opérateurs transposés R_i^T sont des opérateurs d'extensions de $\mathcal{N}_i$ dans $\mathcal{N}$. On note $A_{ii} = R_i A R_i^T$.

La forme compacte de l'algorithme de Block Jacobi est ainsi donnée par :

$$(4.16) \quad \mathbf{U}^{n+1} = \mathbf{U}^n + (R_1^T (R_1 A R_1^T)^{-1} R_1 + R_2^T (R_2 A R_2^T)^{-1} R_2) \mathbf{r}^n.$$

Connection avec l'algorithme de Schwarz

La résolution de $\mathbf{U}_1^{n+1}$ correspond à la résolution d'un problème muni des conditions aux bords de Dirichlet dans le sous-domaine Ω_1 avec les données de Dirichlet de l'autre sous-domaine à l'étape précédente.

Algorithme algébrique : Partition de l'unité discrète :

Notre objectif est de présenter les algorithmes algébriques associés aux algorithmes continus RAS et ASM : La façon la plus simple de le faire est d'écrire la méthode itérative en termes de résidus comme dans l'équation (4.16).

La partition de l'unité algébrique

Au niveau discret, les principaux ingrédients de la partition de l'unité sont

- Un ensemble d'indices de degré de liberté $\mathcal{N}$ et une décomposition en N sous-ensembles $\mathcal{N} = \bigcup_{i=1}^N \mathcal{N}_i$.
- Un vecteur $\mathbf{U} \in \mathbb{R}^{dim(\mathcal{N})}$.
- La restriction d'un vecteur $\mathbf{U} \in \mathbb{R}^{dim(\mathcal{N})}$ à un sous-domaine $\Omega_i, 1 \leq i \leq N$ peut être exprimer comme $R_i \mathbf{U}$ où R_i est une matrice booléenne rectangulaire de dimension $dim(\mathcal{N}_i) \times dim(\mathcal{N})$. L'opérateur d'extension est la matrice transposée R_i^T .
- Les fonctions partition de l'unité au niveau discret correspondent aux matrices de dimension $dim(\mathcal{N}_i) \times dim(\mathcal{N}_i)$ avec des entrées non-négatives telle que pour tous les vecteurs $\mathbf{U} \in \mathbb{R}^{dim(\mathcal{N})}$

$$\mathbf{U} = \sum_{i=1}^N R_i^T D_i R_i \mathbf{U},$$

ceci implique

$$\mathbf{I}_d = \sum_{i=1}^N R_i^T D_i R_i$$

où $\mathbf{I}_d \in \mathbb{R}^{dim \mathcal{N} \times dim \mathcal{N}}$ est la matrice identité.

4.2.4 Les problèmes Multi-dimensionnels avec plusieurs sous-domaines

Dans le cas général, l'ensemble des indices $\mathcal{N}$ peut être partitionné par un partitionneur graphique automatique comme METIS [84] ou SCOTCH [28]. De la matrice d'entrée A , un graphique de connectivité est créé. Deux indices $i, j \in \mathcal{N}$ sont reliés si le coefficient de la matrice $A_{ij} \neq 0$. Normalement, même si la matrice A n'est pas symétrique, le graphe de connectivité est symétrisé. Ensuite, les algorithmes qui trouvent un bon partitionnement des graphes non-structurés sont fortement utilisés. Cette distribution doit être faite de telle sorte que le nombre d'éléments attribués à chaque processeur est à peu près le même, et le nombre d'éléments adjacents attribués à différents processeurs est minimisé. Le but de la première condition est d'équilibrer le calcul entre les processeurs. Alors que la deuxième condition a pour but de réduire au minimum la communication résultante de la place des éléments adjacents aux différents processeurs.

La figure (4.3) nous montre un exemple des partitionnements en 4 sous-domaines dans le cas du cube en 3D et le premier cas industriel considérés dans notre problème.

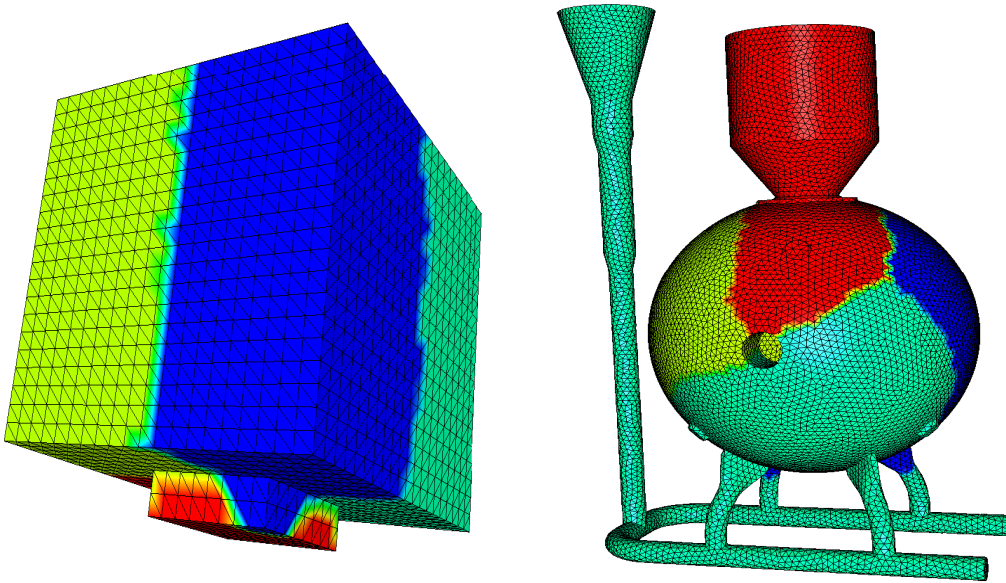


FIGURE 4.3 – Exemples des maillages partitionnés avec METIS

4.2.5 Algorithme de décomposition de domaine utilisé avec le logiciel Freefem++

L'algorithme de décomposition de domaine se résume alors par les étapes suivantes :

- a) La géométrie est partitionnée et distribuée entre les processeurs MPI. A ce niveau, toutes les tâches peuvent être calculées en même temps, ce qui signifie que chaque processeurs MPI est en charge d'un seul sous-domaine et les variables sont locales dans chaque processeur.
- b) Afin de trouver l'approximation par éléments finis de la solution, on résout le système algébrique global correspondant. FreeFem++ contient les solveurs :
 - un solveur de GMRES parallèle qui est le solveur parallèle par défaut dans FreeFem ++,
 - une méthode Schwarz à un niveau, soit ASM ou RAS,
 - la méthode Schwarz à deux niveaux avec un espace grossier Geneo.
- c) Les matrices de rigidité locales sont assemblées en même temps : L'opérateur local est attaché à une structure distribuée dschwarz.
- d) Le système linéaire distribué est résolu par un appel qui inclu des arguments supplémentaires en ligne de commandes qui sont automatiquement transmis aux solveurs.

Dans ce chapitre, on a utilisé La méthode de Schwarz à un niveau avec le solveur parallèle RAS pour les 5 matrices de l'algorithme du remplissage.

On rappelle que les cinq matrices à résoudre sont les problèmes algébriques associés aux formulations variationnelles discrètes suivantes :

1- Chercher $\mathbf{u}_h^* \in X_{h,U_{in},0}$ tel que pour tout $\mathbf{v}_h \in X_h$, on a :

$$(4.17) \quad \left(\rho^n \frac{\mathbf{u}_h^* - \mathbf{u}_h^n o X_h^n}{\Delta t}, \mathbf{v}_h \right) + (2\mu^n \mathbb{D}\mathbf{u}_h^*, \mathbb{D}\mathbf{v}_h) + (\nabla p_h^n, \mathbf{v}_h) = (\rho^n \mathbf{g}, \mathbf{v}_h).$$

2- Chercher $p_h^{n+1} \in M_h$ tel que pour tout $q_h \in M_h$, on a :

$$(4.18) \quad (\text{div} \mathbf{u}_h^*, q_h) + \left(\frac{\Delta t}{\rho^n} \nabla (p_h^{n+1} - p_h^n), \nabla q_h \right) = 0.$$

3- Chercher $\mathbf{u}_h^{n+1} \in L_h$ tel que pour tout $\mathbf{v}_h \in L_h$, on a :

$$(4.19) \quad \left(\mathbf{u}_h^{n+1}, \mathbf{v}_h \right) + (\varepsilon \nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h) = (\mathbf{u}_h^*, \mathbf{v}_h) - \left(\frac{\Delta t}{\rho^n} \nabla (p_h^{n+1} - p_h^n), \mathbf{v}_h \right).$$

4- Chercher $\phi_h^{n+1} \in Y_h$ tel que pour tout $\mathbf{r}_h \in Y_h$, on a :

$$(4.20) \quad \left(\frac{\phi_h^{n+1}}{\Delta t}, r_h \right) + \varepsilon (\nabla \phi_h^{n+1}, \nabla r_h) - \epsilon \int_{\Gamma_1} G(\mathbf{u}_h^n) r_h = \left(\frac{\phi_h^n o X_h^n}{\Delta t}, r_h \right)$$

5- Chercher $\phi_h^{n+1} \in Y_h$ tel que pour tout $\mathbf{r}_h \in Y_h$, on a :

$$(4.21) \quad \left(\frac{\Phi_h^{n+1} - \Phi_h^n X_{(\mathbf{w}_h^n, \Delta\tau)}^n}{\Delta\tau}, r_h \right) - (\text{sign}(\phi)_h^n, r_h) = 0.$$

Après parallélisation du code, une différence entre les solutions des matrices des codes séquentiels et parallèles a été remarqué après des dizaines d'itérations (figure 4.8), en particulier les matrices qui contiennent les termes des courbes caractéristiques. Pour corriger ce problème on a modifié dans la section suivante l'algorithme pour qu'il soit approprié à la méthode de caractéristiques.

4.3 Méthode de décomposition de domaine avec la méthode de caractéristiques.

Dans le cas séquentiel (sans décomposition de domaine), on sait que les courbes caractéristiques peuvent sortir du domaine de calcul à côté des bords (Voir figure 4.4). Près du bord, durant l'évolution des caractéristiques, on détecte les points qui tentent d'aller hors du domaine. On les arrête à la frontière et on calcule l'intégrale correspondante.

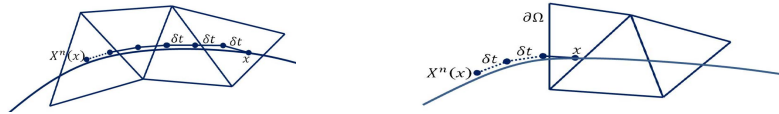


FIGURE 4.4 – A gauche : le point caractéristique se déplaçant avec le pas de temps δt à l'intérieur du domaine ; A droite : le point caractéristique sortant du domaine.

En utilisant la méthode de décomposition de domaine avec 1 seul recouvrement (Voir figure 4.6 et 4.7), chaque sous-domaine avance d'une couche vers le sous-domaine adjacent, donc le recouvrement entre 2 sous-domaines comporte 2 couches de triangle (zone verte). Si on utilise la technique mentionnée ci-dessus dans chaque sous-domaine, la solution sera affectée par de petites erreurs numériques qui s'accumulent et changent les résultats avec l'évolution du temps (comme on va voir dans les résultats numériques). Afin de corriger ce défaut, on prolonge le sous-domaine en ajoutant une couche supplémentaire (Voir la zone beue dans les figures 4.6 et 4.7). La région étendue ne sera utilisée que pour le calcul des courbes caractéristiques qui sortent du sous-domaine.

De plus, on a remarqué durant les tests numériques sur différents moules avec différent pas de temps, que plus que le pas de temps est plus grand, plus on a besoin d'ajouter des couches de recouvrement. Avec un pas de temps $dt = 2h$, il suffit de mettre 4 couches, pour le pas de temps $dt = 3h$ il suffit d'ajouter 6 couches. Pour ne pas risquer que les courbes caractéristiques sortent des sous-domaines, on a ajouté pour nos simulations 20

couches. Ce nombre de recouvrement n'affecte pas le temps de calcul des factorizations des matrices qui est l'étape la plus chère dans le calcul. Cette zone est juste utilisée pour le calcul des intégrales qui contiennent des courbes caractéristiques.

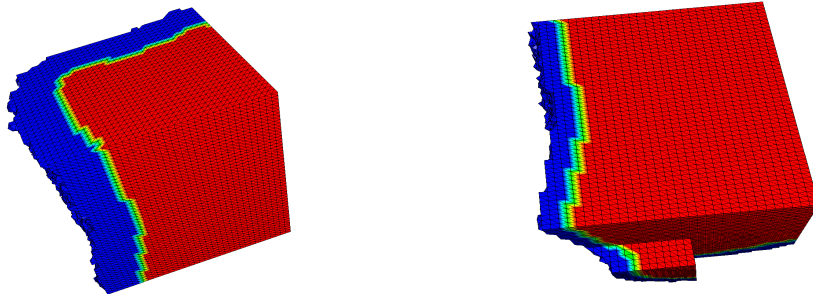


FIGURE 4.5 – Des sous-domaines du cube avec la region ajoutée pour le calcul des courbes caractéristiques.

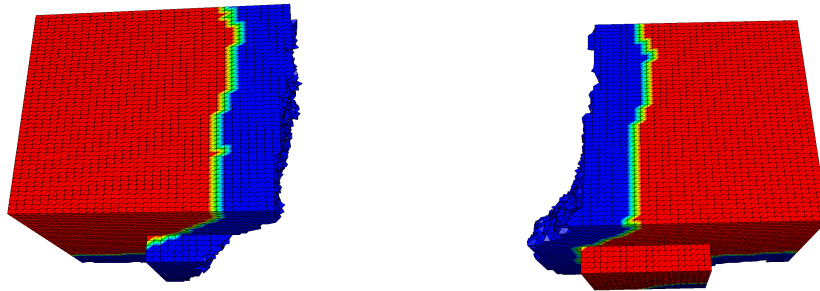


FIGURE 4.6 – Des sous-domaines du cube avec la region ajoutée pour le calcul des courbes caractéristiques.

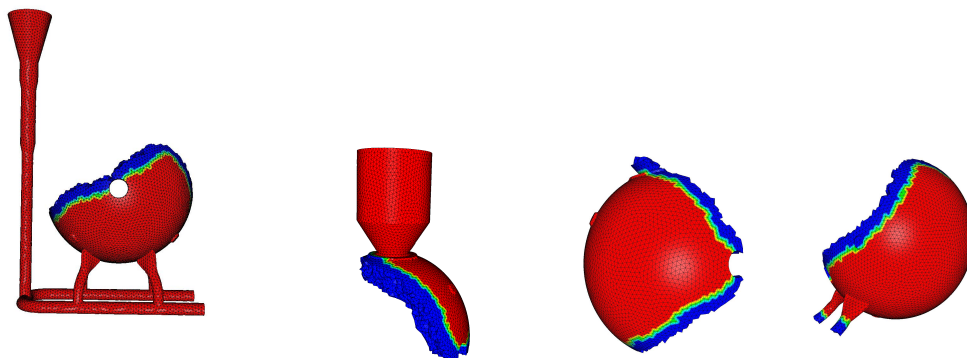


FIGURE 4.7 – Des sous-domaines du premier cas industriel avec la region ajoutée pour le calcul des courbes caractéristiques.

4.4 Validation des résultats numériques : Comparaison des résultats séquentiels et parallèles

Dans cette section, on va comparer les résultats numériques des codes séquentiels et parallèles. On a choisi pour ce but de comparer l'erreur en volume dans le cas des solutions avec les conditions aux bords de type Navier pour la vitesse et Neumann non-homogène pour la fonction Level-Set avant la conservation de masse pour vérifier que les solutions des 5 matrices donne le même résultat pour le problème trois dimensionnel industriel. Dans une autre étape, on va présenter pour chaque cas le temps de calcul de chaque matrice pour la première itération en fixant le nombre des inconnus et en faisant évoluer le nombre des coeurs de 1 à 128 coeurs.

4.4.1 Premier cas industriel

Dans cette section, on considère le même exemple de la section 2.4.3. On compare l'erreur en volume en fonction du temps pour la simulation avec le code séquentiel, avec le code parallèle avec 1 recouvrement pour l'algorithme de Schwarz puis la simulation avec 1 recouvrement pour l'algorithme de Schwarz et 20 recouvrements pour le calcul de l'intégral des courbes caractéristiques.

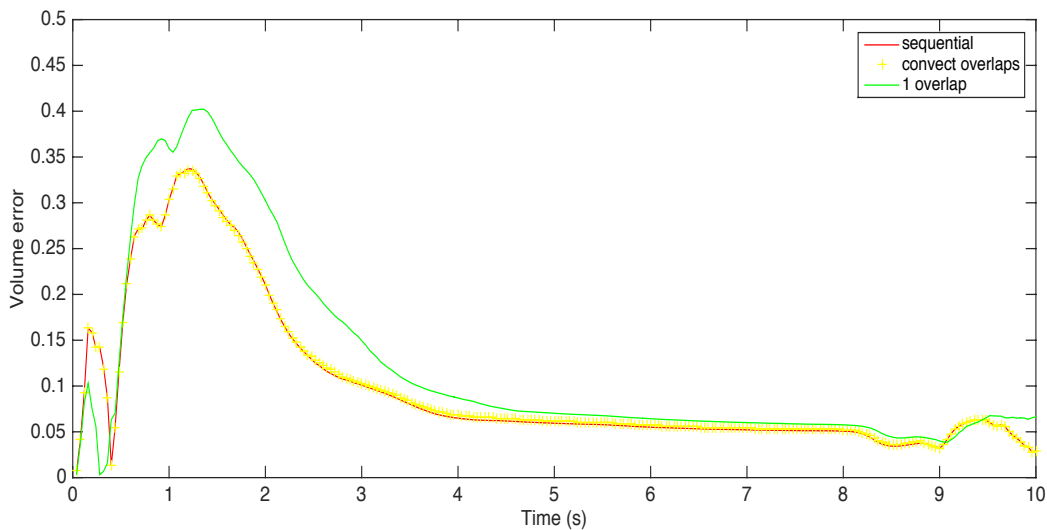


FIGURE 4.8 – L'erreur en volume en fonction du temps pour $dt = 2h$ dans le cas séquentiel, dans le cas parallèle avec 1 recouvrement et dans le cas parallèle avec 20 recouvrements.

On remarque que l'algorithme séquentiel et parallèle donne le même résultat avec l'algorithme parallèle désigné pour la méthode de caractéristiques. L'erreur qui a été accumulée

4.4. Validation des résultats numériques : Comparaison des résultats séquentiels et parallèles

dans le code parallèle classique avec 1 seul recouvrement se corrige avec cette modification. De plus, graphiquement on obtient le même résultat.

Dans la figure (4.28) on présente le temps de calcul de la simulation avec le code parallèle avec 128 coeurs. On remarque que le temps de calcul à diminuer à peu près de 30 fois du code séquentiel. La diminution n'est pas proche à 128 fois parceque le nombre des inconnues n'est pas assez grand. On va voir dans l'exemple suivant que l'algorithme de calcul parallèle qu'on a utilisé est plus efficace quand le nombre des inconnues est assez grand.

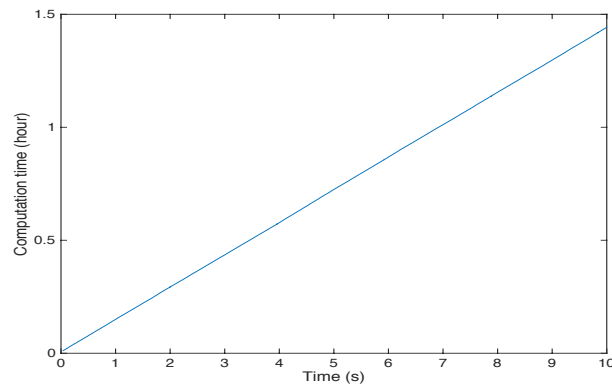


FIGURE 4.9 – Temps de calcul de la simulation du premier cas industriel avec le code parallèle.

Dans les figures (4.10, 4.11, 4.12, 4.13 et 4.14) on présente le temps de calcul en échelle logarithmique ("strong scalability") de la première itération des cinq matrices de l'algorithme en faisant évoluer le nombre des coeurs de 1 à 128 coeurs.

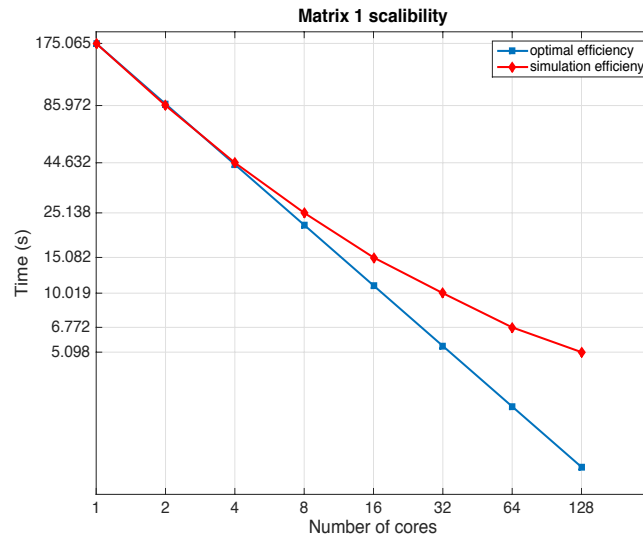


FIGURE 4.10 – "Strong scalability" de la matrice 1 pour le premier problème industriel.

4.4. Validation des résultats numériques : Comparaison des résultats séquentiels et parallèles

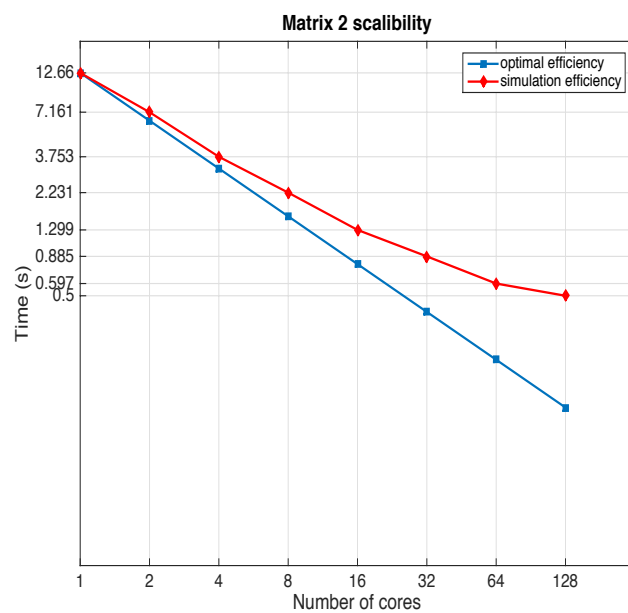


FIGURE 4.11 – "Strong scalability" de la matrice 2 pour le premier problème industriel.

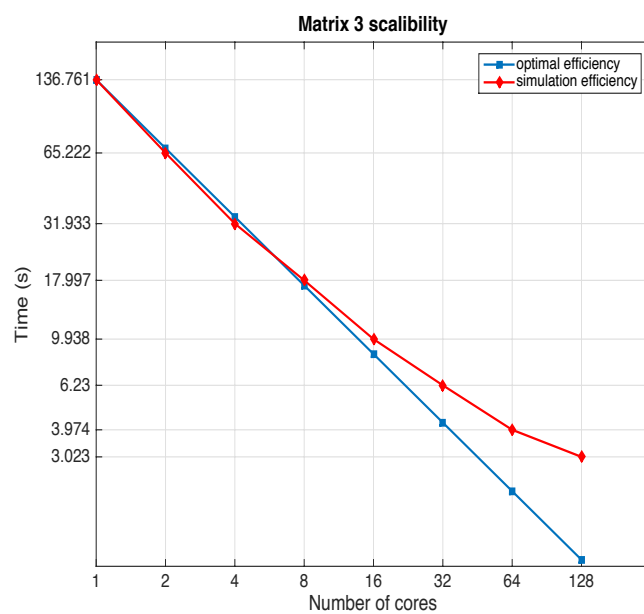


FIGURE 4.12 – "Strong scalability" de la matrice 3 pour le premier problème industriel.

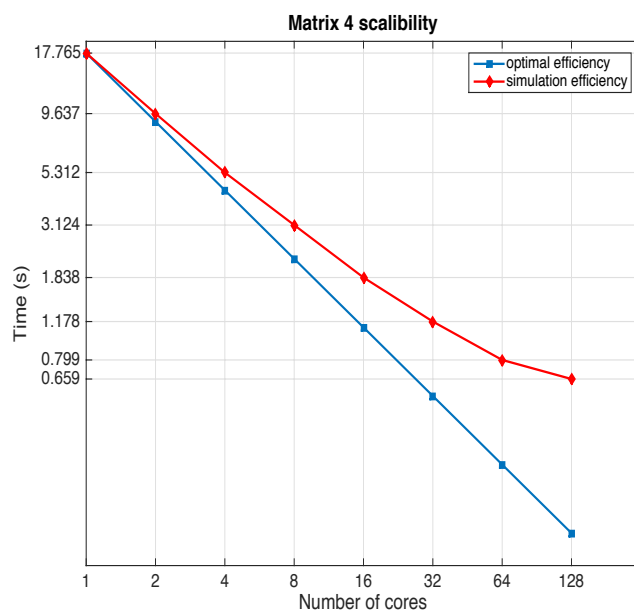


FIGURE 4.13 – "Strong scalability" de la matrice 4 pour le premier problème industriel.

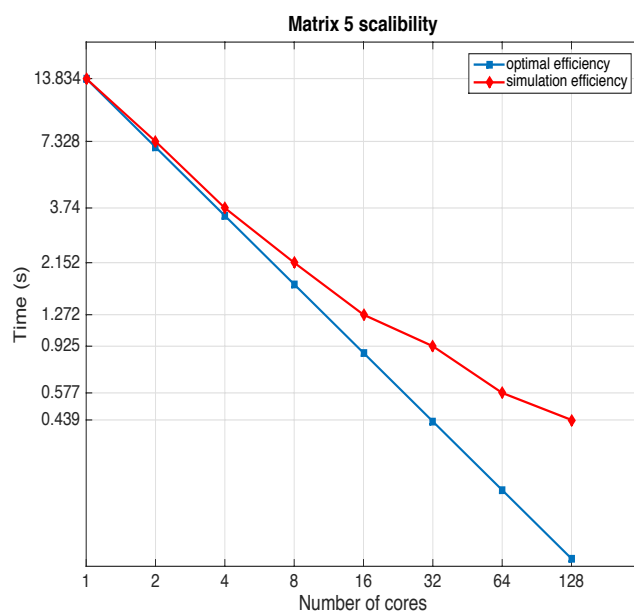


FIGURE 4.14 – "Strong scalability" de la matrice 5 pour le premier problème industriel.

On remarque que le temps de calcul des matrices diminue avec une moyenne de 30 fois avec 128 coeurs. La diminution du temps du calcul est d'une moyenne de 2 fois en passant de 1 coeur à 2 coeurs, puis elle diminue à une moyenne de 1.8 en passant de 2 à 4 coeurs, puis 1.8 en passant de 4 coeurs à 8 coeurs, puis 1.6 en passant de 8 coeurs à 16 coeurs, puis 1.5 fois en passant de 16 coeurs à 32 coeurs, puis 1.4 en passant de 32 coeurs à 64 coeurs, puis 1.3 en passant de 64 coeurs à 128 coeurs.

On a comparé le code séquentiel et parallèle avec des autres moules et on a obtenu toujours les mêmes solutions. Dans la suite, on utilise l'algorithme parallèle pour des moules de plus grande taille qui ne marche pas avec le code séquentiel.

4.5 Deuxième cas industriel

Dans cette section, on modélise le remplissage d'un moule industriel de plus grande taille, la hauteur maximale dans ce moule est de 1.7 m ; tandis que le plus grand diamètre est de 1.56 m . La géométrie du moule est donnée par la figure (4.15).

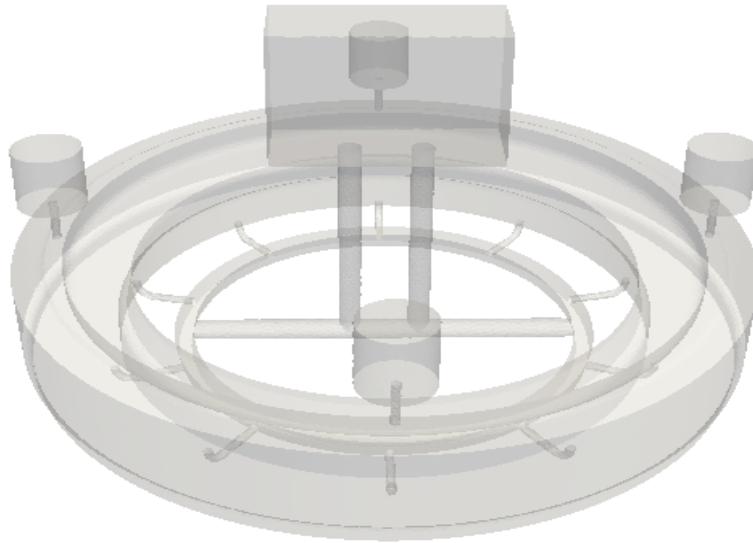


FIGURE 4.15 – La deuxième moule de l'industrie

La densité de l'air est 1 Kg.m^{-3} , la viscosité cinématique est $1.818e^{-3}\text{ m}^2.\text{s}^{-1}$ et la viscosité dynamique est $1.818e^{-3}\text{ Kg.m}^{-1}.\text{s}^{-1}$; tandis que la densité de l'aluminium liquide est 2385 Kg.m^{-3} , la viscosité cinématique est $1e^{-4}\text{ m}^2.\text{s}^{-1}$ et la viscosité dynamique est $2.385e^{-5}\text{ Kg.m}^{-1}.\text{s}^{-1}$. De plus, $U_{\text{in}} = 0.25\text{ m/s}$, $h = 0.02$, $\Delta t = 3h$.

Le maillage considéré contient 2,646,497 tétraèdres et 468,362 sommets.

On a utilisé le code parallèle avec 20 recouvrements pour le calcul des intégrales qui contiennent les courbes caractéristiques.

Les figures (4.16, 4.17, 4.18, 4.19, 4.20, 4.21) nous montrent l'évolution de l'interface approchée aux instants $t = 0 \text{ s}$, $t = 3 \text{ s}$, $t = 6 \text{ s}$, $t = 9 \text{ s}$, $t = 12 \text{ s}$, $t = 15 \text{ s}$.

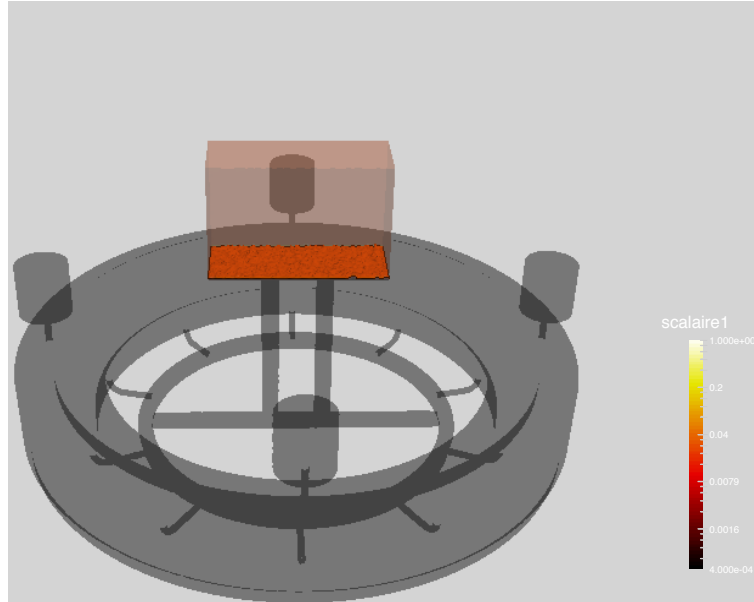


FIGURE 4.16 – Interface approchée à $t=0 \text{ s}$

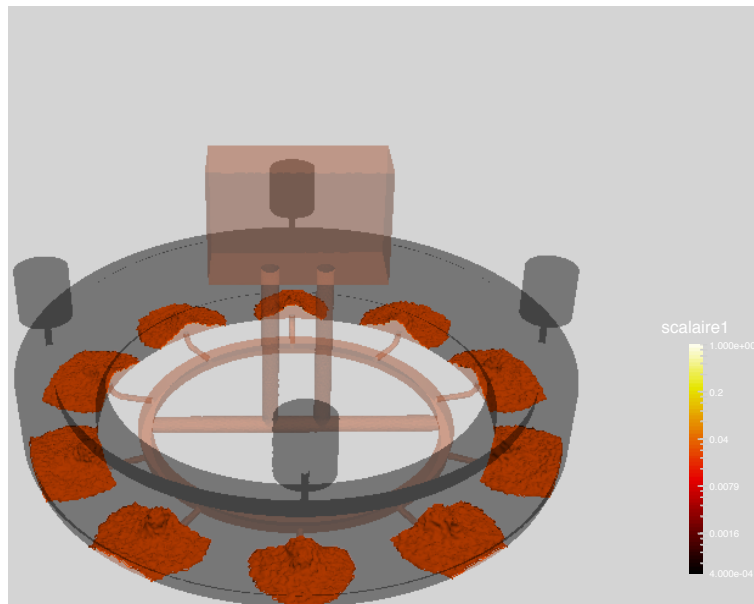


FIGURE 4.17 – Interface approchée à $t=3 \text{ s}$

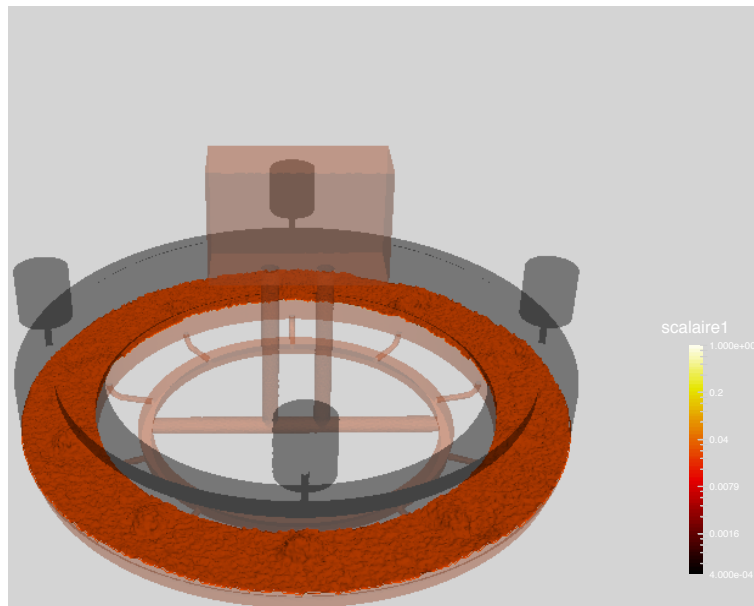


FIGURE 4.18 – Interface approchée à $t=6$ s

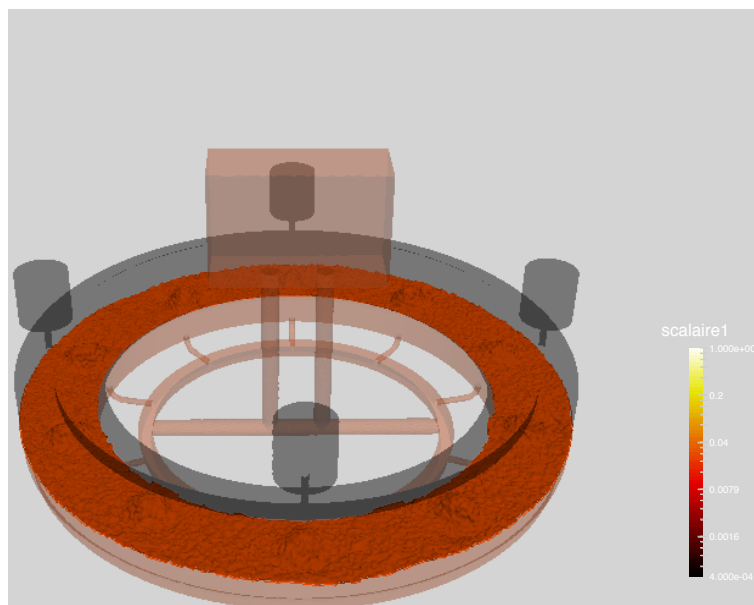


FIGURE 4.19 – Interface approchée à $t=9$ s

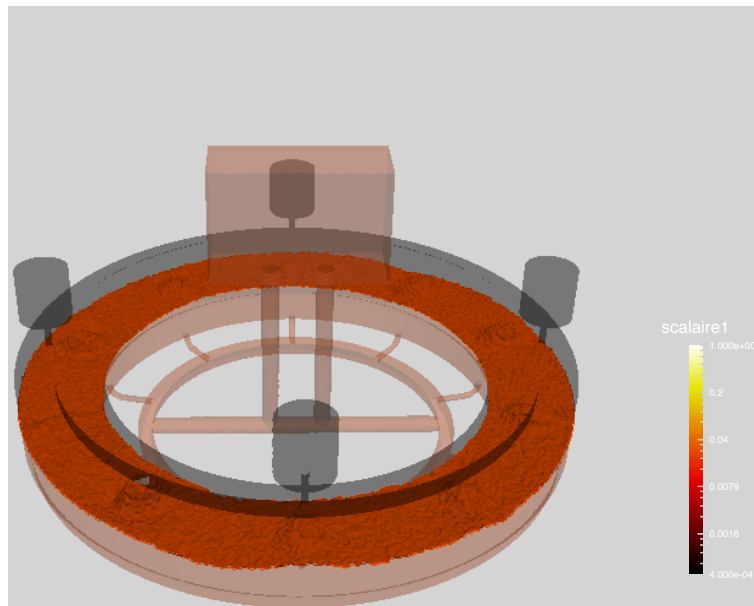


FIGURE 4.20 – Interface approchée à $t=12$ s

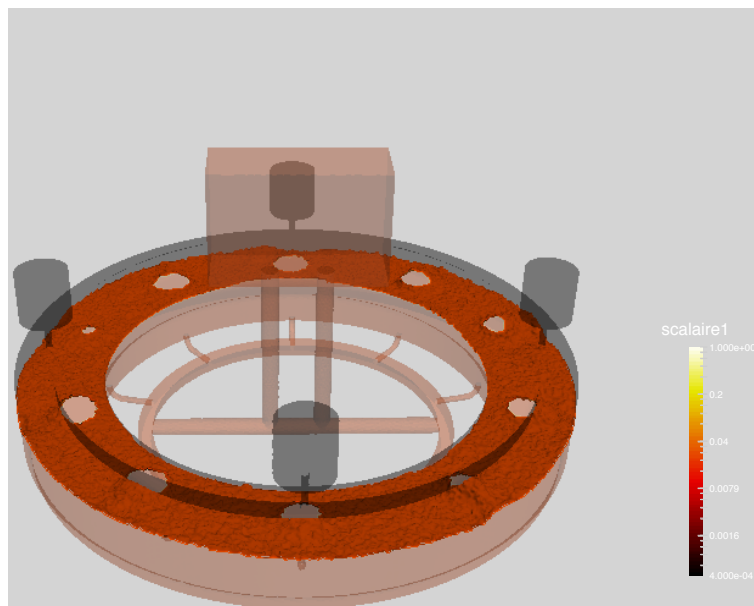


FIGURE 4.21 – Interface approchée à $t=15$ s

La figure (4.22) nous montre la variation de l'erreur en volume en fonction du temps durant le remplissage du moule.

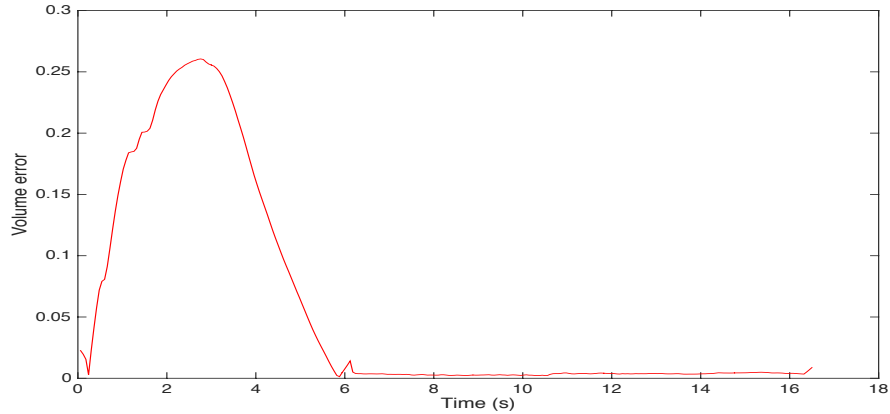


FIGURE 4.22 – Variation de l'erreur en volume en fonction du temps.

Il est important de noter dans ce cas qu'on a trouvé une difficulté à appliquer l'algorithme de conservation de masse dans les tuyaux. A ce niveau, la diffusion numérique est grande à chaque pas de temps, même avec des pas de temps plus petites. On a commencé la conservation de masse quand le fluide est arrivé en dehors des tuyaux. Pour corriger ces problèmes de diffusion numériques dans les tuyaux, on désire dans le futur faire un couplage 1D-3D de l'algorithme. On approche les tuyaux à des segments en 1D et on attend une diminution considérable de la diffusion numérique.

Dans les figures (4.23), (4.24), (4.25), (4.26) et (4.27) on présente le temps de calcul en échelle logarithmique de la première itération des cinq matrices de l'algorithme du remplissage en faisant évoluer le nombre des coeurs de 2 à 128 coeurs.

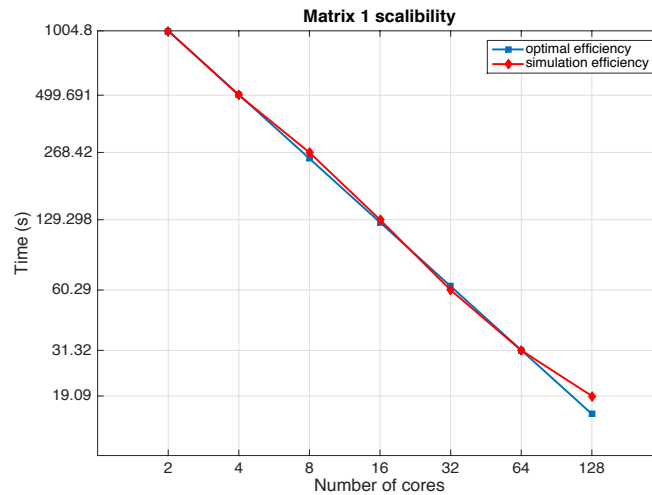


FIGURE 4.23 – "Strong scalability" de la matrice 1 pour le deuxième problème industriel.

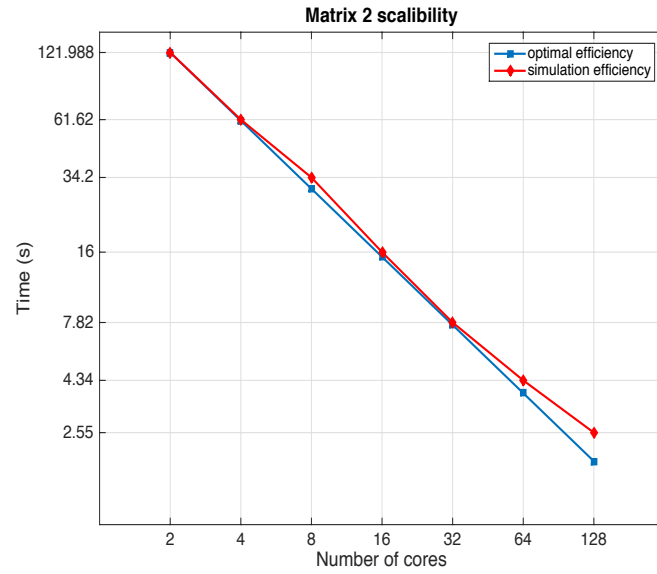


FIGURE 4.24 – "Strong scalability" de la matrice 2 pour le deuxième problème industriel.

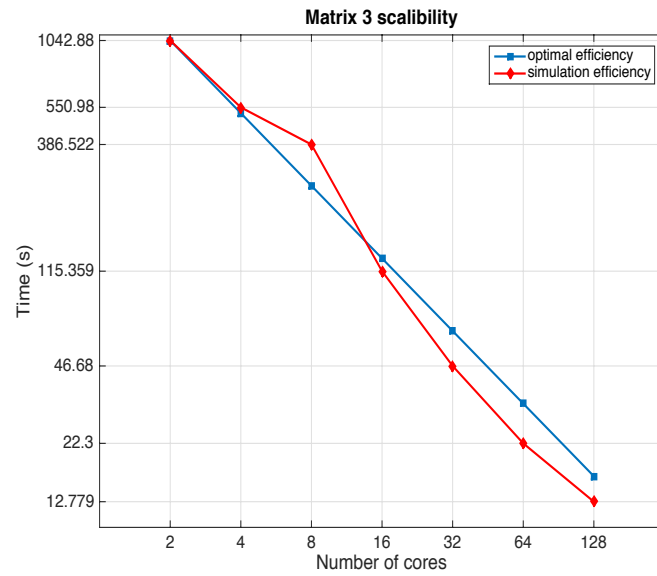


FIGURE 4.25 – "Strong scalability" de la matrice 3 pour le deuxième problème industriel.

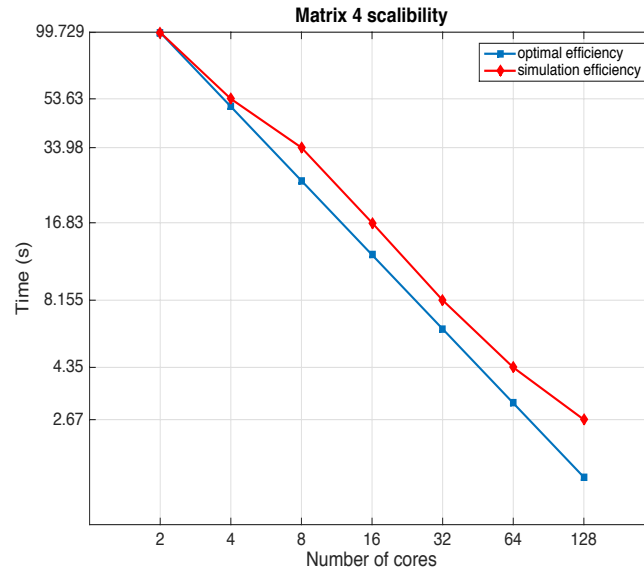


FIGURE 4.26 – "Strong scalability" de la matrice 4 pour le deuxième problème industriel.

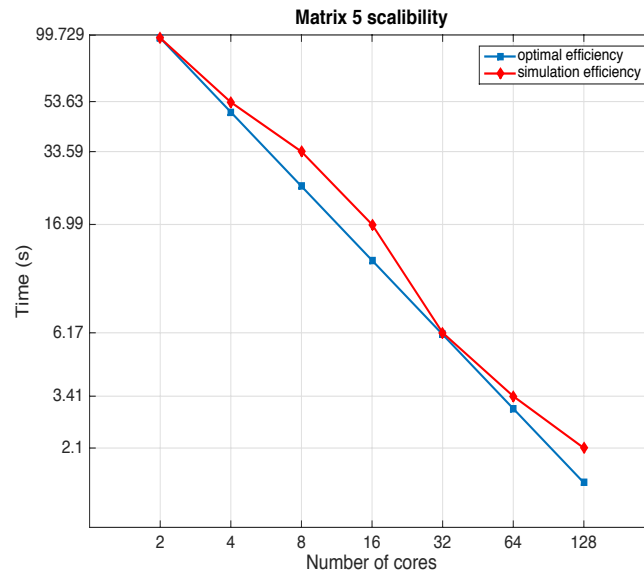


FIGURE 4.27 – "Strong scalability" de la matrice 5 pour le deuxième problème industriel.

De même, on remarque que le temps de calcul des matrices diminue avec une moyenne de 52 fois de 2 à 128 coeurs pour la première matrice, de 47 fois pour la deuxième, de 104 fois pour la troisième, de 37 fois pour la quatrième et de 46 fois pour la cinquième matrice. La diminution du temps du calcul est d'une moyenne de 1.9 fois en passant de 2 coeur à 4 coeurs, puis elle diminue à 1.7 fois en passant de 4 à 8 coeurs, puis 2.5 fois en passant de 8 coeurs à 16 coeurs, puis 2.5 fois en passant de 16 à 32 coeurs, puis 1.9 fois en passant de 32 coeurs à 64 coeurs, puis 1.7 fois en passant de 64 coeurs à 128 coeurs.

La forte "scalability" en temps de calcul diminue en passant à un nombre de coeurs élevé. Ce problème vient d'un part de la présence du recouvrement de Schwarz. D'autre part, dans les méthodes de Schwarz à un niveau, il y a un manque d'échange global d'informations. Les données sont échangées seulement d'un sous-domaine à ses voisins directs, mais la solution dans chaque sous-domaine dépend du second membre dans tous les sous-domaines. Les résultats théoriques et numériques montrent que les méthodes de décomposition de domaine basées uniquement sur la résolution des sous-domaines locales ne sont pas proportionnelles par rapport au nombre de sous-domaines. Ceci peut être réparé par l'utilisation d'une méthode à deux niveaux dont le préconditionneur est enrichi par la résolution d'un problème grossier dont la taille est de l'ordre du nombre de sous-domaines.

Dans la figure (4.28) on présente le temps de calcul de la simulation avec le code parallèle avec 128 coeurs. On remarque que la simulation prend à peu près 5 heures.

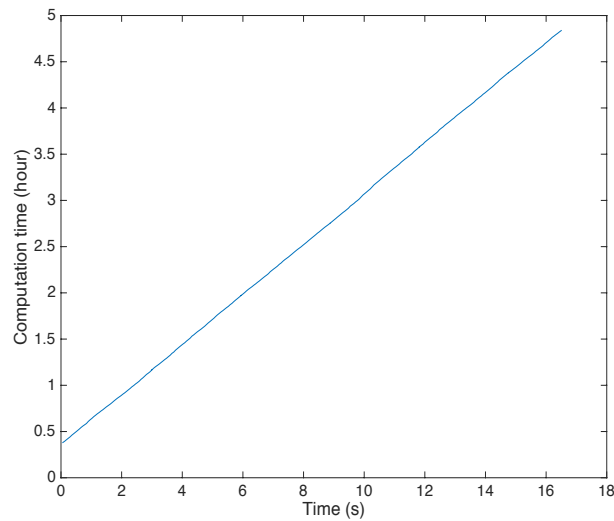


FIGURE 4.28 – Temps de calcul de la simulation du deuxième cas industriel avec le code parallèle.

Chapitre 5

Conclusions et perspectives

Sommaire

5.1	Conclusions	109
5.2	Perspectives	109

5.1 Conclusions

L'objectif de notre étude était de modéliser la phase remplissage en fonderie en particulier pour des moules trois dimensionnels industrielles.

Dans ce travail, on a proposé un algorithme pour le transport de l'interface par la vitesse de fluide pour un écoulement diphasique incompressible visqueux non-miscible par la méthode de caractéristiques. Cet algorithme a été validé par l'expérience de l'écroulement de barrage de Martin et Moyce et par le benchmark de Campbell. Les résultats obtenus sont en accord avec les expériences. Dans une autre étape, on a fait une parallélisation du code pour réduire le temps de calcul pour les moules à moyenne taille et pour pouvoir modéliser les moules de plus grande taille.

5.2 Perspectives

Pour améliorer la qualité de l'algorithme proposé, plusieurs pistes s'offrent dans le futur :

- a) Améliorer l'étape de reinitialisation, il existe dans la littérature des algorithmes qui effectue la reinitialisation dans un voisinage de l'interface. Ce qui réduit énormément le coût de calcul. Dans nos applications, l'état d'équilibre a été atteint en un nombre des itérations qui ne dépasse pas 6 itérations ce qui a été acceptable.
- b) Prendre en considération la force de tension pour détecter les bulles qui se forment sur l'interface lors des changements topologiques.
- c) Faire un couplage 1D-3D pour la modélisation des tuyaux-moules pour réduire la diffusion numérique dans les tuyaux.
- d) Prendre en considération les changements thermiques entre les deux phases des fluides et modéliser la phase de solidification.
- e) Améliorer le calcul parallèle en utilisant le calcul parallèle à deux niveaux avec des conditions optimisées ce qui permet d'augmenter la communication globale d'informations entre les sous-domaines. Par conséquent, la performance de l'algorithme de calcul parallèle augmente en faisant évoluer le nombre de processeurs.
- f) Introduire de modèles de turbulence. Ce type de modèle serait intéressant dans l'étude des écoulements à plus haut "Reynolds number". Ce qui permettrait de simuler les effets de la turbulence qui donnent les meilleurs résultats pour la modélisation des métaux liquide-air [135].

Bibliographie

Bibliographie

- [1] H. ABBOUD, Schémas à deux-grilles pour la résolution du problème de Navier-Stokes instationnaire incompressible. *Mathématiques [math]. Université Pierre et Marie Curie - Paris VI; Université Saint-Joseph, Beyrouth (2006)*
- [2] Y. ACHDOU AND J.L. GUERMOND , Convergence analysis of a finite element projection/Lagrange-Galerkin method for the incompressible Navier- Stokes equations , *SIAM J. Numer. Anal.* 37, no. 3, 799-826(2000).
- [3] P. R. AMESTOY, I. S. DUFF AND J.-Y. L'EXCELLENT, Multifrontal Parallel Distributed Symmetric and Unsymmetric Solvers, *Comput. Methods in Appl. Mech. Eng.*, Vol. 184, pp. 501-520(2000).
- [4] P. R. AMESTOY, A. GUERMOUCHE, J.-Y. L'EXCELLENT AND S. PRALET, Hybrid Scheduling for the Parallel Solution of Linear Systems, *Parallel Computing*, Vol. 32, No. 2, pp. 136-156(2006).
- [5] Y. AMIRAT, D. BRESCH, J. LEMOINE, J. SIMON , Effect of rugosity on a ow governed by stationary Navier- Stokes equations, *Quart. Appl. Math.* 59, no. 4, 769-785(2001).
- [6] D.M. ANDERSON, G.B. MCFADDEN, A.A. WHEELER, Diffuse-interface methods in fluid mechanics, *Annu. Rev. Fluid Mech.*,30 : 139-165(1998).
- [7] N. ASHGRIZ AND J.Y. POO, FLAIR, Flux line-segment model for advection and interface reconstruction, *J. Comput. Phys.* 93, 449-468(1991).
- [8] G.BARLES, Solutions de viscosité et équations elliptiques du deuxième ordre , *Cours, Université de tours(1997)*.
- [9] G.K. BATCHELOR , An Introduction to Fluid Dynamics, *Cambridge University Press, 1967*.
- [10] G.S. BEAVERS, D.D. JOSEPH, Boundary conditions at a naturally permeable wall, *J. Fluid Mech.* 30-01,pp. 197-207(1967).
- [11] J.B. BELL AND D.L. MARCUS, A second-order projection method for variable-density flows , *J. Comput. Phys.* 101, 334-348(1992).
- [12] S. H. BENTON , The Hamilton-Jacobi equation : A global approach, *Academic Press, New York(1977)*.

-
- [13] J.P. BENQUÉ, B. IBLER, A. KERAMSI, G. LABADIE, A finite element method for the Navier- Stokes equations, Proceedings of the third international conference on finite elements in flow problems, Banff. Alberta, Canada , *10-13 June (1980)*.
 - [14] K. BOUKIR ET AL , A Hight Order Characteristics/Finite Elements Method For The Incompressible Navier-Stokes Equations, Internat , *J. Numer. Methods Fluids, Vol. 25, pp. 1421-1454(1997)*.
 - [15] A. BOURLIOUX, A coupled level-set volume-of-fluid algorithm for tracking material interfaces, *In Sixth International Symposium on Computational Fluid Dynamics, Lake Tahoe, California, United States(1995)*.
 - [16] J. BRACKBILL, D.B. KOTHE AND C. ZEMACH, A continuum method for modeling surface tension , *J. Comput. Phys. 100, 335-354(1992)*.
 - [17] F. BREZZI, J. PITKARANTA , On the Stabilization of Finite Element Approximations of The Stokes Problem, in : W. Hackbush (Ed.), Efficient Solution of Elliptic Systems, *Notes on Numerical Fluid Mechanics, vol. 10, Friedr. Vieweg Sohn, Braunschweig(1984)*.
 - [18] M.O. BRISTEAU, R. GLOWINSKI AND J. PÉRIAUX, Numerical methods for the Navier-Stokes equations , *Application to the simulation of compressible and incompressible viscous flows. Comput. Phys. Rep. 6, 73-187(1987)*.
 - [19] A. BROOKS, T. HUGHES, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering 32 (1-3), 199-259(1982)*.
 - [20] D. BUCUR, E. FEIREISL, S. NECASOVA, J. WOLF , On the asymptotic limit of the Navier-Stokes system on domains with rough boundaries , *J. Differential Equations 244, no. 11, 2890-2908(2008)*.
 - [21] D. BUCUR, E. FEIREISL, S. NECASOVA, Boundary behavior of viscous fluids : influence of wall roughness and friction-driven boundary conditions, *Arch. Ration. Mech. Anal. 197, no. 1, 117-138(2010)*.
 - [22] M. BULICEK, J. MALEK, K. R. RAJAGOPAL, Navier's slip and evolutionary Navier-Stokes-like systems with pressure and shear-rate dependent viscosity , *Indiana Univ. Math. J. 56, no. 1, 51-85(2007)*.
 - [23] X.C CAI, C. FARHAT, AND M. SARKIS, A minimum overlap restricted additive Schwarz preconditioner and applications to 3D flow simulations, *Contemporary mathematics, 218 :479-485(1998)*.
 - [24] X.C CAI AND M. SARKIS, A restricted additive Schwarz preconditioner for general sparse linear systems, *Siam journal on scientific computing, 21 :239-247(1999)*.
 - [25] P. JOLIVET, F. HECHT, F. NATAF AND C. PRUD'HOMME, Scalable Domain Decomposition Preconditioners For Heterogeneous Elliptic Problems, *Proceeding SC'13 Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis Article No. 80*.

-
- [26] Y.C. CHANG, T.Y.HOU, B. MERRIMAN AND S. OSHER, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, *J. Comput. Phys.* 124 :449-464 (1996)
 - [27] L. CHEN, S.V. GARIMELLA, J.A. REIZES AND E. LEONARDI, The development of a bubble rising in a viscous liquid., *J. Fluid Mech.* 387, 61-69(1999).
 - [28] C. CHEVALIER AND F. PELLEGRINI. PT-SCOTCH, a tool for efficient parallel graph ordering, *Parallel computing*, 6-8(34) :318-331(2008).
 - [29] A.J. CHORIN, Numerical solution of the Navier-Stokes equations, *Math. Comp.* 22, 745-762(1968).
 - [30] A.J. CHORIN, On the convergence of discrete approximations to the Navier- Stokes equations, *Math. Comp.* 23, 341-353(1969).
 - [31] R. CODINA , Stabilized fractional step finite element methods for incompressible flows , *proceedings of European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2000)*, compact-disk.
 - [32] G.-H. COTTET AND P. PONCET , Advances in direct numerical simulations of 3D wall-bounded flows by vortex-in-cell methods, *Journal of Computational Physics*, 193,136-158, (2003).
 - [33] M. G. CRANDALL AND P. L. LIONS , Viscosity solutions of Hamilton-Jacobi equations , *Trans. Amer. Math. Soc.* 277, 1-42(1983).
 - [34] W.P. CROWLEY, FLAG, A free-Lagrange method for numerically simulating hydrodynamic flows in two dimensions , *In : Lecture Notes in Physics 8, Springer-Verlag, New York*, 37-43 (1971).
 - [35] C. CUVELIER, R.M. SCHULKES, Some numerical methods for the computation of capillary free boundaries governed by the Navier-Stokes equations, *D. Reidel Publishing Company, Dordrecht*(1986).
 - [36] C. CUVELIER, A. SEGAL AND A.A. VAN STEENHOVEN, Finite Element Methods and Navier-Stokes Equations, *D. Reidel Publishing Company, Dordrecht*(1986).
 - [37] T. A. DAVIS, AND I. S. DUFF, A Combined Unifrontal/Multifrontal Method for Unsymmetric Sparse Matrices, *ACM Trans. Math. Softw.*, Vol. 25, No. 1, pp. 1-19(1997).
 - [38] A.DERVIEUX, F.THOMASSET, A finite element method for the simulation of a Rayleigh-Taylor instability, in Approximation Methods for Navier-Stokes problems, *Lecture Notes in Mathematics ; 771, Springer- Verlag, Berlin*,145-158(1980).
 - [39] C. DAPOGNY, Shape optimization, level set methods on unstructured meshes and mesh evolution, *Thèse Université Paris 7-Denis Diderot et Université Pierre et Marie Curie*(2013).
 - [40] V. DOLEAN, P. JOLIVET, F. NATAF, An Introduction to Domain Decomposition Methods : algorithms, theory and parallel implementation, *Master. France.* 2015.

-
- [41] J. DONEA, S. GIULIANI, H. LAVAL AND L. QUARTAPELLE , Finite element solution of the unsteady Navier-Stokes equations by a fractional step method , *Comput. Meth. Appl. Mech. Engrg.* 30, 53-73 (1982).
 - [42] A. DOUGLIS, The continuous dependence of generalized solutions of nonlinear partial differential equations upon initial data , *Comm. Pure Appl. Math.* 14, 267-284(1961).
 - [43] M.S. ENGELMAN AND R.L. SANI, Finite element simulation of incompressible fluid flow with a free/moving surface, *Pineridge, Swansea*, 389-400(1983).
 - [44] D. ENRIGHT, R. FEDKIW, J. FERZIGER, AND I. MITCHELL, A hybrid particle level set method for improved interface capturing , *Journal of Computational Physics*, 183 : 83-116, (2002).
 - [45] A. ESMAEELI AND G. TRYGGVASON, Direct numerical simulations of bubbly flows. Part 1. Low Reynolds number arrays, *J. Fluid Mech.* 377, 313-345(1998)
 - [46] A. ESMAEELI AND G. TRYGGVASON, Direct numerical simulations of bubbly flows. Part 2. Moderate Reynolds number arrays, *J. Fluid Mech.* 385, 325-358(1999).
 - [47] W. H. FLEMING, The Cauchy problem for degenerate parabolic equations , *J. Math. Mech.* 13 (1964), 987-1008.
 - [48] W. H. FLEMING, The Cauchy problem for a nonlinear first order partial differential equation , *J. Differential Equations* 5 (1969), 5 15-530.
 - [49] W. H. FLEMING , Nonlinear partial differential equations -probabilistic and game theoretic methods , . *Problems in Nonlinear Analysis, CIME, Ed. Cremonese, Roma*, 1971.
 - [50] A. FRIEDMAN , The Cauchy problem for first order partial differential equations , *Indiana Univ. Math. J.* 23 (1973), 27-40.
 - [51] M.J. FRITTS AND J.P. BORIS, The Lagrangian solution of the transient problems in hydrodynamics using a triangular mesh, *J. Comput. Phys.* 31, 173-215(1979).
 - [52] D.E. FYFE, E.S. ORAN AND M.J. FRITTS, Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh, *J. Comput. Phys.* 76, 349-384(1988).
 - [53] G.FOURESTEY , Une méthode des caractéristiques d'ordre deux sur maillages mobiles pour la résolution des équations de Navier-Stokes incompressible par éléments finis , *Rapport de recherche, Thème 4, Numéro 4448*(2002).
 - [54] P.GALDI, W.LAYTON, Approximation of the larger eddies in fluid motion ii : a model for space filtered flow, *Math. Models Methods Appl.Sci.* 10 (3), 343-350(2000).
 - [55] M. J. GANDER, Schwarz methods over the course of time, *Electronic transactions on numerical analysis*, 31 :228-255(2008).
 - [56] R.H. GALLAGHER, G.F. CAREY, J.T. ODEN, O.C. ZIENKIEWICZ , Finite Elements in Fluids , Vol. 6, *J. Wiley and sons*(1985).
 - [57] M. J. GANDER, Schwarz methods over the course of time, *Electronic transactions on numerical analysis*, 31 :228-255(2008).

-
- [58] J. GLIMM, O. MCBRYAN, R. MENIKOFF AND D.H. SHARP, Front tracking applied to Rayleigh-Taylor instability , *SIAM J. Sci. Stat. Comput.*7, 230-251(1986).
 - [59] J. GLIMM, J. GROVE, B. LINDQUIST, O. MCBRYAN AND G. TRYGGVASON, The bifurcation of tracked scalar waves, *SIAM J. Sci. Stat. Comput.*9, 61-79(1988).
 - [60] R. GLOWINSKI AND P. LE TALLEC, Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics , *SIAM, Philadelphia*(1989).
 - [61] R. GLOWINSKI AND O. PIRONNEAU , Finite element methods for Navier-Stokes equations, *Annu. Rev. Fluid Mech.* 24,167-204(1992).
 - [62] R. GLOWINSKI, T.W. PAN, T.I. HESLA, D.D. JOSEPH, J. PERIAUX , A distributed Lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies : application to particulate flow , *Comput. Methods Appl. Mech. Engrg.* 184, no. 2-4,241-267(2000).
 - [63] P.M. GRESHO AND S.T. CHAN, On pressure boundary conditions for the incompressible Navier-Stokes equations, *Int. J. Num. Meth. Fluids* 7,1111- 1145(1987).
 - [64] P.M. GRESHO, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix , *Part 1 : Theory. Int. J. Num. Meth. Fluids* 11, no. 5, 587-620(1990).
 - [65] P.M. GRESHO AND S.T. CHAN, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix , *Part 2 : Implementation. Int. J. Num. Meth. Fluids* 11, no. 5, 621-659(1990).
 - [66] P.M. GRESHO AND R.L. SANI (IN COLLABORATION WITH M.S. ENGELMAN), Incompressible Flow and the Finite Element Method, *John Wiley and Sons*(1998).
 - [67] J.L. GUERMOND AND L. QUARTAPELLE , On the approximation of the unsteady Navier-Stokes equations by finite element projection methods, *Numer. Math.* 80, 207-238(1998).
 - [68] J.L. GUERMOND AND L. QUARTAPELLE , On stability and convergence of projection methods based on pressure Poisson equation , *Int. J. Num. Meth. Fluids* 26,1039-1053(1998).
 - [69] J.L. GUERMOND, P. MINEV , JIE SHEN , An overview of projection methods for incompressible flows , *Comput. Methods Appl. Mech. Engrg.* 195,6011-6045(2006).
 - [70] M.D. GUNZBURGER, Finite Element Methods for Viscous Incompressible Flows , *A Guide to Theory, Practice and Algorithms. Academic Press, San Diego*(1989).
 - [71] P. K. GUPTA AND K. V. PAGALTHIVARTHI, Application of Multifrontal and GMRES Solvers for Multisize Particulate Flow in Rotating Channels, *Prog. Comput Fluid Dynam.*, Vol. 7, pp. 323-336(2007).
 - [72] HARLOW F.H., WELCH J.E., Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* ; 8 : 2182-2189(1965).

-
- [73] S. E. HIEBER, P. KOUMOUTSAKOS , A Lagrangian particle level set method, *Journal of Computational Physics* 210, 342-367(2005).
 - [74] C.W. HIRT AND J.L. COOK, Calculating three-dimensional flows around structures and over rough terrain, *J. Comput. Phys.* 10 : 324-340(1972).
 - [75] HIRT C.W. AND NICHOLS B.D., Volume of fluid (VOF) method for the dynamics of free boundaries., *Journal of Computational Physics*, 39 :201-225 (1981).
 - [76] J.M. HYMAN , Numerical methods for tracking interfaces, *Physics* 12, 396-407(1984).
 - [77] M. HAYASHI, K. HATANAKA AND M. KAWAHARA, Lagrangian finite element method for free surface Navier-Stokes flow using fractional step methods., *Int. J. Num. Meth. Fluids* 13,805-840(1991).
 - [78] F. HECHT , New development in FreeFem++, *J. Numer. Math.* 20, no. 3-4, 251-265. 65Y15(2012).
 - [79] C.W. HIRT, J.L. COOK AND T.D. BUTLER , A Lagrangian method for calculating the dynamics of an incompressible fluid with free surface , *J. Comput. Phys.* 5,103-124(1970).
 - [80] W. JAGER AND A. MIKELIC , On the interface boundary condition of Beavers, Joseph, and Saffman, *SIAM J. Appl. Math.* 60 (2000), no. 4, 1111-1127.
 - [81] W. JAGER AND A. MIKELIC, On the roughness-induced effective boundary conditions for an incompressible viscous flow , *J. Differential Equations* 170 (2001), no. 1, 96-122.
 - [82] H. JIA , Characteristic stabilized finite element method for the transient Navier-Stokes equations , *Comput. Methods Appl. Mech. Eng*, doi :10.1016/j.cma.2010.06.010 (2010).
 - [83] V. JOHN , Slip with friction and penetration with resistance boundary conditions for the Navier-Stokes equations numerical tests and aspects of the implementation, *Journal of Computational and Applied Mathematics* 147, 287-300(2002).
 - [84] G. KARYPIS AND V. K. METIS, unstructured graph partitioning and sparse matrix ordering system. version 2.0. Technical report , *Minneapolis, MN 55455 : University of Minnesota, Department of Computer Science, August 1995.*
 - [85] G.E. KARNIADAKIS, M. ISRAELI AND S.A. ORSZAG , High-order splitting methods for the incompressible Navier-Stokes equations , *Int . J. Comput. Phys.* 97, 414-443(1991).
 - [86] P. KJELLGREN , A semi-implicit fractional step finite element method for viscous incompressible flows , *Comput. Mech.* 20, 541-550(1997).
 - [87] S. N. KRUKOV , Generalized solutions of nonlinear first order equations and certain quasilinear parabolic equations, *Vestnik Moscov , Univ. Ser. I Mat. Meh.* 6, 67-74(1964).

-
- [88] S. N. KRUKOV, Generalized solutions of first order nonlinear equations in several independent variables. I , *Mat. Sb.* 70 (112) (1966), 394-415; II, *Mat. Sb. (N. S.)* 72 (114), 93-116(1967).
 - [89] S. N. KRUKOV , Generalized solution of the Hamilton-Jacobi equations of Eikonal type, *Math. USSR-Sb.* 2 (7), 406-446(1975).
 - [90] D. KUZMIN , Numerical Simulation of Reactive Bubbly Flows , *PhD thesis, University of Jyväskylä*, Jyväskylä, 1999.
 - [91] O.A. LADYZHENSKAYA, The Mathematical Theory of Viscous Incompressible Flow , 2nd ed. *Gordon and Breach, New York*(1969).
 - [92] H. LAVAL AND L. QUARTAPELLE , A fractional-step Taylor-Galerkin method for unsteady incompressible flows , *Int. J. Num. Meth. Fluids* 11 : 501-513(1990).
 - [93] R.W. LEWIS, A.S. USMANI AND J.T. CROSS , Efficient mould filling simulation in castings by an explicit finite element method, *Int. J. Num. Meth. Fluids* 20 : 493-506(1995).
 - [94] H. LAMB, Hydrodynamics., *Cambridge University Press, 1932*.
 - [95] P. L. LIONS , Generalized solutions of Hamilton-Jacobi equations , *Pitman Research Notes Series, Pitman, London*(1982).
 - [96] P.-L. LIONS, On the Schwarz alternating method. II. In Domain decomposition methods. Tony Chan, Roland Glowinski, Jacques Periaux, and Olof Widlund, editors., *SIAM, Philadelphia, PA, pages* 47-70(1989).
 - [97] P.-L. LIONS, On the Schwarz alternating method. III : a variant for nonoverlapping subdomains, *In First international symposium on domain decomposition methods for partial differential equations. Tony F. Chan, Roland Glowinski, Jacques Periaux, and Olof Widlund, editors. SIAM, Philadelphia, PA*(1990).
 - [98] J. LIU, S. KOSHISUKA, AND Y. OKA, A hybrid particle-mesh method for viscous, incompressible, multiphase flows , *Journal of Computational Physics*, 202 :65 ? 93, (2005).
 - [99] J-C. MARTIN ET M.J.MOYCE, Some gravity wave problems in the motion of perfect liquids, *Philos. Trans. Roy. Soc. London Ser. A*, 244, 231-334 (1952).
 - [100] S. MARUSIC, On the Navier-Stokes system with pressure boundary condition, *Ann. Univ. Ferrara Sez. VII Sci. Mat.* 53, no. 2, 319-331(2007).
 - [101] T. MENARD, S. TANGUY, AND BERLEMONT A. , Coupling level set / VOF / ghost fluid methods : validation and application to 3D simulation of the primary break-up of a liquid jet , *International Journal of Multiphase Flow*, 33 :510-524(2007).
 - [102] C.L.M.H. NAVIER, Sur les lois de l'équilibre et du mouvement des corps élastiques, *Mem. Acad. R. Sci. Inst. Vol. 6, France* 369(1827).
 - [103] W.F. NOH AND P.WOODWARD, SLIC (simple line interface calculation), *In Lecture Notes in Physics* 59, *Springer-Verlag, New York* 330-340(1976).

-
- [104] T. OKAMOTO AND M. KAWAHARA , Two-dimensional sloshing analysis by Lagrangian finite element method, *Int. J. Num. Meth. Fluids* 11,453-477(1990).
 - [105] E. OLSSON, G. KREISS, A conservative level set method for two phase flow, , *J. Comp. Phys* 210 :225-246(2005).
 - [106] S.A. ORSZAG, M. ISRAELI AND M.O. DEVILLE, Boundary conditions for incompressible flows, *J. Sci. Comput.* 1, no. 1,75-111(1986).
 - [107] S. OSHER AND J.A. SETHIAN, Fronts propagating with curvature-dependent speed : algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics*, 79 :12- 49(1988).
 - [108] D. PENG, B.MERRIMAN, S. OSHER , H.ZHAO, AND M.KANG, A PDE-based fast local level set method, *Journal of Computational Physics*, 155, pp. 410-438(1999).
 - [109] O. PIRONNEAU , On the transport-diffusion algorithm and its applications to the Navier-Stokes equations , 38(3) :309-332 (1982).
 - [110] O. PIRONNEAU , Méthodes des éléments finis pour les fluides , *Collection Recherches en Mathématiques Appliquées*, Masson, Paris(1988) .
 - [111] O. PIRONNEAU , Finite Element Methods for Fluids, *John Wiley and Sons, Chichester*(1989).
 - [112] O. PIRONNEAU, J. LIOU, AND T. TEZDUYAR, Characteristic-Galerkin and Galerkin/least-squares space-time formulations for the advection-diffusion equation with time-dependent domains., *Computer Methods in Applied Mechanics and Engineering*, 100 :117-141(1992).
 - [113] PIRONNEAU O.AND AND TABATA M., Stability and convergence of a Galerkin-characteristics finite element scheme of lumped mass type , *Int. J. Numer. Meth. Fluids* 64 :1240-1253 (2010).
 - [114] A. PROHL , Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations , *Advances in Numerical Mathematics. B.G. Teubner, Stuttgart*, 1997.
 - [115] E.G. PUCKETT, A.S. ALMGREN, J.B. BELL, D.L. MARCUS AND W.J. RIDER, A high order projection method for tracking fluid interfaces in variable density incompressible flows , *J. Comput. Phys.* 130, 269-282(1997).
 - [116] L. QUARTAPELLE , Solution of the Time-Dependent Incompressible Navier-Stokes Equations, *Birkh-sauser Verlag, Basel*(1993).
 - [117] A. QUARTERONI AND A. VALLI , Numerical Approximation of Partial Differential Equations , *Springer-Verlag, Berlin*(1994).
 - [118] P.E. RAAD AND R. BIDOAE, Three-dimensional Eulerian-Lagrangian marker and micro cell method for the simulation of free surface flows , *Journal of Computational Physics*, 203 :668-699, (2005).
 - [119] M. P. RAJU, Parallel Computation of Finite Element Navier- Stokes Codes Using MUMPS Solver, *International Journal of Computer Science Issues*, Vol. 4, No. 2, Sept.(2009).

-
- [120] J.D. RAMSHAW AND J.A. TRAPP , A numerical technique for low-speed homogeneous two-phase flow with sharp interfaces, *J. Comput. Phys.* 21 : 438-453(1976).
 - [121] R. RANNACHER , On Chorin's projection method for the incompressible Navier-Stokes equations, I : The Navier-Stokes Equations II : Theory and Numerical Methods, *Springer-Verlag, Berlin*, 167-183(1992).
 - [122] W.J. RIDER AND D.B. KOTHE , Reconstructing volume tracking, *J. Comput. Phys.* 14, 112-152(1998).
 - [123] M. RUDMAN , Volume-tracking methods for interfacial flow calculations, *Int. J. Num. Meth. Fluids* 24, 671-691(1997).
 - [124] Y. SAAD AND M. SCHULTZ, A generalized minimum residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 7, pp. 856-869(1986).
 - [125] Y. SAAD, Iterative Methods for Sparse Linear Systems, *SIAM Publishers*, 2nd edition(2000).
 - [126] E.SAEZ , Etude numérique du remplissage 3D en fonderie, *Thèse de l'Ecole des Mines de Paris* (2003).
 - [127] S. SARRA, The Method of Characteristics with applications to Conservation Laws. , *Journal of Online Mathematics and its Applications (JOMA)*, Vol 3, February (2003).
 - [128] SCHÄFER M., TUREK S., DURST F., KRAUSE E. AND RANNACHER R. Benchmark Computations of Laminar Flow Around a Cylinder, *Publisher : Vieweg+Teubner Verlag-Adress : Wiesbaden - pages 547-566, isbn=978-3-322-89849-4, doi=10.1007/978-3-322-89849-4 39* (1996)
 - [129] H. A. SCHWARZ, Über einen Grenzübergang durch alternierendes Verfahren., *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zurich*, 15 : 272-286 (1870).
 - [130] L.I. SEDOV, Mechanics of Continuous Media , Vol. 1, 2. *World Scientific Publishing Inc., River Edge, NJ*, 1997.
 - [131] J. SERRIN, Mathematical principles of classical uid mechanics, *Handbuch der Physik*, pp. 125-263, *Springer- Verlag*, 1959.
 - [132] J. SHEN AND R. TEMAM , A new fractional scheme for the approximation of incompressible flows , *Mat. Appl. Comput.* 8 , no.1, 3-22(1989).
 - [133] J. SHEN, Pseudo-Compressibility Methods for the Unsteady Incompressible Navier-Stokes Equation, in Proceedings of the 1994 Beijing Symposium on Nonlinear Evolution Equations and Infinite Dynamical Systems , pp. 68-78, *Ed. Boling Guo, Zhong-Shan University Press*(1997).
 - [134] J. SHEN, On a new pseudo-compressibility methods for the incompressible Navier-Stokes equations , *Appl. Numer. Math.* Vol 21, pp. 71-90(1996).
 - [135] SIRRELL B., HOLLIDAY M., CAMPBELL J., The benchmark test 1995,, *J. in : Proceedings of the 7th Conference on Modeling of Casting Welding and Solidification Processes, London, UK*, pp. 915-933 (1995).

-
- [136] W.SHYY , H.S. UDAYKUMAR, M.M. RAO, R.W. SMITH, Computational Fluid Dynamics with moving Boundaries,, *Taylor and Francis (1996)*.
 - [137] M.SUSSMAN, P.SMEREKA, S.OSHER, A level set approach for computing solutions to incompressible two-phase flows, *J. Comput. Phys.* ; 114 : 146-159(1994).
 - [138] M.SUSSMAN, A.ALMGREN, J.BELL, PH.COLELLA, L. HOWELL AND M.WELCOME, An adaptive level set approach for incompressible two-phase flows, *J. Comput. Phys.* 148 : 81-124(1999).
 - [139] M. SUSSMAN AND E.G. PUCKETT, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, *Journal of Computational Physics*, 162 :301-337(2000).
 - [140] A. SMALONSKI , Numerical Modeling of Two-Fluid Interfacial Flows , *PhD thesis University of Jyväskylä Finland (2001)*.
 - [141] M.SUSSMAN AND E.FATEMI, An efficient interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, *SIAM J. Sci. Comput.* 20 , no. 4, 1165-1191 (1999).
 - [142] T.T. M. TA , Modélisation des problèmes bi-fluides par la méthode des lignes de niveau et l'adaptation du maillage. *Thèse de doctorat. Université Pierre et Marie-Curie(2015)*.
 - [143] R. TEMAM, Sur l'approximation de la solution des équations de Navier- Stokes par la méthode des pas fractionnaires (I), *Arch. Rational Mech. Anal.* 32 : 135-153(1969).
 - [144] R.TEMAM , Sur l'approximation de la solution des équations de Navier- Stokes par la méthode des pas fractionnaires (II) , *Arch. Rational Mech. Anal.* 33 : 377-385 (1969).
 - [145] R. TEMAM, Navier-Stokes Equations , *Theory and Numerical Analysis. 3rd ed., North-Holland, Amsterdam, 1984*.
 - [146] A. TOSELLI AND O.WIDLUND, Domain decomposition methods -algorithms and theory, *Volume 34 of Springer Series in Computational Mathematics. Springer(2005)*.
 - [147] S. TUREK , A comparative study of time stepping techniques for the incompressible Navier-Stokes equations : From fully implicit nonlinear schemes to semi-implicit projection methods, *Int. J. Num. Meth. Fluids* 22, 987-1011(1996).
 - [148] S. TUREK , Efficient Solvers for Incompressible Flow Problems : An Algorithmic and Computational Approach, *Springer-Verlag, Berlin(1999)*.
 - [149] S. VINCENT, G. BALMIGERE, J.-P. CALTAGIRONE, AND E. MEILLOT, Eulerian-Lagrangian multiscale methods for solving scalar equations - Application to incompressible two-phase flows. , *Journal of Computational Physics*, 229 :73- 106, (2010).
 - [150] Z. WANG, J. YANG, B. KOO, AND STERN F , A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves, *Journal of Multiphase Flow*, 35 :227-246(2009).

-
- [151] Y. XIAO, Z. XIN, ZHOUPING, On the vanishing viscosity limit for the 3D Navier-Stokes equations with a slip boundary condition, *Comm. Pure Appl. Math.* 60, no. 7, 1027-1055(2007).
- [152] ZHANG H., ZHENG L.L., PRASAD V. AND HOU T.Y., A curvilinear level set formulation for highly deformable free surface problems with application to solidification, *Numer. Heat Transfer, part B*, 34 :120(1998).
- [153] ZHAO H.K., MERRIMAN B., OSHER S. AND WANG L., Capturing the behavior of bubbles and drops using the variational level set approach, *J. Comput. Phys.* 143 :495-518(1998).

Résumé

Dans cette thèse, on utilise des outils mathématiques et numériques pour modéliser les écoulements tridimensionnels incompressibles à surface libre instantionnaires. L'application industrielle visée est l'étude de la phase de remplissage des moules dans une fonderie. On développe un algorithme pour le transport de l'interface par la vitesse du fluide pour un fluide diphasique incompressible visqueux non-miscible de rapport de densité important en utilisant la méthode de caractéristiques pour traiter le problème de convection.

Il y a des défis majeurs dans le contexte de la modélisation des fluides diphasiques. Tout d'abord, on doit prendre en considération l'évolution de l'interface et de ses changements topologiques. Deuxièmement, on doit traiter la non-linéarité convective de l'interface et de l'écoulement. Troisièmement, les équations de Navier-Stokes et du transport doivent être munies des conditions aux bords appropriées. En outre, il faut traiter soigneusement les singularités géométriques et topologiques à travers l'interface en particulier dans le cas de rapport de densité et viscosité important. On doit également maintenir la résolution d'une interface d'épaisseur nulle durant les cas du pliage, la rupture et la fusion de l'interface. Quatrièmement, on doit respecter les propriétés physiques telles que la conservation de la masse pour tout écoulement d'un fluide incompressible. Cinquièmement, il faut toujours penser aux limitations du temps de calcul et de mémoire pour résoudre ce genre de problème dans les cas pratiques. Notre but est de trouver un schéma fiable capable de modéliser le remplissage des moules tridimensionnelles industrielles.

La première partie de cette thèse est dédiée à la description mathématique du schéma de transport de l'interface par la vitesse du fluide. Le mouvement des fluides est décrit par les équations de Navier-Stokes. L'interface est capturée par la fonction Level-Set. Le problème est discrétisé en espace par la méthode des éléments finis et en temps par la méthode de caractéristiques. Des conditions aux bords appropriées pour le problème du remplissage d'un moule sont introduites et un algorithme de calcul de la solution est présentée. Finalement, des résultats numériques montrent et valident l'efficacité du schéma proposé.

Dans la deuxième partie de cette thèse, on introduit une méthode de décomposition de domaine qui correspond à la discrétisation par la méthode des caractéristiques dans le but d'améliorer la performance de l'algorithme proposé lors de la modélisation du remplissage des moules industrielles à moyennes séries. Des résultats numériques de comparaison valident la précision du code parallèle.

MOTS CLÉS. Ecoulement diphasique, fonction Level-Set, méthode des éléments finis, conditions aux bords sous effet de la pression, méthode de projection.

Summary

In this thesis, we use mathematical and numerical tools to model three dimensional incompressible laminar flows with free surface. The described industrial application is the study of the mould filling phase in a foundry. We develop an algorithm for the transport of the interface by the fluid velocity for a viscous incompressible immiscible fluids of high density ratio in two-phase flow using the method of characteristics for the convection problem.

There are, however, major challenges in the context of two-phase flow modeling. First, we have to take into account the evolution of the interface and its topological changes. Second, we have to deal with the non-linearity for the convection of the flow and the interface. Third, we must assign appropriate boundary conditions to the flow and transport equations. In addition, care must be taken in treating the geometrical and topological singularities across the interface. We also have to maintain a sharp interface resolution, including the cases of interface folding, breaking and merging. Furthermore, we should respect the physical properties such as the mass conservation for any incompressible fluid flows. Finally, we have to keep in mind the limitations in the time of computation and memory to solve this kind of problem in practical cases. Our purpose is to find a reliable scheme able to model the filling of three dimensional industrial moulds.

The first part of the thesis is devoted to the mathematical description of the interface transport scheme by the fluid velocity. The fluids motion is described by the Navier-Stokes equations. The interface is captured by the Level-Set function. The problem is discretized by the characteristics method in time and finite elements method in space. The interface is captured by the Level-Set function. Appropriate boundary conditions for the problem of mould filling are investigated, a new natural boundary condition under pressure effect for the transport equation is proposed and an algorithm for computing the solution is presented. Finally, numerical experiments show and validate the effectiveness of the proposed scheme.

In the second part of the thesis, we introduce a domain decomposition method that suits the discretization by the method of characteristics in order to improve the performance of the proposed algorithm to model the filling phase for moulds of medium series. Numerical results of comparison validate the precision of the parallel code.

KEYWORDS. Two-phase flow, Level-Set function, finite element method, characteristics method, boundary condition under pressure effect, projection method.