



HAL
open science

Simulation numérique directe multiphasique de la déformation d'un alliage Al-Cu à l'état pâteux – Comparaison avec des observations par tomographie aux rayons X in situ en temps réel

Jean-François Zaragoci

► **To cite this version:**

Jean-François Zaragoci. Simulation numérique directe multiphasique de la déformation d'un alliage Al-Cu à l'état pâteux – Comparaison avec des observations par tomographie aux rayons X in situ en temps réel. Autre. Ecole Nationale Supérieure des Mines de Paris, 2012. Français. NNT : 2012ENMP0088 . pastel-00820727

HAL Id: pastel-00820727

<https://pastel.hal.science/pastel-00820727>

Submitted on 7 May 2013

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.

Ecole doctorale n°364 : Sciences Fondamentales et Appliquées

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École nationale supérieure des mines de Paris

Spécialité « Mécanique Numérique »

présentée et soutenue publiquement par

Jean-François ZARAGOCI

le 9 juillet 2012

**Simulation numérique directe multiphasique de la déformation d'un alliage Al-Cu
à l'état pâteux – Comparaison avec des observations par tomographie
aux rayons X in situ en temps réel.**

Directeur de thèse : **Michel BELLET, Charles André GANDIN**

Co-encadrement de la thèse : **Luisa SILVA**

Jury

M. Bernard BILLIA, Directeur de recherche CNRS, IM2NP, Université d'Aix-Marseille III
M. Michel RAPPAZ, Professeur, LSMX, Ecole Polytechnique Fédérale de Lausanne
M. Stéphane VINCENT, Maître de conférence, I2M, Institut Polytechnique de Bordeaux
M. Michel SUERY, Collaborateur bénévole, SIMaP, Institut National Polytechnique de Grenoble
M. Michel BELLET, Professeur, CEMEF, MINES ParisTech
M. Charles-André GANDIN, Directeur de Recherche CNRS, CEMEF, MINES ParisTech
Mme Luisa SILVA, Chargée de recherche, CEMEF, MINES ParisTech

Examinateur
Rapporteur
Rapporteur
Examinateur
Examinateur
Examinateur
Examinateur

**T
H
È
S
E**

- Remerciements -

Une thèse n'est pas seulement une expérience professionnelle, c'est aussi et surtout une aventure humaine, au cours de laquelle j'ai pu apprendre énormément de choses sur le fonctionnement de la recherche en France, sur les autres et sur moi-même. J'ai connu des moments de doute, des moments de joie, des moments de remise en question, des moments de motivation extrême suivis de moments de démotivation totale, des nuits blanches et des week-ends passés au laboratoire, des arrivées très matinales dans mon bureau en courant, curieux de voir le résultat de mes simulations, d'autres arrivées plus tardives sur la pointe des pieds et à reculons, ne sachant pas ce que j'allais pouvoir trouver à faire pour m'occuper, écrasé par l'ampleur de la tâche se dressant devant moi, des soirs à regarder l'horloge pour réaliser que le jour s'était écoulé beaucoup trop rapidement, des journées à regarder cette même horloge en espérant qu'elle s'accélère afin d'atteindre une heure décente pour pouvoir vaquer à d'autres activités, des moments à penser que je pourrais m'absenter plusieurs mois sans que personne ne s'en rende compte et que je ne serais jamais en mesure d'écrire ces remerciements et d'autres à penser que la mission qui était la mienne présentait un intérêt important pour de nombreuses personnes.

Fort heureusement, je n'étais pas seul pour préparer ce travail. Par soucis de perfection et surtout de procrastination, j'ai mis très longtemps à me motiver pour écrire ces remerciements. Pour le faire, j'ai évidemment dû revoir mes exigences à la baisse et il est fort possible que je fasse preuve d'ingratitude en oubliant de mentionner certaines personnes sans le soutien desquelles vous ne liriez pas ces lignes actuellement. Par conséquent, si quelqu'un se sent lésé à la lecture de ces remerciements, je l'incite à me contacter afin que j'essaie de réparer le terrible affront subi autour d'un verre de jus de fruits.

Je voudrais dans un premier temps témoigner ma gratitude à Michel Bellet, mon directeur de thèse. Alors qu'il est pourtant très occupé, j'ai adoré la disponibilité dont il fait preuve avec ses étudiants. Non content de posséder des qualités humaines et scientifiques indéniables, son réflexe de toujours laisser la porte de son bureau ouverte lorsque il est libre et l'impression qu'il donne de ne jamais être dérangé, et ce même lorsque je lui posais des questions totalement indignes d'un doctorant, sont plus qu'appréciables. Je le recommande vivement en tant qu'encadrant.

Avant de faire sa connaissance, les termes liquidus et solidus évoquaient irrémédiablement dans mon esprit le nom de deux légionnaires romains dans la bande dessinée Astérix. Grâce à lui, une requête faite sur un célèbre moteur de recherche et concernant la microstructure dendritique renvoie à présent parmi les premiers résultats une image tirée d'une de nos simulations. Ce changement, je le dois bien sûr à Charles-André Gandin, mon co-directeur de thèse. Même si nous nous sommes quelque peu éloignés de ses domaines de prédilection au cours de ce travail, sa présence à mes côtés a été primordiale. Sans son habileté à me mettre le couteau sous la gorge je suis notamment convaincu que je ne saurais pas encore aujourd'hui ce qu'est un élément fini. Que ses doctorants actuels se rassurent, je parlais évidemment au sens figuré.

Je tiens également à remercier Luisa Silva, ma troisième encadrante, grâce à qui j'ai pu finir ce travail à peu près dans les temps. Je me souviendrai en particulier de son aide précieuse

juste avant une réunion avec nos partenaires académiques. J'avais pu apprécier à cette occasion l'heure de modification tardive des fichiers utilisés pour lancer un calcul important. Je dois aussi ajouter que la capacité de Luisa à me faire garder les pieds sur terre a été vitale. En effet, au moment où j'ai réalisé que les outils utilisés dans notre librairie de calcul éléments finis présentaient un certain nombre de limitations et que j'ai eu envie de tout redévelopper en employant une autre méthode, elle m'a à raison conseillé d'insister avec nos outils actuels, ce qui nous a permis d'obtenir des résultats satisfaisants en un délai beaucoup plus court.

J'aimerais également manifester ma sympathie pour Bernard Billia, Michel Rappaz, Stéphane Vincent et Michel Suéry, les membres de mon jury de thèse qui ont eu le courage d'examiner ce travail, ainsi que pour les partenaires du projet SIMUZAL Luc Salvo et Bastien Mireux, dont la collaboration a été précieuse.

Je souhaite aussi exprimer ma reconnaissance envers les membres permanents du groupe Calcul Intensif en Mise en Forme des Matériaux, et plus particulièrement Hugues Digonnet, qui m'a aidé à de multiples reprises, Thierry Coupez, chef du groupe avec qui je regrette de ne pas avoir davantage interagi, Marc Bernacki, qui m'a permis d'avoir de nombreuses conversations intéressantes sur mon travail, Patrice Laure, qui m'a encadré en stage, de même qu'Elie Hachem et Rudy Valette, qui n'ont cessé de m'encourager.

Parmi toutes les personnes côtoyées durant mon séjour au laboratoire, j'ai eu avec certaines une relation dépassant le cadre du travail. Je pense notamment à Innocent, Bruno, Séverine ou encore à mes condisciples Ludovic, Armand et Nicolas. Je veux en outre féliciter Rachid, Nicolas, Marie-Cécile, Jérôme, Mohamed et Stéphanie avec qui j'ai partagé un bureau au cours de ces années et qui ont eu le grand mérite de réussir à me supporter. Mention spéciale à mes stagiaires Mohamed Ali et Damir, grâce à qui j'ai pu réaliser la difficulté de la tâche d'encadrement.

A présent, comment ne pas avoir une pensée pour ceux et celles qui ont eu le courage et l'inconscience de se lancer tout comme moi dans la préparation d'une thèse ? Je congratulate les doctorants des promotions précédentes, Guillaume, Larbi, Emile, Christophe, Grégory, Laurence, Céline et Erwan, ceux des promotions suivantes, Ugo, Tommy, Alexis, Michel, Olivier, Jérémy et Ghina et bien évidemment ceux de ma propre promotion, Grégory, Trystan, Nadine, Paméla, Karim, Haithem, Esteban, Nuno, Okba et Audrey.

Je désire enfin mentionner dans ces remerciements tous ceux et celles qui ont contribué à rendre plus faciles ma vie et mon travail au laboratoire, parmi lesquels les directeurs successifs du Centre de Mise en Forme des Matériaux, Jean-Loup Chenot, Yvan Chastel et Elisabeth Massoni, Patrick Coels, le directeur administratif, les assistantes de direction Marie-Françoise et Sylvie, les membres du service informatique, Serge, Carole, Valéry et Ann-Mari, les bibliothécaires, Sylvie et Brigitte, Geneviève, qui m'a aidé dans l'organisation de mes déplacements et les membres de la délégation, Jean-François Agassant, Christian Laruelle et Muriel, en charge de l'accueil et de la reprographie.

Je m'excuse encore une fois auprès de toutes les personnes oubliées.

- Table des matières -

- Introduction générale	1
- Chapitre 1 - Etat de l'art	5
1. Introduction	6
2. Solidification	6
3. Comportement mécanique de la zone pâteuse	7
4. Notre approche	9
5. Conclusion	10
- Chapitre 2 - Transformation des données issues de microtomographies en représentations éléments finis	11
1. Introduction	12
2. Acquisition des images	13
2.1. L'installation européenne de rayonnement synchrotron	13
2.2. La microtomographie aux rayons X	13
2.3. Description des images obtenues	14
3. Traitement des images	16
3.1. Filtre médian	16
3.2. Segmentation	17
4. Maillage surfacique	19
4.1. En deux dimensions	19
4.1.1. Principe de base	19
4.1.2. Algorithme	19
4.1.3. Calcul de la normale	21
4.1.4. Ambiguïté	22
4.1.5. Cohérence	23
4.1.6. Image en niveaux de gris	25
4.2. En trois dimensions	26
4.2.1. Principe	26
4.2.2. Ambiguïté	29
4.2.3. Cohérence	30
4.2.4. Performances	30
4.2.5. Améliorations possibles	32
5. Maillage volumique	32
5.1. Calcul de la distance	33
5.1.1. Méthode linéaire	33
5.1.2. Méthode hiérarchique	34
5.1.3. Algorithme parallèle	36
5.1.4. Remarque	37
5.2. Construction du maillage volumique	37
5.2.1. Maillage initial	37
5.2.2. Remaillage	40
5.3. Traitements	41
5.3.1. Calcul de la surface spécifique	41
5.3.2. Calcul de la fraction volumique	43
5.3.3. Comparaisons	44

6. Conclusion	45
- Chapitre 3 - Solveur mécanique	47
1. Introduction	48
2. Formulation	48
2.1. Formulation forte	48
2.2. Formulation faible	49
2.3. Formulation faible discrétisée	50
2.4. L'élément P1+/P1	50
2.5. Calcul des matrices locales	52
2.5.1. Terme en vitesse-vitesse	52
2.5.2. Terme en vitesse-pression (et pression-vitesse)	53
2.5.3. Terme en pression-pression	53
2.5.4. Calcul des intégrales	54
2.6. Assemblage	56
2.7. Rajout des conditions aux limites	57
2.8. Calcul de la viscosité	57
2.9. Stockage des matrices	58
3. Résolution des systèmes linéaires	59
3.1. Méthode directe	59
3.2. Méthode itérative	60
3.3. Applications simples	62
3.3.1. Problème mal posé	62
3.3.2. Dimension des résidus	63
3.3.3. Solveur direct ou itératif	67
4. Test de l'inclusion	69
4.1. Maillage utilisé	69
4.2. Solution numérique	70
4.3. Rapport des viscosités et précision	72
4.4. Utilisation d'un maillage plus adapté	73
5. Comportement viscoplastique	73
6. Conclusion	77
- Chapitre 4 - Déplacement d'interfaces	79
1. Introduction	80
2. Etude bibliographique	81
3. La méthode Level Set dans CIMLib	84
3.1. Solveur de convection ConvectionP1	84
3.2. Solveur de réinitialisation ReinitLevelSet	87
3.3. Solveur de réinitialisation convective Leveller	90
3.4. Solveur de réinitialisation convective LevellerS	93
3.5. Solveur de réinitialisation convective LevellerT	95
4. La méthode PLS (Particle Level Set)	97
4.1. Convection de la level set	97
4.2. Création des particules	98
4.3. Déplacement des particules	99
4.4. Correction de la level set	100
4.5. Réinitialisation de la level set	100
4.6. Remarques	101
4.7. Effet de l'apport des particules	101
5. Comparaison des deux méthodes	102
5.1. Rotation d'un disque	102

5.2. Test du vortex	105
5.3. Conclusion	107
6. Test de l'inclusion	108
6.1. Solution numérique	109
6.2. Epaisseur de lissage	110
6.3. Remaillage	114
6.4. Maillage anisotrope	116
6.5. Plusieurs inclusions	120
6.6. Période de réinitialisation de GMRES	121
7. Conclusion	125
- Chapitre 5 - Applications	127
1. Introduction	128
2. Test de l'inclusion	128
3. Calculs sur une image 2D	131
3.1. Représentation éléments finis	131
3.2. Comportement newtonien	132
3.3. Comportement viscoplastique	139
3.4. Influence du pas de temps et de la taille de maille	143
4. Calculs sur une image 3D	147
5. Conclusion	155
- Conclusion générale	157
- Annexe 1 - Code pour les Marching Squares	161
- Annexe 2 - Code pour les Marching Cubes	165
- Références bibliographiques	177

- Introduction générale -

« Non quia difficilia sunt, non audemus, sed quia non audemus, difficilia sunt. »

Sénèque

D'après une étude américaine, plus de quarante millions de tonnes d'aluminium ont été produites dans le monde en 2010 [USG 11a]. Cette valeur classe l'aluminium au deuxième rang du métal le plus utilisé, loin derrière l'acier, avec ses 1400 millions de tonnes produites en 2010 [USG 11a]. Il faut toutefois légèrement relativiser le rapport important entre les deux productions en considérant que la masse volumique de l'aluminium est trois fois plus faible que celle de l'acier. Malgré cela, l'acier est toujours beaucoup plus utilisé, en raison de ses caractéristiques mécaniques plus élevées et de son coût inférieur [Reb 05]. La production de l'aluminium est cependant en pleine expansion ces dernières années, comme en atteste la figure I.1, qui nous montre la production annuelle depuis 1900 [USG 11b].

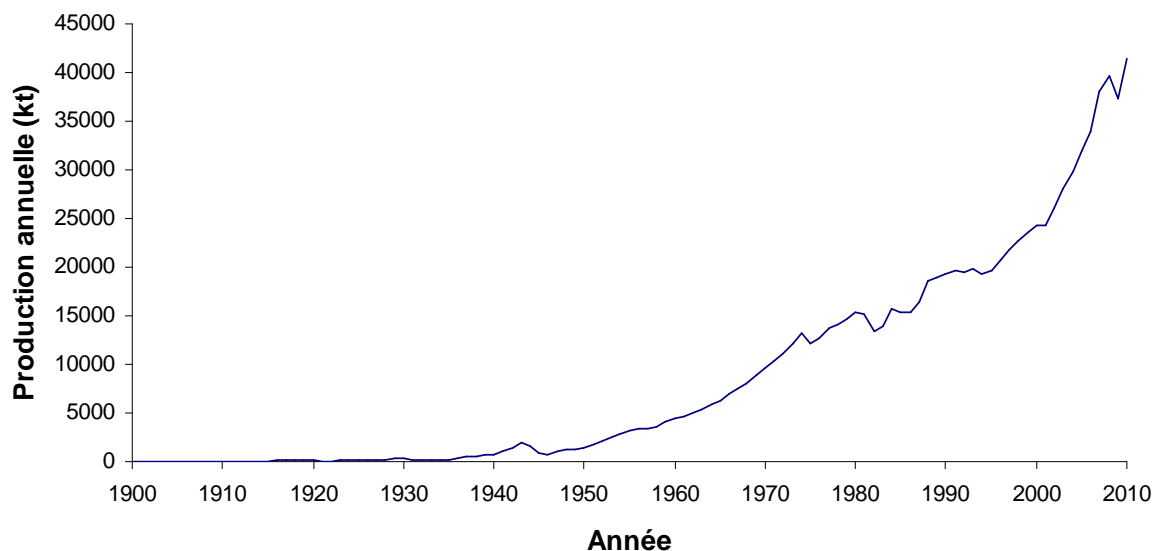


Figure I.1. Production annuelle d'aluminium, courbe tracée d'après les données fournies par une étude américaine [USG 11a, USG 11b].

Dans un ouvrage de Grinberg publié en 2003, il est clairement illustré que « l'histoire de l'aluminium est intimement liée à celle des transports modernes : la conquête de l'air s'est faite grâce à des appareils utilisant ce métal, conjuguant légèreté et haute résistance mécanique. Parce qu'il contribue à réduire les consommations d'énergie et qu'il est hautement recyclable, l'aluminium progresse dans l'automobile comme dans la construction navale [Gri 03]. » Les autres domaines importants d'utilisation de l'aluminium sont la construction, l'emballage ainsi que les équipements électriques et mécaniques. Nous tenons également à signaler que bien qu'il soit le troisième élément le plus abondant sur Terre, après l'oxygène et le silicium, l'aluminium est loin d'être inépuisable, puisque d'après une étude menée par le magazine Science & Vie ses gisements exploitables à un coût acceptable seront épuisés en 2139 [Sci 08].

Introduction générale

La coulée (continue ou en lingots) est une étape importante de la mise en forme des alliages d'aluminium ; c'est celle qui nous intéresse ici. Ce procédé de fabrication ayant déjà été décrit dans de nombreux travaux [Lud 04, Mat 07, Gra 04], nous ne revenons pas dessus en détail dans ce document. Lors d'une coulée, le métal initialement liquide se refroidit progressivement pour se retrouver à l'état solide à la fin du procédé. Pendant le processus de solidification, états liquide et solide coexistent pour définir un état dit pâteux. La phase solide est plus dense que la phase liquide. En outre, les gaz présents initialement dans le liquide sont beaucoup moins solubles dans le solide. Pour compenser le retrait de solidification important dans ces alliages (de l'ordre de 4 à 7 %) et la contraction volumique à l'état solide, des microporosités vont donc apparaître et dégénérer dans certains cas en fissuration à chaud, un défaut rédhibitoire pour la qualité des pièces, qui est associé à l'ouverture de criques dans les régions à haute fraction de solide [Cer 07].

Afin de pouvoir prédire l'apparition de la fissuration à chaud en fonction des conditions de coulée, de la composition de l'alliage utilisé et de la géométrie du moule, de nombreux chercheurs ont travaillé à établir un certain nombre de critères. Eu égard à la dimension des pièces à modéliser (parfois plusieurs mètres), ces critères s'appuient sur des modèles macroscopiques, incapables de prédire la complexité des phénomènes mis en jeu. Pour alimenter ces critères plus précisément, il peut alors être intéressant de réaliser des simulations à des échelles plus petites. Pour initialiser ces simulations, le recours à des méthodes statistiques, telles que celle décrite par Phillion et al. [Phi 08b], semble indiqué.

En parallèle des avancées de la simulation numérique, la dernière décennie a vu la percée d'une prometteuse technique expérimentale connue sous le nom de microtomographie aux rayons X [Sal 03, Bar 06a, Sal 10]. Cette technique permet d'imager in situ et en trois dimensions (3D) des échantillons soumis à des sollicitations thermomécaniques, de manière précise et non destructive. Des résultats très intéressants ont ainsi été obtenus récemment sur des alliages d'aluminium-cuivre pour des expériences de maintien isotherme [Lim 07], de solidification [Lim 09] et de traction isotherme [Ter 09].

Le projet SIMUZAL (acronyme de Numerical Simulation of the MUshy Zone behaviour in Aluminium alloys ou simulation numérique du comportement des alliages d'aluminium à l'état pâteux en français), financé par l'Agence Nationale de la Recherche (ANR) et dans le cadre duquel cette thèse s'inscrit, réunit les trois partenaires académiques suivants : le laboratoire SIMaP (Science et Ingénierie des Matériaux et des Procédés) de l'INPG (Institut National Polytechnique de Grenoble), le laboratoire Navier de l'Ecole des Ponts ParisTech et le Centre de Mise en Forme des Matériaux (CEMEF) de MINES ParisTech.

Ce projet a pour ambition de combiner des approches expérimentale et numérique pour mieux comprendre le comportement des alliages d'aluminium à l'état pâteux, en utilisant respectivement la microtomographie aux rayons X et des simulations numériques à une petite échelle. Contrairement à une approche statistique, nous avons souhaité initialiser nos simulations numériques à partir des données de microtomographie, afin de pouvoir obtenir des comparaisons directes entre expérience et simulation.

Le travail à effectuer a été réparti de la manière suivante entre les trois partenaires. Le laboratoire SIMaP s'est chargé d'effectuer les expériences de microtomographie à l'ESRF (European Synchrotron Radiation Facility ou installation européenne de rayonnement synchrotron) de Grenoble et d'analyser les données obtenues, ainsi que de déterminer la loi de comportement de la phase solide par des tests mécaniques. L'objectif du laboratoire Navier a été de développer et d'utiliser des outils de corrélation d'image afin d'évaluer les conditions aux limites à utiliser dans les simulations et de faciliter la comparaison entre résultats numériques et expérimentaux. Le CEMEF s'est quant à lui concentré sur la simulation numérique des expériences.

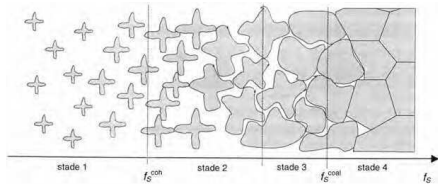
Introduction générale

Afin de profiter de l'expérience reconnue du CEMEF dans le domaine de la simulation numérique du comportement des matériaux, nous avons décidé dès le début du projet de faire appel à CIMLib [Dig 07] (bibliothèque du groupe de recherche Calcul Intensif en Mise en forme des matériaux), bibliothèque écrite en C++ utilisant des maillages tétraédriques non structurés et la méthode des éléments finis pour résoudre des équations aux dérivées partielles. CIMLib est interfacée avec PETSc [Bal 10] (Portable, Extensible Toolkit for Scientific Computation) pour la résolution des systèmes linéaires et avec MPI [MPI 03] (Message Passing Interface) pour la gestion du parallélisme. Le déplacement des interfaces γ est assuré par une méthode level set, décrite par Basset et Ville [Bas 06, Vil 11b].

Au cours de cette thèse, nous nous sommes focalisés sur la simulation d'un test de traction d'un alliage d'aluminium-cuivre à l'état pâteux, effectué expérimentalement par Terzi et al. [Ter 09]. Les images de microtomographie issues de ce test de traction ont été traitées et ont permis d'obtenir une représentation éléments finis directement liée à la réalité expérimentale tridimensionnelle (3D). Grâce à cette représentation de l'échantillon, nous avons pu effectuer un test de traction numérique et ainsi comparer résultats expérimentaux et numériques.

Ce manuscrit s'articule de la manière suivante. Le premier chapitre est dédié à la description de l'état de l'art dans le domaine de la simulation du comportement mécanique des alliages d'aluminium à l'état pâteux. Dans le deuxième chapitre, nous expliquons la méthode utilisée pour transformer les données de microtomographie en représentations éléments finis. Les troisième et quatrième chapitres sont respectivement consacrés à la description des deux solveurs utilisés pour effectuer le test de traction : le solveur mécanique, grâce auquel nous accédons aux valeurs des champs de vitesse, de pression et de contrainte, et le solveur level set permettant de déplacer les interfaces en présence. Enfin dans le cinquième et dernier chapitre, nous couplons ces deux solveurs pour réaliser le test de traction numérique et effectuer une comparaison avec l'expérience.

Introduction générale



- Chapitre 1 - Etat de l'art

« Where all men think alike, no one thinks very much. »

Walter Lippman

1. Introduction	6
2. Solidification	6
3. Comportement mécanique de la zone pâteuse	7
4. Notre approche	9
5. Conclusion	10

1. Introduction

En raison de leur importante utilisation dans de nombreux domaines industriels, les alliages d'aluminium ont fait et font encore aujourd'hui l'objet de très nombreuses études. Pour ce qui concerne la solidification, ces études ont pour objectif de pouvoir prédire les caractéristiques d'une pièce d'aluminium coulée dans des conditions données et spécifiquement l'occurrence de défauts rédhibitoires pour les applications visées, afin de pouvoir orienter les conditions du procédé vers des solutions fiables.

Lee, Chirazi & See, expliquent dans leur revue sur la modélisation de la microporosité dans les alliages d'aluminium-silicium [Lee 01] qu'aucun des modèles disponibles ne fournit de prédictions précises sur la gamme complète des conditions de coulée rencontrées industriellement. Ils ajoutent qu'un modèle idéal devrait inclure l'ensemble des phénomènes physiques à l'oeuvre, ce qui le rendrait trop complexe pour être réellement exploitable.

Notre objectif dans ce chapitre n'est pas de faire un inventaire de tous les travaux ayant pour sujet la solidification des alliages d'aluminium. Nous préférons donc nous concentrer sur les recherches en rapport direct avec notre propre étude. Nous accordons ainsi une attention particulière aux travaux impliquant la microtomographie et les simulations numériques à petite échelle et ayant pour but de mieux comprendre le comportement mécanique des alliages d'aluminium à l'état pâteux.

Nous résumons dans un premier temps les éléments de base indispensables pour comprendre la solidification des alliages d'aluminium. Nous évoquons ensuite les travaux utilisant la microtomographie aux rayons X et des calculs mécaniques sur des volumes élémentaires représentatifs (VER) qui semblent intéressants pour notre étude, avant de présenter l'approche retenue pour la suite de cette thèse.

2. Solidification

La solidification est une étape prépondérante de la mise en forme des alliages d'aluminium, au cours de laquelle le métal, initialement liquide, passe progressivement à l'état solide au cours de son refroidissement. Pour une très bonne analyse des aspects fondamentaux de la solidification, nous conseillons l'ouvrage de Kurz & Fisher [Kur 98] ainsi que celui de Dantzig & Rappaz [Dan 09].

Ludwig donne une description générale de la solidification d'un alliage d'aluminium-cuivre hypoeutectique [Lud 04], en précisant que l'évolution d'une structure de solidification est constituée de quatre étapes : la germination, la croissance, la croissance dendritique et la solidification eutectique.

Durant la solidification, la zone où les phases liquide et solide coexistent est appelée zone pâteuse. Dahle & Arnberg décomposent la progression de la solidification dans cette zone pâteuse en quatre étapes différentes [Dah 96], illustrées sur la figure 1.1. Tout d'abord pendant le premier stade, les dendrites sont isolées dans le liquide et flottent librement. La zone pâteuse n'a alors pas de résistance mécanique en traction. Lors du deuxième stade, la fraction de solide dite de cohérence est atteinte : les dendrites forment un réseau connecté. Le liquide circule aisément dans ce réseau. En revanche durant la troisième phase, la perméabilité du squelette solide chute et le liquide ne peut plus s'écouler aussi librement. La quatrième et dernière étape est atteinte pour une fraction solide dite de cohérence. Des ponts solides sont formés entre les grains, ce qui consolide la structure et isole le liquide.

La masse volumique du solide est plus importante que celle du liquide. Elle augmente également au fur et à mesure que la température diminue. Par conséquent, un apport de matière est nécessaire pour compenser le retrait de solidification et la contraction à l'état

solide. Durant les deux premières phases cet apport se fait par une alimentation en liquide. En fin de solidification il ne reste plus suffisamment de liquide pour compenser le retrait et nous assistons à une chute de pression dans les poches de liquide isolées, ce qui peut mener à la création de pores, dont la formation est aussi reliée à la concentration en gaz dans le liquide. L'explication de la figure 1.1 est également donnée par Cerri et Ludwig [Cer 07, Lud 04].

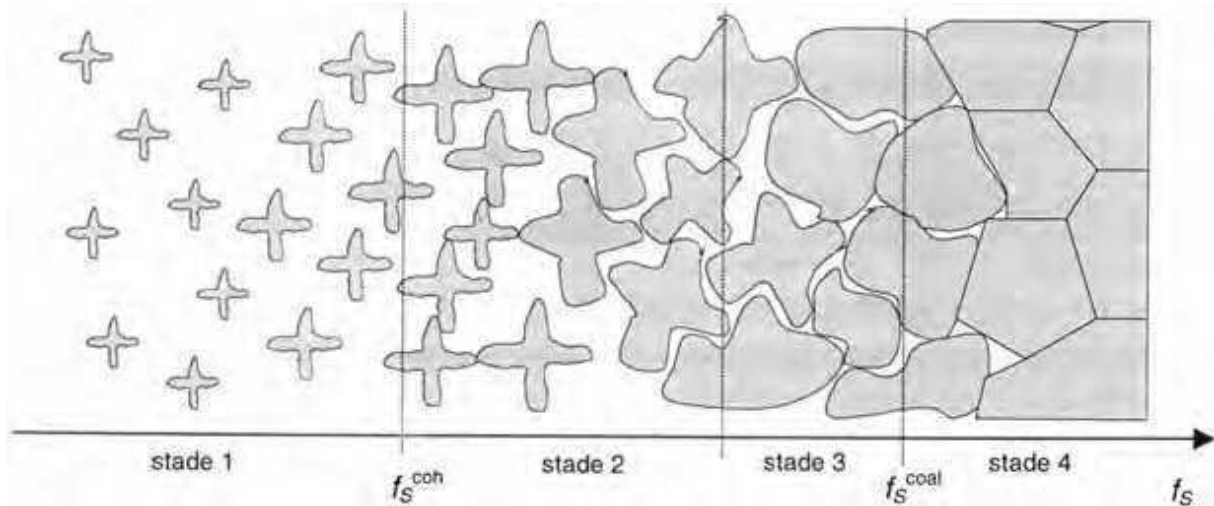


Figure 1.1. Les différents stades de la solidification d'un alliage métallique, d'après Dahle & Arnberg [Dah 96].

Dans certains cas, ces pores vont dégénérer en fissuration à chaud. Il s'agit de l'apparition d'une fissure au sein d'un alliage pâteux [Esk 04]. La fissuration à chaud est un défaut majeur rencontré durant la solidification des alliages d'aluminium. Elle entraîne le rebut de produits pour lesquels la solidification joue un rôle important, tel que la coulée de lingots, la coulée continue ou le soudage. Ce défaut réduisant la productivité dans les procédés de coulée, il est nécessaire d'en comprendre les phénomènes sous-jacents. Comme illustré dans la revue d'Eskin, Suyitno & Katgerman [Esk 04], de nombreux travaux ont été menés sur la fissuration à chaud et le mécanisme général d'apparition est bien compris. En effet, ce défaut est associé à l'impossibilité pour le liquide de nourrir les zones dans lesquelles des pores ont commencé à apparaître, ne permettant pas de résorber ces défauts avant qu'ils ne croissent.

Des études montrent que la fissuration à chaud peut seulement apparaître en fin de solidification, quand la fraction de solide est supérieure à 85 % [Esk 07]. Pour de telles fractions de solide, des ponts solides existent entre les grains et le liquide restant est seulement formé de minces couches et de petites poches autour des grains [Fab 06]. Il ne peut plus s'écouler librement. La déformation due à la solidification continue et mène à des contraintes en traction et des cavitations dans la phase liquide [Gir 10]. Ces pores coalescent et se propagent sous forme de fissures à travers le réseau granulaire et peuvent entraîner une fissuration à chaud si les ponts solides et la coalescence des grains ne ralentissent pas suffisamment le phénomène [Bel 09]. Il est donc crucial de développer une bonne connaissance du comportement mécanique des alliages d'aluminium pâteux [Gir 10].

3. Comportement mécanique de la zone pâteuse

Les vitesses de déformation rencontrées en solidification sont faibles. Par conséquent, nous écartons de cette partie l'étude du comportement thixotrope des alliages à l'état pâteux, autrement dit le comportement des suspensions chargées de cristaux équiaxes et soumises à de

hautes vitesses d'écoulement et de déformation, caractéristiques de l'injection [Mot 02]. Deux approches principales ont été utilisées pour étudier le comportement mécanique des alliages métalliques à l'état pâteux en rapport avec la fissuration à chaud : expérimentale et numérique. Eskin donne un bon aperçu des différents dispositifs expérimentaux utilisés au fil des ans [Esk 07]. Récemment, Phillion et al. ont observé en trois dimensions l'évolution des porosités au sein d'un échantillon d'aluminium-magnésium à l'état pâteux, soumis à un test de traction interrompu, grâce à la microtomographie aux rayons X [Phi 08a]. Cette technique d'imagerie surpasse les techniques précédentes, basées sur des observations surfaciques. Elle a permis de conclure que les porosités préexistantes dans l'échantillon entraînent une augmentation des contraintes locales et permettent d'accommoder les déformations par croissance de ces porosités. Mais le travail de Phillion présente une importante limitation, due à l'interruption du test. Limodin et al., pour leur expérience de maintien isotherme, ont en effet montré l'importance de faire des observations in situ [Lim 07], ce que Terzi et al. ont pu réaliser pour une expérience de traction isotherme d'un alliage d'aluminium-cuivre à l'état pâteux [Ter 09]. Les deux résultats importants obtenus sont l'accumulation de liquide aux surfaces intergranulaires perpendiculaires à l'axe de traction, notamment au milieu de l'éprouvette, ainsi que la propagation de l'air dans l'échantillon à partir de sa surface extérieure.

Sun donne une explication du comportement mécanique d'un alliage pâteux [Sun 10a] : séparément, le comportement mécanique des phases liquide et solide est connu. En revanche, compte tenu de la complexité morphologique de l'alliage à l'état pâteux, la compréhension du comportement du mélange des phases liquide et solide est difficile. Le matériau pâteux peut être vu comme une structure polycristalline, avec du liquide situé au niveau des joints de grains. Ce liquide diminue grandement la résistance mécanique, laquelle est alors beaucoup moins importante que celle du solide pur.

Fuloria & Lee [Ful 09] expliquent qu'une approche classique pour accéder au comportement mécanique d'un alliage pâteux consiste à considérer cet alliage comme un milieu poreux saturé par un liquide visqueux [Mar 02]. Cela permet d'introduire une variable interne qui représente la cohésion partielle du milieu poreux et d'identifier les paramètres d'un modèle à l'aide de tests mécaniques. Ce modèle peut ensuite être implémenté dans un code éléments finis [Lud 05]. Cependant, ce type de modèle est construit pour un alliage particulier et il est difficile de l'appliquer à d'autres systèmes [Ful 09].

Vernède, Jarry & Rappaz [Ver 06, Ver 07] ont mis au point un modèle granulaire qui permet de simuler la transformation de la zone pâteuse d'un alliage durant sa solidification, passant d'un film liquide continu à un solide totalement cohérent. Grâce à ce modèle, ils parviennent à calculer l'approvisionnement en liquide de la zone pâteuse ainsi que la percolation des grains solides. Mais aucun couplage mécanique entre le comportement des phases solide et liquide n'est pris en compte.

Phillion, Cockcroft & Lee ont développé une simulation éléments finis directe pour étudier le comportement mécanique des alliages métalliques à l'état pâteux en initialisant leur domaine de calcul par une tessellation de Voronoï [Phi 08b]. Ils ont ainsi réussi à déduire une équation décrivant le comportement mécanique et incluant les effets de paramètres microstructuraux tels que la fraction de solide, la taille des grains ou encore la fraction de porosité [Phi 09a]. Cependant, les simulations effectuées présentent les limitations suivantes : (i) elles sont seulement bidimensionnelles, (ii) la déformation maximale atteignable est faible, (iii) le liquide est approximé comme un matériau plastique avec une très faible contrainte d'écoulement, au lieu d'être considéré newtonien.

Fuloria & Lee ont quant à eux utilisé des données obtenues par microtomographie afin d'effectuer des simulations éléments finis sur des structures réelles [Ful 09]. Grâce à cela, ils ont pu déterminer un terme de dépendance à la fraction de solide au sein d'une équation

constitutive qui permet d'estimer la contrainte d'écoulement d'un alliage pâteux. Ce terme est seulement dépendant de la géométrie de l'éprouvette, ce qui signifie qu'il peut être utilisé sur d'autres types d'alliage, à condition de connaître le comportement du solide pur. Leurs résultats sont valables pour une fraction de solide allant de 0,1 à 0,9. Seulement, la présence de liquide n'est pas considérée dans leur étude. De plus, seuls des tests de compression uniaxiale sur des structures colonnaires ont été simulés.

Sistaninia et al. ont récemment développé à l'Ecole Polytechnique Fédérale de Lausanne (EPFL) une simulation basée sur un couplage des méthodes éléments finis et éléments discrets capable de prédire le comportement mécanique des alliages métalliques à l'état pâteux durant leur solidification [Sis 11]. La géométrie 3D initiale de l'alliage pâteux est obtenue en utilisant le modèle granulaire précédemment développé [Ver 06, Phi 09b]. Malheureusement, leur modèle souffre de plusieurs défauts. En effet il ne peut prédire le comportement mécanique de la zone pâteuse (i) que pour des fractions solides entre 0,9 et 1, (ii) que pour des faibles déformations et des taux de déformation limités, (iii) qu'avant initiation de dommage à l'intérieur de l'éprouvette.

4. Notre approche

Afin d'essayer de pallier les inconvénients des approches proposées par Phillion, Fuloria et Sistaninia, nous avons décidé de poursuivre les travaux de Sun, développés au CEMEF avec l'aide de CIMLib [Sun 10a, Sun 10b]. Mais contrairement à lui, nous souhaitons initialiser nos simulations éléments finis à partir de données de microtomographie et non d'une construction arbitraire basée sur une tessellation de Voronoï [Hit 11]. Nous nous proposons ainsi de résoudre les équations de conservation de la masse et de la quantité de mouvement de manière monolithique sur l'éprouvette expérimentale de Terzi et al. [Ter 09]. Cette résolution monolithique implique le calcul des champs de vitesse et de pression à la fois dans les phases solide, liquide et gazeuse, en une seule étape et en considérant une loi de comportement pour chaque phase. Les équations de conservation de la masse et de la quantité de mouvement sont données ci-dessous :

$$\begin{cases} \rho \frac{d\bar{v}}{dt} = \nabla \cdot \bar{\bar{\sigma}} + \rho \bar{g} \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{v}) = 0 \end{cases}$$

avec ρ la densité, \bar{v} la vitesse de l'écoulement, \bar{g} la gravité et $\bar{\bar{\sigma}}$ le tenseur des contraintes de Cauchy.

Le nombre de Reynolds de l'écoulement est un nombre adimensionnel nous renseignant sur le régime de l'écoulement. Il exprime le rapport des forces d'inertie sur les forces visqueuses [Pur 77] :

$$Re = \frac{\rho v L}{\eta}$$

avec v la norme de la vitesse, L la longueur caractéristique de l'écoulement et η la viscosité dynamique.

Chapitre 1

Il est très utile de connaître le régime de l'écoulement. Pour la phase solide, il est évident que $Re \ll 1$. Pour le liquide, chacun des termes doit être évalué. ρ est la densité de l'aluminium : $\rho = 2700 \text{ kg.m}^{-3}$. L , la dimension caractéristique des veines liquides est à peu près de $100 \mu\text{m}$ [Ter 09]. En ce qui concerne la viscosité, Vernède donne la valeur 10^{-3} Pa.s [Ver 07]. Nous avons $Re = 270$ v.

Dans le cas que nous voulons simuler ici, un test de traction uniaxiale sur un échantillon millimétrique à une vitesse imposée de $0,1 \mu\text{m.s}^{-1}$ [Ter 09] sur le plan supérieur, nous considérons que le nombre de Reynolds est très faible, et que les termes d'inertie et de gravité peuvent être négligés. Nous devons donc résoudre les équations de Stokes sans second membre :

$$\begin{cases} \nabla \cdot \bar{\sigma} = \vec{0} \\ \nabla \cdot \vec{v} = 0 \end{cases}$$

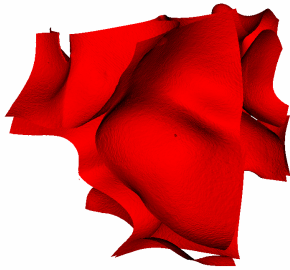
La méthode de résolution de ces équations est détaillée dans le troisième chapitre.

Une fois le champ de vitesse obtenu sur l'ensemble du domaine de calcul, il est nécessaire de déplacer les différentes phases en présence. Sun a montré une grande similitude entre les formalismes lagrangien et eulérien pour ses tests de traction [Sun 10a]. La fenêtre d'observation étant fixe dans l'expérience de Terzi [Ter 09], nous avons choisi d'utiliser le formalisme eulérien afin de garder le domaine de calcul fixe également. Ce formalisme est décrit dans le quatrième chapitre.

5. Conclusion

Nous avons donné dans ce chapitre une courte description du procédé de solidification et du phénomène de fissuration à chaud. Les travaux effectués dans le cadre de cette thèse étant assez éloignés de ces considérations et n'ayant pas porté sur la mise au point d'une équation décrivant le comportement mécanique d'un alliage d'aluminium à l'état pâteux, nous avons volontairement écourté cette présentation de l'état de l'art, mais invitons les lecteurs intéressés à étudier les références évoquées.

La microtomographie aux rayons X et les calculs mécaniques sur des VER à l'état pâteux ont été développés assez récemment et les travaux en rapport direct avec notre étude, qui concerne le couplage entre microtomographie et calculs mécaniques, ne sont pas encore très nombreux. D'ailleurs, compte tenu de la grande différence de propriétés mécaniques entre les phases liquide et solide, il n'existe pas à notre connaissance dans la littérature de méthodes permettant de simuler avec précision l'évolution de la microstructure obtenue expérimentalement par Terzi et al. [Ter 09]. Nous allons dans la suite de cette thèse nous attacher à en développer une, en nous appuyant sur les travaux de Sun [Sun 10a, Sun 10b]. Il est utile de préciser que si ses travaux nous paraissent adaptés pour atteindre notre objectif, ils ont toutefois mis en avant de nombreux problèmes de convergence au cours des résolutions mécaniques, ce qui nous a poussé à examiner le fonctionnement du solveur de Stokes avec la plus grande attention au cours du troisième chapitre de ce document.



- Chapitre 2 -

Transformation des données issues de microtomographies en représentations éléments finis

« Quoi de plus insultant à la science que l'humilité d'un esprit remarquable qui renonce à tout honneur pour satisfaire un maître invisible. »

Marie-Claire Blais

1. Introduction	12
2. Acquisition des images	13
2.1. L'installation européenne de rayonnement synchrotron	13
2.2. La microtomographie aux rayons X	13
2.3. Description des images obtenues	14
3. Traitement des images	16
3.1. Filtre médian	16
3.2. Segmentation	17
4. Maillage surfacique	19
4.1. En deux dimensions	19
4.1.1. Principe de base	19
4.1.2. Algorithme	19
4.1.3. Calcul de la normale	21
4.1.4. Ambiguïté	22
4.1.5. Cohérence	23
4.1.6. Image en niveaux de gris	25
4.2. En trois dimensions	26
4.2.1. Principe	26
4.2.2. Ambiguïté	29
4.2.3. Cohérence	30
4.2.4. Performances	30
4.2.5. Améliorations possibles	32
5. Maillage volumique	32
5.1. Calcul de la distance	33
5.1.1. Méthode linéaire	33
5.1.2. Méthode hiérarchique	34
5.1.3. Algorithme parallèle	36
5.1.4. Remarque	37
5.2. Construction du maillage volumique	37
5.2.1. Maillage initial	37
5.2.2. Remaillage	40
5.3. Traitements	41
5.3.1. Calcul de la surface spécifique	41
5.3.2. Calcul de la fraction volumique	43
5.3.3. Comparaisons	44
6. Conclusion	45

1. Introduction

Nous consacrons ce chapitre à la description des différentes étapes nécessaires pour transformer les images obtenues par microtomographie et fournies par nos partenaires du laboratoire SIMaP en représentations éléments finis utilisées pour effectuer des simulations numériques avec la librairie CIMLib.

Pour construire une représentation éléments finis à partir de données microtomographiques, nous retenons les trois étapes proposées par Madi et Youssef [Mad 07, You 05]. La première est la segmentation, qui consiste à identifier les différents milieux (ou phases) en présence dans l'échantillon. Nous devons ensuite construire des maillages surfaciques qui représentent les interfaces entre ces milieux. Ces maillages surfaciques permettent enfin de calculer un maillage volumique sur lequel les propriétés de chaque milieu peuvent être définies.

Des travaux employant des images expérimentales pour initialiser des simulations ont déjà été menés au CEMEF. Dans des études concernant le domaine médical, Paccini et Odin [Pac 05, Odi 08] se servent du logiciel commercial Amira[®] [Ami 10]. Ce logiciel, également plébiscité par Madi et Youssef [Mad 06, You 05], utilise l'algorithme des *marching cubes* (MC) [Lor 87] pour calculer les maillages surfaciques associés aux images segmentées et une méthode par suivi de front [Fre 98] pour la construction des maillages volumiques. Cependant, les maillages tétraédriques obtenus par ces auteurs sont homogènes et isotropes. Pour pallier cet inconvénient, Paccini choisit d'utiliser le logiciel Forge[®] [Che 98] afin de remailler son domaine de calcul avec une taille de maille hétérogène, adaptée au calcul éléments finis.

Milesi et al. [Mil 11], pour avoir une représentation numérique d'inclusions de sulfure de manganèse contenues dans une matrice acier et faire des simulations d'essais de fatigue, limitent l'usage d'Amira[®] au seul calcul des maillages surfaciques. Les maillages volumiques correspondants sont alors construits grâce à la technique d'immersion de volume [Hac 10] développée dans CIMLib. Avec cette méthode, basée sur la génération et l'adaptation d'un maillage anisotrope par optimisation locale [Gru 05], les interfaces entre la matrice et les inclusions peuvent être fidèlement représentées avec un nombre de degrés de liberté beaucoup moins important que pour des maillages isotropes.

Utiliser Amira[®] pour le maillage surfacique et CIMLib pour le maillage volumique nécessite cependant l'ajout d'une étape intermédiaire afin de rendre les fichiers de sortie d'Amira[®] compatibles avec les fichiers d'entrée de CIMLib. En outre, le développement perpétuel des techniques d'imagerie facilite de plus en plus la transformation de données expérimentales en représentations éléments finis et les projets susceptibles de s'appuyer sur une telle technologie vont potentiellement se multiplier au CEMEF au cours des prochaines années. Dans ce contexte, il nous est apparu souhaitable d'éviter le recours au logiciel commercial Amira[®], d'autant plus que l'algorithme des *marching cubes* employé pour obtenir les maillages surfaciques est désormais bien connu [New 06] et libre de droit. C'est pourquoi nous développons ici notre propre version de cet algorithme.

Pour commencer, nous donnons une description de l'installation européenne de rayonnement synchrotron (ESRF), de la technique de la microtomographie aux rayons X, ainsi que des images obtenues. Nous détaillons ensuite les trois étapes de construction de la représentation éléments finis : la segmentation des images expérimentales, le calcul du maillage surfacique par l'algorithme des *marching cubes* et celui du maillage volumique par la méthode d'immersion de volume. Nous expliquons alors comment calculer la surface spécifique d'une interface et une fraction volumique à partir d'une représentation numérique, afin de pouvoir établir des comparaisons avec les résultats expérimentaux.

2. Acquisition des images

2.1. L'installation européenne de rayonnement synchrotron

Il existe actuellement plus de cinquante installations de rayonnement synchrotron dans le monde et de nouvelles sont en phase de conception ou de construction [Win 07]. Ces installations de plus en plus prisées permettent une utilisation efficace de nombreuses techniques expérimentales, dont la microtomographie aux rayons X, et sont employées dans de multiples domaines de recherche, tel que la science des matériaux [Har 09].

Le rayonnement synchrotron a été observé pour la première fois en 1947 [Eld 48, Pol 83]. Il est émis par des électrons circulant à une vitesse proche de celle de la lumière et dont la trajectoire est déviée par un champ magnétique. A l'ESRF, les électrons émis par un canon à électrons sont accélérés dans un accélérateur linéaire avant d'être transmis à un accélérateur circulaire qui amène les électrons à une énergie de 6 GeV [Har 09]. Ces électrons sont ensuite injectés dans un anneau de stockage de 844 m de circonférence où ils circulent pendant de nombreuses heures [Har 09]. Pour les lecteurs intéressés, Sands donne une introduction des phénomènes physiques à l'œuvre dans les anneaux de stockage d'électrons [San 79].

Les aimants de courbure font partie des différents dispositifs présents dans un anneau de stockage. Ils entraînent une modification de la trajectoire des électrons de plusieurs degrés qui provoque une perte d'énergie par émission de photons, il s'agit du rayonnement synchrotron. La lumière ainsi produite est extrêmement brillante : mille milliards de fois plus que celles produites par des sources conventionnelles de rayons X [Har 09]. Cette lumière, dont les propriétés sont mises en équation de manière simplifiée par Margaritondo [Mar 00], est envoyée dans des pièces expérimentales appelées lignes de lumière et placées autour de l'anneau de stockage.

Les lignes de lumière ID15 et ID19 de l'ESRF, auxquelles nous avons eu accès pendant le projet SIMUZAL, sont composées de trois cabines. La cabine optique est destinée à ajuster les propriétés du faisceau envoyé par l'anneau de stockage, telles que sa taille, son énergie ou sa longueur d'onde, en fonction de l'application visée. L'échantillon, ainsi que les dispositifs utilisés (four et machine de traction dans notre cas) sont placés dans la cabine expérimentale en compagnie des détecteurs qui enregistrent les informations issues de l'interaction entre le faisceau et l'échantillon. Enfin, les scientifiques utilisent la cabine de contrôle pour démarrer leur expérience, s'assurer de son bon fonctionnement et collecter les données associées.

2.2. La microtomographie aux rayons X

La microtomographie aux rayons X est une technique d'imagerie très intéressante. Non destructive, elle permet d'obtenir des images volumiques d'un échantillon in situ. Son essor est important au cours de ces dernières années, avec des temps d'acquisition et une qualité des images qui ne cessent de s'améliorer, ce qui en fait actuellement une méthode de choix en science des matériaux [Sal 03, Bar 06a, Sal 10].

Le principe de la microtomographie aux rayons X est donné par Bernard [Ber 05] et illustré schématiquement sur la figure 2.1 pour une expérience de solidification [Lim 09]. Nous le résumons à présent en omettant volontairement les considérations d'ordre technique.

Nous considérons un matériau homogène sur lequel est envoyé un faisceau d'une certaine énergie. Le rapport entre le nombre de photons incidents et le nombre de photons transmis est alors fonction de l'épaisseur du matériau et de son coefficient d'atténuation, qui dépend du numéro atomique du matériau, de sa densité et de l'énergie du faisceau [Sal 10]. Pour un matériau hétérogène, le nombre de photons transmis est simplement lié à l'intégrale du

coefficient d'atténuation le long du chemin emprunté par les photons à travers l'échantillon [Sal 03]. Un dispositif optique adéquat permet alors d'obtenir une image appelée radiographie, qui est une mesure des coefficients d'atténuation des milieux traversés. Une seule radiographie ne permettant pas d'avoir des informations volumiques, l'échantillon analysé subit une rotation et plusieurs centaines de radiographies sont obtenues sous différents angles. Grâce à ces radiographies, une image volumique de l'échantillon est reconstruite en utilisant les outils mathématiques appropriés [Ber 05].

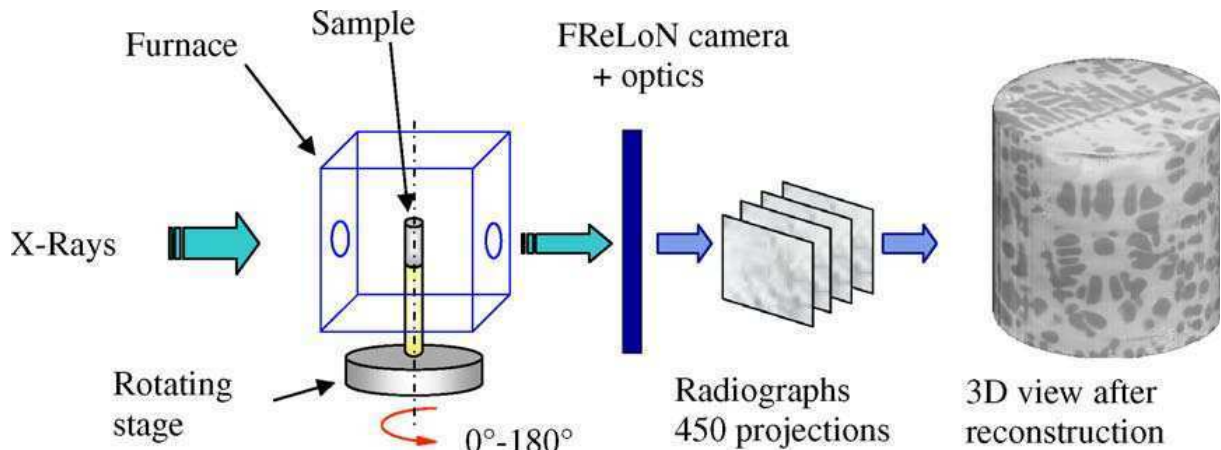


Figure 2.1. Principe de la microtomographie aux rayons X d'après Limodin et al. [Lim 09]

Salvo et al. décrivent différents moyens d'utiliser la tomographie en science des matériaux [Sal 10] en classant par intérêt croissant la tomographie post mortem, la tomographie ex situ, la tomographie in situ interrompue et la tomographie in situ continue. Nous bénéficions dans notre cas d'expériences de tomographie in situ continue, qui demandent un important travail de calibration. Pour avoir une bonne idée des conditions expérimentales utilisées dans le projet SIMUZAL, nous renvoyons le lecteur aux travaux récents du laboratoire SIMaP [Lim 07, Lim 09, Ter 09].

2.3. Description des images obtenues

Après traitement, les données sont converties en images tridimensionnelles codées sur un seul octet. Concrètement, cela signifie que pour chaque voxel (un voxel est à une image 3D ce que le pixel est à une image 2D) est défini un niveau de gris dont la valeur est comprise entre 0 et 255, 0 étant noir et 255 blanc. Ce niveau de gris représente « la moyenne des coefficients d'atténuation des différents solides et fluides » contenus dans le voxel durant l'acquisition [Ber 05]. Il est donc en grande partie relié au numéro atomique des constituants présents dans le voxel.

Plus le niveau de gris est petit, plus le coefficient d'atténuation est faible et le numéro atomique bas. Nous rappelons également que pour des alliages d'aluminium-cuivre à l'état pâteux, la phase liquide contient plus de cuivre que la phase solide. Cela signifie que les voxels gris foncé appartiennent à la phase solide tandis que les voxels gris clair appartiennent à la phase liquide, les voxels très sombres représentant la phase gazeuse.

Les volumes de données obtenus en microtomographie sont très importants. Ainsi, pour une expérience de solidification d'un alliage Al-8 % pds Cu faite sur la ligne de lumière ID19, décrite par Madi [Mad 06], les images possèdent 1024 voxels suivant chacune des directions spatiales. Un voxel occupant une place d'un octet, une seule image nécessite donc 1 Go pour

son stockage. Afin de pouvoir étudier les évolutions microstructurales durant la solidification, 92 images ont été reconstruites, avec un intervalle de 60 s entre 2 images. Pour cette seule expérience, nous avons ainsi 92 Go de données. La figure 2.2 illustre deux des images obtenues.

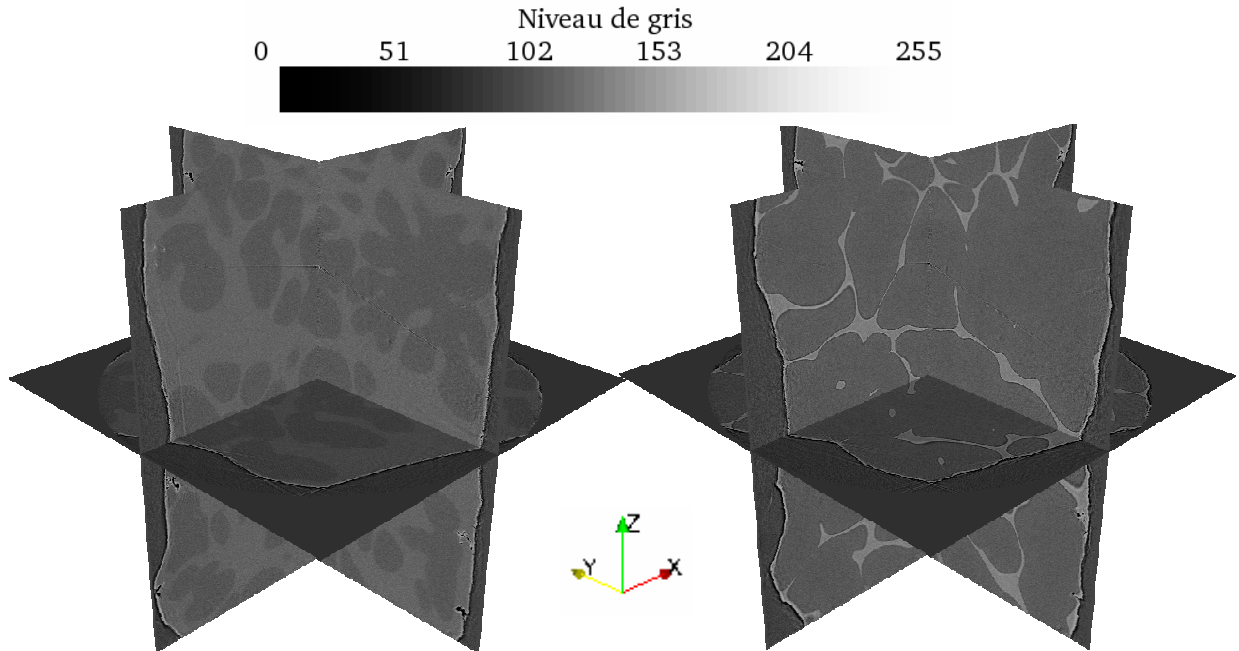


Figure 2.2. Images obtenues après 24 et 72 min de solidification d'un échantillon cylindrique d'Al-8 % Cu. La vitesse de refroidissement est de $0,05 \text{ }^\circ\text{C}\cdot\text{s}^{-1}$ et le champ de visualisation est un cube de 1,43 mm de côté. Pour des raisons de clarté, seules trois coupes perpendiculaires aux axes X, Y et Z et passant par le centre de l'image sont montrées.

Pour pouvoir exploiter ce type d'images comme point de départ de nos calculs, nous devons tout d'abord séparer ces différentes phases liquide, solide et gazeuse, les images pouvant être recadrées pour ne pas considérer le bord de l'échantillon [Lim 07]. Nous devons donc obtenir une image possédant seulement deux ou trois niveaux de gris, selon que nous voulons représenter la phase gazeuse ou pas. En procédant de cette manière, nous détruisons de l'information et nous privons de la connaissance des gradients de composition en cuivre dans le solide et dans le liquide, laquelle pourrait être très importante pour nous aider à initialiser nos simulations. Nous faisons ce choix car la technique d'acquisition actuelle des images ne nous permet pas d'obtenir avec précision la composition en cuivre pour chaque voxel. Rucalvaba et al., dans leur étude de la fragmentation dendritique [Ruc 07], arrivent à reconstruire le champ de composition dans le liquide, mais ils considèrent un alliage Al-20 % pds Cu, de composition en cuivre supérieure à celle des alliages utilisés lors du projet SIMUZAL, ce qui leur facilite la tâche.

A l'œil nu, il est très facile de distinguer les différentes phases en présence dans les images obtenues mais il est beaucoup plus difficile de le faire par ordinateur. En effet, nos yeux interprètent automatiquement les images que nous voyons, tandis que les ordinateurs ont simplement accès à des séries de nombres représentant l'image. La prochaine partie est donc dédiée à la méthode de séparation des phases en présence dans l'échantillon.

3. Traitement des images

3.1. Filtre médian

Nous étudions l'image 2D représentée sur la figure 2.3a. Nous souhaitons segmenter cette image en déterminant la phase correspondante à chaque pixel. Comme il n'y a que deux phases, nous parlons aussi de binarisation. Pour binariser l'image, nous choisissons simplement une valeur de niveau de gris comme seuil. Si un pixel possède une valeur inférieure à ce seuil, il est assigné à la phase solide et représenté en noir, sinon il correspond à la phase liquide et est représenté en blanc. Pour cette image, quel que soit le seuil choisi, il n'est pas possible d'obtenir une binarisation satisfaisante. Sur la figure 2.3b, nous voyons en effet que pour un seuil de 151, des pixels blancs (liquide) sont présents dans le solide (en noir) et inversement. Ce résultat lié à la méthode d'acquisition de l'image ne correspond pas à la réalité expérimentale et ne permet pas de poursuivre le traitement.

Pour résoudre ce problème, nous utilisons un filtre médian, tout comme dans d'autres travaux impliquant la microtomographie [Lim 07, Lim 09]. Ce filtre se caractérise non seulement par sa facilité d'implémentation mais également par son aptitude à conserver la définition des frontières tout en réduisant le bruit contenu dans les images de manière significative.

Le filtre médian utilisé a le fonctionnement suivant en deux dimensions. Nous parcourons l'ensemble des pixels de l'image. Pour chaque pixel, nous plaçons dans un tableau le niveau de gris de ce pixel ainsi que celui de ses huit voisins. Nous trions ensuite ce tableau de neuf valeurs par ordre croissant et le pixel se voit affecter la valeur médiane du tableau trié, c'est-à-dire la 5^{ème}. Nous ajoutons que le sens de parcours n'a aucune importance, le niveau de gris des pixels étant réactualisé à la fin du parcours uniquement. Sur les bords de l'image, le nombre de valeurs à placer dans le tableau est pair. Dans ce cas, le niveau de gris du pixel après application du filtre est la moyenne des deux valeurs médianes du tableau.

Nous illustrons le principe de notre filtre médian en donnant dans le tableau 2.1 les niveaux de gris du pixel encadré en rouge sur la figure 2.3b et de ses voisins. Ce pixel a initialement une valeur de 150 et appartient donc à la phase solide. Après le filtrage, son niveau de gris est de 162 et le pixel appartient bien à la phase liquide.

162	155	153										
171	150	155	→	150	153	155	155	162	165	169	171	171
171	165	169										

Tableau 2.1. Niveau de gris d'un pixel mal segmenté avant et après application du filtre médian. A gauche, niveau de gris du pixel (en rouge) et de ses voisins. A droite, tableau donnant les niveaux de gris triés par ordre croissant. En pratique, notre algorithme permet de récupérer la valeur médiane sans trier le tableau pour une question d'efficacité.

La figure 2.3c montre l'image après application du filtre médian et la figure 2.3d la binarisation associée avec un seuil de 151. Le filtre a ainsi permis de correctement binariser l'image : il n'y a plus de pixels blancs dans le solide et vice versa. Par ailleurs, les contours entre les deux phases obtenus avant et après filtrage sont très proches.

La description que nous venons de faire pour le filtre médian en 2D s'applique également au filtre médian en 3D. Un voxel qui n'est pas situé sur le bord de l'image possède alors 26 voisins. Le tableau à remplir compte donc 27 éléments et nous récupérons la valeur du 14^{ème}.

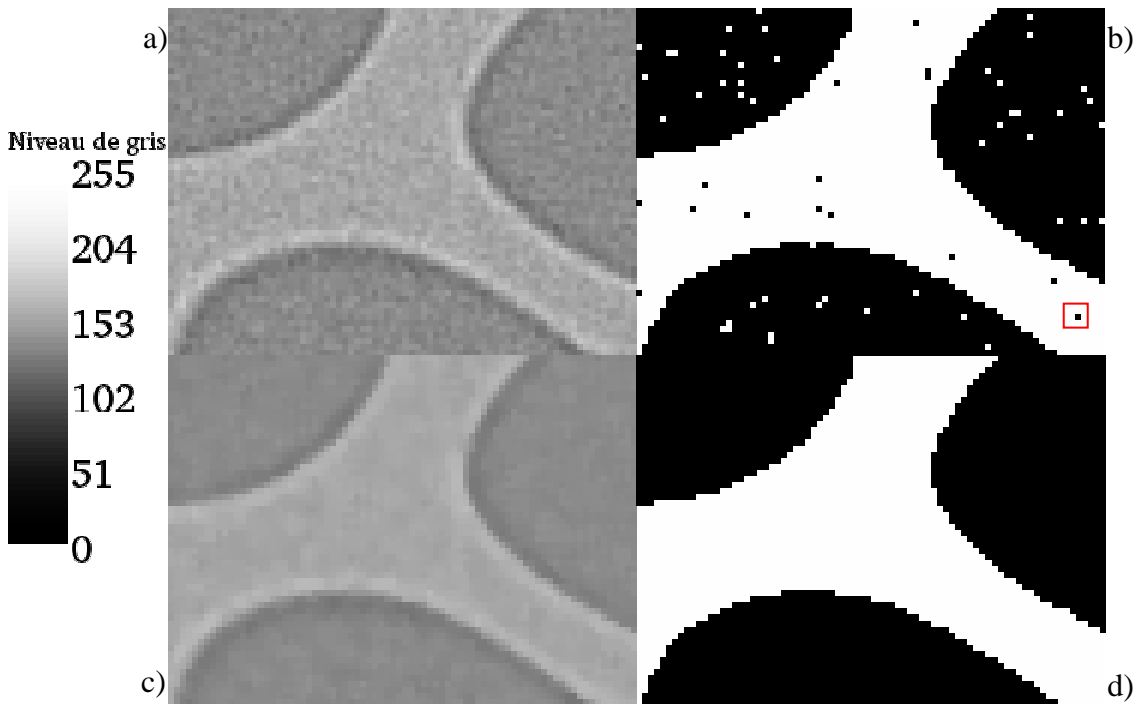


Figure 2.3. Intérêt du filtre médian. a) Image initiale. b) Image initiale binarisée. c) Image filtrée. d) Image filtrée binarisée.

3.2. Segmentation

Pour déterminer le niveau du seuil à appliquer, nous utilisons l'approche de Madi [Mad 06], qui consiste à tracer un histogramme de distribution des niveaux de gris. Cet histogramme pour la figure 2.3c est représenté par la courbe bleue sur la figure 2.4. Sachant que deux phases seulement sont présentes dans cette image, nous identifions les pics correspondants et prenons une valeur de seuil située entre les deux pics, 151 par exemple.

Ce type de segmentation manuelle est tout à fait valable si nous voulons traiter un petit nombre d'images. Cependant, si le nombre d'images à traiter est grand, il peut être intéressant de déterminer le seuil de manière automatique. Pour cela, nous procédons à un lissage de l'histogramme, en ajoutant pour chaque niveau de gris le nombre de pixels des niveaux de gris adjacents. Par exemple, avec une épaisseur de lissage de 4 niveaux de gris, la valeur correspondante au niveau 151 est calculée en ajoutant le nombre de pixels possédant un niveau entre 147 et 155. L'histogramme lissé correspondant est tracé en rose sur la figure 2.4. Il montre les mêmes tendances que l'histogramme non lissé, mais permet d'identifier de manière beaucoup plus claire le maximum de la distribution pour chaque pic ainsi que la valeur minimale entre les deux pics, que nous prenons comme seuil.

Pour l'expérience de solidification, la composition des phases solide et liquide évolue au cours du temps et il n'est pas possible de correctement segmenter l'ensemble des images en utilisant une seule valeur pour le seuil. Ainsi, nous extrayons un cube de 256 voxels de côté pour différentes images et appliquons le seuillage automatique que nous venons de décrire. La figure 2.5 montre le niveau de gris de cette portion d'image pour des fractions volumiques de solide de 0,56 et 0,86. Nous voyons que le liquide, étant enrichi en cuivre, a un niveau de gris bien supérieur pour la fraction volumique la plus importante. Ainsi, le seuil optimal pour séparer liquide et solide doit augmenter durant la solidification. La figure 2.6, qui donne la valeur du seuil calculée en fonction de la fraction volumique de solide pour une épaisseur de

Chapitre 2

lissage de 4 niveaux de gris, confirme cette augmentation. Pour l'ensemble de ces seuils, nous avons vérifié que la binarisation est correctement effectuée.

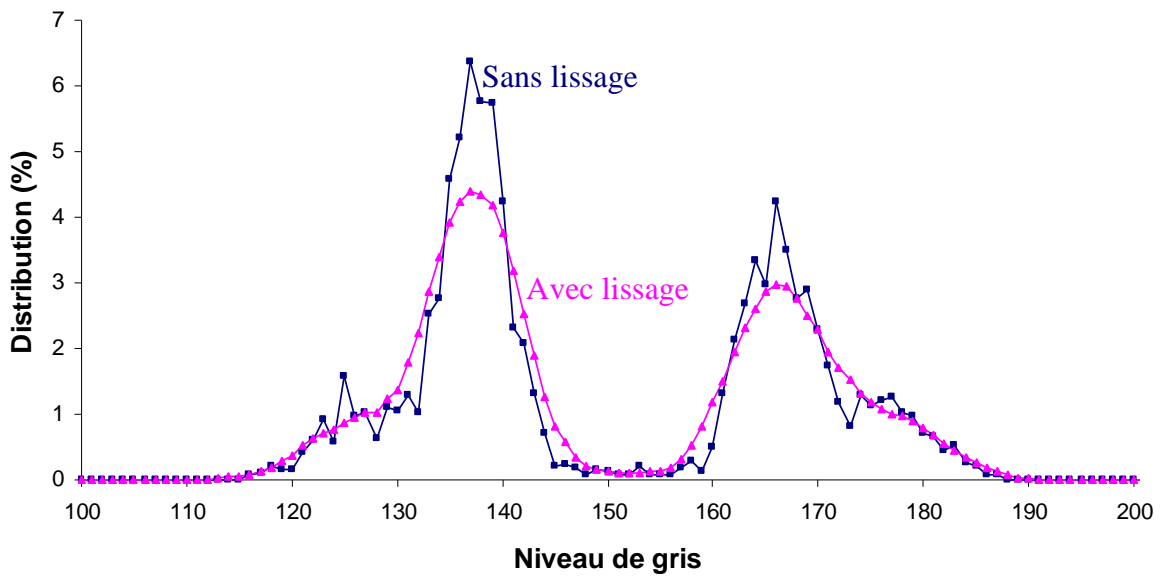


Figure 2.4. Histogramme de distribution des niveaux de gris pour la figure 2.3c.

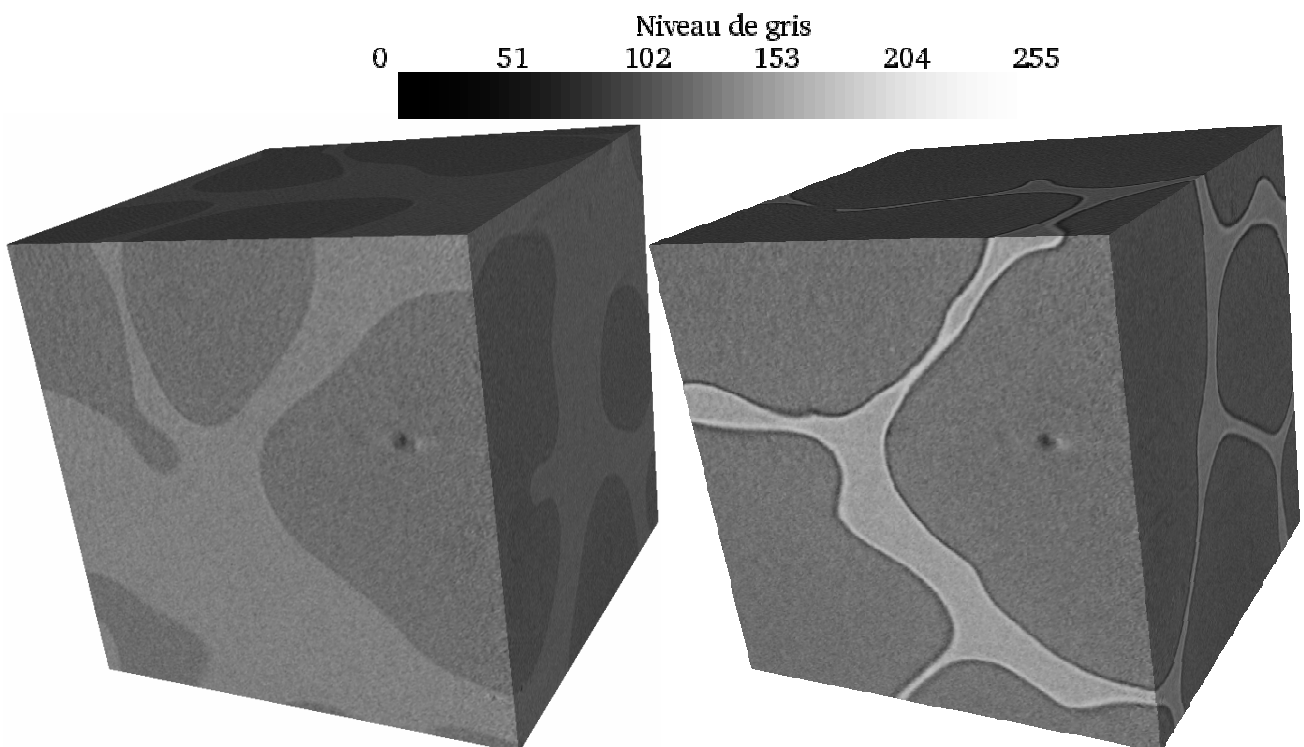


Figure 2.5. Images en niveau de gris pour des fractions solides de 0,56 (à gauche) et 0,86 (à droite).

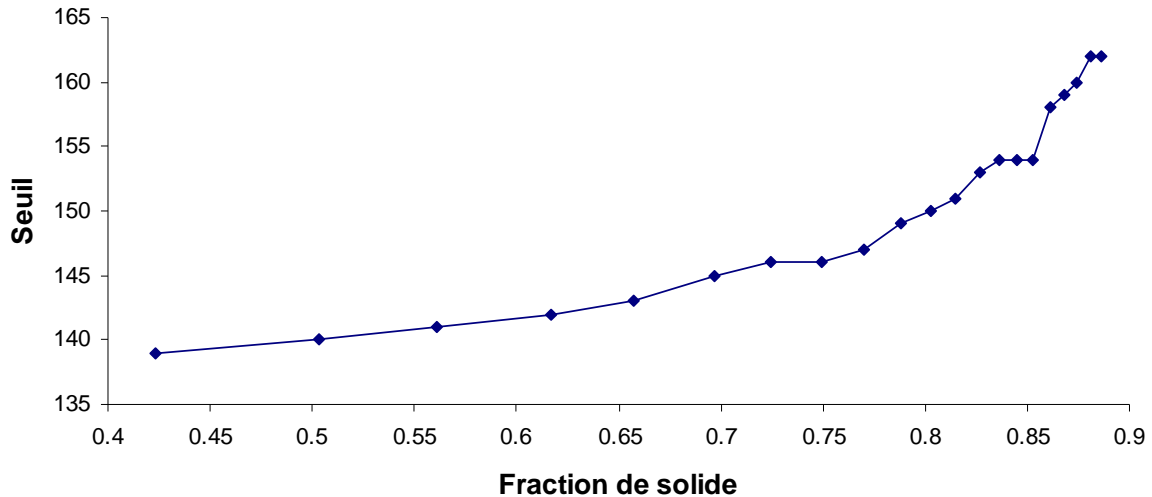


Figure 2.6. Evolution de la valeur du seuil en fonction de la fraction volumique de solide.

4. Maillage surfacique

Nous appelons maillage surfacique l'ensemble des facettes qui représente l'interface entre deux milieux. Nous parlons de maillage surfacique quelle que soit la dimension, alors que ces facettes sont des triangles en 3D et des arêtes en 2D. Nous détaillons à présent la manière de calculer ces maillages. Les codes correspondants, écrits en C++, sont donnés en annexe.

4.1. En deux dimensions

4.1.1. Principe de base

Pour introduire l'algorithme des marching cubes (MC) [Lor 87], Rajon & Bolch [Raj 03] commencent par une présentation en deux dimensions. Ils parlent alors de marching squares (MS). Nous allons faire de même en illustrant le principe de cet algorithme à l'aide d'un exemple simple. Nous considérons l'image binaire représentée sur la figure 2.7a. Cette image comprend 3 pixels dans chaque direction. Le liquide est représenté en blanc et le solide en gris.

Des carrés (appelés marching squares) sont superposés à l'image (figure 2.7b). Les 4 sommets de ces carrés sont situés au centre des pixels de l'image. Nous plaçons alors un point noir sur chaque sommet situé sur un pixel blanc (figure 2.7b). L'arête d'un MS est traversée par l'interface si un seul de ses deux sommets comporte un point. Un carré bleu est alors placé au milieu de l'arête traversée, comme indiqué à la figure 2.7c. Ces carrés bleus constituent les nœuds du maillage surfacique. Ces nœuds sont reliés entre eux pour former le maillage surfacique extrait (figure 2.7d).

4.1.2. Algorithme

Un seul carré (MS) est traité à la fois. Les 4 sommets sont utilisés pour savoir si le maillage surfacique intersecte le carré et pour déterminer la position des nœuds et la façon dont ces derniers sont reliés le cas échéant. Une fois qu'un carré est traité, le carré suivant est analysé

jusqu'à ce que toute l'image soit parcourue. Pour établir l'algorithme, il suffit ainsi d'étudier l'ensemble des possibilités pour un seul MS. Comme chacun des 4 sommets peut appartenir à deux phases, il existe 16 configurations possibles, indexées sur la figure 2.8, sur laquelle sont également tracées les arêtes du maillage surfacique associé. En considérant que certaines des configurations sont une rotation (configurations 1 et 2) ou l'inverse (configurations 1 et 14) d'une autre configuration, les possibilités peuvent se réduire à 4 formes de base [Raj 03] illustrées sur la figure 2.9.

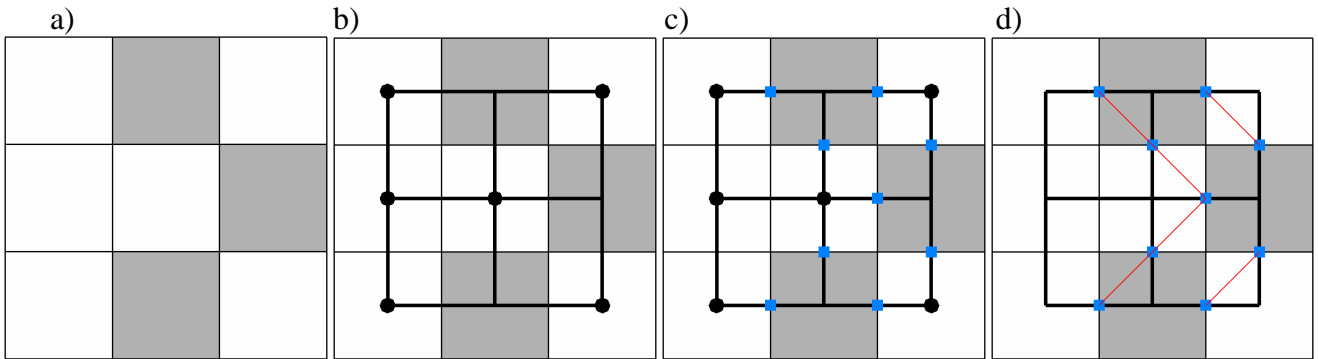


Figure 2.7. Illustration du principe de l'algorithme des marching squares.

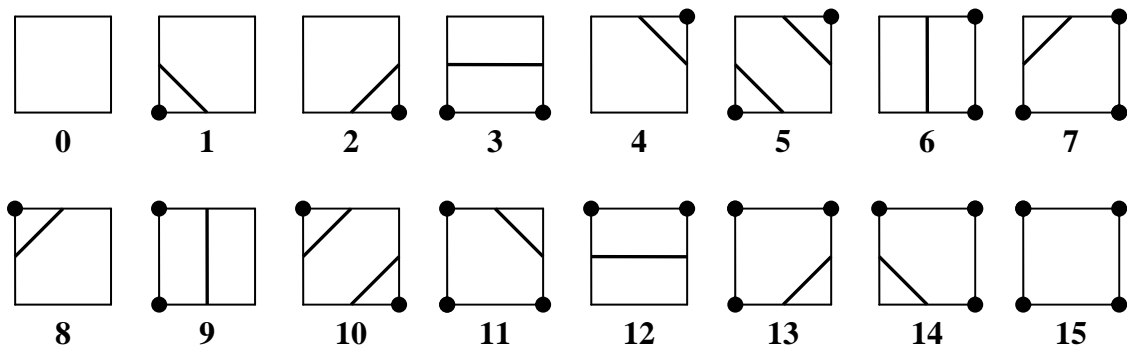


Figure 2.8. Représentation des 16 configurations possibles pour un marching square. Lorsqu'un sommet est situé dans un pixel liquide, un point est dessiné.

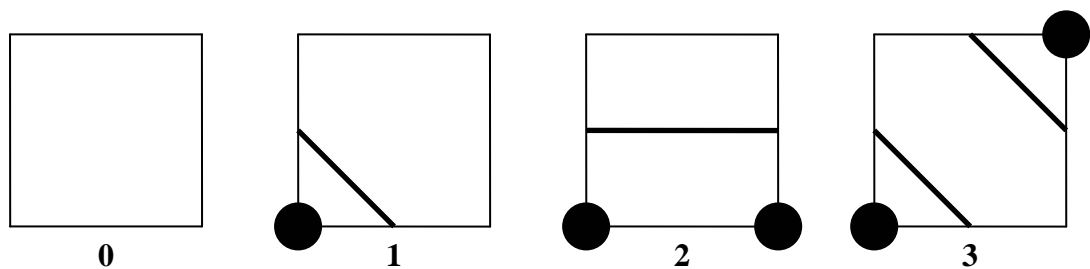


Figure 2.9. Représentation des 4 formes indépendantes obtenues en considérant les rotations et l'inversion.

Afin d'accélérer l'algorithme, la numérotation des différentes arêtes est précalculée pour chacune des 16 configurations et répertoriée dans une table de connectivité. Cette table est une matrice de 16 lignes (une pour chaque configuration) et 5 colonnes. Les 4 premières sont

utilisées pour donner le numéro des arêtes intersectées et la dernière vaut toujours 4. En effet, dès que la valeur 4 est rencontrée, cela signifie que le calcul sur le MS actuel est terminé. Pour déterminer la configuration sur chaque MS, nous calculons un index, indiqué sous les configurations de la figure 2.8. Pour cela, nous attribuons un bit à chaque sommet et sa valeur est mise à un si le pixel correspondant appartient au liquide, sinon le bit vaut zéro. La numérotation utilisée pour les sommets et les arêtes d'un MS est donnée sur la figure 2.10 en même temps que l'index et la partie de la table de connectivité associés à la configuration numéro 4.

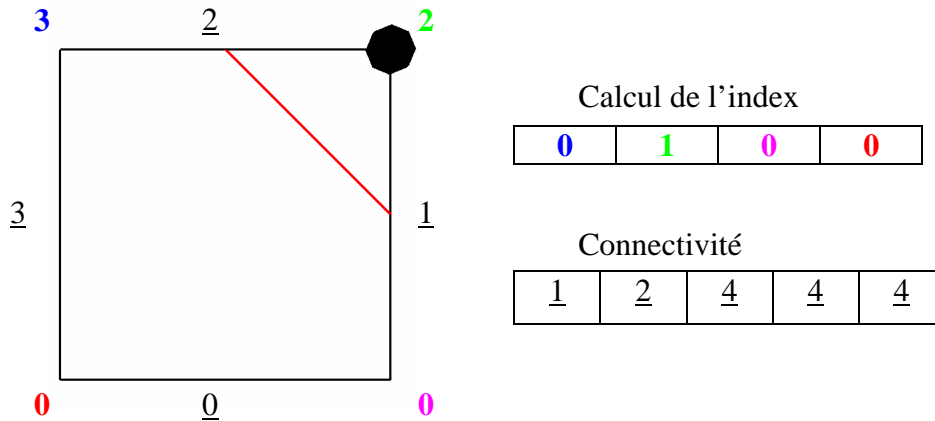


Figure 2.10. Numérotation des arêtes et des sommets d'un marching square, calcul de l'index et connectivité pour la configuration numéro 4 de la figure 2.8. Les numéros d'arête sont soulignés et les numéros des sommets sont écrits avec différentes couleurs.

En utilisant la méthode illustrée sur la figure 2.10 pour toutes les configurations de la figure 2.8, la table de connectivité est rapide et facile à créer. Toutefois, nous remarquons qu'il est également possible de seulement déterminer les arêtes intersectées pour les 4 formes de la figure 2.9 et d'utiliser ensuite une rotation et/ou une inversion pour accéder aux arêtes intersectées pour les 16 configurations. Cela peut sembler inutile en 2D mais se révèle très pratique en 3D comme nous allons le voir plus loin.

4.1.3. Calcul de la normale

Sur la figure 2.10 illustrant une configuration intersectée par un seul segment, nous devons faire un choix pour l'ordre dans lequel les deux centres des arêtes sont donnés. De ce choix dépend l'orientation de la normale au maillage surfacique. Cette normale est calculée comme suit :

$$\vec{n} = \vec{v}_{12} \wedge \vec{v}_{13}$$

avec \vec{n} la normale, \vec{v}_{12} le vecteur ayant pour origine le premier point du segment et pour extrémité le deuxième point et \vec{v}_{13} le vecteur ayant pour origine le premier point du segment et pour extrémité un troisième point, de mêmes abscisse et ordonnée que le deuxième et de cote unitaire.

Nous pouvons employer une normale pointant vers le liquide ou vers le solide, à condition de rester uniforme pour l'ensemble de la table de connectivité. Ainsi l'ordre des deux points pour les configurations 4 et 11 est impérativement différent. La figure 2.11 illustre les deux normales obtenues quand l'ordre des points est modifié.

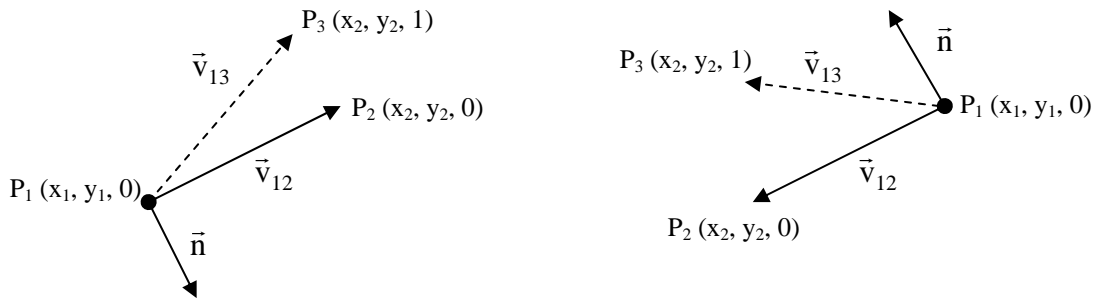


Figure 2.11. Détermination de la normale à partir des deux points du segment.

4.1.4. Ambiguïté

Pour les configurations 5 et 10, la surface de liquide est de 25 % de la surface du carré tandis que la surface du solide est de 75 %. Pourtant la symétrie de la forme 3 (figure 2.9) devrait impliquer une répartition égale pour les deux surfaces [Raj 03]. Ce choix qualifié de direct par Rajon & Bolch a donc tendance à diminuer la surface totale de liquide. Il est ainsi possible de considérer une géométrie inversée pour laquelle la surface de liquide est alors de 75 %. Une autre possibilité illustrée par Van Gelder & Wilhelms et désignée comme ambiguë [Gel 94] consiste à créer 4 zones différentes à l'intérieur du carré. Cependant, même si cette option respecte la symétrie du MS, elle nécessite la création d'un nœud supplémentaire au centre du MS et de 4 arêtes pour obtenir un maillage surfacique. Les trois configurations possibles sont illustrées sur la figure 2.12.

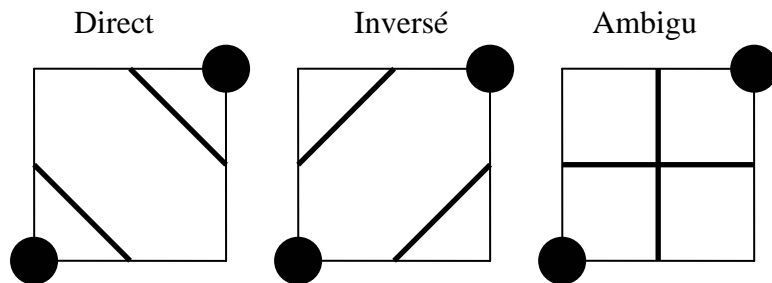


Figure 2.12. Différents maillages surfaciques pouvant être associés à une configuration ambiguë.

Toutefois, cette forme ambiguë a une occurrence tellement limitée que nous ne jugeons pas nécessaire d'y accorder une attention particulière. En outre, quel que soit le choix retenu, le maillage surfacique obtenu est fermé (hormis sur les bords de l'image), ce qui est suffisant pour le calcul du maillage volumique.

Nous signalons cependant que plusieurs auteurs ont mis au point des méthodes pour déterminer précisément le maillage surfacique lorsque ce cas ambigu est rencontré. Même si les techniques sont développées pour des applications 3D, certaines comme celles de Rajon & Bolch et Nielson & Hamann [Raj 03, Nie 91] sont tout à fait utilisables pour l'algorithme des MS.

Dans notre cas, nous faisons le choix de la méthode directe de la figure 2.12.

4.1.5. Cohérence

Nous avons pour le moment expliqué que le calcul sur chaque MS est indépendant, en mettant l'accent sur la connectivité du maillage surfacique. Deux choix sont possibles en ce qui concerne la détermination de la position des nœuds. Le premier consiste à continuer de procéder indépendamment sur chaque MS, en écrivant à chaque fois la position des nœuds et la connectivité correspondante, alors très facile à déterminer. Cette technique présente l'avantage de la simplicité mais elle a plusieurs inconvénients. Tout d'abord, les nœuds du maillage surfacique étant communs à deux MS (hormis aux bords de l'image), nous calculons et écrivons deux fois leurs coordonnées, ce qui a des répercussions en terme d'efficacité et de stockage. De plus, un fichier maillage de CIMLib est composé des coordonnées de l'ensemble des nœuds puis de la connectivité et il n'est donc pas possible d'écrire le maillage au fur et à mesure, ce qui nécessite un stockage ou une écriture intermédiaire des données calculées.

La deuxième solution utilise la notion de cohérence introduite par Lorensen & Cline [Lor 87], pour laquelle les coordonnées d'un nœud ne sont calculées et écrites qu'une seule fois. Cependant, ces deux auteurs ne détaillant pas la prise en compte de cette amélioration, nous avons mis au point notre propre algorithme.

Pour coder l'algorithme des MC dans un environnement parallèle, Nicod propose de parcourir l'image une première fois afin d'évaluer le nombre de nœuds et d'estimer le nombre d'éléments du maillage surfacique, ce qui lui permet d'équilibrer la charge de travail sur chaque partition [Nic 97]. Tout comme lui, nous parcourons l'image une première fois, en commençant par l'axe des abscisses et en continuant par celui des ordonnées (voir figure 2.13). A chaque fois que deux pixels voisins appartiennent à une phase différente, cela signifie qu'un nœud du maillage surfacique est situé entre ces deux pixels. Nous écrivons les coordonnées de ces nœuds au fur et à mesure du parcours et parcourons ensuite l'image avec les MS pour écrire la connectivité.

Dans ces conditions, les connectivités ne sont plus aussi faciles à déterminer. Nous avons en effet besoin de stocker des informations obtenues lors du calcul des nœuds. Pour cela, nous remplissons un vecteur qui nous donne le nombre de nœuds sur chaque ligne et sur chaque colonne et que nous appelons \overline{NN} . A l'image du codage au format Compressed Sparsed Row (CSR) dont le principe est donné par Saad [Saa 03], le nombre de nœuds est cumulé. Cela signifie par exemple que la valeur correspondante à la ligne 3 est la somme du nombre de nœuds présents sur les lignes 1, 2 et 3. Pour obtenir le numéro du premier nœud de la ligne 3, il nous suffit donc d'ajouter 1 au nombre de nœuds contenus sur les lignes 1 et 2. Lors du calcul de la connectivité, nous incrémentons le vecteur \overline{NN} afin de prendre en compte le nombre de nœuds déjà utilisés sur chaque ligne (et colonne). Pour incrémenter ce vecteur, les MS situés sur la colonne la plus à droite sont traités de manière différente puisque le nœud correspondant à l'arête numéro 1 de la figure 2.10 n'est pas partagée avec un autre MS.

Reprenons l'exemple de la figure 2.7. La figure 2.13 montre le sens de parcours de l'image pour écrire les coordonnées des nœuds et déterminer leur nombre sur chaque ligne et colonne, ainsi que le sens de parcours des MS. \overline{NN} s'écrit dans ce cas {0 ; 2 ; 3 ; 5 ; 5 ; 7 ; 9}. Pour le MS 1, la première arête intersectée est celle de droite. Cette arête correspond à la flèche numéro 5 de la figure 2.13 et donc à la cinquième valeur du vecteur \overline{NN} , ce qui signifie que 5 nœuds précèdent le nœud étudié. La connectivité est donc de 6. Ce nœud étant aussi présent sur le MS 2, nous n'incrémentons pas le vecteur \overline{NN} . Pour l'arête inférieure du MS 1, en appliquant la même méthode, nous trouvons une connectivité de 1. Puisque le nœud considéré ne sera pas réutilisé, l'élément correspondant du vecteur \overline{NN} est cette fois-ci incrémenté, ce qui nous permet de trouver la bonne connectivité pour l'arête inférieure du MS 2. Sur ce MS,

le nœud situé au milieu de l'arête de droite n'appartenant qu'au MS 2, l'élément associé du vecteur \overrightarrow{NN} est incrémenté et l'arête de droite du MS 4 située sur la même colonne pourra ainsi être traitée correctement.

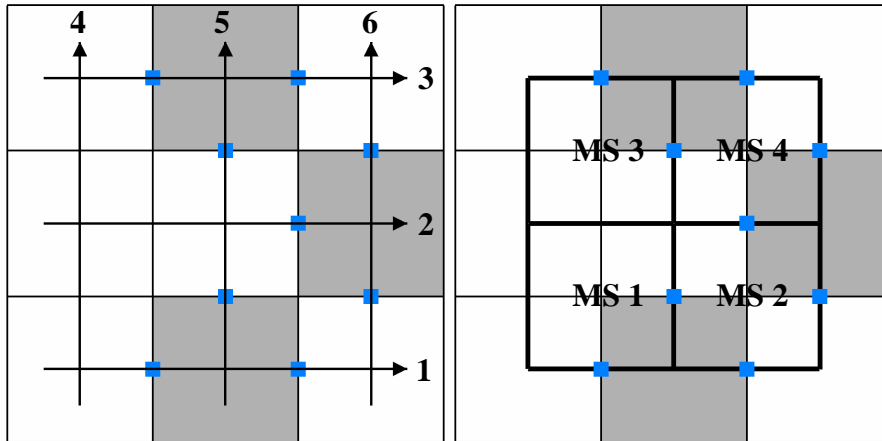


Figure 2.13. Sens de parcours pour écrire les coordonnées des nœuds (à gauche) et ordre des marching squares considérés (à droite).

Comme nous venons de le voir, quand l'arête de droite d'un MS non situé sur le bord droit de l'image est intersectée, le vecteur \overrightarrow{NN} n'est pas réactualisé, car cette arête correspond à l'arête de gauche du MS suivant. Ce n'est pas le cas pour les arêtes supérieures, même si elles correspondent à l'arête inférieure du MS de la ligne du dessus. En effet, le parcours se faisant de gauche à droite, il est nécessaire de tenir compte du nombre d'arêtes supérieures intersectées, même quand le MS n'est pas situé au bord de l'image. Toutefois, ces nœuds devant être utilisés pour la ligne située au dessus de la ligne en cours, nous utilisons une variable pour garder en mémoire le nombre d'arêtes supérieures intersectées sur une ligne. Cette variable est remise à 0 lorsque nous passons au parcours de la ligne suivante.

Le tableau 2.2 donne le maillage obtenu pour les calculs avec et sans cohérence. Ce tableau montre bien que le calcul sans cohérence implique une connectivité très simple à déterminer et que celui avec cohérence permet de ne pas dupliquer les nœuds du maillage surfacique. Dans ce cas, nous considérons que les pixels font $2 \mu\text{m}$ de côté chacun.

Si une zone d'ombre devait persister dans l'esprit du lecteur à la suite de ce passage, nous l'encourageons à regarder avec attention le code fourni en annexe de cette thèse.

Chapitre 2

9	2	6	2	12	2	6	2
2	1			3	2		
4	1			2	1		
4	3			4	1		
2	5			5	2		
4	5			4	3		
3	2			3	2		
3	4			2	5		
5	2			3	4		
5	4			3	4		
				4	3		
				5	4		
				4	5		
6	1			1	2		
2	8			3	4		
3	6			5	6		
4	7			7	8		
7	3			9	10		
9	5			11	12		

Tableau 2.2. Maillage surfacique obtenu par l'algorithme des marching squares utilisé sur l'image de la figure 2.7. La première ligne représente l'en-tête du maillage, c'est-à-dire le nombre de nœuds, la dimension spatiale, le nombre d'éléments et le nombre de nœuds par élément. Les lignes suivantes donnent l'abscisse et l'ordonnée de chaque nœud et celles d'après la connectivité du maillage. Calcul avec cohérence à gauche et sans cohérence à droite.

4.1.6. Image en niveaux de gris

Nous travaillons jusqu'à présent avec des images binaires, pour lesquelles les nœuds du maillage surfacique sont placés entre deux pixels, au milieu des arêtes des MS. Sur des images en niveaux de gris, Lorensen & Cline [Lor 87] adoptent une solution plus élégante en calculant la position des nœuds par une interpolation linéaire. En procédant de la sorte, un MS n'est plus nécessairement intersecté au centre de ses arêtes. Ainsi pour l'arête inférieure, l'ordonnée du nœud est toujours la même mais son abscisse est calculée comme suit :

$$x = x_G + \frac{S - N_G}{N_D - N_G} a$$

Avec x l'abscisse du nœud, x_G l'abscisse du centre du pixel de gauche (coin inférieur gauche du MS), a la longueur d'un pixel, S le seuil entre les deux phases, N_G le niveau de gris du pixel de gauche et N_D le niveau de gris du pixel de droite.

De cette façon le nœud se situe sur l'arête du MS entre les sommets inférieur gauche et inférieur droit. Cela permet d'obtenir une surface plus lisse, comme l'illustre la figure 2.14. Il faut cependant prendre des précautions lorsque la valeur d'un pixel est égale à celle du seuil. En effet, le nœud associé se situe alors sur un sommet du MS. Si deux arêtes contiguës du MS sont intersectées, cela signifie par conséquent que deux nœuds du maillage surfacique ont les

mêmes coordonnées, ce qui pose deux problèmes. En premier lieu, une arête du maillage surfacique se réduisant à un seul nœud, sachant que ce nœud est déjà contenu dans au moins une autre arête, n'a aucun intérêt. Le temps de calcul consommé pour calculer cette arête et l'écrire est donc vain. Nous ne sommes toutefois pas dans l'obligation de corriger ce problème car il ne diminue que légèrement l'efficacité de l'algorithme.

Le deuxième problème, plus important, est le suivant. L'arête étant constituée d'un seul nœud, il n'est pas possible de calculer une normale à cette arête. Or, la méthode par immersion de volume, détaillée dans la partie suivante, nécessite une normale par arête pour construire le maillage volumique. Une solution pour régler les deux problèmes consiste à intégrer des tests supplémentaires dans l'algorithme pour traiter les cas pour lesquels le niveau de gris d'un pixel est égal à celui du seuil. Mais il faut pour cela modifier à la fois les parties concernant l'écriture des nœuds et l'écriture des éléments, ce qui complique la mise en œuvre et ralentit l'exécution. Pour garder la structure actuelle du code, nous préférons plutôt modifier très légèrement la valeur du seuil. Ainsi, dans la figure 2.14b, le maillage surfacique est déterminé avec un seuil de 150,999 au lieu de 151. Cela ne change quasiment rien dans le résultat, d'autant plus que la valeur du seuil est imprécise à plusieurs niveaux de gris près. En revanche, les deux nœuds confondus avec un seuil de 151 ne le sont plus, ce qui permet de calculer une normale bien orientée. Nous remarquons que cette stratégie est également utilisée par Van Gelder & Wilhelms [Gel 94].

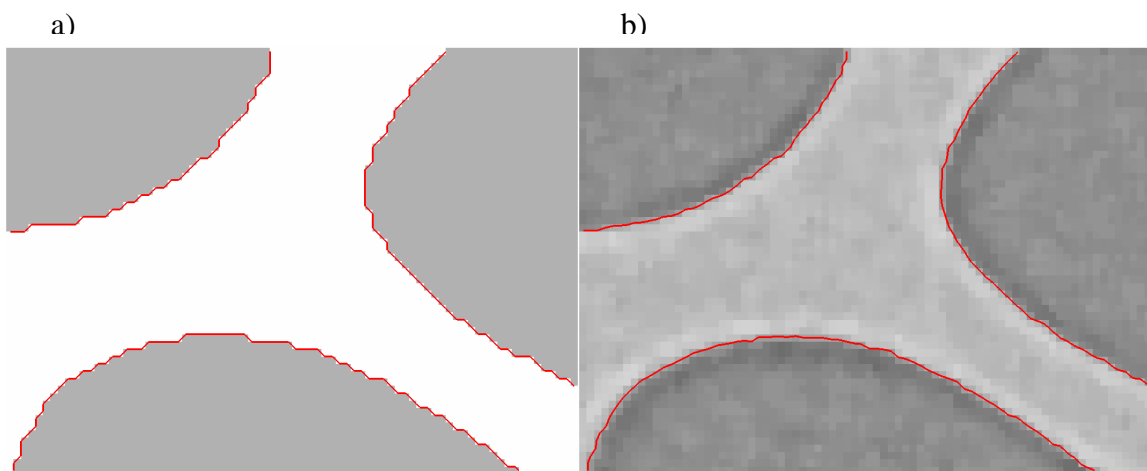


Figure 2.14. Maillage surfacique extrait à partir de la figure 2.3d (en trait continu rouge). a) Avec l'image binaire. b) Avec l'image en niveaux de gris.

4.2. En trois dimensions

En dépit de sa popularité très limitée dans la littérature, nous avons choisi de largement détailler l'algorithme des MS, car il aide grandement à comprendre, et ce de manière simplifiée, la transposition à des cas 3D, connue sous le nom d'algorithme des marching cubes (MC). Rajon & Bolch utilisent d'ailleurs tout comme nous les MS à des fins purement explicatives pour le lecteur [Raj 03].

4.2.1. Principe

Le principe de l'algorithme est très similaire à celui des MS. Chaque MC est analysé indépendamment et l'exécution se termine quand toute l'image a été parcourue. Chacun des 8

sommets d'un MC ayant deux états possibles, il existe 256 configurations. Lorensen & Cline [Lor 87], en utilisant les équivalences par inversion et par rotation, réduisent l'étude à 15 formes de base et il n'est donc pas nécessaire de représenter les 256 configurations. Ces formes sont montrées sur la figure 2.15 avec les triangulations correspondantes. Une triangulation est la collection de triangles représentant le maillage surfacique. Sur la figure 2.16, nous illustrons les 24 rotations [Nie 02] associées à la forme numéro 6 de la figure 2.15. Déterminer la table de connectivité des 256 configurations n'est pas trivial. Pour Lorensen & Cline, cela peut même être une étape propice aux erreurs [Lor 87]. Il faut dans un premier temps numéroter les sommets d'un MC pour pouvoir calculer l'index, et numéroter les arêtes afin de pouvoir déterminer la connectivité. La numérotation choisie, similaire à celle de Lorensen & Cline, est indiquée sur la figure 2.17. Il est ensuite conseillé d'utiliser les triangulations des formes indépendantes et d'opérer les rotations adéquates. Pour nous aider dans cette entreprise, certains auteurs comme Nielson et al [Nie 02] ou encore Lewiner et al. [Lew 03] n'hésitent pas à fournir des tables complètes ou leur propre code. La normale à la facette se calcule avec la même formule qu'en 2D en prenant cette fois-ci pour troisième point le troisième point du triangle. L'interpolation linéaire nécessaire à la détermination des coordonnées des nœuds est elle aussi directement transposée en 3D.

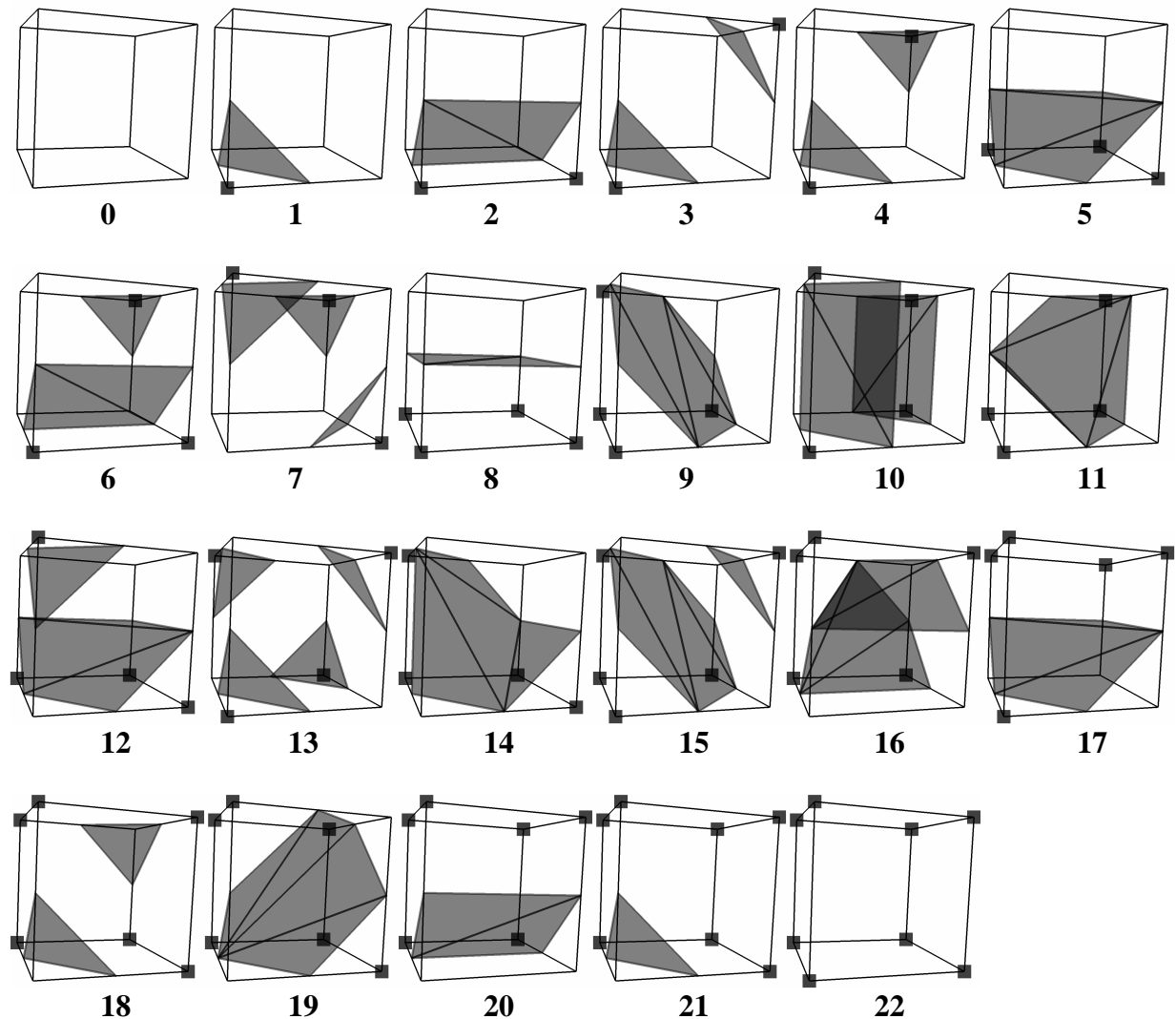


Figure 2.15. De 0 à 14, formes utilisées par Lorensen & Cline, en considérant la rotation et l'inversion. De 0 à 22, formes utilisées par Nielson et al., en considérant seulement la rotation.

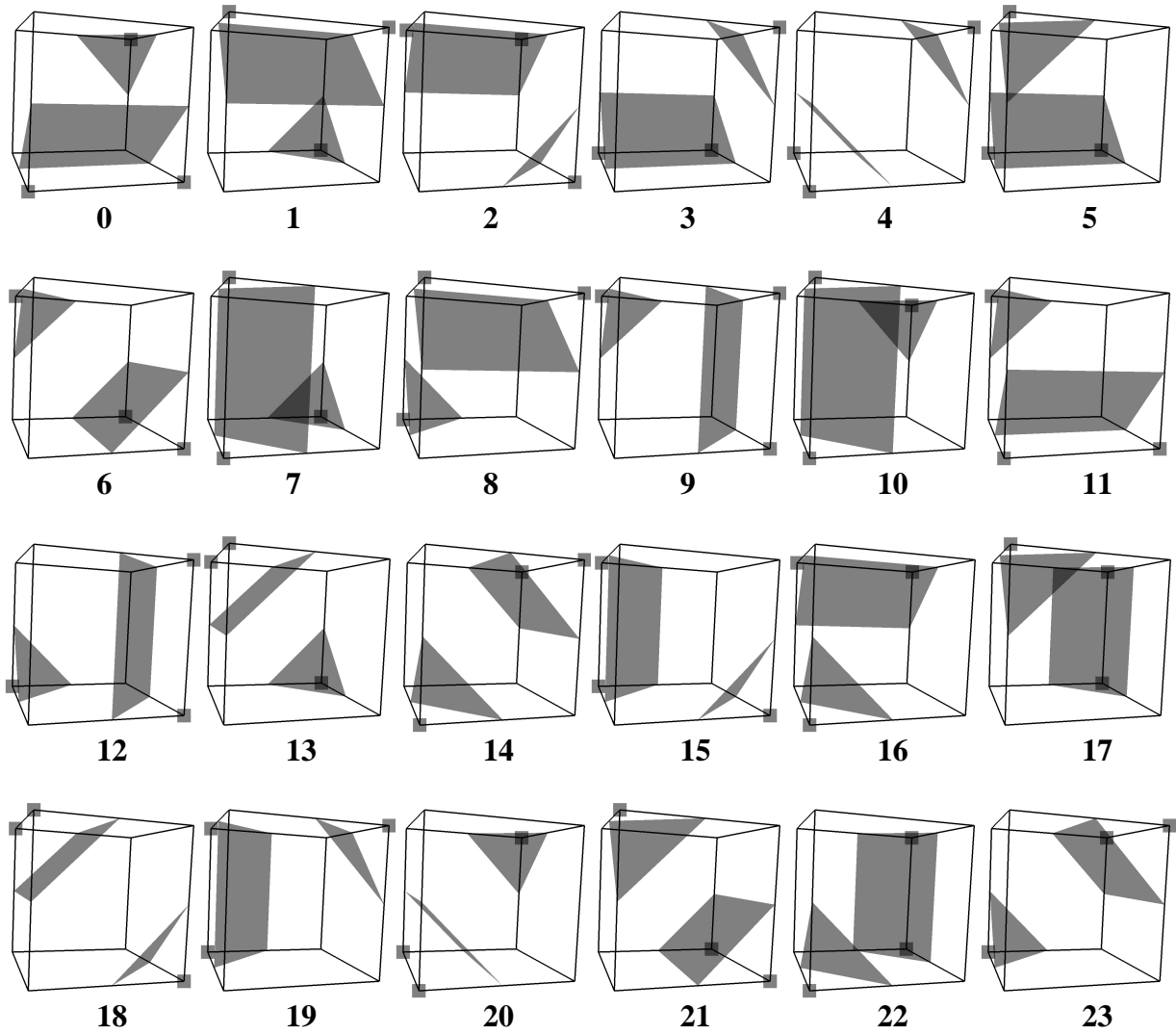


Figure 2.16. Représentation des 24 triangulations obtenues par rotation de la forme 6.

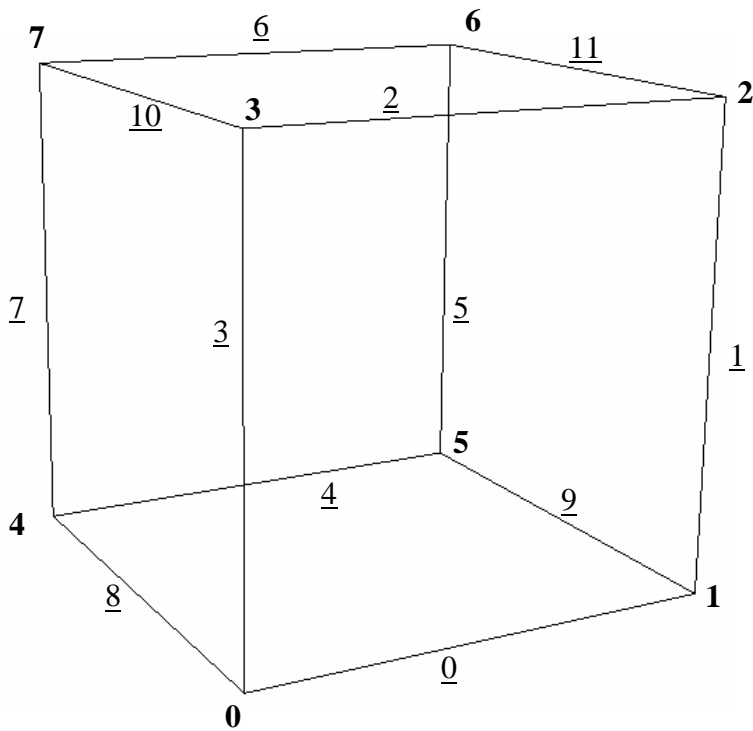


Figure 2.17. Numérotation des sommets et des arêtes d'un marching cube.

4.2.2. Ambiguïté

Avec les 15 formes de base de Lorensen & Cline, nous retrouvons le problème d'ambiguïté rencontré en 2D, à la différence près que dans ce cas, le maillage n'est plus fermé, ce qui peut poser des problèmes lors des calculs subséquents. Ainsi, Nielson et al. [Nie 02], entre autres auteurs, proposent de ne pas considérer l'inversion pour établir 23 formes indépendantes, montrées sur la figure 2.15. De cette façon, le maillage surfacique obtenu, troué en utilisant seulement les 15 formes originelles, est bien fermé, comme le montre la figure 2.18.

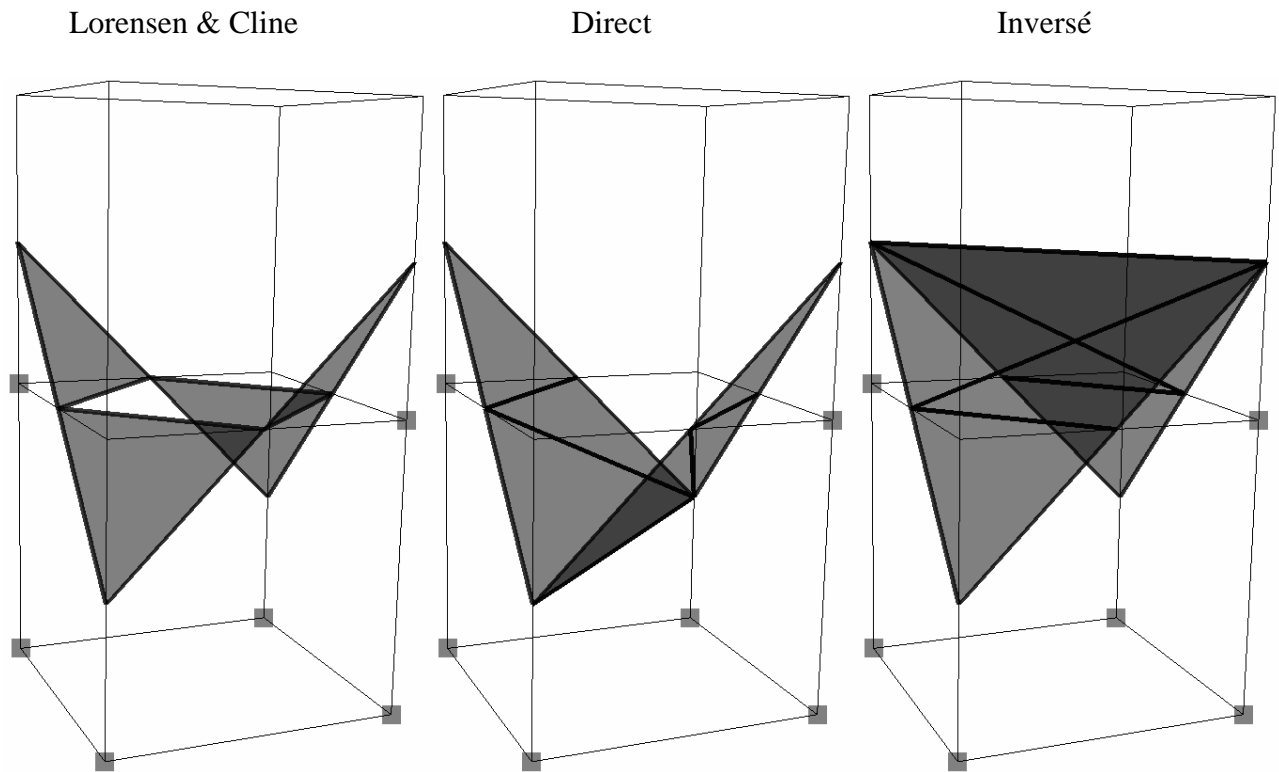


Figure 2.18. Illustration du problème de surface non fermée obtenu avec l'algorithme original de Lorensen & Cline et résolu par les configurations supplémentaires de Nielson et al.

Sur cette figure représentant la triangulation associée à 2 MC contigus sous la forme numéro 3 de la figure 2.15, nous pouvons voir que le maillage surfacique n'est pas fermé. En ajoutant la forme numéro 19, nous remédions à ce problème avec la méthode directe, dont le principe est le même qu'en 2D. Il est aussi possible d'utiliser la méthode inverse, pour laquelle les triangulations associées aux formes numéro 3 et 19 sont interchangées.

Au final, la table de connectivité est calculée en utilisant trois tables différentes. Ces trois tables nous donnent respectivement la triangulation pour les 23 formes de la figure 2.15, le changement de numérotation de chaque arête pour les 24 rotations, ainsi que la forme et la rotation associées à chacune des 256 configurations. Ces deux dernières sont données par Nielson et al. [Nie 02], mais leur numérotation des arêtes et des sommets d'un MC étant différente de la nôtre, nous avons dû modifier ces tables.

4.2.3. Cohérence

Comme en 2D, nous utilisons deux étapes distinctes pour le calcul du maillage surfacique, à savoir l'écriture des nœuds et l'écriture de la connectivité. Le principe pour écrire les nœuds et calculer le nombre de nœuds par ligne est également similaire. Simplement, le vecteur \overrightarrow{NN} comprend beaucoup plus d'éléments en 3D. En effet, si nous appelons L, H et P la longueur, la hauteur et la profondeur d'une image, c'est-à-dire le nombre de voxels de l'image suivant l'abscisse, l'ordonnée et la cote, ce vecteur comprend $H + L + 1$ éléments en 2D et $H.P + L.P + L.H + 1$ éléments en 3D.

Pour écrire la connectivité, nous incrémentons à nouveau le vecteur \overrightarrow{NN} durant le parcours. Par rapport au cas 2D, il faut déterminer le comportement de l'algorithme pour douze arêtes au lieu de quatre, et davantage de cas particuliers sont rencontrés sur les bords de l'image. En outre, un sommet pouvant être commun à plusieurs triangles au sein d'un même MC, nous réactualisons seulement le vecteur \overrightarrow{NN} après l'écriture de toutes les facettes du maillage surfacique intersectant le MC.

Le principe pour prendre en compte la cohérence en 3D étant semblable à celui en 2D, nous n'entrons pas davantage dans les détails de la méthode. Nous nous contentons simplement d'encourager le lecteur à bien assimiler la méthode en 2D pour comprendre celle en 3D. Les codes figurant en annexe devraient pouvoir l'aider à cet effet.

4.2.4. Performances

4.2.4.1. Temps de calcul

Notre code écrit en C++ supporte seulement une utilisation séquentielle. Le temps d'exécution pour extraire le maillage surfacique associé à l'image de la figure 2.5 pour une fraction volumique de solide de 0,86 et montré sur la figure 2.19, est de 9,3 s sur un PC de 3,25 Go de RAM avec un processeur Intel Core de fréquence 2,99 GHz, pour une image qui comprend plus de 16 000 000 de voxels et un maillage surfacique de 1 148 530 éléments. Nous remarquons que la lecture de l'image de départ et l'écriture du maillage surfacique sont responsables de 70 % du temps d'exécution et le calcul à proprement parler dure donc moins de 3 s.

Comme nous voulons seulement extraire le maillage surfacique au tout début de notre simulation et compte tenu du temps de calcul très court, nous ne jugeons pas nécessaire d'optimiser davantage notre code. Nous tenons toutefois à signaler que son application à des images de plus grande taille ne pose pas de problème, puisque le temps de calcul est proportionnel à la taille de l'image et au nombre de nœuds du maillage surfacique. Pour une analyse pertinente du temps de calcul associé à l'utilisation des marchings cubes et la proposition d'une version parallèle performante de l'algorithme, nous renvoyons le lecteur à la thèse de Nicod [Nic 97].

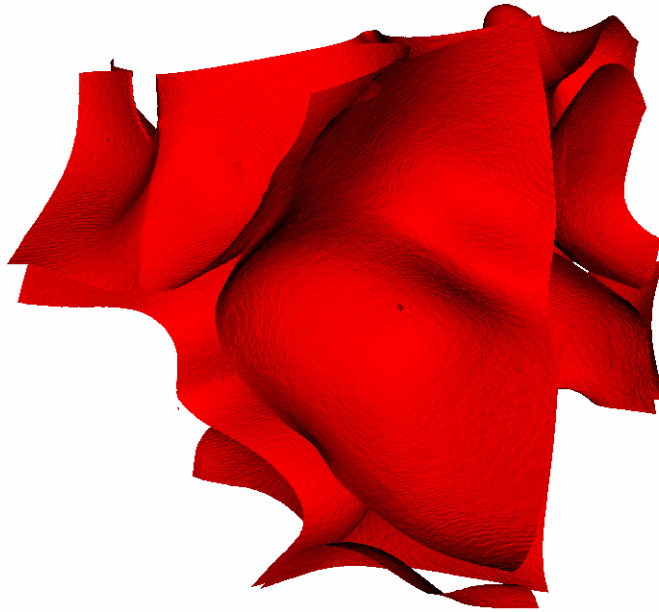


Figure 2.19. Maillage surfacique extrait de la figure 2.5 pour une fraction volumique de solide de 0,86. Ce maillage présentant 1 148 530 facettes triangulaires a été obtenu à partir d'une image de 16 777 216 voxels (16 Mo) en 9,3 s.

4.2.4.2. Mémoire

Il est courant qu'un algorithme soit davantage limité par la mémoire utilisée que par son temps d'exécution. D'ailleurs, l'augmentation de la mémoire disponible est une des principales motivations de Nicod pour paralléliser l'algorithme des MC [Nic 97]. De nos jours, les ordinateurs disposent de beaucoup plus de mémoire vive que dans les années 90, mais les images à traiter étant elles aussi plus volumineuses, le problème peut toujours se poser.

N'ayant finalement besoin de travailler qu'avec des images de taille modeste pour calculer nos maillages surfaciques (inférieure à 100 Mo), nous n'avons pas rencontré de problème de mémoire dans notre cas. Nous tenons toutefois à donner notre sentiment sur cette notion importante, afin de pouvoir envisager dans l'avenir le traitement d'images de taille beaucoup plus grande.

Actuellement, nous chargeons l'ensemble de l'image en mémoire, afin de simplifier et d'accélérer l'algorithme. Nous stockons en supplément le vecteur \overrightarrow{NN} , qui représente en mémoire l'équivalent de 12 couches supplémentaires de l'image (en supposant une image cubique de même dimension sur les trois axes), avec 4 octets par élément et un parcours suivant les trois axes. Toutes les autres variables utilisées dans notre calcul sont des scalaires. Contrairement à Nicod qui souhaite visualiser directement ses résultats [Nic 97], nous n'avons pas besoin de stocker le maillage surfacique que nous écrivons au fur et à mesure, ce qui représente un avantage substantiel. En effet, le maillage montré sur la figure 2.19, une fois écrit, occupe une place de 37 Mo, contre 16 Mo pour l'image de départ en voxel.

Si malgré ce stockage relativement économique, la mémoire vient à manquer, il existe une solution, que nous pouvons appliquer en utilisant un chargement partiel de l'image. Cette solution est employée par Lorensen & Cline [Lor 87]. Elle nécessite seulement des modifications mineures de l'algorithme et n'occasionne qu'un très léger surcoût de calcul. C'est donc la méthode que nous préconiserions.

4.2.5. Améliorations possibles

Pour dégager des pistes d'amélioration cohérentes, nous devons nous concentrer sur l'utilisation que nous faisons du maillage surfacique obtenu. Nous détaillons cela dans la prochaine partie mais tenons d'ores et déjà à préciser que la méthode par immersion de volume n'est pour le moment pas entièrement parallélisée dans CIMLib. Ainsi, la parallélisation de l'algorithme des MC, que nous avons déjà évoquée, ne semble avoir qu'un intérêt limité. Cependant, si pour une raison quelconque cet intérêt devait croître, nous conseillerions l'étude du papier de Newman & Li qui résume plusieurs approches parallèles [New 06].

Newman & Li donnent dans le même article un inventaire des méthodes utilisées pour accélérer l'algorithme de base [New 06]. Ils évoquent notamment des techniques permettant de minimiser le nombre d'opérations effectuées dans ce qu'ils appellent les cellules inactives, c'est-à-dire les *marching cubes* non intersectés par le maillage surfacique. Ils expliquent également les moyens adoptés dans la littérature pour déterminer l'isosurface dans les cas ambigus, avec entre autres l'utilisation d'une interpolation de degré supérieur.

Une fois la construction du maillage surfacique terminée, ajouter un post-traitement dans le but de diminuer le nombre de facettes est une perspective intéressante. Cela permettrait de diminuer la taille de nos fichiers et d'accélérer les calculs ultérieurs. Si nous travaillons sur des images binaires, il est tout d'abord possible de réduire le nombre de triangles sans que la surface résultante en soit affectée. Effectivement, les possibilités d'intersection sont très limitées dans ce cas et il est fréquent que la surface associée à plusieurs triangles contigus de normale identique puisse être représentée par un nombre de triangles moins important. C'est d'ailleurs le cas sur la triangulation directe de la figure 2.19 pour laquelle 2 triangles suffiraient là où 6 sont pourtant présents.

Pour des images en niveaux de gris, diminuer le nombre de facettes sans changer la surface est beaucoup moins évident. Toutefois, la surface construite résultant d'une discrétisation imparfaite, des auteurs proposent de la modifier en réduisant le nombre d'éléments du maillage, sans pour autant obtenir une surface qui pourrait être qualifiée de moins précise. Garland et Heckbert [Gar 99, Hec 99] se servent pour cela d'éléments triangulaires, du même type que les éléments présents avant la réduction. Mais il est également possible d'utiliser des formes de degré supérieur, tels que les B-splines rationnelles non uniformes [Pie 96].

5. Maillage volumique

La construction du maillage volumique avec Amira[®] [Ami 10] s'appuie sur une représentation explicite de l'interface, par la méthode avec suivi de front [Fre 98, Fre 99]. Avec ce type d'approche, les nœuds du maillage volumique situés à l'interface sont superposés aux nœuds du maillage surfacique. Des nœuds sont ainsi positionnés près des interfaces et reliés aux triangles du maillage surfacique pour constituer une première couche d'éléments, appelée front. D'autres nœuds sont alors ajoutés près du front pour établir une autre couche d'éléments, et ce jusqu'à remplissage intégral du volume.

Avec des maillages surfaciques de taille importante, il faut soit diminuer le nombre de nœuds, soit faire en sorte que chaque facette triangulaire du maillage surfacique ne serve pas de point de départ au calcul du maillage volumique. Dans le cas contraire, ce dernier ne peut pas être construit car le nombre de nœuds résultant serait trop important. En outre, si les éléments sont généralement de bonne qualité au voisinage de l'interface, des difficultés sont rencontrées à l'endroit où plusieurs fronts se rejoignent.

Dans CIMLib, nous préférons représenter les interfaces de manière implicite, avec une fonction level set (distance signée) définie aux nœuds du maillage et dont l'isovaleur 0

représente l'interface. Nous avons donc un maillage volumique qui comporte les différentes phases. Pour cela, nous utilisons la méthode par immersion de volume décrite par Hachem et al. [Hac 10], qui consiste à travailler à l'amélioration d'un maillage de fond pour représenter de manière implicite le maillage surfacique considéré. Cette méthode d'immersion de volume comporte trois étapes, qui sont : (i) le calcul de la distance signée, (ii) la construction d'un maillage adapté au calcul et (iii) le mélange des propriétés physiques. Nous détaillons les deux premières dans cette partie et le mélange des propriétés dans le chapitre suivant. Afin de pouvoir comparer les représentations en voxels et celles en éléments finis, nous expliquons également comment utiliser la fonction level set pour calculer le volume de chaque phase et l'aire de l'interface entre deux phases.

5.1. Calcul de la distance

Nous souhaitons déterminer la distance signée entre une surface constituée d'une collection de facettes et l'ensemble des nœuds du maillage volumique. L'opération élémentaire de ce calcul est donc la détermination de la distance entre un point et un segment dans un espace 2D ou celle entre un point et un triangle dans un espace 3D. Si n est le nombre de nœuds et m le nombre de facettes, il suffit donc d'effectuer m fois cette opération pour chaque nœud, de prendre la valeur absolue minimale et d'y associer le signe correspondant. Cette méthode que nous qualifions de linéaire est simple à utiliser mais nécessite $n.m$ calculs de distance élémentaire, ce qui peut prendre beaucoup de temps, en sachant que maillages surfacique et volumique peuvent contenir respectivement plusieurs millions de triangles et de nœuds. Afin d'éviter cet inconvénient majeur, l'algorithme employé utilise le constat selon lequel la distance entre un nœud et des facettes qui en sont très éloignées n'est pas nécessaire à calculer précisément par rapport à celle entre ce même nœud et des facettes qui en sont très proches. Pour éviter des calculs superflus, nous nous servons ainsi d'un système de boîtes hiérarchiques englobant chacune un nombre de facettes défini.

5.1.1. Méthode linéaire

En 2D, pour déterminer la distance entre un nœud et un segment, nous pouvons calculer la position du projeté orthogonal du nœud sur la droite associée au segment. Si le projeté se situe sur le segment alors la distance est la distance entre le projeté et le nœud. Sinon, il s'agit de la distance entre le nœud et un des deux sommets de l'arête, selon le côté duquel est situé le projeté. Les trois configurations possibles sont illustrées sur la figure 2.20.

En 3D, nous calculons les coordonnées du projeté orthogonal du nœud sur le plan défini par la facette triangulaire. La distance entre le nœud et le triangle dépend alors de la région dans laquelle se situe ce projeté. Il peut s'agir de la distance à un sommet du triangle, à une arête du triangle ou au triangle lui-même. Si le projeté se trouve à l'intérieur du triangle, la distance est alors directement égale à la distance entre le projeté et le nœud. Si la distance la plus courte est la distance à une arête, le projeté orthogonal du projeté dans le plan sur cette arête nous donne le point le plus proche du nœud. Les différentes configurations sont illustrées sur la figure 2.20. L'algorithme est donné par Schneider & Eberly [Sch 02].

Cet algorithme permet de calculer la distance. Pour calculer la distance signée, nous utilisons la normale à la facette la plus proche du nœud, à l'aide de l'expression suivante :

$$\phi(\vec{x}) = \begin{cases} +d(\vec{x}) & \text{si } (\vec{x} - \vec{x}_p) \cdot \vec{n} < 0 \\ -d(\vec{x}) & \text{si } (\vec{x} - \vec{x}_p) \cdot \vec{n} > 0 \end{cases}$$

Avec \bar{x} et \bar{x}_p les coordonnées du nœud et de son projeté orthogonal sur le segment (respectivement le plan), \vec{n} la normale à la facette du maillage surfacique, d la distance et ϕ la level set (ou distance signée).

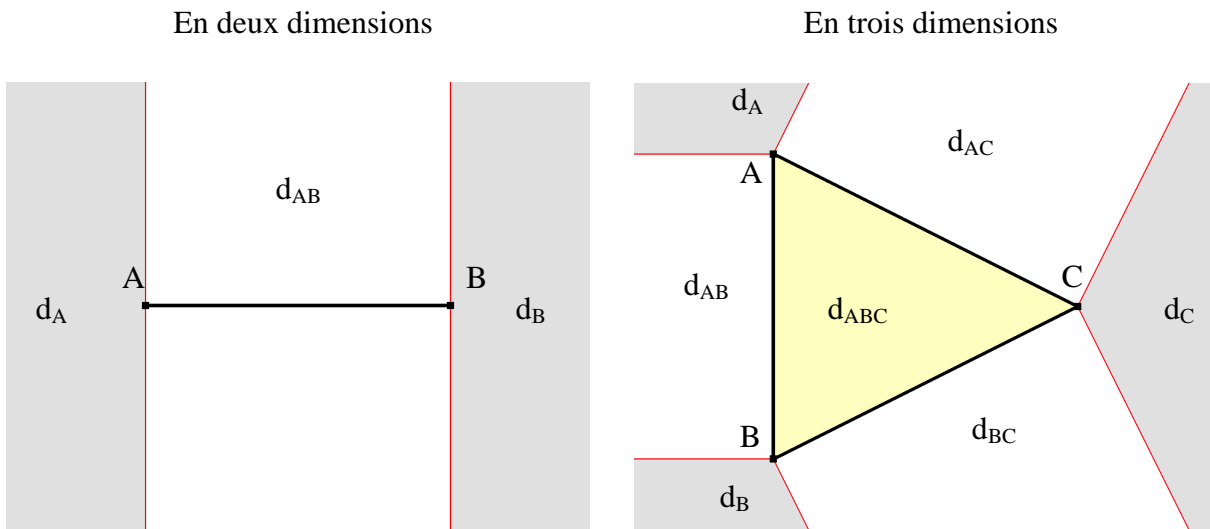


Figure 2.20. Représentation des différentes zones utilisées pour calculer la distance entre un nœud du maillage volumique et une facette du maillage surfacique. En noir, représentation de la facette. En rouge, délimitation des différentes zones. d_A , d_B , d_C , d_{AB} , d_{AC} , d_{BC} et d_{ABC} sont respectivement les zones dans lesquelles les points A, B, C, les arêtes AB, AC, BC et le triangle ABC sont les plus proches du nœud.

Lorsque pour un nœud particulier, le point le plus proche du maillage surfacique est un sommet partagé par deux facettes contiguës, la formule ci-dessus ne suffit pas toujours à calculer le signe de manière univoque. Pour un cas 2D, si ces deux arêtes forment un angle obtus, il existe en effet une région de l'espace pour laquelle le signe est négatif par rapport à une arête et positif par rapport à une autre, alors que la distance est la même en valeur absolue. Bruchon et al., qui ont implémenté l'algorithme du calcul de la distance dans CIMLib, illustrent ce constat et choisissent le signe donné par l'arête permettant de calculer une projection de meilleure qualité [Bru 09].

5.1.2. Méthode hiérarchique

Le système de représentation hiérarchique est détaillé par Bruchon et al. [Bru 09]. Nous le décrivons en prenant le maillage surfacique montré sur la figure 2.14b pour exemple. Nous englobons ce maillage avec un rectangle, qui représente la première boîte de notre structure hiérarchique. Si cette boîte comprend un nombre de facettes trop important, nous la divisons en 4 en 2D (8 en 3D) boîtes de taille identique. Le nombre de facettes contenues à l'intérieur de chacune de ces nouvelles boîtes est évalué. S'il est trop important, la boîte est à son tour divisée en 4 plus petites boîtes qui englobent la boîte de niveau supérieur. L'algorithme se termine quand aucune des boîtes de plus bas niveau ne contient un nombre d'arêtes trop important. Ce nombre est en réalité fixé par l'utilisateur et nous voyons son influence sur la figure 2.21.

Chapitre 2

Nombre maximal d'arêtes par boîte = 32

Nombre maximal d'arêtes par boîte = 16

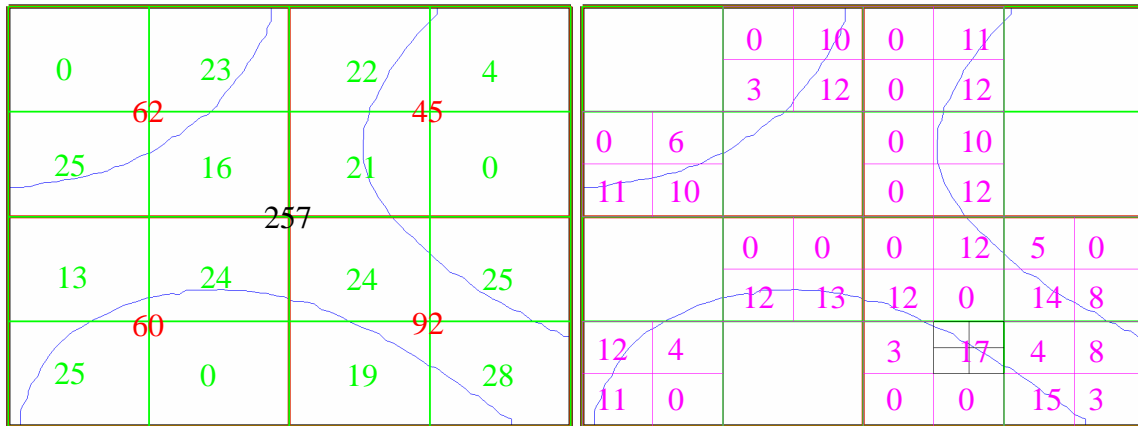


Figure 2.21. Représentation hiérarchique du maillage surfacique. Les chiffres indiqués correspondent au nombre de segments du maillage surfacique contenus dans chaque boîte.

Pour calculer la distance à un nœud, nous déterminons dans un premier temps la distance du nœud aux boîtes ayant la plus grande taille, ou plus exactement la distance de ce nœud à tous les coins de ces boîtes, ce qui est très simple. Nous obtenons ainsi pour chacune une distance minimale et une distance maximale. En comparant ces valeurs de distance, nous éliminons du calcul les boîtes qui ne peuvent pas contenir la facette la plus proche du nœud, c'est-à-dire celles pour lesquelles la distance minimale est plus grande que la plus petite des distances maximales. Nous poursuivons ces comparaisons pour les boîtes de niveau inférieur, jusqu'à ce que plus aucune ne puisse être éliminée. Les distances entre les arêtes contenues dans les boîtes restantes et le nœud sont alors calculées par la méthode de projection décrite précédemment et la plus petite valeur est retenue.

La fonction level set (distance signée) obtenue à partir de ce maillage surfacique de 257 segments sur un maillage volumique de 9901 nœuds est montrée sur la figure 2.22. Le temps nécessaire pour calculer cette fonction est de 0,26 s avec 32 arêtes par boîte, 0,25 s avec 16 arêtes et 1,01 s avec la méthode linéaire. Si l'accélération obtenue dans ce cas est déjà appréciable, elle peut devenir considérable pour des cas 3D et un maillage surfacique comprenant un grand nombre de facettes.

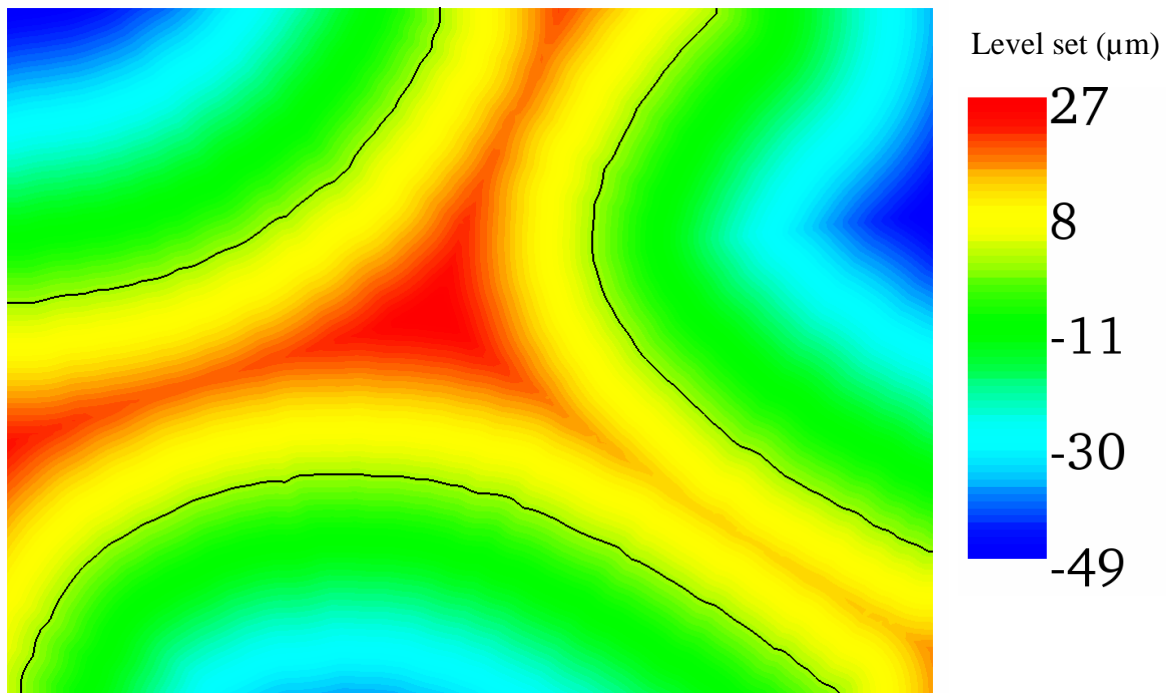


Figure 2.22. Fonction level set obtenue à partir du maillage surfacique de la figure 2.21. La partie négative de la fonction représente le domaine solide, la partie positive le domaine liquide et le trait noir l'interface.

5.1.3. Algorithme parallèle

Lorsque nous souhaitons travailler avec un maillage volumique comportant un très grand nombre de degrés de liberté, nous pouvons avoir recours à un ordinateur parallèle. Dans la version parallèle, le maillage volumique est distribué entre plusieurs partitions réparties sur autant de processeurs. Chaque partition est traitée comme s'il s'agissait d'un maillage indépendant. Cela nécessite de charger le maillage surfacique intégralement sur chaque processeur, ce qui signifie que l'algorithme est limité par la mémoire vive d'un seul processeur et non de l'ensemble des processeurs. Ainsi, pour des maillages surfaciques comportant un très grand nombre de facettes, le calcul devient impossible à réaliser, même si nous augmentons le nombre de processeurs.

Nous avons échappé à ce problème dans les applications de cette thèse mais nous tenons toutefois à proposer plusieurs solutions qui pourront permettre d'utiliser nos algorithmes sur des images de plus grande dimension. La première consiste à réduire le nombre de facettes du maillage surfacique [Gar 99, Hec 99]. Un deuxième moyen, très simple à considérer mais qui augmente le temps de calcul, est de transformer le maillage surfacique en plusieurs maillages plus petits. Il suffit ainsi de garder en mémoire la level set calculée pour le premier maillage, de la comparer à la level set calculée pour le deuxième maillage en choisissant la plus petite valeur absolue et ainsi de suite. Il est aussi possible de ne calculer la distance signée que dans une certaine épaisseur autour de l'interface, là où la fonction level set présentera un intérêt pour la suite du calcul (voir la méthode de lissage de la level set présentée dans le quatrième chapitre). Cela permettrait de ne charger qu'une partie du maillage surfacique pour chaque partition.

Une solution plus élégante serait sans conteste une solution permettant de calculer la distance signée pour l'ensemble des nœuds, tout en ne chargeant pour chaque processeur que la partie

utile du maillage surfacique. Toutefois, proposer un algorithme efficace permettant de traiter l'ensemble des cas n'est pas trivial.

Si la mémoire peut poser problème, le temps de calcul, en revanche, se réduit lorsque le nombre de processeurs augmente. Nous ne donnons pas d'analyse précise car la détermination de la level set à partir du maillage surfacique n'intervient qu'une seule fois par simulation. De plus, le temps de calcul nécessaire pour cette étape est négligeable en comparaison du temps de calcul des autres étapes (calcul du maillage volumique, calcul mécanique). A titre d'exemple, le temps pour déterminer la distance avec un maillage volumique de 3 000 000 de nœuds, un maillage surfacique de plus d'1 000 000 de facettes et un nombre maximal de 512 facettes par boîte est de l'ordre d'1 min avec 32 processeurs (62 Amd Opteron 6134 2,3 Ghz) du cluster CEMEF.

5.1.4. Remarque

Nous utilisons en principe un maillage volumique possédant les mêmes dimensions physiques que l'image utilisée pour calculer le maillage surfacique. Le calcul de la distance peut cependant poser un léger problème sur certains nœuds du bord. En effet, nous pouvons voir sur la figure 2.7 que le maillage surfacique n'est pas défini sur une épaisseur d'une moitié de pixel au bord de l'image. Pour résoudre ce problème, il nous suffit de prolonger l'image d'un pixel (ou voxel) sur tous ses bords ou d'immerger le maillage surfacique dans un maillage volumique légèrement plus petit que l'image de départ.

5.2. Construction du maillage volumique

La construction du maillage volumique dans CIMLib s'appuie sur une librairie indépendante, appelée mtc et développée par Coupez et al. [Cou 00, Gru 04, Gru 05] et par Digonnet et al. pour le parallélisme [Dig 01, Dig 07, Mes 08]. N'étant pas intéressé par les aspects techniques de la génération du maillage, nous renvoyons le lecteur concerné à ces références et nous contentons de donner dans cette partie des éléments d'explication importants pour l'utilisateur de la librairie mtc et de CIMLib.

5.2.1. Maillage initial

Le principe de génération d'un maillage volumique est facile à comprendre du point de vue de l'utilisateur. Nous introduisons une fonction appelée taille de maille et associée aux nœuds du maillage. Lorsque cette fonction est identique pour tous les nœuds, nous parlons de taille de maille homogène. Il s'agit sinon d'une taille de maille hétérogène. Si la taille est identique suivant toutes les directions, le maillage est isotrope et la taille de maille est définie comme un scalaire. Au contraire, si nous souhaitons travailler avec des tailles différentes dans chaque direction, la taille de maille devient une fonction tensorielle et le maillage est alors anisotrope. L'introduction de cette fonction taille de maille, qu'il ne faut pas confondre avec la métrique calculée en interne par le mailleur, est destinée à simplifier l'utilisation de mtc. Nous notons cette fonction h pour un maillage isotrope et \bar{h} pour un maillage anisotrope. h_1 , h_2 et h_3 représentent alors les tailles de mailles souhaitées dans les directions de l'espace. Nous illustrons à présent les différents types de taille de maille sur des exemples simples en 2D.

Commençons avec un maillage isotrope et homogène représentant le même domaine de calcul que celui de la figure 2.22, à savoir un rectangle de 156 μm de longueur et de 116 μm de hauteur. Pour cela, nous partons du maillage le plus grossier possible, comprenant 2 éléments,

montré sur la figure 2.23. Pour obtenir le maillage voulu, l'utilisateur doit seulement renseigner une valeur de taille de maille. En prenant une taille de 5 μm , nous obtenons le maillage montré à droite de la figure 2.23.

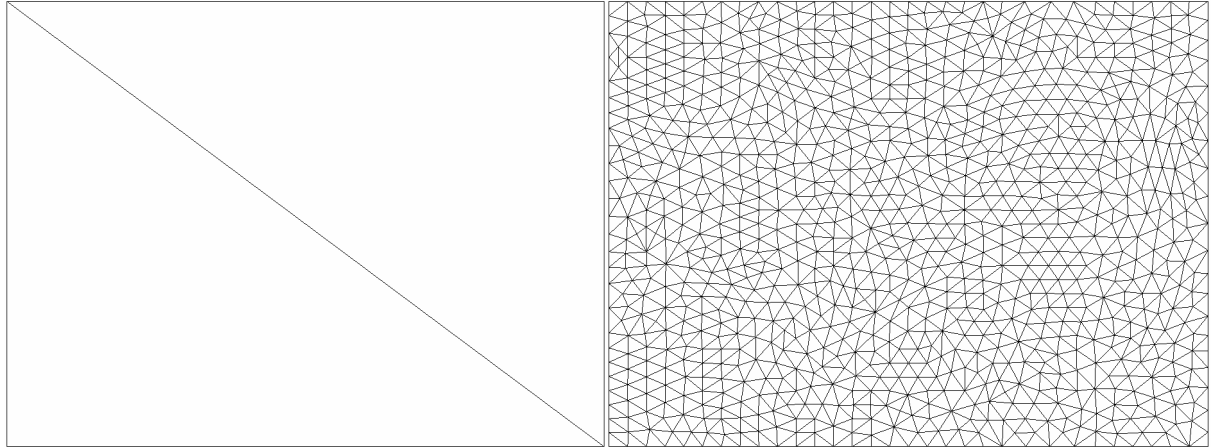


Figure 2.23. Obtention d'un maillage isotrope et homogène à partir d'un maillage de 2 éléments.

A partir d'un maillage de ce type, nous pouvons calculer une taille de maille hétérogène qui dépend de la distance à l'interface. La figure 2.24 illustre le résultat obtenu en utilisant la taille de maille suivante :

$$h = 1 + \frac{|\phi|}{5}$$

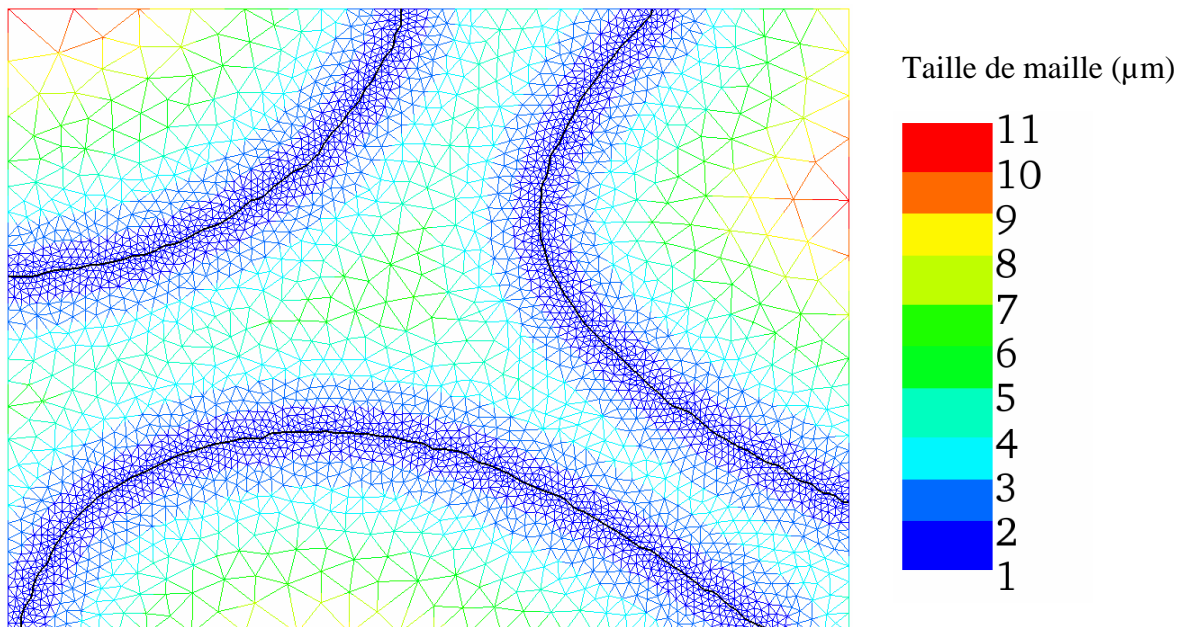


Figure 2.24. Exemple de taille de maille hétérogène. Le trait noir représente l'interface.

De cette façon, la taille de maille à l'interface est de 1 μm et elle croît de manière continue avec la distance à l'interface.

Pour obtenir un maillage anisotrope, nous utilisons une taille de maille matricielle, qui comprend 4 composantes en 2D et 9 en 3D. Les valeurs propres de cette matrice sont les inverses des tailles de maille dans les 2 (ou 3) directions, qui sont quant à elles données par les vecteurs propres de la matrice. Un moyen classique pour avoir un maillage anisotrope adapté à la représentation d'une interface est l'utilisation du gradient de la fonction level set [Ber 09]. Nous utilisons ainsi la taille de maille ci-dessous :

$$\bar{h} = \begin{bmatrix} \frac{1}{h_2} + \frac{\partial\phi}{\partial x} \frac{\partial\phi}{\partial x} \left(\frac{1}{h_1} - \frac{1}{h_2} \right) & \frac{\partial\phi}{\partial x} \frac{\partial\phi}{\partial y} \left(\frac{1}{h_1} - \frac{1}{h_2} \right) \\ \frac{\partial\phi}{\partial x} \frac{\partial\phi}{\partial y} \left(\frac{1}{h_1} - \frac{1}{h_2} \right) & \frac{1}{h_2} + \frac{\partial\phi}{\partial y} \frac{\partial\phi}{\partial y} \left(\frac{1}{h_1} - \frac{1}{h_2} \right) \end{bmatrix}$$

Les valeurs propres sont $1/h_1$ et $1/h_2$ et les vecteurs propres sont le gradient de la level set et le vecteur orthogonal au gradient de la level set. En principe, des éléments anisotropes sont seulement employés dans une certaine épaisseur autour de l'interface. Dans le reste du domaine, la taille de maille est isotrope et nous avons donc $h_1 = h_2$. En utilisant la même taille h_2 que celle de la figure 2.24 et une taille $h_1 = 0,2 \mu\text{m}$ dans une épaisseur de $2 \mu\text{m}$, nous obtenons le maillage montré sur la figure 2.25. L'anisotropie souhaitée des éléments est bien prise en compte par le mailleur mtc, comme illustré sur l'agrandissement fait autour de l'interface.

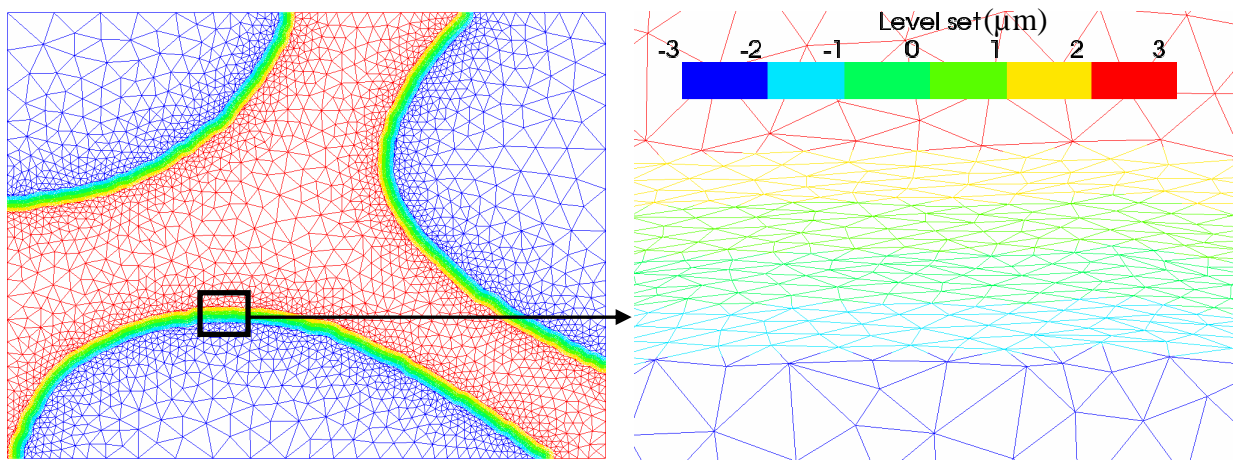


Figure 2.25. Exemple de taille de maille anisotrope.

Nous tenons à préciser que l'utilisateur doit se montrer prudent dès lors qu'il souhaite utiliser une taille de maille hétérogène. En effet, si la fonction taille de maille présente des oscillations importantes dans une zone où le nombre d'éléments du maillage initial est trop faible, il est possible que la taille de maille considérée par le mailleur ne puisse pas refléter celle demandée par l'utilisateur. Il est ainsi conseillé d'effectuer plusieurs itérations de remaillage afin de converger vers le maillage souhaité. Toutefois, cela ne suffit pas toujours si les maillages voulu et initial sont trop différents. Nous préconisons donc également d'utiliser un maillage de départ suffisamment proche du résultat attendu.

En modifiant les paramètres utilisés pour calculer la taille de maille associée à la figure 2.25, nous pouvons obtenir de nombreux maillages utilisables pour nos calculs futurs. Mais il est

difficile de contrôler l'erreur d'interpolation ou le nombre d'éléments du maillage de cette façon. C'est pourquoi il existe dans CIMLib des moyens plus sophistiqués de calculer la taille de maille. Hamide et al. proposent ainsi un estimateur d'erreur piloté par l'erreur d'interpolation [Ham 08a, Ham 08b], tandis que Mesri et al. ont mis au point une méthode de calcul de la taille de maille qui permet de choisir le nombre d'éléments du maillage [Mes 08, Zer 10]. Ces deux techniques s'appuient sur la détermination de la matrice hessienne, c'est-à-dire la matrice carrée des dérivées partielles secondes (ou encore le gradient du gradient), d'une fonction choisie pour remailler notre domaine de calcul. Si le but est de raffiner le maillage de manière anisotrope à l'interface, il est judicieux de choisir pour cette fonction une level set lissée [Zer 10].

Les deux approches permettent d'obtenir une valeur de la taille de maille par élément, que nous projetons aux nœuds et qui devient un paramètre d'entrée du mailleur mtc (le second paramètre étant le maillage actuel). Coupeux a développé récemment un estimateur d'erreur, basé sur le calcul d'un tenseur de distribution des longueurs d'arêtes, qui permet de calculer la taille de maille directement aux nœuds et qui semble donner des résultats encore plus performants [Cou 11]. Même si ces outils paraissent très attrayants, nous ne les utilisons pas ici, pour des raisons exposées dans le quatrième chapitre.

5.2.2. Remaillage

Nous tenons également à préciser que le mailleur mtc, dont nous venons de présenter certaines applications, peut très bien être utilisé pour remailler le domaine de calcul, durant une simulation. Une telle adaptation du maillage a le même fonctionnement que le calcul du maillage initial. Nous prenons pour maillage de départ le maillage de l'incrément précédent, calculons une taille de maille aux nœuds et nous servons du mailleur mtc pour obtenir le nouveau maillage. Il existe cependant une différence fondamentale entre le calcul du maillage initial et un remaillage au cours de la simulation. Pour le maillage initial, nous recalculons en effet la level set à partir du maillage surfacique, ce qui n'est pas possible pour un remaillage. L'ensemble des fonctions définies sur l'ancien maillage, parmi lesquelles la fonction level set, doivent ainsi être transportées sur le nouveau maillage, par interpolation linéaire. Un exemple de remaillage unidimensionnel est illustré sur la figure 2.26. Un champ de température fictif est montré avant et après adaptation du maillage, les différents points symbolisant les valeurs aux nœuds du maillage. Nous remarquons que l'intégrale de la température sur le domaine avant et après remaillage est différente, ce qui signifie que l'énergie n'est pas conservée. Il est par conséquent nécessaire de prendre le plus de précautions possibles quant au choix des tailles de maille souhaitées et de la fréquence des remaillages.

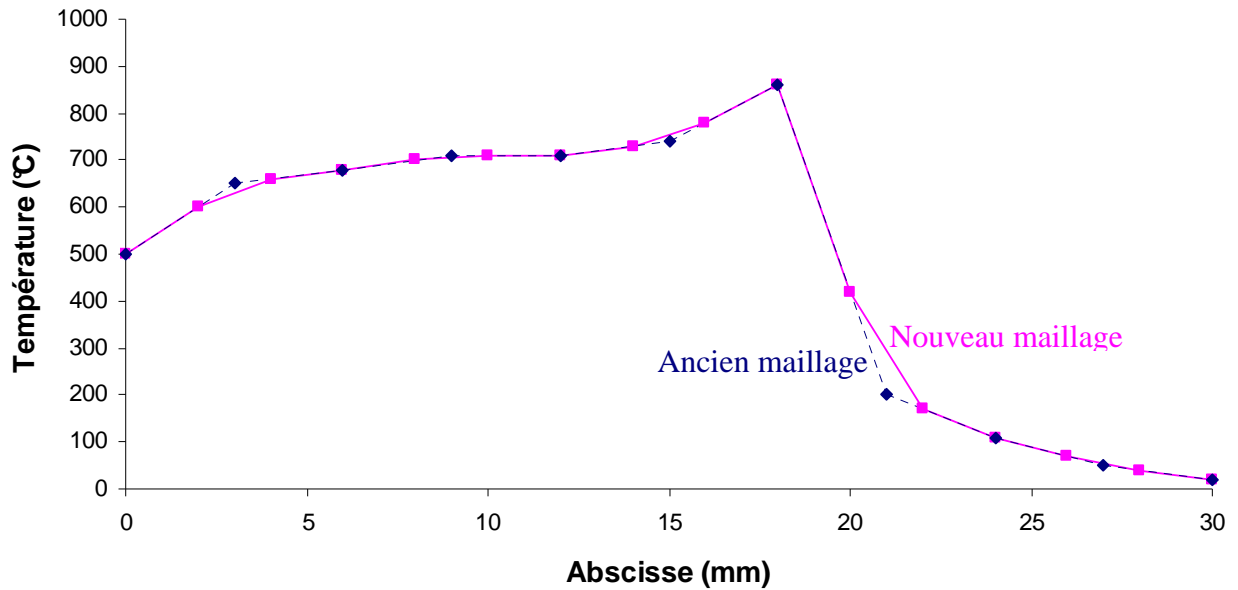


Figure 2.26. Illustration de la non conservation d'une grandeur durant un remaillage.

5.3. Traitements

Dans les travaux de Limodin et al. portant sur les changements microstructuraux au sein d'un alliage d'aluminium-cuivre durant sa refusion partielle [Lim 07], les données expérimentales sont utilisées pour déterminer l'évolution de la fraction volumique de solide ainsi que la surface spécifique de l'interface solide-liquide. Afin de pouvoir effectuer des comparaisons entre expériences et simulations, nous expliquons comment calculer ces grandeurs sur un maillage éléments finis.

5.3.1. Calcul de la surface spécifique

La surface spécifique s'écrit $S_v^{s/l} = \frac{S^{s/l}}{V}$ avec $S^{s/l}$ l'aire de l'interface entre les phases solide et liquide et V le volume total du matériau étudié. Nous nous concentrons ici sur le calcul de l'aire d'une interface. Cette interface est représentée par l'isovaleur 0 de la fonction level set. Il s'agit d'une isoligne en 2D et d'une isosurface en 3D. La fonction level set étant interpolée aux nœuds, son signe est différent sur les nœuds d'un élément traversé par l'isovaleur 0. Il faut alors calculer la longueur ou la surface (en 3D) de l'interface sur cet élément.

Prenons le cas bidimensionnel d'un élément représenté sur la figure 2.27 par le triangle ABC. A, B et C sont les trois nœuds de l'élément. Supposons alors que nous avons $\phi_A < 0$, $\phi_B > 0$ et $\phi_C > 0$ avec ϕ_A , ϕ_B et ϕ_C les valeurs de la fonction level set aux nœuds A, B et C. Nous pouvons affirmer que l'isovaleur zéro de la level set coupe le triangle ABC en un point D situé sur le segment [AB] et en un point E situé sur le segment [AC], comme illustré par la figure 2.27. En outre, la level set étant interpolée linéairement par élément, nous calculons la position de ces deux points en utilisant les formules suivantes :

Chapitre 2

$$x_D = x_A - \left(\frac{x_A - x_B}{\phi_A - \phi_B} \right) \phi_A \quad x_E = x_A - \left(\frac{x_A - x_C}{\phi_A - \phi_C} \right) \phi_A$$

$$y_D = y_A - \left(\frac{y_A - y_B}{\phi_A - \phi_B} \right) \phi_A \quad y_E = y_A - \left(\frac{y_A - y_C}{\phi_A - \phi_C} \right) \phi_A$$

avec A, B, C, D et E les indices désignant les différents points. x et y sont l'abscisse et l'ordonnée de ces points.

Nous n'avons alors plus qu'à calculer la longueur du segment [DE] puis à sommer la contribution de chaque élément pour déterminer le périmètre de notre contour.

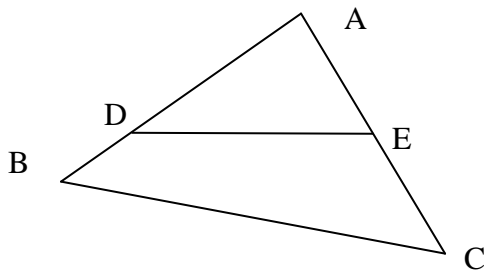


Figure 2.27. Illustration de la méthode permettant de déterminer la contribution d'un élément au calcul du périmètre d'une isoligne

Nous utilisons le même principe en 3D. Au lieu d'avoir des éléments triangulaires, nous avons des éléments tétraédriques à quatre nœuds. Nous devons donc calculer l'aire d'un triangle dans le cas où trois nœuds ont une level set de même signe et l'aire d'un quadrilatère si seulement deux nœuds ont une level set de même signe.

Pour valider notre implémentation, nous plaçons une sphère de rayon 40 cm au centre d'un domaine cubique de côté 1 m, calculons la level set associée sur les nœuds d'un maillage éléments finis et déterminons la surface de la sphère à partir de cette level set. Sur la figure 2.28, nous pouvons voir que la surface obtenue par le calcul se rapproche bien de la surface analytique lorsque la taille de maille diminue.

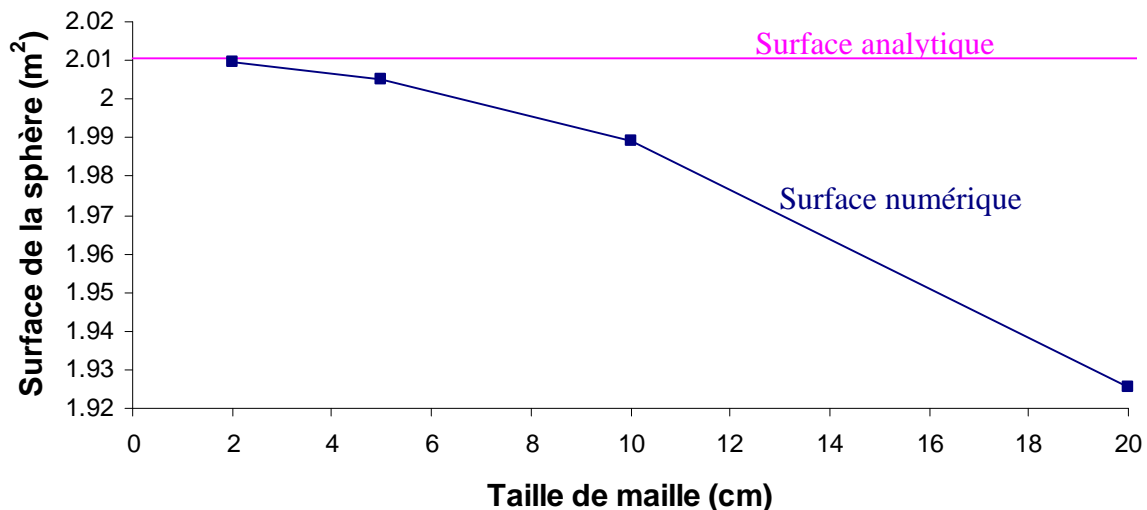


Figure 2.28. Evolution de la surface d'une sphère en fonction de la taille de maille considérée.

5.3.2. Calcul de la fraction volumique

La fraction volumique de solide s'écrit $g^s = \frac{V^s}{V}$ avec V^s le volume de solide. Pour calculer ce volume, nous utilisons une technique similaire à celle employée pour déterminer la surface de l'isovaleur 0. Simplement, pour reprendre l'exemple 2D de la figure 2.27, nous ne calculons plus la longueur du segment [DE] mais la surface du triangle ADE ou celle du quadrilatère BCED selon que nous sommes intéressé par le volume d'une phase ou de l'autre. Cela amène des configurations plus complexes en 3D. Nous en décrivons cinq qui permettent de calculer le volume associé à la partie négative de la fonction level set pour un tétraèdre ABCD. Pour calculer le volume associé à la partie positive, il suffit donc de changer le signe de la level set sur tous les nœuds du maillage. Ces cinq situations sont représentatives de l'ensemble des possibilités rencontrées :

- la level set associée aux quatre nœuds est positive. La contribution de l'élément est alors nulle ;
- la level set est négative pour tous les nœuds. L'apport de l'élément dans le calcul du volume est alors égal à son propre volume, dont nous obtenons par exemple la valeur en utilisant la formule de Héron :

$$V^T = \frac{1}{6} \left\{ \begin{array}{l} + (x_B - x_A)((y_C - y_A)(z_D - z_A) - (z_C - z_A)(y_D - y_A)) \\ - (x_C - x_A)((y_B - y_A)(z_D - z_A) - (z_B - z_A)(y_D - y_A)) \\ + (x_D - x_A)((y_B - y_A)(z_C - z_A) - (z_B - z_A)(y_C - y_A)) \end{array} \right\}$$

V^T est alors le volume du tétraèdre ABCD tandis que x, y et z sont l'abscisse, l'ordonnée et la cote des différents points ;

- la level set est négative en A et positive pour les trois autres nœuds. Nous calculons la position des points E, F et G situés sur les segments [AB], [AC] et [AD] pour lesquels la level set vaut zéro. La contribution de l'élément est alors le volume du tétraèdre AEFG ;
- la level set est positive en A et négative pour les trois autres nœuds. Comme dans le cas précédent nous calculons la position de E, F et G. L'apport de l'élément est alors la différence entre le volume du tétraèdre ABCD et celui du tétraèdre AEFG ;
- la level set est négative en A et en B et positive en C et D. Nous calculons la position des points E, F, G et H situés respectivement sur les segments [AC], [AD], [BC] et [BD], pour lesquels la level set vaut zéro. La contribution de l'élément est égale au volume de l'hexaèdre ABEFGH, qui peut être déterminé en ajoutant le volume des tétraèdres AEFG, AFGH et ABGH dans le cas où EFHG forme un quadrilatère.

Pour vérifier le code correspondant, nous calculons le volume de la sphère utilisée pour tracer la figure 2.28. Nous obtenons à nouveau une valeur numérique de plus en plus proche de la valeur analytique à mesure que la taille de maille diminue. Nous comparons également les résultats obtenus avec ceux donnés par une méthode s'appuyant sur le théorème de flux-divergence, qui permet de calculer le volume d'une surface fermée. La formule employée est la suivante :

$$\iiint_V \nabla \cdot \vec{F} \, dV = \oiint_{\Sigma} \vec{F} \cdot \vec{n} \, dS$$

avec V un volume délimité par son contour fermé Σ , \vec{F} un champ vectoriel et \vec{n} la normale unitaire à la surface dirigée vers l'extérieur.

Si nous prenons $\vec{F} = x \vec{i}$, avec x la première coordonnée spatiale et \vec{i} le premier vecteur de base, nous avons $\nabla \cdot \vec{F} = 1$ et nous obtenons par conséquent $V = \oiint_{\Sigma} x \vec{i} \cdot \vec{n} \, dS$.

Nous devons donc calculer la contribution sur chaque élément traversé par l'isosurface puis sommer l'ensemble des contributions pour obtenir le volume de l'objet. En 3D, sur le triangle ou le quadrilatère représentant l'intersection, l'abscisse est définie de manière linéaire. Les vecteurs \vec{i} et \vec{n} étant constants sur cette facette triangulaire ou quadrilatérale, l'intégrale de surface peut donc s'évaluer en un seul point central : le centre de gravité.

Pour calculer un vecteur normal à la surface nous effectuons le produit vectoriel entre deux arêtes du triangle ou du quadrilatère et nous divisons par sa norme pour obtenir la normale unitaire. Pour nous assurer que nous calculons bien la normale sortante, nous déterminons le signe du produit scalaire obtenu entre cette normale et un vecteur dirigé vers l'extérieur, par exemple le vecteur \vec{AE} , avec A un nœud situé à l'intérieur du volume et E un point situé sur la surface de l'objet.

Le théorème de flux-divergence donne les mêmes volumes que l'autre méthode pour la sphère placée à l'intérieur du domaine cubique. Cependant, le code que nous avons mis en commun dans la librairie CIMLib est celui qui correspond à la première des deux techniques présentées, même si elle semble beaucoup moins élégante. En effet, cette dernière présente un avantage très important sur la méthode du théorème de flux divergence, puisqu'elle permet de calculer le volume associé à une surface non fermée présente sur les bords du domaine.

5.3.3. Comparaisons

Les outils tout juste décrits permettent bien d'effectuer des comparaisons entre les résultats expérimentaux pouvant être obtenus par nos partenaires du laboratoire SIMaP et des résultats obtenus par simulation numérique. Ils peuvent également permettre d'avoir une idée sur la précision de la transformation des images en représentations éléments finis. Il nous faut pour cela être capable de calculer des fractions volumiques et des surfaces spécifiques à partir des fichiers de tomographies. Une possibilité pour la fraction volumique consiste à effectuer un simple décompte du nombre de voxels appartenant à la phase étudiée et de diviser le résultat par le nombre total de voxels du matériau étudié. En ce qui concerne la détermination de l'aire d'une interface, il est possible d'employer le concept d'intercepts linéaires décrit par Underwood [Und 70].

Nous tenons cependant à rappeler que les maillages surfaciques calculés par l'algorithme des marching cubes constituent une bonne représentation des interfaces entre les phases. Il est donc justifié d'utiliser ces maillages pour déterminer les fractions volumiques et les surfaces des interfaces. Les surfaces sont directement données par la sommation de l'aire de l'ensemble des facettes triangulaires. Si nous avons décidé de ne pas employer le théorème de flux-divergence pour les maillages volumiques, il semble être une bonne option pour déterminer le volume associé à un maillage surfacique. Afin d'employer ce théorème, il est nécessaire de fermer les surfaces qui ne le sont pas sur les bords. Pour cela, nous ajoutons une couche de voxels blancs autour des images considérées avant d'extraire les maillages surfaciques. Pour calculer les volumes de manière rigoureuse, un traitement spécifique doit alors être employé sur les bords.

Idéalement, nous aimerions pouvoir établir une correspondance entre la taille de maille utilisée pour le maillage volumique (plus exactement le rapport entre la taille de maille et la taille d'un voxel de l'image) et la précision de la transformation, c'est-à-dire les pourcentages d'écart entre les surfaces des interfaces et les volumes de chaque phase obtenus avec les maillages surfacique et volumique. Nous avons essayé de le faire pour plusieurs images et avons évidemment obtenu des représentations plus fiables en diminuant la taille de maille ou en utilisant des maillages beaucoup plus raffinés aux interfaces, obtenus avec les outils décrits par Mesri et al. [Mes 08]. Cependant, établir une telle correspondance n'est pas possible en toute généralité, car trop dépendant des morphologies rencontrées. Il est en effet plus facile d'obtenir un écart entre le volume donné par les maillages surfacique et volumique de 2 % pour les grains de taille importante contenus dans la figure 2.19 que pour des grains de taille plus modeste.

Nous choisissons ainsi de ne pas montrer de graphique représentant l'évolution du volume d'une phase en fonction de la taille de maille. Nous encourageons toutefois les futurs utilisateurs des outils présentés dans ce chapitre à effectuer leurs propres expérimentations afin de déterminer le maillage volumique qui correspond au mieux à leurs attentes. Nous leur conseillons également de prendre davantage en considération les possibilités offertes par les solveurs qu'ils ont l'intention d'utiliser que la précision de la représentation elle-même. Obtenir une représentation numérique parfaite d'une image expérimentale ne présente en effet aucun intérêt si le maillage associé ne permet pas d'effectuer une simulation. Milesi et al. présentent un bon exemple de ce qu'il faut parfois savoir se résoudre à faire. Conscient de la difficulté et de l'utilité limitée de représenter numériquement leurs images avec une très grande précision, ils préfèrent en supprimer les inclusions les plus petites [Mil 11].

6. Conclusion

Nous venons de présenter la méthode retenue pour établir des représentations éléments finis à partir d'images de microtomographie obtenues par nos partenaires grenoblois à l'installation européenne de rayonnement synchrotron. En filtrant les images, puis en les segmentant et en calculant des maillages surfaciques qui correspondent aux interfaces entre les différentes phases en présence, nous calculons des fonctions level set qui permettent d'initialiser un maillage volumique du matériau imagé, à partir duquel des simulations avec la librairie CIMLib peuvent être effectuées.

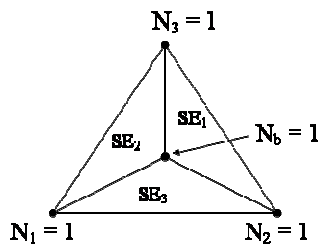
Avant de développer la technique présentée ici pour initialiser les représentations par éléments finis, nous avons employé une autre méthode décrite dans un précédent rapport [Zar 09] et consistant à calculer de proche en proche une distance à l'interface directement sur les voxels. Nous avons également utilisé le logiciel Morph-M, développé au Centre de Morphologie Mathématique de MINES ParisTech [Enf 07], afin de calculer plus précisément cette distance à l'interface sur les voxels. Les résultats étant moins satisfaisants que ceux obtenus par marching cubes et immersion de volume, nous avons choisi de ne pas détailler ces deux méthodes. Il existe également d'autres techniques pour extraire un maillage surfacique à partir d'une image, telle que celle des marching tetrahedra [Elv 92].

Les moyens décrits dans ce chapitre ne sont pas nécessairement optimaux. Mais même si un certain nombre d'améliorations peuvent être envisagées, les travaux effectués permettent de répondre au cahier des charges fixé au début du projet SIMUZAL. Nous satisfaisons notamment aux contraintes dictées par le mailleur mtc de CIMLib, qui nous impose l'utilisation de maillages non structurés composés de tétraèdres. En outre, les outils développés sont employés dans d'autres études du CEMEF [Huk 11].

Chapitre 2

Si nous n'avions pas dû utiliser des tétraèdres, il aurait pu être intéressant d'essayer des éléments différents pour la représentation numérique et particulièrement des éléments cubiques. Maire et al. décrivent notamment la méthode voxel/élément qui consiste à considérer chaque voxel de l'image de départ comme un élément du maillage de la représentation numérique [Mai 03]. Cette technique simple à implémenter nécessite un très grand nombre d'éléments. Pour pallier cet inconvénient, Maire propose de baisser la résolution de l'image, ce qui diminue le nombre d'éléments. Il est également possible de découper les cubes en tétraèdres afin de pouvoir utiliser CIMLib.

La méthode voxel/élément présente un autre défaut : un voxel appartenant soit à une phase soit à une autre, la représentation de l'interface est très abrupte. Pour corriger cela, tout en gardant la résolution de l'image inchangée, nous pourrions faire appel à une représentation level set de l'interface en calculant une distance au maillage surfacique sur un maillage volumique de type octree [Los 04], pour lequel les éléments cubiques sont beaucoup plus gros loin des zones d'intérêt. En faisant coïncider les éléments de ce maillage avec les voxels de l'image au niveau de l'interface, l'isovaleur zéro de la fonction level set serait plus proche de la surface définie par le maillage surfacique qu'en utilisant des tétraèdres non structurés.



- Chapitre 3 -

Solveur mécanique

« If you understood what I said, I must have misspoken. »

Alan Greenspan

1. Introduction	48
2. Formulation	48
2.1. Formulation forte	48
2.2. Formulation faible	49
2.3. Formulation faible discrétisée	50
2.4. L'élément P1+/P1	50
2.5. Calcul des matrices locales	52
2.5.1. Terme en vitesse-vitesse	52
2.5.2. Terme en vitesse-pressure (et pressure-vitesse)	53
2.5.3. Terme en pressure-pressure	53
2.5.4. Calcul des intégrales	54
2.6. Assemblage	56
2.7. Rajout des conditions aux limites	57
2.8. Calcul de la viscosité	57
2.9. Stockage des matrices	58
3. Résolution des systèmes linéaires	59
3.1. Méthode directe	59
3.2. Méthode itérative	60
3.3. Applications simples	62
3.3.1. Problème mal posé	62
3.3.2. Dimension des résidus	63
3.3.3. Solveur direct ou itératif	67
4. Test de l'inclusion	69
4.1. Maillage utilisé	69
4.2. Solution numérique	70
4.3. Rapport des viscosités et précision	72
4.4. Utilisation d'un maillage plus adapté	73
5. Comportement viscoplastique	73
6. Conclusion	77

1. Introduction

La méthode de résolution que nous décrivons dans ce chapitre a été précédemment décrite dans un grand nombre de travaux menés au CEMEF, dans différents contextes. Cette méthode s'appuie sur une résolution éléments finis en vitesse-pression avec un champ de vitesse linéaire enrichi par l'apport d'une fonction bulle au centre de l'élément et un champ de pression linéaire. Il s'agit de la désormais classique formulation P1+/P1 très largement inspirée par l'approche MINI-élément introduite par Arnold, Brezzi & Fortin [Arn 84].

Cette formulation P1+/P1 est maintenant utilisée depuis assez longtemps au CEMEF [Cou 97, Per 00, Pic 97]. Dans CIMLib, le solveur de Stokes avec formulation P1+/P1 a été implémenté en 2004 par Basset [Bas 06]. Le codage adopté profite pleinement de la programmation orientée objets et a permis le développement de solveurs plus complexes s'appuyant sur cette base commune. Ainsi, un solveur de Navier Stokes incompressible [Bas 06], un solveur de Navier Stokes compressible avec extra-contrainte [Pha 12] et un solveur général mixte [Bru 07] s'appuient sur ce solveur de Stokes.

Les équations de Navier Stokes ont fait l'objet de très nombreux travaux dans la littérature et les méthodes pour les résoudre sont multiples. Nous avons choisi de ne pas mener une étude bibliographique approfondie sur ce domaine, car la méthode utilisée au CEMEF depuis de nombreuses années déjà semble donner satisfaction. Nous nous contentons par conséquent de conseiller au lecteur désireux de se familiariser avec la résolution des équations de Navier Stokes l'analyse des cours de Burkard [Bur 05] et de Scheid [Sch 08].

Dans ce chapitre, nous détaillons dans un premier temps la formulation utilisée pour résoudre les équations de Stokes incompressibles et sans second membre. De cette formulation découle un système linéaire pour lequel nous décrivons deux techniques classiques de résolution : une méthode directe et une méthode itérative. Nous illustrons ensuite la précision de cette méthode sur un test de traction uniaxiale d'une inclusion circulaire en comparant nos résultats numériques à la solution analytique décrite dans la littérature. Une fois le solveur de Stokes validé, nous expliquons comment prendre en compte un comportement viscoplastique. Enfin, nous concluons sur le solveur mécanique utilisé et donnons des pistes à explorer pour l'améliorer.

2. Formulation

2.1. Formulation forte

Nous rappelons les équations de conservation de la quantité de mouvement et de la masse pour un écoulement incompressible, des termes d'inertie négligés et sans force de masse :

$$\begin{cases} -\nabla \cdot \bar{\sigma} = \vec{0} \\ \nabla \cdot \bar{v} = 0 \end{cases}$$

\bar{v} représente le vecteur vitesse de l'écoulement et $\bar{\sigma}$ le tenseur des contraintes de Cauchy.

Afin de pouvoir résoudre ces équations, nous devons y adjoindre une loi de comportement. Dans cette partie, nous considérons une loi newtonienne : $\bar{\sigma} = 2\eta \bar{\epsilon}(\bar{v}) - p\bar{I}$ avec p la pression, \bar{I} le tenseur identité, $\bar{\epsilon}(\bar{v})$ le tenseur des vitesses de déformation tel que $\bar{\epsilon}(\bar{v}) = \frac{1}{2}(\bar{\nabla}\bar{v} + \bar{\nabla}\bar{v}^T)$ et η la viscosité dynamique supposée constante.

En remplaçant le tenseur des contraintes par son expression, nous obtenons :

$$\begin{cases} -\nabla \cdot (2\eta \bar{\bar{\epsilon}}(\bar{v})) + \bar{\nabla} p = 0 \\ \nabla \cdot \bar{v} = 0 \end{cases}$$

Ces deux équations constituent la formulation forte des équations de Stokes. Cette formulation est ainsi appelée car pour en être solution, une paire de fonctions \bar{v} et p doit respecter ces deux équations en tout point du domaine considéré. Cette paire de fonctions est alors appelée solution forte des équations de Stokes.

Les équations de Stokes incompressibles sont bien souvent écrites de la manière suivante dans la littérature :

$$\begin{cases} -\eta \Delta(\bar{v}) + \bar{\nabla} p = 0 \\ \nabla \cdot \bar{v} = 0 \end{cases}$$

En effet, $\nabla \cdot \bar{v} = 0$ implique que $\nabla \cdot (\bar{\bar{\epsilon}}(v)) = \frac{1}{2} \Delta(\bar{v})$ quand la viscosité est uniforme.

Nous avons cependant choisi d'écrire l'équation de conservation de la quantité de mouvement sous sa forme compressible dans un souci de généralité. En outre, le solveur a été codé suivant la formulation en vitesse de déformation et non suivant la formulation en laplacien afin que puissent en découler des solveurs compressibles.

2.2. Formulation faible

Si une paire de fonction \bar{v} et p est solution forte des équations de Stokes, elle en est toujours solution forte si les équations de Stokes sont respectivement multipliées par des fonctions test \bar{w} et q . De plus, comme ces équations sont respectées pour tous les points du domaine considéré, nous pouvons les intégrer sur l'ensemble de ce domaine pour obtenir [Bur 05] :

$$\begin{cases} \int_{\Omega} (-\nabla \cdot (2\eta \bar{\bar{\epsilon}}(\bar{v})) + \bar{\nabla} p) \cdot \bar{w} \, dV = 0 \\ \int_{\Omega} (\nabla \cdot \bar{v}) q \, dV = 0 \end{cases}$$

avec Ω le domaine d'étude sur lequel nous voulons résoudre les équations.

En utilisant une fonction \bar{w} qui s'annule sur la frontière du domaine Ω , en réalisant une intégration par parties pour les deux termes de l'équation de conservation de la quantité de mouvement et avec des termes de bord nuls (conditions de Dirichlet sur la vitesse), nous avons :

$$\left\{ \begin{array}{l} \int_{\Omega} (2\eta \bar{\bar{\epsilon}}(\bar{\bar{v}}) : \bar{\bar{\epsilon}}(\bar{\bar{w}}) - p \nabla \cdot \bar{\bar{w}}) dV = 0 \\ \int_{\Omega} (-q \nabla \cdot \bar{\bar{v}}) dV = 0 \end{array} \right.$$

Le symbole $:$ représente le produit doublement contracté.

Ces deux équations constituent la formulation faible des équations de Stokes. Une paire de fonctions $\bar{\bar{v}}$ et p est solution faible des équations de Stokes si elle satisfait ces deux équations pour toutes les fonctions test $\bar{\bar{w}}$ et q , avec les fonctions $\bar{\bar{w}}$ qui s'annulent au bord du domaine [Bur 05].

2.3. Formulation faible discrétisée

Pour appliquer la méthode des éléments finis, nous devons discrétiser notre domaine de calcul. Cette discrétisation s'appuie sur l'utilisation du maillage décrit dans le chapitre précédent et elle consiste à décomposer le domaine en une suite d'éléments triangulaires en deux dimensions et tétraédriques en trois dimensions (appelés d-simplexes par Gruau [Gru 04]). A l'intérieur de chaque élément la valeur des champs est calculée par une interpolation linéaire qui dépend de la valeur des champs aux nœuds (ou sommets) de l'élément et de leur position.

De cette façon nous pouvons définir le champ de pression en tout point de l'espace comme la somme de chaque valeur nodale multipliée par un coefficient de valeur unitaire si le point se situe exactement sur ce nœud du maillage, de valeur nulle si le point est en dehors d'un élément comprenant ce nœud et de valeur intermédiaire sinon. Ces coefficients, dont le nombre est égal au nombre de nœuds du domaine, sont appelés fonctions de base. Nous considérons que les fonctions test introduites précédemment sont également des fonctions de base.

La formulation faible discrétisée des équations de Stokes s'écrit de la manière suivante :

$$\left\{ \begin{array}{l} \int_{\Omega} (2\eta \bar{\bar{\epsilon}}(\bar{\bar{v}}_h) : \bar{\bar{\epsilon}}(\bar{\bar{w}}_h) - p_h \nabla \cdot \bar{\bar{w}}_h) dV = 0 \\ \int_{\Omega} (-q_h \nabla \cdot \bar{\bar{v}}_h) dV = 0 \end{array} \right.$$

en utilisant la lettre h pour représenter les grandeurs discrétisées.

En considérant qu'il y a N fonctions de base et que le champ de vitesse est un vecteur, nous décomposons la formulation faible discrétisée en un ensemble de $(d + 1) N$ équations, où d est la dimension du domaine et N le nombre de nœuds. Le nombre d'équations obtenues est égal au nombre d'inconnues à déterminer.

2.4. L'élément P1+/P1

L'utilisation d'une interpolation linéaire en vitesse et en pression (P1/P1) semble évidente, mais elle ne respecte pas la condition de stabilité de Brezzi-Babuška [Bab 73, Bre 74]. En revanche, la condition de stabilité est respectée en rajoutant un degré de liberté au centre de

Chapitre 3

chaque élément pour le champ de vitesse [Arn 84]. Nous parlons alors d'élément P1+/P1. La vitesse est décomposée en une partie linéaire \vec{v} et une partie bulle \vec{b} :

$$\vec{\tilde{v}}_h = \vec{v}_h + \vec{b}_h$$

Pour la création des fonctions bulle, chaque élément est subdivisé en $d + 1$ sous-éléments en reliant le centre de l'élément à chacun de ses nœuds. Ainsi, à l'image des fonctions de base linéaires, la fonction bulle a une valeur unitaire au centre de l'élément et nulle sur le contour de l'élément.

La formulation faible discrétisée des équations de Stokes s'écrit alors :

$$\begin{cases} \int_{\Omega} (2\eta \bar{\bar{\epsilon}}(\vec{\tilde{v}}_h) : \bar{\bar{\epsilon}}(\vec{\tilde{w}}_h) - p_h \nabla \cdot \vec{\tilde{w}}_h) dV = 0 \\ \int_{\Omega} (-q_h \nabla \cdot \vec{\tilde{v}}_h) dV = 0 \end{cases}$$

En utilisant la décomposition de la vitesse [Bas 06] et la propriété d'orthogonalité de la bulle [Per 00], nous obtenons le système suivant :

$$\begin{cases} \int_{\Omega} (2\eta \bar{\bar{\epsilon}}(\vec{v}_h) : \bar{\bar{\epsilon}}(\vec{w}_h) - p_h \nabla \cdot \vec{w}_h) dV = 0 \\ \int_{\Omega} (2\eta_b \bar{\bar{\epsilon}}(\vec{b}_h) : \bar{\bar{\epsilon}}(\vec{c}_h) - p_h \nabla \cdot \vec{c}_h) dV = 0 \\ \int_{\Omega} (-q_h \nabla \cdot \vec{v}_h - q_h \nabla \cdot \vec{b}_h) dV = 0 \end{cases}$$

\vec{c}_h représente la fonction test associée à la bulle et η_b la viscosité de la bulle, égale à la viscosité au centre de l'élément multipliée par un coefficient de stabilisation dont la valeur par défaut est 1/4 en 2D et 1/36 en 3D. Cette valeur par défaut est celle présente dans le code et il n'y a pas à notre connaissance de référence pour la justifier.

Par un procédé de condensation statique qui consiste à obtenir l'expression de \vec{b}_h à partir de la deuxième ligne et à remplacer \vec{b}_h par cette expression dans la troisième ligne [Bas 06, Hac 09], nous incluons les degrés de liberté de la bulle dans l'équation de conservation de la masse. Le système à résoudre est alors :

$$\begin{cases} \int_{\Omega} (2\eta \bar{\bar{\epsilon}}(\vec{v}_h) : \bar{\bar{\epsilon}}(\vec{w}_h) - p_h \nabla \cdot \vec{w}_h) dV = 0 \\ \int_{\Omega} -q_h \nabla \cdot \vec{v}_h dV + \int_{\Omega} -q_h \nabla \cdot \vec{b}_h dV \left(\int_{\Omega} 2\eta_b \bar{\bar{\epsilon}}(\vec{b}_h) : \bar{\bar{\epsilon}}(\vec{c}_h) dV \right)^{-1} \int_{\Omega} p_h \nabla \cdot \vec{c}_h dV = 0 \end{cases}$$

Le terme contenant le produit doublement contracté et la viscosité de la bulle dans la deuxième ligne représente une matrice carrée d'ordre d , qu'il nous faut inverser (c'est la signification du -1 en exposant).

2.5. Calcul des matrices locales

Pour appliquer la méthode des éléments finis, nous devons déterminer les valeurs numériques des intégrales du système précédent sur chaque élément. Ces valeurs constituent les matrices locales du système à résoudre.

2.5.1. Terme en vitesse-vitesse

Nous voulons d'abord calculer la matrice locale pour le terme en vitesse-vitesse :

$$\int_T (2\eta \bar{\bar{\epsilon}}(\bar{\mathbf{v}}_h) : \bar{\bar{\epsilon}}(\bar{\mathbf{w}}_h)) dV$$

T représente un élément du maillage éléments finis.

Cette intégrale ressemble à celle rencontrée dans le cas classique du laplacien [Dem 08] pour lequel la matrice locale comporte 9 composantes en 2D et 16 en 3D. Toutefois, nous ne devons pas oublier que les fonctions de base sont utilisées ici pour une équation vectorielle et non pour une équation scalaire comme dans le cas du laplacien. Ainsi, puisque le champ de vitesse admet 2 composantes en 2D et 3 en 3D, la matrice locale associée à ce terme comporte 36 composantes en 2D et 144 en 3D. Cette matrice locale est symétrique.

Pour calculer les valeurs des éléments de cette matrice, nous devons effectuer deux boucles sur les fonctions de base (une pour la fonction de base associée au champ de vitesse et l'autre pour la fonction de base associée à la fonction test). Pour chacune de ces boucles, une boucle sur les composantes est également nécessaire.

Nous illustrons à présent le calcul dans un cas en 2D. Pour le premier nœud de l'élément, $\bar{\mathbf{v}}_h$ vaut $\begin{pmatrix} N_1 \\ 0 \end{pmatrix}$ pour la composante Ox et $\begin{pmatrix} 0 \\ N_1 \end{pmatrix}$ pour la composante Oy où N_i est la fonction de base associée au nœud i de l'élément considéré.

Ainsi, pour le premier des 36 termes de la matrice locale :

$$\bar{\bar{\epsilon}}(\bar{\mathbf{v}}_h) : \bar{\bar{\epsilon}}(\bar{\mathbf{w}}_h) = \bar{\bar{\epsilon}}\begin{pmatrix} N_1 \\ 0 \end{pmatrix} : \bar{\bar{\epsilon}}\begin{pmatrix} N_1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{\partial N_1}{\partial x} & \frac{1}{2} \frac{\partial N_1}{\partial y} \\ \frac{1}{2} \frac{\partial N_1}{\partial y} & 0 \end{pmatrix} : \begin{pmatrix} \frac{\partial N_1}{\partial x} & \frac{1}{2} \frac{\partial N_1}{\partial y} \\ \frac{1}{2} \frac{\partial N_1}{\partial y} & 0 \end{pmatrix} = \left(\frac{\partial N_1}{\partial x}\right)^2 + \frac{1}{2} \left(\frac{\partial N_1}{\partial y}\right)^2$$

Et pour le deuxième terme :

$$\bar{\bar{\epsilon}}(\bar{\mathbf{v}}_h) : \bar{\bar{\epsilon}}(\bar{\mathbf{w}}_h) = \bar{\bar{\epsilon}}\begin{pmatrix} N_1 \\ 0 \end{pmatrix} : \bar{\bar{\epsilon}}\begin{pmatrix} 0 \\ N_1 \end{pmatrix} = \begin{pmatrix} \frac{\partial N_1}{\partial x} & \frac{1}{2} \frac{\partial N_1}{\partial y} \\ \frac{1}{2} \frac{\partial N_1}{\partial y} & 0 \end{pmatrix} : \begin{pmatrix} 0 & \frac{1}{2} \frac{\partial N_1}{\partial x} \\ \frac{1}{2} \frac{\partial N_1}{\partial x} & \frac{\partial N_1}{\partial y} \end{pmatrix} = \frac{1}{2} \left(\frac{\partial N_1}{\partial x}\right) \left(\frac{\partial N_1}{\partial y}\right)$$

De manière générale, nous avons :

$$A_{nkml}^{vv} = \int_T \eta \left[\frac{\partial N_m}{\partial x_j} \frac{\partial N_n}{\partial x_j} \delta_{kl} + \frac{\partial N_m}{\partial x_k} \frac{\partial N_n}{\partial x_j} \delta_{jl} \right] dV$$

avec A^{vv} la matrice locale pour l'élément T, n et m le numéro des fonctions test, k et l les différentes composantes, δ_{ij} le symbole de Kronecker égal à 1 lorsque $i = j$ et nul dans le cas contraire et x_j la $j^{\text{ème}}$ coordonnée.

Pour cette expression, nous avons adopté la convention de l'indice muet.

2.5.2. Terme en vitesse-pression (et pression-vitesse)

Les termes en vitesse-pression sont les suivants :

$$\int_T (-p_h \nabla \cdot \bar{w}_h) dV \quad \text{et} \quad \int_T (-q_h \nabla \cdot \bar{v}_h) dV$$

Pour ces termes, la matrice locale est :

$$A_{nm1}^{vp} = \int_T \left(-N_n \frac{\partial N_m}{\partial x_1} \right) dV$$

2.5.3. Terme en pression-pression

Le terme en pression-pression est donné ci-dessous :

$$- \int_{\Omega} q_h \nabla \cdot \bar{b}_h dV \left(\int_{\Omega} 2\eta_b \bar{\bar{\epsilon}}(\bar{b}_h) : \bar{\bar{\epsilon}}(\bar{c}_h) dV \right)^{-1} \int_{\Omega} p_h \nabla \cdot \bar{c}_h dV$$

Nous considérons dans un premier temps le terme en vitesse bulle-vitesse bulle :

$$\int_{\Omega} 2\eta_b \bar{\bar{\epsilon}}(\bar{b}_h) : \bar{\bar{\epsilon}}(\bar{c}_h) dV$$

La matrice locale correspondante est donnée par :

$$A_{kl}^{bb} = \int_T \eta \left[\frac{\partial N_b}{\partial x_j} \frac{\partial N_b}{\partial x_j} \delta_{kl} + \frac{\partial N_b}{\partial x_k} \frac{\partial N_b}{\partial x_j} \delta_{jl} \right] dV$$

avec N_b la fonction de base associée à la bulle.

Cette matrice ressemble beaucoup à la matrice du terme en vitesse-vitesse, mais nous n'avons pas de boucle sur les fonctions de base. La matrice possède ainsi 4 composantes en 2D et 9 en 3D. Nous pouvons donc facilement l'inverser pour le calcul du terme en pression-pression.

Pour le terme en vitesse bulle-pression, qui correspond au deux autres termes, la matrice locale s'écrit :

$$A_{nk}^{bp} = \int_T \left(N_n \frac{\partial N_b}{\partial x_k} \right) dV$$

Pour calculer la matrice en pression-pression, nous effectuons le produit suivant :

$$A^{pp} = -A^{bp} (A^{bb})^{-1} (A^{bp})^T$$

où le T représente la transposition d'une matrice. Cette matrice en pression-pression comprend 9 composantes en 2D et 16 en 3D.

2.5.4. Calcul des intégrales

Pour calculer les valeurs numériques des matrices locales, nous devons déterminer la valeur des dérivées partielles de chaque fonction de base par rapport aux différentes coordonnées. Pour cela, nous rappelons que sur un élément, une fonction de base associée à un nœud vaut 1 sur ce nœud et 0 sur les autres nœuds de l'élément. Comme une fonction de base est linéaire, la résolution du système suivant nous permet de déterminer la valeur de la fonction de base associée au premier nœud en tout point d'un élément triangulaire :

$$\begin{cases} ax_1 + by_1 + c = 1 \\ ax_2 + by_2 + c = 0 \\ ax_3 + by_3 + c = 0 \end{cases}$$

avec x_i et y_i les coordonnées du $i^{\text{ème}}$ nœud de l'élément et a, b et c les valeurs à déterminer.

Une fois ce système résolu, il est facile d'obtenir le gradient des fonctions de base [Dem 08] :

$$\bar{\nabla} N_1 = \frac{1}{D} \begin{bmatrix} y_2 - y_3 \\ x_3 - x_2 \end{bmatrix}; \quad \bar{\nabla} N_2 = \frac{1}{D} \begin{bmatrix} y_3 - y_1 \\ x_1 - x_3 \end{bmatrix}; \quad \bar{\nabla} N_3 = \frac{1}{D} \begin{bmatrix} y_1 - y_2 \\ x_2 - x_1 \end{bmatrix}$$

avec $D = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)$.

Les dérivées partielles des fonctions de base étant constantes sur l'élément, elles peuvent être sorties des intégrales. Pour intégrer un polynôme de premier degré, un seul point d'intégration est nécessaire au centre de l'élément. Ainsi :

Chapitre 3

$$\int_T N_i dV = \frac{1}{d+1} V$$

où V représente la surface de l'élément T pour $d = 2$ et son volume pour $d = 3$.

Pour calculer les intégrales dans lesquelles la bulle intervient, nous ajoutons les contributions calculées sur l'ensemble des sous-éléments, représentés sur la figure 3.1 en 2D. Il nous faut ainsi déterminer les valeurs des dérivées partielles de la fonction bulle par rapport à chaque coordonnée sur chaque sous-élément ainsi que la valeur des fonctions de base associées à chaque nœud de l'élément au centre de gravité de tous les sous-éléments.

Une méthode simple pour calculer ces valeurs consiste à résoudre les systèmes à $d + 1$ équations et $d + 1$ inconnues appropriés. Pour la fonction bulle sur le sous-élément SE_1 , il s'agit du système suivant :

$$\begin{cases} a \frac{x_1 + x_2 + x_3}{3} + b \frac{y_1 + y_2 + y_3}{3} + c = 1 \\ ax_2 + by_2 + c = 0 \\ ax_3 + by_3 + c = 0 \end{cases}$$

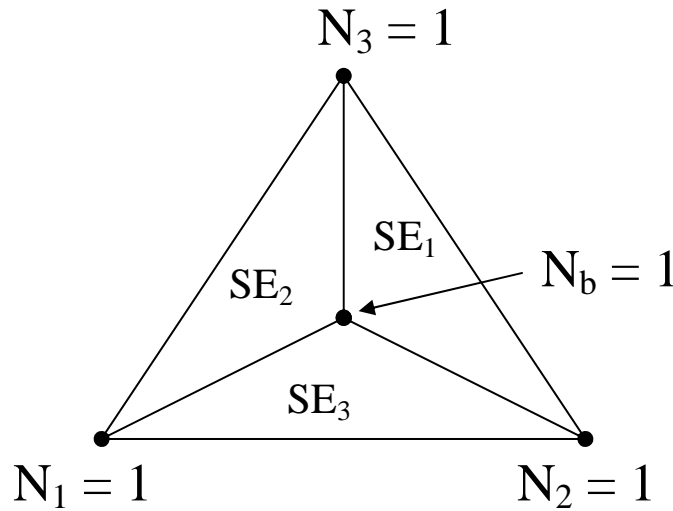


Figure 3.1. Représentation des sous-éléments (SE) pour un élément triangulaire.

Pour le calcul de N_i au centre du sous-élément 1, nous prenons la moyenne des valeurs de N_i sur les trois sommets du sous élément 1. De manière générale, nous avons :

$$\bar{\nabla} N_b = (d+1) \bar{\nabla} N_i \text{ sur } SE_i ; \bar{N}_i = \frac{1}{(d+1)^2} \text{ sur } SE_i ; \bar{N}_i = \frac{d+2}{(d+1)^2} \text{ sur } SE_j$$

avec \bar{N}_i la valeur d'une fonction de base au centre du sous élément et j différent de i .

2.6. Assemblage

Jusqu'à présent, nous avons seulement calculé les intégrales sur chaque élément. Nous devons calculer ces intégrales sur l'ensemble du domaine de calcul. Pour cela, nous utilisons l'égalité suivante :

$$\int_{\Omega} N_i N_j dV = \sum_{k=1}^{NE} \int_{T_k} N_i N_j dV$$

avec NE le nombre d'éléments sur notre domaine de calcul et T_k le $k^{\text{ème}}$ élément.

L'intégrale sur l'élément est généralement nulle, sauf si N_i et N_j sont des fonctions de base associées à deux nœuds d'un même élément. En pratique, une fois la matrice locale à un élément calculée, nous ajoutons toutes les contributions dans une matrice globale à la place qui convient. La matrice globale est une matrice carrée de dimension $(d + 1) N$.

Considérons un élément en 2D. Les nœuds appartenant à cet élément ont pour numéros C_1, C_2 et C_3 , où C correspond à la connectivité. Dans la matrice globale, les contributions de cet élément apparaîtront donc dans les lignes et les colonnes correspondant à C_1, C_2 et C_3 .

Nous donnons ci-dessous la matrice locale à cet élément avec les contributions à rajouter sur les lignes correspondant à C_2 et les colonnes correspondant à C_3 encadrées en noir :

A_{1x1x}^{vv}	A_{1x1y}^{vv}	A_{1x2x}^{vv}	A_{1x2y}^{vv}	A_{1x3x}^{vv}	A_{1x3y}^{vv}	A_{11x}^{vp}	A_{21x}^{vp}	A_{31x}^{vp}
A_{1y1x}^{vv}	A_{1y1y}^{vv}	A_{1y2x}^{vv}	A_{1y2y}^{vv}	A_{1y3x}^{vv}	A_{1y3y}^{vv}	A_{11y}^{vp}	A_{21y}^{vp}	A_{31y}^{vp}
A_{2x1x}^{vv}	A_{2x1y}^{vv}	A_{2x2x}^{vv}	A_{2x2y}^{vv}	A_{2x3x}^{vv}	A_{2x3y}^{vv}	A_{12x}^{vp}	A_{22x}^{vp}	A_{32x}^{vp}
A_{2y1x}^{vv}	A_{2y1y}^{vv}	A_{2y2x}^{vv}	A_{2y2y}^{vv}	A_{2y3x}^{vv}	A_{2y3y}^{vv}	A_{12y}^{vp}	A_{22y}^{vp}	A_{32y}^{vp}
A_{3x1x}^{vv}	A_{3x1y}^{vv}	A_{3x2x}^{vv}	A_{3x2y}^{vv}	A_{3x3x}^{vv}	A_{3x3y}^{vv}	A_{13x}^{vp}	A_{23x}^{vp}	A_{33x}^{vp}
A_{3y1x}^{vv}	A_{3y1y}^{vv}	A_{3y2x}^{vv}	A_{3y2y}^{vv}	A_{3y3x}^{vv}	A_{3y3y}^{vv}	A_{13y}^{vp}	A_{23y}^{vp}	A_{33y}^{vp}
A_{11x}^{vp}	A_{11y}^{vp}	A_{12x}^{vp}	A_{12y}^{vp}	A_{13x}^{vp}	A_{13y}^{vp}	A_{11}^{pp}	A_{12}^{pp}	A_{13}^{pp}
A_{21x}^{vp}	A_{21y}^{vp}	A_{22x}^{vp}	A_{22y}^{vp}	A_{23x}^{vp}	A_{23y}^{vp}	A_{21}^{pp}	A_{22}^{pp}	A_{23}^{pp}
A_{31x}^{vp}	A_{31y}^{vp}	A_{32x}^{vp}	A_{32y}^{vp}	A_{33x}^{vp}	A_{33y}^{vp}	A_{31}^{pp}	A_{32}^{pp}	A_{33}^{pp}

Pour une meilleure lisibilité, nous avons préféré utilisé x et y dans la notation des matrices locales pour représenter les composantes de la vitesse.

Après assemblage, nous obtenons :

$$\overline{\overline{A}} \vec{x} = \vec{b}$$

où $\overline{\overline{A}}$ est la matrice globale du système, \vec{b} le vecteur second membre global, dont les composantes sont pour le moment toutes nulles et \vec{x} le vecteur solution, que nous cherchons à déterminer. Les composantes de ce vecteur sont respectivement la vitesse suivant Ox, la vitesse suivant Oy, la vitesse suivant Oz et la pression du premier nœud, la vitesse suivant Ox, la vitesse suivant Oy, la vitesse suivant Oz et la pression du deuxième nœud, et ainsi de suite.

Nous avons décidé d'utiliser une loi de mélange inspirée de l'approche Volume of Fluid [Bas 06], pour laquelle la viscosité est calculée en P0 (constante sur l'élément) :

$$\eta = \tau\eta_1 + (1 - \tau)\eta_2$$

où η est la viscosité sur l'élément, η_1 et η_2 sont les viscosités des deux milieux et τ le taux de remplissage relatif au milieu 1 tel que :

$$\tau = \frac{\sum_{i=1}^{d+1} \phi_i + \sum_{i=1}^{d+1} |\phi_i|}{2 \sum_{i=1}^{d+1} |\phi_i|}$$

avec ϕ_i la distance signée par rapport à l'interface entre les deux milieux, pour le $i^{\text{ème}}$ nœud de l'élément.

De cette manière, la viscosité est égale à la viscosité de chacun des milieux dans les éléments non traversés par l'isovaleur zéro de la distance et a une valeur intermédiaire sur les éléments traversés.

D'autres lois de mélange sont envisageables. Par exemple, nous pouvons définir des lois P1, pour lesquelles la viscosité devient linéaire sur l'élément, mais il faut alors spécifier une épaisseur de mélange. Il est aussi possible de ne pas considérer une moyenne arithmétique pondérée par le taux de remplissage, comme nous avons choisi de le faire, mais une moyenne géométrique.

Il pourrait ainsi être intéressant d'utiliser ces différentes lois de mélange sur des cas simples afin de déterminer laquelle permet d'obtenir les meilleurs résultats.

2.9. Stockage des matrices

Afin de tirer partie des caractéristiques de la matrice globale, dont la plupart des termes sont nuls, un stockage en matrice creuse est utilisé. En effet, si un stockage classique, qui consiste à garder tous les termes de la matrice en mémoire, présente l'avantage de la simplicité, il est loin d'être optimal. Ainsi, si nous voulions résoudre les équations de Stokes sur un maillage 3D d'un million de nœuds en stockant tous les termes de la matrice, il faudrait déjà que nous soyons en mesure de garder en mémoire plus de 58 To de données (si nous faisons l'hypothèse que chaque valeur est codée sur 4 octets), ce qui représente une mémoire supérieure à celle disponible sur l'ensemble des nœuds du cluster du CEMEF.

En supposant que chaque nœud d'un maillage 3D a en moyenne 24 voisins (valeur légèrement supérieure à la réalité selon nous), c'est-à-dire 24 nœuds avec lesquels il possède au moins un élément en commun, cela signifie que sur une des 4 lignes correspondantes à ce nœud dans la matrice globale, seulement 100 valeurs seront non nulles, et ce quel que soit le nombre de degrés de liberté. Donc, dans notre exemple avec 1 000 000 de nœuds, il n'y aurait que 400 000 000 de valeurs non nulles sur une matrice comprenant 16 000 000 000 000 éléments, soit un pourcentage de remplissage de 0,0025 %.

Le stockage réellement utilisé porte le nom de CSR (Compressed Sparsed Row) [Saa 03] et il consiste à utiliser trois vecteurs : (i) un pour stocker les termes non nuls de la matrice, (ii) un

pour stocker les colonnes correspondantes à chaque terme non nul, (iii) un pour définir le numéro du premier élément de chaque ligne.

Pour la mise en œuvre numérique dans un contexte parallèle, nous renvoyons le lecteur à la thèse de Marie [Mar 97]. Nous tenons aussi à signaler que le parallélisme est assuré par l'utilisation de la librairie MPI [MPI 03].

3. Résolution des systèmes linéaires

Nous venons de voir dans la partie précédente comment obtenir les valeurs numériques du système $\overline{\overline{A}}\overline{\overline{x}} = \overline{\overline{b}}$. La matrice $\overline{\overline{A}}$ ne dépendant pas du vecteur solution $\overline{\overline{x}}$, ce système est dit linéaire.

La résolution des systèmes linéaires est cruciale pour obtenir des résultats cohérents. Il s'agit aussi de la partie de nos applications qui demande le plus de temps de calcul. Pour cette raison, il existe de très nombreux travaux dans la littérature à ce sujet. Nous avons volontairement choisi de ne pas nous attarder sur l'étude bibliographique de cette question extrêmement vaste, comme peut par exemple en attester le nombre de références (1699) de la revue de Fadeev & Fadeeva [Fad 81]. Malgré tout, si nous ne devons retenir que deux documents, il s'agirait sans doute des ouvrages de Saad [Saa 03] et de van der Vorst [Vor 03], centrés sur la résolution itérative de grands systèmes avec matrice creuse.

Comme il n'est pas possible de résoudre exactement les systèmes linéaires, nous définissons un vecteur résidu, que nous cherchons à annuler, de la manière suivante :

$$\overline{\overline{r}} = \overline{\overline{A}}\overline{\overline{x}} - \overline{\overline{b}}.$$

Quand le vecteur résidu est nul, le système est résolu. Cela étant impossible en pratique, nous faisons en sorte que la norme du résidu ne dépasse pas une valeur très faible à la fin de la résolution. Comme nous allons le voir, cela peut poser des problèmes. En effet il est possible que le résidu soit très faible sur certains nœuds alors que la solution obtenue est encore très éloignée de la vraie solution.

Si la résolution des systèmes linéaires a déjà été analysée précisément au CEMEF par Marie [Mar 97] ou encore Perchat [Per 00], il n'existe pas d'étude publiée pour la librairie CIMLib, au sein de laquelle la résolution des systèmes linéaires est confiée à PETSc [Bal 10]. Le seul travail pour lequel cette résolution a fait l'objet d'une attention particulière est la thèse de Basset [Bas 06], dans laquelle une indication est donnée au sujet du type de préconditionneur à utiliser en fonction de la dimension du problème. Le préconditionnement est une modification d'un système linéaire le rendant plus facile à résoudre par une méthode itérative [Saa 03].

Nous décrivons à présent succinctement les deux méthodes disponibles dans CIMLib pour résoudre les systèmes linéaires : la méthode directe et la méthode itérative. Puis nous discutons des résultats obtenus sur quelques applications très simples avec les deux méthodes.

3.1. Méthode directe

Une méthode directe de résolution permet d'obtenir une solution exacte en un nombre d'étapes défini. Par exacte, il faut comprendre à la précision machine près. Ce type de méthode est très robuste mais demande une grande capacité mémoire et une importante puissance de calcul. En outre, la parallélisation d'une méthode directe est beaucoup plus

délicate à implémenter que celle d'une méthode itérative. Pour cette raison, la résolution directe n'est accessible que de manière séquentielle dans CIMLib, ce qui limite donc son utilisation à des cas avec un nombre de degrés de liberté réduit, que nous estimons inférieur à 500 000 sur un PC de bureau standard, ce qui correspond à un peu plus de 100 000 nœuds en 3D (avec 4 degrés de liberté par nœud).

Pour une description de méthodes directes avec des matrices creuses, nous renvoyons le lecteur à l'ouvrage de Duff et al. [Duf 86]

Les méthodes directes utilisent l'équivalence entre $\overline{\overline{A}}\overline{\overline{x}} = \overline{\overline{b}}$ et $\overline{\overline{x}} = \overline{\overline{A}}^{-1}\overline{\overline{b}}$.

Ainsi la méthode de factorisation LU codée dans PETSc et utilisée dans CIMLib donne une norme du résidu finale élevée si la matrice du système est singulière, c'est-à-dire non inversible. De plus quand une solution est obtenue par une méthode directe, elle ne dépend pas du vecteur solution initial. En réalité, ce n'est pas exactement le cas dans CIMLib, puisque la factorisation LU est définie comme un préconditionnement et qu'une méthode de résolution itérative est tout de même utilisée. Mais elle converge en une itération.

Utiliser le solveur direct nous permet donc de déterminer facilement si le problème est bien posé, ce qui est le cas quand il admet une solution unique. Nous estimons que cette vérification devrait constituer une étape prépondérante dans toute construction d'application CIMLib.

3.2. Méthode itérative

Pour résoudre les systèmes linéaires découlant de la formulation éléments finis utilisée ici de manière itérative, nous utilisons la même méthode que Basset [Bas 06], à savoir un préconditionnement de Jacobi par bloc [Saa 03] de 4 en 3D et de 1 en 3D avec une méthode ILU pour chaque bloc (factorisation LU incomplète) [Cha 94], associée à la méthode du résidu conjugué [Duf 64] pour la résolution du système. Cette méthode du résidu conjugué, très proche de celle du gradient conjugué, permet de résoudre des systèmes avec une matrice symétrique non définie positive, comme c'est le cas ici. La méthode du gradient conjugué a été initialement proposée par Hestenes & Stiefel [Hes 52]. Shewchuk en donne quant à lui une bonne introduction (qualifiée par l'auteur d'indolore) [She 94].

Pour la méthode itérative du résidu conjugué, le vecteur résidu est utilisé pour définir un vecteur direction, permettant de calculer une solution à partir de la solution à l'itération précédente. A chaque fois qu'un nouveau vecteur solution est calculé, le vecteur résidu et sa norme sont également recalculés. Si la norme est suffisamment faible, nous considérons que le système linéaire est résolu. Il y a deux façons de calculer le vecteur résidu, que nous appelons méthode initiale et méthode par cumul. La méthode initiale consiste à calculer le résidu actuel de la même façon que le résidu initial ($\overline{\overline{r}}_i = \overline{\overline{A}}\overline{\overline{x}}_i - \overline{\overline{b}}$ avec i qui désigne le numéro de l'itération actuelle) et la méthode par cumul utilise le résidu et la direction à l'itération précédente. La méthode par cumul demande un nombre d'opérations moins important, elle prend donc un temps de calcul inférieur et c'est celle que nous utilisons. Toutefois, cette méthode cumulant les erreurs d'arrondis dues à la précision machine, il est possible qu'elle donne un résultat sensiblement différent de celui donné par l'autre méthode. C'est cette observation qui pousse Dufour à utiliser la méthode initiale toutes les 16 itérations, afin de limiter ces erreurs [Duf 64].

Les matrices obtenues par la méthode des éléments finis sont parfois mal conditionnées et les systèmes linéaires sont difficiles à résoudre par une méthode itérative. L'utilisation d'un préconditionneur est ainsi primordiale. Cela consiste à résoudre le système suivant :

Chapitre 3

$$\overline{\overline{C}}^{-1}\overline{\overline{A}}\overline{\overline{x}} = \overline{\overline{C}}^{-1}\overline{\overline{b}}$$

Avec $\overline{\overline{C}}^{-1}$ la matrice de préconditionnement, $\overline{\overline{C}}^{-1}\overline{\overline{A}}$ plus facilement inversible que $\overline{\overline{A}}$ et $\overline{\overline{C}}^{-1}$ plus rapide à calculer que $\overline{\overline{A}}^{-1}$.

Quand un préconditionneur est utilisé, la convergence se fait par défaut sur la norme du résidu préconditionné, qui est calculé de la manière suivante initialement :

$$\overline{\overline{r}} = \overline{\overline{C}}^{-1}\overline{\overline{A}}\overline{\overline{x}} - \overline{\overline{C}}^{-1}\overline{\overline{b}}$$

Si l'utilisateur décide de ne fixer aucun paramètre pour la résolution du système linéaire donné par le solveur de Stokes, la résolution par défaut est employée. Il s'agit d'une résolution par la méthode de généralisation de minimisation du résidu (GMRES) [Saa 86] avec préconditionnement ILU(1) (le 1 entre parenthèses signifiant par bloc de 1) [Cha 94, Saa 03], 3 000 itérations au maximum et une précision relaxée de paramètre $P = 10^{-7}$. La précision est la valeur en dessous de laquelle la norme du résidu doit être pour que la résolution prenne fin. La relaxation sert à changer la valeur de la précision en fonction du nombre d'itérations. Par défaut, la valeur exigée pour la norme du résidu est donnée par :

$$\|\overline{\overline{r}}\| < \frac{P}{200} i.$$

En couplant GMRES et ILU(1), cette résolution par défaut est orientée vers la résolution de problèmes 3D pour lesquels les termes d'inertie sont présents et qui donnent donc lieu à des matrices non symétriques [Bas 06]. Utiliser ces options n'est pas judicieux ici.

Si la précision souhaitée pour la norme du résidu n'est pas atteinte lorsque le nombre maximal d'itérations a été effectué, la résolution s'arrête et la solution calculée à la dernière itération est renvoyée. En général, atteindre le nombre maximal d'itérations n'est pas bon signe car la dernière solution calculée peut être moins bonne qu'une solution calculée à une itération précédente. Cependant, définir un nombre maximal d'itérations présente deux avantages. Premièrement, le temps de calcul maximal nécessaire peut facilement être estimé. En outre, cela permet d'obtenir une solution même si la précision demandée a été renseignée de manière hasardeuse. Dans certains cas, il est ainsi possible de calculer une solution satisfaisante même si la norme du résidu exigée est tellement faible qu'elle sera impossible à obtenir en un nombre d'itérations très important.

La précision peut être intuitivement perçue comme le paramètre clef d'une résolution itérative. En effet, si elle est trop importante, c'est-à-dire que la norme finale du résidu est importante, la résolution du système linéaire s'effectue en un nombre d'itérations faible, mais la solution n'est pas nécessairement satisfaisante. Au contraire, si la précision demandée est très faible, il est possible que la résolution s'embourbe dans des calculs inutiles alors qu'une solution admissible a déjà été obtenue. Dans certains cas, une solution correcte peut-être maintenue jusqu'à atteindre le nombre maximal d'itérations, ce qui n'a donc que pour conséquence d'augmenter le temps de calcul de manière significative. Mais dans d'autres cas plus complexes, la solution pourtant correcte en considérant une précision plus importante peut dégénérer et entraîner l'interruption de notre simulation.

Vouloir résoudre parfaitement nos systèmes linéaires n'a de toute façon pas de sens. Dans notre tentative de simuler des phénomènes physiques, des erreurs sont déjà introduites par l'utilisation d'un modèle numérique et d'une discrétisation éléments finis et obtenir une

norme du résidu très faible n'y changera rien. Il est donc indispensable de définir une précision intermédiaire et c'est là que réside toute la difficulté. Tout d'abord, le résidu étant préconditionné, il n'a pas d'unité et il est difficile de le relier à l'écart à la solution exacte. Ensuite, la convergence étant définie sur la norme du résidu, il faut sans aucun doute prévoir d'être de moins en moins exigeant à mesure que nous augmentons le nombre de degrés de liberté.

3.3. Applications simples

Dans cette partie, nous avons choisi d'illustrer les résultats des méthodes introduites sur des cas simples, afin de dégager un certain nombre d'observations. Nous utilisons toujours la méthode du résidu conjugué pour résoudre le système linéaire. Couplé au résidu conjugué, nous avons la factorisation LU (solveur direct), une absence de préconditionnement ou le préconditionnement ILU(4). La norme du résidu exigée est prise constante, sans relaxation, et sera précisée dans les cas où c'est nécessaire.

3.3.1. Problème mal posé

Nous donnons tout d'abord un exemple pour lequel, contrairement au solveur itératif, le solveur direct permet de détecter si un problème est mal posé. Notre objectif étant de simuler un test de traction d'un alliage d'aluminium cuivre à l'état pâteux, nous nous intéressons ici à un test de traction très simple.

Pour cela nous considérons un domaine carré de côté 1 m et possédant une viscosité de 1 Pa.s. Nous appliquons sur le bord supérieur de ce carré une vitesse de 1 m.s⁻¹ suivant Oy et une vitesse nulle suivant Oy sur le bord inférieur. Les conditions initiales sont une vitesse et une pression nulles. Le maillage de 13 216 nœuds et 26 015 éléments utilisé, avec une taille de maille moyenne h de 1 cm, est montré sur la figure 3.2 (en haut à gauche), ainsi que la vitesse suivant Ox calculée par le solveur direct (en haut à droite) et par le solveur itératif avec une précision de paramètre P = 10⁻⁷, un préconditionnement ILU(4) et une vitesse Ox initiale nulle (en bas à gauche) et de 10 m.s⁻¹ (en bas à droite).

Sans donner aucune condition aux limites sur le champ de vitesse suivant Ox, les équations de conservation admettent une infinité de solutions, qui sont données ci-dessous :

$$v_x = C + 0,5 - x ; v_y = y ; p = -2Pa$$

avec C une valeur réelle et le coin inférieur gauche l'origine du repère.

Toutes ces solutions ont la même vitesse suivant Oy et la même pression. C'est également le cas des solutions que nous avons obtenues. Nous pouvons aussi constater que les champs de vitesse suivant Ox calculés correspondent à une des solutions admissibles. Cependant, si la norme finale du résidu est inférieure à 10⁻⁷ pour les résolutions itératives, elle est supérieure à 10⁴ pour la résolution directe, ce qui signifie que la matrice du système linéaire est singulière (ou non inversible).

Il faut aussi remarquer que la valeur de C semble correspondre exactement à la valeur de la vitesse initiale suivant Ox pour les résolutions itératives. Cette observation pourrait permettre de conclure qu'il est possible de résoudre des problèmes mal posés de manière itérative sur des exemples plus complexes, comme le cas de l'inclusion dont la solution analytique est proposée par Gilormini & Montheillet [Gil 84]. Cela n'est toutefois pas rigoureux, car des glissements non souhaités peuvent apparaître en cours de simulation.

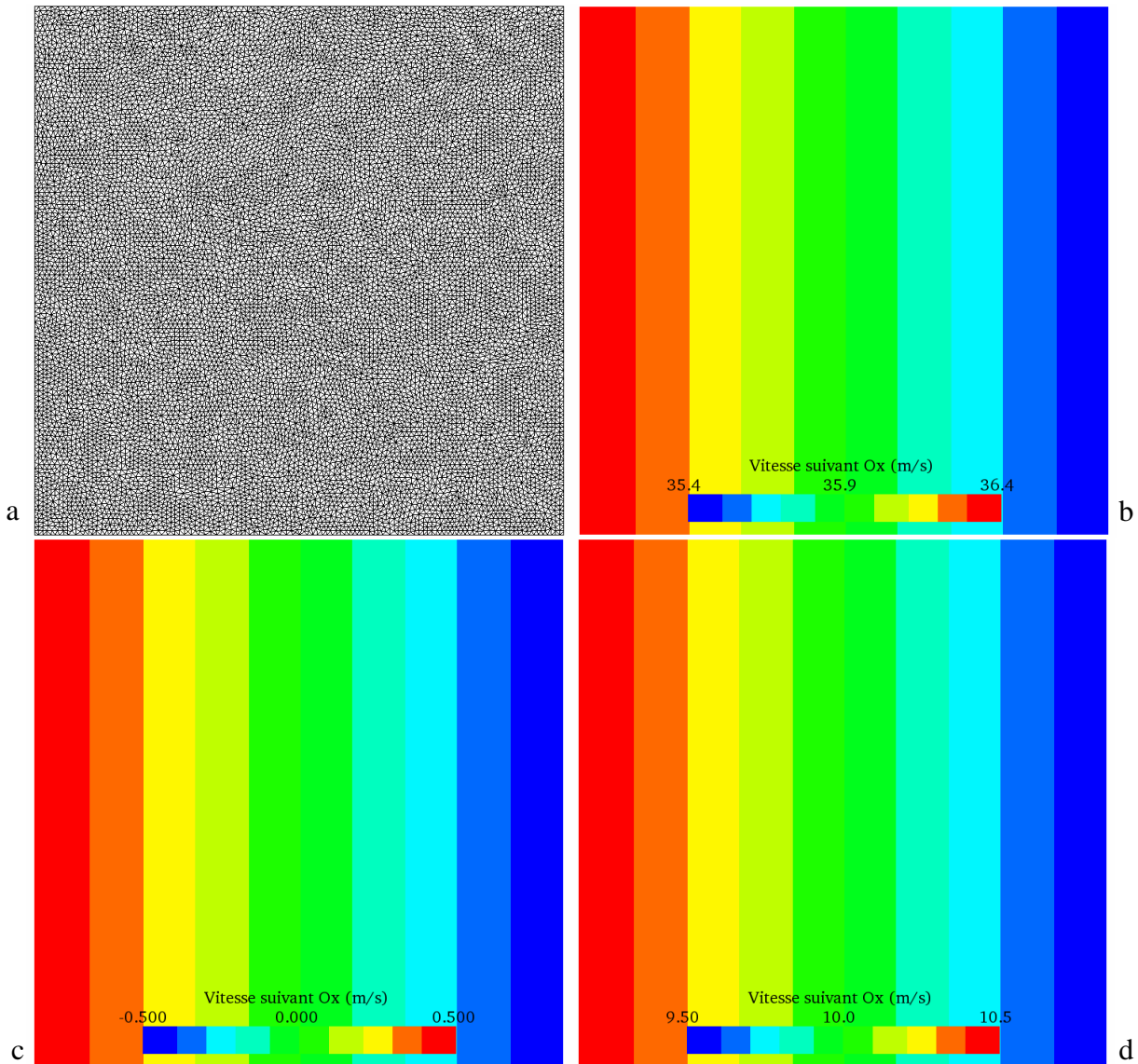


Figure 3.2. De gauche à droite et de haut en bas, (a) le maillage utilisé et les champs de vitesse suivant Ox calculés (b) par le solveur direct et par le solveur itératif avec une vitesse initiale de (c) 0 m.s^{-1} et de (d) 10 m.s^{-1} . L'échelle de couleur est différente pour les trois solutions.

3.3.2. Dimension des résidus

A partir des expressions des matrices locales données précédemment, il est possible de faire une analyse dimensionnelle du système linéaire à résoudre. Ainsi, le résidu en vitesse s'exprime en kg.s^{-2} en 2D et en kg.m.s^{-2} en 3D et le résidu en pression en $\text{m}^2.\text{s}^{-1}$ en 2D et en $\text{m}^3.\text{s}^{-1}$ en 3D. Les résidus ne sont donc pas homogènes. Ce constat, bien que gênant, ne doit pas poser de problème si les résidus en pression et en vitesse sont du même ordre de grandeur. Nous considérons le même carré de côté 1 m que précédemment. Cette fois-ci, nous appliquons une condition de symétrie sur le plan de gauche ($v_x = 0$) afin que le problème soit bien posé. La solution est très proche de celles du cas précédent :

Chapitre 3

$$v_x = -x ; v_y = y ; p = -2 \text{ Pa} .$$

Puisque les résidus ne sont pas homogènes, il est fort probable que le fait de changer la taille de maille ait une influence sur le rapport entre les résidus en vitesse et en pression. Pour nous en assurer, nous résolvons le système linéaire obtenu de manière itérative et sans préconditionnement pour différents maillages en utilisant une précision $P = 10^{-7}$. Pour chaque calcul, nous déterminons une erreur moyenne en vitesse et une autre en pression, de la manière suivante :

$$\bar{E} = \frac{\left(\int_{\Omega} e_h^2 dS \right)^{1/2}}{S_{\Omega}}$$

avec S_{Ω} la surface du domaine et e_h l'erreur en chaque nœud du domaine donnée par la différence entre valeur analytique et valeur numérique.

La figure 3.3 montre l'évolution des erreurs en fonction de la taille de maille. Contrairement au résultat recherché, diminuer la taille de maille entraîne une augmentation de l'erreur. En effet, cela a pour conséquence de diminuer la valeur des résidus en pression, qui ont alors un poids inférieur dans le calcul de la norme du résidu. Il est ainsi possible d'atteindre une norme du résidu faible sans que la pression soit déterminée de manière précise.

Afin d'obtenir une pression plus fiable, il est nécessaire d'augmenter la valeur des résidus en pression. Une méthode simple est de changer le système d'unités en utilisant des mm comme unité de longueur au lieu des m.

Par exemple, pour une taille de maille de 1 cm, l'erreur moyenne en pression passe de 0,102 mPa lorsque l'unité de longueur est le m à 0,190 μ Pa lorsque l'unité de longueur est le mm. Cependant, la norme du résidu initial étant beaucoup plus élevée, il faut un nombre d'itérations beaucoup plus important pour parvenir à une norme finale inférieure à 10^{-7} (et ainsi satisfaire $P = 10^{-7}$). En outre, obtenir des solutions dépendantes du système d'unité employé n'a rien de rassurant et il est donc nécessaire d'employer une autre méthode.

Pour pallier ce problème au CEMEF, Jaouen transforme la matrice globale du système de manière à rendre les résidus en vitesse et en pression tous les deux homogènes à une force [Jao 98]. Cette technique a montré son efficacité dans le logiciel THERCAST [Cos 04, Jao 98]. Malheureusement, elle n'est pas implémentée dans CIMLib et nous n'avons pas de données démontrant son efficacité dans le cas très simple que nous sommes en train d'examiner.

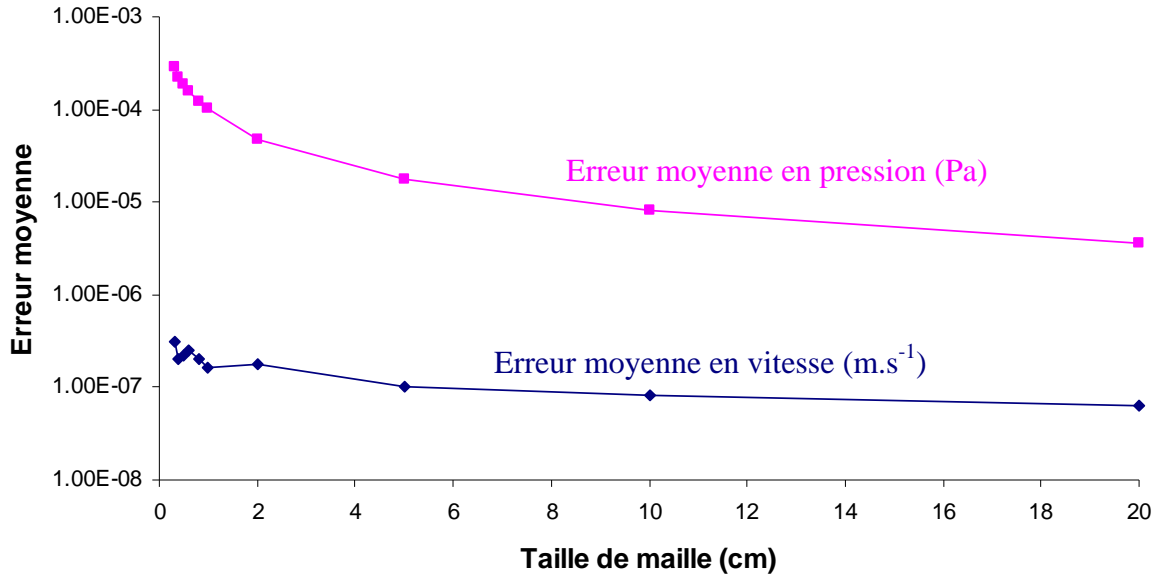


Figure 3.3. Erreurs moyennes en vitesse et en pression sans preconditionnement.

Pour s'affranchir du problème d'hétérogénéité des résidus en vitesse et en pression, c'est donc le preconditionnement ILU(k) qui est utilisé. La figure 3.4 montre les erreurs moyennes obtenues en vitesse et en pression pour un preconditionnement ILU(4). Ce preconditionnement permet, hormis une diminution considérable du nombre d'itérations pour atteindre une norme finale du résidu inférieure à 10^{-7} , de donner des erreurs beaucoup plus faibles qui semblent diminuer lorsque la taille de maille décroît, malgré des oscillations. Ces oscillations sont sans doute provoquées par des erreurs d'arrondis dues à la précision machine et par l'utilisation d'une méthode de calcul du résidu par cumul.

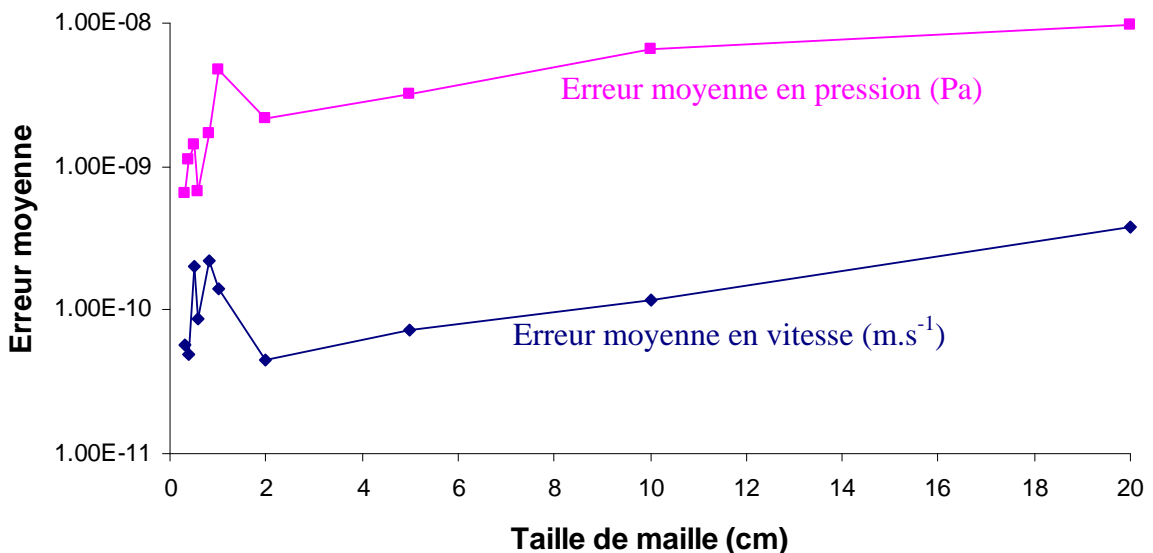


Figure 3.4. Erreurs moyennes en vitesse et en pression avec preconditionnement.

Même si nous ne pouvons nier le caractère indispensable du preconditionneur dans le but de construire des applications de plus grande ampleur, il faut tout de même souligner que l'introduction de ce preconditionneur entraîne une diminution plus erratique de la norme du

résidu au fil des itérations. Il arrive ainsi que la norme du résidu soit plus importante à une itération ultérieure, ce qui n'est pas le cas sans préconditionneur. Si ce phénomène se révèle anecdotique sur ce cas simple, il peut devenir prépondérant dans d'autres situations en empêchant d'atteindre la norme finale souhaitée. Pour sortir de la boucle de résolution, c'est alors le paramètre du nombre maximal d'itérations qui prend le relais. Dans ce cas, il est fort probable que la norme du résidu finale ne soit pas la valeur la plus faible calculée au cours de la résolution et par conséquent que la solution finale ne soit pas non plus la meilleure obtenue.

Dans le cas que nous venons d'examiner, le préconditionneur ILU(4) a pu gommer l'effet d'hétérogénéité des résidus, mais il est légitime de nous questionner sur sa capacité à le faire si nous utilisons des systèmes d'unité complètement différents. Nous considérons la même application en changeant l'unité utilisée pour la longueur. C'est en effet elle qui a l'impact le plus important sur l'ordre de grandeur des termes contenus dans les matrices locales. En passant de m à des mm, le rapport entre les termes de la matrice en vitesse-vitesse et ceux de la matrice en pression-pression est divisé par 10^9 (10^{12} en 3D). Les figures 3.5 et 3.6 montrent respectivement les erreurs moyennes en vitesse et en pression ainsi que les normes initiale et finale du vecteur résidu pour une taille de maille de 1 cm et différentes unités.

Dans tous les cas, les erreurs sont très faibles. Nous pouvons tout de même observer que pour les normes initiales du résidu plus importantes, l'erreur obtenue est plus faible. Dans ces cas, la norme du résidu final calculé par cumul est inférieure à 10^{-7} , mais la norme du vrai résidu final est en fait plus importante, ce qui signifie que le système linéaire ne peut pas être mieux résolu. Pour le calcul en m, la résolution se termine quand la norme est inférieure à 10^{-7} , alors qu'il serait possible de descendre encore un peu plus et donc d'obtenir une erreur plus faible. En conclusion, le système d'unité utilisé ne semble pas beaucoup affecter les résultats.

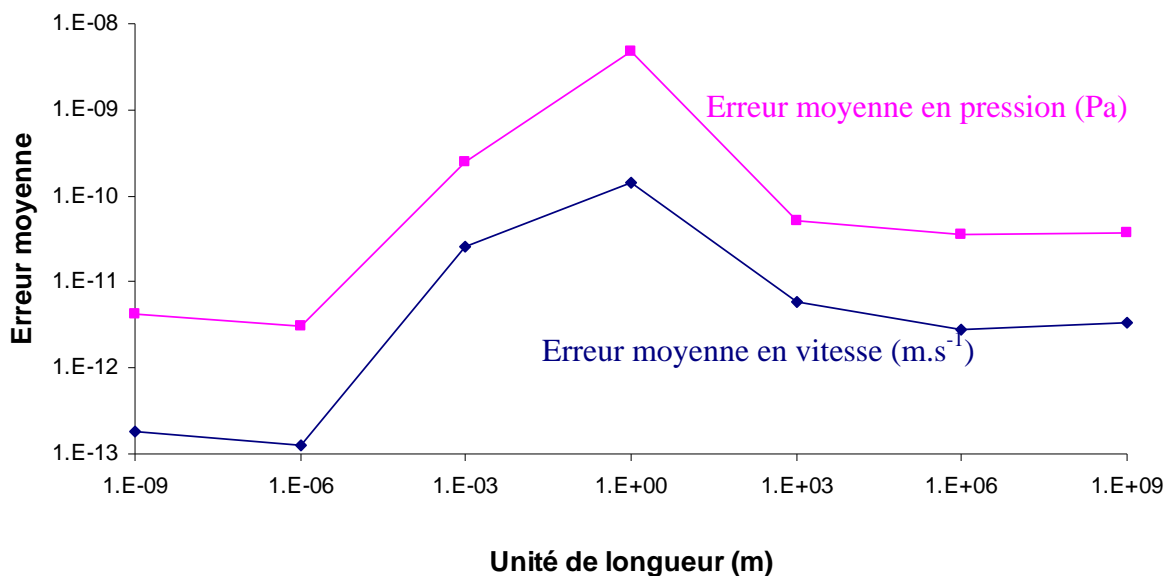


Figure 3.5. Erreurs moyennes en vitesse et en pression en fonction de l'unité de longueur.

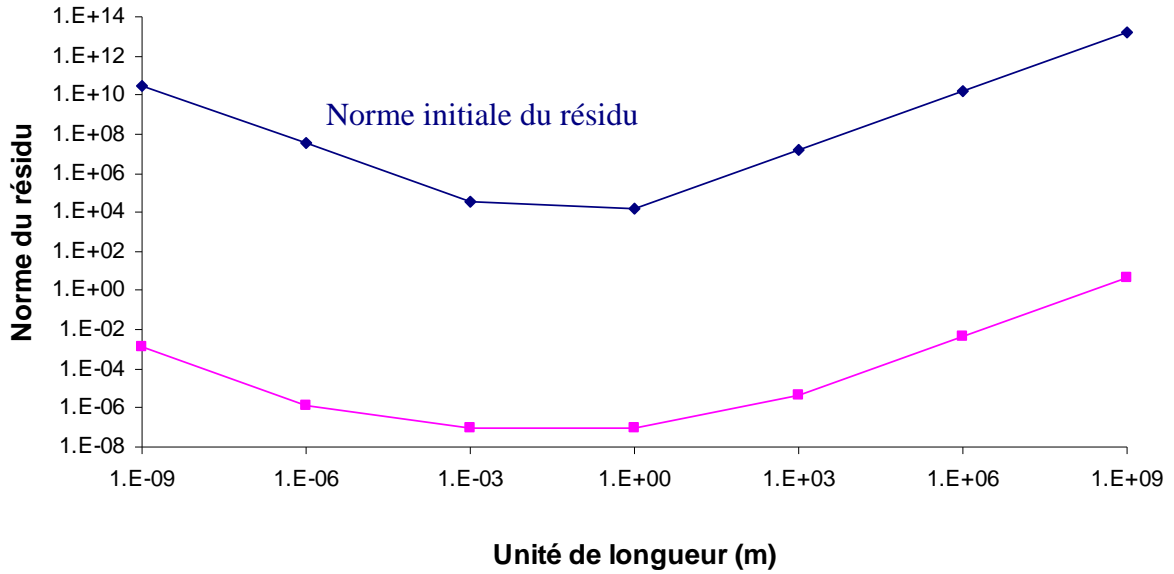


Figure 3.6. Normes initiale et finale du résidu en fonction de l'unité de longueur.

3.3.3. Solveur direct ou itératif

Perchat explique que le temps de résolution des solveurs directs est une fonction du nombre de nœuds au carré [Per 00]. Il parvient à concurrencer les solveurs directs par un solveur itératif en terme de temps de calcul sur des problèmes de plus de 5 000 nœuds, qui correspondent aujourd'hui à des problèmes de taille très modeste. Pour savoir si cette tendance est la même dans l'environnement CIMLib/PETSc, nous comparons le temps de calcul et l'erreur en utilisant le solveur direct et le solveur itératif (ILU(4) + résidu conjugué) pour différentes précisions. La machine utilisée est un PC de 3,25 Go de RAM avec un processeur Intel Core de fréquence 2,99 GHz. La figure 3.7 nous montre la comparaison du temps de calcul et la figure 3.8 celle de l'erreur moyenne en vitesse.

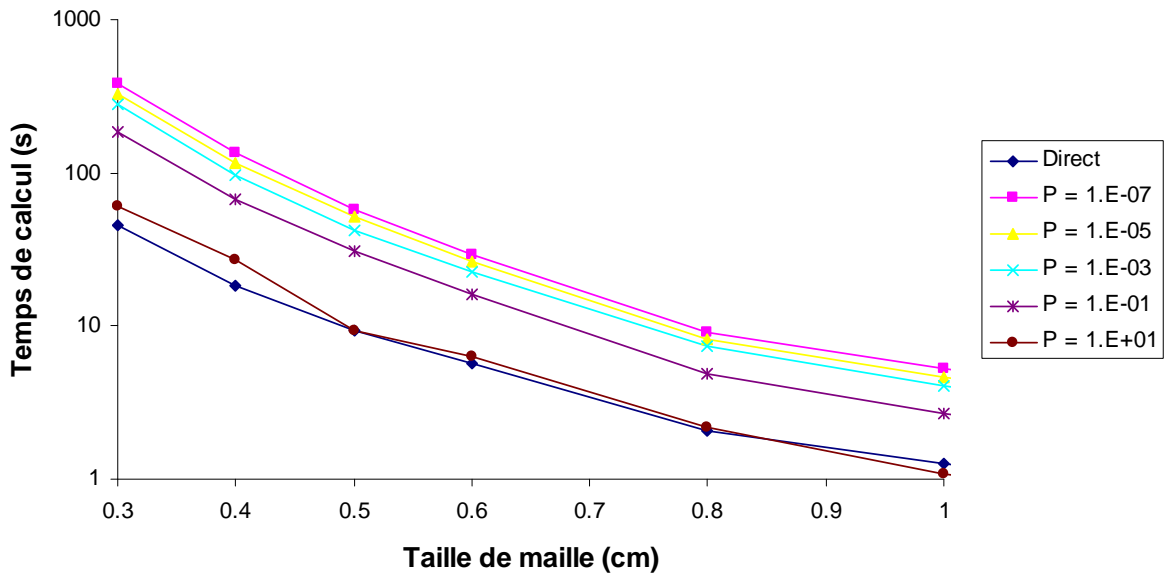


Figure 3.7. Temps de calcul pour les solveurs direct et itératif avec différentes précisions.

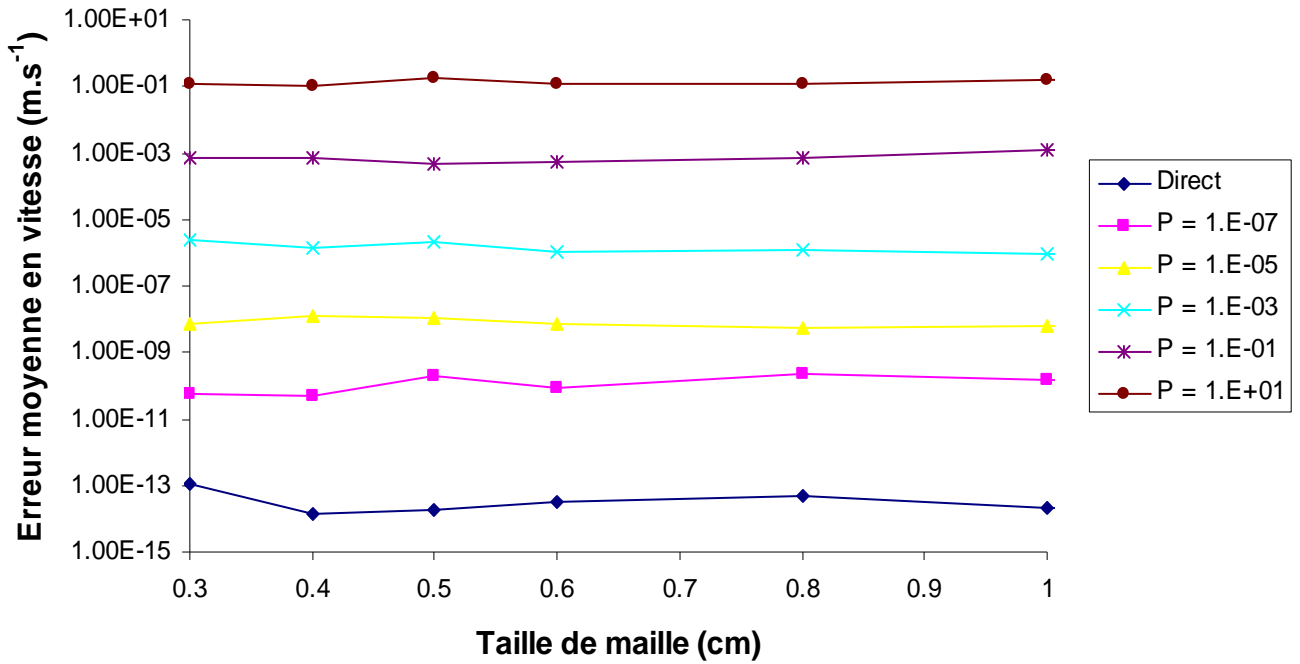


Figure 3.8. Erreur moyenne en vitesse pour les solveurs direct et itératif avec différentes précisions.

Nous constatons que le solveur direct est beaucoup plus précis que le solveur itératif. De plus, si pour le solveur itératif le temps de calcul dépend de la précision souhaitée, il est plus important que pour le solveur direct, même lorsque l'erreur en vitesse obtenue est grande. Nous nous attendions à ce que le solveur direct soit plus rapide que le solveur itératif lorsque la taille de maille est grande et le nombre de nœuds faible, et moins rapide pour des maillages comprenant un plus grand nombre de degrés de liberté. Or la figure 3.9 nous montre que c'est le contraire qui se produit. Plus la taille de maille diminue et plus le rapport entre le temps de calcul pour le solveur itératif et le temps de calcul pour le solveur direct augmente.

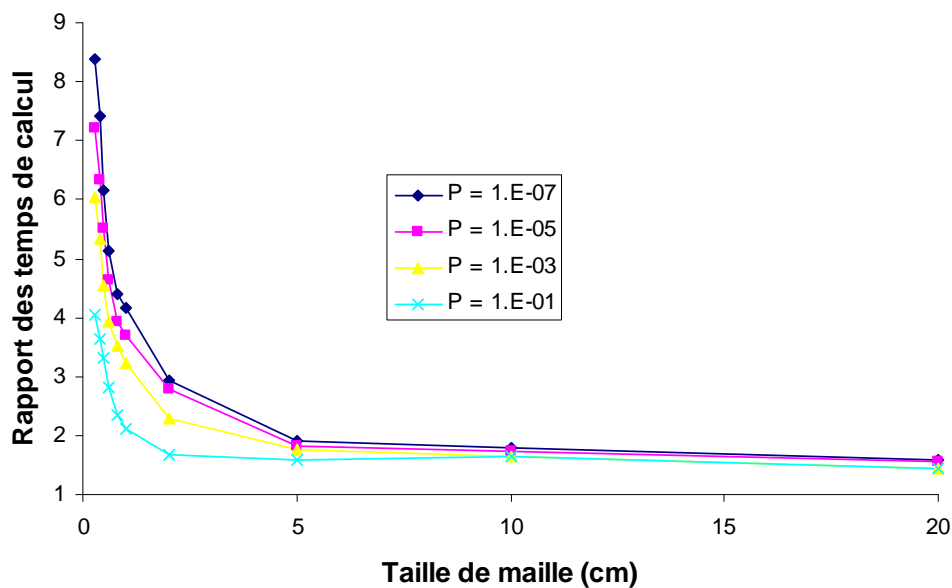


Figure 3.9. Rapport entre le temps de calcul pour les solveurs itératif et direct.

Nous avons ajusté une courbe de tendance sur la figure 3.7 pour estimer l'évolution du temps de calcul consommé par le solveur direct en fonction de la taille de maille. L'exposant calculé est très proche de -3. En considérant que le nombre de nœuds est donné en fonction de la taille de maille par la relation $N = Ch^{-2}$ (avec C une constante), nous pouvons donc affirmer que le temps de calcul est une fonction du nombre de nœuds à la puissance 1,5. Le solveur direct proposé par CIMLib/PETSc est donc beaucoup plus efficace que celui utilisé par Perchat [Per 00]. L'explication vient sans doute d'une bien meilleure exploitation de la structure creuse de la matrice.

Pour $h = 3$ mm, la plus petite taille de maille utilisée, le maillage comprend 136 944 nœuds. Il serait bien sûr très intéressant de continuer la comparaison entre les solveurs direct et itératif pour des systèmes de plus grande taille, mais cela n'a malheureusement pas été possible, car le solveur direct nécessite une mémoire trop importante et qu'il n'est pas utilisable dans un environnement parallèle.

Puisque nous abordons actuellement la question des efficacités respectives des deux solveurs utilisés, nous devons également remarquer que, pour l'application présentée ici, la viscosité est constante. Si ce n'était plus le cas, le solveur direct permettrait de résoudre les systèmes linéaires en un temps identique, les valeurs non nulles de la matrice globale étant situées à la même position. Au contraire, le solveur itératif pourrait souffrir d'un net ralentissement, dû à un moins bon conditionnement de la matrice, et donc voir son attractivité par rapport à son concurrent direct encore davantage réduite.

En guise de conclusion, nous préconisons fortement l'emploi du solveur direct pour la résolution d'un système linéaire découlant de l'application du solveur local de Stokes pour un problème bidimensionnel. Le recours au solveur itératif ne devrait donc se faire que si le solveur direct est pris à défaut faute de mémoire.

4. Test de l'inclusion

Nous nous intéressons à présent à un cas de traction plus complexe que précédemment. Il s'agit d'un problème de traction d'une inclusion sphérique placée dans une matrice infinie, avec un comportement newtonien de l'inclusion et de la matrice. Ce problème nous permet donc de considérer deux phases de viscosité différente, ce que nous n'avons pas fait jusqu'à présent. Ce test est très intéressant car il présente une solution analytique. Elle est donnée par Gilormini, Montheillet & Germain [Gil 84, Gil 86, Gil 87] et permet de connaître parfaitement la valeur des champs de vitesse et de pression sur l'ensemble du domaine. De plus, même si nous nous contentons ici d'une étude mécanique statique bidimensionnelle, il faut souligner que la solution analytique s'étend à la dynamique de l'inclusion tridimensionnelle, que nous considérons dans le cinquième et dernier chapitre.

Pour la solution en 2D, nous nous reportons au premier des trois articles de Gilormini [Gil 84]. Un aspect prépondérant de la solution est l'uniformité de la vitesse de déformation équivalente et de la pression à l'intérieur de l'inclusion.

4.1. Maillage utilisé

Ne pouvant considérer un domaine de calcul infini, nous prenons une inclusion de 25 μm de rayon placée au milieu d'un carré de côté 1 mm. Nous imposons une vitesse suivant Oy de 0,05 $\mu\text{m}\cdot\text{s}^{-1}$ sur le plan supérieur et de -0,05 $\mu\text{m}\cdot\text{s}^{-1}$ sur le plan inférieur. Pour bien poser le problème, nous appliquons une vitesse suivant Ox nulle sur le nœud situé au milieu du plan inférieur. Le rapport entre le côté du carré et le rayon de l'inclusion est suffisant dans la mesure où les champs de vitesse et de pression analytiques au bord d'un domaine de cette

taille ne sont presque pas modifiés par la présence de l'inclusion. Une autre option, privilégiée par El Khaoulani dans sa thèse [Elk 10] consiste à appliquer directement sur les bords du domaine les valeurs analytiques du champ de vitesse. Il est également possible de ne considérer qu'un quart de la géométrie en tenant compte de la symétrie du problème.

L'écoulement étant perturbé par la présence de l'inclusion, il est logique de considérer un maillage avec une taille de maille plus fine à l'interface entre la matrice et l'inclusion et de plus en plus grossière à mesure que nous nous éloignons de cette interface. Nous utilisons la taille de maille suivante :

$$\begin{aligned} h &= h_{\min} && \text{si } |\phi| < E \\ h &= h_{\min} + f(|\phi| - E) && \text{si } |\phi| > E \\ h &= h_{\max} && \text{si } h > h_{\max} \end{aligned}$$

Avec h_{\min} la taille de maille à l'interface, E l'épaisseur de la zone avec une taille de maille constante, ϕ la distance signée à l'interface, f le coefficient qui régit l'augmentation de la taille de maille en fonction de la distance et h_{\max} la taille de maille maximale, utilisée pour ne pas obtenir de trop gros éléments loin de l'interface.

Ainsi la figure 3.10 nous donne la taille de maille en fonction de la distance à l'interface, pour des paramètres $h_{\min} = 1 \mu\text{m}$, $h_{\max} = 50 \mu\text{m}$, $E = 35 \mu\text{m}$ et $f = 0,2$, ainsi que le maillage associé.

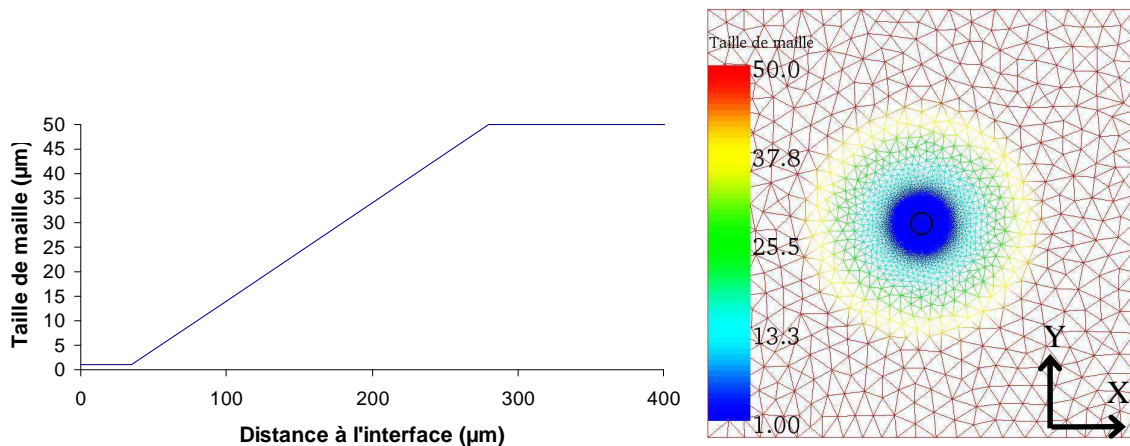


Figure 3.10. Taille de maille en fonction de la distance à l'interface (à gauche) et maillage associé (à droite). Le trait noir représente le contour de l'inclusion.

4.2. Solution numérique

En utilisant le maillage de la figure 3.10 et une résolution itérative avec une précision non relaxée $P = 10^{-7}$, nous avons lancé la simulation pour des viscosités dans la matrice et dans l'inclusion de 10 GPa.s et 10 kPa.s, soit un rapport de 10^6 entre les deux.

Sur la figure 3.11, nous représentons les valeurs des champs de vitesse analytique et numérique sur le contour de l'inclusion. Pour tracer la solution numérique, nous avons interpolé les valeurs données par la simulation sur des capteurs répartis uniformément sur l'interface matrice inclusion (un capteur tous les $4,5^\circ$ pour un total de 80 capteurs).

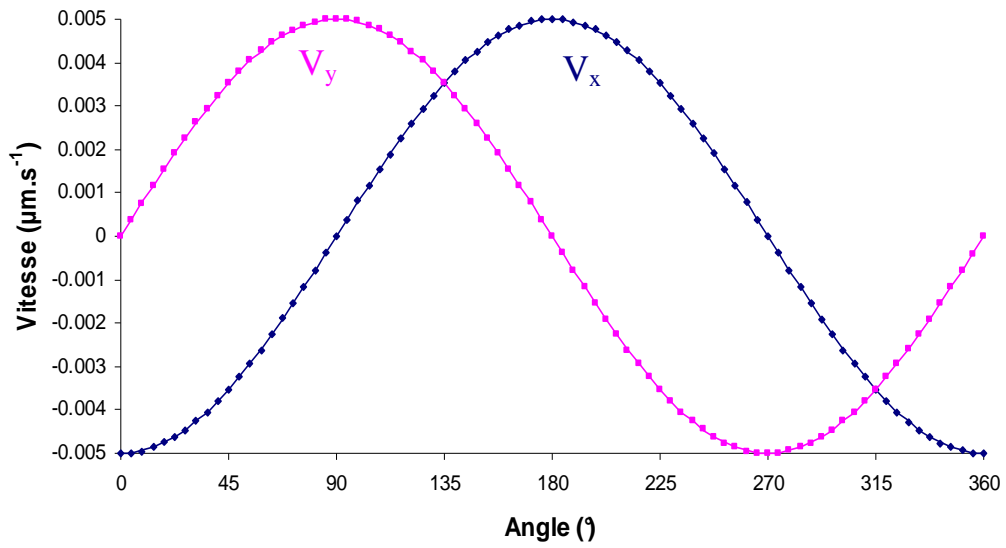


Figure 3.11. Champ de vitesse sur le contour de l'inclusion. L'angle est calculé par rapport à l'axe des abscisses. Les vitesses analytique et numérique sont respectivement représentées par un trait plein et des marques ponctuelles.

Les solutions analytique et numérique sont très proches dans ce cas et il en est de même pour le champ de pression, malgré la présence de quelques oscillations à l'interface, comme le montre la figure 3.12.

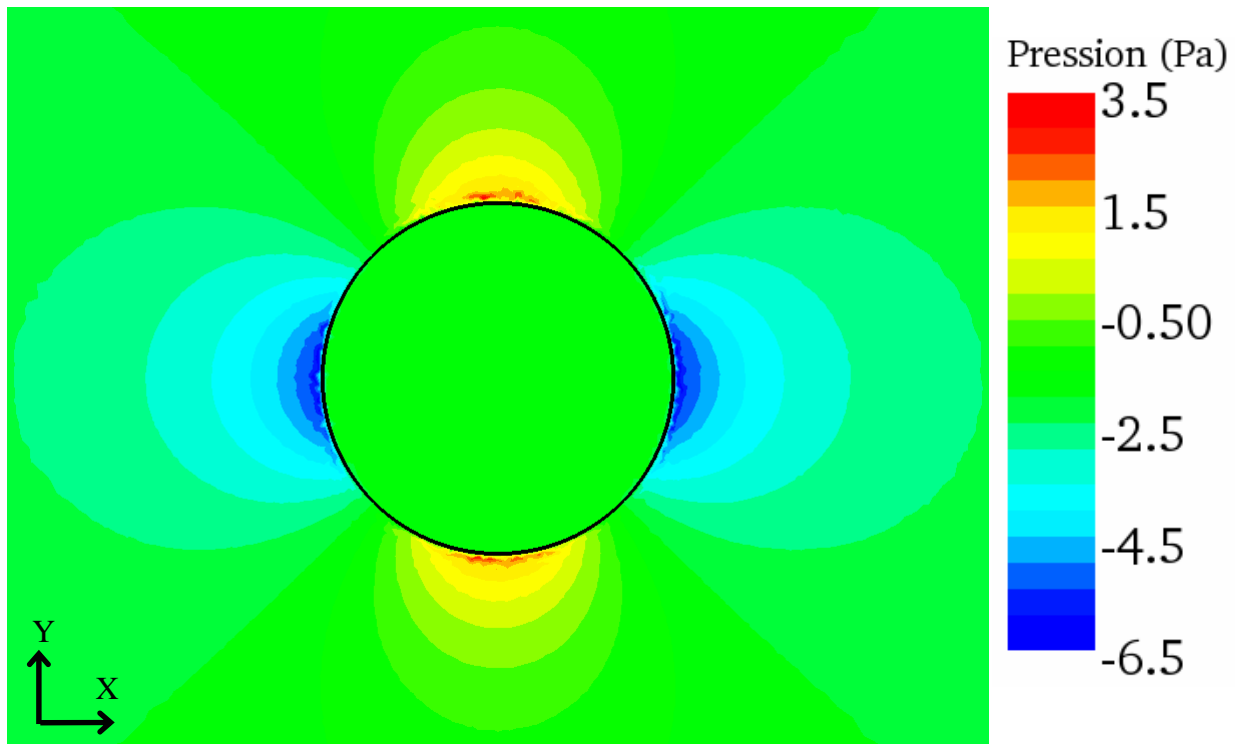


Figure 3.12. Champ de pression autour de l'inclusion.

4.3. Rapport des viscosités et précision

Un autre aspect important de la solution analytique est la très faible influence du rapport des viscosités de la matrice et de l'inclusion lorsqu'une certaine valeur est dépassée. Concrètement, au-delà d'un rapport de 1 000 entre les deux viscosités, il n'est plus possible d'observer de différences à l'œil nu sur les courbes analytiques de la figure 3.11. Nous souhaitons maintenant déterminer si les résultats numériques obtenus sont toujours très proches de la solution analytique pour un rapport de viscosité encore plus grand que 10^6 . Pour cela, nous relançons le calcul en considérant différentes viscosités de l'inclusion et différentes précisions pour la résolution du système linéaire. Pour avoir une idée de la justesse de chaque résolution, nous calculons la moyenne de la valeur absolue de l'écart à la solution analytique pour les 80 capteurs utilisés pour le tracé de la figure 3.11. La figure 3.13 représente cette erreur moyenne pour la composante du champ de vitesse suivant l'axe Ox pour des rapports de viscosité entre la matrice et l'inclusion de 10^6 , 10^8 , 10^{10} , 10^{12} et 10^{14} et des précisions P s'échelonnant entre 1 et 10^{-12} .

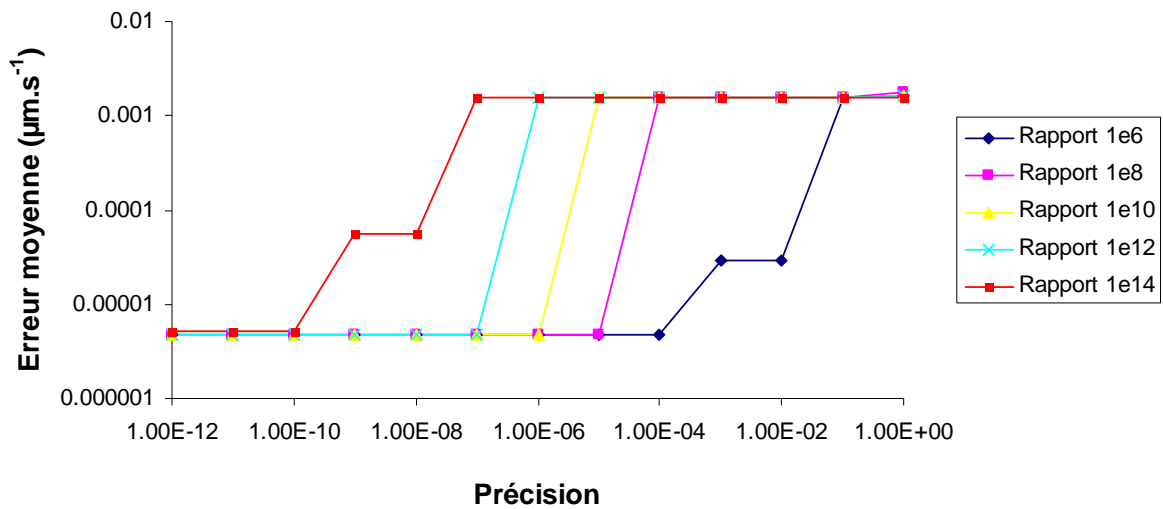


Figure 3.13. Ecart moyen à la solution analytique pour le champ de vitesse suivant Ox en fonction de la précision de la résolution du système linéaire et du rapport des viscosités.

Quel que soit le rapport des viscosités, une norme finale du résidu trop grande implique une erreur importante, quasiment identique pour toutes les simulations. En effet, les points à droite de la figure 3.13 représentent une erreur moyenne de $1,6 \text{ nm.s}^{-1}$, alors que la vitesse maximale suivant Ox sur le contour de l'inclusion est de 5 nm.s^{-1} . De manière similaire, en dessous d'une certaine norme finale du résidu, les erreurs sont très faibles, elles correspondent au résultat montré sur la figure 3.11.

La figure 3.13 nous montre aussi que la précision nécessaire pour obtenir une erreur faible diminue quand le rapport des viscosités augmente, ce qui augmente également le temps de calcul. Nous devons aussi préciser que s'il est possible d'obtenir une norme du résidu finale très faible dans ce cas simple, ce ne sera plus le cas dans d'autres simulations. Par conséquent, il apparaît plus prudent de ne pas considérer de rapports de viscosité trop importants, même si nous devons utiliser des viscosités non réalistes pour cela.

4.4. Utilisation d'un maillage plus adapté

La taille de maille utilisée dans cette partie a été choisie de manière très simple et n'est pas optimale dans l'optique de minimiser le nombre de nœuds et par conséquent le temps de calcul pour une précision donnée. Pour obtenir un maillage plus efficace, nous pourrions avoir recours à une métrique anisotrope obtenue avec un estimateur d'erreur. Nous avons cependant choisi de ne pas présenter une telle étude, car nous verrons dans le prochain chapitre que le solveur d'interface, qui est couplé avec le solveur mécanique, ne permet pas l'utilisation d'un maillage anisotrope dans notre cas.

5. Comportement viscoplastique

Nos partenaires du SIMaP ont effectué des tests de compression sur des échantillons solides d'aluminium-cuivre, pour diverses compositions en cuivre et différentes températures. Ils en ont déduit une loi de comportement viscoplastique, pour laquelle la contrainte équivalente répond à une loi de type puissance à une température fixée :

$$\bar{\sigma} = K \dot{\bar{\epsilon}}^m$$

avec K la consistance, $\dot{\bar{\epsilon}}$ la vitesse de déformation équivalente et m le coefficient de sensibilité à la vitesse de déformation.

Nous utilisons les expressions de la contrainte équivalente et de la vitesse de déformation équivalente données ci-dessous :

$$\bar{\sigma} = \sqrt{\frac{3}{2} \bar{\mathbf{s}} : \bar{\mathbf{s}}} ; \dot{\bar{\epsilon}} = \sqrt{\frac{2}{3} \bar{\bar{\boldsymbol{\epsilon}}}(\bar{\mathbf{v}}) : \bar{\bar{\boldsymbol{\epsilon}}}(\bar{\mathbf{v}})}$$

avec $\bar{\mathbf{s}}$ le tenseur déviateur des contraintes donné par $\bar{\mathbf{s}} = 2\eta_{\text{app}} \bar{\bar{\boldsymbol{\epsilon}}}(\bar{\mathbf{v}})$ où η_{app} est la viscosité apparente du solide.

En combinant les expressions de la loi de comportement et celles de la contrainte équivalente, de la vitesse de déformation équivalente et du déviateur des contraintes, nous en déduisons la relation entre la viscosité apparente du solide et les deux paramètres de la loi de comportement :

$$\eta_{\text{app}} = \frac{K}{3} \dot{\bar{\epsilon}}^{m-1}$$

La loi de comportement considérée ici est en accord avec celles décrites dans la littérature pour les alliages d'aluminium à chaud [Ful 09, Phi 08b]. Contrairement au comportement newtonien, cette loi est non linéaire : la viscosité apparente est en effet une puissance $m-1$ de la vitesse de déformation. Ainsi la matrice globale du système dépend du vecteur solution. Une manière classique de résoudre ce type de comportement est l'utilisation de la méthode de Newton-Raphson, comme par exemple dans la thèse de Hamide [Ham 08] et de beaucoup d'autres thèses au CEMEF. Cependant, dans le contexte de résolution CIMLib/PETSc, cette méthode n'est pas implémentée de façon optimale. Richard a ainsi montré que le codage actuel peut présenter de graves problèmes de convergence pour les faibles coefficients de sensibilité à la vitesse de déformation [Ric 10]. Il a également proposé une méthode pour

mieux interfacer CIMLib et les solveurs non linéaires de PETSc mais ses développements n'ont malheureusement pas encore été pérennisés.

Nous avons par conséquent décidé de résoudre les équations de conservation résultantes avec le solveur linéaire présenté précédemment dans ce chapitre. Pour cela, nous calculons la viscosité apparente du solide, que nous utilisons pour assembler le système linéaire à résoudre. La résolution du système linéaire nous donne un champ de vitesse, qui nous permet de déterminer la vitesse de déformation équivalente et la nouvelle viscosité apparente. Avec cette nouvelle viscosité apparente, nous reconstruisons un système linéaire et ainsi de suite, jusqu'à ce que suffisamment d'itérations linéaires aient été menées. Cette méthode est appelée méthode du point fixe.

En ce qui concerne la convergence pour la résolution de ce comportement non linéaire, nous pouvons définir un rapport entre la norme initiale du résidu de la première itération linéaire et celle de l'itération en cours, au dessus duquel nous considérons que la résolution est terminée. Il est également possible d'utiliser un rapport entre la norme initiale du résidu de l'itération linéaire en cours et la norme finale que nous exigeons. En effet si ce rapport est faible, cela signifie que notre itération linéaire actuelle ne peut qu'apporter de légères modifications sur les champs de vitesse et de pression et il apparaît donc superflu de poursuivre la résolution.

Sur la figure 3.14, nous avons tracé l'évolution de la viscosité apparente du solide en fonction de la vitesse de déformation équivalente avec une consistance de 30 MPa et un coefficient de sensibilité à la vitesse de déformation de 0,2, qui correspondent à peu près aux paramètres identifiés par nos partenaires du SIMaP. Avec une telle sensibilité à la vitesse de déformation, la viscosité est multipliée par 6,31 à chaque fois que la vitesse de déformation est divisée par 10. Ainsi la viscosité devient très importante pour les très faibles vitesses de déformation. Avec une vitesse de déformation de l'ordre de 10^{-4} s^{-1} , qui correspond à la vitesse de déformation de l'expérience de Terzi et al. [Ter 09] que nous souhaitons reproduire, la viscosité apparente de la phase solide est d'environ 10 GPa.s. Mais durant le calcul, la vitesse de déformation peut localement avoir des valeurs beaucoup plus faibles, ce qui entraîne une viscosité très importante.

Or, nous avons vu dans la partie précédente que le rapport entre les viscosités maximale et minimale joue un rôle important dans la convergence de notre méthode. Sachant que la viscosité réelle de la phase liquide est de l'ordre de 10^{-3} Pa.s , nous utiliserons donc une viscosité du liquide augmentée pour réduire ce rapport. Nous nous servirons également d'une vitesse de régularisation $\dot{\bar{\epsilon}}_r$ en imposant $\dot{\bar{\epsilon}} \geq \dot{\bar{\epsilon}}_r$, ce qui nous permettra de diminuer la viscosité apparente maximale de la phase solide. La valeur à donner au paramètre $\dot{\bar{\epsilon}}_r$ fera l'objet d'une étude dans notre dernier chapitre.

Nous avons pour habitude de prendre des valeurs initiales nulles pour les champs de vitesse et de pression. Si cela ne pose aucun problème lorsque le comportement est newtonien, ce n'est plus le cas pour une loi puissance, car la viscosité équivalente ne peut pas alors être calculée. Pour la définir, nous utilisons une vitesse de déformation initiale donnée par les conditions aux limites et en considérant que le domaine a une viscosité homogène, même quand plusieurs phases sont présentes. De cette façon, nous pouvons laisser la vitesse nulle.

Pour calculer la viscosité dans un cas newtonien à deux phases, nous faisons un mélange en utilisant la formule $\eta = \tau\eta_1 + (1 - \tau)\eta_2$ définie dans la sous-partie 2.8 de ce chapitre. Pour calculer la viscosité dans une application à deux phases, avec une des deux phases viscoplastique et l'autre newtonienne, nous définissons un coefficient de sensibilité à la vitesse de déformation unitaire et une consistance égale à la viscosité dans la phase newtonienne, puis nous mélangeons le coefficient de sensibilité à la vitesse de déformation : $m = \tau m_1 + (1 - \tau)m_2$ et la consistance : $K = \tau K_1 + (1 - \tau)K_2$. m_1 , m_2 , K_1 et K_2 sont respectivement les coefficients de sensibilité à la vitesse de déformation et les consistances

des deux phases. Nous régularisons ensuite la vitesse de déformation : $\dot{\epsilon} \geq \dot{\epsilon}_r$ et déterminons la viscosité dans tout le domaine à l'aide du résultat des mélanges : $\eta_{app} = K\dot{\epsilon}^{m-1} / 3$. En procédant de la sorte, la viscosité apparente de la phase newtonienne est bien égale à sa viscosité.

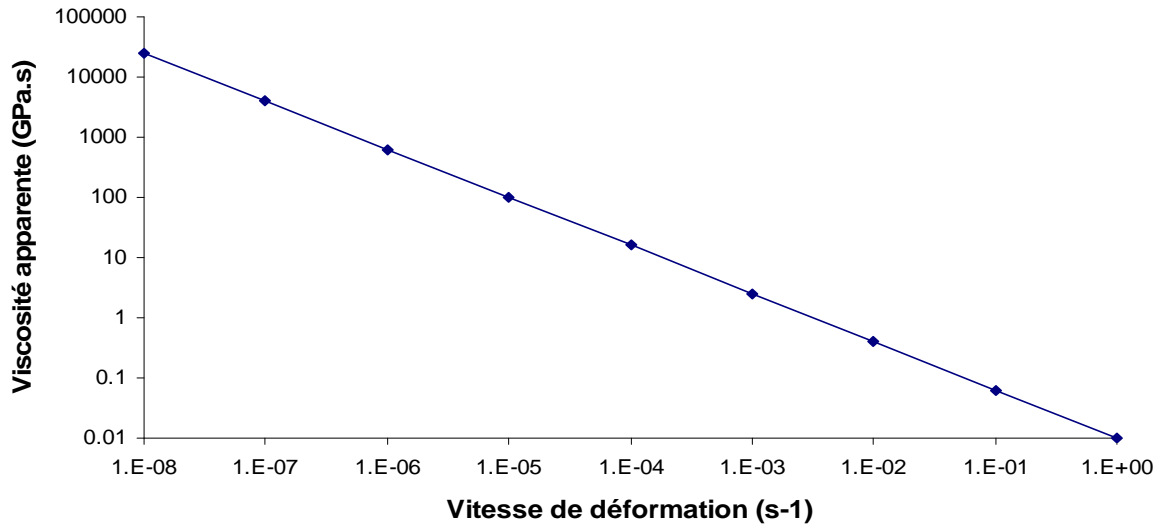


Figure 3.14. Viscosité apparente du solide en fonction de la vitesse de déformation équivalente.

Pour déterminer une loi de comportement du solide, les essais de compression du laboratoire SIMaP ont été menés à vitesse de déformation variable par paliers successifs. Dans le tableau 3.1, nous pouvons trouver les valeurs des paliers de vitesses de déformation considérés ainsi que les valeurs de déformation maximales associées. Le coefficient de sensibilité à la vitesse de déformation et la consistance ont pu être déterminés à partir des courbes en contrainte déformation. A une température de 525 °C, nous avons ainsi obtenu une valeur constante de 0,206 pour la sensibilité à la vitesse de déformation et des consistances respectives de 19,32 MPa, 29,06 MPa et 36,28 MPa pour des compositions en cuivre de 1, 2 et 4 %.

Pour simuler ces expériences, nous avons considéré un domaine cubique d'1 mm de côté, une taille de maille de 100 μm et des vitesses de traction allant de 0,1 $\mu\text{m.s}^{-1}$ à 20 $\mu\text{m.s}^{-1}$. La valeur du pas de temps a été déterminée pour obtenir des résultats tous les 0,5 % de déformation. La figure 3.15 montre les courbes obtenues numériquement et expérimentalement pour des compositions en cuivre de 1 %, 2 % et 4 % à une température de 525 °C.

Vitesse de déformation	10^{-4}	$2 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	10^{-3}	$8 \cdot 10^{-4}$	$2 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	$8 \cdot 10^{-3}$	10^{-2}	$2 \cdot 10^{-2}$
Déformation	0,04	0,08	0,12	0,16	0,2	0,25	0,325	0,4	0,475	0,55

Tableau 3.1. Vitesses de déformation utilisées au cours des essais

Au final, même s'il existe une différence pour les vitesses de déformation les plus élevées et les compositions de 2 et 4 %, nous constatons un bon accord entre la simulation et l'expérience et ce au moyen de la simple loi de Norton, sans écrouissage et à coefficients constants.

Chapitre 3

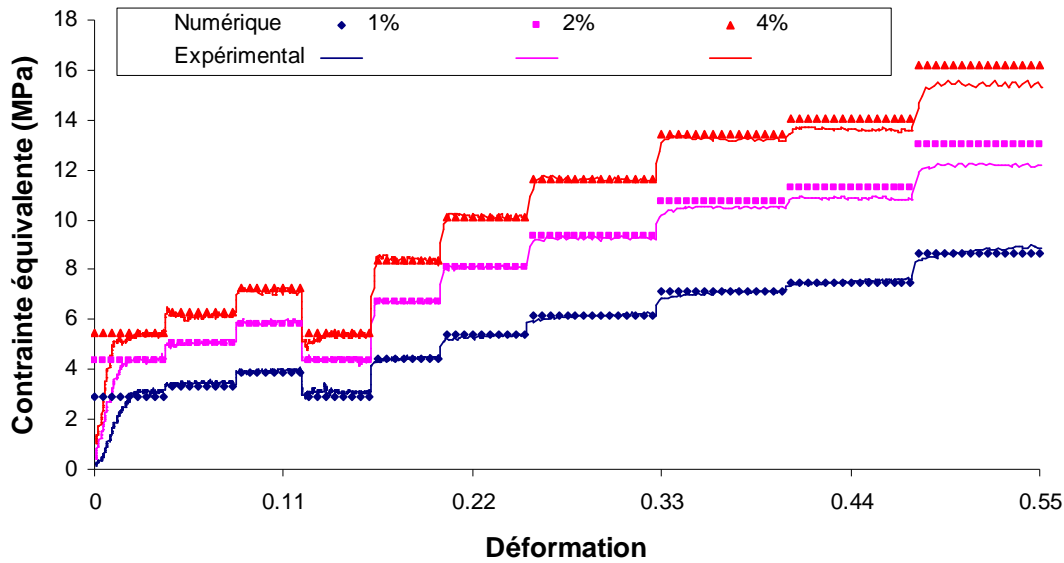


Figure 3.15. Comparaison des contraintes obtenues à 525°C pour différentes compositions en cuivre.

Pour résoudre le comportement non linéaire, nous utilisons à chaque incrément trois itérations de point fixe, c'est-à-dire trois résolutions de système linéaire. Ce nombre paraît faible, mais il est suffisant dans ce cas simple. Pour le prouver, nous avons tracé sur la figure 3.16 la norme du résidu initial pour les systèmes linéaires résolus pour l'alliage à 4 % de cuivre. Sur cette figure, nous voyons que seules quelques normes initiales sont élevées. Elles correspondent soit au début de la simulation, pour laquelle la vitesse est nulle, soit à un changement de vitesse de traction et donc de viscosité. Très souvent, la norme initiale du résidu a des valeurs faibles qui sont inférieures à la précision souhaitée et aucune résolution n'est par conséquent nécessaire.

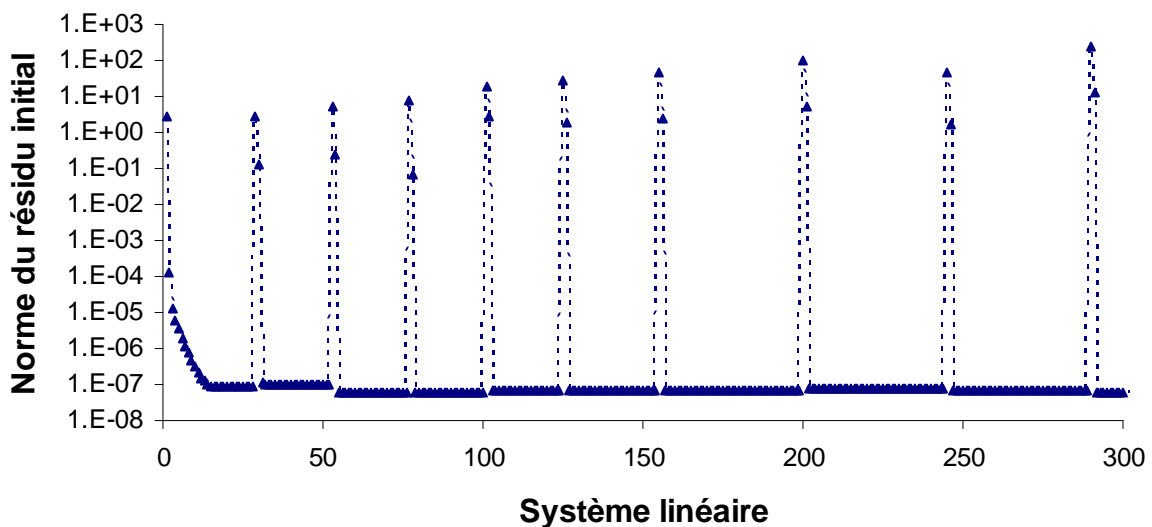


Figure 3.16. Norme du résidu initial pour chaque système linéaire.

6. Conclusion

Nous avons détaillé dans ce chapitre la méthode de résolution des équations de conservation de la masse et de la quantité de mouvement utilisée dans CIMLib. Pour cela nous avons dans un premier temps considéré un comportement newtonien pour lequel nous devons résoudre un système linéaire, par une méthode directe ou une méthode itérative. Même si la méthode de résolution itérative s'est montrée moins efficace que la méthode directe, l'utilisation de cette dernière est seulement séquentielle et reste donc limitée à des applications de taille modeste. Travailler à rendre la méthode directe parallèle semble donc une piste intéressante.

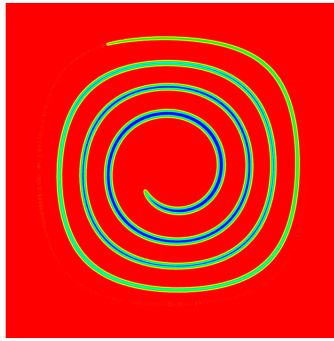
Nous nous sommes par la suite focalisé sur l'utilisation de la méthode itérative, qui consiste en un couplage d'un préconditionnement ILU(k) et de la méthode du résidu conjugué. Cette méthode nous a permis d'obtenir des résultats en accord avec la solution analytique du comportement d'une inclusion placée dans une matrice infinie soumise à un test de traction uniaxiale.

Nous avons également constaté que la méthode de résolution itérative décrite présente un certain nombre d'inconvénients. Pour commencer, le préconditionnement utilisé ne s'appuie pas sur la structure de la matrice du système et même si ce dernier est indispensable, il peut induire des augmentations de la norme du résidu au cours de la résolution et une certaine difficulté à converger. Une combinaison de la méthode actuelle avec la méthode d'homogénéisation des résidus utilisée dans THERCAST pourrait ainsi constituer une solution élégante. Pour la méthode du résidu conjugué, une perspective séduisante serait l'ajout de conditions d'arrêt supplémentaires et d'un calcul différent du vecteur résidu [Duf 64].

En outre, la résolution présentée nécessite de ne pas utiliser un rapport de viscosité entre les différentes phases trop important. Pour augmenter ce rapport, il serait peut-être utile de développer une méthode pour laquelle le conditionnement de la matrice ne dépendrait que de la taille de maille considérée [Bui 08]. Il est également possible de recourir à une viscosité bulle n'étant pas directement proportionnelle à la viscosité au centre de l'élément, ce qui a pour effet de réduire les oscillations dues au changement de viscosité à l'interface [Pha 12].

En ce qui concerne la prise en compte d'un comportement non linéaire, nous avons fait le choix d'une méthode très simple de point fixe, entièrement basée sur la résolution linéaire. L'utilisation de la méthode de Newton-Raphson serait sans doute plus efficace et plus robuste de manière générale. Pour cela, un meilleur couplage entre CIMLib et PETSc doit être implémenté.

Chapitre 3



- Chapitre 4 -

Déplacement d'interfaces

« Chaque science, chaque étude, a son jargon inintelligible, qui semble n'être inventé que pour en défendre les approches. »

Voltaire

1. Introduction	80
2. Etude bibliographique	81
3. La méthode Level Set dans CIMLib	84
3.1. Solveur de convection ConvectionP1	84
3.2. Solveur de réinitialisation ReinitLevelSet	87
3.3. Solveur de réinitialisation convective Leveller	90
3.4. Solveur de réinitialisation convective LevellerS	93
3.5. Solveur de réinitialisation convective LevellerT	95
4. La méthode PLS (Particle Level Set)	97
4.1. Convection de la level set	97
4.2. Création des particules	98
4.3. Déplacement des particules	99
4.4. Correction de la level set	100
4.5. Réinitialisation de la level set	100
4.6. Remarques	101
4.7. Effet de l'apport des particules	101
5. Comparaison des deux méthodes	102
5.1. Rotation d'un disque	102
5.2. Test du vortex	105
5.3. Conclusion	107
6. Test de l'inclusion	108
6.1. Solution numérique	109
6.2. Epaisseur de lissage	110
6.3. Remaillage	114
6.4. Maillage anisotrope	116
6.5. Plusieurs inclusions	120
6.6. Période de réinitialisation de GMRES	121
7. Conclusion	125

1. Introduction

Nous venons de détailler dans le chapitre précédent la méthode utilisée pour déterminer un champ de vitesse stationnaire qui satisfait aux conditions aux limites imposées, les équations de conservation de la masse et de la quantité de mouvement ainsi que les lois de comportement des différents milieux en présence. Notre but étant de simuler le comportement d'un échantillon au cours du temps, il est nécessaire de pouvoir déplacer les différents milieux durant une simulation, en fonction du champ de vitesse calculé. Nous allons décrire comment faire dans ce chapitre intitulé déplacement d'interfaces.

Avoir choisi un tel titre peut de prime abord sembler étrange. En effet, si nous supposons que chaque volume de matière élémentaire possède ses propres valeurs de propriétés physiques, il est nécessaire de résoudre une équation permettant le transport de ces valeurs sur l'ensemble du domaine de calcul et prenant en compte la vitesse de l'écoulement en tout nœud du maillage. Cependant, nous considérons ici que chaque milieu a des propriétés homogènes, ce qui limite nos besoins au seul déplacement des interfaces entre les différents milieux.

Comme expliqué précédemment, nous représentons une interface entre deux milieux de manière implicite avec une fonction level set, pour laquelle la valeur définie en chaque nœud du maillage représente la distance signée à l'interface. Le champ de vitesse est utilisé pour actualiser la level set dans un contexte eulérien, qui contrairement au contexte lagrangien, s'appuie sur l'utilisation d'un maillage fixe. Le calcul de la nouvelle fonction se fait en deux étapes. Il faut tout d'abord résoudre une équation de convection pour propager la level set à la vitesse définie sur l'ensemble du domaine. A la suite de cette résolution, la level set ne correspond plus à une distance signée loin de l'interface. En outre, comme la vitesse a seulement une signification physique sur l'interface, nous déterminons la solution d'une équation de réinitialisation, grâce à laquelle la level set redevient une distance signée, tout en conservant la position de l'interface inchangée.

Les équations de convection et de réinitialisation étant toutes deux de type hyperbolique, leur résolution peut être combinée pour former une équation de réinitialisation convective. De plus, la valeur de la level set n'étant utile que proche de l'interface, il est possible d'utiliser une level set lissée, qui n'est plus égale à la distance signée, pour optimiser la résolution. Au CEMEF, les solveurs de convection et de réinitialisation ont été implémentés dans CIMLib par Basset pendant sa thèse [Bas 06]. Des solveurs de réinitialisation convective, sans lissage [Bas 06], avec lissage en fonction sinus et en fonction tangente hyperbolique ont été codés par Coupez [Cou 06, Cou 07, Vil 11a].

Le déplacement d'interfaces étant un sujet très largement étudié, nous donnons dans un premier temps un aperçu de différentes méthodes utilisées dans la littérature. Nous décrivons ensuite la technique utilisée dans cette thèse et dans CIMLib. Nous expliquons également le fonctionnement de la méthode level set particulière, que nous abrégeons par PLS (Particle Level Set). L'analyse des résultats obtenus par les deux approches sur des cas test nous permet ainsi de juger plus objectivement des capacités réelles de la méthode employée dans CIMLib. Après cette comparaison, nous considérons de nouveau l'étude du test de l'inclusion afin de formuler un certain nombre d'observations sur les conditions d'utilisation de notre méthode. Pour conclure, nous donnons notre sentiment global sur la résolution CIMLib, en résumant les améliorations envisageables.

2. Etude bibliographique

Le déplacement d'interfaces peut-être utilisé partout où il est nécessaire de représenter une phase ou plus en mouvement. Il trouve ainsi des applications dans des domaines tels que la mécanique des fluides, la mécanique des solides, les changements de phases, ou encore le cinéma ou les jeux vidéo.

Il existe tellement de méthodes dans la littérature pour déplacer des interfaces qu'il serait trop long de toutes les évoquer. En outre, il n'est pas facile de déterminer rapidement l'ensemble des caractéristiques pour chacune. Connaître les limites, les avantages, les conditions d'application, la difficulté d'implémentation, la vitesse de calcul ou les perspectives d'évolution demande en effet pour chaque méthode une longue manipulation que nous n'avons malheureusement pas eu le temps d'effectuer.

Cependant, deux mesures concrètes sont prises pour aider les chercheurs à choisir plus rapidement une méthode adaptée à leurs besoins. Tout d'abord, nous notons dans la littérature l'utilisation de nombreux cas test pour lesquels une solution analytique est définie. Ainsi, réaliser des simulations pour comparer ces solutions analytiques à des résultats numériques est actuellement une étape obligatoire dans le développement de toute nouvelle approche. Cela permet, outre la validation de sa méthode, d'avoir un regard critique sur ses performances en la situant par rapport aux autres méthodes.

Toutefois, les éléments de comparaison sont parfois trop peu nombreux dans la bibliographie et il devient alors utile de construire soi-même des applications avec d'autres techniques de déplacement d'interfaces afin de pouvoir positionner correctement sa propre approche. Pour nous aider dans ce travail, nous pouvons profiter des codes disponibles gratuitement sur Internet comme par exemple la boîte à outils de méthodes level set de Mitchell [Mit 08], la librairie de Mokheri & Faloutsos [Mok 06] ou encore le logiciel CLAWPACK développé par Leveque [Lev 96]. Grâce à cette deuxième mesure, utiliser une technique totalement inconnue devient beaucoup plus facile et il est possible d'établir des comparaisons pertinentes sans avoir besoin de maîtriser tous les aspects de chaque méthode, à condition de faire confiance au contenu des outils proposés et à leur description.

Pour donner un aperçu des différentes approches, il est plus commode de les classer par catégorie. Un des critères de classification les plus pertinents est le mode de représentation de l'interface. Ce dernier est soit explicite, soit implicite.

Pour une représentation explicite, le contour du maillage et l'interface sont superposés. Le maillage est alors déplacé avec la vitesse de l'écoulement dans un contexte lagrangien, de telle sorte que son contour est toujours superposé avec l'interface. Ce type de méthode, connu sous le nom de suivi d'interface ou suivi de front (interface tracking ou front tracking), est très avantageux pour imposer directement une condition aux limites sur l'interface. Toutefois, puisqu'il faut calculer un nouveau maillage à chaque pas de temps, il n'est pas trivial de développer une procédure pouvant rendre compte de déplacements complexes de l'interface, spécialement en trois dimensions. Pour obtenir plus de détails sur ce type de méthode, nous renvoyons le lecteur intéressé à l'analyse des travaux de Tryggvason et de ses collaborateurs sur la croissance dendritique [Jur 96, Alr 02, Alr 04].

Pour une représentation explicite de l'interface, nous pouvons utiliser une fonction scalaire définie sur l'ensemble du domaine de calcul. Selon que la fonction est définie aux nœuds ou aux éléments, la méthode s'intitule level set (ligne de niveau) ou VOF (volume of fluid ou volume de fluide en français). Pour la méthode VOF, généralement attribuée à Hirt & Nichols [Hir 81], la fonction représente sur chaque élément la fraction volumique d'une phase. Ainsi, dans une simulation où les deux phases en présence sont de l'eau et de l'air, un élément liquide a une fraction volumique de 1, un élément gazeux une fraction volumique de 0 et un

élément situé à l'interface une fraction volumique intermédiaire. Nous notons qu'utiliser une fonction définie seulement aux éléments empêche une reconstruction très précise de l'interface. Pour la méthode level set, introduite par Osher & Sethian en 1988 [Osh 88], la fonction représente la distance signée à l'interface. La fonction level set est donc positive dans une phase, négative dans l'autre et son isovaleur zéro nous donne la position de l'interface. Un des avantages de cette méthode est la simplicité du calcul de la normale à l'interface et de sa courbure [Jon 06].

Il est également envisageable de représenter l'interface de manière implicite avec des particules. C'est la méthode MAC (marker and cell) décrite par McKee et al. [Mck 08]. Dans la méthode MAC, des points matériels ou particules sont placés près de l'interface et permettent sa reconstruction. La vitesse calculée sur une grille eulérienne est alors interpolée pour chaque particule, permettant leur déplacement et par extension le déplacement de l'interface associée.

A la limite entre les modes de représentation implicite et explicite, nous pouvons trouver la méthode développée par Bargteil & Strain [Bar 06a, Bar 06b], connue sous le nom de contour semi-lagrangien (semi-Lagrangian contouring). Elle consiste à utiliser un maillage surfacique qui définit explicitement la surface en combinaison avec un maillage eulérien. Le maillage surfacique permet à chaque pas de temps de calculer une distance signée à l'interface sur la grille eulérienne, laquelle est convectée avec la méthode semi-lagrangienne de Strain [Str 01] et utilisée pour reconstruire le maillage surfacique. Nous signalons que Wojtan & Thürey utilisent une méthode assez proche de celle de Bargteil et en profitent pour replacer leurs travaux dans la littérature de manière pertinente [Woj 09, Thü 10].

Un autre moyen de classer les méthodes est le formalisme utilisé pour déplacer les interfaces, même si cela peut grandement recouper le classement fait suivant le mode de représentation de l'interface. Nous avons déjà dit que le suivi de front utilise un contexte lagrangien, mais une formulation arbitrairement lagrangienne eulérienne (ALE) est possible [Jim 04]. Au contraire, même si des exceptions existent [Res 10], les méthodes VOF et level set s'inscrivent en principe dans un contexte purement eulérien. Elles nécessitent par conséquent de résoudre une équation de convection décrivant l'évolution des fonctions associées. Cette équation dérive directement du principe de conservation de la masse pour la méthode VOF, ce qui permet de maintenir le volume de chaque phase constant dans des simulations incompressibles. En revanche, nous assistons à une diffusion de la fraction volumique [Bas 06], qui complique encore davantage la reconstruction de l'interface. Pour la méthode level set, la résolution de l'équation de convection entraîne également une diffusion numérique, qui se caractérise cette fois-ci par une non conservation du volume. En outre, la fonction, initialement égale à une distance signée, n'en est plus une après avoir été convectée, à moins de s'être servi d'une extension performante du champ de vitesse [Ada 99].

Comme expliqué en introduction, il est alors nécessaire de procéder à une réinitialisation. Hysing & Turek [Hys 05] donnent une étude comparative de plusieurs méthodes courantes de réinitialisation sur des maillages formés de quadrilatères, parmi lesquelles figurent la méthode marche rapide (fast marching) développée par Sethian [Set 96, Set 99] et une méthode qu'ils qualifient de brutale (brute force redistancing). Cette dernière, qui consiste simplement à extraire l'interface comme une suite de segments (une suite de facettes triangulaires en 3D), à partir desquels la distance minimale est calculée sur les nœuds de la grille, entraîne des temps de calcul importants. Hysing & Turek arrivent à la conclusion que la méthode marche rapide est la plus performante en terme de vitesse, précision et robustesse. Ils évoquent également la méthode de réinitialisation impliquant la résolution d'une équation hyperbolique aux dérivées partielles [Sus 94a, Cou 07].

Pour avoir une bonne idée sur la variété des schémas numériques utilisés pour résoudre cette équation de réinitialisation, mais aussi l'équation de convection, nous recommandons la lecture de l'ouvrage d'Osher & Fedkiw [Osh 03], qui est de surcroît une bonne introduction aux méthodes level set. Nous tenons néanmoins à souligner que les schémas présentés s'appliquent généralement à des maillages structurés constitués d'éléments carrés en deux dimensions et cubiques en trois dimensions. Ce type de maillage, très simple à construire, est d'ailleurs considéré dans une grande partie des travaux sur la méthode level set.

Certains auteurs ont pris conscience de la nécessité de proposer des solutions adaptées à des maillages de type différent, qui peuvent dans certains cas présenter des avantages importants. Ainsi, Barth & Sethian, le co-inventeur de la méthode level set, proposent un schéma numérique fonctionnant sur des maillages triangulaires [Bar 98]. Frey et son équipe de recherche vont même plus loin puisqu'ils utilisent une méthode level set en combinaison avec des maillages triangulaires ou tétraédriques anisotropes [Duc 08, Bui 10, Bui 11], avec lesquels ils parviennent à représenter précisément les interfaces de leurs objets tout en limitant grandement le nombre d'éléments. Des maillages tétraédriques anisotropes sont également utilisés au CEMEF dans la librairie CIMLib [Mes 08, Cou 11] mais les publications de ce laboratoire ne donnent pour l'instant pas de résultats sur des cas analytiques 3D avec ce type de maillage, contrairement aux travaux de Bui, Dapogny & Frey [Bui 10].

Les scientifiques utilisant des maillages non structurés ont bien sûr pour objectif de réduire au maximum la diffusion et la perte de matière engendrée par l'utilisation d'une méthode level set en minimisant les temps de calcul. Mais il existe d'autres moyens pour cela. Heo & Ko introduisent par exemple une approche purement eulérienne très conservative en reconstruisant la distance signée avec un polynôme d'ordre élevé près de l'interface [Heo 10]. Certains auteurs, quant à eux, préconisent un couplage entre plusieurs méthodes, profitant des avantages de chacune tout en atténuant ses défauts par l'apport de l'autre méthode.

C'est le cas de Sussman & Puckett [Sus 00], qui font des comparaisons entre des résultats de simulations obtenus par la méthode VOF, par la méthode level set et par une méthode hybride qu'ils intitulent méthode CLSVOF (coupled level set and volume of fluid). Cette nouvelle méthode est généralement supérieure aux deux méthodes découplées et elle est à présent utilisée dans de nombreux travaux [Men 07, Nic 10, Yan 06, Lv 10].

En parallèle, Enright et al. présentent un couplage entre une approche particulière et une méthode level set, nommée PLS (particle level set) [Fos 01, Enr 02a, Enr 02b, Enr 02c, Enr 05]. Ils utilisent la level set pour sa capacité à prendre en compte les modifications topologiques complexes de l'interface et les particules pour reconstruire une fonction level set donnant une meilleure conservation. Cette reconstruction fonctionne d'ailleurs tellement bien qu'ils peuvent aisément ne considérer qu'un schéma numérique d'ordre faible pour convecter et réinitialiser la level set [Enr 05]. Le temps de calcul économisé de la sorte compense très largement la légère baisse de précision par rapport à un schéma numérique plus précis [Enr 02c]. Une très grande émulation existe autour de cette méthode, pour laquelle de nombreuses améliorations sont proposées [Hie 05, Gau 08, Los 04, Los 07, Pig 05, Wan 09]. De plus, les chercheurs intéressés par la méthode PLS peuvent télécharger la librairie mise à disposition par Mokheri & Faloutsos [Mok 06], afin d'accélérer leur compréhension et de comparer les résultats obtenus à leurs propres codes [Cun 08a, Cun 08b, Mei 08].

Dans la méthode PLS, les particules sont situées dans les cellules autour de l'interface. Mihalef et al. remarquent que ce n'est pas une nécessité absolue et ils obtiennent de meilleurs résultats en introduisant la méthode marker level set (MLS), qui consiste à placer directement les particules sur l'interface [Mih 06, Mih 07, Mih 08, Mih 09]. Pour encore améliorer cette méthode, Elmi propose d'utiliser des marqueurs dirigés [Elm 11]. Cela lui permet de réduire le nombre de marqueurs puisqu'il n'en utilise plus qu'un au maximum par élément. Les

résultats qu'il obtient sont comparables avec ceux donnés par la méthode MLS. Cependant, aucun résultat en 3D ou avec un champ de vitesse calculé par un solveur mécanique n'est présenté. Il faut aussi noter que ce travail a juste été soumis pour publication à l'heure actuelle.

3. La méthode Level Set dans CIMLib

Dans cette partie, nous décrivons l'approche level set implémentée dans la librairie CIMLib. Pour obtenir une description la plus exhaustive possible, nous orientons le lecteur vers l'étude des documents suivants : [Bas 06, Cou 06, Cou 07, Vil 11a, Vil 11b].

Ayant fait la constatation qu'il est parfois bien difficile de relier les équations énoncées dans une thèse aux lignes de code déployées dans un logiciel de calcul, nous n'hésitons pas à employer l'appellation locale des différents solveurs que nous nous apprêtons à décrire. Nous nous en excusons par avance auprès des lecteurs extérieurs à CIMLib.

Nous tenons en outre à signaler que certains détails de la méthode des éléments finis déjà évoqués dans le chapitre traitant de la résolution mécanique sont volontairement passés sous silence.

3.1. Solveur de convection ConvectionP1

Nous rappelons la formulation forte de l'équation de convection, en notation eulérienne [Bas 06, Osh 01], avec ϕ la fonction level set et \vec{v} la vitesse :

$$\frac{\partial \phi}{\partial t} + \vec{v} \cdot \vec{\nabla} \phi = 0$$

Dans notre méthode, aucune condition aux limites n'est imposée sur les bords.

En multipliant cette équation par une fonction test χ et en l'intégrant sur l'ensemble du domaine de calcul Ω , nous obtenons la formulation faible donnée ci-dessous :

$$\int_{\Omega} \frac{\partial \phi}{\partial t} \chi \, dV + \int_{\Omega} \vec{v} \cdot \vec{\nabla} \phi \chi \, dV = 0$$

En utilisant le même type de notation que dans le chapitre précédent, la formulation faible discrète est alors la suivante :

$$\int_{\Omega} \frac{\partial \phi_h}{\partial t} \chi_h \, dV + \int_{\Omega} \vec{v} \cdot \vec{\nabla} \phi_h \chi_h \, dV = 0$$

En choisissant un schéma d'Euler implicite, la discrétisation temporelle s'écrit tout simplement [Bas 06] :

$$\frac{\partial \phi_h}{\partial t} = \frac{\phi_h - \phi_h^-}{\Delta t}$$

où Δt est le pas de temps choisi pour la simulation et ϕ_h^- la valeur de la level set au pas de temps précédent.

Une recommandation classique pour le choix du pas de temps est le respect de la condition CFL (Courant-Friedrichs-Lewy [Cou 67]), qui impose un pas de temps inférieur ou égal à la taille de maille divisée par la norme de la vitesse, afin de ne pas franchir plus d'un élément par incrément.

Comme Basset [Bas 06], nous utilisons un schéma semi-implicite en temps pour la convection, pour lequel le terme $\vec{v} \cdot \vec{\nabla} \phi$ est calculé en utilisant à parts égales les valeurs du gradient au pas de temps précédent et au pas de temps actuel. En mettant au second membre les termes dépendant de la fonction level set au pas de temps précédent, qui sont connus, nous obtenons :

$$\int_{\Omega} \frac{\phi_h}{\Delta t} \chi_h \, dV + \frac{1}{2} \int_{\Omega} \vec{v} \cdot \vec{\nabla} \phi_h \chi_h \, dV = \int_{\Omega} \frac{\phi_h^-}{\Delta t} \chi_h \, dV - \frac{1}{2} \int_{\Omega} \vec{v} \cdot \vec{\nabla} \phi_h^- \chi_h \, dV .$$

Résoudre cette équation de convection par la méthode de Galerkin standard, qui consiste à prendre pour fonctions test les fonctions d'interpolation nodales N_n de valeur unitaire sur le nœud n et de valeur nulle sur les autres nœuds de l'élément, entraîne la présence d'oscillations numériques dans la solution [Bas 06]. Nous utilisons par conséquent la méthode SUPG (Streamline Upwind/Petrov-Galerkin), introduite par Brooks & Hughes [Bro 82], pour laquelle une nouvelle fonction test \tilde{N}_n est employée. Cette fonction test s'écrit :

$$\tilde{N}_n = N_n + \tau \vec{v} \cdot \vec{\nabla} N_n$$

avec τ un taux de stabilisation qui dépend pour chaque élément de sa taille dans le sens de l'écoulement h_e et de la vitesse en son centre \vec{v} , tel que :

$$\tau = \frac{h_e}{2 \|\vec{v}\|} \quad \text{avec} \quad h_e = \frac{2 \|\vec{v}\|}{\sum_{n=1}^{d+1} |\vec{v} \cdot \vec{\nabla} N_n|} \quad [\text{Tez 86, Tez 03}]$$

avec d la dimension du problème.

En utilisant la fonction test SUPG, la formulation faible discrétisée devient :

$$\forall n, \quad \int_{\Omega} \frac{\phi_h}{\Delta t} \tilde{N}_n \, dV + \frac{1}{2} \int_{\Omega} \vec{v} \cdot \vec{\nabla} \phi_h \tilde{N}_n \, dV = \int_{\Omega} \frac{\phi_h^-}{\Delta t} \tilde{N}_n \, dV - \frac{1}{2} \int_{\Omega} \vec{v} \cdot \vec{\nabla} \phi_h^- \tilde{N}_n \, dV .$$

La matrice et le second membre locaux pour l'élément T sont donnés ci-dessous :

$$A_{nm} = \int_T \frac{N_m \tilde{N}_n}{\Delta t} \, dV + \frac{1}{2} \int_T \vec{v} \cdot \vec{\nabla} N_m \tilde{N}_n \, dV ; \quad B_n = \int_T \frac{\phi_h^-}{\Delta t} \tilde{N}_n \, dV - \frac{1}{2} \int_T \vec{v} \cdot \vec{\nabla} \phi_h^- \tilde{N}_n \, dV .$$

Pour intégrer ces polynômes du deuxième ordre, il nous faut $d + 1$ points d'interpolation, dont les coordonnées sont données dans de nombreux ouvrages sur la méthode des éléments finis, comme celui de Dhatt & Touzot [Dha 84].

Après l'étape d'assemblage, nous obtenons un système linéaire pour lequel nous avons un nombre d'inconnues égal au nombre de nœuds du maillage. De par la présence d'un terme de convection, la matrice de ce système est non symétrique. Aucune condition aux limites n'est ajoutée. Pour la résolution du système linéaire, nous utilisons la méthode de généralisation de la méthode de minimisation du résidu (GMRES) disponible dans la librairie PETSc [Bal 10] et inventée par Saad & Schultz [Saa 86]. La méthode utilisée est plus exactement la méthode GMRES (30) où la valeur 30 représente la période de réinitialisation de la base de vecteurs utilisée dans la résolution. Nous conseillons au lecteur voulant se familiariser avec cette méthode de résolution itérative le livre de Quarteroni et al., dans lequel il pourra trouver une implémentation Matlab élémentaire de GMRES [Qua 00].

La résolution itérative est combinée avec une méthode de préconditionnement de Jacobi par bloc [Saa 03]. Tout comme Basset dans sa thèse [Bas 06], nous avons utilisé des blocs de 1 en 3D et des blocs de 4 en 2D.

Pour toutes les simulations effectuées dans ce chapitre, nous utilisons une précision non relaxée $P = 10^{-8}$ pour la résolution des systèmes linéaires. Le nombre d'itérations est en général très faible et la précision demandée est donc atteinte rapidement. Cependant, ce n'est pas toujours le cas et nous présentons dans la sous-partie 6.6 de ce chapitre un exemple qui montre que la résolution des systèmes linéaires peut être améliorée pour nos solveurs d'interfaces.

Pour illustrer le fonctionnement du solveur de convection, nous étudions l'exemple très simple suivant. Nous considérons un domaine rectangulaire de longueur 10 m et de hauteur 1 m. L'interface entre les deux milieux se situe sur le segment d'abscisse 1 m, elle est donc parallèle à l'axe des ordonnées. La fonction level set est alors donnée par $\phi = x - 1$. Le champ de vitesse est constant, avec $v_x = x$ et $v_y = 0$.

Nous prenons une taille de maille de 10 cm, un pas de temps de 10 ms et un temps final de 1 s. Nous plaçons des capteurs sur la ligne d'ordonnée 0,5 m pour différentes abscisses. La figure 4.1 montre le maillage, ainsi que la position de l'interface initiale et celle des capteurs tandis que la figure 4.2 montre l'évolution de la fonction level set sur ces différents capteurs au cours du temps, ainsi que la solution analytique. Cette figure nous donne l'évolution attendue, pour laquelle $\phi(x, t) = x \exp(-t) - 1$. L'interface est située sur la droite d'équation $x = 2,718$ m à la fin du calcul, qui est aussi le résultat analytique, donné par $\exp(t)$.

Comme le montre la figure 4.2, les différentes lignes de niveau sont de plus en plus resserrées au fil du temps. En effet avec la vitesse considérée, la valeur analytique du gradient suivant Ox au cours du temps est donnée par $d\phi/dx = \exp(-t)$ (le gradient suivant Oy étant nul).

Ainsi au début de la résolution, la norme du gradient est unitaire sur tout le domaine. Après un temps de résolution de 1 s, elle a une valeur de 0,368, égale à la solution analytique.

La norme du gradient n'étant plus unitaire, la fonction level set ne représente plus une distance signée à l'interface. Il est alors nécessaire de la réinitialiser. C'est l'objet de la sous-partie suivante.

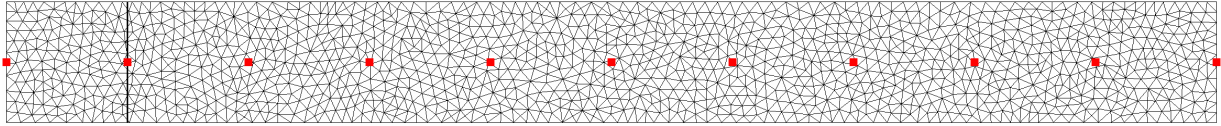


Figure 4.1. Domaine de calcul, maillage et capteurs associés. L'interface est représentée par le trait noir et les capteurs par les carrés rouges.

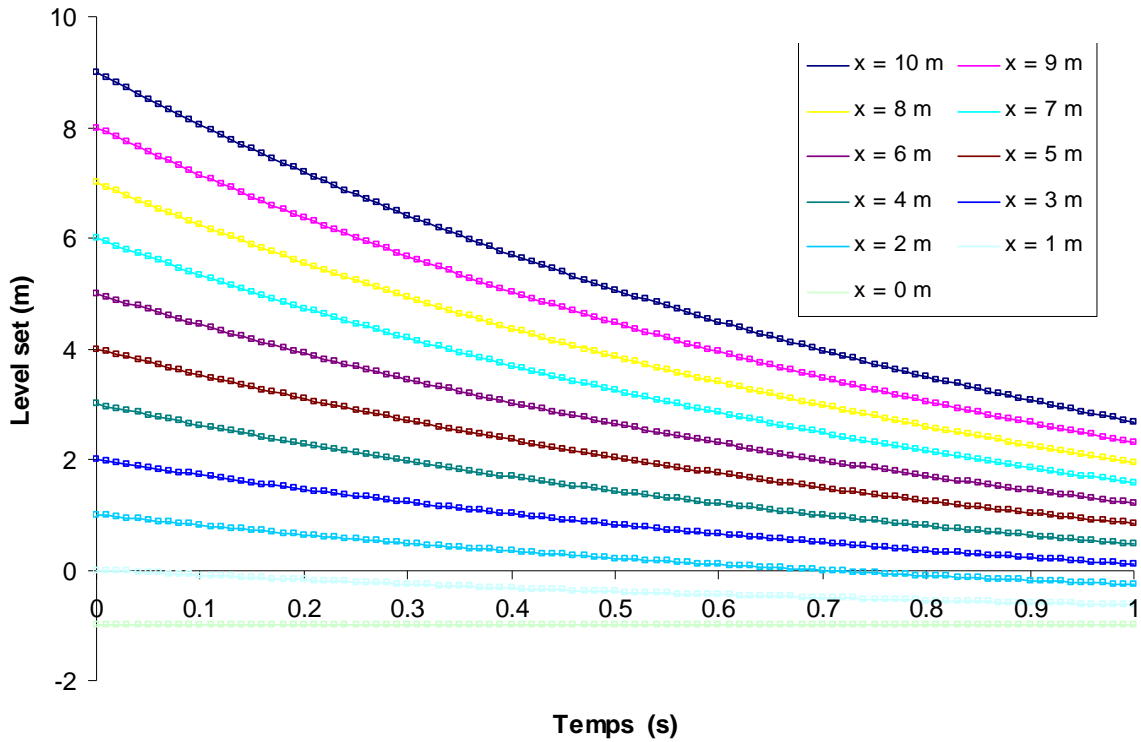


Figure 4.2. Evolution de la fonction level set au cours du temps lors de la résolution de l'équation de convection pour différents capteurs. Les carrés représentent les résultats numériques et les traits continus la solution analytique.

3.2. Solveur de réinitialisation ReinitLevelSet

Pour l'algorithme de réinitialisation, Basset a employé la méthode présentée par Sussman et al. [Sus 94a, Sus 94b], qui consiste à résoudre une équation aux dérivées partielles, construite à partir de la propriété de norme de gradient unitaire de la level set. La démonstration de cette propriété est donnée par Osher & Fedkiw [Osh 03].

Nous introduisons une fonction signe S tel que :

$$S(\phi) = \begin{cases} -1 & \text{si } \phi < 0 \\ 1 & \text{si } \phi > 0 \\ 0 & \text{si } \phi = 0 \end{cases}$$

L'équation de réinitialisation s'écrit alors :

$$\frac{\partial \phi}{\partial \tau} = S(\phi)(1 - \|\bar{\nabla} \phi\|)$$

avec τ homogène à une distance et appelé temps fictif par abus de langage [Cou 06].

En résolvant cette équation, la fonction level set est modifiée partout où la norme de son gradient n'est pas unitaire. La présence de la fonction signe permet de garder la fonction level set inchangée au niveau de l'interface et de savoir de quel côté de l'interface chaque point est situé, afin de retrouver une distance signée.

La fonction signe n'étant pas différentiable en $\phi = 0$, il est préférable d'utiliser une fonction signe lissée, comme conseillé par Peng et al [Pen 99]. Nous avons [Bas 06] :

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + \|\bar{\nabla} \phi\|^2 h^2}}$$

avec h la taille de l'élément.

En l'état actuel des choses, l'équation de réinitialisation est une équation non linéaire [Sus 94a]. Basset la linéarise en introduisant le gradient à l'incrément fictif précédent et en approximant le signe de la level à l'incrément précédent par le signe à l'incrément actuel :

$$\|\bar{\nabla} \phi\| = \frac{\|\bar{\nabla} \phi\|^2}{\|\bar{\nabla} \phi\|} \approx \frac{\bar{\nabla} \phi \cdot \bar{\nabla} \phi^-}{\|\bar{\nabla} \phi^-\|}.$$

L'équation de réinitialisation devient alors $\frac{\partial \phi}{\partial \tau} + S(\phi^-) \frac{\bar{\nabla} \phi^- \cdot \bar{\nabla} \phi}{\|\bar{\nabla} \phi^-\|} \bar{\nabla} \phi = S(\phi^-)$.

Ceci nous permet de définir une vitesse de réinitialisation $\bar{w} = S(\phi^-) \frac{\bar{\nabla} \phi^-}{\|\bar{\nabla} \phi^-\|}$, grâce à laquelle

nous avons finalement $\frac{\partial \phi}{\partial \tau} + \bar{w} \cdot \bar{\nabla} \phi = S(\phi^-)$.

La vitesse de réinitialisation et le signe étant connus, nous obtenons une équation de réinitialisation similaire à l'équation de convection. Ainsi, nous utilisons pour la résoudre les mêmes outils que pour l'équation de convection, à savoir une méthode éléments finis SUPG avec un schéma en temps semi-implicite et un préconditionnement de Jacobi par bloc combiné à la méthode GMRES pour la résolution du système linéaire.

La vitesse de réinitialisation ayant une norme unitaire, résoudre une fois l'équation a pour effet de transformer la level set en une distance signée sur une épaisseur égale au pas de temps fictif considéré $\Delta \tau$ (lui aussi homogène à une distance). Pour obtenir une norme unitaire sur une épaisseur plus importante, nous devons donc résoudre plusieurs fois l'équation de réinitialisation. En ce qui concerne le choix du pas de temps fictif, Coupez préconise une valeur égale ou inférieure à la taille de maille [Cou 07].

Reprenons l'exemple précédent là où nous l'avons laissé (après une convection d'une seconde) pour illustrer le fonctionnement du solveur de réinitialisation. Nous utilisons un pas de temps fictif de 50 mm, égal à la moitié de la taille de maille, et un temps fictif final de 10 m. Sur la figure 4.3, nous observons l'évolution de la level set au cours du temps fictif sur les mêmes capteurs que précédemment. Nous constatons que plus le capteur est situé loin de l'interface, plus la réinitialisation prend de temps. En effet les droites de la figure 4.3 associées aux capteurs voisins de l'interface (par exemple pour $x = 3\text{m}$), deviennent rapidement parallèles à l'axe des abscisses ; la level set n'y est plus modifiée et la réinitialisation est donc terminée. En outre, l'évolution temporelle de ϕ est identique au signe près dans les zones en cours de réinitialisation. En effet, le gradient de la level set étant nul suivant Oy , nous avons donc :

$$\bar{w} = S(\phi^-) \frac{\bar{\nabla}\phi^-}{\|\bar{\nabla}\phi^-\|} = \begin{pmatrix} S(\phi^-) \\ 0 \end{pmatrix}$$

L'équation de réinitialisation nous donne par conséquent $\frac{\partial\phi}{\partial\tau} = S(\phi^-) \left(1 - \frac{\partial\phi}{\partial x}\right) = \pm 0,632$. Cette valeur est obtenue car $\partial\phi/\partial x = 0,368$ après l'étape de convection.

La figure 4.4 permet de réaliser de manière claire que la réinitialisation se fait à partir de l'interface. Elle représente la fonction level set ainsi que son gradient à différents temps.

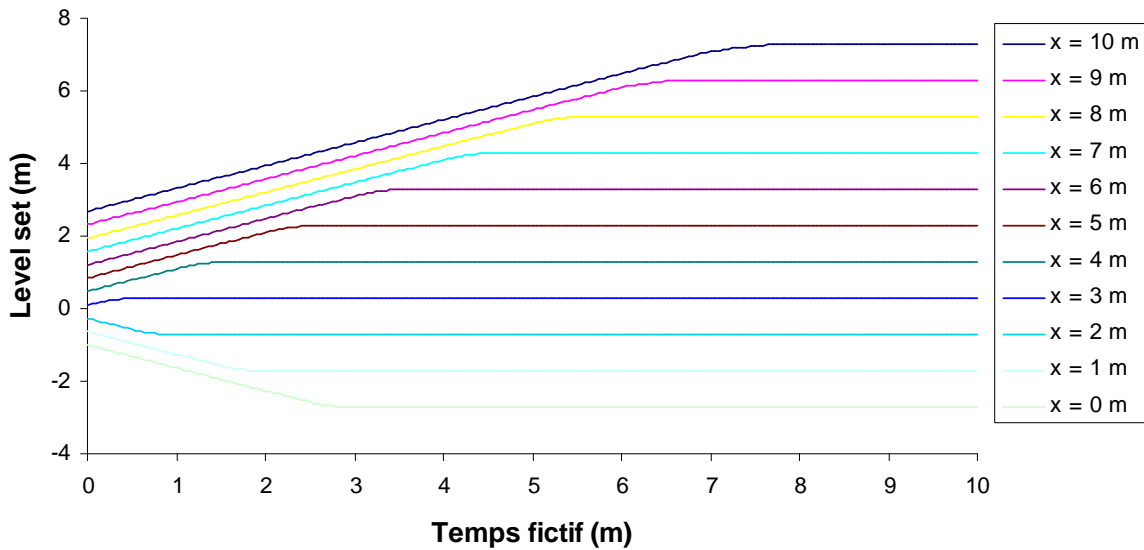


Figure 4.3. Evolution de la fonction level set au cours du temps lors de la résolution de l'équation de réinitialisation pour différents capteurs.

Il faut signaler que réinitialiser une fonction level set en utilisant une équation aux dérivées partielles entraîne un très léger déplacement de l'interface. Ce déplacement minime à chaque réinitialisation peut devenir important et donc préjudiciable à la qualité de la simulation si de nombreuses réinitialisations sont nécessaires.

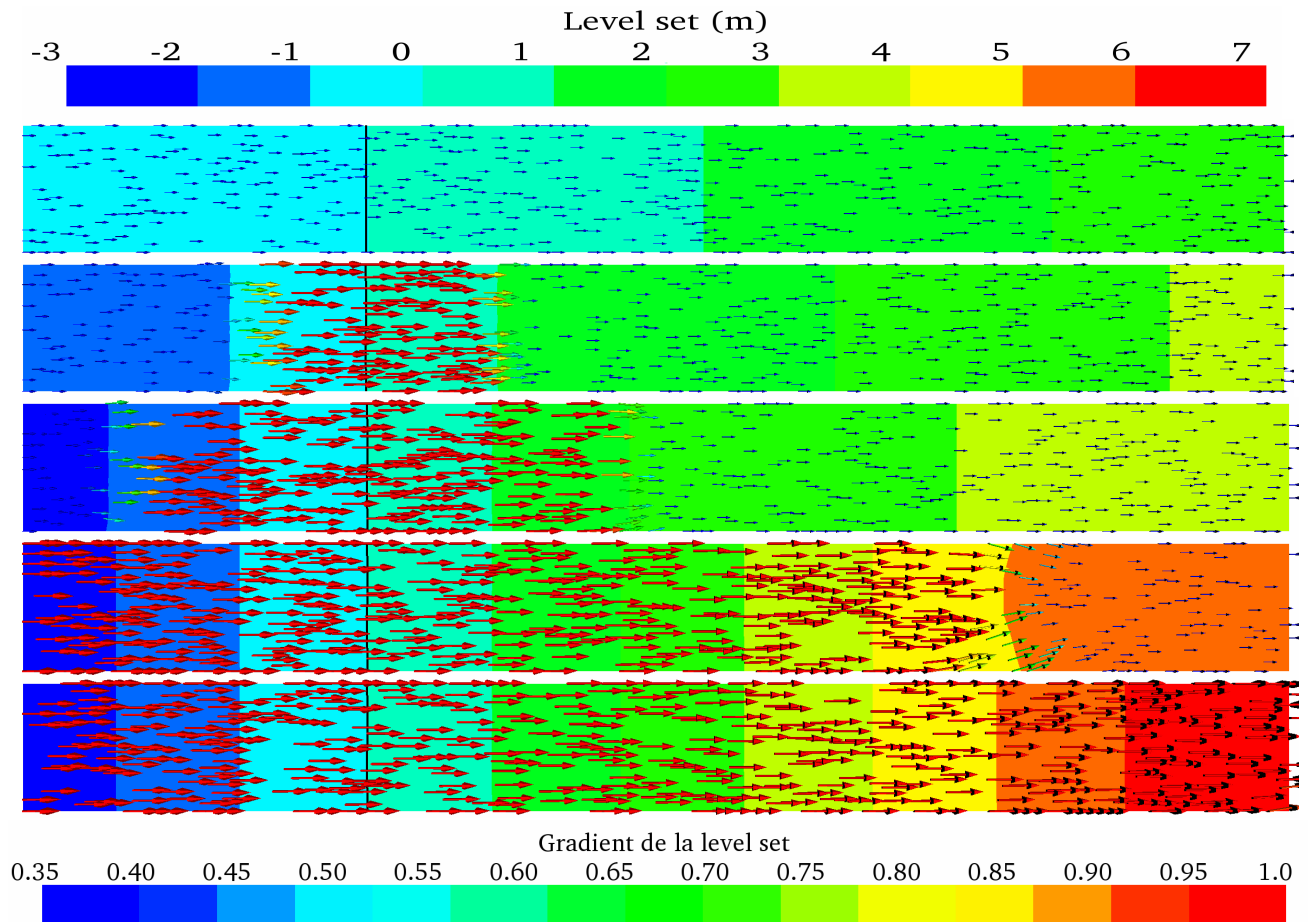


Figure 4.4. Fonction level set et gradient de la level set pour des temps fictifs respectifs de 0, 1, 2, 5 & 8 m. La couleur représente la level set tandis que les flèches représentent son gradient.

3.3. Solveur de réinitialisation convective Leveller

Pour résoudre correctement un problème impliquant le transport d'une fonction level set, il est préférable de toujours travailler avec une distance signée. Ainsi, la réinitialisation doit être utilisée plus souvent que sur notre exemple précédent. Comme cela occasionne un surcoût de calcul et des déplacements non souhaitables de l'interface, Basset suggère de ne pas réinitialiser à chaque incrément. Il propose un critère basé sur la dégradation de la norme du gradient que nous sommes prêts à accepter pour déterminer quand la réinitialisation doit s'opérer [Bas 06].

Une approche qui semble plus élégante est envisagée par Coupeuz. Les équations de convection et de réinitialisation étant du même type, il a montré qu'il était possible de coupler leur résolution en une seule et même étape [Cou 06, Cou 07]. Cette approche a d'ailleurs été reprise par Ville et al. [Vil 11a, Vil 11b], qui introduisent le paramètre λ dans l'égalité suivante :

$$\frac{\partial \phi}{\partial \tau} = \frac{\partial \phi}{\partial t} \frac{\partial t}{\partial \tau} = \frac{1}{\lambda} \frac{\partial \phi}{\partial t}$$

Chapitre 4

Par la suite le paramètre λ est pris égal à la taille de maille divisée par le pas de temps, comme dans le papier de Coupez [Cou 06].

Ceci permet d'écrire l'équation de réinitialisation en faisant apparaître la dérivée de la level set par rapport au temps réel (et non plus par rapport à un temps fictif) de la façon qui suit :

$$\frac{\partial \phi}{\partial t} + \lambda \bar{w} \cdot \bar{\nabla} \phi = \lambda S(\phi^-)$$

Coupez et al. obtiennent alors une équation de réinitialisation convective en rajoutant à l'équation précédente la contribution apportée par la vitesse de convection [Bas 06, Cou 06, Cou 07, Vil 11a, Vil 11b] :

$$\frac{\partial \phi}{\partial t} + (\bar{v} + \lambda \bar{w}) \cdot \bar{\nabla} \phi = \lambda S(\phi^-).$$

En résolvant cette équation, le but de Coupez n'est pas d'obtenir à chaque incrément une fonction level set égale à une distance signée sur l'ensemble du domaine, mais seulement sur une zone suffisamment large autour de l'interface pour que l'équation de convection soit résolue correctement. Ainsi, Basset explique que l'application d'une troncature de la fonction level set initiale apporte une stabilisation numérique [Bas 06]. Cette troncature consiste à définir une valeur maximale E pour la valeur absolue de la level set, en utilisant $|\phi| < E$.

Dans cette configuration, l'expression suivante est utilisée pour la fonction signe :

$$S(\phi) = \frac{\phi}{|\phi| + \rho h + \|\bar{\nabla} \phi\|^2 h^2}$$

avec ρ un paramètre ajustable relié à l'épaisseur sur laquelle nous lisons le signe, dont la valeur par défaut est fixée à 1. C'est cette valeur par défaut que nous utilisons.

Nous tenons à signaler que cette équation correspond au code du solveur Leveler, même si elle n'est mentionnée dans aucun papier. Nous n'avons par conséquent trouvé aucune justification en faveur de cette expression, d'autant plus que les monômes du dénominateur ne sont pas tous homogènes à une distance.

Puisque la troncature de la level set entraîne un gradient nul loin de l'interface, la vitesse de réinitialisation est définie en utilisant une norme du gradient régularisée [Cou 06], pour laquelle un paramètre ε (de valeur 0,5 par défaut) est introduit :

$$\bar{w} = S(\phi^-) \frac{\bar{\nabla} \phi^-}{\|\bar{\nabla} \phi^-\|_r} \quad \text{avec} \quad \|\bar{\nabla} \phi\|_r = \varepsilon + (1 - \varepsilon) \|\bar{\nabla} \phi\|$$

En utilisant la norme du gradient régularisée, nous changeons la norme de \bar{w} , qui doit être unitaire (en supposant que le signe l'est). Pour cette raison, nous modifions l'équation de réinitialisation en multipliant son second membre par $\|\bar{\nabla} \phi^-\| / \|\bar{\nabla} \phi^-\|_r$. L'équation finale à résoudre, qui correspond au solveur Leveler de CIMLib, rapidement introduit par Basset [Bas 06], est la suivante :

$$\frac{\partial \phi}{\partial t} + \left(\bar{v} + \lambda S(\phi^-) \frac{\bar{\nabla} \phi^-}{\|\bar{\nabla} \phi^-\|_r} \right) \cdot \bar{\nabla} \phi = \lambda S(\phi^-) \frac{\|\bar{\nabla} \phi^-\|}{\|\bar{\nabla} \phi^-\|_r}$$

Nous résolvons cette équation pour l'exemple introduit pour le solveur de convection. Le pas de temps choisi est de 5 ms. Nous lançons deux simulations avec $E = \infty$ et $E = 1$ m. La figure 4.5 montre les résultats obtenus pour les deux simulations à différents temps.

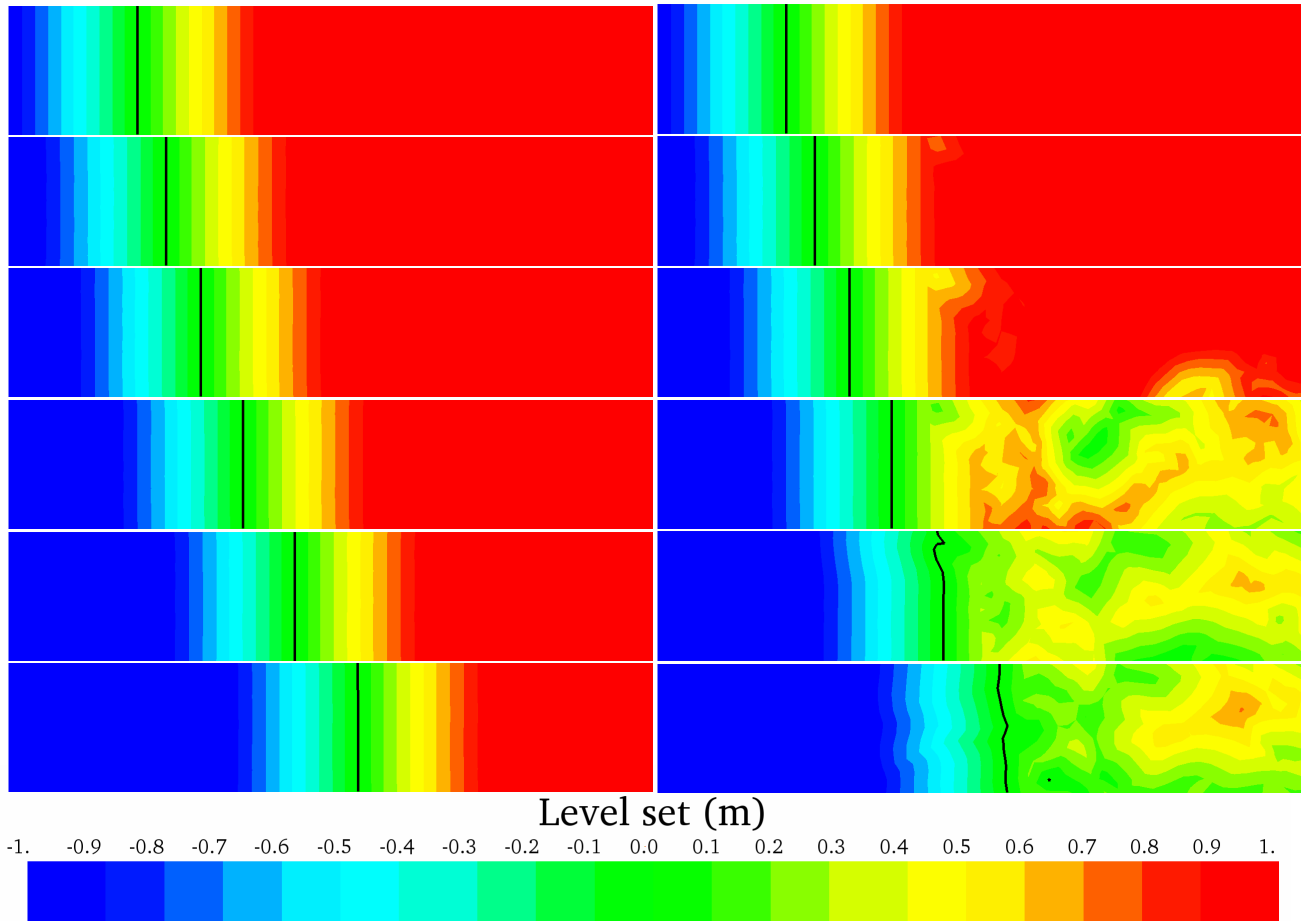


Figure 4.5. Fonction level set à différents temps. A gauche, $E = \infty$. A droite, $E = 1$ m. Les temps, de haut en bas sont les suivants : 0 ; 0,2 ; 0,4 ; 0,6 ; 0,8 et 1 s.

Lorsque la level set est tronquée, des instabilités apparaissent loin de l'interface. Au bout d'un certain temps, ces instabilités perturbent le déplacement de l'interface, ce qui compromet la simulation. Au contraire, sans troncature, la fonction level set est proche d'une fonction distance dans le voisinage de l'interface.

Cette figure est en réalité très dépendante de l'échelle de couleur utilisée pour représenter la fonction level set. En effet, si nous utilisons une échelle dont les bornes sont à chaque incrément les valeurs maximale et minimale de la level set, nous voyons beaucoup plus d'instabilités, y compris dans le calcul sans troncature.

3.4. Solveur de réinitialisation convective LevellerS

D'après Coupez, il existe deux moyens qui permettent d'améliorer le solveur de réinitialisation convective [Cou 06]. Il est dans un premier temps possible de ne prendre en compte la vitesse de convection que dans le voisinage proche de l'interface. En effet, pour obtenir une distance signée loin de l'interface, la vitesse de réinitialisation est suffisante. Nous pouvons également agir sur la troncature de la fonction level set. Précédemment, nous avons défini la level set comme une distance signée dans une zone d'épaisseur $2E$ autour de l'interface, la level set ayant pour valeur $\pm E$ dans le reste du domaine de calcul. Nous créons ainsi une transition très importante du gradient de la level set, sa norme passant brutalement de 1 à 0. C'est ce qui explique la création des instabilités observées à droite sur la figure 4.5. Pour créer une transition plus souple, Ville et al. se servent d'une nouvelle fonction level set $\tilde{\phi}$ [Vil 11a], représentée sur la figure 4.6 et définie de la manière suivante :

$$\tilde{\phi} = \begin{cases} \frac{-2E}{\pi} & \text{si } \phi < -E \\ \frac{2E}{\pi} \sin\left(\frac{\pi}{2E}\phi\right) & \text{si } -E < \phi < E \\ \frac{2E}{\pi} & \text{si } \phi > E \end{cases}$$

Dans cette expression, ϕ représente la distance signée sur l'ensemble du domaine.

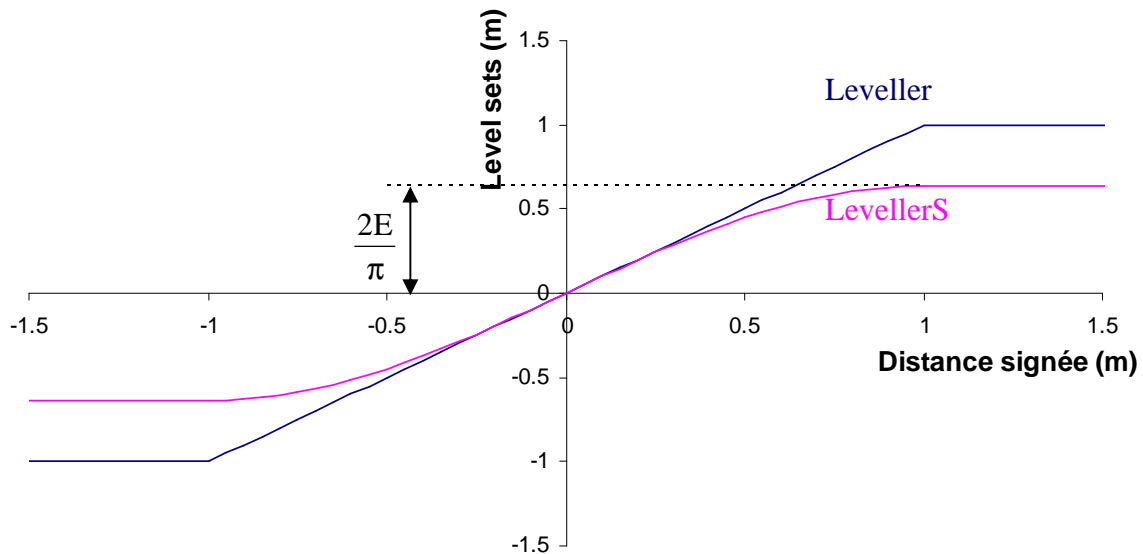


Figure 4.6. Comparaison des troncatures utilisées pour le Leveller et pour le LevellerS, avec $E = 1$ m.

Dans la zone de variation de la nouvelle level set, nous avons :

$$\frac{\partial \tilde{\phi}}{\partial \phi} = \cos\left(\frac{\pi}{2E}\phi\right) = \sqrt{1 - \left(\frac{\pi}{2E}\tilde{\phi}\right)^2}$$

Par conséquent, en considérant que $\|\vec{\nabla}\phi\| = 1$, la norme du gradient de $\tilde{\phi}$ s'écrit :

$$\|\vec{\nabla}\tilde{\phi}\| = \sqrt{1 - \left(\frac{\pi}{2E} \tilde{\phi}\right)^2}$$

Cette norme est unitaire à l'interface, puis diminue progressivement, jusqu'à atteindre une valeur nulle pour $|\phi| = E$. Pour réinitialiser cette level set, la norme du gradient ne doit donc plus être ramenée vers l'unité, mais vers son expression donnée ci-dessus, en résolvant l'équation de réinitialisation suivante :

$$\frac{\partial \tilde{\phi}}{\partial \tau} = S(\tilde{\phi}) \left(\sqrt{1 - \left(\frac{\pi}{2E} \tilde{\phi}^-\right)^2} - \|\vec{\nabla}\tilde{\phi}\| \right)$$

En théorie l'intérieur de la racine carrée est toujours positif, mais il arrive en pratique que ce ne soit pas le cas, à cause d'instabilités numériques. Dans ce cas, elle est prise égale à 0.

En combinant cette équation avec l'équation de convection, nous obtenons une équation de réinitialisation convective proche de celle établie sans le lissage de la level set en fonction sinus. Elle est donnée par :

$$\frac{\partial \tilde{\phi}}{\partial t} + \left(\bar{v} + \lambda S(\tilde{\phi}^-) \frac{\vec{\nabla}\tilde{\phi}^-}{\|\vec{\nabla}\tilde{\phi}^-\|_r} \right) \cdot \vec{\nabla}\tilde{\phi} = \lambda S(\tilde{\phi}^-) \sqrt{1 - \left(\frac{\pi}{2E} \tilde{\phi}^-\right)^2} \frac{\|\vec{\nabla}\tilde{\phi}^-\|}{\|\vec{\nabla}\tilde{\phi}^-\|_r}$$

$\|\vec{\nabla}\tilde{\phi}^-\|_r$ représente toujours la norme du gradient de la fonction level set régularisée avec $\|\vec{\nabla}\tilde{\phi}\|_r = \varepsilon + (1 - \varepsilon)\|\vec{\nabla}\tilde{\phi}\|$. Simplement, la valeur de ε par défaut n'est plus la même que pour le Leveller. Pour le levellerS, $\varepsilon = h/E$.

Pour le Leveller, les vitesses de convection et de réinitialisation jouent un rôle dans l'ensemble du domaine. Il est pourtant légitime de ne prendre en compte la vitesse de convection que sur l'interface et celle de réinitialisation dans tout le domaine sauf sur l'interface. Pour cette raison, un paramètre e est introduit dans le LevellerS. Il donne l'épaisseur de la zone sur laquelle seule la convection est résolue. Au-delà de cette zone, seule la réinitialisation est résolue.

$$\text{Nous avons ainsi : } S(\tilde{\phi}) = \begin{cases} -1 & \text{si } \tilde{\phi} \leq -e \\ 1 & \text{si } \tilde{\phi} \geq +e ; \bar{v} = \vec{0} \quad \text{si } |\tilde{\phi}| \geq e \\ 0 & \text{si } |\tilde{\phi}| < e \end{cases}$$

Si les paramètres e et E ne sont pas renseignés, ils prennent les valeurs suivantes :

$$e = 2h ; E = 20h$$

Comme la taille de maille h est calculée sur chaque élément et qu'elle varie donc autour de la taille de maille moyenne, e et E ont dans ce cas des valeurs différentes sur chaque élément. Pour éviter cela, nous préférons fixer nous-mêmes ces valeurs comme des multiples de la taille de maille moyenne.

Nous essayons le `LevelerS` avec les mêmes paramètres que dans la section 3.3, en prenant $E = 1$ m et $e = 0,2$ m. La figure 4.7 montre la valeur de la level set dans la zone pour laquelle seule la vitesse de convection intervient. Comme la vitesse n'est pas la même sur l'ensemble de cette zone, les distances entre les différentes isovaleurs changent au cours de la simulation, créant des instabilités numériques qui perturbent l'interface.

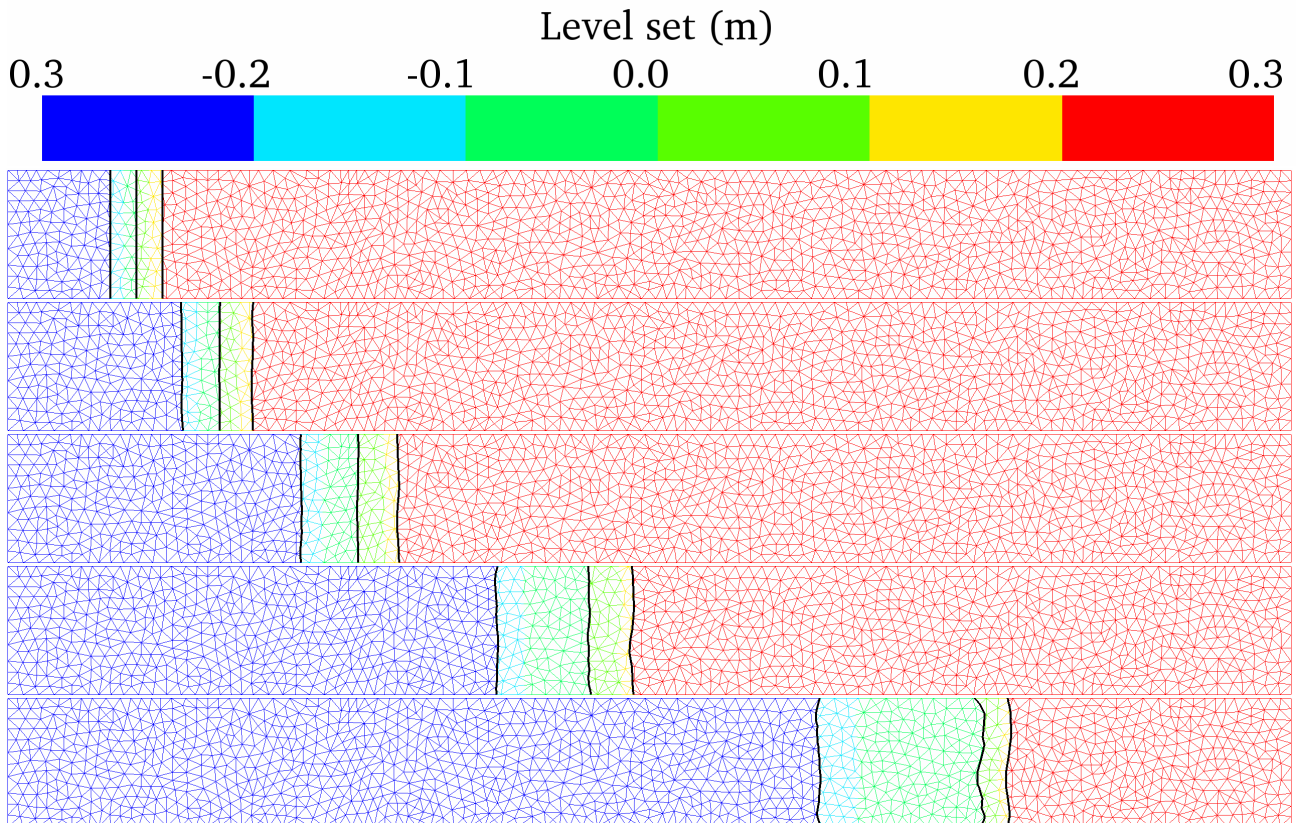


Figure 4.7. Fonction level set $\tilde{\phi}$ pour 0 s ; 0,5 s ; 1 s ; 1,5 s et 2 s. Les traits noirs représentent les isovaleurs -0,2 ; 0 et 0,2 de la fonction level set.

3.5. Solveur de réinitialisation convective `LevelerT`

Récemment Coupez a développé un nouveau solveur de réinitialisation convective, appelé `LevelerT`. Ce solveur a été intégré à `CIMLib` en mars 2011. Même s'il n'a pas encore été décrit dans la littérature, Coupez l'utilise dans un article sur la construction de métrique [Cou 11]. Il y définit une nouvelle fonction level set de la manière suivante :

$$\bar{\phi} = E \tanh\left(\frac{\phi}{E}\right)$$

Ce lissage présente des avantages par rapport à celui du LevellerS. Tout d'abord, il n'existe qu'une formule quelle que soit la valeur de la distance signée. Théoriquement, il est donc possible de retrouver une vraie distance signée à partir de cette fonction level set en utilisant l'argument tangente hyperbolique. De plus, la valeur de la level set loin de l'interface est ici directement égale à la valeur du paramètre E.

A partir de cette expression nous exprimons la norme du gradient de la level set, comme nous l'avons fait pour le LevellerS. Nous obtenons cette fois-ci :

$$\|\bar{\nabla}\bar{\phi}\| = 1 - \left(\frac{\bar{\phi}}{E}\right)^2$$

L'équation de réinitialisation convective de la fonction level set lissée en tangente hyperbolique est très proche de celle pour la level set lissée en sinus. Elle s'écrit :

$$\frac{\partial\bar{\phi}}{\partial t} + \left(\bar{v} + \lambda S(\bar{\phi}^-) \frac{\bar{\nabla}\bar{\phi}^-}{\|\bar{\nabla}\bar{\phi}^-\|_r} \right) \cdot \bar{\nabla}\bar{\phi} = \lambda S(\bar{\phi}^-) \left(1 - \left(\frac{\bar{\phi}}{E}\right)^2 \right) \frac{\|\bar{\nabla}\bar{\phi}^-\|}{\|\bar{\nabla}\bar{\phi}^-\|_r}$$

$$\text{avec } \|\bar{\nabla}\bar{\phi}^-\|_r = \varepsilon + (1 - \varepsilon)\|\bar{\nabla}\bar{\phi}\| ; \varepsilon = 0,1 \frac{h}{E} \text{ et } S(\bar{\phi}) = \frac{\bar{\phi}}{E}$$

A partir de cette expression, la démarche pour calculer les matrices locales, assembler la matrice et le second membre globaux et résoudre le système linéaire est similaire à celle détaillée pour le solveur ConvectionP1.

Sur l'exemple précédent, avec $E = 1$ m, le LevellerT permet d'obtenir une meilleure réinitialisation que le LevellerS. La figure 4.8 nous montre qu'après 2 s, les isovaleurs sont seulement légèrement décalées par rapport au début de la simulation. Cependant, elles ne sont plus tout à fait parallèles à l'axe des ordonnées. La qualité de la réinitialisation dépend fortement du paramètre E. En effet, sur le bas de la figure 4.8, prendre $E = \infty$ permet d'obtenir des isovaleurs parfaitement droites mais très éloignées par rapport au début du calcul. Dans ce cas, plus aucune réinitialisation n'est faite et les résultats sont similaires au solveur de convection.

Malgré ces résultats, le solveur LevellerT est pour le moment le plus performant dans CIMLib. C'est donc celui que nous utilisons par la suite. Des travaux sont actuellement en cours au CEMEF pour développer des solveurs de réinitialisation convective plus efficaces.

Pour relativiser cette observation, nous pouvons signaler que le champ de vitesse considéré ici est très éloigné de ceux calculés par notre solveur mécanique. Ainsi, si nous appliquons un champ de vitesse possédant une composante suivant Ox constante, quitte à la faire évoluer en fonction du temps, les isovaleurs sont bien mieux calculées. Cela semble suggérer que l'ajout d'une procédure pour étendre la vitesse à l'interface sur les nœuds voisins comme proposé par Tan & Zabaras [Tan 06] ainsi qu'Adalsteinsson & Sethian [Ada 99] pourrait constituer une amélioration importante de la résolution level set dans CIMLib.

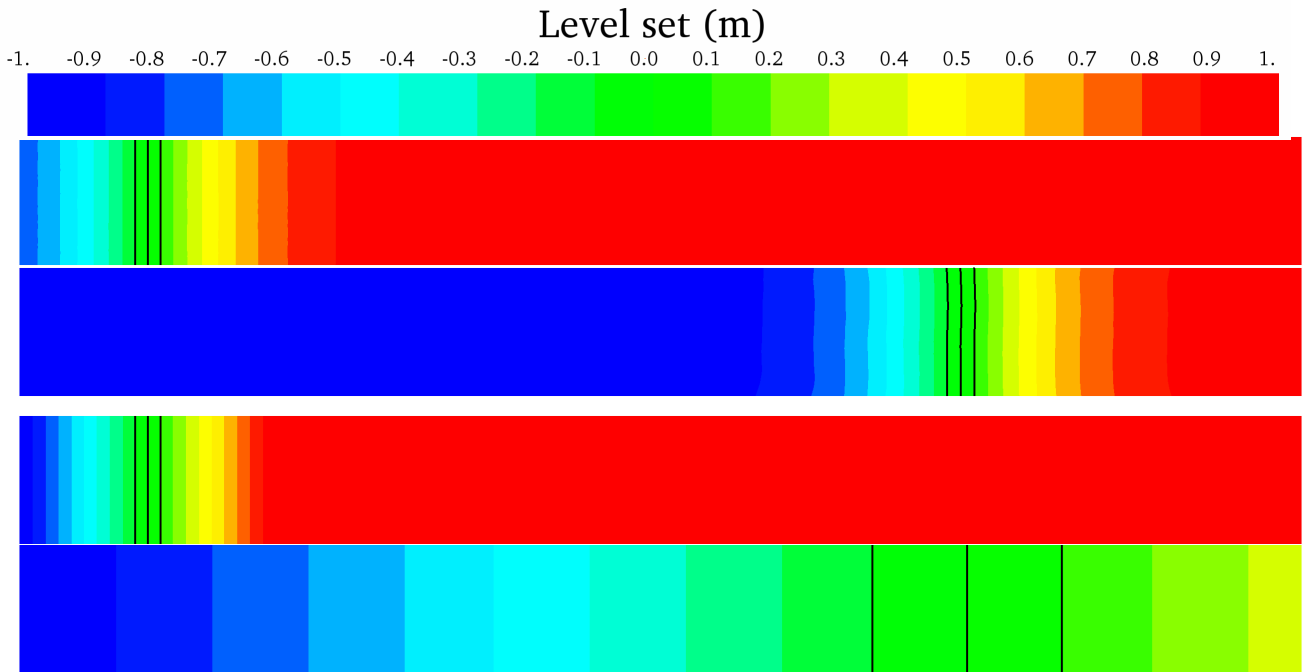


Figure 4.8. De haut en bas : fonction level set $\bar{\phi}$ pour $E = 1$ m et $t = 0$ s et 2 s, puis pour $E = \infty$ et $t = 0$ s et 2 s. Les traits noirs représentent les isovaleurs $-0,1$; 0 et $0,1$ de la fonction level set.

4. La méthode PLS (Particle Level Set)

Nous nous intéressons à présent à la méthode PLS développée à l'université Stanford par l'équipe de recherche de Fedkiw [Fos 01, Enr 02a, Enr 02b, Enr 02c, Enr 05]. Cette méthode s'appuie sur un maillage régulier constitué d'éléments carrés en 2D et cubiques en 3D, au sommet de chacun desquels est stockée une valeur de la fonction level set. Des particules sont positionnées près des interfaces entre les différents milieux. Chacune de ces particules possède une position, un rayon et un signe, permettant de savoir de quel côté de l'interface elle se situe.

Une équation de convection est résolue pour la level set et pour les particules. Les particules étant déplacées de manière plus précise, elles sont utilisées pour reconstruire la level set à l'interface. Après cette reconstruction, une méthode de type marche rapide est utilisée pour réinitialiser la fonction level set.

Pour nous faciliter la tâche, nous utilisons la librairie fournie gratuitement sur Internet par Mokheri & Faloutsos [Mok 06]. Nous y avons apporté des adaptations mineures pour la visualisation et le post-traitement des résultats. Nous décrivons à présent l'ensemble des étapes nécessaires au fonctionnement de la méthode PLS.

4.1. Convection de la level set

Pour convecter la level set, un schéma numérique très simple, connu sous le nom de semi-lagrangien et de premier ordre, est employé. Le principe de la méthode semi-lagrangienne, utilisée par Strain [Str 99, Str 00] repose sur la formule suivante [Mok 06] :

$$\phi(\bar{x}, t + \Delta t) = \phi(\bar{x} - \bar{v}(\bar{x})\Delta t, t)$$

avec ϕ la fonction level set, \bar{v} le vecteur vitesse, \bar{x} le vecteur position, t le temps et Δt le pas de temps.

Ainsi, pour déterminer la nouvelle valeur de la level set sur un nœud de la grille, il suffit de calculer la position $\bar{x} - \bar{v}(\bar{x})\Delta t$. Pour obtenir la valeur de la level set pour cette position et par conséquent la level set sur le nœud, nous appliquons une interpolation bilinéaire (trilinéaire en 3D) en utilisant les valeurs de la level set des nœuds appartenant à l'élément considéré. L'expression correspondante en 2D est donnée par Enright et al. [Enr 05]

Enright explique également que ce schéma est inconditionnellement stable et cite les travaux de Strain pour davantage de détails [Str 99]. Comme le schéma est inconditionnellement stable, le choix du pas de temps n'est pas gouverné par la condition CFL et Enright utilise un nombre CFL de 4,9, c'est-à-dire $\Delta t(|u|_{\max} / \Delta x + |v|_{\max} / \Delta y) = 4,9$ avec $|u|_{\max}$ et $|v|_{\max}$ les normes maximales des composantes de la vitesse suivant les axes Ox et Oy et Δx et Δy la taille des éléments suivant les axes Ox et Oy [Enr 05]. Cette valeur est beaucoup plus élevée que celle utilisée dans d'autres méthodes, pour lesquelles elle ne peut dépasser l'unité.

En choisissant le pas de temps de cette manière, cela signifie que l'interface ne peut jamais traverser plus de cinq éléments en un seul pas de temps et donc que la valeur de la level set en un nœud ne peut dépendre que de la level set des nœuds situés dans une zone d'épaisseur $5 \Delta x$ autour de l'interface. Pour cette raison, l'équation de convection n'est résolue que dans cette zone, occasionnant un gain de temps important. De manière analogue, une diminution ou une augmentation du pas de temps entraîneraient respectivement une diminution ou une augmentation de l'épaisseur de la zone où il est nécessaire de résoudre la convection de la level set.

4.2. Création des particules

Les particules étant utilisées pour corriger le caractère diffusif de la méthode level set, elles sont seulement placées autour de l'interface. Dans le travail de Foster & Fedkiw [Fos 01], les particules ne sont placées que d'un seul côté de l'interface. Mais il a été prouvé depuis qu'en placer des deux côtés est plus judicieux [Enr 02a]. Par défaut, nous plaçons 4^d particules par élément sur une épaisseur de $2\Delta x$ autour de l'interface (avec d la dimension considérée).

Le générateur de nombre aléatoire utilisé pour placer les particules est celui employé par Mokberi & Faloutsos. Il s'agit du Mersenne Twister de Matsumoto & Nishimura [Mat 00]. Nous l'utilisons tel quel sans chercher à comprendre son fonctionnement. En réalité, un générateur de nombres aléatoires n'est pas indispensable pour faire fonctionner la méthode PLS. En effet, il est tout à fait possible de déterminer des positions fixes pour les particules au sein de chaque élément. Avec 16 particules par élément en 2D, il nous suffirait simplement de disposer les particules de manière régulière à l'intérieur de chaque cellule. Mais le nombre de particules étant un paramètre ajustable par l'utilisateur, une disposition automatique paraît fastidieuse à envisager sans un tel générateur.

Afin de pouvoir corriger la fonction level set ultérieurement, il faut garder en mémoire pour chaque particule, en plus de sa position, deux informations supplémentaires. Il s'agit du signe et du rayon associé. Le signe nous permet de savoir de quel côté de l'interface la particule se situe. Quant au rayon, il nous donne une indication sur la distance de la particule à l'interface. Il est choisi de la manière suivante [Enr 02c] :

$$r_p = \begin{cases} r_{\max} & \text{si } |\phi_p| > r_{\max} \\ |\phi_p| & \text{si } r_{\min} < |\phi_p| < r_{\max} \\ r_{\min} & \text{si } |\phi_p| < r_{\min} \end{cases}$$

avec r_p le rayon de la particule, ϕ_p la level set interpolée sur la particule, r_{\min} et r_{\max} les rayons minimal et maximal d'une particule, dont les valeurs par défaut sont $r_{\min} = 0,1 \min(\Delta x, \Delta y)$ et $r_{\max} = 0,5 \min(\Delta x, \Delta y)$.

La figure 4.9 nous montre un exemple de positionnement initial des particules dans un cas où $\Delta x = \Delta y$.

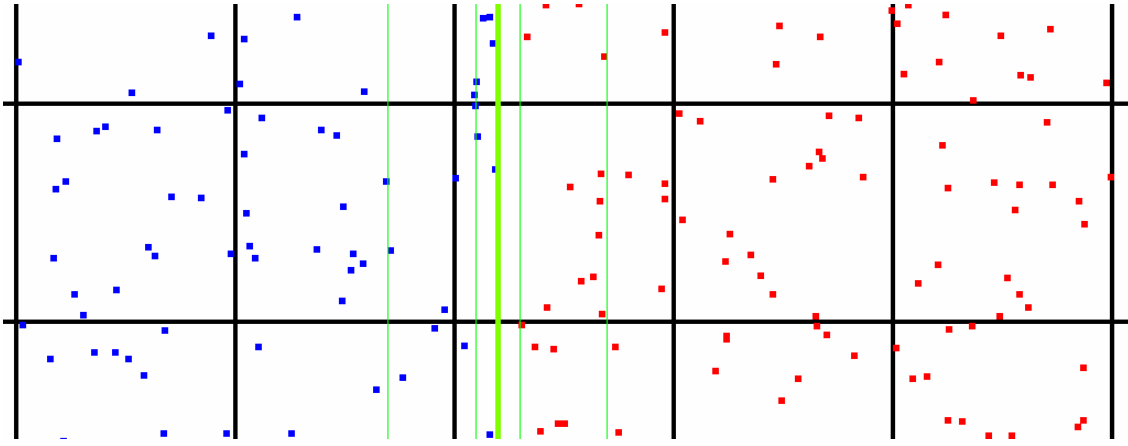


Figure 4.9. Exemple de positionnement des particules. Le quadrillage noir représente les éléments. Les particules bleues sont associées à une level set négative tandis que les particules rouges sont associés à une level set positive. Les traits verts correspondent de gauche à droite aux isovaleurs $\phi = -r_{\max}$, $\phi = -r_{\min}$, $\phi = 0$, $\phi = r_{\min}$ et $\phi = r_{\max}$.

4.3. Déplacement des particules

Pour que les particules puissent permettre de corriger la fonction level set, un schéma numérique plus précis est utilisé pour les déplacer. Il s'agit de la méthode de Runge-Kutta d'ordre 2 pour laquelle le calcul est le suivant :

$$\begin{aligned} \bar{x}_1 &= \bar{x}_{p0} + \bar{v}(\bar{x}_{p0})\Delta t \\ \bar{x}_2 &= \bar{x}_1 + \bar{v}(\bar{x}_1)\Delta t \\ \bar{x}_{p1} &= \frac{\bar{x}_{p0} + \bar{x}_2}{2} \end{aligned}$$

avec \bar{x}_{p0} et \bar{x}_{p1} la position de la particule à l'incrément actuel et à l'incrément suivant. La figure 4.10 illustre les formules ci-dessus.

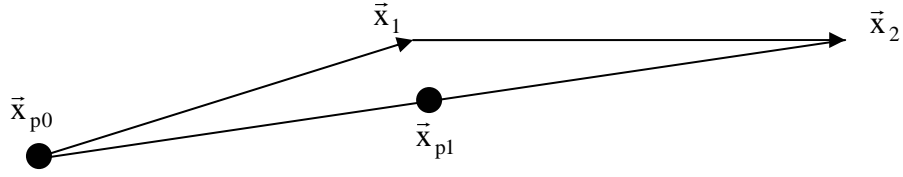


Figure 4.10. Déplacement d'une particule.

4.4. Correction de la level set

Après convection de la level set et des particules, les particules ayant traversé l'interface sont utilisées pour reconstruire la level set. Pour les identifier, la valeur de la level set est interpolée sur chaque particule. Si elle est négative et que le signe de la particule est positif, ou inversement, la particule est alors employée.

Pour cela, une distance signée à la particule est calculée sur chaque nœud de l'élément auquel elle appartient. Cette distance est déterminée en assimilant la particule à un cercle de rayon r_p (respectivement une sphère). Pour un nœud donné, nous avons :

$$\phi'(\bar{x}) = S_p (r_p - \|\bar{x} - \bar{x}_p\|)$$

avec $\phi'(\bar{x})$ la distance signée entre la particule et le nœud, \bar{x}_p la position de la particule et S_p le signe de la particule.

Les particules de signe positif sont utilisées pour reconstruire la zone pour laquelle $\phi > 0$ et celles de signe négatif permettent la reconstruction de la zone pour laquelle $\phi < 0$. Comme il peut y avoir dans un même élément des particules de signes positif et négatif qui ont traversé l'interface, Enright et al. [Enr 05] utilisent des level sets intermédiaire ϕ^+ et ϕ^- . Avant la correction, $\phi^+ = \phi^- = \phi$. Pour les particules de signe négatif, $\phi^+ = \max(\phi', \phi^+)$ et pour celles de signe positif, $\phi^- = \min(\phi', \phi^-)$.

Une fois toutes les particules échappées prises en compte pour corriger ϕ^+ et ϕ^- , il faut en déduire une seule valeur pour ϕ grâce à l'expression qui suit :

$$r_p = \begin{cases} \phi^+ & \text{si } |\phi^+| \leq \phi^- \\ \phi^- & \text{si } |\phi^+| > \phi^- \end{cases}$$

Ainsi, la level set reçoit la distance la plus proche de l'interface.

4.5. Réinitialisation de la level set

Comme dans CIMLib, la level set après convection n'est plus une distance signée. Il faut donc procéder à sa réinitialisation. Pour cela, la technique de premier ordre connue sous le nom de marche rapide est utilisée [Set 96, Set 99]. Puisque la convection n'est résolue que sur une

certaine épaisseur autour de l'interface, il en est de même pour la réinitialisation, qui n'est faite que sur une épaisseur de $6 \Delta x$.

Nous orientons le lecteur désirant une description précise et un code fonctionnel de l'algorithme marche rapide vers la librairie PLS de Mokhberi & Faloutsos [Mok 06].

4.6. Remarques

Comme la réinitialisation marche rapide peut entraîner de très légers déplacements de l'interface, une correction de la level set par les particules est également utilisée après la réinitialisation. Nous sommes toutefois forcé de constater que cette correction ne se fait que sur les éléments situés à l'interface et donc que la propriété de norme de gradient unitaire retrouvée par l'étape de réinitialisation n'est plus tout à fait vérifiée.

Il peut également être intéressant de redéfinir le rayon des particules à la fin d'un incrément en se basant sur les nouvelles valeurs de la level set. Mais puisqu'une telle redéfinition entraîne des erreurs, il vaut mieux ne pas en abuser. Enright et al. proposent de l'effectuer toutes les 20 itérations [Enr 05]. Cependant, nous choisissons ici de garder les valeurs initiales. Une idée similaire est l'ajout ou la suppression de particules pendant la simulation, en fonction de leur densité de distribution. Dans les cas simples que nous considérons, nous conservons l'ensemble des particules initial.

4.7. Effet de l'apport des particules

Nous avons décidé de ne pas montrer les résultats de la méthode PLS sur l'exemple du rectangle considéré précédemment pour les différents solveurs CIMLib. En effet, compte tenu du champ de vitesse incompressible et du positionnement initial aléatoire des particules, la correction de l'interface par ces dernières entraîne des oscillations. Pour obtenir la bonne solution, nous pouvons prolonger le champ de vitesse à l'interface [Ada 99, Tan 06], ce qui est assez simple à réaliser sur le type de maillage considéré ici.

Pour démontrer l'intérêt du couplage entre les particules et la level set, nous préférons considérer l'exemple bien connu de la rotation d'un cercle. A l'intérieur d'un domaine carré de 1 m de côté est placé un cercle de rayon 15 cm et de centre (0,5 m ; 0,75 m). Nous appliquons le champ de vitesse incompressible dont les deux composantes sont données ci-dessous :

$$v_x = 2\pi(0,5 - y)$$

$$v_y = 2\pi(x - 0,5)$$

Avec un tel champ de vitesse, le cercle tourne autour du centre du domaine dans le sens trigonométrique tout en gardant sa forme initiale. Il met 1 s pour revenir à sa position de départ. Ses positions analytiques sont montrées sur la figure 4.11 pour différents temps. En prenant $\Delta x = 1$ cm et $\Delta t = 5$ ms, nous lançons le calcul avec et sans particules. La level set est montrée dans les deux cas au début du calcul, après un tour et après deux tours sur la figure 4.12. Comme la level set n'a pas de signification dans la zone où elle n'est pas réinitialisée, nous faisons le choix de la tronquer à $\pm 6\Delta x$ pour la visualisation.

Sans les particules, le cercle perd très vite de sa surface à cause d'une résolution level set très diffusive [Mok 06]. Avec l'apport des particules, malgré l'ordre peu élevé des schémas numériques pour la convection et la réinitialisation, la conservation est plutôt bonne. Nous observons une perte de moins de 1 %.

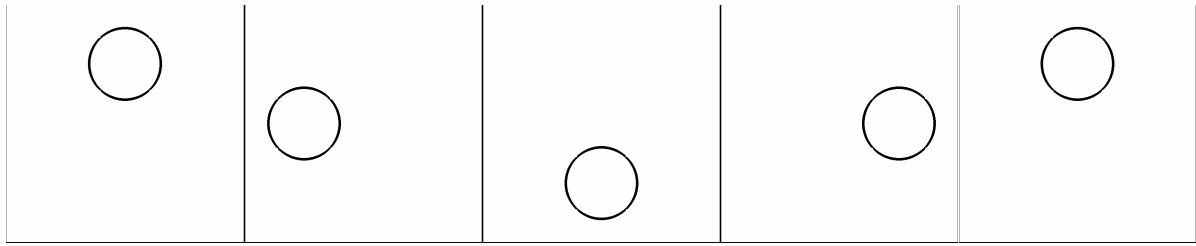


Figure 4.11. Position analytique du cercle pour différents temps. De gauche à droite, $t = 0$; 0,25 ; 0,5 ; 0,75 et 1 s.

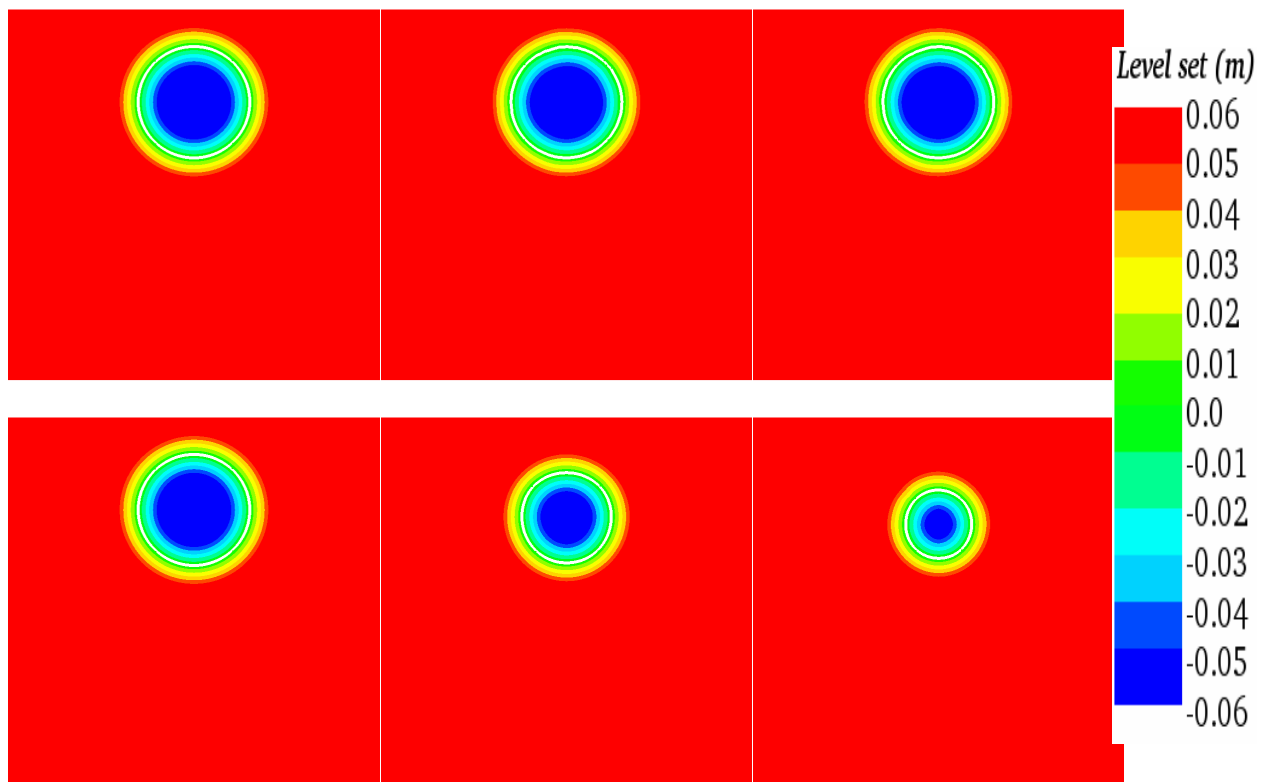


Figure 4.12. Level set au début de la simulation (à gauche), après un tour (au milieu) et après deux tours (à droite). Calcul avec particules (en haut) et sans particules (en bas). L'ensemble du domaine de calcul est montré. Le trait blanc et fin représente l'interface.

5. Comparaison des deux méthodes

Nous allons à présent évaluer la précision des deux méthodes introduites en utilisant deux tests classiques de la littérature pour lesquels les champs de vitesse sont fixés.

5.1. Rotation d'un disque

Nous reprenons le cas de la rotation d'un disque exposé ci-dessus. Ce cas vraiment très simple a été largement utilisé dans des travaux du CEMEF [Bru 04, Bas 06, Vil 11a, Vil 11b]. Ville et al. [Vil 11a, Vil 11b] ont notamment effectué des calculs avec plusieurs tailles de maille et

pas de temps en déterminant à chaque fois la conservation de la surface du disque. Ils ont pour cela utilisé le solveur LevellerS. Nous nous proposons ici d'utiliser le solveur LevellerT et de comparer les résultats obtenus avec ceux obtenus par la méthode PLS.

Pour évaluer la surface de l'objet au cours de la simulation avec la méthode PLS, nous utilisons la technique décrite par Caiden, Fedkiw & Anderson [Cai 01]. Avec CIMLib, nous employons la méthode introduite dans le deuxième chapitre.

Nous considérons des tailles de maille entre 1 cm et 1 mm et des rapports entre la taille de maille et le pas de temps de 1 m.s^{-1} à 16 m.s^{-1} pour les deux méthodes. Nous n'allons pas montrer les conservations obtenues avec CIMLib pour $h/\Delta t = 1 \text{ m.s}^{-1}$, car les résultats sont alors pollués par des oscillations apparaissant au bord du domaine. En effet, pour les pas de temps correspondants, la condition CFL n'est plus respectée, en particulier sur les quatre coins du domaine de calcul pour lesquels la norme de la vitesse est la plus grande.

Sur les coins, nous avons $\|\vec{v}\| = 4,44 \text{ m.s}^{-1}$. Sur le contour du disque, $\|\vec{v}\| = 2,51 \text{ m.s}^{-1}$ au maximum durant la simulation. Cette valeur est analytique et la norme maximale de la vitesse sur le contour peut être légèrement supérieure en pratique. La valeur de la vitesse loin de l'interface n'est censée jouer aucun rôle dans la convection du cercle. Par conséquent, nous décidons de la tronquer en imposant $\|\vec{v}\| \leq 3 \text{ m.s}^{-1}$. La figure 4.13 représente les champs de vitesse initial et tronqué, ainsi que les level sets obtenues avec le LevellerT après 1 s pour une taille de maille de 10 mm et un pas de temps de 10 ms avec ces deux champs de vitesse. Des oscillations sont présentes dans les deux cas de figure. Toutefois, elles sont beaucoup plus importantes avec la vitesse initiale, puisqu'elles entraînent la création d'une nouvelle interface dans le coin inférieur droit du domaine, ce qui n'est pas le cas en utilisant la troncature de la vitesse. Bien évidemment, pour une simulation plus complexe, nécessitant par exemple un couplage avec d'autres solveurs éléments finis, une telle apparition de matière a bien souvent des conséquences catastrophiques sur les solutions obtenues. Avec la méthode PLS, puisque l'équation de convection n'est résolue que sur une fine bande autour de l'interface, le champ de vitesse loin de l'interface n'est pas utilisé lors des calculs et le tronquer ne modifie pas les résultats obtenus.

Notre étude sur la taille de maille et le pas de temps est illustrée sur la figure 4.14. Cette figure représente la perte de surface du cercle initial, exprimée en pourcent, après une rotation. Pour nous placer dans les mêmes conditions que Basset et Ville, nous n'avons pas tronqué la vitesse. Cette figure nous permet de tirer plusieurs conclusions importantes.

Pour $h/\Delta t = 2 \text{ m.s}^{-1}$ (à gauche de la figure 4.14), CIMLib donne des meilleurs résultats que la méthode PLS. Nous notons cependant que l'écart entre les résultats est de plus en plus faible à mesure que la taille de maille est diminuée. Nous utilisons les huit points correspondant à $h/\Delta t = 2 \text{ m.s}^{-1}$ pour tracer la figure 4.15, qui nous donne la perte de surface en fonction de la taille de maille pour les deux méthodes. La réduction de l'écart entre les résultats obtenus par les deux méthodes est clairement visible sur cette figure, avec une vitesse de convergence supérieure pour la méthode PLS.

Revenons à l'analyse de la figure 4.14. Pour une simulation à une taille de maille donnée, il n'est pas facile de déterminer le pas de temps optimal pour lequel nous avons un bon compromis entre temps de calcul et erreur commise. Pour les deux méthodes, nous n'obtenons pas la bonne solution si le pas de temps est trop grand. En le diminuant, nous arrivons à avoir des résultats cohérents, qui sont encore améliorés lorsque le pas de temps est diminué pour la méthode PLS. Même si pour réduire le temps de calcul, il ne paraît pas judicieux de considérer $h/\Delta t > 4 \text{ m.s}^{-1}$, nous voyons que les résultats ne sont pas dégradés si le pas de temps diminue. Au contraire pour CIMLib, prendre un pas de temps trop petit augmente à la fois le temps de calcul et la perte de surface. Le nombre de simulations est sans doute trop faible pour approfondir cette observation, mais il nous semble tout de même que

l'augmentation de la perte en fonction de l'inverse du pas de temps est plutôt régulière, les trois points avec les pas de temps les plus petits pour chaque taille de maille formant des droites parallèles les unes aux autres. Nous tenons tout de même à proposer l'explication suivante. Diminuer le pas de temps implique un plus grand nombre de résolutions de l'équation de réinitialisation convective et donc un plus grand nombre de réinitialisations. Or, une réinitialisation ne permet pas de conserver exactement la définition de l'interface et il est donc préférable d'en effectuer le moins possible.

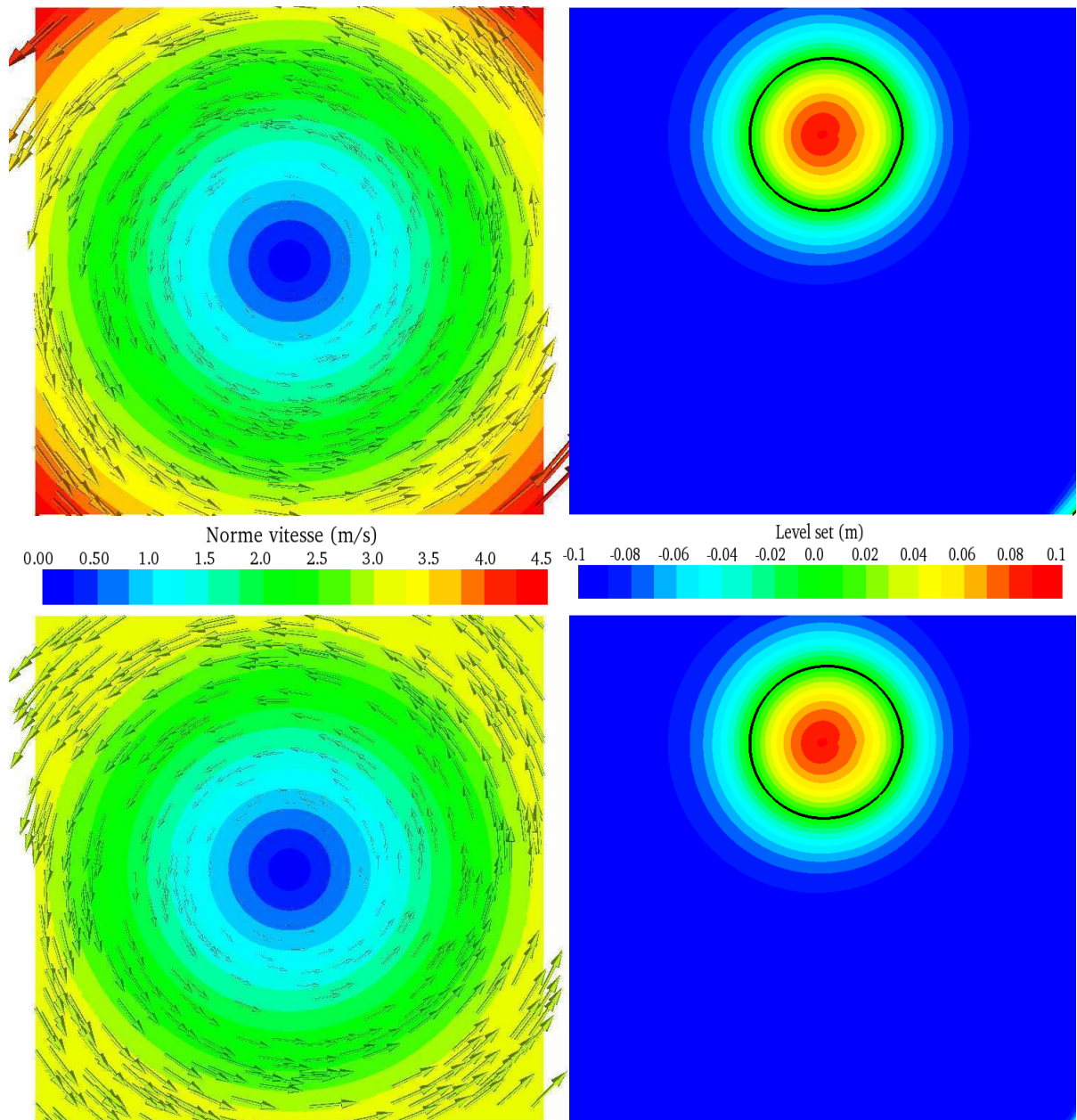


Figure 4.13. En haut à gauche : champ de vitesse initial. En haut à droite : level set pour le champ de vitesse initial après 1 s. En bas à gauche : champ de vitesse tronqué. En bas à droite : level set pour le champ de vitesse tronqué après 1 s. Le trait noir représente l'interface. Ces résultats sont obtenus avec le solveur LevellerT.

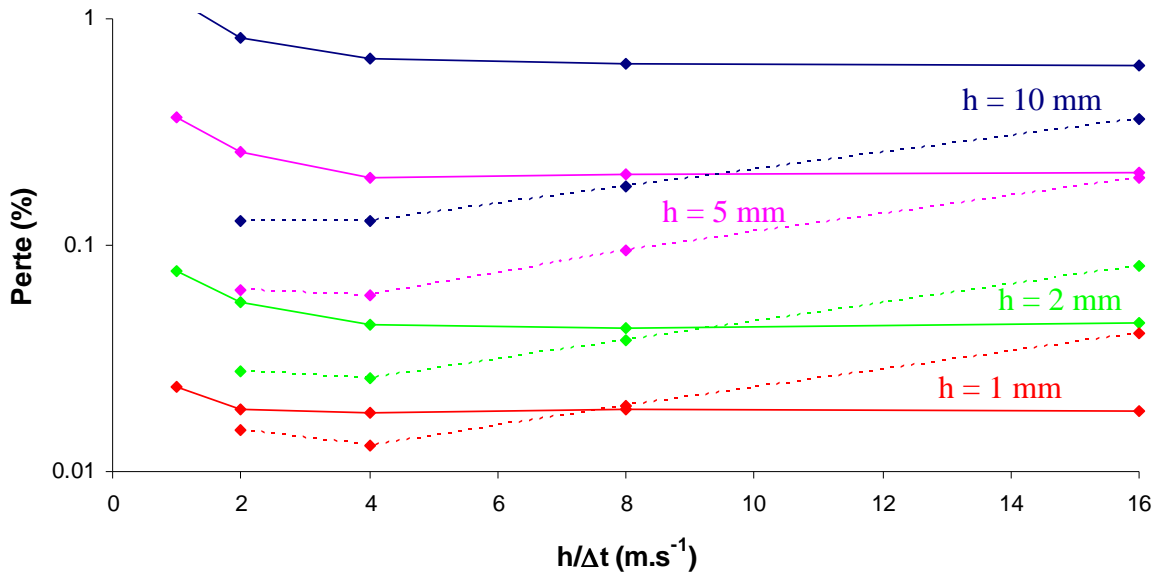


Figure 4.14. Conservation obtenue par le solveur LevellerT de CIMLib et la méthode PLS pour différents tailles de maille et pas de temps. En pointillés, résultats CIMLib et en traits pleins résultats PLS. Les tailles de maille sont indiquées directement sur la figure.

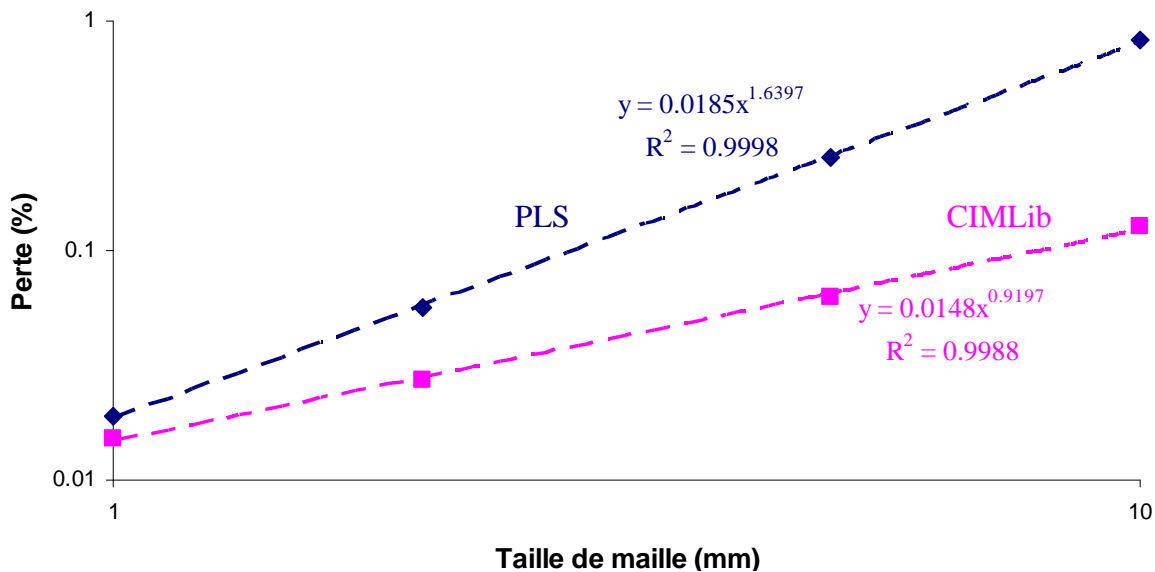


Figure 4.15. Vitesse de convergence des méthodes CIMLib et PLS.

5.2. Test du vortex

Nous voulions initialement présenter des résultats sur la rotation du disque de Zalesak [Zal 79], mais ce cas test déjà étudié sommairement par Ville [Vil 11a, Vil 11b] ne nous apprend pas grand-chose de plus que l'étude menée sur la rotation du disque. Nous nous contentons donc de souligner que ce test est un bon indicateur des erreurs de diffusion rencontrées dans une méthode de déplacement d'interfaces et qu'il est largement utilisé dans la littérature. Mais il ne nous renseigne aucunement sur la capacité de nos deux méthodes à représenter de manière précise des structures filamenteuses longues et fines, qui peuvent

apparaître lors de certains écoulements, comme ceux rencontrés par Bell et al. [Bel 89]. Nous utilisons donc le champ de vitesse donné par Rider & Kothe [Rid 98]. Ce test est déjà employé dans CIMLib par Olivier Basset [Bas 06]. Le cercle initial est placé au même endroit que précédemment et le champ de vitesse incompressible considéré est donné par :

$$v_x = -2 \sin^2(\pi x) \sin(\pi y) \cos(\pi y)$$

$$v_y = 2 \sin(\pi x) \cos(\pi x) \sin^2(\pi y)$$

Pour les deux méthodes, le pas de temps est proportionnel à la taille de maille. Pour une taille de maille de 2 mm, nous prenons $\Delta t = 1$ ms pour CIMLib et 5 ms pour la PLS. Nous souhaitons étudier la conservation de la surface après un temps de 5 s. La figure 4.16, représentant la level set pour la PLS avec $h = 1$ mm, nous donne une idée assez précise de la forme de l'objet à la fin de la simulation.

Etant donné que pour une taille de maille identique, le temps de calcul de la méthode PLS est beaucoup plus faible, nous comparons les résultats des deux méthodes à temps de calcul équivalent et non à taille de maille équivalente. Ainsi la figure 4.17 nous donne la conservation en fonction du temps de calcul pour les deux méthodes. Afin de prendre une échelle logarithmique pour cette figure, nous montrons la valeur absolue de la conservation calculée comme suit :

$$C = \frac{|S_f - S_i|}{S_i}$$

avec S_i et S_f les surfaces initiale et finale et C la conservation.

Le temps de calcul représenté ne comprend ni le temps du traitement nécessaire pour calculer la surface, ni celui des différentes sorties effectuées. Si pour les deux méthodes, la conservation s'améliore lorsque le temps de calcul augmente et donc lorsque la taille de maille diminue, les résultats sont tout de même très différents. Notamment, nous constatons un gain de surface avec CIMLib et une perte avec la méthode PLS. CIMLib est également beaucoup plus lent pour donner un même pourcentage de conservation. Ainsi pour une taille de maille de 2 mm, la méthode PLS donne une perte de 0,41 % pour un temps de calcul de 50 s, alors que CIMLib donne un gain de 7 % en un temps plus de 1000 fois supérieur. Les résultats avec CIMLib sont comparables avec ceux obtenus par Olivier Basset qui évoque un gain de 5 % pour un temps plus court et une taille de maille plus importante [Bas 06].

Des tests avec des champs de vitesse fixés existent également en trois dimensions. Toutefois, compte tenu de l'étude 2D que nous venons de mener, nous sommes convaincu que présenter une étude 3D n'aurait qu'un intérêt assez faible. Pour avoir un aperçu de la comparaison entre les résultats CIMLib et PLS, nous conseillons au lecteur les travaux de Ville et al. [Vil 11a, Vil 11b] et ceux de Enright et al. [Enr 05]

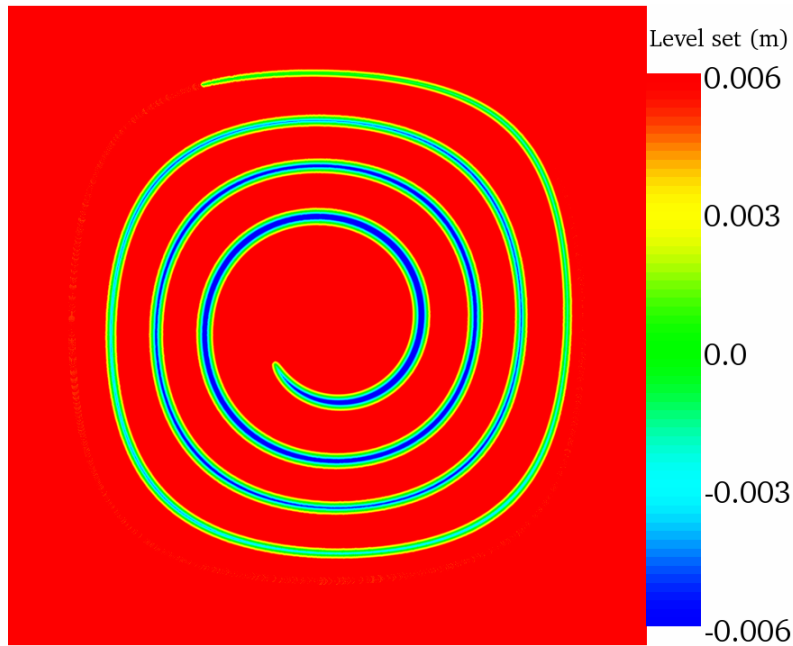


Figure 4.16. Level set après 5 s pour la méthode PLS avec $h = 1$ mm.

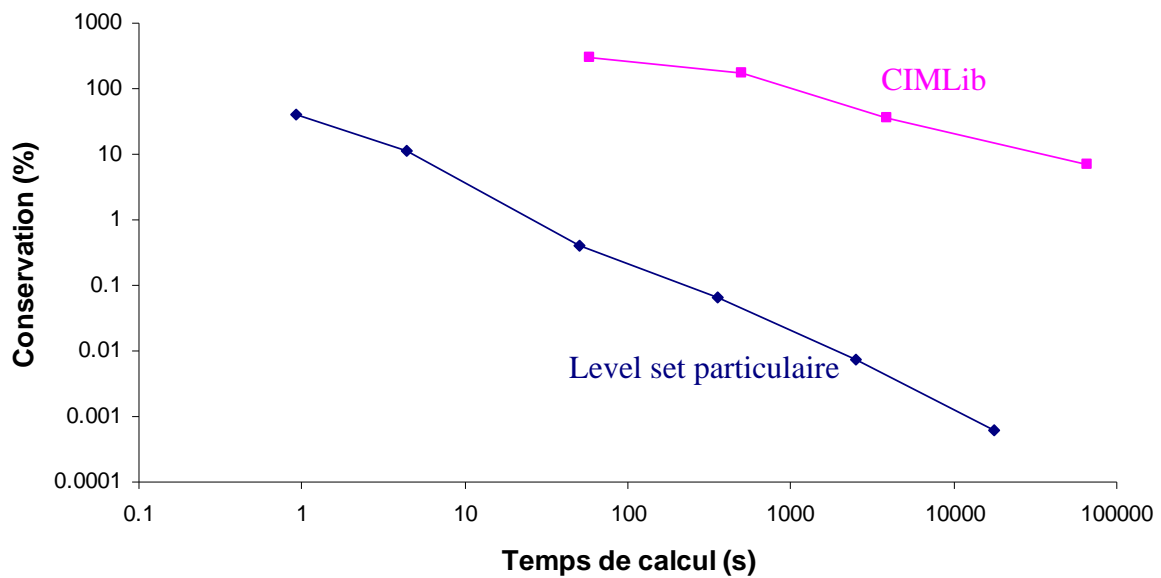


Figure 4.17. Conservation de la surface du cercle en fonction du temps de calcul pour les méthodes CIMLib et PLS.

5.3. Conclusion

La comparaison entre les deux méthodes nous semble à l'avantage de la méthode PLS. Elle est beaucoup plus rapide à précision équivalente. En outre elle est très utilisée dans la littérature, dans laquelle de nombreuses améliorations et variantes sont proposées [Hie 05, Gau 08, Los 04, Los 07, Pig 05, Wan 09], parmi lesquelles nous pouvons citer la méthode MLS [Mih 06, Mih 07, Mih 08, Mih 09, Elm 11] qui permet de limiter grandement le nombre de particules et donc le coût de calcul et la mémoire nécessaire, surtout en 3D.

Malheureusement, les résultats PLS présentés ici n'ont été obtenus qu'avec la librairie de Mokheri & Faloutsos [Mok 06]. Intégrer cette méthode à CIMLib n'est pas un travail simple. En effet, l'architecture de notre librairie de calculs éléments finis est totalement différente de celle nécessaire à l'implémentation de la méthode PLS. Les solveurs de CIMLib sont tous utilisables en parallèle sur des maillages tétraédriques (ou triangulaires en 2D) non structurés, alors que nous avons utilisé des calculs séquentiels sur des maillages rectangulaires structurés pour la méthode PLS. Nous n'avons en outre pas détaillé d'applications 3D sur des maillages hexaédriques.

Pour incorporer la méthode PLS telle quelle dans CIMLib, il nous faudrait ainsi être capable de la rendre parallèle et de transférer les données entre maillages tétraédriques et hexaédriques. Les différentes étapes de la méthode PLS sont facilement parallélisables, à l'exception de la réinitialisation marche rapide, qui est séquentielle par nature. Mais des implémentations parallèles de cet algorithme existent, comme celles proposées par Tugurlan [Tug 08] et Herrmann [Her 03]. En ce qui concerne le transfert de données entre les deux types de maillage, certains outils, comme l'interpolation des champs lors du remaillage, sont déjà disponibles. De plus, un tel transfert a déjà été mis en place dans CIMLib par Carozzani et al., dans le cadre d'un couplage entre automates cellulaires et éléments finis [Car 12]. Cependant, les interactions entre les deux maillages y étant différentes de celles dont nous avons besoin, une réflexion plus approfondie est nécessaire pour déterminer si ces outils sont exploitables dans notre situation.

Il est également envisageable d'adapter la méthode PLS à des éléments tétraédriques, afin d'améliorer la précision du LevellerT. Une telle adaptation n'existe pas à notre connaissance dans la littérature.

Une autre solution consisterait à ne considérer que le maillage hexaédrique nécessaire au fonctionnement de la méthode PLS. Il faudrait alors que nous développions des solveurs compatibles avec l'utilisation d'un tel maillage. Pour cela, nous étudierions sans aucun doute les travaux de Fedkiw et de son équipe de manière détaillée et notamment l'article de Losasso, Gibou & Fedkiw, dans lequel une méthode de résolution des équations de Navier Stokes sur un maillage de type octree est décrite [Los 04].

6. Test de l'inclusion

Comme dans le chapitre précédent, nous avons choisi d'utiliser le test de traction de l'inclusion pour valider notre solveur LevellerT. Nous considérons une inclusion sphérique de rayon 25 μm placée au centre d'une matrice cubique d'1 mm de côté. Comme nous voulons simplement valider le solveur d'interface dans ce chapitre, matrice et inclusion ont même viscosité, les deux matériaux étant choisis newtoniens. Le repère orthonormé utilisé possède le même centre que l'inclusion. Le champ de vitesse restant alors identique quelle que soit la position de l'inclusion, nous utilisons sa valeur analytique dont les composantes sont données ci-dessous :

$$v_x = -\frac{kx}{2} ; v_y = ky ; v_z = -\frac{kz}{2} \text{ avec } k = \frac{V_h - V_b}{H}$$

V_h et V_b sont les vitesses imposées sur les plans supérieur et inférieur de la matrice et H la hauteur de la matrice. Nous prenons $V_h = -V_b = 0,05 \mu\text{m.s}^{-1}$.

D'après Gilormini & Montheillet, une inclusion sphérique placée dans une matrice soumise à ce champ de vitesse se transforme en ellipsoïde de révolution [Gil 86]. Il est ainsi possible de

déterminer l'évolution analytique de la forme de l'inclusion au cours du temps. En considérant que a est la longueur du demi-axe de l'inclusion suivant l'axe de traction, la vitesse maximale sur le contour de l'inclusion suivant Oy s'écrit :

$$v_y = ka .$$

Ce qui, après intégration et prise en compte de la condition initiale nous donne :

$$a = R \exp(kt)$$

R étant le rayon initial de l'inclusion.

6.1. Solution numérique

Nous utilisons un maillage du même type que celui présenté dans la sous-partie 4.1 du troisième chapitre, avec une taille constante de $2 \mu\text{m}$ sur une épaisseur de $35 \mu\text{m}$ autour de l'interface, un coefficient directeur f de 0,2 (voir test de l'inclusion dans le chapitre précédent) et une taille de maille maximale de $50 \mu\text{m}$. Nous simulons la traction sur une durée de 4000 s. Pour déterminer la longueur du demi-axe suivant Oy à chaque incrément, nous calculons simplement l'ordonnée des deux points d'intersection entre la surface de l'inclusion et l'axe des ordonnées. La différence entre ces deux ordonnées nous donne alors la longueur de l'axe complet de l'ellipse, que nous n'avons plus qu'à diviser par deux.

La figure 4.18 montre le contour de l'inclusion au début de la simulation, après 2000 s et après 4000 s en utilisant un pas de temps de 100 s et une épaisseur de lissage de la fonction level set de 10 fois la taille de maille moyenne.

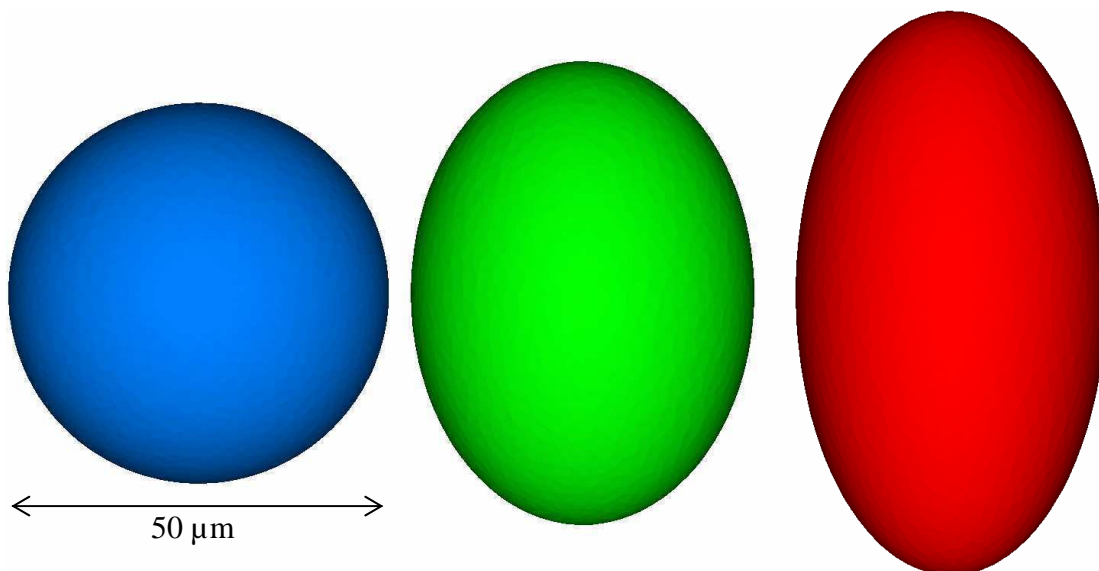


Figure 4.18. Contour de l'inclusion au début du calcul, après 2000 s et après 4000 s.

Sur la figure 4.19 nous pouvons voir l'évolution de la longueur du demi-grand axe de l'ellipsoïde de révolution au cours du temps. La valeur numérique est très proche de la solution analytique. Après 4000 s, le volume de l'inclusion est de $65263 \mu\text{m}^3$ pour un volume initial de $65315 \mu\text{m}^3$. Cela représente une perte de volume de 0,079 %.

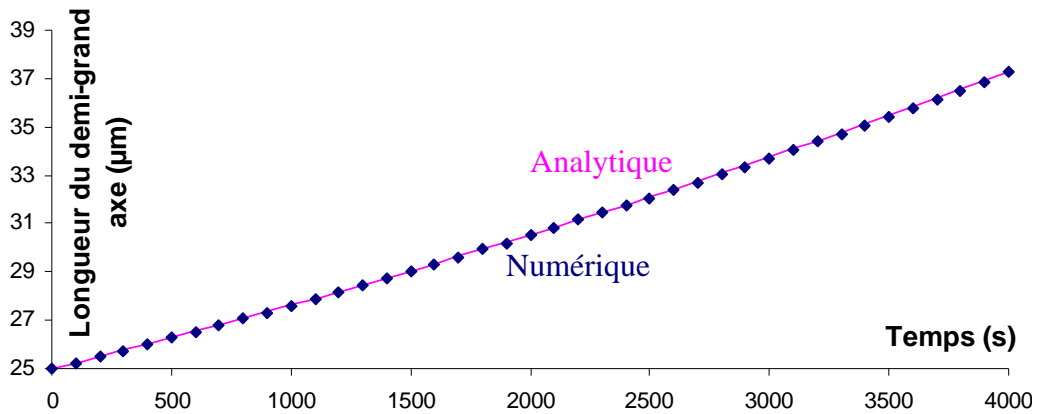


Figure 4.19. Evolution de la longueur du demi-grand axe au cours de la simulation. Solution analytique en trait plein et solution numérique en marques ponctuelles.

Pour ce calcul, nous avons mené une étude en fonction de la taille de maille et du pas de temps. Comme prévu, nous obtenons de meilleurs résultats pour une taille de maille plus faible. En revanche le choix du pas de temps est assez délicat, puisque sa diminution entraîne parfois une dégradation des résultats. Ces observations étant similaires à celles obtenues pour la rotation du disque, nous jugeons inutile de les présenter.

6.2. Epaisseur de lissage

Nous relançons le calcul précédent pour différentes épaisseurs de lissage et montrons les volumes obtenus après 4000 s sur la figure 4.20. Nous constatons que plus l'épaisseur de lissage est grande, meilleure est la conservation. Comme illustré par la figure, cette amélioration est loin d'être négligeable, puisque prendre une épaisseur de 10^{50} h (point non montré sur la figure pour lequel la level set initiale est une distance signée) permet d'obtenir une perte de 0,00011 %, 70 fois plus petite que celle pour $E = 10$ h.

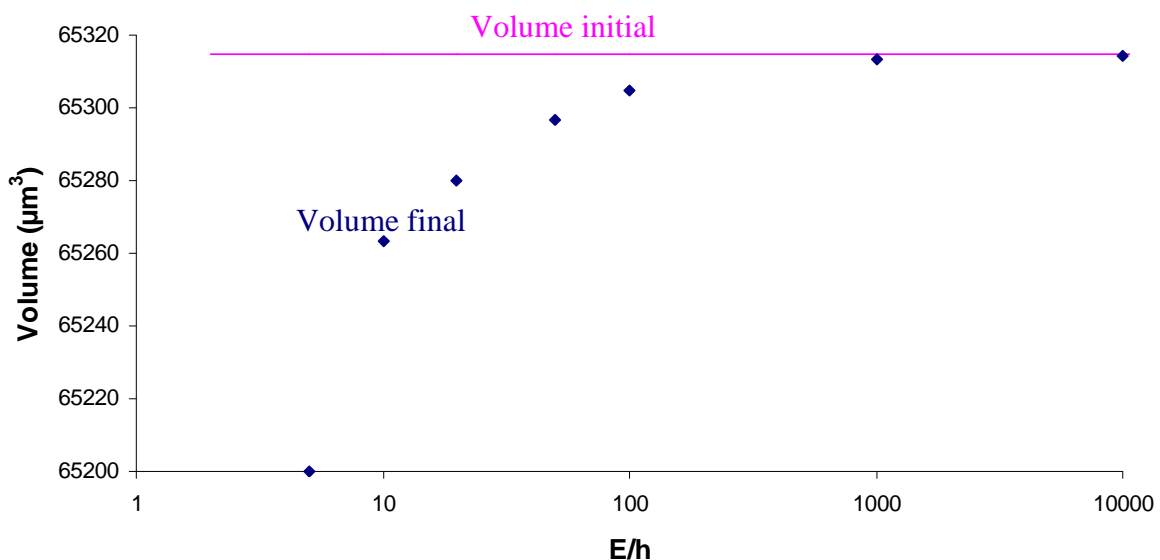


Figure 4.20. Volumes initial et final en fonction de l'épaisseur de lissage. Volume initial en rose et volume final en bleu.

Nous avons vu, lors de la présentation du LevellerT, que prendre une épaisseur de lissage très grande revient à ne pas faire de réinitialisation et permet de mieux calculer la position de

l'interface. Cela est confirmé par comparaison des isovaleurs analytique et numérique pour $E = 10^{50} h$. Il faut tout de même souligner que cette observation est valable car nous considérons des champs de vitesse linéaires, pour lesquels le gradient de la level set, même s'il n'a plus une norme unitaire, possède des valeurs très régulières. Avec le champ de vitesse unidirectionnel utilisé pour tracer la figure 4.4, le gradient avait d'ailleurs exactement la même valeur sur tout le domaine, avant réinitialisation de la level set.

Nous pouvons ainsi affirmer que dans cet exemple, la réinitialisation est responsable de la plus grosse partie de la perte de volume. Il est donc nécessaire de connaître l'épaisseur de lissage minimale à considérer pour obtenir une bonne réinitialisation, afin de respecter au

mieux $\|\bar{\nabla}\bar{\phi}\| = 1 - \left(\frac{\bar{\phi}}{E}\right)^2$, tout en minimisant la perte de volume.

Connaissant la forme de l'ellipsoïde après 4000 s, nous sommes capables de déterminer la valeur analytique de la distance signée à l'interface sur l'ensemble du domaine. A partir de la level set et de l'épaisseur de lissage, nous utilisons l'argument tangente hyperbolique pour accéder à la valeur de la distance signée pour chaque simulation. Pour simplifier notre analyse, nous n'allons considérer que les valeurs de la level set sur l'axe des abscisses et celui des ordonnées, ce qui est suffisant pour avoir une idée de la qualité de la réinitialisation. Nous pouvons ainsi nous contenter d'utiliser le calcul de la distance d'un point à une ellipse décrit par Eberly dans sa configuration la moins complexe [Ebe 08].

Les figures 4.21 et 4.22 représentent la différence entre les distances signées analytique et numérique suivant les axes Ox et Oy, pour des abscisses et des ordonnées proches du centre du repère. Nous choisissons de tracer cette différence plutôt que la valeur de la distance signée elle-même, car cela permet d'utiliser une échelle beaucoup plus petite et de mieux visualiser les résultats.

Comme attendu, augmenter l'épaisseur de lissage entraîne une erreur plus importante, rendant ainsi la réinitialisation moins performante. Cependant, même avec une épaisseur de lissage faible, l'erreur est loin d'être négligeable. Par exemple, la distance signée pour $E/h = 5$ et le point de coordonnées $x = 40 \mu\text{m}$ et $y = 0$ est de $20,70 \mu\text{m}$ (point montré par une flèche rouge sur la figure 4.21) à comparer à la valeur analytique de $19,53 \mu\text{m}$. Cela représente donc une erreur de 6 %. Pour $E/h = 5$ et le point de coordonnées $x = 0$ et $y = 60 \mu\text{m}$ (point montré par une flèche rouge sur la figure 4.22), les valeurs numérique et analytique sont de $20,13 \mu\text{m}$ et $22,70 \mu\text{m}$ et l'erreur est alors supérieure à 11 %.

Pour calculer l'erreur commise, il semble toutefois intéressant et plus logique de rapporter l'écart sur les valeurs analytique et numérique de la level set à son déplacement analytique. Pour $x = 40 \mu\text{m}$ et $y = 0$, la level set initiale est de $15 \mu\text{m}$ et nous avons donc un déplacement analytique de $4,53 \mu\text{m}$ ($19,53 - 15$) pour un écart de $1,17 \mu\text{m}$ ($20,7 - 19,53$). Pour $x = 0 \mu\text{m}$ et $y = 60 \mu\text{m}$, la level set initiale est de $35 \mu\text{m}$ et le déplacement analytique est alors de $12,3 \mu\text{m}$ ($19,53 - 15$) pour un écart de $2,57 \mu\text{m}$ ($22,70 - 20,13$). Nous avons alors respectivement 26 et 21 % d'erreur, ce qui semble important.

Les deux figures 4.21 et 4.22, nous donnent un autre renseignement. En effet, elles nous permettent de savoir si la distance calculée est supérieure ou inférieure à la distance analytique : quand l'erreur est négative, la valeur numérique est supérieure et quand l'erreur est positive la valeur numérique est inférieure à la valeur analytique. Nous remarquons ainsi que pour les points à l'extérieur de l'inclusion sur l'axe Ox (ceux pour lesquels $|x| > 20 \mu\text{m}$ sur la figure 4.21), la valeur numérique est supérieure à la valeur analytique et inversement sur l'axe Oy (pour les points ayant $|y| > 37 \mu\text{m}$ sur la figure 4.22). Cela s'explique par le champ de vitesse que nous imposons. Si nous ne résolvons que l'équation de convection, les isovaleurs analytiques à l'extérieur de l'inclusion sont de plus en plus éloignées suivant l'axe Ox et de plus en plus rapprochées suivant l'axe Oy. C'est ce qui est observé numériquement,

avec ou sans réinitialisation. Simplement, la réinitialisation permet d'atténuer plus ou moins le rapprochement ou l'éloignement des isovalues, en fonction de l'épaisseur de lissage.

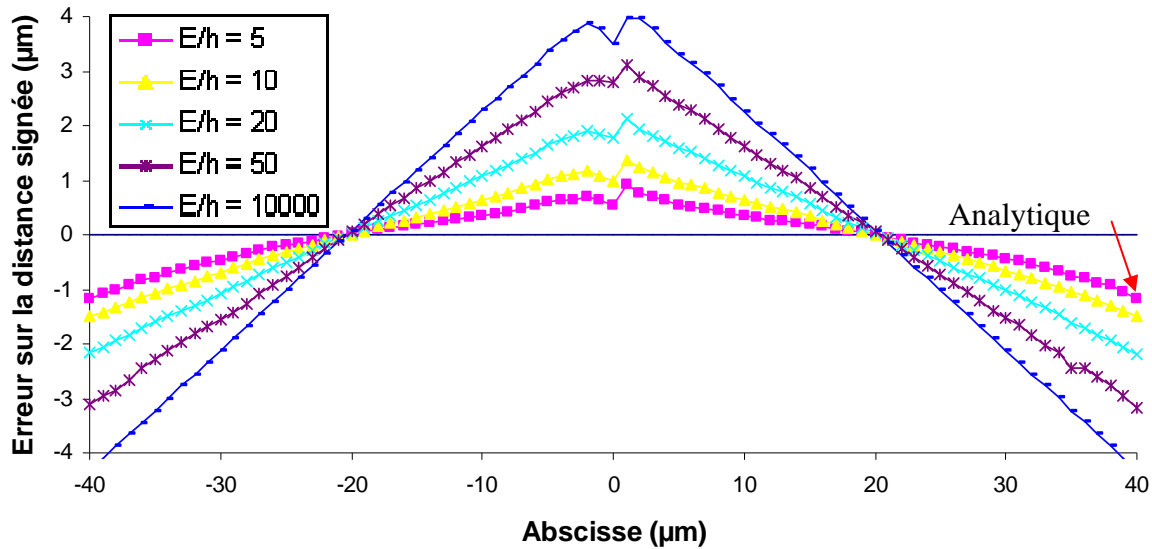


Figure 4.21. Erreur sur la distance signée le long de l'axe Ox pour différentes épaisseurs de lissage.

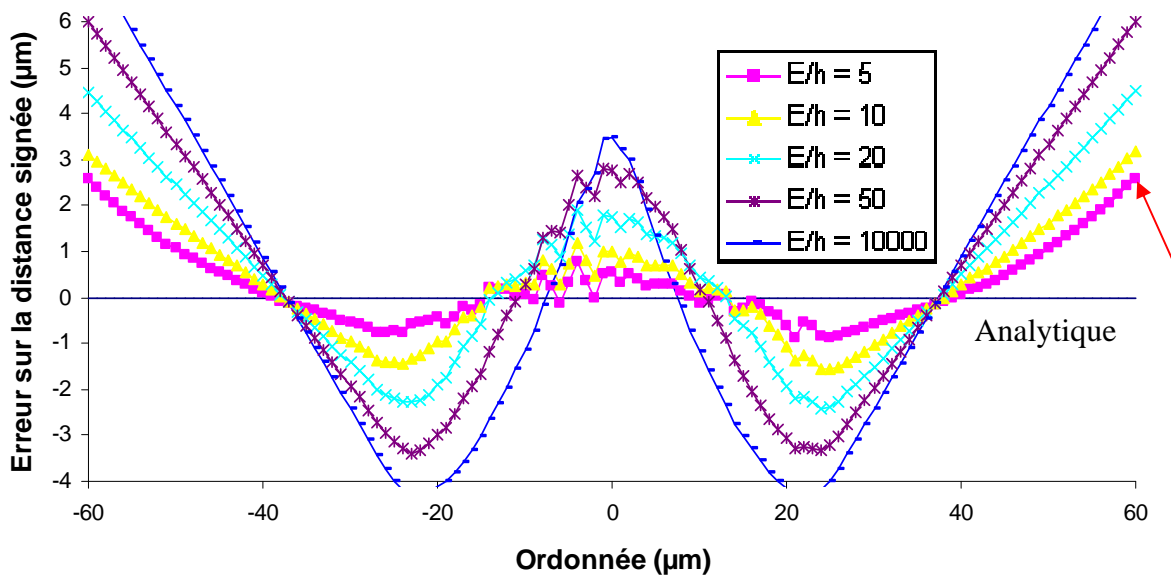


Figure 4.22. Erreur sur la distance signée le long de l'axe Oy pour différentes épaisseurs de lissage.

Nous pouvons tronquer le champ de vitesse pour mieux réinitialiser notre fonction level set. Au cours du calcul, la longueur du demi axe de l'inclusion suivant l'axe Ox décroît, sa valeur initiale et maximale étant de $25 \mu\text{m}$. Pour une abscisse de $30 \mu\text{m}$, le champ de vitesse suivant Ox vaut $1,5 \text{ nm}\cdot\text{s}^{-1}$. Donc si nous prenons $|v_x| \leq 1,5 \text{ nm}\cdot\text{s}^{-1}$ et $|v_z| \leq 1,5 \text{ nm}\cdot\text{s}^{-1}$, nous ne modifions pas le champ de vitesse dans le voisinage proche de l'interface tout au long du calcul et la forme finale de l'inclusion reste donc identique. En revanche, la figure 4.23, qui

est l'équivalent de la figure 4.21 avec une vitesse tronquée, nous montre que la distance signée numérique est plus proche de la distance signée analytique, en particulier pour les abscisses supérieures à 30 μm (abscisse à partir de laquelle la vitesse suivant Ox est tronquée) et pour les épaisseurs de lissage les plus importantes.

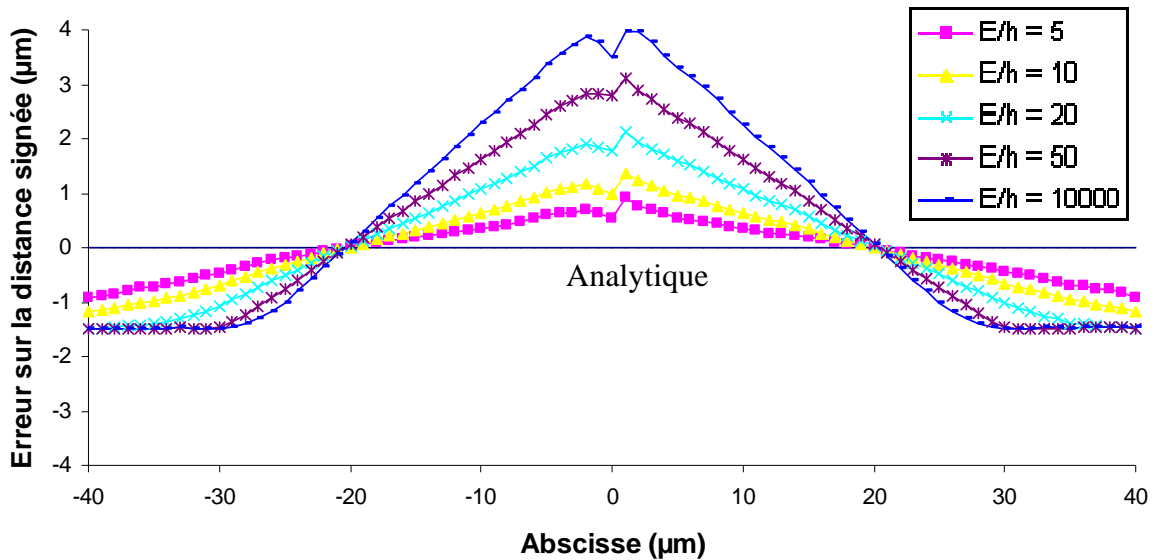


Figure 4.23. Erreur sur la distance signée le long de l'axe Ox pour différentes épaisseurs de lissage avec une vitesse tronquée.

Nous venons de voir de manière claire que modifier le champ de vitesse loin de l'interface peut permettre une meilleure réinitialisation de la level set. Pour cela, il semble logique de propager la vitesse à l'interface, comme suggéré par Tan & Zabaras sur des maillages rectangulaires et hexaédriques structurés [Tan 06]. Cependant, proposer un tel traitement sur des applications parallèles, basées sur des maillages 3D non structurés, pouvant présenter une forte anisotropie et quels que soient le champ de vitesse et les interfaces considérés, est loin d'être trivial.

Pour modifier la réinitialisation, il est aussi possible de changer la valeur du paramètre λ . Dans tous les calculs, nous avons pris $\lambda = h/\Delta t$. Diminuer cette valeur dégrade la réinitialisation et entraîne une meilleure conservation. Cela a donc le même effet qu'une augmentation de l'épaisseur de lissage. Au contraire, utiliser $\lambda > h/\Delta t$ dégrade la conservation mais améliore la réinitialisation. Cependant, ce résultat n'est observé dans notre cas que pour $\lambda < 1,8 h/\Delta t$. Il est tout à fait probable que cette limite varie si un autre pas de temps est considéré.

En conclusion, bien que conscient des limites de notre solveur de réinitialisation convective, nous utilisons par la suite, sauf précision contraire, une épaisseur de lissage de dix fois la taille de maille à l'interface, pour laquelle nous estimons qu'un bon compromis entre conservation du volume et qualité de la réinitialisation est obtenu. Nous utilisons également $\lambda = h/\Delta t$.

6.3. Remaillage

Nous avons pour le moment considéré un seul maillage pour l'ensemble de la simulation. Pour obtenir des résultats cohérents, nous avons utilisé la solution analytique afin de déterminer la configuration de l'inclusion dans la matrice au cours du temps. De manière très simple, nous avons alors employé un maillage fixe avec une taille de maille constante dans le voisinage de l'inclusion tout au long du calcul. Procéder de cette façon n'est pas optimal. En effet, il est possible de considérer une taille de maille constante dans le voisinage de l'inclusion en effectuant des remaillages, ce qui permet de diminuer l'épaisseur sur laquelle la taille est constante.

Pour le remaillage, nous adoptons la même formule donnant la taille de maille en fonction de la distance que celle utilisée au début de la simulation, ce qui permet de comparer les résultats avec et sans remaillage. Pour cela, il est nécessaire de recalculer la distance signée avant chaque remaillage. L'argument tangente hyperbolique n'étant pas défini pour des valeurs supérieures ou égales à un, nous devons alors faire attention au choix de l'épaisseur de lissage. En effet, prendre une épaisseur de lissage trop faible est impossible dans ce problème, puisque compte tenu de la précision machine limitée, la level set utilisée pour le LevelSet à une valeur égale à l'épaisseur de lissage loin de l'interface, et il est donc impossible de déterminer de nouveau une distance signée. Nous pourrions sans doute appliquer directement la taille maximale de $50 \mu\text{m}$ sur les nœuds pour lesquels la distance signée ne peut être obtenue. Cependant, cela pourrait entraîner une variation de la taille de maille moins régulière que dans les calculs sans remaillage. Nous nous contentons par conséquent d'utiliser une épaisseur de lissage de $10^{50} h$ (la level est alors une distance signée), pour laquelle ce problème est évité.

Les deux paramètres clefs de notre étude sont l'épaisseur et la période de remaillage. La figure 4.24 illustre le volume final de l'inclusion pour différentes valeurs de ces deux paramètres. Afin que le remaillage présente un intérêt, nous prenons un temps final de 12 000 s, pour lequel, avec l'épaisseur de $35 \mu\text{m}$, l'inclusion n'est plus entièrement contenue dans la sphère initiale de rayon $60 \mu\text{m}$ possédant une taille de maille de $2 \mu\text{m}$ à la fin du calcul. En effet, après 12 000 s, la longueur analytique du demi-axe suivant Oy est de $83 \mu\text{m}$.

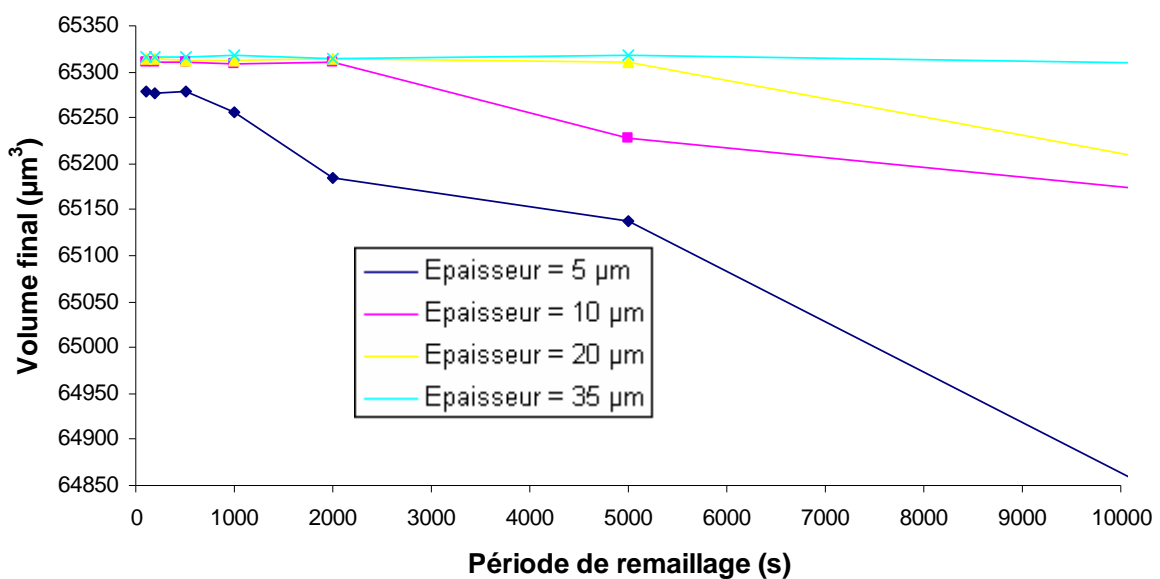


Figure 4.24. Volume final de l'ellipse pour différentes épaisseurs et périodes de remaillage.

Comme prévu, pour une épaisseur plus importante, un remaillage moins fréquent est nécessaire. Cependant, remailler plus souvent ne semble pas dégrader les résultats.

Nous traçons l'évolution du volume au cours de la simulation pour une période de remaillage de 5000 s et les 4 épaisseurs sur la figure 4.25. Pour l'épaisseur de 5 μm , nous notons une baisse du volume avant le premier remaillage entre 3000 et 5000s, ce qui signifie que l'interface n'est plus située dans une zone pour laquelle la taille de maille est de 2 μm . Le remaillage permet alors de remédier à cela. Cependant puisque la taille de maille à l'interface est modifiée, il en résulte une perte de volume, symbolisée par les traits verticaux de la figure 4.25. Après le remaillage, le volume se stabilise, jusqu'à ce que l'interface sorte de nouveau de la zone finement remaillée.

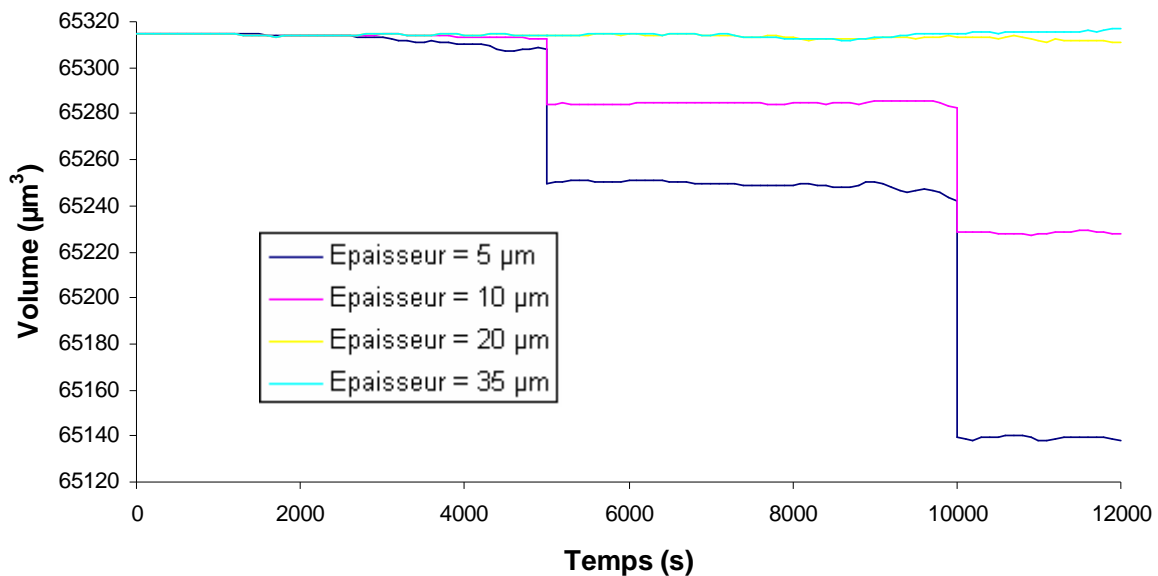


Figure 4.25. Evolution du volume de l'inclusion au cours du temps pour différentes épaisseurs de remaillage.

Il faut toutefois noter que la perte de volume avec l'épaisseur de 5 μm et la période de remaillage de 5000 s est relativement limitée, puisque nous ne calculons qu'une différence de 0,48 % par rapport au volume analytique (de 65 450 μm). A titre de comparaison, pour la simulation avec une épaisseur de 35 μm et une période de remaillage de 100 s, soit un remaillage à chaque incrément, la différence par rapport au volume analytique est de 0,20 %. La figure 4.26 révèle pourtant que le temps de calcul est très différent. Il s'agit respectivement de 1343 s (22 min) et 65 640 s (18 h), soit un rapport de 50 entre ces deux simulations. Nous remarquons toutefois que la différence entre les volumes numériques initial et final peut justifier l'emploi d'un calcul plus précis. En effet, le volume initial numérique de l'inclusion est de 65 315 μm pour l'ensemble des calculs, tandis que les volumes finaux pour les simulations durant 65 640 et 1343 s sont de 65 316 et 65 138 μm , ce qui représente respectivement une différence de 1 et de 177 μm . Cependant, vouloir à tout prix minimiser l'erreur faite au cours de la simulation au détriment du temps de calcul alors que l'erreur initiale de discrétisation de la sphère est déjà de 135 μm n'est pas forcément une bonne idée. Dans notre cas, le meilleur compromis semble être obtenu avec une épaisseur et une période de remaillage de 10 μm et 2000 s, puisque nous obtenons une différence entre les volumes analytique et numérique de 0,21 % en 3100 s (52 min). Les trois points utilisés dans ce paragraphe sont mis en avant sur la figure 4.26.

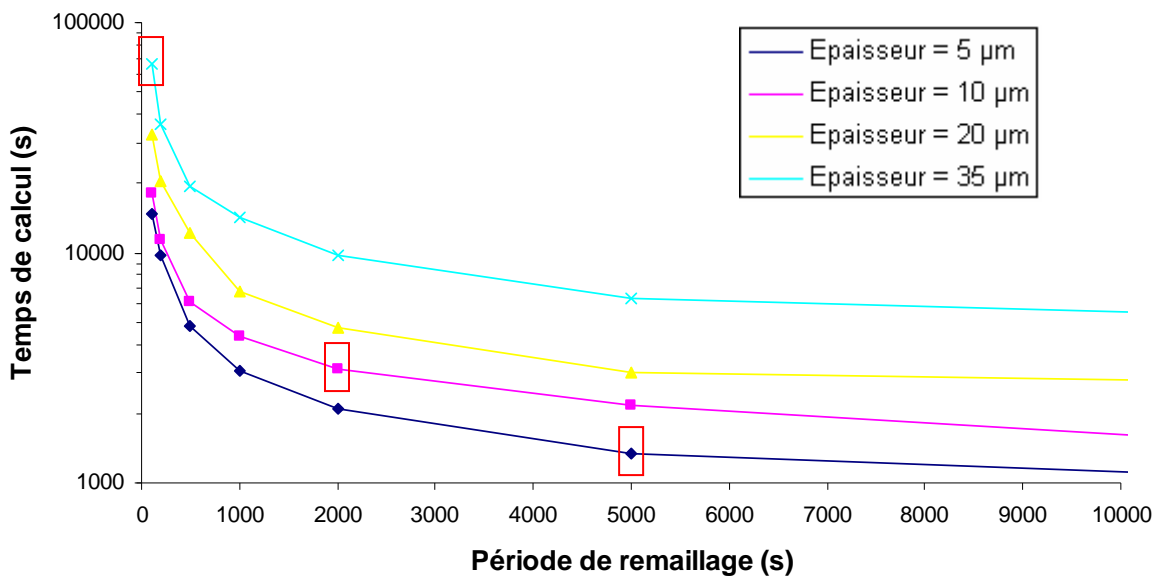


Figure 4.26. Temps de calcul en fonction de l'épaisseur et de la période de remaillage.

6.4. Maillage anisotrope

Jusqu'à présent, nous avons considéré des maillages isotropes, pour lesquels la taille des éléments est similaire dans toutes les directions. Pour essayer d'optimiser la résolution détaillée dans ce chapitre, il est possible d'avoir recours à des éléments anisotropes, possédant une taille différente suivant la direction. Cela devrait a priori nous permettre de réduire le temps de calcul à précision identique.

Coupez et Zerguine utilisent dans leurs applications un estimateur d'erreur pour capturer au mieux les interfaces lors de calculs dynamiques [Cou 11, Zer 10]. C'est ce que nous souhaitons également faire. Toutefois, comme les différents paramètres d'un tel outil ne sont pas faciles à calibrer, nous tenons à procéder par petites étapes. Nous étudions tout d'abord le comportement du LevelSet sur un maillage anisotrope n'évoluant pas au cours du temps. Ce maillage anisotrope est obtenu en utilisant la démarche détaillée dans la sous-partie 5.2 du deuxième chapitre et dans un article de Bernacki et al. [Ber 09]

Comme nous l'avons vu, la détermination de la taille de maille associée à ce type de remaillage passe par le calcul du gradient de la fonction distance. La distance étant définie aux nœuds, le gradient est défini aux éléments. Pour définir le gradient aux nœuds, nous effectuons une moyenne de la valeur sur l'ensemble des éléments entourant chaque nœud. Cependant, comme cela entraîne un manque de précision et que nous remaillons seulement sur la sphère initiale, nous préférons utiliser la valeur analytique du gradient donnée ci-dessous :

$$\frac{\partial \phi}{\partial x} = \frac{x}{r} ; \frac{\partial \phi}{\partial y} = \frac{y}{r} ; \frac{\partial \phi}{\partial z} = \frac{z}{r} \text{ avec } r = \sqrt{x^2 + y^2 + z^2} .$$

Nous définissons h_1 , la taille de maille dans la direction du gradient de la level set et h_2 , la taille de maille dans les deux autres directions. Ces deux valeurs sont utilisées pour définir le maillage à l'intérieur d'une sphère de 60 µm centrée sur l'origine du repère. A l'extérieur de cette sphère, la taille de maille est isotrope. Elle est calculée comme précédemment, en

utilisant la taille h_2 , pour opérer la jonction entre les deux zones. Une coupe 2D d'un tel maillage est montrée sur la figure 4.27. Dans ce cas, nous avons $h_1 = 1 \mu\text{m}$ et $h_2 = 8 \mu\text{m}$. La sphère rose de rayon $25 \mu\text{m}$ représente le contour initial de l'inclusion. Le trait noir est un cercle de rayon $60 \mu\text{m}$, délimitant la frontière entre les zones de remaillages anisotrope et isotrope.

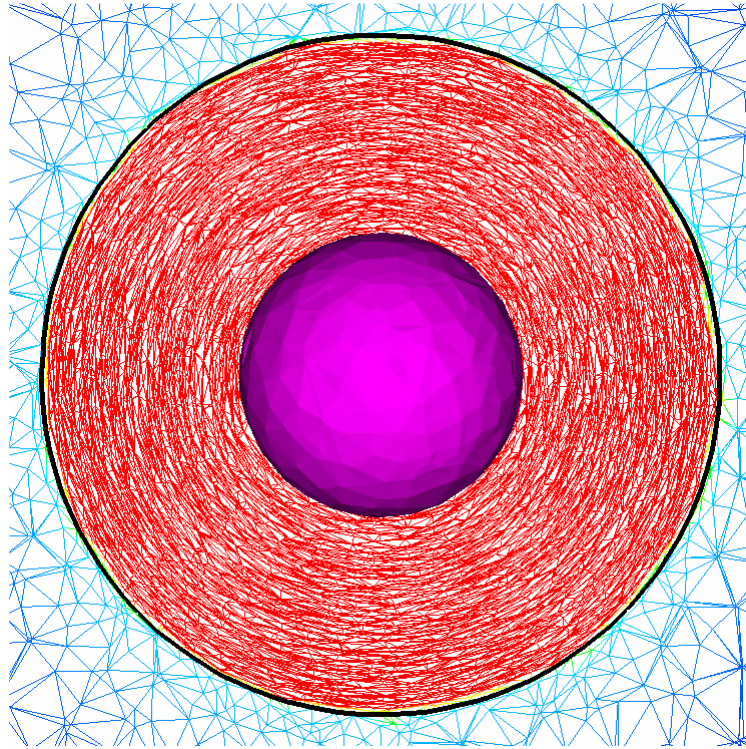


Figure 4.27. Coupe 2D montrant l'anisotropie des éléments autour de l'inclusion pour $h_1 = 1 \mu\text{m}$ et $h_2 = 8 \mu\text{m}$.

Le choix de l'épaisseur de lissage pose également problème avec un maillage anisotrope. En effet, il est commun d'utiliser une épaisseur proportionnelle à la taille des éléments à l'interface, mais cette taille varie suivant les directions. Nous devons par conséquent considérer soit une épaisseur très grande, soit une épaisseur proportionnelle à la taille de maille, dont la valeur optimale change vraisemblablement en fonction de l'étirement des éléments. Les résultats pour des épaisseurs égales à dix fois h_1 et dix fois h_2 ne montrent pas une amélioration de la conservation à temps de calcul équivalent pour des maillages anisotropes. Nous décidons de commenter seulement les résultats pour une épaisseur de lissage infinie (en pratique $10^{50} h_1$), pour laquelle aucune réinitialisation n'est faite, pour simplifier l'analyse.

Afin de respecter la condition CFL, le pas de temps doit s'adapter à la taille de maille minimale. Ainsi, nous choisissons respectivement $\Delta t = 50 \text{ s}$, 100 s , 150 s , 200 s , 250 s et 300 s pour $h_1 = 1 \mu\text{m}$, $2 \mu\text{m}$, $3 \mu\text{m}$, $4 \mu\text{m}$, $5 \mu\text{m}$ et $6 \mu\text{m}$.

Comme l'augmentation de la taille de maille entraîne une perte de volume de la sphère initiale, nous décidons de ne pas montrer la conservation pour chaque simulation mais plutôt de représenter les différents volumes obtenus, que nous pouvons comparer au volume analytique de la sphère. La figure 4.28 illustre le volume après 4000 s (après 4050 s pour $h_1 = 3 \mu\text{m}$) en fonction de h_1 et h_2 . En augmentant h_2 , le volume final s'éloigne de la solution analytique. En revanche l'augmentation de h_1 a un effet beaucoup moins important sur le

volume final, ce qui indique que le résultat est davantage influencé par la taille de maille maximale.

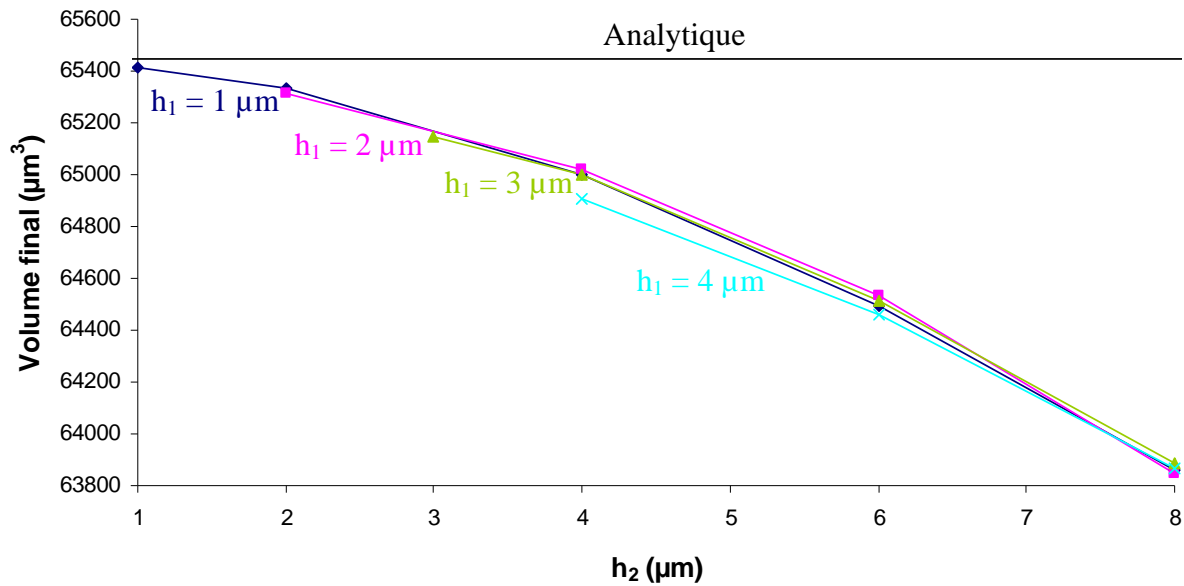


Figure 4.28. Volume final de l'inclusion en fonction des tailles de maille h_1 et h_2 .

Ce constat ne plaide pas en faveur de l'utilisation d'un maillage anisotrope. Cependant, il nous manque encore une donnée importante pour chaque simulation, qui n'est autre que le temps de calcul.

Ainsi, la figure 4.29 donne pour chaque simulation le volume final de l'inclusion en fonction du temps de calcul associé. Sur cette figure, les points rouges représentent les simulations avec un maillage isotrope, pour lesquelles $h_1 = h_2$. Les points situés à gauche (ou au-dessus) de ce trait correspondent alors à des simulations pour lesquelles le compromis entre le temps de calcul et la conservation du volume de l'inclusion est meilleur que pour les simulations à maillage isotrope. Or, aucun point symbolisant les simulations à maillage anisotrope n'est situé dans ce domaine.

Nous déconseillons par conséquent vivement l'utilisation d'un maillage anisotrope calculé à partir d'une fonction distance pour simuler avec le LevelSet le déplacement d'interfaces soumises à un champ de vitesse issu d'un test de traction uniaxial.

Nous tenons également à apporter une réponse à l'objection selon laquelle l'anisotropie ne dégrade la qualité de nos simulations que parce que nous considérons un maillage fixe, pour lequel les éléments ne sont plus adaptés pour définir l'évolution de la fonction level set au cours du temps. Pour cela, nous utilisons le remaillage adaptatif pour la simulation avec $h_1 = 2 \mu\text{m}$ et $h_2 = 8 \mu\text{m}$. Le gradient de la level set ne peut alors plus être calculé à l'aide de la formule analytique. Nous remarquons que le volume final obtenu après 12 000 s est moins élevé que celui à maillage fixe, et ce quelle que soit la période de remaillage. En outre, plus les remaillages sont fréquents, plus la perte de volume est importante.

Cela s'explique par une différence fondamentale entre le remaillage isotrope de la sous-partie précédente et le remaillage anisotrope utilisé ici. Si le remaillage isotrope est effectué au bon moment, la taille de maille à l'interface reste constante et les nœuds sont dans le pire des cas seulement très légèrement déplacés, ce qui n'entraîne qu'une modification minimale du volume de l'inclusion. Au contraire, avec le remaillage anisotrope, la taille de maille varie pour chaque remaillage, car elle est fonction du gradient de la level set, dont le calcul au nœud est

de plus relativement imprécis. La position des nœuds change par conséquent beaucoup plus à chaque remaillage, ce qui provoque une importante perte de volume.

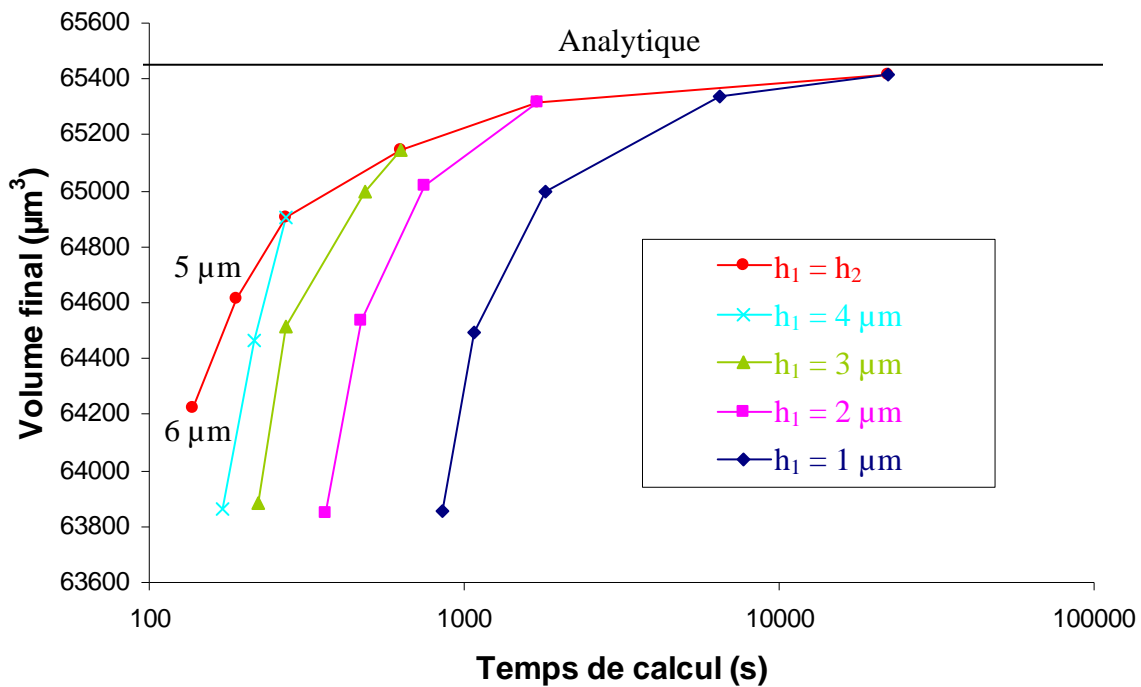


Figure 4.29. Temps de calcul en fonction des tailles de maille h_1 et h_2 .

Nous n'avons pas encore donné d'explication quant à l'inefficacité du maillage fixe anisotrope. Deux nous semblent pourtant évidentes. Nous observons tout d'abord que la sphère rose de la figure 4.27 semble mal définie malgré une taille de maille minimale de $1 \mu\text{m}$. Pour ce type de géométrie, il semblerait plus logique d'utiliser la courbure de l'interface pour définir un maillage adapté. La courbure est cependant d'abord calculée sur chaque élément à partir du gradient de la level set, puis elle est réinterpolée aux nœuds, avec une perte de précision. Un remaillage avec une telle fonction entraîne donc des pertes de volume importantes.

L'autre explication est la suivante : le maillage étant seulement adapté avec la fonction level set, il ne correspond pas nécessairement au champ de vitesse appliqué. La taille de maille dans le sens de l'écoulement peut ainsi varier de h_1 à h_2 , ce qui est peut-être préjudiciable pour la précision de nos simulations.

Pour confirmer cette hypothèse, nous considérons les maillages anisotropes déjà utilisés pour le test de traction de l'inclusion, mais cette fois-ci avec un champ de vitesse égal au gradient analytique de la level set. Le rayon de la sphère augmente alors d' $1 \mu\text{m}$ à chaque s. Nous arrêtons la simulation pour un temps de 30 s et le pas de temps est pris proportionnel à h_1 . Nous avons $\Delta t = 0,5 \text{ s}$ pour $h_1 = 1 \mu\text{m}$. Les contours initial et final obtenus avec $h_1 = h_2 = 4 \mu\text{m}$ sont montrés sur la figure 4.30.

Nous comparons les volumes numériques au volume analytique et obtenons des résultats semblables à ceux obtenus pour le champ de vitesse de traction. Comme sur la figure 4.28, h_1 ne semble pas beaucoup influencer le volume et comme sur la figure 4.29, nous obtenons des temps de calcul supérieurs pour les maillages anisotropes à volume final équivalent.

Par conséquent, si les maillages anisotropes ne donnent pas satisfaction dans notre cas, il semble bien que c'est à cause d'un problème de discrétisation initiale de la sphère. A la lumière des résultats obtenus, nous décidons de renoncer à l'utilisation de maillages

anisotropes par la suite. Cela signifie que nous renonçons également à l'utilisation des estimateurs d'erreur présentés par Zerguine et Coupez [Zer 10, Cou 11].

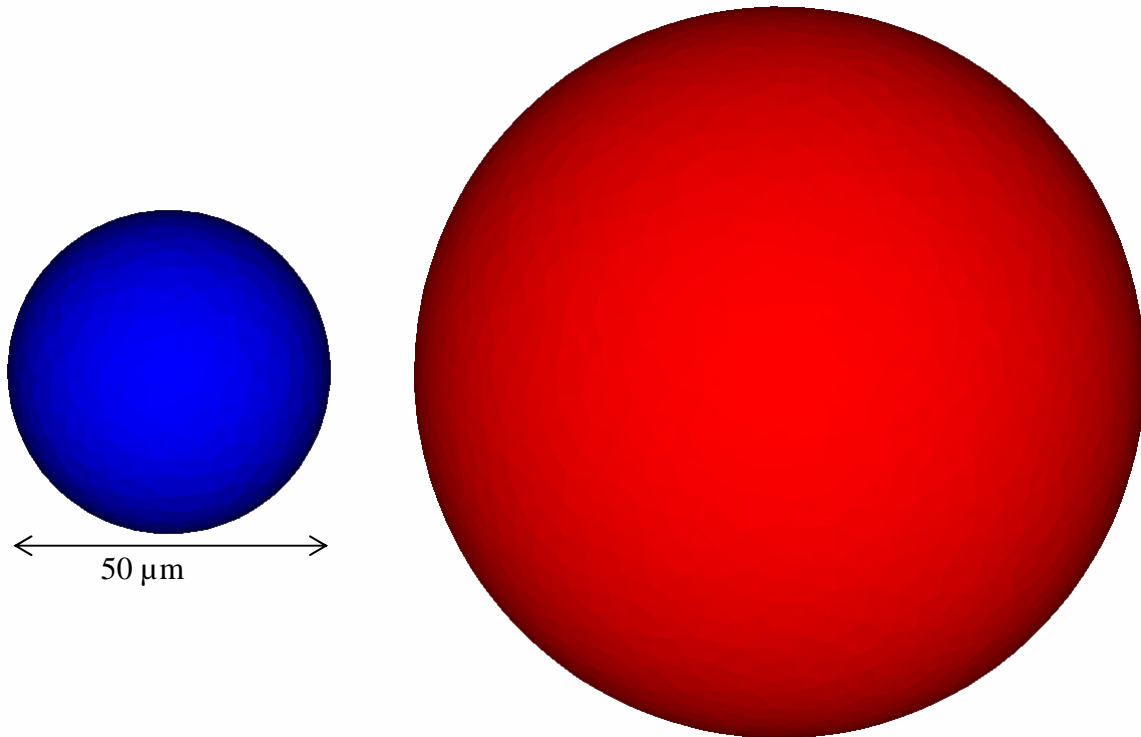


Figure 4.30. Contour de l'inclusion au début et à la fin de la simulation pour $h_1 = h_2 = 4 \mu\text{m}$.

6.5. Plusieurs inclusions

Après avoir mené plusieurs tests avec une seule inclusion, nous tenons à nous assurer que prendre en compte plusieurs inclusions très proches les unes des autres ne pose aucun problème. Pour cela, nous positionnons sept inclusions dans la matrice. Nous situons une inclusion au centre du repère, comme précédemment. Les six autres sont accolées à l'inclusion centrale et leur centre est situé sur les trois axes du repère orthonormé. Comme dans la sous-partie 6.3, nous utilisons une taille de maille minimale de $2 \mu\text{m}$ dans le voisinage des inclusions avec un remaillage adaptatif isotrope et une épaisseur de lissage de $10^{50} h$. Le résultat numérique en terme de conservation et des longueurs des axes est très proche de la solution analytique. Le contour des inclusions avant et après déformation est montré sur la figure 4.31.

Nous faisons aussi des tests en positionnant une inclusion près du bord supérieur de la matrice, afin d'étudier le comportement du solveur quand un objet sort du domaine de calcul. Cette inclusion est présentée à différents temps sur la figure 4.32. Nous obtenons également dans ce cas un bon accord entre les résultats numériques et la solution analytique.

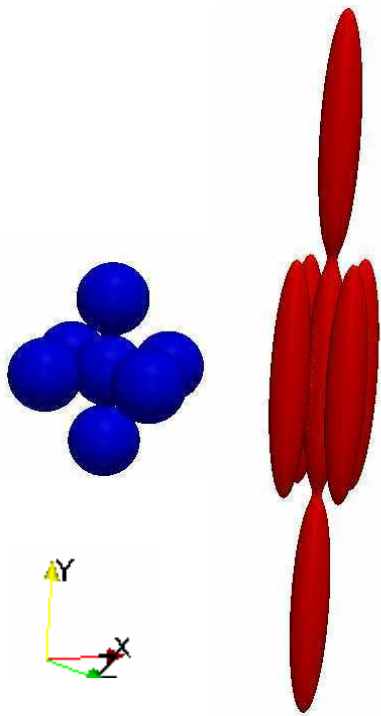


Figure 4.31. Contour des inclusions. Contour initial en bleu et contour final en rouge.

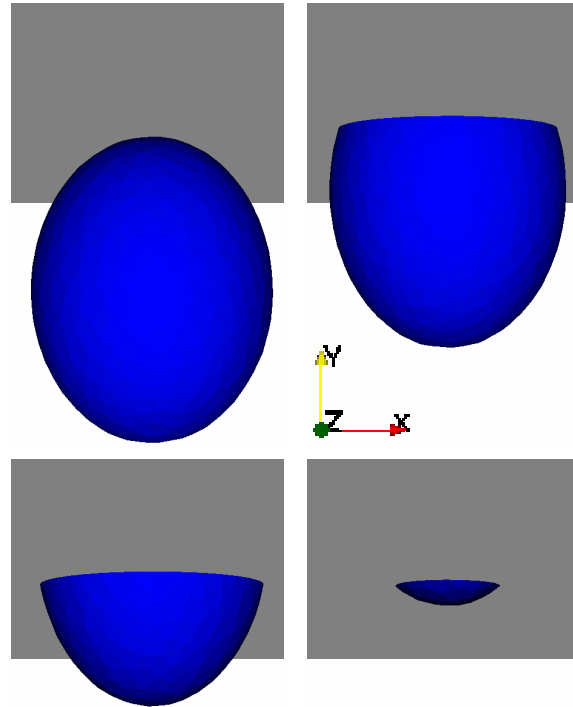


Figure 4.32. Contour d'une inclusion sortant du domaine de calcul. Plan supérieur du domaine en gris. La position de l'inclusion est montrée à quatre différents temps.

6.6. Période de réinitialisation de GMRES

En 3D, la résolution du système linéaire associé à un solveur de réinitialisation convective dans CIMLib est parfois très instable. Comme nous avons remarqué que le LevellerT est moins sujet à cette observation sur des cas simples, nous utilisons le LevellerS pour illustrer ce phénomène.

Nous lançons donc le même calcul que dans la sous-partie 6.1 en prenant une taille de maille minimale de $8 \mu\text{m}$, un pas de temps de 400 s et une épaisseur de lissage de $80 \mu\text{m}$ (soit 10 fois la taille de maille minimale). Le maillage considéré comprend 19 893 nœuds, ce qui permet d'avoir un temps de calcul beaucoup plus faible qu'en prenant une taille minimale de $2 \mu\text{m}$, taille pour laquelle la résolution du système linéaire est également instable avec le LevellerS. Nous souhaitons une norme finale du résidu en dessous de 10^{-8} et fixons le nombre maximal d'itérations à 5000.

Les figures 4.33, 4.34 et 4.35 représentent respectivement le nombre d'itérations, le temps de résolution ainsi que les normes du résidu finale et initiale associés à chaque système linéaire.

Chapitre 4

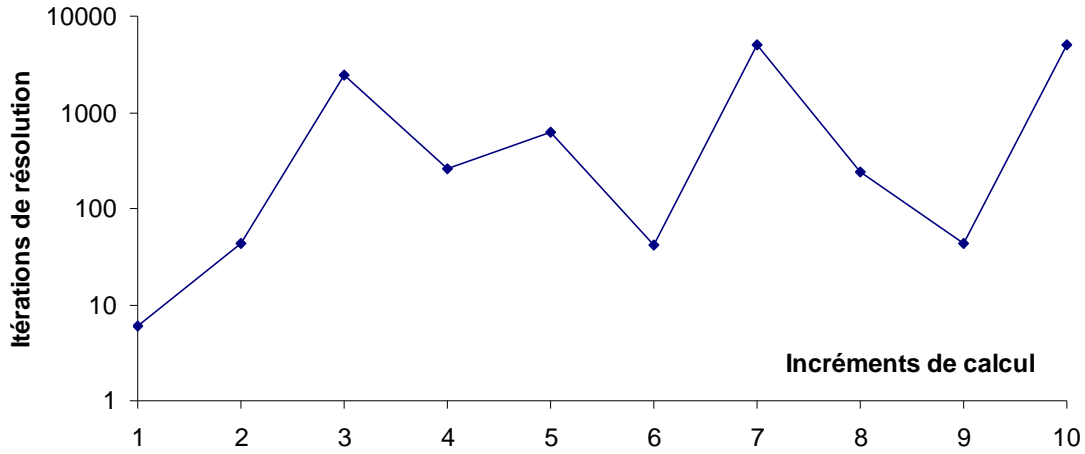


Figure 4.33. Nombre d'itérations de résolution du système linéaire pour les différents incréments de calcul.

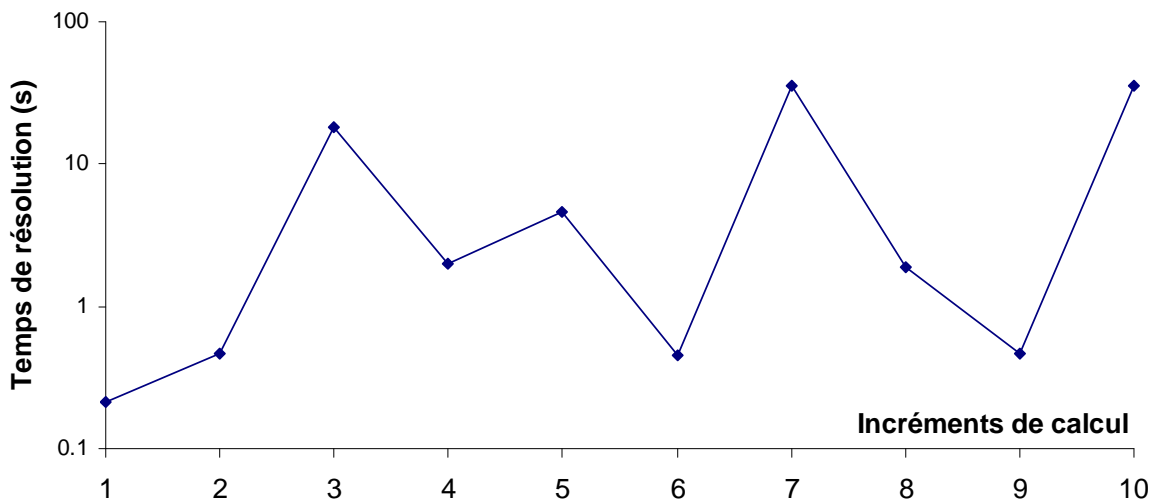


Figure 4.34. Temps de résolution du système linéaire pour les différents incréments de calcul.

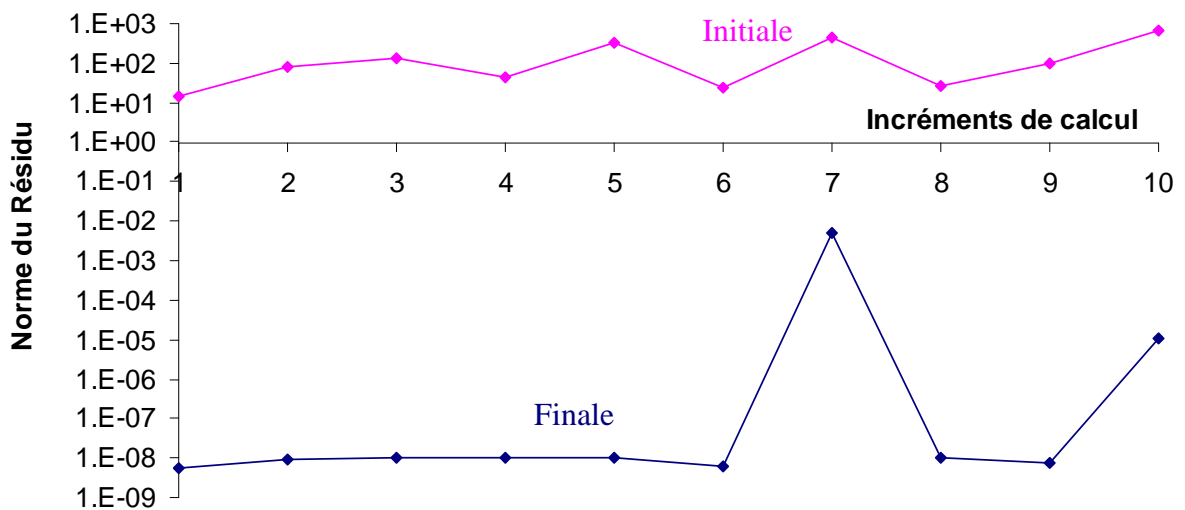


Figure 4.35. Normes initiale et finale du résidu des systèmes linéaires pour les différents incréments de calcul.

Ces calculs sont effectués avec l'option par défaut de PETSc, pour laquelle la période de réinitialisation de la base de vecteurs utilisée pour la méthode GMRES [Saa 86] est de 30 itérations. Comme les figures ci-dessus l'illustrent, le nombre d'itérations nécessaire à la résolution des systèmes linéaires peut tantôt être très faible, tantôt atteindre la limite du nombre maximal demandé, ce qui signifie que la norme finale du résidu est au dessus de celle souhaitée, occasionnant ainsi un temps de calcul beaucoup plus important. Cela peut aussi avoir pour conséquence de donner une solution erronée, et c'est ce que nous observons pour une taille de maille plus petite.

Pour pallier ce problème, nous avons rajouté dans CIMLib une option qui permet de changer la période de réinitialisation de la base de vecteurs. Nous utilisons cette option pour résoudre le septième système linéaire, pour lequel le nombre maximal d'itérations est atteint avec une période de 30 itérations. La figure 4.36 illustre l'évolution de la norme du résidu au cours de la résolution pour différentes périodes de réinitialisation et la figure 4.37 le temps de résolution pour chaque période. Pour des périodes trop faibles, la norme du résidu n'atteint jamais la valeur souhaitée et réinitialiser la base de vecteurs n'a aucun intérêt. Pour des périodes suffisamment importantes, une seule réinitialisation suffit pour obtenir la convergence. Nous constatons qu'après la réinitialisation, la norme du résidu subit une augmentation, due au recalcul du vecteur résidu par la méthode initiale (présentée dans le chapitre précédent), puis elle décroît très vite. D'après le tableau 4.1, qui donne la valeur de la norme du résidu et le nombre d'itérations nécessaire pour la convergence après réinitialisation, il est inutile de prendre une période trop importante. En effet la résolution est quasiment identique quelle que soit la période de réinitialisation, tandis que le temps de calcul augmente si la période est supérieure.

Dans ce cas, considérer une période de réinitialisation de 100 itérations est beaucoup plus efficace. Cependant, nous trouvons une valeur optimale de la période différente pour d'autres systèmes linéaires. Idéalement, nous pensons que la période de réinitialisation de la base de vecteurs devrait s'appuyer sur un critère prenant en compte l'évolution du vecteur résidu au cours des itérations. Une telle procédure est d'ailleurs présentée par Joubert [Jou 94].

Période de réinitialisation	Norme du résidu après réinitialisation (* 10 ⁵)	Nombre d'itérations nécessaire pour la convergence, après réinitialisation
100	9,874358	71
200	9,864202	70
500	9,862052	70
1000	9,861549	72
2000	9,861325	71

Tableau 4.1. Effet de la réinitialisation sur la norme du résidu et sur le nombre d'itérations en fonction de la période de réinitialisation.

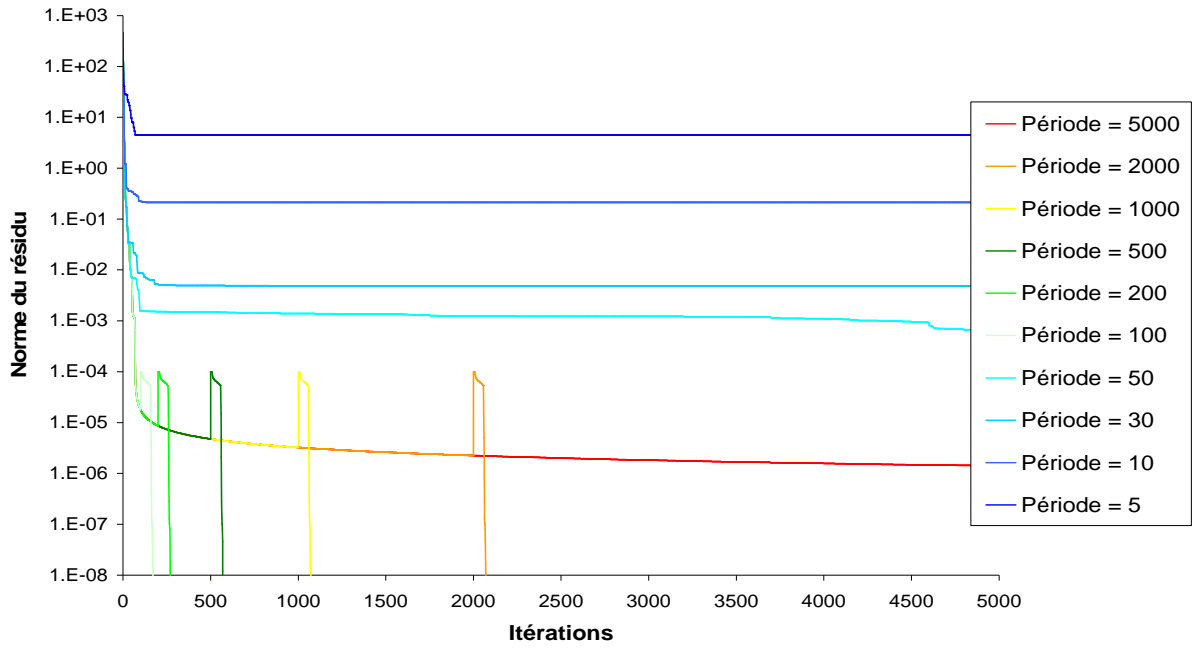


Figure 4.36. Evolution de la norme du résidu au cours de la résolution pour différentes périodes de réinitialisation de la base de vecteurs.

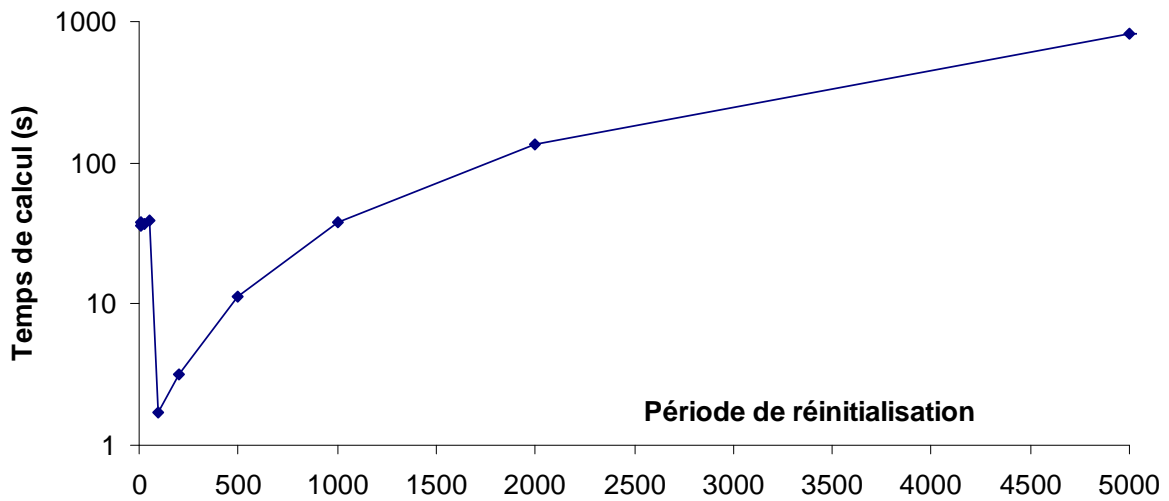


Figure 4.37. Temps de résolution du système linéaire en fonction de la période de réinitialisation de la base de vecteurs.

7. Conclusion

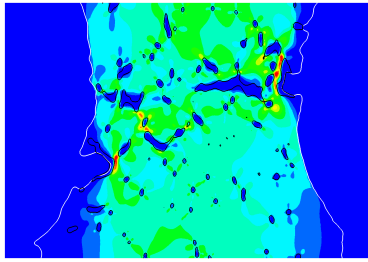
Nous avons décrit dans ce chapitre la technique de déplacement des interfaces utilisée dans CIMLib. Pour cela, nous avons détaillé la méthode employée pour résoudre les équations de convection, de réinitialisation et de réinitialisation convective. Nous avons également expliqué le fonctionnement de la méthode level set particulière (PLS), ce qui nous a permis d'effectuer des comparaisons avec l'approche originale développée au CEMEF (solveur LevellerT). Pour les cas simples testés, ces comparaisons ont été favorables à la méthode PLS, laquelle est beaucoup plus rapide que le LevellerT à précision équivalente.

Le LevellerT souffre en effet de plusieurs limitations. La réinitialisation n'est pas satisfaisante et elle entraîne une diffusion de la position de l'interface, avec en outre un pas de temps dont il n'est pas aisé de déterminer la valeur optimale. Nous rappelons également que pour déterminer la nouvelle valeur de la fonction level set, nous devons assembler et résoudre un système linéaire avec une matrice globale non symétrique et dont le nombre d'inconnues est égal au nombre de nœuds du maillage, et ce quel que soit le périmètre (ou la surface en 3D) de l'interface. Le lissage de la level set en tangente hyperbolique permet donc de limiter les instabilités entrevues sur la figure 4.5 par l'apport d'un lissage du gradient de la level set et de nous sécuriser sur la non imposition de conditions aux limites lorsque la valeur de la level set au bord du domaine est égale à sa borne maximale (respectivement minimale). Cependant, cette méthode de lissage, appelée level set locale par Ville [Vil 11b], ne permet pas de réduire l'effort de calcul nécessaire, contrairement par exemple à la méthode PLS, pour laquelle la level set n'est calculée que dans le voisinage proche de l'interface.

Pour rendre le LevellerT plus performant, nous proposons les améliorations suivantes. Pour obtenir une meilleure conservation, le couplage avec une méthode VOF [Sus 00], particulière [Enr 05] ou utilisant des marqueurs [Mih 07], semble tout indiqué. Concernant la réinitialisation, nous avons montré qu'une extension du champ de vitesse serait intéressante à intégrer [Ada 99, Tan 06]. Nous suggérons également d'ajouter une option permettant de réinitialiser la fonction level set par la méthode "brute force redistancing" [Hys 05]. Hormis cette dernière méthode, pour laquelle les outils sont déjà présents dans CIMLib, comme expliqué dans le deuxième chapitre, les autres améliorations proposées nécessiteraient une réflexion importante pour être intégrées à notre bibliothèque de calcul. En effet, l'extension du champ de vitesse et les méthodes level set particulières n'existent pas à notre connaissance dans la littérature pour des maillages tétraédriques anisotropes. En revanche, la méthode CLSVOF a déjà été étudiée pour des maillages tétraédriques [Lv 10] et la méthode VOF non couplée existe dans CIMLib [Bru 04].

Malgré les limitations actuelles du LevellerT, ce solveur est actuellement le plus performant dans la librairie CIMLib pour effectuer des calculs parallèles de déplacements d'interfaces couplés avec la résolution mécanique décrite dans le chapitre précédent. Puisque nous allons l'utiliser par la suite, nous retenons que des solutions analytiques ont été retrouvées avec succès. Nous listons à présent les conditions d'utilisation retenues : maillages isotropes, épaisseur de lissage de la level de dix fois la taille de maille à l'interface et pas de temps respectant la condition CFL. Avant de pouvoir utiliser des maillages anisotropes pour nos applications, nous recommandons d'étudier les outils de Coupe sur les cas test 3D présentés par Bui et al. [Bui 10]

Dans l'ensemble de ce chapitre, nous avons seulement traité des cas pour lesquels deux phases sont présentes. Pour traiter les problèmes impliquant trois phases du chapitre suivant, nous utilisons alors deux fonctions level set, tout en sachant que cela peut créer des problèmes aux joints triples [Zha 96, Smi 02].



- Chapitre 5 - Applications

« The right to search for truth implies also a duty; one must not conceal any part of what one has recognized to be true. »

Albert Einstein

1. Introduction	128
2. Test de l'inclusion	128
3. Calculs sur une image 2D	131
3.1. Représentation par éléments finis	131
3.2. Comportement newtonien	132
3.3. Comportement viscoplastique	139
3.4. Influence du pas de temps et de la taille de maille	143
4. Calculs sur une image 3D	147
5. Conclusion	155

1. Introduction

Notre objectif dans ce chapitre est de réaliser une simulation numérique représentative de l'expérience de traction menée par Terzi et al. [Ter 09] sur un échantillon cylindrique d'Al-8% pds Cu à l'état pâteux. Nous disposons pour cela d'une série d'images de microtomographie segmentées fournies par nos partenaires du SIMaP, dans lesquelles les phases gazeuse, solide et liquide ont été séparées. Ces images nous permettent d'initialiser nos simulations numériques, en utilisant les outils décrits dans le deuxième chapitre, et également de comparer la morphologie expérimentale à celle que nous obtenons numériquement. Deux résultats importants montrés par les images expérimentales de Terzi et al. sont l'accumulation de liquide aux surfaces intergranulaires perpendiculaires à l'axe de traction, notamment au milieu de l'éprouvette, ainsi que la propagation de l'air dans l'échantillon à partir de sa surface extérieure [Ter 09].

Notre but n'est pas facile à atteindre car pour être représentatif de l'expérience, nous avons besoin d'employer un maillage très volumineux sur lequel les calculs sont complexes. Dans un premier temps, nous devons coupler les solveurs mécanique et level set respectivement décrits dans les troisième et quatrième chapitres. Pour cela, nous utilisons de nouveau le test de l'inclusion. Nous utilisons ensuite une coupe 2D de l'image expérimentale pour faire des analyses nous permettant d'étudier l'influence des différents paramètres de la simulation, ce qui nous permet finalement de réaliser un calcul en trois dimensions, pour lequel nous obtenons une comparaison avec les résultats expérimentaux.

2. Test de l'inclusion

Nous considérons à nouveau le test de l'inclusion en couplant cette fois-ci les résolutions mécanique et level set. La solution analytique nous est donnée par Gilormini & Montheillet [Gil 86]. Même en utilisant le logiciel Matlab, la forme de cette solution est trop complexe pour obtenir une intégration exacte nous donnant la longueur du demi-grand axe en fonction du temps et des différents paramètres : rayon initial de l'inclusion sphérique, rapport des viscosités, vitesse imposée sur les bords du domaine. Nous utilisons par conséquent une intégration numérique. Cependant, puisque nous voulons étudier dans cette partie l'influence du rapport de viscosité sur la résolution, plutôt que la longueur du demi-grand axe nous préférons considérer le volume final de l'inclusion, dont la valeur analytique reste constante quel que soit le rapport de viscosité.

Nous résolvons ici les équations de Stokes $\nabla \cdot \bar{\sigma} = \bar{0}$ et $\nabla \cdot \bar{v} = 0$ dans une configuration newtonienne, avec la méthode présentée dans le troisième chapitre et en mélangeant la viscosité des deux phases : $\eta = \tau \eta_M + (1 - \tau) \eta_I$ avec η , η_M et η_I respectivement la viscosité résultant du mélange et les viscosités constantes de la matrice et de l'inclusion. τ est le taux de remplissage de l'élément. Le champ de vitesse est recalculé à chaque pas de temps et utilisé pour actualiser la fonction level set grâce au solveur LevellerT, détaillé dans le chapitre précédent.

La figure 5.1 montre le volume de l'inclusion après 4000 s, en utilisant le même maillage que celui utilisé dans la sous partie 6.1 du chapitre précédent, un pas de temps de 100 s, une viscosité de la matrice de 10 GPa.s et une épaisseur de lissage de la level set de 20 μm , soit 10 fois la taille de maille à l'interface. La vitesse imposée au bord du domaine et le rayon initial de l'inclusion (25 μm) sont également ceux employés au chapitre précédent. Nous représentons le volume final pour trois précisions de résolution des systèmes linéaires

associés aux résolutions de Stokes et pour des rapports de viscosité entre la matrice et l'inclusion allant de 1 à 10^{13} . Dans cette figure, la viscosité de l'inclusion varie donc de 10 GPa.s à 1 mPa.s, qui est approximativement la valeur réelle de viscosité du métal liquide à considérer pour nos applications.

Cette figure nous permet de faire plusieurs observations. Tout d'abord le volume final de l'inclusion est plus éloigné du volume initial pour un rapport de viscosité de 10 que pour un rapport unitaire et il l'est encore plus pour un rapport de 100. Il semble ensuite se stabiliser. Nous voyons trois raisons pouvant expliquer cela : (i) pour un rapport unitaire, la viscosité reste constante sur tout le domaine au cours du temps et seule une résolution des équations de Stokes est donc nécessaire, (ii) la matrice n'étant pas de taille infinie, la meilleure solution numérique ne correspond pas tout à fait à la solution analytique donnée par Gilormini & Montheillet [Gil 86] lorsque le rapport de viscosité n'est pas unitaire, (iii) plus le rapport de viscosité augmente, plus l'inclusion se déforme rapidement et le pas de temps adapté pour le rapport de viscosité unitaire du quatrième chapitre ne l'est donc plus tout à fait ici.

Les autres observations sont à rapprocher de celles faites lors de l'étude sur le rapport de viscosité et la précision de la résolution du système linéaire faite dans le troisième chapitre. En effet, pour des rapports de viscosité entre 1 et 10^4 , le volume final est le même pour les trois précisions. Lorsque le rapport de viscosité augmente et que la norme finale du résidu exigée n'est plus suffisamment faible (par exemple pour la précision 10^{-3} et le rapport 10^6), nous remarquons tout d'abord que le volume de l'inclusion diminue. Les résultats sont alors moins précis mais toujours cohérents. En augmentant encore le rapport de viscosité, l'erreur sur la résolution mécanique devient très importante et l'inclusion n'est plus déformée correctement.

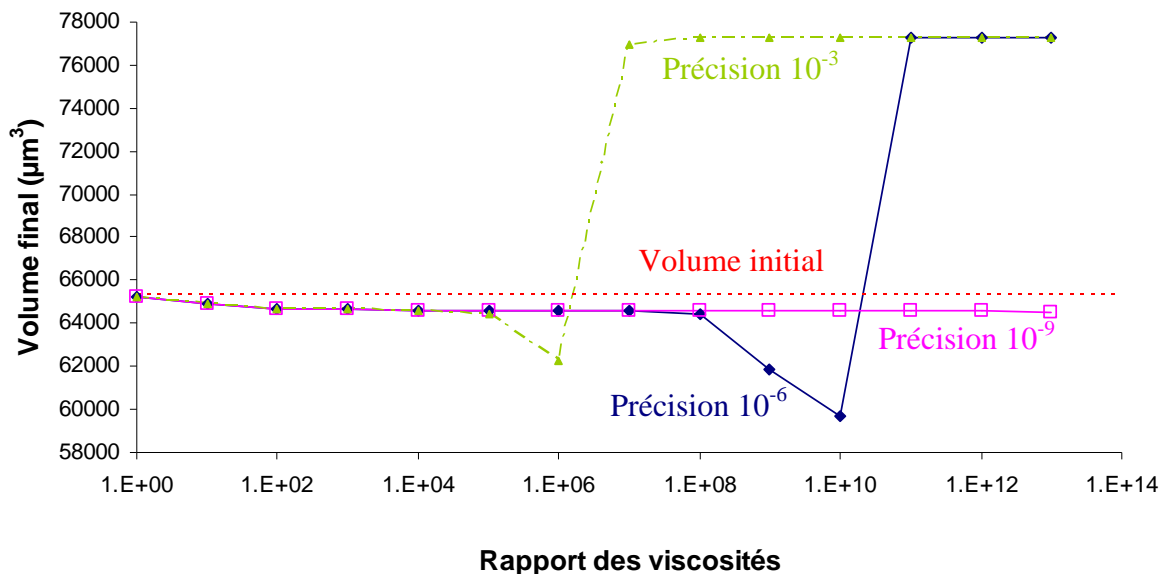


Figure 5.1. Volume final de l'inclusion en fonction du rapport de viscosité et de la précision de la résolution du système linéaire.

Plus la norme du résidu final exigée est faible et plus le temps de résolution des systèmes linéaires est important. Il est donc important de déterminer la précision (ou norme) requise en fonction du rapport de viscosité considéré. Fort heureusement, nous nous rendons compte qu'il est en pratique inutile de considérer un rapport de viscosité trop important, puisqu'à partir de 10^6 , notre solution analytique (intégrée numériquement) ne varie quasiment plus. Ce

constat est illustré sur la figure 5.2, qui nous montre la longueur analytique du demi-grand axe de l'inclusion en fonction du rapport de viscosité.

Nous avons vu dans le chapitre précédent que l'épaisseur de lissage de la level set est un paramètre très important lorsque seul le solveur de déplacement d'interface est utilisé. En effet, une faible épaisseur permet une meilleure réinitialisation au détriment de la conservation, tandis qu'une épaisseur plus élevée entraîne une meilleure conservation mais une moins bonne réinitialisation. Nous observons le même résultat lorsque le solveur d'interface est couplé avec le solveur de Stokes. La figure 5.3 illustre d'ailleurs l'évolution du volume de l'inclusion au cours du temps en fonction de l'épaisseur de réinitialisation pour un rapport de viscosité de 10^6 . Comme attendu, la conservation est meilleure pour les plus grandes épaisseurs mais la réinitialisation moins bonne.

Il existe cependant une différence importante par rapport au chapitre précédent. Dans ce cas, en raison du calcul mécanique, la conservation est beaucoup moins bonne même lorsque la level set correspond à une distance signée. Ainsi avec une épaisseur de lissage de 10^{50} h (la level set est alors une distance signée), nous obtenons une perte de volume de 0,80 %. Elle est de 1,06 % avec une épaisseur de 10 h. La différence entre ces deux conservations est par conséquent beaucoup plus faible que sans calcul mécanique, puisque nous avons obtenu un rapport de 70 entre les deux (cf sous partie 6.2 du chapitre précédent). Pour cette raison, nous considérons par la suite des épaisseurs de lissage de la level set égales à 10 tailles de maille.

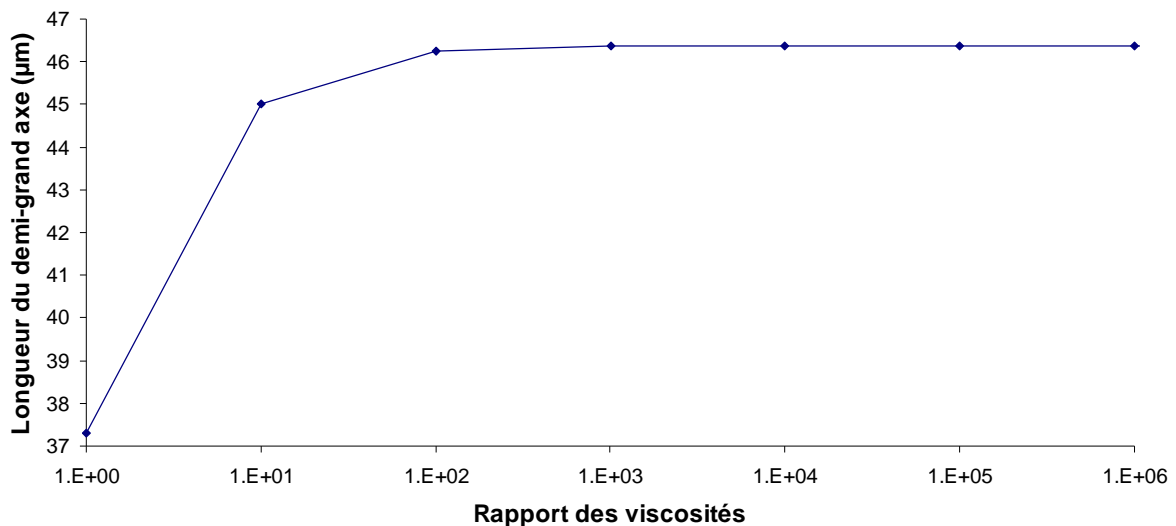


Figure 5.2. Longueur analytique du demi-grand axe de l'inclusion après 4000 s en fonction du rapport de viscosité.

Cette étude sur la déformation d'une inclusion sphérique nous permet ainsi de tirer la conclusion suivante. Pour cette application, il est inutile de considérer un rapport de viscosité supérieur à 10^6 . De plus, ce constat est valable à la fois analytiquement et numériquement. Dans la prochaine partie, nous traitons de la déformation de la coupe 2D d'une image obtenue par microtomographie, afin d'étudier le rôle joué par le rapport de viscosité dans un cas plus complexe.

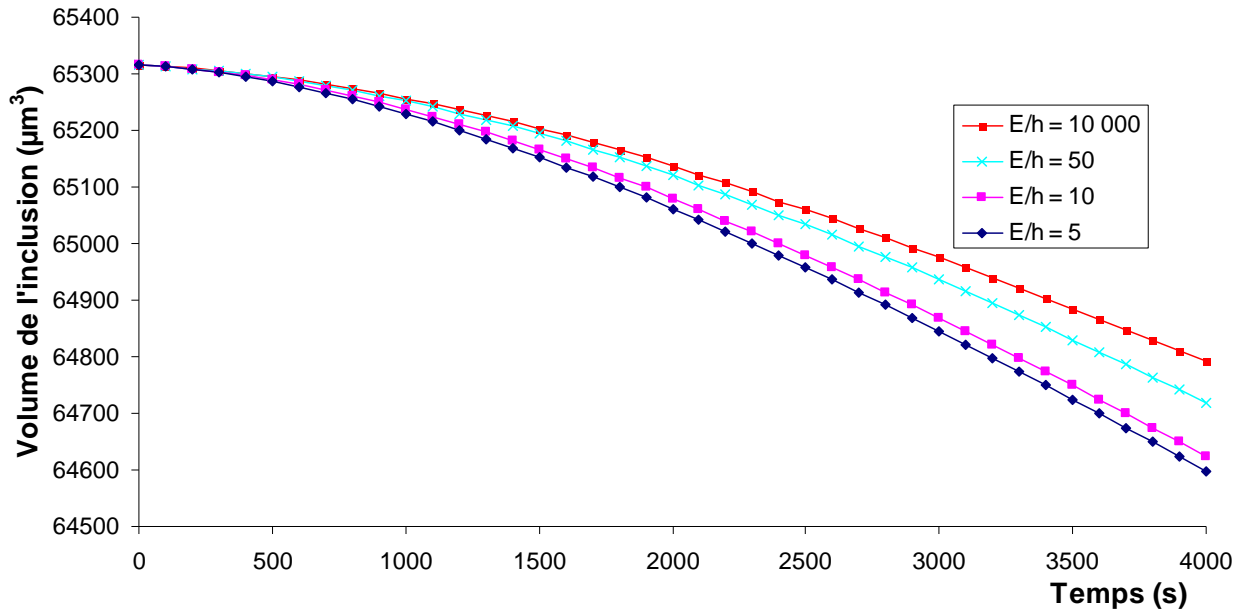


Figure 5.3. Evolution du volume de l'inclusion au cours du temps en fonction de l'épaisseur de lissage de la level set pour un rapport de viscosité $\eta_M / \eta_I = 10^6$.

3. Calculs sur une image 2D

3.1. Représentation par éléments finis

Pour simuler le comportement de l'éprouvette expérimentale, nous devons rajouter une troisième phase dans nos calculs. Plutôt que de considérer un cas test comme le cas de l'inclusion, nous préférons commenter les résultats obtenus directement sur une coupe bidimensionnelle (2D) des images expérimentales. En raison du temps de calcul relativement court qu'elle nécessite, cette approche 2D est particulièrement intéressante pour étudier les paramètres de la simulation. En effet, pour une soixantaine d'incrément, une taille de maille de 10 μm et un comportement viscoplastique du solide, un calcul 2D prend moins de deux heures sur un processeur et un calcul 3D environ une semaine sur 32 processeurs. La coupe 2D considérée est montrée sur la figure 5.4. Le maillage utilisé a une taille de maille homogène et isotrope de 10 μm tandis que le domaine de calcul est un rectangle de 2,24 mm de longueur et 1,43 mm de hauteur.

Pour obtenir cette représentation, nous utilisons les outils décrits au deuxième chapitre. Puisque trois phases sont présentes, nous décomposons l'image segmentée en deux images binaires : la première représentant le liquide et le mélange solide + air et la deuxième représentant l'air et le mélange solide + liquide. Un maillage surfacique est calculé pour chacune de ces deux images, ce qui nous permet d'obtenir sur le maillage volumique une level set associée à la phase liquide et une level set associée à la phase gazeuse. Nous pouvons aussi calculer une level set associée à la phase solide comme l'opposé du minimum des level sets liquide et gazeuse : $\phi_s = -\min(\phi_l, \phi_g)$ avec ϕ_s , ϕ_l et ϕ_g respectivement level sets solide, liquide et gazeuse. A partir de ces différentes level sets, il est facile d'obtenir la représentation par phase montrée sur la figure 5.4. Pour cela, nous définissons une fonction de phase Φ , dont nous calculons la valeur grâce à la formule ci-dessous :

$$\Phi = \begin{cases} 0 & \text{si } \phi_g < 0 \\ 1 & \text{si } \phi_l < 0 \\ 2 & \text{si } \phi_s < 0 \end{cases}$$

$\Phi = 0$, $\Phi = 1$ et $\Phi = 2$ correspondent respectivement aux domaines gazeux, liquide et solide.

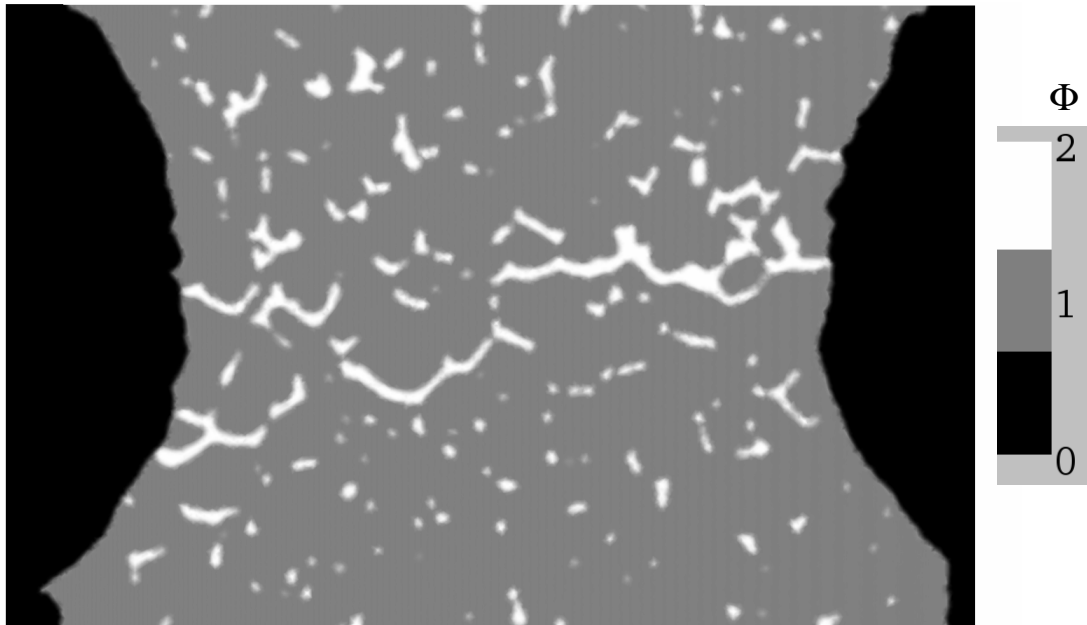


Figure 5.4. Coupe bidimensionnelle utilisée pour les simulations. Gaz, solide et liquide sont respectivement représentés en noir, gris et blanc.

3.2. Comportement newtonien

Dans un premier temps et dans un souci de simplification, nous considérons les trois phases comme newtoniennes. Pour le solide, nous utilisons une viscosité de 10 GPa.s, qui est du même ordre de grandeur que la viscosité apparente pouvant être obtenue avec la loi puissance évoquée dans le quatrième chapitre à la vitesse de déformation considérée. Pour convaincre le lecteur nous rappelons la formule pour calculer la viscosité apparente : $\eta_{app} = K\dot{\epsilon}^{m-1} / 3$. Avec une consistance K de 30 MPa, une sensibilité à la vitesse de déformation m de 0,2 et une vitesse de déformation généralisée $\dot{\epsilon}$ de $2 \cdot 10^{-4}$ (valeur de vitesse de déformation donnée par Terzi et al. [Ter 09]), nous obtenons ainsi une viscosité apparente η_{app} de 9,1 GPa.s.

Les viscosités réelles à considérer pour le liquide et l'air sont de l'ordre de 1 mPa.s et de 0,01 mPa.s, ce qui implique un rapport entre les viscosités maximale et minimale de 10^{15} et une convergence très difficile à obtenir avec ces valeurs. Cependant nous pensons que, comme pour le test de l'inclusion, travailler avec les viscosités réelles ne présente aucun intérêt dès lors que nous prenons un rapport suffisamment important entre la viscosité du solide et celles de l'air et du liquide. Ne possédant pas de solution analytique dans ce cas, nous faisons varier les viscosités et étudions les résultats obtenus.

Pour déterminer la viscosité de l'éprouvette sur chaque élément, nous devons donc opérer deux mélanges. Tout d'abord, nous faisons un mélange en utilisant la level set de l'air avec la

viscosité de l'air et du liquide : $\eta_{g/1} = \tau_{g/1}\eta_l + (1 - \tau_{g/1})\eta_g$ avec $\eta_{g/1}$ le résultat du premier mélange, η_l et η_g les viscosités de du liquide et de l'air et $\tau_{g/1}$ le taux de remplissage

associé à la level set de l'air : $\tau_{g/1} = \left(\sum_{i=1}^{d+1} \phi_g + \sum_{i=1}^{d+1} |\phi_g| \right) / \left(2 \sum_{i=1}^{d+1} |\phi_g| \right)$. Pour le deuxième

mélange, la level set du solide est utilisée avec la viscosité du solide et les résultats du premier mélange : $\eta = \tau_{s/(g/1)}\eta_{g/1} + (1 - \tau_{s/(g/1)})\eta_s$ où η est la viscosité finale mélangée de l'éprouvette, η_s la viscosité du solide et $\tau_{s/(g/1)}$ le taux de remplissage associée à la level set

solide : $\tau_{s/(g/1)} = \left(\sum_{i=1}^{d+1} \phi_s + \sum_{i=1}^{d+1} |\phi_s| \right) / \left(2 \sum_{i=1}^{d+1} |\phi_s| \right)$.

Notre résolution des équations de Stokes prend ainsi en compte les propriétés des trois phases. Le champ de vitesse obtenu nous permet alors de recalculer les level sets du liquide et du gaz, en utilisant pour chacune des deux level sets le solveur LevellerT. La level set solide, quant à elle, est recalculée à chaque incrément à partir des level sets liquide et gazeuse : $\phi_s = -\min(\phi_l, \phi_g)$.

Pour une première série de 14 simulations, nous considérons des viscosités comprises entre 10 GPa.s et 1 mPa.s pour le liquide et une viscosité de l'air 100 fois plus petite que celle du liquide. Etant donné que les viscosités de l'air et du liquide sont très faibles devant celle du solide, nous lançons également une série de 14 simulations pour lesquelles nous faisons varier la viscosité du liquide et prenons une viscosité du gaz égale à celle du liquide. De cette façon, nous n'avons besoin que d'effectuer un seul mélange pour déterminer la viscosité de l'éprouvette : en utilisant la level set solide en combinaison avec les viscosités du liquide (ou de l'air) et du solide. En revanche, même si nous pourrions nous contenter de ne convecter que la level set associée au solide, nous convectons toujours la level set liquide et la level set gazeuse, afin de pouvoir continuer de suivre indépendamment les évolutions des phases liquide et gazeuse au sein de l'éprouvette.

Le pas de temps choisi est de 40 s et nous arrêtons les calculs après 2400 s, soit après 60 incréments de calcul. Comme conditions aux limites, nous imposons une vitesse verticale de $0,1 \mu\text{m.s}^{-1}$ sur le plan supérieur et une vitesse verticale nulle sur le plan inférieur, représentatives de l'essai de traction à modéliser. En effet, lorsque nous regardons l'évolution de la coupe 2D expérimentale (figure 5.5), le plan inférieur semble fixe et l'épaisseur de l'éprouvette ne varie pas au cours du temps. En revanche, la figure 5.5 nous montre clairement une diminution d'épaisseur sur le plan supérieur. Nous avons choisi d'appliquer en tout point de ce plan la vitesse donnée par Terzi et al. pour leur expérience de traction : $0,1 \mu\text{m.s}^{-1}$ [Ter 09]. Il serait intéressant de considérer des conditions aux limites plus complexes sur le plan supérieur, calculées par corrélation d'image et correspondant plus précisément aux résultats expérimentaux. Nous n'avons malheureusement pas reçu les données du laboratoire Navier pour cela.

Afin de résoudre un problème bien posé, nous imposons également une vitesse horizontale nulle sur le nœud situé au centre de l'éprouvette sur le plan inférieur.

Pour nous affranchir de la précision inhérente à la résolution des systèmes linéaires par une méthode itérative, nous résolvons les systèmes linéaires provenant des équations de Stokes par la méthode directe, ce qui est rendu possible par le faible nombre de nœuds dans le maillage considéré : environ 40 000. Ce maillage est illustré sur la figure 5.6.

Pour déterminer les bonnes viscosités à utiliser, c'est-à-dire les viscosités maximales pour lesquelles les résultats ne semblent plus varier lorsqu'elles sont diminuées, nous étudions la forme des éprouvettes déformées ainsi que la répartition des champs obtenus par le calcul mécanique à la fin de la déformation.

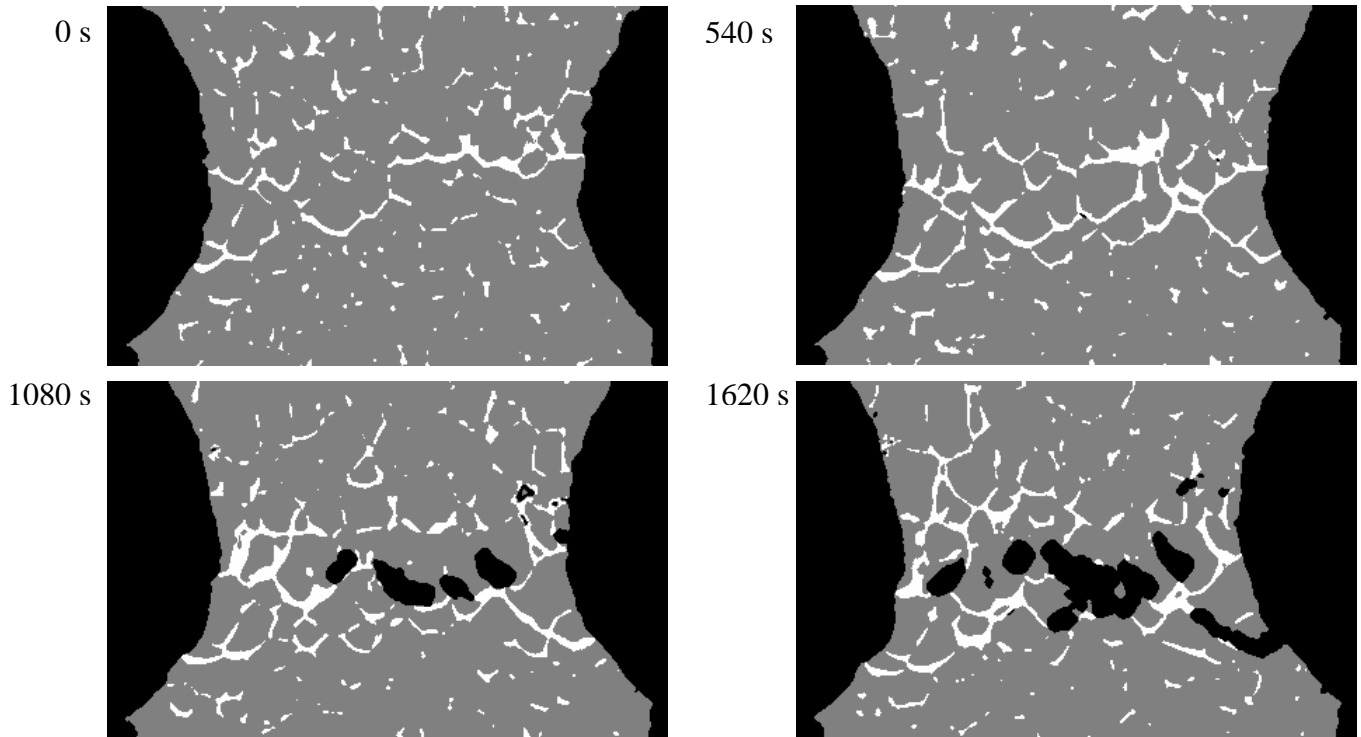


Figure 5.5. Evolution de la coupe 2D expérimentale au cours du temps.

La figure 5.7 illustre la forme de l'éprouvette à la fin de la simulation pour trois différentes viscosités liquide et gazeuse, ainsi que la répartition des contraintes équivalentes de von Mises au sein de l'éprouvette. En 2D, $\bar{\sigma} = 3\sqrt{(\sigma_{xx} - \sigma_{yy})^2/4 + \sigma_{xy}^2}$, avec $\bar{\sigma}$ la contrainte équivalente et σ_{xx} , σ_{xy} et σ_{yy} les composantes du tenseur des contraintes. Nous remarquons que ces trois figures sont très proches les unes des autres. Elles montrent chacune une propagation de l'air au sein de l'éprouvette dans les zones où du liquide était initialement présent. Cependant, pour la plus importante des trois viscosités, nous observons un endommagement de l'éprouvette moins prononcé, matérialisé par une flèche sur la figure 5.7. En prenant une viscosité de l'air et du liquide de 10 kPa.s, nous obtenons un endommagement beaucoup plus proche de celui observé pour la solution que nous qualifions de référence, pour laquelle les viscosités liquide et gazeuse sont respectivement de 100 mPa.s et 1 mPa.s. Ces viscosités proches des viscosités réelles à considérer sont en effet les plus faibles pour lesquelles nous avons réussi à mener le calcul à son terme.

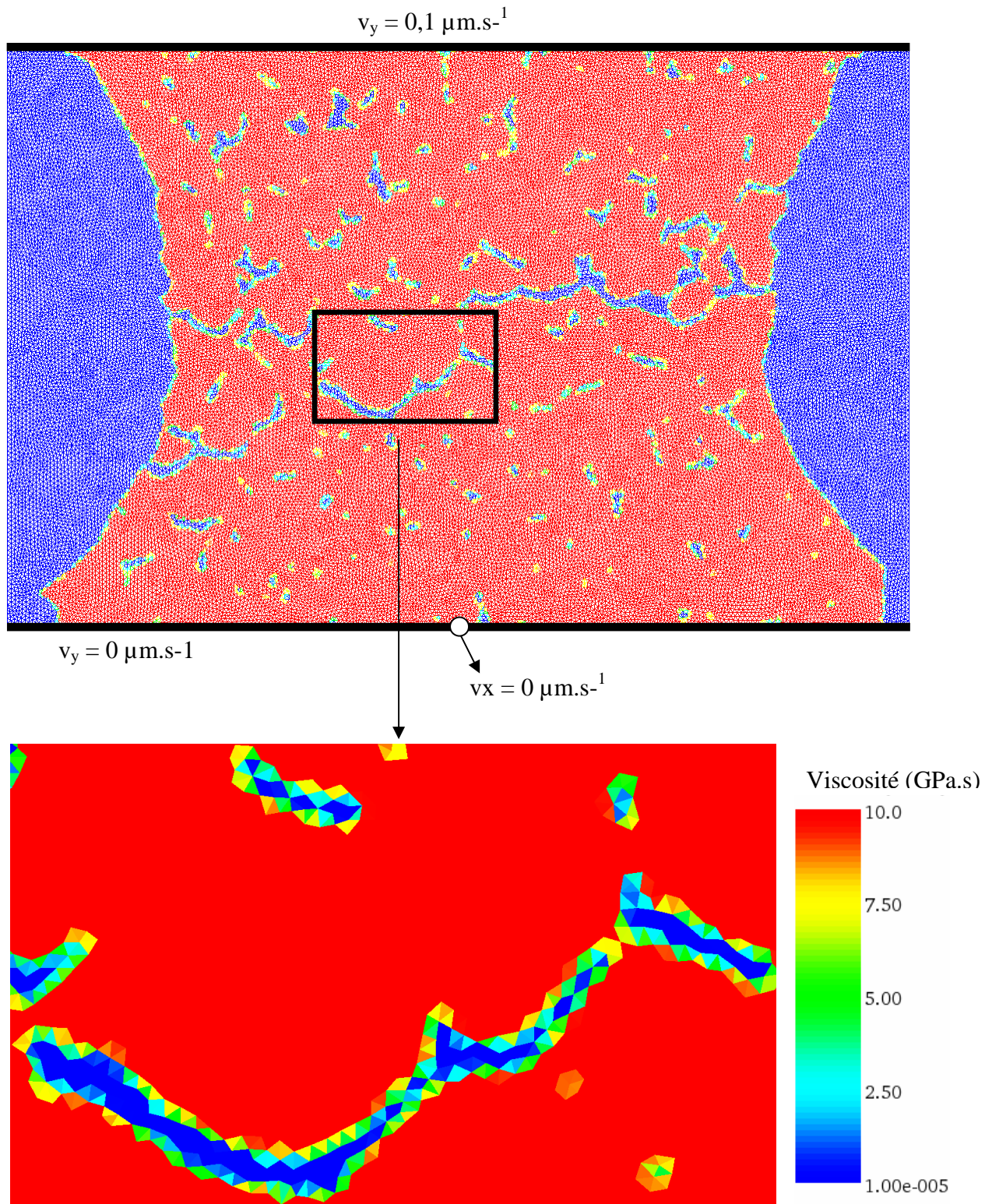
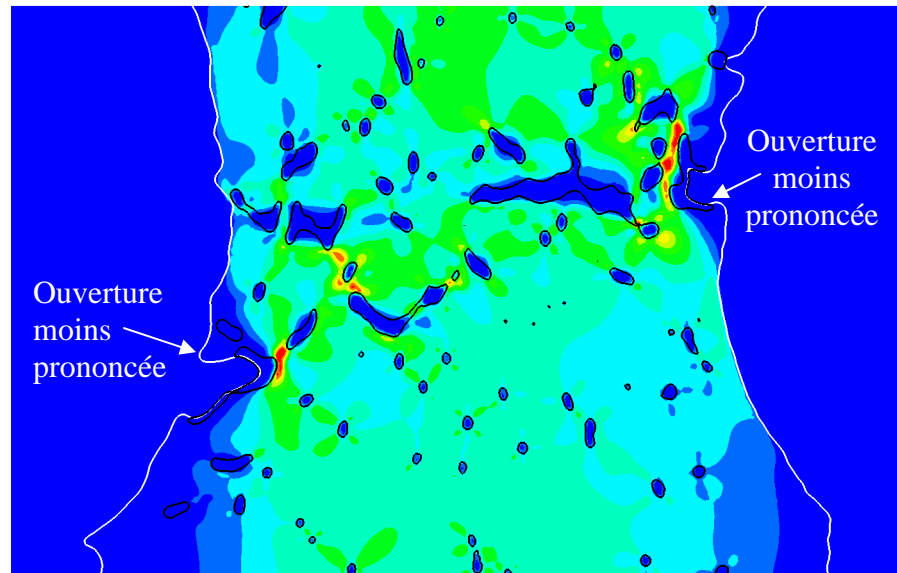


Figure 5.6. En haut : maillage utilisé pour les simulations sur la coupe 2D. En bas : agrandissement sur des veines de liquide.

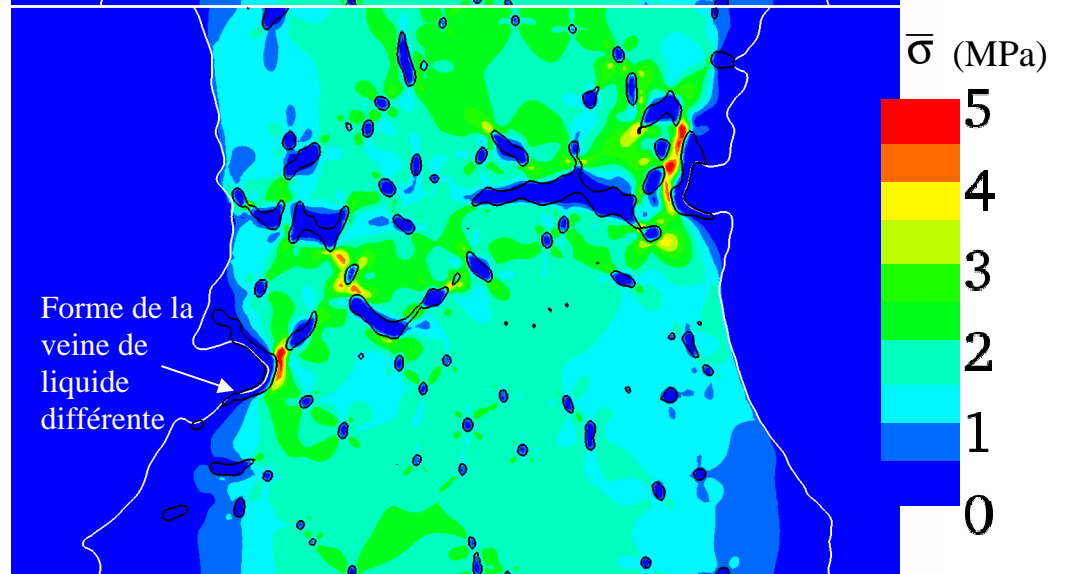
$$\eta^l = 1 \text{ MPa}\cdot\text{s}$$

$$\eta^g = 1 \text{ MPa}\cdot\text{s}$$



$$\eta^l = 10 \text{ kPa}\cdot\text{s}$$

$$\eta^g = 10 \text{ kPa}\cdot\text{s}$$



$$\eta^l = 100 \text{ mPa}\cdot\text{s}$$

$$\eta^g = 1 \text{ mPa}\cdot\text{s}$$

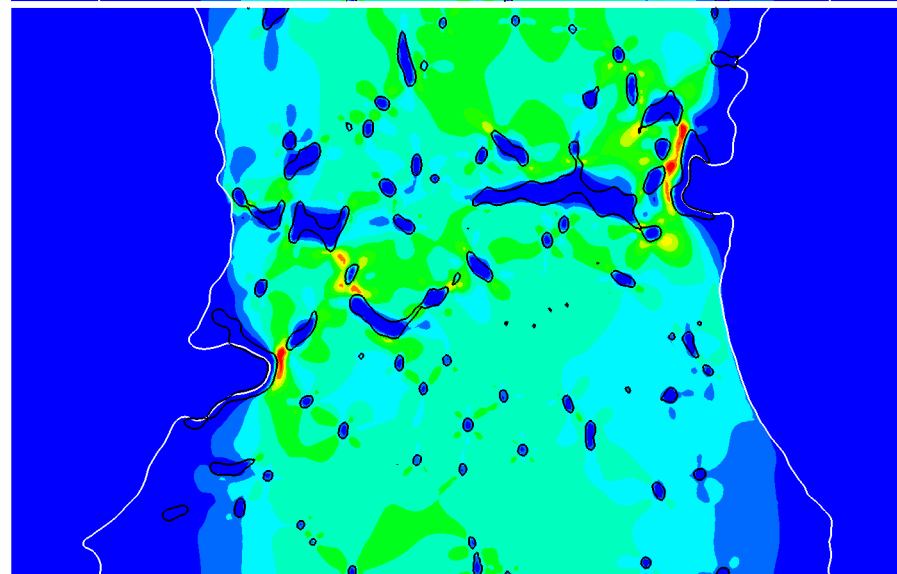


Figure 5.7. Eprouvette après 2400 s. Les contours blanc et noir représentent respectivement les isovaleurs zéro des level sets gazeuse et liquide. La couleur représente la contrainte équivalente.

Nous pouvons donc en conclure que malgré une très légère différence (la forme des veines de liquide indiquée sur la figure 5.7), considérer un rapport entre la viscosité du solide et les viscosités du gaz et du liquide de 10^6 semble tout à fait justifié. Par conséquent après avoir démontré précédemment que le solveur mécanique utilisé ne nous permet pas de réaliser des calculs avec des rapports de viscosité trop importants, nous démontrons ici que considérer des rapports trop importants est inutile. En effet, nous ne présentons ici que la similarité des résultats en contrainte équivalente, mais il en est de même pour tous les autres champs mécaniques, comme la vitesse de déformation généralisée. Ce résultat confirme donc les constatations effectuées dans le cas de l'inclusion, présentées dans la partie précédente.

Nous étudions également les évolutions de la contrainte équivalente moyenne dans la phase solide et de la surface de liquide présente dans l'éprouvette au cours du temps en fonction du rapport de viscosité. La surface de liquide est calculée par la méthode décrite dans le point 5.3.1 du deuxième chapitre. Pour la contrainte équivalente moyenne dans la phase solide, nous calculons d'abord la contrainte équivalente sur chaque élément du domaine. Avec l'outil décrit en 5.3.2 du deuxième chapitre et la valeur de la level set solide, nous calculons la surface de solide S_T présente sur chaque élément, ce qui permet, par une simple sommation

d'obtenir S_s , la surface totale de solide : $S_s = \sum_{T=1}^{N_T} S_T$, N_T étant le nombre d'éléments du

maillage. $\bar{\sigma}_s$, la contrainte équivalente moyenne dans le solide s'écrit alors de la manière

suivante : $\bar{\sigma}_s = \frac{\sum_{T=1}^{N_T} S_T \bar{\sigma}_T}{S_s}$ avec $\bar{\sigma}_T$ la contrainte équivalente sur l'élément T.

Les évolutions de la contrainte équivalente moyenne dans le solide et de la surface de liquide sont respectivement montrées sur les figures 5.8 et 5.9 avec seulement 4 différents rapports, afin d'accroître la lisibilité des résultats. Le rapport 10^{11} correspond à une viscosité de 100 mPa.s pour le liquide et de 1 mPa.s pour le gaz. Nous constatons que plus le rapport est important, plus nous nous rapprochons des résultats associés au rapport 10^{11} . Ainsi, comme pour la figure 5.7, nous réalisons qu'à partir d'un rapport de viscosité de 10^6 , les résultats ne varient plus que très légèrement. En outre, comme le montre la figure 5.7 (en comparant la déformée du milieu à celle du bas) ne considérer qu'une seule valeur de viscosité pour l'air et le liquide est aussi une bonne approximation qui permet de simplifier les calculs. Nous considérons donc par la suite que l'air et le liquide ont tous deux une viscosité de 10 kPa.s.

Dans cette application, la surface de liquide n'est pas conservée en raison des conditions aux limites du calcul. En effet, compte tenu d'une vitesse suivant l'axe de traction nulle sur le plan inférieur et positive sur le plan supérieur ($0,1 \mu\text{m.s}^{-1}$) et de la représentation eulérienne utilisée, le liquide s'échappe par le haut de l'éprouvette au cours du calcul.

Pour les calculs en trois dimensions, le nombre de nœuds du maillage est très important et la résolution des systèmes linéaires associées aux équations de Stokes ne peut plus être directe. Par conséquent nous étudions à présent l'influence de la précision de la résolution itérative sur les solutions obtenues pour notre image 2D.

La figure 5.10 montre l'évolution de la surface de liquide au sein de l'éprouvette pour deux précisions différentes et pour la résolution directe. Les résultats obtenus pour le solveur direct sont quasiment superposés avec ceux obtenus pour la précision de paramètre $P = 10^{-3}$ et c'est donc cette dernière que nous utilisons pour les calculs ultérieurs. En outre, la contrainte équivalente moyenne dans le solide et la configuration des éprouvettes déformées est également très proche. En revanche pour une précision de paramètre $P = 10^{-2}$, qui n'est pas suffisante, nous observons davantage de différences.

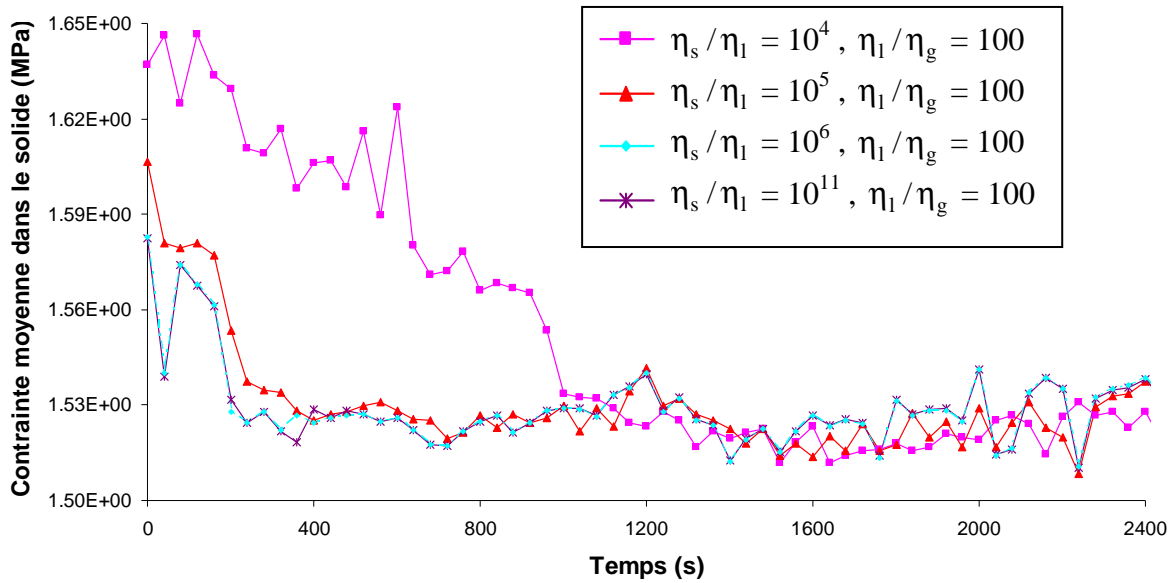


Figure 5.8. Evolution de la contrainte équivalente moyenne dans le solide pour différents rapports entre les viscosités des trois phases, avec une résolution directe.

Nous remarquons que pour le test de l'inclusion, avec un rapport de viscosité $\eta_s/\eta_l = 10^6$ et une précision $P = 10^{-3}$, nous avons obtenu un résultat assez différent de celui obtenu pour une précision $P = 10^{-6}$ (voir figure 5.1). Il semble donc que la précision à employer pour la résolution des systèmes linéaires par une méthode itérative dépend du rapport de viscosité mais également de la configuration des phases au sein de la matrice.

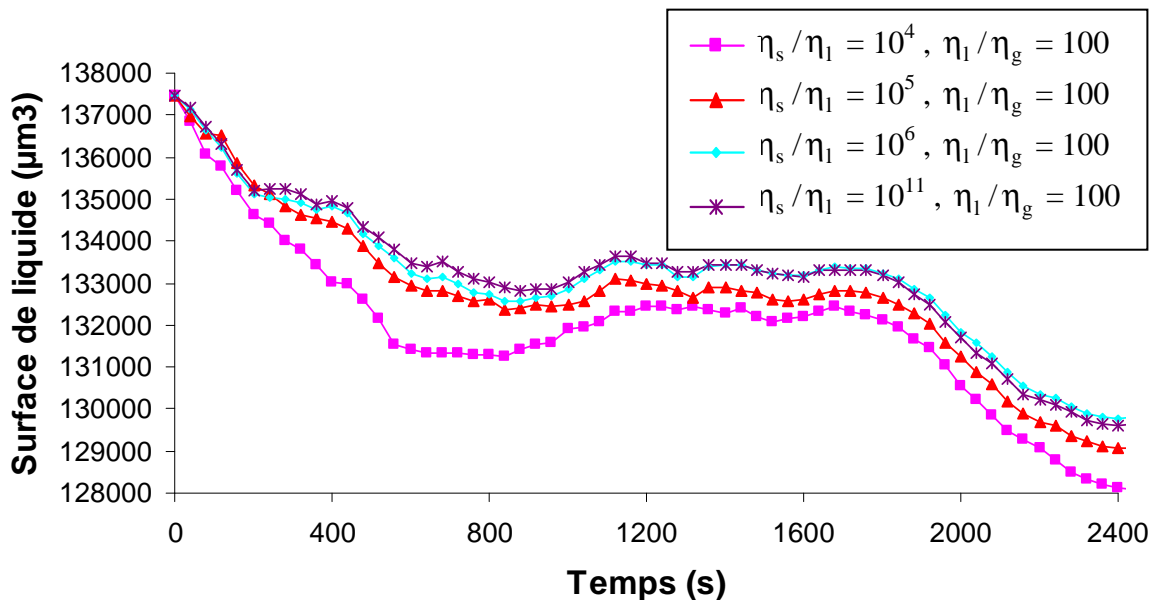


Figure 5.9. Evolution de la surface de liquide pour différents rapports entre la viscosité du solide et du liquide. La viscosité du gaz est 100 fois plus faible que celle du liquide.

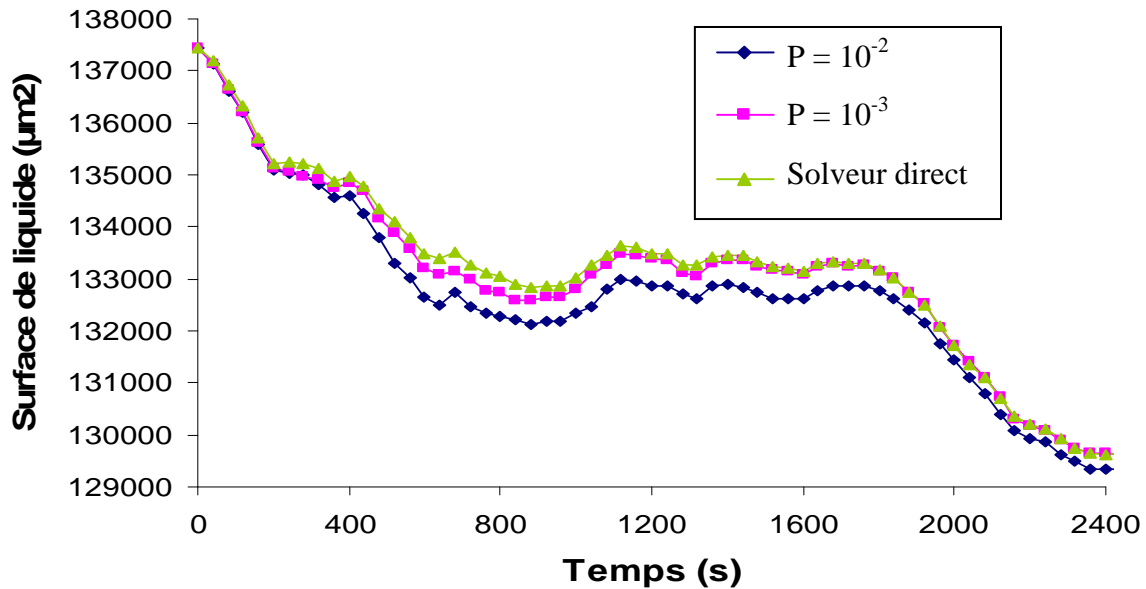
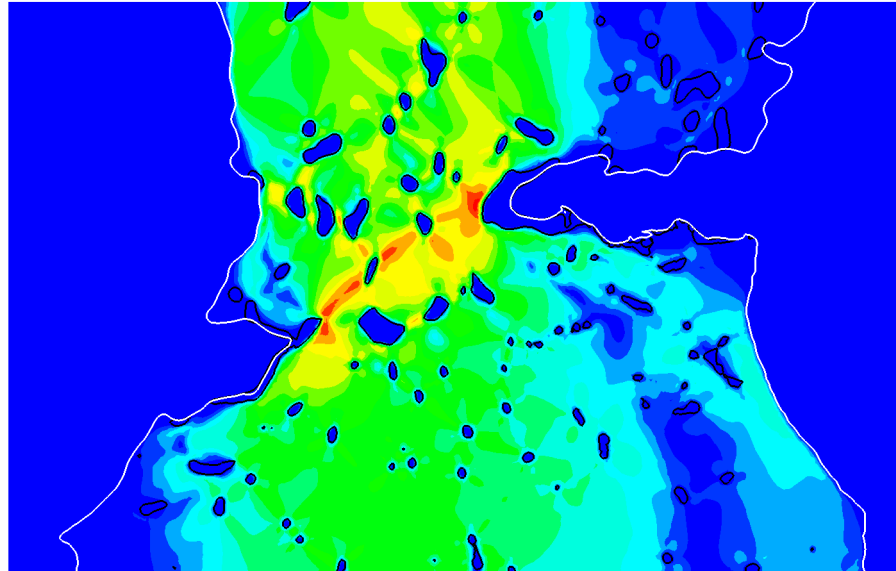


Figure 5.10. Evolution de la surface de liquide pour différentes précisions de résolution des systèmes linéaires associés aux équations de Stokes.

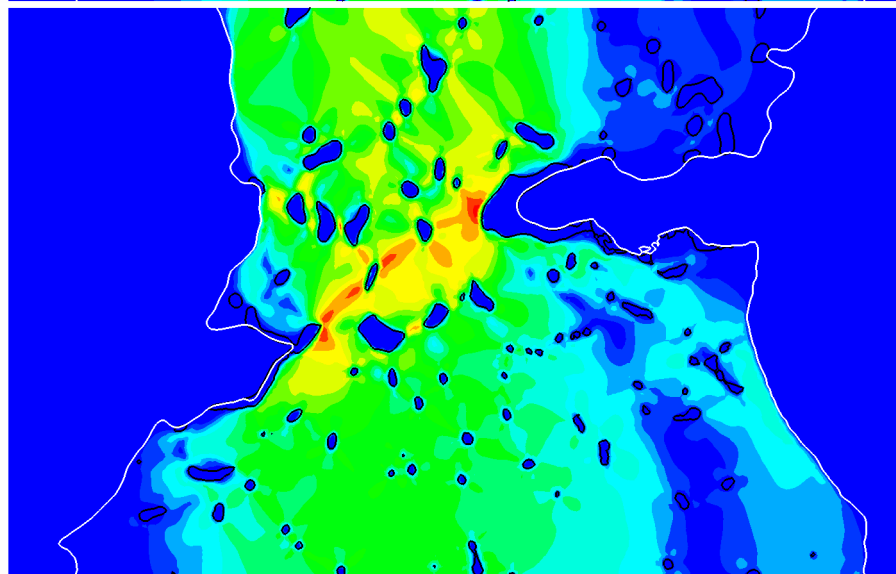
3.3. Comportement viscoplastique

Après avoir déterminé le rapport de viscosité et la précision à utiliser pour résoudre les systèmes linéaires pour une phase solide ayant un comportement newtonien, nous travaillons à présent sur la prise en compte de la loi puissance. Comme mentionné dans le troisième chapitre, nous résolvons le comportement non linéaire par une méthode de point fixe, pour laquelle nous effectuons plusieurs itérations linéaires, en réactualisant à chaque fois la viscosité apparente de l'éprouvette. Pour optimiser la résolution, il nous faut donc déterminer dans un premier temps le nombre d'itérations de point fixe nécessaire. Afin de nous affranchir de la précision de la résolution du système linéaire, nous utilisons pour cela le solveur direct. Nous utilisons une consistance $K_s = 35,2$ MPa et une sensibilité à la vitesse de déformation $m_s = 0,2$, déduites de la campagne expérimentale menée par nos partenaires du SIMaP. Afin de limiter la viscosité apparente maximale dans la phase solide, nous régularisons la vitesse de déformation à une valeur de 10^{-6} s^{-1} : $\dot{\epsilon} \geq 10^{-6} \text{ s}^{-1}$. Les déformées calculées sont montrées après 2400 s sur la figure 5.11 en utilisant respectivement 5, 20 et 50 itérations de point fixe pour chaque résolution non linéaire des équations de Stokes. Nous pouvons voir que l'éprouvette est beaucoup plus endommagée que dans le cas newtonien. Cela s'explique aisément. Dans le cas newtonien, il y a une relation linéaire entre la contrainte équivalente et la vitesse de déformation généralisée. La viscosité est la même partout dans le solide, que ce soit dans les bords inférieurs gauche et droit de l'éprouvette où la contrainte est très faible ou dans les ponts solides situés entre les veines liquides où la contrainte est très élevée. Au contraire, en utilisant la loi puissance, les zones solides où la vitesse de déformation est la plus forte sont aussi celles où la viscosité (du solide) est la plus faible et par conséquent les plus propices à l'endommagement.

5 itérations



20 itérations



50 itérations

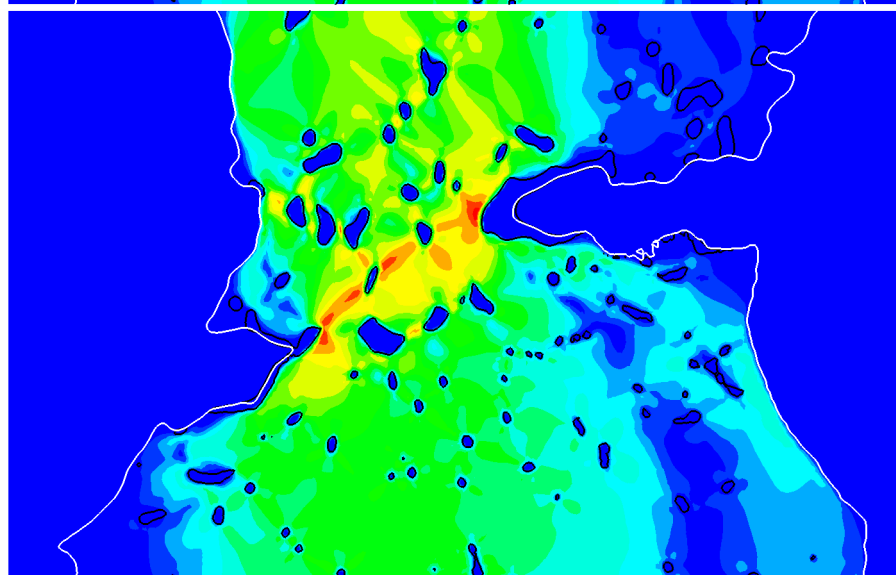


Figure 5.11. Epreuve après 2400 s. Les contours blanc et noir représentent respectivement les isovaleurs zéro des level sets gazeuse et liquide. La couleur représente la contrainte équivalente.

Nous remarquons également que les déformées montrent assez peu de différences pour les trois nombres d'itérations. Par conséquent, nous n'utilisons que cinq itérations de point fixe pour la suite des calculs. Nous tenons malgré tout à souligner qu'en utiliser plus dans ce cas implique seulement un temps de calcul plus long et n'est donc pas rédhibitoire pour le bon déroulement des simulations. La figure 5.12 nous montre l'évolution de la norme du résidu au début de chaque itération de point fixe, durant la première résolution mécanique et lorsque 50 itérations sont considérées. Afin d'éviter la confusion dans l'esprit du lecteur, nous rappelons à présent succinctement les différentes étapes pour résoudre les équations de conservation de la quantité de mouvement et de la masse, dans le cas d'une phase solide ayant un comportement viscoplastique.

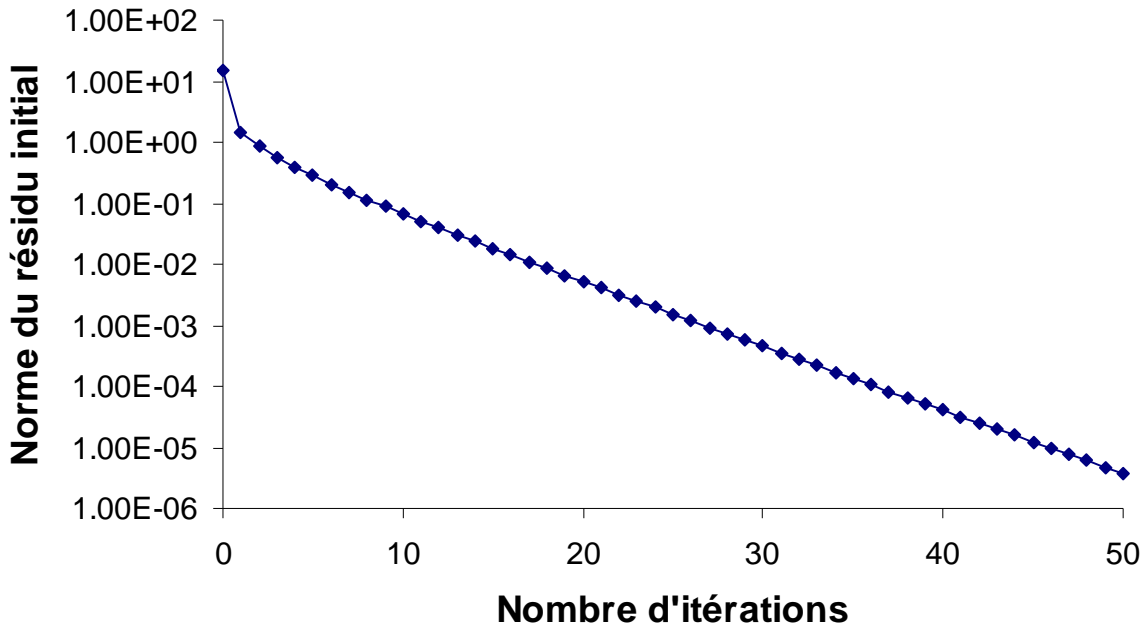


Figure 5.12. Evolution de la norme du résidu initial lors de la première résolution mécanique.

Nous initialisons tout d'abord la vitesse de déformation généralisée en considérant une valeur constante dans l'éprouvette, donnée par la formule suivante : $\dot{\epsilon}_0 = (V_H - V_B)/H$ avec V_H , V_B et H les vitesses suivant l'axe Oy appliquées sur les plans supérieur et inférieur de l'éprouvette, ainsi que la hauteur de l'éprouvette. La vitesse et la pression initiale sont nulles. Nous effectuons ensuite un mélange pour obtenir la consistance et la sensibilité à la vitesse de déformation sur chaque élément du domaine de calcul : $m = \tau_{s/(g/l)}m_1 + (1 - \tau_{s/(g/l)})m_s$ et $K = \tau_{s/(g/l)}K_1 + (1 - \tau_{s/(g/l)})K_s$ avec $\tau_{s/(g/l)}$ le taux de remplissage de l'élément calculé grâce à la level set solide ϕ_s , $m_1 = m_g = 1$ la sensibilité à la vitesse de déformation dans le liquide et le gaz, $K_1 = K_g = 10$ kPa la consistance dans les phases liquide et gazeuse. La viscosité apparente est alors obtenue : $\eta_{app} = K\dot{\epsilon}^{m-1}/3$. Cette viscosité nous permet d'assembler le système linéaire suivant : $\overline{\overline{A}}\vec{x} = \vec{b}$, pour la résolution duquel nous utilisons le solveur direct. Le vecteur résidu s'écrit $\vec{r} = \overline{\overline{A}}\vec{x} - \vec{b}$ et nous montrons sa norme sur la figure 5.12 (premier point en haut à gauche). Une fois le système résolu, une nouvelle vitesse est obtenue permettant de recalculer la viscosité apparente et le nouveau système linéaire associé à résoudre. La norme du résidu correspondante est le deuxième point de la figure 5.12 (nombre

d'itérations = 1). Cette procédure est répétée jusqu'à atteindre le nombre d'itérations de point fixe considéré : 50 dans le cas de la figure 5.12.

La figure 5.12 nous montre que la norme du résidu initial (calculée avant la résolution directe du système linéaire) décroît continûment en fonction du nombre d'itérations de point fixe. Si cette norme initiale n'a pas de rapport avec la norme finale du résidu dans le cas d'une résolution itérative avec un comportement newtonien de l'éprouvette étudié précédemment, nous faisons malgré tout la même observation la concernant : il est inutile de vouloir que cette norme soit trop faible, car les résultats obtenus ne sont plus améliorés en dessous d'une certaine valeur.

Nous avons pour l'instant calculé la viscosité apparente du solide en prenant pour valeur minimale de la vitesse de déformation équivalente 10^{-6} s^{-1} . Concrètement la vitesse de déformation équivalente atteint parfois des valeurs de l'ordre de 10^{-9} s^{-1} sur certains éléments appartenant à la phase solide. Sans troncature, la viscosité apparente associée à de tels éléments deviendrait très importante et le rapport entre les viscosités maximale et minimale nous empêcherait de mener les calculs à bien. Nous avons décidé de changer la valeur de la troncature afin d'étudier son influence. La figure 5.13 montre le contour de l'air pour trois différentes valeurs. Avec 10^{-4} s^{-1} , la viscosité maximale dans le solide est moins importante dans les zones à faible vitesse de déformation et l'air prend plus facilement la place du solide dans les coins inférieurs gauche et droite.

Par conséquent l'air a aussi plus de difficultés à se propager au sein de l'éprouvette que dans un cas avec une vitesse de troncature plus faible. Cette valeur de 10^{-4} s^{-1} correspond à la vitesse de déformation minimale pour laquelle la loi de comportement du solide a été identifiée et il est donc difficile de savoir si une autre troncature est plus appropriée. Cependant, nous notons que compte tenu de la vitesse appliquée en haut de l'éprouvette et de sa hauteur, la vitesse de déformation généralisée dans le cas d'un échantillon purement solide serait inférieure à 10^{-4} s^{-1} et il nous paraît donc important de considérer une valeur de troncature plus faible. Nous considérons par la suite une valeur de 10^{-6} s^{-1} pour laquelle une diminution n'entraîne plus beaucoup de changements dans les résultats obtenus.

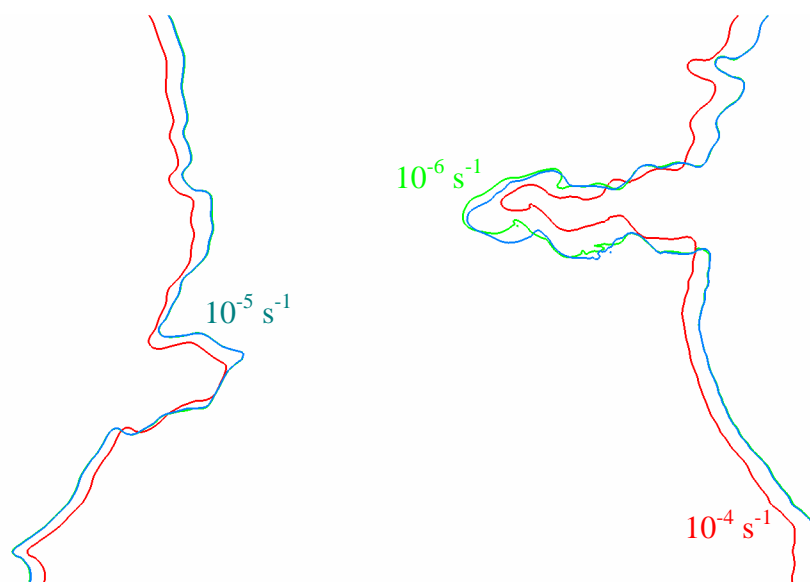


Figure 5.13. Isovaleur zéro de la level set gazeuse pour différentes vitesses de régularisation. A gauche de la figure, les isovaleurs pour 10^{-5} s^{-1} et 10^{-6} s^{-1} sont quasiment confondues, ce qui empêche de voir l'isovaleur pour 10^{-6} s^{-1} .

Les résultats avec la loi de comportement viscoplastique sont pour l'instant obtenus avec le solveur direct. Comme pour le comportement newtonien, nous faisons à présent des tests avec le solveur itératif. Dans un premier temps, nous demandons une précision constante pour chacune des cinq itérations de point fixe. Une précision $P = 10^{-4}$ semble la plus adaptée dans ce cas. Nous estimons cependant qu'il n'est pas nécessairement utile de prendre la même précision pour toutes les itérations de point fixe. En effet, après une seule itération, même une solution obtenue avec une résolution parfaite du système linéaire sera encore très éloignée de la solution non linéaire. Au contraire, il semble indispensable pour la dernière itération de point fixe, pour laquelle nous sommes beaucoup plus proche de la véritable solution, d'exiger une norme du résidu suffisamment basse. Ce constat est illustré sur la figure 5.14, sur laquelle nous représentons les déformées après 2400 s pour trois différentes résolutions itératives.

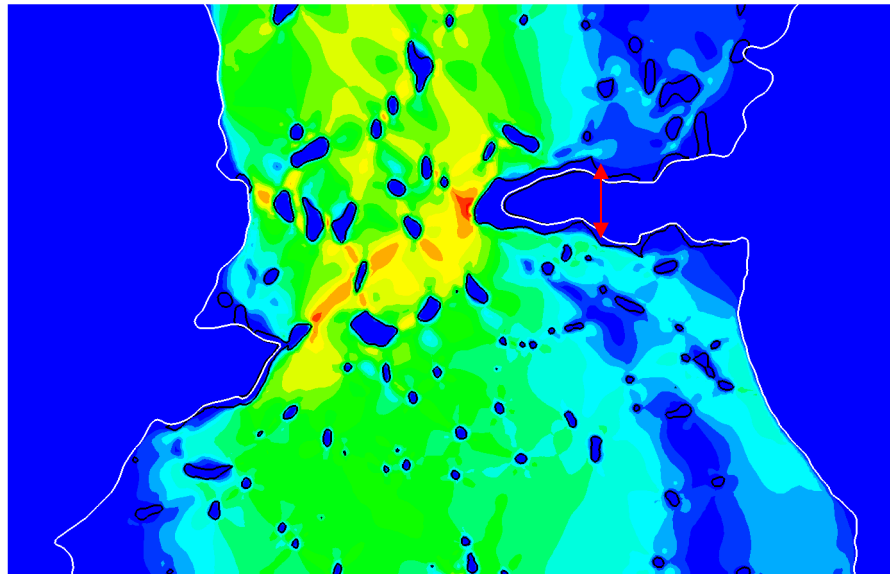
Les deux premières déformées sont obtenues avec une précision constante pour les cinq itérations tandis que nous avons continuellement modifié la précision pour la déformée du bas de la figure 5.14 : les normes du résidu exigées étant respectivement 1 ; 0,1 ; 10^{-2} ; 10^{-3} et 10^{-4} . Les disparités observables sur cette figure en analysant les trois éprouvettes sont assez limitées mais il existe tout de même une différence assez importante matérialisée par les trois flèches rouges, ayant même dimension pour les trois déformées. Dans le cas avec la précision de 10^{-3} , l'ouverture provoquée par la propagation de l'air au sein de l'éprouvette est moins importante. Ainsi, le résultat obtenu en diminuant successivement la norme du résidu final est plus proche de la solution obtenue avec la précision de 10^{-4} (la plus précise des 3) que celle obtenue avec 10^{-3} , et elle nécessite pourtant un temps de calcul plus faible. Exiger une norme du résidu variable en fonction du numéro de l'itération de point fixe permet donc d'obtenir une meilleure solution tout en diminuant le temps de calcul : c'est ce que nous appliquons par la suite.

3.4. Influence du pas de temps et de la taille de maille

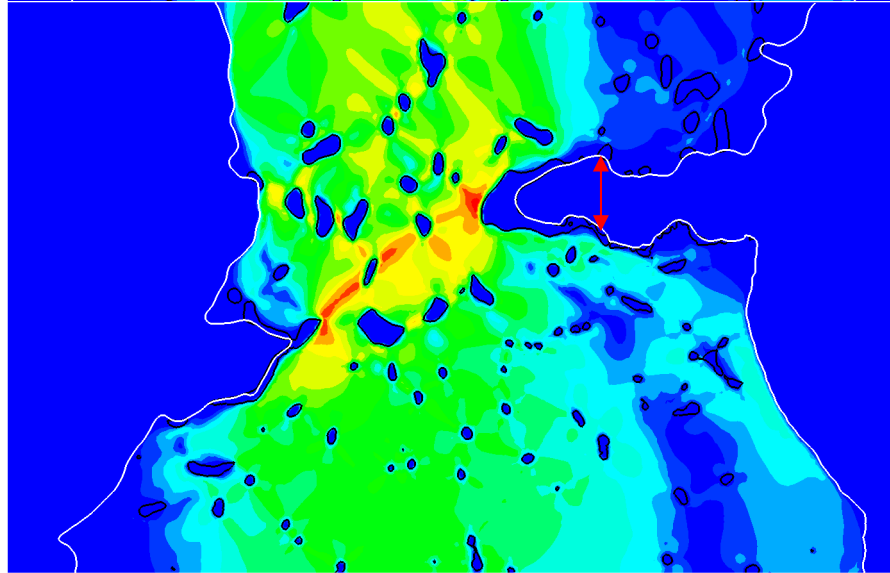
Le pas de temps fait partie des paramètres importants de la simulation. Il ne doit pas être choisi trop grand, afin de respecter la condition CFL, évoquée au chapitre précédent et selon laquelle l'interface ne doit pas franchir plus d'un élément par incrément. Si le pas de temps est choisi trop petit, le temps de calcul est inutilement augmenté. En outre comme vu au chapitre précédent, la conservation peut alors se détériorer par rapport à un calcul avec un pas de temps plus grand. Dans les cas où nous imposons un champ de vitesse fixe pour tester le solveur de convection, nous connaissons la valeur maximale de la vitesse au voisinage de l'interface et il est donc très facile de choisir un pas de temps adapté. Mais à présent, le champ de vitesse étant calculé par notre solveur mécanique, la détermination du pas de temps n'est plus aussi simple.

Il est possible de travailler avec un pas de temps adaptatif, calculé en fonction de la valeur maximale du champ de vitesse au voisinage de l'interface. Mais cela rendrait la simulation plus instable car l'obtention d'une valeur erronée et très importante en un seul nœud du domaine de calcul pourrait réduire drastiquement la valeur du pas de temps, qui ne correspondrait plus alors à la vitesse calculée sur les autres nœuds. De plus, l'utilisation d'un pas de temps fixe permet des comparaisons de la forme des éprouvettes après un même temps de simulation, comme illustré précédemment. Et cela nous donne également l'occasion de pouvoir estimer facilement le temps de calcul maximal des simulations que nous effectuons. Malgré notre choix d'utiliser un pas de temps de fixe, nous tenons à signaler que l'emploi d'un pas de temps adaptatif fait actuellement l'objet de travaux au CEMEF, afin d'optimiser le temps nécessaire aux simulations.

Précision 10^{-3}



Précision 10^{-4}



Précision 1 ;
0,1 ; 10^{-2} ;
 10^{-3} ; 10^{-4}

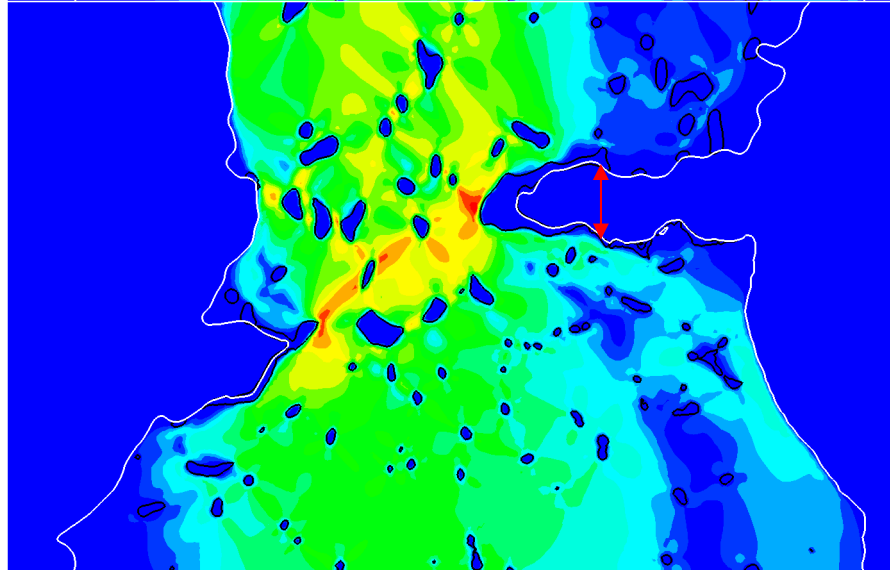


Figure 5.14. Eprouvette après 2400 s pour différentes résolutions itératives.

Pour déterminer le pas de temps fixe à utiliser, nous considérons les conditions aux limites appliquées au bord du domaine. Nous rappelons qu'il s'agit d'une vitesse verticale de $0,1 \mu\text{m}\cdot\text{s}^{-1}$ sur le plan supérieur et d'une vitesse verticale nulle sur le plan inférieur. Avec une éprouvette composée de solide pur, la vitesse verticale appliquée sur le plan supérieur constitue donc la vitesse verticale maximale calculée sur le domaine de calcul. Avec notre échantillon multiphasique, ce résultat n'est plus valable car la vitesse peut localement être plus importante dans des veines de liquide ou dans l'air pénétrant à l'intérieur de l'éprouvette. Cependant, après une observation minutieuse des résultats obtenus, nous pouvons affirmer que bien souvent la norme maximale de la vitesse au sein de l'éprouvette ne dépasse pas $0,5 \mu\text{m}\cdot\text{s}^{-1}$ et le pas de temps à considérer est donc du même ordre de grandeur que ceux considérés pour les calculs sans résolution des équations de Stokes. En utilisant les paramètres déterminés lors des études détaillées plus tôt dans ce chapitre, nous avons donc relancé la simulation avec différents pas de temps. L'évolution de la surface de liquide dans l'éprouvette au cours de ces simulations est montrée sur la figure 5.15. Le pas de temps de 40 s est celui utilisé précédemment.

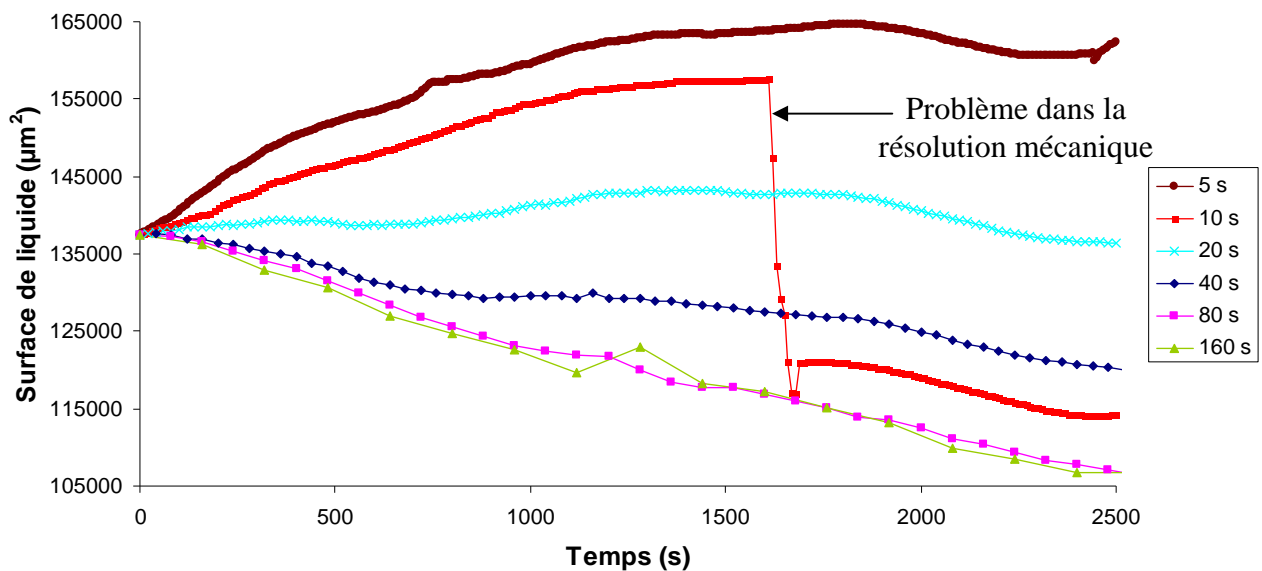


Figure 5.15. Evolution de la surface de liquide dans l'éprouvette au cours de la simulation pour différents pas de temps.

Avec un pas de temps plus important, les résultats obtenus sont comparables. Nous notons toutefois que certaines petites poches et veines de liquide disparaissent au cours de la simulation, d'où la plus faible surface de liquide observée sur la figure 5.15. Pour le pas de temps de 20 s, la déformée finale ressemble également beaucoup à celle obtenue pour le pas de temps de 40 s, avec davantage de liquide à la fin du calcul. Nous remarquons toutefois que durant la première moitié de la simulation, la surface de liquide augmente. Ce résultat n'est pas étonnant, car la non conservativité de notre solveur level set peut aussi bien entraîner des pertes que des gains de surface, d'ailleurs obtenus pour le cas du vortex étudié dans le chapitre précédent. Nous ajoutons cependant qu'avec les conditions aux limites et le suivi eulérien de la matière considérés ici, la quantité de liquide présente initialement dans l'éprouvette ne devrait pouvoir que diminuer, car le liquide s'échappe par le haut de l'éprouvette au cours du calcul et aucune alimentation n'est prise en compte. Un calcul au cours duquel la surface de liquide est en augmentation importante, comme par exemple pour les pas de temps de 5 et 10 s, trahit par conséquent un manque flagrant de conservation de la matière et les résultats associés doivent être considérés avec les plus grandes précautions.

Pour le pas de temps de 10 s, nous constatons également une diminution très rapide de la surface de liquide après 1600 s. En analysant le déroulement du calcul à ce moment précis, nous nous rendons compte que cette diminution coïncide avec une non convergence de la résolution mécanique. La figure 5.16 illustre d'ailleurs la variation de la norme du résidu pour la cinquième et dernière itération de point fixe associée.

Nous pouvons voir sur cette figure que la norme du résidu oscille énormément et ne descend jamais en dessous de la valeur exigée de 10^{-4} . La résolution continue ainsi jusqu'à atteindre le nombre maximal d'itérations que nous fixons à 50 000. Seulement, la norme du résidu à la dernière itération est supérieure à 76 et la solution obtenue est donc très mauvaise. Pourtant, la norme du résidu à la 338^{ème} itération est de $1,15 \cdot 10^{-4}$, valeur très proche de la norme demandée, et il est donc certain que la solution correspondante est satisfaisante.

Pour éviter ce problème, il nous suffirait par conséquent de garder en mémoire la solution associée à la plus faible norme du résidu calculée au cours de la résolution itérative avant de la poursuivre afin d'obtenir une norme encore plus faible. Et en cas d'échec, comme montré sur la figure 5.16, nous récupérerions la solution mise de côté. Cette idée semble a priori extrêmement simple à appliquer, mais ce n'est malheureusement pas le cas dans l'environnement CIMLib/PETSc. Pour pallier ce problème, nous avons donc recours à une solution moins élégante : pour les calculs en trois dimensions, obtenir une norme du résidu aussi faible que celle demandée en deux dimensions pour les 4^{ème} et 5^{ème} itérations de point fixe est très compliqué et nous utilisons donc la notion de relaxation pour laquelle la norme du résidu exigée augmente au fil des itérations (cf 3^{ème} chapitre). Cette différence est justifiée par le nombre de degrés de liberté beaucoup plus important en 3D : il est plus difficile d'obtenir une norme du vecteur résidu inférieure à 10^{-4} car ce vecteur comporte alors beaucoup plus d'éléments. Concrètement la norme du résidu exigée est la suivante : $\|\bar{r}\| < P_i/200$ avec i le numéro de l'itération de résidu conjugué, et P respectivement égal à 1 ; 0,1 ; 10^{-2} ; 10^{-3} et 10^{-4} pour les cinq itérations de point fixe.

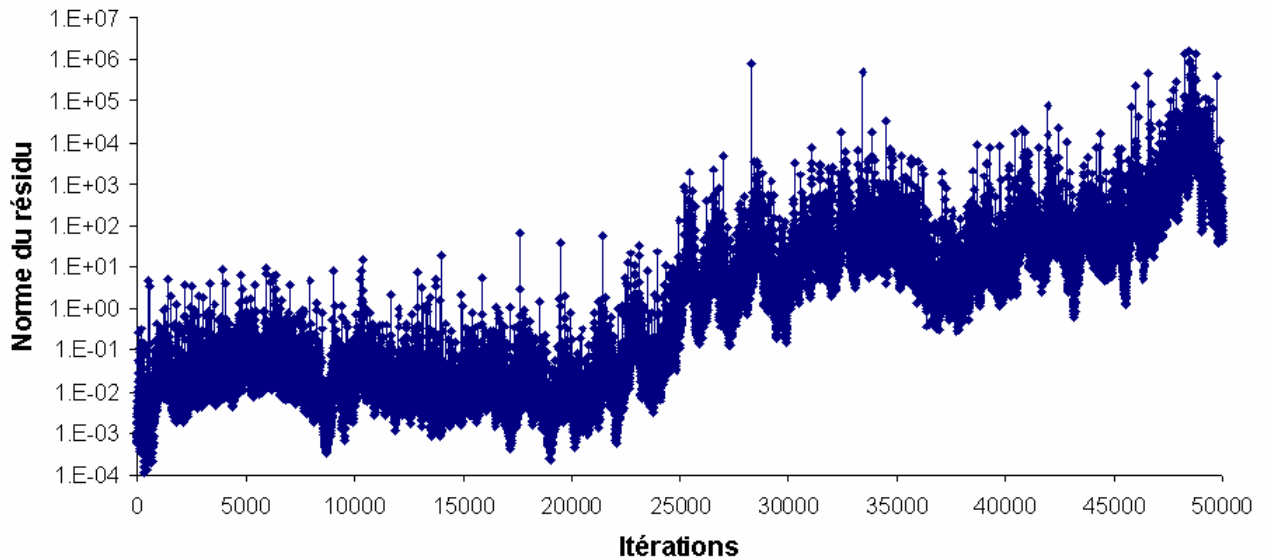


Figure 5.16. Evolution de la norme du résidu lors d'une résolution où la convergence n'est pas obtenue.

Pour résumer brièvement les conclusions liées à cette étude sur le pas de temps à considérer, nous nous contentons de préciser que nous conservons le pas de temps pris dans les études précédentes, c'est-à-dire 40 s pour une taille de maille de 10 μm .

A l'image du pas de temps, la taille de maille est un autre paramètre clef de la résolution. Nous avons pour le moment considéré une taille de 10 μm , plus importante que la taille du pixel de l'image de départ, égale à 5,6 μm . En prenant les mêmes paramètres que précédemment et le même facteur de proportionnalité pour calculer le pas de temps et l'épaisseur de lissage de la level set, nous relançons le calcul pour des tailles de maille de 5, 4 et 3 μm . En comparant les déformées de la figure 5.17 avec celle calculée pour la taille de maille de 10 μm (bas de la figure 5.14), nous constatons deux différences importantes : (i) de petites poches de liquide ayant disparu avec la taille de maille de 10 μm sont encore présentes avec les tailles plus faibles, (ii) l'air s'est propagé de manière différente à l'intérieur de l'éprouvette à droite des images. Nous remarquons également une grande ressemblance des déformées pour les trois tailles de maille, ce qui signifie qu'il est alors inutile de raffiner encore davantage le maillage. Une taille de maille de 5 μm semble par conséquent adaptée.

4. Calculs sur une image 3D

Tous les tests effectués en deux dimensions nous permettent de choisir les paramètres à employer pour les calculs tridimensionnels. En trois dimensions, le principe de résolution est le même, à une différence fondamentale près. Le nombre de degrés de liberté nécessaire pour avoir une représentation suffisamment fine des échantillons est très important, ce qui implique des calculs beaucoup plus longs et difficiles à réaliser. Ainsi, nous préférons procéder par étape en étudiant dans un premier temps le comportement du solveur LevelSet.

Pour cela, nous créons la représentation éléments finis de départ à partir d'une image expérimentale, en ne considérant que la phase liquide et la phase solide (figure 5.20). Nous appliquons alors un champ de vitesse incompressible donné par la solution analytique d'un test de traction uniaxial : $v_x = -kx/2$; $v_y = ky$; $v_z = -kz/2$ avec $k = V_h - V_b/H$. V_h et V_b sont les vitesses imposées sur les plans supérieur et inférieur de la matrice et H la hauteur de la matrice. Ce n'est ni plus ni moins que la reproduction du test de l'inclusion utilisé dans le chapitre précédent et appliqué à une morphologie réelle de veines et de poches liquides.

Comme nous tenons à voir l'amélioration de la conservation avec la diminution de la taille de maille, nous ne considérons qu'une portion de 15 ° du cylindre expérimental afin de limiter la taille des maillages employés. En outre nous rajoutons une couche de solide au-dessus de l'éprouvette pour que le liquide reste contenu dans le domaine de calcul tout au long de la simulation. L'éprouvette fait alors 1,85 mm de hauteur et le champ de vitesse appliqué suivant l'axe de traction est de 0,1 $\mu\text{m}\cdot\text{s}^{-1}$ sur le plan supérieur. Elle est nulle sur le plan inférieur.

La figure 5.18 montre l'évolution du volume de liquide au cours du temps pour différentes tailles de maille. L'épaisseur de lissage de la level set vaut 10 tailles de maille et le pas de temps est également proportionnel à la taille de maille : il est de 50 s pour la taille de 10 μm . Le temps final est de 4000 s. Le volume de liquide est de plus en plus important au début de la simulation à mesure que le maillage est raffiné. C'est aussi le cas pour le volume de liquide à la fin du calcul. En outre la perte de volume est également de moins en moins élevée quand la taille de maille est diminuée, comme l'illustre la figure 5.19. Nous avons en effet une perte de près de la moitié du volume initial pour la taille de maille de 15 μm et elle n'est plus que de 0,72 % pour la taille de 4 μm . La figure 5.20 illustre les contours initial et final du liquide pour cette taille de 4 μm .

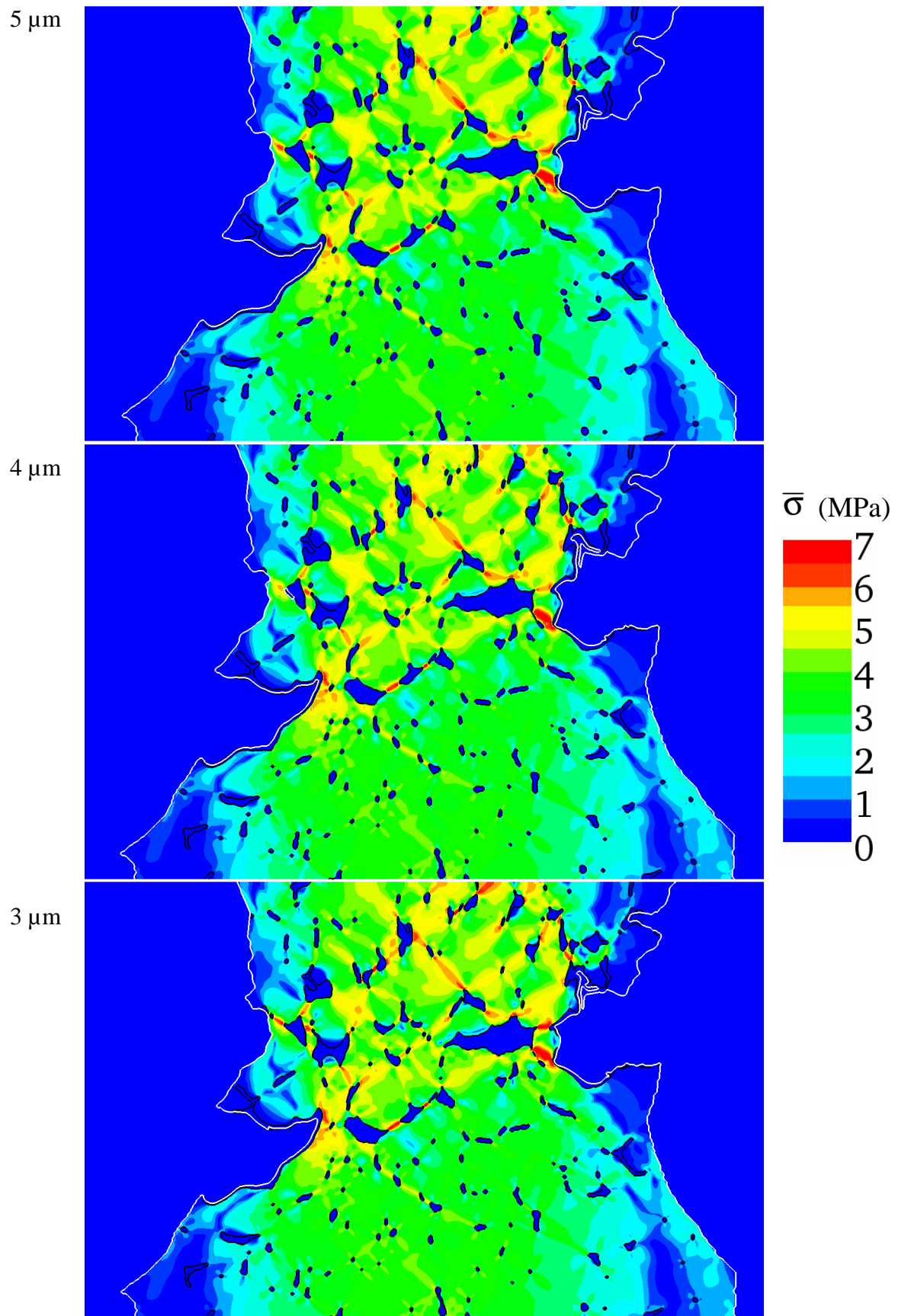


Figure 5.17. Eprouvette après 2400 s pour différentes tailles de maille.

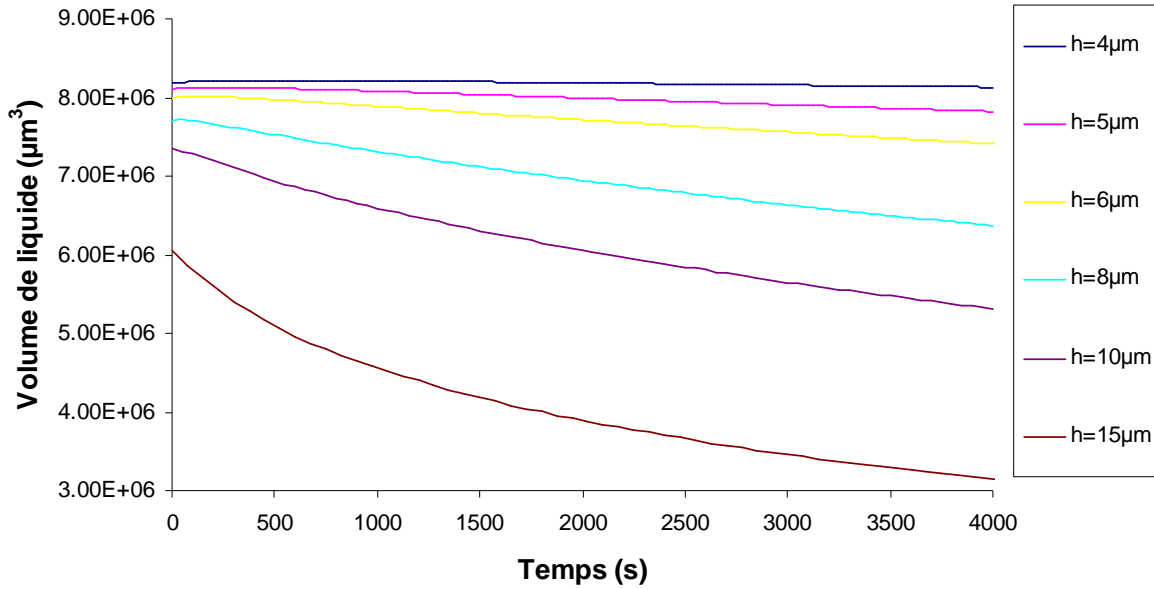


Figure 5.18. Evolution du volume de liquide pour un champ de vitesse incompressible en fonction de la taille de maille.

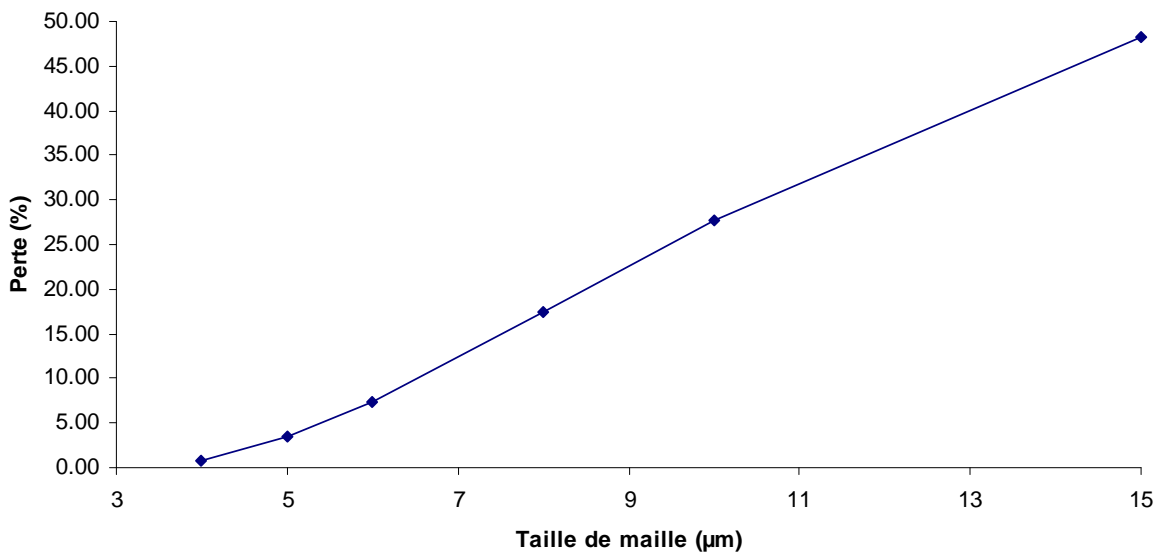


Figure 5.19. Evolution de la conservation en fonction de la taille de maille après 4000 s.

Même si l'obtention d'une aussi bonne conservation pour la taille de maille la plus faible semble encourageante, nous devons préciser que le calcul associé à la figure 5.20 a nécessité environ 5 h avec 50 processeurs. Le maillage correspondant possède presque 9 000 000 de nœuds et il ne permet pourtant de ne représenter qu'une portion de 15 ° de l'éprouvette expérimentale. Nous venons ainsi de montrer la convergence du solveur level set pour déplacer des veines liquides mais nous devons malheureusement nous résoudre à lancer des calculs avec une plus grosse taille de maille si nous voulons représenter une plus grosse partie de l'échantillon. Pour pallier ce problème, nous avons également testé des calculs avec remaillage, mais n'y avons pas donné suite pour les raisons décrites dans les deux paragraphes suivants.

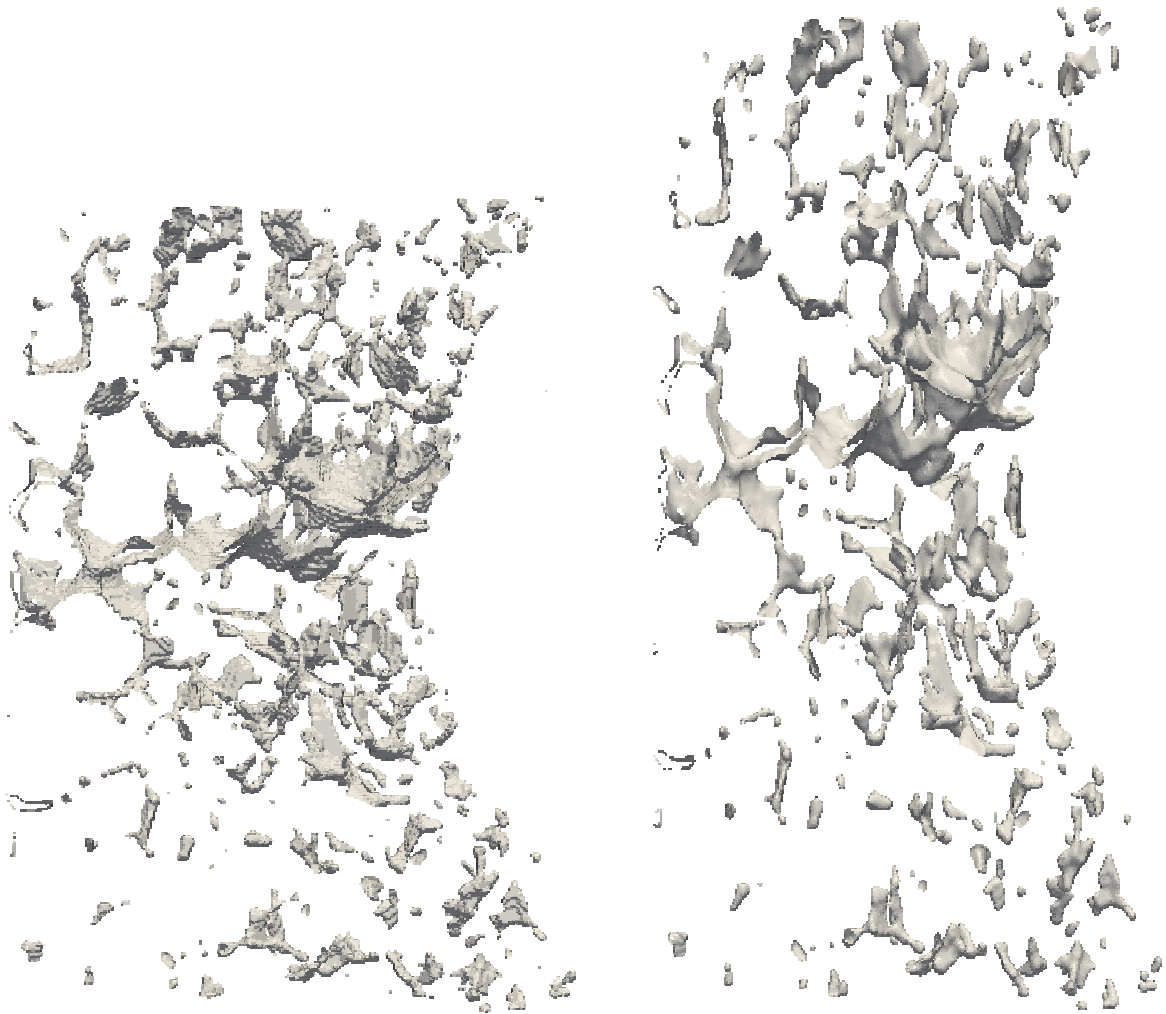


Figure 5.20. Contours initial (à gauche) et final (à droite) du liquide pour la taille de maille de $4\ \mu\text{m}$.

Avec le test de l'inclusion, pour lequel le remaillage adaptatif était tout à fait justifié, l'interface présentait une faible surface par rapport à la taille du domaine de calcul et était extrêmement localisée. Au contraire, dans le cas d'une image réelle, l'aire de l'interface est beaucoup plus importante et le liquide est plus ou moins réparti uniformément dans tout le domaine de calcul. Avoir une taille uniforme sur une certaine épaisseur autour de l'interface nécessiterait donc beaucoup plus de nœuds que pour l'inclusion unique située au centre de la matrice. Et si l'utilisation d'un maillage anisotrope permettrait de limiter le nombre de nœuds, nous avons vu au chapitre précédent qu'elle limiterait surtout la conservation du liquide.

Nous précisons également qu'utiliser des remaillages adaptatifs avec un aussi grand nombre de processeurs prend en proportion beaucoup plus de temps que sur un seul processeur. En effet le remaillage, bien que parallèle, nécessite de nombreuses communications entre les différents processeurs et le repartitionnement du domaine de calcul est lui aussi loin d'être immédiat. Si les deux raisons exposées ci-dessus nous ont déjà semblé amplement suffisantes pour seulement utiliser des maillages fixes, il est également utile de signaler que les calculs avec remaillage s'interrompent parfois brusquement sur le cluster du CEMEF, pour des raisons qui nous sont malheureusement encore inconnues. La perte de précision due au transport des champs n'est, elle non plus, pas à négliger pour justifier l'emploi du maillage fixe.

Nous nous intéressons à présent à une résolution mécanique, sans déplacement d'interface. Pour cela nous considérons un quart de l'éprouvette expérimentale, que nous discrétisons avec un maillage homogène et isotrope de $10\ \mu\text{m}$. Le maillage considéré comprend plus de 3 000 000 de nœuds et 16 000 000 d'éléments. Le domaine de calcul est un parallélépipède rectangle de dimension $1,12\ \text{mm} \times 1,43\ \text{mm} \times 1,08\ \text{mm}$. La figure 5.21a montre l'interface expérimentale entre l'air et le mélange liquide/solide 513 s après le début du test de traction. Elle est obtenue à partir d'une représentation en voxels associée à une image de microtomographie. La figure 5.21b montre la même interface pour la représentation éléments finis. Nous pouvons observer la très bonne reconstruction géométrique de l'interface avec le maillage éléments finis.

Les conditions aux limites appliquées sont les suivantes : (i) vitesse verticale de $0,1\ \mu\text{m}\cdot\text{s}^{-1}$ sur le plan supérieur, (ii) vitesse verticale nulle sur le plan inférieur, (iii) vitesse normale nulle sur les deux plans intérieurs de l'éprouvette, (iv) aucune condition sur les deux plans extérieurs. Ces conditions sont appliquées quelle que soit la phase présente aux bords du domaine et aucune couche de solide n'est cette fois-ci rajoutée. La résolution est alors la même qu'en deux dimensions. La figure 5.22 montre la répartition de la contrainte équivalente et de la vitesse de déformation généralisée sur une coupe 2D de la microstructure associée à la représentation éléments finis de la figure 5.21b. Nous pouvons voir que les valeurs de vitesse de déformation maximale sont situées dans les veines de liquide. La contrainte équivalente, quant à elle, est très hétérogène dans le solide, et ses maxima sont situés entre des veines de liquide, dans les ponts solides.

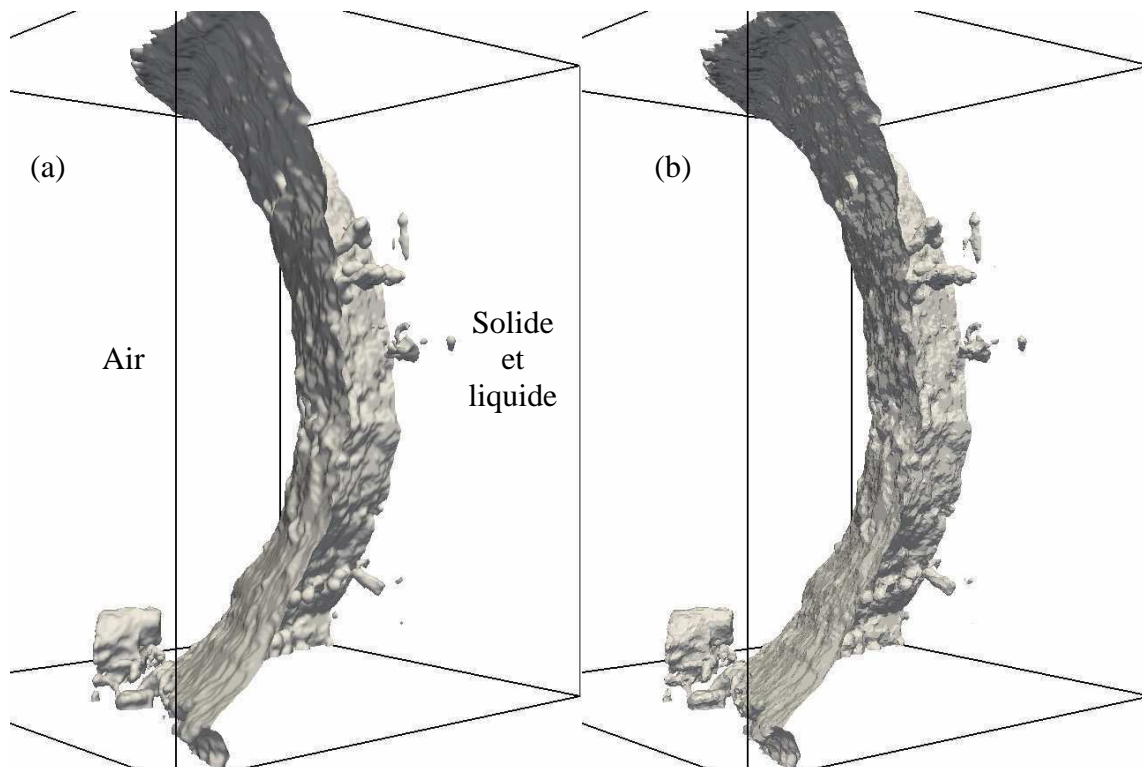


Figure 5.21. Frontière entre l'air et le mélange solide/liquide 513 s après le début du test de traction. Frontières (a) expérimentale et (b) numérique.

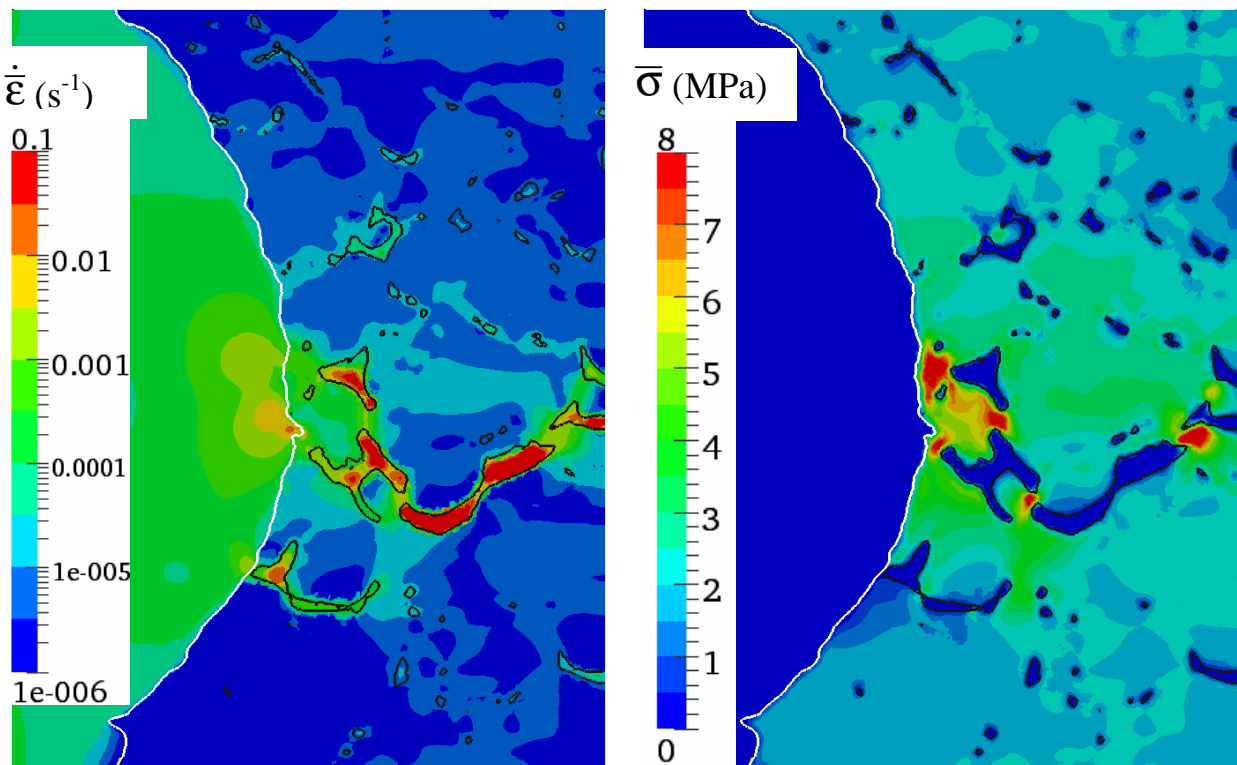


Figure 5.22. Résultats mécaniques sur une coupe 2D de l'éprouvette.

Après avoir testé les résolutions mécanique et level set séparément, nous lançons un calcul dynamique pour lequel les deux solveurs sont couplés. Nous pouvons grâce à cette simulation directement comparer les évolutions morphologiques des interfaces numérique et expérimentale. La figure 5.23 illustre une telle comparaison pour l'interface entre l'air et le mélange solide/liquide 1323 s après le début du test de traction expérimentale. Nous avons pris pour représentation numérique initiale celle montrée sur la figure 5.21b, juste avant la propagation expérimentale de l'air dans l'échantillon. Avec un pas de temps de 13,5 s, les 60 incréments de calcul effectués pour tracer la figure 5.23b ont nécessité un temps de simulation d'un peu plus d'une semaine avec 32 processeurs du cluster CEMEF. Nous constatons que l'accord avec l'expérience est globalement satisfaisant. En particulier la pénétration de l'air dans l'éprouvette s'effectue dans les mêmes régions et avec une amplitude du même ordre de grandeur.

Pour conforter cette comparaison qualitative et puisque nous possédons de nombreuses images de l'expérience de traction, nous pouvons faire un calcul purement mécanique à différents temps pour voir l'évolution de la contrainte équivalente moyenne dans la phase solide, et la comparer à l'évolution calculée pour le calcul dynamique. C'est ce que nous montre la figure 5.24. Nous voyons que la contrainte diminue au cours du temps dans les deux cas, en raison de l'endommagement de l'échantillon dû à la propagation de l'air. En outre, les valeurs calculées de manière statique et dynamique sont assez proches, ce qui est plutôt encourageant.

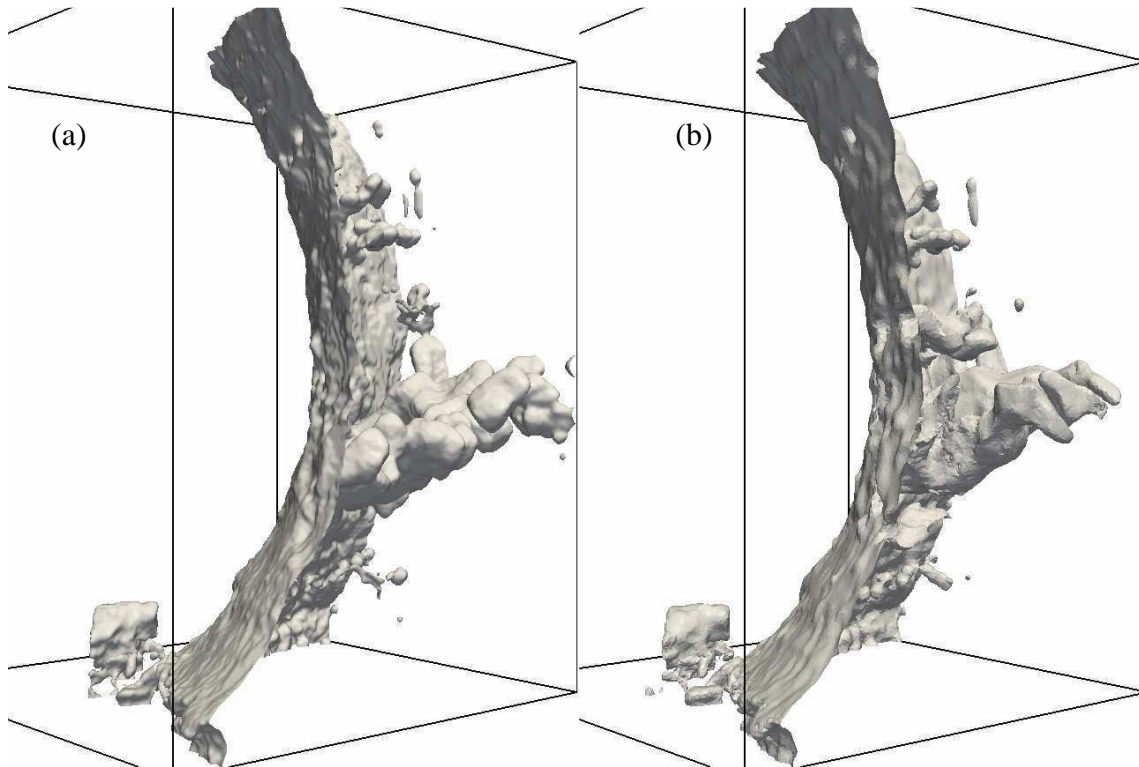


Figure 5.23. Frontière entre l'air et le mélange solide/liquide 1323 s après le début du test de traction. Frontières (a) expérimentale et (b) numérique.

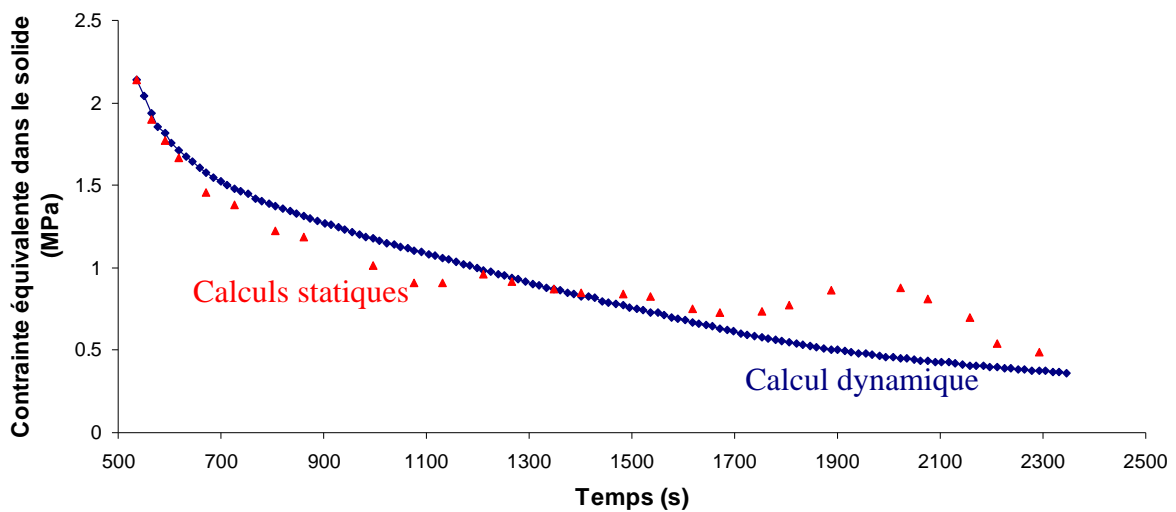


Figure 5.24. Evolution de la contrainte équivalente moyenne dans le solide au cours du temps.

Pour déterminer la pertinence du calcul dynamique, il est judicieux d'étudier également la variation du volume de liquide au cours de la simulation. Cette étude est illustrée sur la figure 5.25, qui nous donne également la variation du volume de liquide associée à l'expérience et aux calculs statiques. Le volume correspondant aux calculs statiques suit le volume expérimental, avec un certain écart dû à une discrétisation insuffisante du domaine de calcul. Le volume expérimental, quant à lui, augmente au début de l'expérience, comme expliqué par Terzi et al. [Ter 09], et il cesse d'augmenter à partir du moment où l'air commence à se propager dans l'éprouvette. Cette augmentation de volume est provoquée par un flux de liquide se déplaçant de la périphérie de l'éprouvette (zones inférieure et supérieure non

imagées) vers la zone centrale d'observation. Ce flux n'est pas pris en compte dans notre modèle. Par conséquent, le volume de liquide devrait constamment diminuer pour le calcul dynamique, au fur et à mesure que des poches de liquide sortent du domaine de calcul eulérien par sa face supérieure. Or ce n'est pas le cas au cours des 500 premières secondes de simulation (entre 513 s et environ 1000 s), ce qui constitue une preuve supplémentaire de l'aspect diffusif (ou non conservatif) de notre méthode level set. Il faut toutefois souligner que cette limitation est peut-être salutaire dans l'obtention de nos résultats, cet enrichissement artificiel en liquide permettant sans doute de faciliter la propagation de l'air au sein de l'échantillon.

Le volume de solide, quant à lui diminue tout au long de la simulation, ce qui est cohérent avec les conditions aux limites considérées. Il est remplacé par du gaz, dont le volume est en constante augmentation durant le calcul. Nous avons bien vérifié que le volume total des trois phases reste constant. La figure 5.26 montre l'évolution du volume des différentes phases au cours de la simulation.

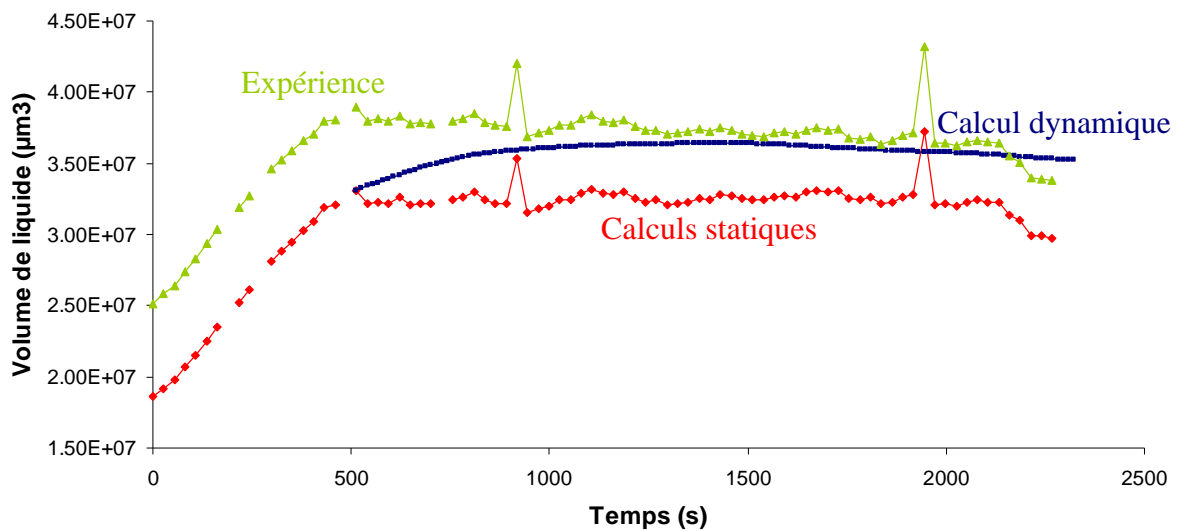


Figure 5.25. Evolution du volume de liquide au cours du temps.

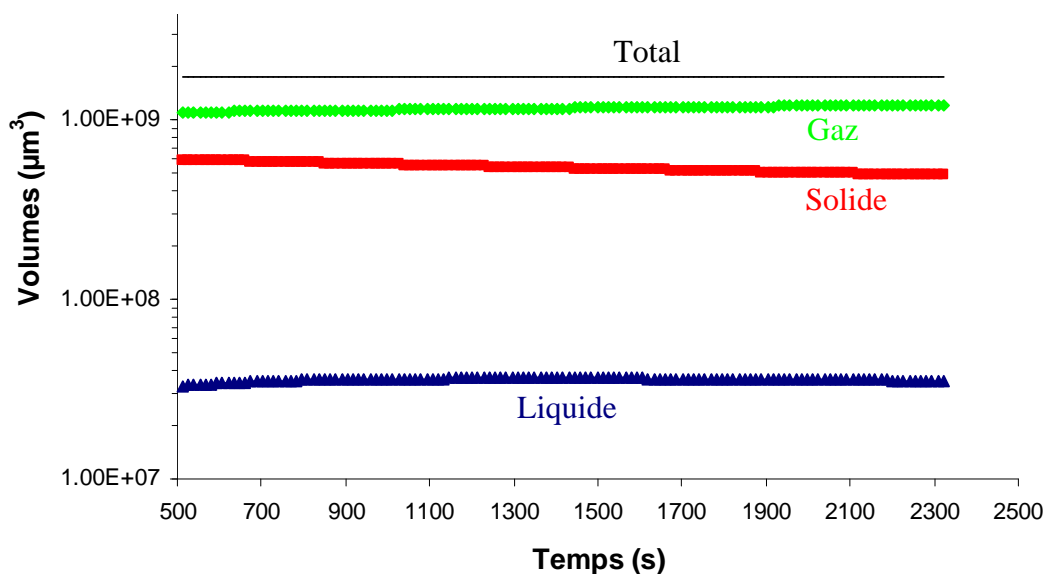


Figure 5.26. Evolution du volume des trois phases au cours du temps.

5. Conclusion

Nous venons de détailler dans ce chapitre la méthodologie employée pour parvenir à réaliser une simulation par éléments finis sur une éprouvette représentative d'un test de traction sur un échantillon cylindrique d'Al-8% Cu, ce qui nous a permis d'obtenir une comparaison encourageante entre simulation et expérience. Pour cela, nous avons essayé au maximum d'isoler chacun des paramètres à considérer afin d'étudier leur influence sur les résultats de simulation pour en déduire les valeurs optimales à adopter.

Nous tenons tout de même à souligner que malgré toutes les difficultés encore présentes, et bien que la physique de notre modèle reste à compléter, une telle simulation constitue à notre connaissance une première dans le domaine de la solidification des alliages métalliques, ce qui ouvre des perspectives intéressantes, que nous décrivons en conclusion générale de cette thèse.

Avant d'évoquer des pistes pour enrichir le modèle très simple actuellement utilisé, nous préférons tout d'abord sensibiliser le lecteur sur la nécessité absolue d'améliorer la fiabilité des solveurs mécanique et level set actuellement utilisés dans CIMLib. A titre d'exemple, les résolutions itératives pour le calcul dynamique présenté à la fin de ce chapitre n'ont pas convergé à 69 reprises. Cela ne signifie pas nécessairement que les résultats obtenus sont faux, mais augmenter le temps de calcul pour ne faire que dégrader des solutions déjà satisfaisantes ne nous semble pas particulièrement indispensable.

Pour le moment, nous avons dû nous résoudre à considérer seulement un quart de la géométrie expérimentale. Il serait évidemment intéressant de considérer l'ensemble de l'éprouvette pour nos simulations. Pour cela, plusieurs solutions nous semblent envisageables : le recours (i) à un maillage adaptatif, (ii) à un plus grand nombre de processeurs, (iii) à un type de résolution plus rapide pour les systèmes linéaires, en faisant par exemple appel à un solveur multigrille [Elk 10].

- Conclusion générale -

« A vaincre sans péril, on triomphe sans gloire. »

Pierre Corneille

Nous venons de présenter une méthode originale permettant de coupler des simulations par éléments finis à des expériences de microtomographie, en initialisant les simulations à partir des données expérimentales. Grâce à cela, nous avons pu effectuer des comparaisons directes entre résultats expérimentaux et numériques pour le test de traction d'un alliage d'aluminium-cuivre à l'état pâteux. Les travaux accomplis dans le cadre de cette thèse souffrent d'un très grand nombre de limitations, mais ils ont permis de nous convaincre du bien-fondé de notre approche, dont les principales étapes sont résumées ci-dessous.

Nous avons dans un premier temps expliqué comment construire une représentation par éléments finis à partir de données de microtomographie. Pour cela, nous traitons les images expérimentales afin d'identifier les différentes phases présentes, ce qui permet l'extraction de maillages surfaciques par l'algorithme des marching cubes et la construction d'un maillage volumique par la méthode d'immersion de volume. Cette méthode consiste à calculer la distance aux maillages surfaciques et à adapter le maillage avec des éléments plus fins à l'interface. Nous rappelons cependant avoir simplement utilisé des maillages homogènes et isotropes pour nos applications.

Nous avons par la suite présenté la méthode de résolution des équations de Stokes utilisée dans CIMLib et dans cette thèse, qui s'appuie sur une formulation monolithique P1+/P1 en vitesse-pression, pour laquelle les champs de vitesse et de pression sont calculés simultanément sur tous les nœuds du maillage, et ce quelle que soit la phase qui leur est associée. Pour cette résolution monolithique, nous utilisons une loi de mélange aux interfaces afin que les viscosités des différents milieux en présence soient considérées. Nous avons également donné quelques pistes concernant la résolution des systèmes linéaires assemblés par la méthode des éléments finis, en établissant une relation entre la précision de la résolution et le rapport des viscosités.

Le champ de vitesse calculé permet d'actualiser les fonctions level set représentatives des interfaces par une méthode de réinitialisation convective. Nous avons détaillé le fonctionnement de cette méthode avant de la soumettre à une étude approfondie et à une comparaison avec la méthode level set particulière, reconnue dans la littérature pour ses qualités. Malgré la supériorité de la level set particulière sur notre méthode, nous avons tout de même obtenu avec cette dernière une conservation du volume satisfaisante pour une inclusion sphérique soumise à un test de traction uniaxial. Toutefois, nous avons mis en avant la difficulté à obtenir une réinitialisation convenable. Nous espérons en outre avoir contribué par ce travail à la remise en cause de certaines assertions relatives aux solveurs de réinitialisation convective développés dans CIMLib. Ainsi, dans la conclusion de leurs travaux de thèse, Ville et François affirment à propos du solveur de réinitialisation convective de CIMLib (le LevellerS dans leur cas) qu'il « permet une représentation très fidèle de l'évolution d'une surface libre et sans aucune diffusion numérique » [Vil 11b] et que « son caractère local permet de réduire le temps de calcul » [Fra 11], ce qui nous semble inexact pour les applications présentées dans cette thèse. Ces deux auteurs évoquent également le recours à un estimateur d'erreur et à des éléments anisotropes à l'interface comme une

Conclusion générale

perspective séduisante, mais nous appelons à la plus grande prudence à ce sujet. En effet, il n'existe pas à notre connaissance d'étude montrant l'évolution du volume de matière au cours du temps pour des champs de vitesse incompressibles sur ce type de maillage.

Dans le dernier chapitre de cette thèse, nous avons établi un couplage entre les solveurs mécanique et level set. Nous avons tout d'abord considéré une inclusion placée au centre d'une matrice cubique et de viscosité beaucoup plus faible que celle de la matrice, soumise à un test de traction uniaxial, avant d'étudier des tests de traction sur une coupe 2D issue d'une image de microtomographie. Ces tests ont permis de déterminer les paramètres à utiliser sur une simulation 3D, pour laquelle nous avons pu faire une comparaison qualitative avec les résultats expérimentaux.

Nous tenons également à signaler que l'utilisation de la méthode décrite tout au long de cette thèse peut présenter un intérêt pour accroître notre compréhension des phénomènes physiques impliqués dans tous les domaines de recherche pour lesquels des microtomographies aux rayons X et des simulations par éléments finis sont réalisables.

En raison des difficultés rencontrées dans l'utilisation des outils disponibles dans CIMLib, nous avons déjà proposé dans les troisième et quatrième chapitres des perspectives afin d'améliorer la robustesse et la qualité des solveurs mécanique et level set. A la lumière des limitations observées, nous pouvons également nous demander si le choix d'utiliser CIMLib était le plus pertinent. En effet pouvoir utiliser des maillages tétraédriques anisotropes calculés par un estimateur d'erreur et adaptés à la définition des interfaces apparaissait a priori séduisant, mais nous avons en réalité dû nous résoudre à employer des maillages isotropes et homogènes. Dans ces conditions, il aurait pu être intéressant d'utiliser un maillage de type octree [Los 04] avec des éléments correspondants aux voxels de l'image initiale à l'interface. Pour cela, l'emploi de PhysBAM [Dub 11], librairie gratuite de simulation multiphysique développée à l'université Stanford et disposant de solveurs mécanique et level set, nous semble particulièrement adapté.

En dehors des perspectives d'amélioration évoquées en ce qui concerne les méthodes de résolution des équations, nous pouvons énoncer un certain nombre d'ajouts qui permettraient d'enrichir notre modèle numérique.

Il pourrait ainsi être intéressant de considérer les aspects suivants : compressibilité de l'air, effets de tension de surface et de mouillabilité, diffusion de l'hydrogène [Car 07, Sas 08, Fel 10] et définition d'un critère d'apparition des porosités. Ces aspects sont tous reliés à la création et à la propagation des pores au sein de l'éprouvette. En l'état actuel, notre modèle ne permet en effet que de simuler l'endommagement par pénétration de l'air depuis l'extérieur de l'échantillon. Ainsi, un pore isolé gardera son volume constant au cours d'un test de traction, ce qui ne correspond pas à la réalité physique, d'où la nécessité de considérer l'air comme un fluide compressible. La tension de surface, la mouillabilité et la gravité ont quant à elles une grande influence sur la géométrie des veines liquides et donc sur le comportement mécanique du mélange pâteux. De plus, définir un critère d'apparition des porosités est obligatoire pour simuler la cavitation dans les films liquides.

En outre, l'expérience que nous avons essayé de reproduire par simulation numérique était un test de traction effectué à température constante, et nous avons estimé légitime de ne pas considérer la solidification dans notre modèle numérique. Toutefois, la solidification ne peut évidemment pas être négligée pour comprendre les conditions d'apparition de la fissuration à chaud et les expériences de traction anisotherme effectuées par nos partenaires du SIMaP montrent d'ailleurs bon nombre de différences avec les expériences de traction isotherme. Il est donc nécessaire de résoudre l'équation de conservation du soluté, l'équation de conservation de l'énergie ne semblant pas primordiale à résoudre car la température peut être

Conclusion générale

considérée comme homogène dans les échantillons millimétriques contenus à l'intérieur d'un four.

Afin d'ajouter la solidification dans notre modèle pour simuler des expériences de traction anisotherme, nous préconisons l'approche suivante consistant à (i) utiliser le solveur de convection diffusion réaction développé dans CIMLib au cours de la thèse de Hachem [Hac 09], (ii) valider son couplage avec la résolution level set en étudiant des problèmes de Stefan [Vui 93] tel que celui de la solidification en coin simulé par Tan & Zabaras [Tan 06] et dont la solution analytique est donnée par Rathjen & Jiji [Rat 71], (iii) rajouter les termes de courbure et d'anisotropie afin de retrouver les évolutions de structures dendritiques décrites dans la littérature [Kim 00, Gib 03, Ram 04, Tan 06], (iv) utiliser les données de microtomographie de Limodin et al. obtenues lors du maintien isotherme d'un échantillon d'aluminium-cuivre [Lim 07] pour initialiser une simulation numérique et la comparer avec les résultats expérimentaux, (v) reproduire cette comparaison pour une expérience de solidification dendritique [Lim 09].

De tels travaux ont déjà initiés au CEMEF. François a par exemple retrouvé un bon accord entre ses résultats numériques et la solution analytique pour le cas de solidification en coin d'un corps pur [Fra 11]. Plusieurs stages sur la modélisation par level set de la croissance dendritique ont également eu lieu [Mzo 09, Vu 07, Zar 08], mais ils n'ont pas permis d'obtenir des résultats en accord avec la théorie de la solvabilité microscopique [Kar 98].

D'autres améliorations de ce travail sont envisageables en ce qui concerne les expériences de microtomographie et la corrélation des images. Nous les résumons à présent brièvement et laissons le soin à nos partenaires des laboratoires SIMaP et Navier de les détailler.

La microtomographie aux rayons X faisant des progrès rapidement, il est maintenant possible d'effectuer des expériences avec des acquisitions d'image à la fois plus rapides et plus précises, permettant donc un accès à de meilleures résolutions et à des vitesses de refroidissement beaucoup plus importantes que celles considérées jusqu'à présent. Des progrès peuvent aussi être obtenus au niveau de l'instrumentation des expériences. Un capteur d'effort permettrait par exemple d'effectuer des comparaisons entre les contraintes calculées par simulation numérique et par l'expérience.

Les outils de corrélation d'images, quant à eux, seraient avantageux pour renforcer le lien entre les expériences et les simulations, que ce soit pour déduire précisément les conditions aux limites à appliquer dans les simulations, ou pour comparer les résultats expérimentaux et numériques sous un angle différent. Il est en effet possible de transformer nos résultats numériques par éléments finis en images de voxels du même format que les données expérimentales, afin d'appliquer des techniques de corrélation similaires à celles appliquées aux images expérimentales [Ter 09].

L'approche employée dans cette thèse est coûteuse en temps de calcul et ne peut être employée que pour des volumes de faible dimension. S'il apparaît indispensable de simuler d'autres expériences de microtomographie pour enrichir notre modèle, il serait également très intéressant d'utiliser des volumes élémentaires représentatifs en modifiant des paramètres tels que la taille des grains, la fraction de solide ou la fraction de porosité, afin de déterminer une équation décrivant le comportement mécanique de l'alliage pâteux, pouvant être employé dans des simulations à plus grande échelle. Un tel travail a déjà été mené à bien par Phillion, Cockcroft & Lee [Phi 09a].

Conclusion générale

- Annexe 1 -

Code pour les Marching Squares

```

#include <fstream>
#include <iostream>

using namespace std;

// Table de connectivité
unsigned char TableSegments[16][5]=
{
    {4,4,4,4,4}, // 0
    {3,0,4,4,4}, // 1
    {0,1,4,4,4}, // 2
    {3,1,4,4,4}, // 3
    {1,2,4,4,4}, // 4
    {3,0,1,2,4}, // 5
    {0,2,4,4,4}, // 6
    {3,2,4,4,4}, // 7
    {2,3,4,4,4}, // 8
    {2,0,4,4,4}, // 9
    {0,1,2,3,4}, // 10
    {2,1,4,4,4}, // 11
    {1,3,4,4,4}, // 12
    {1,0,4,4,4}, // 13
    {0,3,4,4,4}, // 14
    {4,4,4,4,4}, // 15
};

int main()
{
    // Choix des paramètres
    int L=78; // Largeur de l'image
    int H=58; // Hauteur de l'image
    double a=2.0; // Taille d'un pixel
    unsigned char SeuilImage=151; // Seuil à appliquer
    ifstream Entree("78x58.raw",ios_base::binary); // Fichier d'entrée
    if(!Entree) // vérification du fichier d'entrée
    {
        cout << "pas de fichier d'entree\n";return -1;}
    ofstream Sortie("Surface.txt"); // Fichier de sortie
    // Début du code
    double S=static_cast<double>(SeuilImage)-1e-3;
    int L_1=L-1,L_2=L-2,H_1=H-1,Hp1=H+1;
    unsigned char Index,NumeroArete;
    int i,j,Noeuds=0,Elements=0,C1,C2,jp1,jL,jLp1,jp1L,jp1Lp1;
    double NG1,NG2,I,x,y;

```

Annexe 1

```

int * NN = new int[L+Hp1];NN[0]=0;
unsigned char * Image = new unsigned char[L*H];
// Ligne laissée pour l'en-tête
Sortie << "          \n";
// Lecture de l'image
int DecalageInit=H_1*L,Decalage=-L*2;
Entree.seekg(DecalageInit,std::ios_base::beg);
for (j=0;j<H;j++)
{
    jL=j*L;
    for (i=0;i<L;i++)
    {
        Image[i+jL]=Entree.get();
    }
    Entree.seekg(Decalage,std::ios_base::cur);
}
// Ecriture des noeuds suivant l'abscisse
for (j=0;j<H;j++)
{
    jL=j*L;
    jLp1=jL+1;
    for (i=0;i<L_1;i++)
    {
        if(Image[i+jL]<S!=Image[i+jLp1]<S)
        {
            Noeuds+=1;
            NG1=static_cast<double>(Image[i+jL]);
            NG2=static_cast<double>(Image[i+jLp1]);
            I=(S-NG1)/(NG2-NG1)*a;
            x=(i+0.5)*a;y=(j+0.5)*a;
            Sortie << x+I << "    " << y << endl;
        }
    }
    NN[j+1]=Noeuds;
}
// Ecriture des noeuds suivant l'ordonnée
for (i=0;i<L;i++)
{
    for(j=0;j<H_1;j++)
    {
        if(Image[i+j*L]<S!=Image[i+(j+1)*L]<S)
        {
            Noeuds+=1;

            NG1=static_cast<double>(Image[i+j*L]);
            NG2=static_cast<double>(Image[i+(j+1)*L]);
            I=(S-NG1)/(NG2-NG1)*a;
            x=(i+0.5)*a;y=(j+0.5)*a;
            Sortie << x << "    " << y+I << endl;
        }
    }
}

```

Annexe 1

```

    }
    NN[i+H+1]=Noeuds;
}
int NNAS; // Nombre de noeuds utilisés sur l'arête supérieure
// Calcul des éléments
for(j=0;j<H_1;j++)
{
    jL=j*L;jp1=j+1;jLp1=jL+1;jp1L=jp1*L;jp1Lp1=jp1L+1;
    NNAS=0;
    for(i=0;i<L_2;i++)
    {
        Index=0;
        NumeroArete=0;
        // Calcul de l'index
        if(Image[i+jL]>=S)
            Index+=1;
        if(Image[i+jLp1]>=S)
            Index+=2;
        if(Image[i+jp1Lp1]>=S)
            Index+=4;
        if(Image[i+jp1L]>=S)
            Index+=8;
        while(TableSegments[Index][NumeroArete]!=4)
        {
            switch(TableSegments[Index][NumeroArete])
            {
                case 0 : C1=NN[j]+1;NN[j]+=1; break;
                case 1 : C1=NN[i+Hp1]+1; break;
                case 2 : C1=NN[j+1]+NNAS+1;NNAS+=1; break;
                case 3 : C1=NN[i+H]+1;NN[i+H]+=1; break;
            }
            switch(TableSegments[Index][NumeroArete+1])
            {
                case 0 : C2=NN[j]+1;NN[j]+=1; break;
                case 1 : C2=NN[i+Hp1]+1; break;
                case 2 : C2=NN[j+1]+NNAS+1;NNAS+=1; break;
                case 3 : C2=NN[i+H]+1;NN[i+H]+=1; break;
            }
            NumeroArete+=2;
            Elements+=1;
            Sortie << C1 << "    " << C2 << " 0" << endl;
        }
    }
    i=L_2;Index=0;
    NumeroArete=0;
    // Calcul de l'index sur la dernière colonne
    if(Image[i+jL]>=S)
        Index+=1;
    if(Image[i+jLp1]>=S)
        Index+=2;
}

```


Annexe 1

```

if(Image[i+jp1Lp1]>=S)
    Index+=4;
if(Image[i+jp1L]>=S)
    Index+=8;
while(TableSegments[Index][NumeroArete]!=4)
{
    switch(TableSegments[Index][NumeroArete])
    {
        case 0 : C1=NN[j]+1;NN[j]+=1; break;
        case 1 : C1=NN[i+Hp1]+1;NN[i+Hp1]+=1; break;
        case 2 : C1=NN[j+1]+NNAS+1;NNAS+=1; break;
        case 3 : C1=NN[i+H]+1;NN[i+H]+=1; break;
    }
    switch(TableSegments[Index][NumeroArete+1])
    {
        case 0 : C2=NN[j]+1;NN[j]+=1; break;
        case 1 : C2=NN[i+Hp1]+1;NN[i+Hp1]+=1; break;
        case 2 : C2=NN[j+1]+NNAS+1;NNAS+=1; break;
        case 3 : C2=NN[i+H]+1;NN[i+H]+=1; break;
    }
    NumeroArete+=2;
    Elements+=1;
    Sortie << C1 << "    " << C2 << " 0" << endl;
}
}
// Ecriture de l'en tête
Sortie.seekp(0,ios::beg);
Sortie << Noeuds << "    2    " << Elements << "    3";

Sortie.close();
return 0;
}

```

- Annexe 2 -

Code pour les Marching Cubes

```

#include <fstream>
#include <iostream>

using namespace std;

// Table de connectivité
unsigned char TableTriangles[256][16]=
{
    {12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12}, // 0
    {8,3,0,12,12,12,12,12,12,12,12,12,12,12,12,12}, // 1
    {9,0,1,12,12,12,12,12,12,12,12,12,12,12,12,12}, // 2
    {8,1,9,3,1,8,12,12,12,12,12,12,12,12,12,12}, // 3
    {1,2,11,12,12,12,12,12,12,12,12,12,12,12,12,12}, // 4
    {0,8,3,1,2,11,12,12,12,12,12,12,12,12,12,12}, // 5
    {9,2,11,0,2,9,12,12,12,12,12,12,12,12,12,12}, // 6
    {2,8,3,2,11,8,11,9,8,12,12,12,12,12,12,12}, // 7
    {10,2,3,12,12,12,12,12,12,12,12,12,12,12,12,12}, // 8
    {10,0,8,2,0,10,12,12,12,12,12,12,12,12,12,12}, // 9
    {1,9,0,2,3,10,12,12,12,12,12,12,12,12,12,12}, // 10
    {1,10,2,1,9,10,9,8,10,12,12,12,12,12,12,12}, // 11
    {3,11,1,10,11,3,12,12,12,12,12,12,12,12,12,12}, // 12
    {0,11,1,0,8,11,8,10,11,12,12,12,12,12,12,12}, // 13
    {3,9,0,3,10,9,10,11,9,12,12,12,12,12,12,12}, // 14
    {9,8,11,8,10,11,12,12,12,12,12,12,12,12,12,12}, // 15
    {4,7,8,12,12,12,12,12,12,12,12,12,12,12,12,12}, // 16
    {4,3,0,7,3,4,12,12,12,12,12,12,12,12,12,12}, // 17
    {4,7,8,9,0,1,12,12,12,12,12,12,12,12,12,12}, // 18
    {4,1,9,4,7,1,7,3,1,12,12,12,12,12,12,12}, // 19
    {4,7,8,11,1,2,12,12,12,12,12,12,12,12,12,12}, // 20
    {3,4,7,3,0,4,1,2,11,12,12,12,12,12,12,12}, // 21
    {9,2,11,9,0,2,8,4,7,12,12,12,12,12,12,12}, // 22
    {4,7,3,4,3,11,4,11,9,11,3,2,12,12,12,12}, // 23
    {8,4,7,3,10,2,12,12,12,12,12,12,12,12,12,12}, // 24
    {10,4,7,10,2,4,2,0,4,12,12,12,12,12,12,12}, // 25
    {9,0,1,8,4,7,2,3,10,12,12,12,12,12,12,12}, // 26
    {1,10,2,9,10,1,9,7,10,9,4,7,12,12,12,12}, // 27
    {3,11,1,3,10,11,7,8,4,12,12,12,12,12,12,12}, // 28
    {7,0,4,7,11,0,7,10,11,1,0,11,12,12,12,12}, // 29
    {4,7,8,9,0,10,9,10,11,10,0,3,12,12,12,12}, // 30
    {4,7,10,4,10,9,9,10,11,12,12,12,12,12,12,12}, // 31
    {4,9,5,12,12,12,12,12,12,12,12,12,12,12,12,12}, // 32
    {9,5,4,0,8,3,12,12,12,12,12,12,12,12,12,12}, // 33
}

```

Annexe 2

{5,0,1,4,0,5,12,12,12,12,12,12,12,12,12,12},	// 34
{8,5,4,8,3,5,3,1,5,12,12,12,12,12,12,12},	// 35
{5,4,9,11,1,2,12,12,12,12,12,12,12,12,12,12},	// 36
{4,9,5,0,8,3,11,1,2,12,12,12,12,12,12,12},	// 37
{5,2,11,5,4,2,4,0,2,12,12,12,12,12,12,12},	// 38
{8,3,2,4,8,2,4,2,11,4,11,5,12,12,12,12},	// 39
{4,9,5,3,10,2,12,12,12,12,12,12,12,12,12,12},	// 40
{0,10,2,0,8,10,4,9,5,12,12,12,12,12,12,12},	// 41
{0,5,4,0,1,5,2,3,10,12,12,12,12,12,12,12},	// 42
{4,8,10,4,10,1,4,1,5,2,1,10,12,12,12,12},	// 43
{11,3,10,11,1,3,9,5,4,12,12,12,12,12,12,12},	// 44
{5,4,9,11,1,8,11,8,10,8,1,0,12,12,12,12},	// 45
{5,10,11,5,0,10,5,4,0,0,3,10,12,12,12,12},	// 46
{5,4,8,5,8,11,11,8,10,12,12,12,12,12,12,12},	// 47
{9,7,8,5,7,9,12,12,12,12,12,12,12,12,12,12},	// 48
{9,3,0,9,5,3,5,7,3,12,12,12,12,12,12,12},	// 49
{0,7,8,0,1,7,1,5,7,12,12,12,12,12,12,12},	// 50
{5,7,1,1,7,3,12,12,12,12,12,12,12,12,12,12},	// 51
{9,7,8,9,5,7,11,1,2,12,12,12,12,12,12,12},	// 52
{11,1,2,9,5,0,5,3,0,5,7,3,12,12,12,12},	// 53
{11,5,7,11,7,0,11,0,2,8,0,7,12,12,12,12},	// 54
{2,11,5,2,5,3,3,5,7,12,12,12,12,12,12,12},	// 55
{7,9,5,7,8,9,3,10,2,12,12,12,12,12,12,12},	// 56
{9,2,0,9,7,2,9,5,7,7,10,2,12,12,12,12},	// 57
{2,3,10,0,1,8,1,7,8,1,5,7,12,12,12,12},	// 58
{10,2,1,10,1,7,7,1,5,12,12,12,12,12,12,12},	// 59
{1,10,11,1,3,10,5,7,9,7,8,9,12,12,12,12},	// 60
{5,7,0,5,0,9,7,10,0,1,0,11,10,11,0,12},	// 61
{10,11,0,10,0,3,11,5,0,8,0,7,5,7,0,12},	// 62
{5,7,10,11,5,10,12,12,12,12,12,12,12,12,12,12},	// 63
{5,11,6,12,12,12,12,12,12,12,12,12,12,12,12,12},	// 64
{5,11,6,0,8,3,12,12,12,12,12,12,12,12,12,12},	// 65
{9,0,1,5,11,6,12,12,12,12,12,12,12,12,12,12},	// 66
{1,8,3,1,9,8,5,11,6,12,12,12,12,12,12,12},	// 67
{1,6,5,2,6,1,12,12,12,12,12,12,12,12,12,12},	// 68
{1,6,5,1,2,6,3,0,8,12,12,12,12,12,12,12},	// 69
{9,6,5,9,0,6,0,2,6,12,12,12,12,12,12,12},	// 70
{5,2,6,5,8,2,5,9,8,3,2,8,12,12,12,12},	// 71
{6,5,11,10,2,3,12,12,12,12,12,12,12,12,12,12},	// 72
{10,0,8,10,2,0,11,6,5,12,12,12,12,12,12,12},	// 73
{5,11,6,1,9,0,10,2,3,12,12,12,12,12,12,12},	// 74
{5,11,6,1,9,2,9,10,2,9,8,10,12,12,12,12},	// 75
{6,3,10,6,5,3,5,1,3,12,12,12,12,12,12,12},	// 76
{0,5,1,0,10,5,0,8,10,10,6,5,12,12,12,12},	// 77
{6,3,10,5,3,6,5,0,3,5,9,0,12,12,12,12},	// 78
{6,5,9,6,9,10,10,9,8,12,12,12,12,12,12,12},	// 79
{5,11,6,4,7,8,12,12,12,12,12,12,12,12,12,12},	// 80
{4,3,0,4,7,3,6,5,11,12,12,12,12,12,12,12},	// 81
{8,4,7,9,0,1,6,5,11,12,12,12,12,12,12,12},	// 82
{6,5,11,4,7,9,7,1,9,7,3,1,12,12,12,12},	// 83

Annexe 2

{6,1,2,6,5,1,4,7,8,12,12,12,12,12,12,12},	// 84
{4,7,0,0,7,3,6,5,1,6,1,2,12,12,12,12},	// 85
{8,4,7,9,0,5,0,6,5,0,2,6,12,12,12,12},	// 86
{7,3,9,7,9,4,3,2,9,5,9,6,2,6,9,12},	// 87
{4,7,8,6,5,11,3,10,2,12,12,12,12,12,12,12},	// 88
{5,11,6,4,7,2,4,2,0,2,7,10,12,12,12,12},	// 89
{2,3,10,9,0,1,4,7,8,6,5,11,12,12,12,12},	// 90
{9,2,1,9,10,2,9,4,10,7,10,4,5,11,6,12},	// 91
{4,7,8,6,5,10,5,3,10,5,1,3,12,12,12,12},	// 92
{5,1,10,5,10,6,1,0,10,7,10,4,0,4,10,12},	// 93
{5,10,6,5,3,10,5,9,3,0,3,9,4,7,8,12},	// 94
{6,5,9,6,9,10,4,7,9,7,10,9,12,12,12,12},	// 95
{4,11,6,9,11,4,12,12,12,12,12,12,12,12,12,12},	// 96
{4,11,6,4,9,11,0,8,3,12,12,12,12,12,12,12},	// 97
{11,0,1,11,6,0,6,4,0,12,12,12,12,12,12,12},	// 98
{11,6,4,11,4,3,11,3,1,3,4,8,12,12,12,12},	// 99
{1,4,9,1,2,4,2,6,4,12,12,12,12,12,12,12},	// 100
{8,3,0,4,9,2,4,2,6,2,9,1,12,12,12,12},	// 101
{4,0,6,6,0,2,12,12,12,12,12,12,12,12,12,12},	// 102
{8,3,2,8,2,4,4,2,6,12,12,12,12,12,12,12},	// 103
{11,4,9,11,6,4,10,2,3,12,12,12,12,12,12,12},	// 104
{8,2,0,8,10,2,9,11,4,11,6,4,12,12,12,12},	// 105
{10,2,3,11,6,1,6,0,1,6,4,0,12,12,12,12},	// 106
{6,4,1,6,1,11,4,8,1,2,1,10,8,10,1,12},	// 107
{9,1,3,9,3,6,9,6,4,10,6,3,12,12,12,12},	// 108
{8,10,1,8,1,0,10,6,1,9,1,4,6,4,1,12},	// 109
{3,10,6,3,6,0,0,6,4,12,12,12,12,12,12,12},	// 110
{6,4,8,10,6,8,12,12,12,12,12,12,12,12,12,12},	// 111
{7,11,6,7,8,11,8,9,11,12,12,12,12,12,12,12},	// 112
{0,9,11,0,11,7,0,7,3,6,7,11,12,12,12,12},	// 113
{0,1,11,8,0,11,8,11,6,8,6,7,12,12,12,12},	// 114
{11,6,7,11,7,1,1,7,3,12,12,12,12,12,12,12},	// 115
{7,8,9,7,9,2,7,2,6,2,9,1,12,12,12,12},	// 116
{2,6,9,2,9,1,6,7,9,0,9,3,7,3,9,12},	// 117
{7,8,0,7,0,6,6,0,2,12,12,12,12,12,12,12},	// 118
{2,6,7,3,2,7,12,12,12,12,12,12,12,12,12,12},	// 119
{3,10,2,7,8,6,8,11,6,8,9,11,12,12,12,12},	// 120
{2,0,7,2,7,10,0,9,7,6,7,11,9,11,7,12},	// 121
{8,6,7,8,11,6,8,0,11,1,11,0,3,10,2,12},	// 122
{10,2,1,10,1,7,11,6,1,6,7,1,12,12,12,12},	// 123
{8,9,6,8,6,7,9,1,6,10,6,3,1,3,6,12},	// 124
{9,1,0,7,10,6,12,12,12,12,12,12,12,12,12,12},	// 125
{3,10,6,3,6,0,7,8,6,8,0,6,12,12,12,12},	// 126
{6,7,10,12,12,12,12,12,12,12,12,12,12,12,12},	// 127
{6,10,7,12,12,12,12,12,12,12,12,12,12,12,12},	// 128
{7,6,10,8,3,0,12,12,12,12,12,12,12,12,12,12},	// 129
{9,0,1,7,6,10,12,12,12,12,12,12,12,12,12,12},	// 130
{8,1,9,8,3,1,10,7,6,12,12,12,12,12,12,12},	// 131
{11,1,2,6,10,7,12,12,12,12,12,12,12,12,12,12},	// 132
{8,3,0,10,7,6,1,2,11,12,12,12,12,12,12,12},	// 133

Annexe 2

{2,9,0,2,11,9,6,10,7,12,12,12,12,12,12},	// 134
{7,6,10,8,3,11,8,11,9,11,3,2,12,12,12,12},	// 135
{2,7,6,3,7,2,12,12,12,12,12,12,12,12,12},	// 136
{7,0,8,7,6,0,6,2,0,12,12,12,12,12,12},	// 137
{2,7,6,2,3,7,0,1,9,12,12,12,12,12,12},	// 138
{7,9,8,7,2,9,7,6,2,2,1,9,12,12,12,12},	// 139
{11,7,6,11,1,7,1,3,7,12,12,12,12,12,12},	// 140
{0,11,1,8,11,0,8,6,11,8,7,6,12,12,12,12},	// 141
{0,11,9,0,7,11,0,3,7,6,11,7,12,12,12,12},	// 142
{7,6,11,7,11,8,8,11,9,12,12,12,12,12,12},	// 143
{6,8,4,10,8,6,12,12,12,12,12,12,12,12,12},	// 144
{3,6,10,3,0,6,0,4,6,12,12,12,12,12,12},	// 145
{8,6,10,8,4,6,9,0,1,12,12,12,12,12,12},	// 146
{9,3,1,9,6,3,9,4,6,10,3,6,12,12,12,12},	// 147
{6,8,4,6,10,8,2,11,1,12,12,12,12,12,12},	// 148
{11,1,2,6,10,0,6,0,4,0,10,3,12,12,12,12},	// 149
{9,0,11,11,0,2,8,4,6,8,6,10,12,12,12,12},	// 150
{11,9,3,11,3,2,9,4,3,10,3,6,4,6,3,12},	// 151
{8,2,3,8,4,2,4,6,2,12,12,12,12,12,12},	// 152
{4,6,0,6,2,0,12,12,12,12,12,12,12,12,12},	// 153
{9,0,1,8,4,3,4,2,3,4,6,2,12,12,12,12},	// 154
{1,9,4,1,4,2,2,4,6,12,12,12,12,12,12},	// 155
{11,4,6,11,3,4,11,1,3,3,8,4,12,12,12,12},	// 156
{11,1,0,11,0,6,6,0,4,12,12,12,12,12,12},	// 157
{4,6,3,4,3,8,6,11,3,0,3,9,11,9,3,12},	// 158
{4,6,11,9,4,11,12,12,12,12,12,12,12,12,12},	// 159
{4,9,5,7,6,10,12,12,12,12,12,12,12,12,12},	// 160
{5,4,9,7,6,10,0,8,3,12,12,12,12,12,12},	// 161
{5,0,1,5,4,0,7,6,10,12,12,12,12,12,12},	// 162
{6,10,7,5,4,3,5,3,1,3,4,8,12,12,12,12},	// 163
{9,5,4,11,1,2,7,6,10,12,12,12,12,12,12},	// 164
{4,9,5,6,10,7,0,8,3,11,1,2,12,12,12,12},	// 165
{7,6,10,5,4,11,4,2,11,4,0,2,12,12,12,12},	// 166
{4,11,5,4,2,11,4,8,2,3,2,8,7,6,10,12},	// 167
{7,2,3,7,6,2,5,4,9,12,12,12,12,12,12},	// 168
{9,5,4,0,8,6,0,6,2,6,8,7,12,12,12,12},	// 169
{6,3,7,6,2,3,4,0,5,0,1,5,12,12,12,12},	// 170
{6,2,8,6,8,7,2,1,8,4,8,5,1,5,8,12},	// 171
{4,9,5,7,6,1,7,1,3,1,6,11,12,12,12,12},	// 172
{8,1,0,8,11,1,8,7,11,6,11,7,4,9,5,12},	// 173
{4,0,11,4,11,5,0,3,11,6,11,7,3,7,11,12},	// 174
{7,6,11,7,11,8,5,4,11,4,8,11,12,12,12,12},	// 175
{6,9,5,6,10,9,10,8,9,12,12,12,12,12,12},	// 176
{6,10,3,5,6,3,5,3,0,5,0,9,12,12,12,12},	// 177
{0,1,5,0,5,10,0,10,8,10,5,6,12,12,12,12},	// 178
{6,10,3,6,3,5,5,3,1,12,12,12,12,12,12},	// 179
{1,2,11,9,5,10,9,10,8,10,5,6,12,12,12,12},	// 180
{5,0,9,5,3,0,5,6,3,10,3,6,11,1,2,12},	// 181
{10,8,5,10,5,6,8,0,5,11,5,2,0,2,5,12},	// 182
{2,11,5,2,5,3,6,10,5,10,3,5,12,12,12,12},	// 183

Annexe 2

{5,6,2,5,2,8,5,8,9,3,8,2,12,12,12,12},	// 184
{9,5,6,9,6,0,0,6,2,12,12,12,12,12,12},	// 185
{1,5,8,1,8,0,5,6,8,3,8,2,6,2,8,12},	// 186
{1,5,6,2,1,6,12,12,12,12,12,12,12,12,12},	// 187
{1,3,6,1,6,11,3,8,6,5,6,9,8,9,6,12},	// 188
{11,1,0,11,0,6,9,5,0,5,6,0,12,12,12,12},	// 189
{5,6,11,0,3,8,12,12,12,12,12,12,12,12,12},	// 190
{5,6,11,12,12,12,12,12,12,12,12,12,12,12,12},	// 191
{5,10,7,11,10,5,12,12,12,12,12,12,12,12,12},	// 192
{10,5,11,10,7,5,8,3,0,12,12,12,12,12,12,12},	// 193
{5,10,7,5,11,10,1,9,0,12,12,12,12,12,12,12},	// 194
{5,11,7,7,11,10,1,9,8,1,8,3,12,12,12,12},	// 195
{10,1,2,10,7,1,7,5,1,12,12,12,12,12,12,12},	// 196
{0,8,3,1,2,7,1,7,5,7,2,10,12,12,12,12},	// 197
{9,0,2,9,2,7,9,7,5,7,2,10,12,12,12,12},	// 198
{7,5,2,7,2,10,5,9,2,3,2,8,9,8,2,12},	// 199
{2,5,11,2,3,5,3,7,5,12,12,12,12,12,12,12},	// 200
{11,7,5,11,0,7,11,2,0,8,7,0,12,12,12,12},	// 201
{9,0,1,5,11,3,5,3,7,3,11,2,12,12,12,12},	// 202
{9,8,2,9,2,1,8,7,2,11,2,5,7,5,2,12},	// 203
{5,1,7,1,3,7,12,12,12,12,12,12,12,12,12,12},	// 204
{0,8,7,0,7,1,1,7,5,12,12,12,12,12,12,12},	// 205
{9,0,3,9,3,5,5,3,7,12,12,12,12,12,12,12},	// 206
{9,8,7,5,9,7,12,12,12,12,12,12,12,12,12,12},	// 207
{5,8,4,5,11,8,11,10,8,12,12,12,12,12,12,12},	// 208
{5,11,10,5,10,0,5,0,4,0,10,3,12,12,12,12},	// 209
{1,9,0,5,11,4,11,8,4,11,10,8,12,12,12,12},	// 210
{11,10,4,11,4,5,10,3,4,9,4,1,3,1,4,12},	// 211
{4,10,8,4,1,10,4,5,1,2,10,1,12,12,12,12},	// 212
{0,4,10,0,10,3,4,5,10,2,10,1,5,1,10,12},	// 213
{0,2,5,0,5,9,2,10,5,4,5,8,10,8,5,12},	// 214
{4,5,9,3,2,10,12,12,12,12,12,12,12,12,12,12},	// 215
{8,2,3,4,2,8,4,11,2,4,5,11,12,12,12,12},	// 216
{5,11,2,5,2,4,4,2,0,12,12,12,12,12,12,12},	// 217
{4,3,8,4,2,3,4,5,2,11,2,5,9,0,1,12},	// 218
{1,9,4,1,4,2,5,11,4,11,2,4,12,12,12,12},	// 219
{8,4,5,8,5,3,3,5,1,12,12,12,12,12,12,12},	// 220
{5,1,0,4,5,0,12,12,12,12,12,12,12,12,12,12},	// 221
{8,4,5,8,5,3,9,0,5,0,3,5,12,12,12,12},	// 222
{4,5,9,12,12,12,12,12,12,12,12,12,12,12,12,12},	// 223
{4,10,7,4,9,10,9,11,10,12,12,12,12,12,12,12},	// 224
{0,8,3,4,9,7,9,10,7,9,11,10,12,12,12,12},	// 225
{7,4,0,7,0,11,7,11,10,1,11,0,12,12,12,12},	// 226
{3,1,4,3,4,8,1,11,4,7,4,10,11,10,4,12},	// 227
{1,2,10,9,1,10,9,10,7,9,7,4,12,12,12,12},	// 228
{9,7,4,9,10,7,9,1,10,2,10,1,0,8,3,12},	// 229
{10,7,4,10,4,2,2,4,0,12,12,12,12,12,12,12},	// 230
{10,7,4,10,4,2,8,3,4,3,2,4,12,12,12,12},	// 231
{4,3,7,4,11,3,4,9,11,11,2,3,12,12,12,12},	// 232
{9,11,7,9,7,4,11,2,7,8,7,0,2,0,7,12},	// 233

Annexe 2

```

{3,7,11,3,11,2,7,4,11,1,11,0,4,0,11,12}, // 234
{4,8,7,11,2,1,12,12,12,12,12,12,12,12,12}, // 235
{4,9,1,4,1,7,7,1,3,12,12,12,12,12,12}, // 236
{0,8,7,0,7,1,4,9,7,9,1,7,12,12,12,12}, // 237
{4,0,3,7,4,3,12,12,12,12,12,12,12,12,12}, // 238
{4,8,7,12,12,12,12,12,12,12,12,12,12,12,12}, // 239
{9,11,8,8,11,10,12,12,12,12,12,12,12,12,12}, // 240
{3,0,9,3,9,10,10,9,11,12,12,12,12,12,12}, // 241
{0,1,11,0,11,8,8,11,10,12,12,12,12,12,12}, // 242
{3,1,11,10,3,11,12,12,12,12,12,12,12,12,12}, // 243
{1,2,10,1,10,9,9,10,8,12,12,12,12,12,12}, // 244
{3,0,9,3,9,10,1,2,9,2,10,9,12,12,12,12}, // 245
{10,8,0,2,10,0,12,12,12,12,12,12,12,12,12}, // 246
{10,3,2,12,12,12,12,12,12,12,12,12,12,12,12}, // 247
{2,3,8,2,8,11,11,8,9,12,12,12,12,12,12}, // 248
{9,11,2,0,9,2,12,12,12,12,12,12,12,12,12}, // 249
{2,3,8,2,8,11,0,1,8,1,11,8,12,12,12,12}, // 250
{1,11,2,12,12,12,12,12,12,12,12,12,12,12,12}, // 251
{8,9,1,3,8,1,12,12,12,12,12,12,12,12,12}, // 252
{9,1,0,12,12,12,12,12,12,12,12,12,12,12,12}, // 253
{8,0,3,12,12,12,12,12,12,12,12,12,12,12,12}, // 254
{12,12,12,12,12,12,12,12,12,12,12,12,12,12,12} // 255
};

int main()
{
    // Choix des paramètres
    int L=256; // Largeur de l'image
    int H=256; // Hauteur de l'image
    int P=256; // Profondeur de l'image
    double a=1.4; // Taille d'un voxel
    unsigned char SeuilImage=141; // Seuil à appliquer
    ifstream Entree("30.raw",ios_base::binary); // Fichier d'entrée
    if(!Entree) // vérification du fichier d'entrée
    { cout << "pas de fichier d'entree\n";return -1;}
    ofstream Sortie("Surface.txt");
    // Début du code
    double S=static_cast<double>(SeuilImage)-5e-2;
    unsigned char Index,NA;
    int i,j,k,l,Noeuds=0,Elements=0,C,jL;
    int jLpkHL,jLpkHLp1,kHL,P_1,jLpkp1HL,jLp1;
    int kH,kHp1,jpkH,jp1pkH,jpkp1H,jp1pkp1H,jp1Lp1,jp1L;
    int HP,LP,HL,HPpkLp1,HPpLP,HPpLPp1,HPpLPp1pjL,ipjL,jLpkp1HLp1;
    int jp1LpkHL,jp1LpkHLp1,jp1Lpkp1HL,jp1Lpkp1HLp1,kL,kLp1,kp1L,kp1Lp1;
    int L_1=L-1;
    int Lp1=L+1;
    P_1=P-1;
    int H_1=H-1;
    HL=H*L;HP=H*P;LP=L*P;
    HPpLP=HP+LP;HPpLPp1=HPpLP+1;

```

Annexe 2

```

int AreteAppelee[12];for(k=0;k<12;k++){ AreteAppelee[k]=0;}
int * NN = new int[(H+L)*P+HL+1];NN[0]=0;
int * NNDA = new int[(H+L)*P+HL+1];
unsigned char * Image = new unsigned char[P*H*L];
bool * AppelSommetSuivantZ = new bool[HL];
double NG1,NG2,I,x,y,z;
for (k=0;k<HP+LP+HL+1;k++)
{
    NNDA[k]=0;
}
// Ligne laissée pour l'en-tête
Sortie << "                \n";
// Lecture de l'image
int Decalage=-L,DecalageInit;
for (k=0;k<P;k++)
{
    kHL=k*HL;
    DecalageInit=HL+kHL;
    Entree.seekg(DecalageInit,std::ios_base::beg);
    for (j=0;j<H;j++)
    {
        jLpkHL=j*L+kHL;
        Entree.seekg(Decalage,std::ios_base::cur);
        for (i=0;i<L;i++)
        {
            Image[i+jLpkHL]=Entree.get();
        }
        Entree.seekg(Decalage,std::ios_base::cur);
    }
}

// Ecriture des noeuds suivant l'abscisse
for (k=0;k<P;k++)
{
    kH=k*H;
    kHp1=kH+1;
    kHL=k*HL;
    for (j=0;j<H;j++)
    {
        jL=j*L;
        jLpkHL=jL+kHL;
        jLpkHLp1=jLpkHL+1;
        for (i=0;i<L_1;i++)
        {
            if(Image[i+jLpkHL]<S!=Image[i+jLpkHLp1]<S)
            {
                Noeuds+=1;
                NG1=static_cast<double>(Image[i+jLpkHL]);
                NG2=static_cast<double>(Image[i+jLpkHLp1]);
                I=(S-NG1)/(NG2-NG1)*a;
            }
        }
    }
}

```


Annexe 2

```

        x=(i+0.5)*a;y=(j+0.5)*a;z=(k+0.5)*a;
        Sortie << x+I << " " << y << " " << z << endl;
    }
}
NN[j+kHp1]=Noeuds;
}
}
// Ecriture des noeuds suivant l'ordonnée
for (k=0;k<P;k++)
{
    HPpkLp1=HP+k*L+1;
    kHL=k*HL;
    for (i=0;i<L;i++)
    {
        for(j=0;j<H_1;j++)
        {
            if(Image[i+j*L+kHL]<S!=Image[i+(j+1)*L+kHL]<S)
            {
                Noeuds+=1;
                NG1=static_cast<double>(Image[i+j*L+kHL]);
                NG2=static_cast<double>(Image[i+(j+1)*L+kHL]);
                I=(S-NG1)/(NG2-NG1)*a;
                x=(i+0.5)*a;y=(j+0.5)*a;z=(k+0.5)*a;
                Sortie << x << " " << y+I << " " << z << endl;
            }
        }
        NN[HPpkLp1+i]=Noeuds;
    }
}
// Ecriture des noeuds suivant la cote
for(j=0;j<H;j++)
{
    jL=j*L;
    HPpLp1pjL=HPpLp1+jL;
    for (i=0;i<L;i++)
    {
        ipjL=i+jL;
        for(k=0;k<P_1;k++)
        {
            if(Image[ipjL+k*HL]<S!=Image[ipjL+(k+1)*HL]<S)
            {
                Noeuds+=1;
                NG1=static_cast<double>(Image[ipjL+k*HL]);
                NG2=static_cast<double>(Image[ipjL+(k+1)*HL]);
                I=(S-NG1)/(NG2-NG1)*a;
                x=(i+0.5)*a;y=(j+0.5)*a;z=(k+0.5)*a;
                Sortie << x << " " << y << " " << z+I << endl;
            }
        }
        NN[HPpLp1pjL+i]=Noeuds;
    }
}

```

```

    }
}
// Ecriture des éléments
NA=0;
for(k=0;k<P_1;k++)
{
    kH=k*H;
    kHp1=kH+1;
    kHL=k*HL;
    kL=k*L;
    kLp1=kL+1;
    kp1L=kL+L;
    kp1Lp1=kp1L+1;
    for(j=0;j<H;j++)
    {
        NNDA[j+kH]=0;
    }
    for(i=0;i<L;i++)
    {
        NNDA[HP+i+kL]=0;
    }

    for(j=0;j<H;j++)
    {
        jL=j*L;
        for(i=0;i<L;i++)
        {
            AppelSommetSuivantZ[i+jL]=0;
        }
    }

    for(j=0;j<H_1;j++)
    {
        jpkH=j+kH;
        jp1pkH=j+kHp1;
        jpkp1H=j+kH+H;
        jp1pkp1H=jp1pkH+H;

        jL=j*L;
        jLp1=jL+1;
        jp1L=jL+L;
        jp1Lp1=jLp1+L;

        jLpkHL=j*L+kHL;
        jLpkHLp1=jLpkHL+1;
        jLpkp1HL=jLpkHL+HL;
        jLpkp1HLp1=jLpkp1HL+1;
        jp1LpkHL=jLpkHL+L;
        jp1LpkHLp1=jLpkHLp1+L;
    }
}

```

Annexe 2

```

jp1Lpkp1HL=jLpkp1HL+L;
jp1Lpkp1HLp1=jLpkp1HLp1+L;

NNDA[jpkH]=0;
NNDA[jpkp1H]=0;
for(i=0;i<L_1;i++)
{
    Index=0;
    // Calcul de l'index
    if(Image[i+jLpkHL]<S)
        Index+=1;
    if(Image[i+jLpkHLp1]<S)
        Index+=2;
    if(Image[i+jp1LpkHLp1]<S)
        Index+=4;
    if(Image[i+jp1LpkHL]<S)
        Index+=8;
    if(Image[i+jLpkp1HL]<S)
        Index+=16;
    if(Image[i+jLpkp1HLp1]<S)
        Index+=32;
    if(Image[i+jp1Lpkp1HLp1]<S)
        Index+=64;
    if(Image[i+jp1Lpkp1HL]<S)
        Index+=128;
    if(TableTriangles[Index][0]!=12)
    {
        while(TableTriangles[Index][NA]!=12)
        {
            switch(TableTriangles[Index][NA])
            {
                case 0 : C=NN[jpkH]+1; break;
                case 1 : C=NN[HP+i+kLp1]+1; break;
                case 2 : C=NN[jp1pkH]+NNDA[jp1pkH]+1; break;
                case 3 : C=NN[HP+i+kL]+1; break;
                case 4 : C=NN[jpkp1H]+NNDA[jpkp1H]+1; break;
                case 5 : C=NN[HP+i+kp1Lp1]+NNDA[HP+i+kp1Lp1]+1; break;
                case 6 : C=NN[jp1pkp1H]+NNDA[jp1pkp1H]+1; break;
                case 7 : C=NN[HP+i+kp1L]+NNDA[HP+i+kp1L]+1; break;
                case 8 : C=NN[HPpLP+i+jL]+NNDA[HPpLP+i+jL]+1; break;
                case 9 : C=NN[HPpLPp1+i+jL]+NNDA[HPpLPp1+i+jL]+1; break;
                case 10 : C=NN[HPpLP+i+jp1L]+NNDA[HPpLP+i+jp1L]+1; break;
                case 11 : C=NN[HPpLP+i+jp1Lp1]+NNDA[HPpLP+i+jp1Lp1]+1;
break;
            }
            AreteAppelee[TableTriangles[Index][NA]]=1;
            NA+=1;
            if(NA%3==0)
            {
                Elements+=1;
            }
        }
    }
}

```

Annexe 2

```

        Sortie << C << " 0" << endl;
    }
    else
    {
        Sortie << C << " ";
    }
}
for(l=0;l<12;l++)
{
    if(AreteAppelee[l])
    {
        switch(l)
        {
            case 0 : NN[jpkH]+=1; break;
            case 1 : if(i==L-2)
                {
                    NN[HP+i+kL+1]+=1;
                }
                break;
            case 2 : NNDA[jp1pkH]+=1; break;
            case 3 : NN[HP+i+kL]+=1; break;
            case 4 : NNDA[jpkp1H]+=1; break;
            case 5 : if(i==L-2)
                {
                    NNDA[HP+i+kp1L+1]+=1;
                }
                break;
            case 6 : NNDA[jp1pkp1H]+=1; break;
            case 7 : NNDA[HP+i+kp1L]+=1; break;
            case 8 : AppelSommetSuisantZ[i+jL]=1; break;
            case 9 : if(i==L-2)
                {
                    AppelSommetSuisantZ[i+1+jL]=1;
                }
                break;
            case 10 : if(j==H-2)
                {
                    AppelSommetSuisantZ[i+jL+L]=1;
                }
                break;
            case 11 : if(j==H-2 && i==L-2)
                {
                    AppelSommetSuisantZ[i+1+jL+L]=1;
                }
                break;
        }
        AreteAppelee[l]=0;
    }
}
NA=0;

```

Annexe 2

```
        }
    }
}
for(j=0;j<H;j++)
{
    jL=j*L;
    for(i=0;i<L;i++)
    {
        if(AppelSommetSuivantZ[i+jL])
        {
            NNDA[HPpLP+i+jL]+=1;
        }
    }
}

}
Sortie.seekp(0,ios::beg);
Sortie << Noeuds << " 3 " << Elements << " 4";
return 0;
}
```

- Références bibliographiques -

- [Ada 99] Adalsteinsson D., Sethian J. A. The Fast Construction of Extension Velocities in Level Set Methods. *Journal of Computational Physics*, 1999, vol. 148, p. 2-22.
- [Alr 02] Al-Rawahi N., Tryggvason G. Numerical simulation of dendritic solidification with convection: Two-dimensional geometry. *Journal of Computational Physics*, 2002, vol. 180, p. 471-496.
- [Alr 04] Al-Rawahi N., Tryggvason G. Numerical simulation of dendritic solidification with convection: Three-dimensional flow. *Journal of Computational Physics*, 2004, vol. 194, p. 677-696.
- [Ami 10] Amira® 5 - Amira User's Guide, 2010, 476 p.
- [Arn 84] Arnold D., Brezzi F., Fortin M. A stable finite element for the Stokes equations, *Calcolo*, 1984, vol. 23, No 4, p. 337-344.
- [Bab 73] Babuška I. The Finite Element Method with Penalty, *Mathematics of Computation*, 1973, vol. 27, No 122, p. 221-228.
- [Bal 10] Balay S., Buschelman K., Eijkhout V., Gropp W., Kaushik D., Knepley M., Curfman McInnes L., Smith B., Zhang H. PETSc Users Manual – Revision 3.1, Mathematics and Computer Science Division, Argonne National Laboratory, 2010, 189 p.
- [Bar 98] Barth T. J., Sethian J. A. Numerical Schemes for the Hamilton-Jacobi and Level Set Equations on Triangulated Domains. *Journal of Computational Physics*, 1998, vol. 145, p. 1-40.
- [Bar 06a] Bargteil A. W. Tolga G. G. O'Brien J. F., Strain J. A. A semi-Lagrangian contouring for fluid simulation, *ACM Transactions on Graphics (TOG)*, 2006, vol 25, No 1, p. 19-38.
- [Bar 06b] Bargteil A. *Tracking and Texturing Liquid Surfaces*. Ph.D. Thesis in Computer Science. University of California, Berkeley, 2006, 89 p.
- [Bar 06c] Baruchel J., Buffiere J.-Y., Cloetens P., Di Michiel M., Ferrie E., Ludwig W., Maire E., Salvo L. Advances in synchrotron radiation microtomography, *Scripta Materiala*, 2006, vol. 55, p 41-46.
- [Bas 06] Basset O. *Simulation numérique d'écoulements multi fluides sur grille de calcul*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2006, 200 p.
- [Bel 89] Bell J. B., Collela P. A Second-Order Projection Method for the Incompressible Navier-Stokes Equations. *Journal of Computational Physics*, 1989, vol. 85, p. 257-283.
- [Bel 09] Bellet M., Cerri O., Bobadilla M., Chastel Y. Modeling Hot Tearing during Solidification of Steels: Assessment and Improvement of Macroscopic Criteria through the Analysis of Two Experimental Tests, *Metallurgical and Materials Transactions A*, 2009, vol. 40A, p. 2705-2717.

Références bibliographiques

- [Ber 05] Bernard D. 3D Quantification of Pore Scale Geometrical Changes Using Synchrotron Computed Microtomography. *Oil and Gas Science and Technology-Rev. IFP*, 2005, vol. 60, No. 5, p. 747-762.
- [Ber 09] Bernacki M., Resk H., Coupez T., Logé R.E. Finite element model of primary recrystallization in polycrystalline aggregates using a level set framework, *Modelling and Simulation in Materials Science and Engineering*, 2009, vol. 17 064006, 22 p.
- [Bre 74] Brezzi F. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, vol. 8, No 2, p 129-151.
- [Bro 82] Brooks A. N., Hughes T. J. R. Steamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations. *Computer Methods in Applied Mechanics and Engineering*, vol. 32, p. 199-259.
- [Bru 04] Bruchon J. *Etude de la formation d'une structure de mousse par simulation directe de l'expansion de bulles dans une matrice liquide polymère*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2004, 191 p.
- [Bru 07] Bruchon J. *Les lois de comportement avec CIMLIB*. Centre de Mise en Forme des Matériaux, MINES ParisTech, 2007, 49 p.
- [Bru 09] Bruchon J., Digonnet H., Coupez T. Using a signed distance function for the simulation of metal forming processes: Formulation of the contact condition and mesh adaptation. From a Lagrangian approach to an Eulerian approach. *International Journal for Numerical Methods in Engineering*, 2009, vol. 78, p. 980-1008.
- [Bui 08] Bui T. T. C., Frey P., Maury B. Méthode du second membre modifié pour la gestion de rapports de viscosité importants dans le problème de Stokes bifluide. *Comptes Rendus Mécaniques*, 2008, vol. 336, p. 524-529.
- [Bui 10] Bui T. T. C., Dapogny C., Frey P. An accurate anisotropic adaptation method for solving the level set advection equation. *International Journal for Numerical Methods in Fluids*, 2010, accepté.
- [Bui 11] Bui T. T. C., Frey P., Maury B. A coupling strategy based on anisotropic mesh adaptation for solving two-fluid flows. *International Journal for Numerical Methods in Fluids*, 2011, vol. 66, p. 1226-1247.
- [Bur 05] Burkard J., *Finite Element Treatment of the Navier Stokes Equation*. School of Computational Science, Florida State University, 2005, 68 p.
- [Cai 01] Caiden R., Fedkiw R., Anderson C. A Numerical Method for Two Phase Flow Consisting of Separate Compressible and Incompressible Regions. *Journal of Computational Physics*, 2001, vol. 166, p. 1-27.
- [Car 07] Carlson K. D., Lin Z., Beckermann C. Modeling the Effect of Finite-Rate Hydrogen Diffusion on Porosity Formation in Aluminum Alloys. *Metallurgical and Materials Transactions B*, 2007, vol 38B, p. 541-555.
- [Car 12] Carozzani T., Digonnet H., Gandin C.-A. 3D CAFE modeling of grain structures: Application to primary dendritic and secondary eutectic solidification. *Modelling and Simulation in Materials Science and Engineering*, 2012, vol. 015010, 19 p.

Références bibliographiques

- [Cer 07] Cerri O. *Rupture à chaud dans les aciers au cours de leur solidification. Caractérisation expérimentale et modélisation thermomécanique*, Thèse de Doctorat spécialité Science et Génie des Matériaux, Sophia-Antipolis : MINES ParisTech, 2007, 204 p.
- [Cha 94] Chan T. F., van der Vorst H. A. *Approximate and incomplete factorizations*, chapter in *Parallel Numerical Algorithms*, edited by D. Keyes, A. Semah, V. Venkatakrishnan, ICASE/LaRC Interdisciplinary Series in Science and Engineering, Kluwer, 1994, p. 167-202.
- [Che 98] Chenot J.-L. Fourment L., Coupez T., Ducloux R., Wey E. *Forge3® - a general tool for practical optimization of forging sequence of complex three-dimensional parts in industry*. Proceedings of the International Conference on Forging and Related Technology, 1998, p. 113-122.
- [Cos 04] Costes F. *Modélisation thermomécanique tridimensionnelle par éléments finis de la coulée continue d'aciers*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2004, 160 p.
- [Cou 97] Coupez T., Marie S. From a direct solver to a parallel iterative solver in 3D forming simulation. *International Journal of Supercomputer and Applications*, 1997, vol. 11, p. 205-211.
- [Cou 00] Coupez T. Génération et adaptation de maillage par optimisation locale, *La Revue Européenne des éléments finis*, 2000, vol. 9, p. 403-423.
- [Cou 06] Coupez T. *Réinitialisation convective et locale des fonctions Level Set pour le mouvement de surfaces et d'interfaces*. Journées Activités Universitaires de Mécanique, La Rochelle, 2006.
- [Cou 07] Coupez T. *Convection of local Level Set function for moving surfaces and interfaces in forming flow*. Proceedings of the 9th International Conference on Numerical Methods in Industrial Forming Processes, Porto, 2007, vol. 908, p 61-66.
- [Cou 11] Coupez T. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. *Journal of Computational Physics*, 2011, vol. 230, p. 2391-2405.
- [Cou 67] Courant R., Friedrichs R., Lewy H. On the Partial Difference Equations of Mathematical Physics. *IBM Journal of Research and Development*, 1967 vol. 11, No 2, p. 215-234.
- [Cun 08a] Cuntz N., Strzodka R., Kolb A. *Real-Time Particle Level Sets with Application to Flow Visualization*. Computer Graphics and Multimedia Group, University of Siegen, Germany, 2008, 12 p.
- [Cun 08b] Cuntz N. Kolb A., Strzodka R., Weiskopf D. Particle Level Set Advection for the Interactive Visualization of Unsteady 3D Flow. *Computer Graphics Forum*, 2008, Vol. 27, No 3, p. 719-726.
- [Dah 96] Dahle A. K., Arnberg L. Overview: the rheological properties of solidifying aluminium foundry alloys. *Journal of the Minerals, Metals and Materials Society*, 1996, Vol. 48, No 3, p. 34-47.
- [Dan 09] Dantzig J.A., Rappaz M. *Solidification*, EPFL Press, 2009.

Références bibliographiques

- [Dem 08] Demay Y. *Eléments Finis. Eléments Finis Mixtes*. Laboratoire Jean Alexandre Dieudonné, Université de Nice, 2008, 30 p.
- [Dha 84] G. Dhatt, Touzot G. *Une présentation de la méthode des éléments finis*. Maloine S.A. Editeur Paris, 1984, 2^e édition.
- [Dig 01] Digonnet H. *Repartitionnement dynamique, mailleur parallèle et leurs applications à la simulation numérique en mise en forme des matériaux*. Thèse de Doctorat spécialité Science et Génie des Matériaux, Sophia-Antipolis. MINES ParisTech, 2001, 190 p.
- [Dig 07] Digonnet H., Silva L., Coupez T. *Cimlib: A Fully parallel Application for Numerical Simulations Based On Components Assembly*. Proceedings of Numerical Methods in Industrial Forming Processes 2007, Porto, 2007, vol. 208, p. 269-274.
- [Dub 11] Dubey P., Hanrahan P., Fedkiw R., Lentine M., Schroeder C. *PhysBAM: Physically Based Simulation*. SIGGRAPH courses, Stanford University, 2011, 22 p.
- [Duc 08] Ducrot V., Frey P. *Anisotropic level set adaptation for accurate interface capturing*. Proceedings of the 17th International Meshing Roundtable, Sandia National Labs, Pittsburgh, PA, 2008, p 159-176.
- [Duf 86] Duff I. S, Erisman A. M., Reid J. K. *Direct Methods for Sparse Matrices*. Clarendon Press, 1986.
- [Duf 64] Dufour H. M., Résolution des systèmes linéaires par la méthode des résidus conjugués, *Bulletin Géodésique (1946 - 1975)*, 1964, vol. 71, No 1, p. 65-87.
- [Ebe 08] Eberly D. *Distance from a Point to an Ellipse in 2D*. Geometric Tools, LLC, 2008, 15 p.
- [Eld 48] Elder F. R., Langmuir R. V., Pollock H. C. Radiation from Electrons Accelerated in a Synchrotron. *Physical Review*, vol. 74, No 1, p. 52-56.
- [Elk 10] El Khaoulani R. *Prédiction fiable de l'endommagement ductile par la méthode des éléments finis mixtes : endommagement non local et adaptation de maillage*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2010, 146 p.
- [Elm 11] Elmi M. A coupled Directed Marker and Level Set Method for Interface Capturing. *Journal of Computational Physics*, 2011, soumis.
- [Elv 92] Elvins T. T. A survey of algorithms for volume visualization. *Computer Graphics*, 1992, vol. 26, No 3, p. 194-201.
- [Enf 07] Enficiaud R. *Algorithmes multidimensionnels et multispectraux en Morphologie Mathématique : Approche par méta-programmation*. Thèse de Doctorat spécialité Morphologie Mathématique, Fontainebleau. MINES ParisTech, 2007, 294 p.
- [Enr 02a] Enright D. P. *Use of the Particle Level Set Method for enhanced resolution of free surface flows*. Ph.D. Thesis in Scientific Computing and Computational Mathematics. Stanford university, 2002, 110 p.
- [Enr 02b] Enright D., Marschner S., Fedkiw R. *Animation and Rendering of Complex Water Surfaces*, SIGGRAPH, 2002, p. 736-744.

Références bibliographiques

- [Enr 02c] Enright D., Fedkiw R., Ferziger J. Mitchell I., A Hybrid Particle Level Set Method for Improved Interface Capturing, *Journal of Computational. Physics*, 2002, vol. 183, p. 83-116.
- [Enr 05] Enright D., Losasso F., Fedkiw R. A Fast and Accurate Semi-Lagrangian Particle Level Set Method. *Computers and Structures*, 2005, vol. 83, p. 479-490.
- [Esk 04] Eskin D.G., Suyitno, Katgerman L. Mechanical properties in the semi-solid state and hot tearing of aluminium alloys. *Progress in Materials Science*, 2004, vol. 49, p 629-711.
- [Esk 07] Eskin D.G., Katgerman L. A quest for a New Hot Tearing Criterion. *Metallurgical and Materials Transactions A*, 2007, vol. 38A, p. 1511, 1519.
- [Fab 06] Fabrègue A., Deschamps M., Suéry M., Poole W.J. Rheological Behavior of Al-Mg-Si-Cu Alloy in the Mushy State Obtained by Partial Remelting and Partial Solidification at High Cooling Rate, *Metallurgical and Materials Transactions A*, 2006, vol. 37A, p. 1459-1467.
- [Fad 81] Faddeev D. K., Faddeeva V. N. Computational methods of linear algebra, *Journal of Mathematical Sciences*, 1981, vol. 15, No 5, p. 531-650.
- [Fel 10] Felberbaum M. *Porosity in Aluminium Alloys: Visualization, Characterization, and Modeling*. Ph.D. Thesis. EPFL, 2010, 191 p.
- [Fos 01] Foster N. Fedkiw R. *Practical Animation of Liquids*. SIGGRAPH, 2001, p. 15-22.
- [Fra 11] François G. *Éléments Finis Stabilisés pour le remplissage en fonderie à haut Reynolds*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2011, 176 p.
- [Fre 98] Frey P. J., Borouchaki H., George P.-L. 3D Delaunay mesh generation coupled with an advancing-front approach. *Computer Methods in Applied Mechanics and Engineering*, 1998, vol. 157, p. 115-131.
- [Fre 99] Frey P.J., George P.L. *Maillages – Applications aux éléments finis*. Hermes Science, 1999.
- [Ful 09] Fuloria D., Lee P. D. An X-ray microtomographic and finite element modelling approach for the prediction of semi-solid deformation behaviour in Al-Cu alloys. *Acta Materiala*, 2009, vol. 57, p. 5554-5562.
- [Gar 99] Garland M. *Quadric-Based Polygonal Surface Simplification*. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, 1999, 210 p.
- [Gau 08] Gaudlitz D., Adams N. A. On improving mass-conservation properties of the hybrid particle-level-set method. *Computer & Fluids*, 2008, vol. 37, p. 1320-1331.
- [Gel 94] Van Gelder A., Wilhelms J. Topological Considerations in Isosurface Generation. *ACM Transactions on Graphics*, 1994, vol. 13, No 4, p. 337-375.
- [Gib 03] Gibou F., Fedkiw R., Caflisch, R., Osher S. A level set approach for the numerical simulation of dendritic growth. *Journal of Scientific Computing*, 2003, vol. 19, p. 183-199.
- [Gil 84] Gilormini P., Montheillet F. Champs de vitesses et de contraintes autour d'une inclusion cylindrique en traction ou compression plane (calcul analytique). *Journal de Mécanique Théorique et Appliquée*, 1984, vol. 3, No 4, p. 563-576.

Références bibliographiques

- [Gil 86] Gilormini P., Montheillet F. Deformation of an inclusion in a viscous matrix and induced stress concentrations. *Journal of the Mechanics and Physics of Solids*, 1986, vol. 34, No 2, p. 97-123.
- [Gil 87] Gilormini P., Germain Y. A finite element analysis of the inclusion problem for power law viscous materials. *International Journal of Solids and Structures*, 1987, vol. 23, No 3, p. 413-437.
- [Gir 10] Giraud E., Suéry M., Coret M. Mechanical Behavior of AA6061 Aluminium in the Semisolid State Obtained by Partial Melting and Partial Solidification. *Metallurgical and Materials Transactions A*, 2010, vol. 41A, p. 2257-2268.
- [Gra 04] Grasso P.-D. *Coalescence and Mechanical behaviour of semi-solid aluminium alloys in relation to hot tearing*. Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne, 2004, 184 p.
- [Gri 03] Grinberg I. *L'Aluminium : Un si léger métal*. Editions Gallimard, 2003, 127 p.
- [Gru 04] Gruau C. *Génération de métriques pour adaptation anisotrope de maillages, applications à la mise en forme des matériaux*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2004, 230 p.
- [Gru 05] Gruau C., Coupez T. 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Computer Methods in Applied Mechanics and Engineering*, 2005, vol. 157, p. 115-131.
- [Hac 09] Hachem E. *Stabilized finite element method for heat transfer and turbulent flows inside industrial furnaces*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2009, 326 p.
- [Hac 10] Hachem E., Kloczko T., Digonnet H., Coupez T. Stabilized finite element solution to handle complex heat and fluid flows in industrial furnaces using the immersed volume method. *International Journal for Numerical Methods in Fluids*, 2010, doi : 10.1002/flid.2498
- [Ham 08a] Hamide M. *Modélisation numérique du soudage à l'arc des aciers*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2008, 285 p.
- [Ham 08b] Hamide M., Massoni E., Bellet M. Adaptive mesh technique for thermal-metallurgical numerical simulation of arc welding processes. *International Journal for Numerical Methods in Engineering*, 2008, vol. 73, p. 624-641.
- [Har 09] Hardy L. *ESRF : Aspects du fonctionnement des accélérateurs au jour le jour*. ESRF, 2009, 105 p.
- [Hec 99] Heckbert P. S., Garland M. Optimal triangulation and quadric-based surface simplification. *Computational Geometry*, 1999, vol. 14, p. 49-65.
- [Heo 10] Heo N., Ko H.-S. Detail-preserving fully-Eulerian interface tracking framework. *ACM Transactions on Graphics (TOG)*, 2010, vol. 29, No 6.
- [Her 03] M. Herrmann. *A Domain Decomposition Parallelization of the Fast Marching Method*. CTR Annual Research Briefs, 2003, p. 213-225.

Références bibliographiques

- [Hes 52] Hestenes M. R., Stiefel E. Methods of Conjugate Gradients for Solving Linear Systems, *Journal of Research of the National Bureau of Standards*, 1952, Vol. 49, No. 6, p. 409-436.
- [Hie 05] Hieber S. E., Koumoutsakos P. A Lagrangian particle level set method. *Journal of Computational Physics*, 2005, vol. 210, p. 342-367.
- [Hir 81] Hirt C. W., Nichols B. D. Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. *Journal of Computational Physics*, 1981, vol. 39, p. 201-225.
- [Hit 11] Hitti K. *Direct numerical simulation of complex Representative Volume Elements (RVEs): Generation, Resolution and Homogenization*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2011, 191 p.
- [Huk 11] Hukic D. *Reconstruction exacte et statistique de microstructures numériques à partir d'images expérimentales dans un formalisme level-set*. Rapport de stage, MINES ParisTech, 2011, 29 p.
- [Hys 05] Hysing S.-R., Turek S. *The eikonal equation: numerical efficiency vs. algorithmic complexity on quadrilateral grids*. Proceedings of Algoritmy, 2005, p. 22-31.
- [Jao 98] Jaouen O. *Modélisation tridimensionnelle par éléments finis pour l'analyse thermo-mécanique du refroidissement des pièces coulées*. Thèse de Doctorat spécialité Sciences et Génie des Matériaux, Sophia-Antipolis. MINES ParisTech, 1998, 187 p.
- [Jim 04] Jimack P. K. Adaptive Algorithms for Free-Surface Flows Problems. Fourth International Conference on Engineering Computational Technology, Lisbon, 2004.
- [Jon 06] Jones M. W., Baerentzen J. A., Sramek M. *3D Distance Fields: A Survey of Techniques and Applications*, IEEE Transactions on Visualization and Computer Graphics, vol. 12, No 4, pp. 581-599.
- [Jou 94] Joubert W. On the convergence behaviour of the restarted gmres algorithm for solving nonsymmetric linear systems. *Numerical Linear Algebra with Applications*, 1994, vol. 1, No 5, p. 427-447.
- [Jur 96] Juric D., Tryggvason G. A Front-Tracking Method for Dendritic Solidification. *Journal of Computational Physics*, 1996, vol. 123, p. 127-148.
- [Kar 98] Karma A., Rappel W.-J. Quantitative phase-field modeling of dendritic growth in two and three dimensions, *Physical Review E*, 1998, vol. 57, no 4, p. 4323-4349.
- [Kim 00] Kim Y.T, Goldendfeld N., Dantzig J.A. Computation of dendritic microstructures using a level set method, *Physical Review E*, 2000, vol. 62, no 2, p. 2471-2474.
- [Kur 98] Kurz W., Fisher D.-J. *Fundamentals of solidification*, Trans Tech Publications, 1998.
- [Lee 01] Lee P. D., Chirazi A., See D. Modeling microporosity in aluminum-silicon alloys: a review. *Journal of Light Metals*, 2001, vol. 1, p. 15-30.
- [Lev 96] Leveque R. J. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis*, 1996, vol. 33, No 2, p. 627-665.

Références bibliographiques

- [Lew 03] Lewiner T., Lopes H., Wilson Vieira A., Tavares G. Efficient implementation of Marching Cubes' cases with topological guarantees. *Journal of Graphic Tools*, 2003, vol. 8, p. 1-15.
- [Lim 07] Limodin N., Salvo L., Suery M., DiMichiel M. In situ investigation by X-ray tomography of the overall and local microstructural changes occurring during partial remelting of an Al-15.8 wt.% Cu alloy. *Acta Materiala*, 2007, vol. 55, p. 3177-3191.
- [Lim 09] Limodin N., Salvo L., Boller E., Suéry M., Felberbaum M., Gaillière S., Madi K. In situ and real-time 3-D microtomography investigation of dendritic solidification in an Al-10 wt.% Cu alloy. *Acta Materiala*, 2009, vol. 57, p. 2300-2310.
- [Lor 87] Lorensen W.-E., Cline H.-E. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, 1987, vol. 21, p. 163-169.
- [Los 04] Losasso F., Gibou F., Fedkiw R. *Simulating Water and Smoke with an Octree Data Structure*, SIGGRAPH 2004, ACM TOG 23, p. 457-462 (2004).
- [Los 07] Losasso F. *Algorithms for increasing the efficiency and fidelity of fluid simulations*. Ph.D. Thesis. Stanford university, 2007, 118 p.
- [Lud 04] Ludwig O., *Etude expérimentale et modélisation du comportement rhéologique d'alliages d'Al-Cu en cours de solidification dans le cadre de la fissuration à chaud en coulée continue*. Thèse de Doctorat spécialité Science et Génie des Matériaux, Institut National Polytechnique de Grenoble, 2004, 209 p.
- [Lud 05] Ludwig O., Drezet J. M., Martin C. L., Suéry M. Rheological behavior of Al-Cu alloys during solidification: constitutive modeling, experimental identification, and numerical study. *Metallurgical and Materials Transactions A*, 2005, vol. 36, p. 1525-1535.
- [Lv 10] Lv X., Zou Q., Zhao Y., Reeve D. A novel coupled level set and volume of fluid method for sharp interface capturing on 3D tetrahedral grids. *Journal of Computational Physics*, 2010, vol. 229, p. 2573-2604.
- [Mad 06] Madi K. *Influence de la morphologie tridimensionnelle des phases sur le comportement mécanique de réfractaires électrofondus*. Thèse de Doctorat spécialité Science et Génie des Matériaux, Evry : MINES ParisTech, 2006.
- [Mad 07] Madi K., Forest S., Boussuge M., Gaillière S., Lataste E., Buffière J.-Y., Bernard D., Jeulin D. Finite element simulations of the deformation of fused-cast refractories based on X-ray computed tomography. *Computational Materials Science*, 2007, vol. 39, p. 224-229
- [Mai 03] Maire E., Fazekas A., Salvo L., Dendievel R., Youssef S., Cloetens P., Letang J.-M. X-ray tomography applied to the characterization of cellular materials. Related finite element modeling problems, *Composites Science and Technology*, 2003, vol. 63, p. 2431-2443.
- [Mar 00] Margaritondo G. Synchrotron Light in a Nutshell. *Surface Review and Letters*, 2000, vol. 7, No 4, p. 379-387.
- [Mar 97] Marie S. *Un modèle de parallélisation S.P.M.D pour la simulation numérique de procédés de mise en forme des matériaux*. Thèse de Doctorat spécialité Sciences et Génie des Matériaux, Sophia-Antipolis. MINES ParisTech, 1997, 199 p.
- [Mar 02] Martin C. L., Braccini M., Suéry M. Rheological behavior of the mushy zone at small strains. *Materials Science and Engineering A*, 2002, vol. 325, p. 292-301.

Références bibliographiques

- [Mat 07] Mathier V. *Two-phase modelling of hot tearing in aluminium alloys using a semi-coupled method*. Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne, 2007, 167 p.
- [Mat 00] Matsumoto, M., Nishimura, T.: Dynamic creation of pseudorandom number generators. In: Monte-Carlo and Quasi-Monte Carlo Methods 1998, pp. 56–69. Springer, Heidelberg (2000).
- [Mck 08] McKee S., Tomé M. F., Ferreira V. G., Cuminato J. A., Castelo A., Sousa F. S., Mangiavacchi N. The MAC method. *Computer & Fluids*, 2008, vol. 37, p. 907-930.
- [Mei 08] Mei X., Decaudin P., Hu B.-G., Zhang X. Real-Time Marker Level Set on GPU. In *International Conference on Cyberworlds, CW '08*. Hangzhou, Chine, septembre 2008. p. 209-216. IEEE, 2008.
- [Men 07] Menard T. *Développement d'une méthode Level Set pour le suivi d'interface*. Thèse de Doctorat Spécialité Energétique. Université de Rouen, 2007, 203 p.
- [Mes 08] Mesri Y., Zerguine W., Digonnet H., Silva L., Coupez T. *Dynamic Parallel Adaption for Three Dimensional Unstructured Meshes: Application to Interface Tracking*. Proceedings of the 17th International Meshing Roundtable, 2008, p. 195-212.
- [Mih 06] Mihalef V., Unlusu B., Metaxas D., Sussman M., Hussaini M. Y. *Physics-Based Boiling Simulation*. Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, 2006.
- [Mih 07] Mihalef V. *The Marker Level Set Methods: applications to simulation of liquids*. Ph.D. Thesis in Computer and Information Science. State University of New Jersey, 2007, 80 p.
- [Mih 08] Mihalef V., Kadioglu S., Sussman M., Metaxas D., Hurmusiadis V. Interaction of two-phase flow with animated models. *Graphical Models*, 2008, vol. 70, p. 33-42.
- [Mih 09] Mihalef V., Metaxas D., Sussman M. Simulation of two-phase flow with sub-scale droplet and bubble effects. *Computer Graphics Forum*, 2009, vol. 28, No 2, p. 229-338.
- [Mil 11] Milesi M., Bouchard P.-O., Cornuault P. H., Chastel Y., Bernacki M., Hachem E. Advanced numerical method for generation of three-dimensional particles and its application in microstructure-based simulation of fatigue behaviour. *Computational Materials Science*, 2011, doi :10.1016/j.commatsci.2011.04.039
- [Mit 08] Mitchell I. M., The Flexible, Extensible and Efficient Toolbox of Level Set Methods, *Journal of Scientific Computing*, 2008, vol. 35, No 2-3, p. 300-329.
- [Mok 06] Mokhberi E., Faloutsos P. *A Particle Level Set Library*. University of California, Los Angeles, 2006, 12 p.
- [Mot 02] Moto Mpong S. *Etude rhéologique et simulation numérique de l'injection d'un alliage d'aluminium à l'état semi-solide*. Thèse de Doctorat spécialité Science et Génie des Matériaux, Sophia-Antipolis. MINES ParisTech, 2002, 149 p.
- [Mpi 03] *MPI-2: Extensions to the Message- Passing Interface*, 2003, 370 p.
- [Mzo 09] Mzoughi M. A. *Modélisation de la microstructure dendritique basée sur des estimateurs d'erreur*. Rapport de stage, MINES ParisTech, 2009, 42 p.

Références bibliographiques

- [New 06] Newman T. S., Yi H. A survey of the marching cubes algorithm. *Computer & Graphics*, 2006, vol. 30, p. 854-879.
- [Nic 10] Nichita B. A. *An Improved CFD Tool to Simulate Adiabatic and Diabatic Two-Phase Flows*. Thèse de Doctorat. Ecole Polytechnique Fédérale de Lausanne, 2010, 181 p.
- [Nic 97] Nicod J.-M. *Extraction de surface en imagerie médicale : approches parallèles*. Thèse de Doctorat spécialité Informatique, Lyon : Ecole Normale Supérieure, 1997.
- [Nie 91] Nielson G. M., Hamann B. *The asymptotic decider : Resolving the ambiguity in Marching Cubes*. IEEE Proceedings on Visualization, San Diego CA, 1991, No 91CH3046-0, p 83-91.
- [Nie 02] Nielson G. M., Huang A., Sylvester S. *Approximating Normals for Marching Cubes applied to Locally Supported Isosurfaces*. IEEE Proceedings on Visualisation, Boston, MA, 2002, No 0-7803-7498-3, p. 459-466.
- [Odi 08] Odin G. *Modélisation numérique de l'os mandibulaire appliquée à l'implantologie dentaire maxillo-faciale*. Thèse de Doctorat spécialité Sciences et Génie des Matériaux, Sophia-Antipolis. MINES ParisTech, 2008.
- [Osh 88] Osher S., Sethian J. Front Propagating with Curvature Dependant Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 1988, vol. 79, p. 12-49.
- [Osh 01] Osher S., Fedkiw R. Level Set Methods: An Overview and Some Recent Results. *Journal of Computational Physics*, 2001, vol. 169, p. 463-502.
- [Osh 03] Osher S., Fedkiw R. *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New York, 2003.
- [Pac 05] Paccini A. *Développement d'un modèle éléments finis 3D appliqué à la simulation d'opérations chirurgicales des tissus mous*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2005.
- [Pen 99] Peng D., Merriman B., Osher S., Zhao H., Kang M. A PDE-Based Fast Local Level Set Method, *Journal of Computational Physics*, 1999, vol. 155, p. 410-438.
- [Per 00] Perchat E. *MINI-Elément et factorisations incomplètes pour la parallélisation d'un solveur de Stokes 2D. Application au forgeage*. Thèse de Doctorat spécialité Sciences et Génie des Matériaux, Sophia-Antipolis. MINES ParisTech, 2000, 154 p.
- [Pha 12] Pham V. T. *Mixed Velocity-Displacement Formulation for Modeling of Complex Behavior of Polymer*. Thèse de Doctorat Spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2012, 205 p.
- [Phi 08a] Phillion A.B, Lee P.D, Maire E., Cockcroft S.L. Quantitative Assessment of Deformation-Induced Damage in a Semisolid Aluminum Alloy via X-ray Microtomography. *Metallurgical and Materials Transactions A*, 2008, vol 39, p. 2459-2469.
- [Phi 08b] Phillion A. B., Cockcroft S. L., Lee P. D. A three-phase simulation of the effect of microstructural features on semi-solid tensile deformation. *Acta Materiala*, 2008, vol. 56, p. 4328-4338.
- [Phi 09a] Phillion A. B., Cockcroft S. L., Lee P. D. Predicting the constitutive behavior of

Références bibliographiques

semi-solids via a direct finite element simulation: application to AA5182. *Modelling and Simulation in Materials Science Engineering*, 2009, vol. 17, 15 p.

[Phi 09b] Phillion A. B., Desbiolles J. L., Rappaz M. *A 3D Granular Model of Equiaxed-Granular Solidification*. Modeling of Casting, Welding, and Advanced Solidification Processes—Xii, Vancouver, Canada, S. Cockcroft and D. Maijer, eds., TMS, Warrendale, PA, 2009, pp. 353–60.

[Pic 98] Pichelin E., Coupez T. Finite element solution of the 3D mold filling problem for viscous incompressible fluid. *Computer Methods in Applied Mechanics and Engineering*, 1998, vol. 163, p. 359-371.

[Pie 96] Piegl L., Tiller W. *The NURBS Book*. Springer, 1996.

[Pig 05] Pighin F., Patel S., Cohen J., Chu A. *Oriented Particle Level Set for Fluid Simulation*. Eurographics/ACM SIGGRAPH Symposium on Computer Animation, 2005.

[Pol 83] Pollock H. C. The discovery of synchrotron radiation. *American Journal of Physics*, 1983, vol. 51, No 3, p. 278-280.

[Pur 77] Purcell E.M. Life at Low Reynolds Number. *American Journal of Physics*, 1977, vol. 45, p. 3-11.

[Qua 00] Quarteroni A., Sacco R., Saleri F. *Méthodes numériques pour le calcul scientifique*, Springer-Verlag, 2000.

[Raj 03] Rajon D.-A., Bolch W.-E. Marching cube algorithm: review and trilinear interpolation adaptation for image-based dosimetric models. *Computerized Medical Imaging and Graphics*, 2003, vol. 27, p. 411-435.

[Ram 04] Ramirez J.C., Beckermann C., Karma A., Diepers H.J. Phase-field modelling of binary alloy solidification with coupled heat and solute diffusion. *Physical Review E*, 2004, vol. 69, p. 051607-1-16.

[Rat 71] Rathjen K.A., Jiji L.M. Heat Conduction with melting or freezing In a corner. *Journal of heat Transfer*, 1971, ASME 93, p. 101-109.

[Reb 05] Reboul M. *Corrosion des alliages d'aluminium*. Techniques de l'ingénieur, 2005, p. COR 325 1-19.

[Res 10] Resk H. *Finite element modelling of grain-scale heterogeneities in polycrystalline aggregates*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2010, 239 p.

[Ric 10] Richard M. *Intégration d'une méthode de calcul non linéaire Petsc dans Cimlib*. Centre de Mise en Forme des Matériaux, MINES ParisTech, 2010, 17 p.

[Rid 98] Rider W. J., Kothe D. B. Reconstructing Volume Tracking. *Journal of Computational Physics*, 1998, vol. 141, p. 1112-152.

[Ruc 07] Rucalvaba D., Mathiesen R.H., Eskin D.G., Arnberg L., Katgerman L. In situ observations of dendritic fragmentation due to local solute-enrichment during directional solidification of an aluminium alloy. *Acta Materiala*, 2007, vol. 55, p. 4287-4292.

Références bibliographiques

- [Saa 86] Saad Y. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 1986, vol. 7, No 3, p. 856-869.
- [Saa 03] Saad Y. *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 2003.
- [Sal 03] Salvo L., Cloetens P., Maire E., Zabler S., Blandin J.J, Buffière J.Y., Ludwig W., Boller E., Bellet D., Josserond C. X-ray micro-tomography an attractive characterisation technique in materials science. *Nuclear Instruments and Methods in Physics Research B*, 2003, vol. 200, p. 273-286.
- [Sal 10] Salvo L., Suéry M., Marmottant A., Limodin N., Bernard D. 3D imaging in material science: Application of X-ray tomography. *Comptes Rendus Physique*, 2010, vol. 11, p. 641-649.
- [San 79] Sands M. *The physics of electron storage rings. An introduction*. University of California, Santa Cruz, 1979, 190 p.
- [Sas 08] Sasikumar R., Walker M. J., Savithri S., Sundarraj S. Initiation of microporosity from pre-existing bubbles: a computational study. *Modelling and Simulation in Materials Science and Engineering*, 2008, vol. 16 035009, 14 p.
- [Sch 08] Scheid J.-F. *Analyse numérique des équations de Navier-Stokes*. 2008, 63 p.
- [Sch 02] Schneider P., Eberly D. *Geometric Tools for Computer Graphics*. Elsevier, 2002.
- [Sci 08] Science & Vie hors série No 243. *Construire un monde durable*. 2008.
- [Set 96] Sethian J. A. A Fast Marching Level Set Method for Monotically Advancing Fronts. *Proceedings of the National Academy of Sciences*, 1996, vol. 93, No 4, p 1591-1595
- [Set 99] Sethian J. A. *Level Set Methods and Fast Marching Methods*, Springer-Verlag, Cambridge University Press, 1999.
- [She 94] Shewchuk J. R. *An Introduction to the Conjugate Gradient Without the Agonizing Pain*. School of Computer Science, Carnegie Mellon University, 1994, 64 p.
- [Sis 11] Sistaninia M., Phillion A. B., Drezet J.-M., Rappaz M. Simulation of Semi-Solid Material Mechanical behavior Using a Combined Discrete/Finite Element Method. *Metallurgical and Materials Transactions A*, 2011, vol. 42, p. 239-248.
- [Smi 02] Smith K. A., Solis F. J., Chopp D. L. A projection method for motion of triple junctions by level sets. *Interfaces and Free Boundaries*, 2002, vol. 4, No 3, p 263-276.
- [Str 99] Strain J. Semi-Lagrangian Methods for Level Set Equations. *Journal of Computational Physics*, 1999, vol. 151, p. 498-533.
- [Str 00] Strain J. A Fast Modular Semi-Lagrangian Method for Moving Interfaces. *Journal of Computational Physics*, 2000, vol. 161, p. 512-536.
- [Str 01] Strain J. A fast semi-Lagrangian contouring method for moving interfaces. *Journal of Computational Physics*, 2001, vol. 169, p. 1-22.

Références bibliographiques

- [Sun 10a] Sun Z. *Déformation mécanique de la microstructure numérique de matériaux semi-solides dans un contexte des éléments finis*. Centre de Mise en Forme des Matériaux, MINES ParisTech, 2010, 23 p.
- [Sun 10b] Sun Z, Logé R. E., Bernacki M. 3D finite element model of semi-solid permeability in an equiaxed granular structure. *Computational Materials Science*, 2010, vol. 49, p. 158-170.
- [Sus 94a] Sussman M., Smereka P. Osher S. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 1994, vol. 114, p. 146-159.
- [Sus 94b] Sussman M. *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow*. Ph.D. Thesis in Mathematics. University of California, Los Angeles, 1994, 114 p.
- [Sus 00] Sussman M., Puckett E. G. A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 2000, vol. 162, p. 301-337.
- [Tan 06] Tan L., Zabarar N. A level set simulation of dendritic solidification with combined features of front-tracking and fixed-domain methods. *Journal of Computational Physics*, 2006, vol. 211, p. 36-63.
- [Ter 09] Terzi S., Salvo L., Suéry M., Limodin N., Adrien J., Maire E., Pannier Y., Bornert M., Bernard D., Felberbaum M., Rappaz M., Boller E. In situ X-ray tomography observation of inhomogeneous deformation in semi-solid aluminium alloys, *Scripta Materiala*, 2009, vol. 9, p. 449-452.
- [Tez 86] Tezduyar T., Park Y. Discontinuity-capturing finite element formulations for nonlinear convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 1986, vol. 59, p. 307-325.
- [Tez 03] Tezduyar T., Sathe S. Stabilization parameters in SUPG and PSPG formulations. *Journal of Computational and Applied Mechanics*, 2003, vol. 4, No 1, p. 71-88.
- [Thü 10] Thürey N., Wojtan C., Gross M., Turk G. A Multiscale Approach to Mesh-based Surface Tension Flows, *SIGGRAPH*, 2010, vol. 29, No 4, p. 1-10.
- [Tug 08] Tugurlan M. C. *Fast Marching Methods – Parallel Implementation and Analysis*. Ph.D. Thesis in Mathematics. Louisiana State University, 2008, 130 p.
- [Und 70] Underwood E. E. *Quantitative Stereology*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1970, 274 p.
- [Usg 11a] U. S. Geological Survey. *Mineral Commodity Summaries*. 2011, 201 p. Disponible sur Internet le 2 avril 2012 : <http://minerals.usgs.gov/minerals/pubs/mcs/2011/mcs2011.pdf>
- [Usg 11b] U. S. Geological Survey. *Historical Statistics for Mineral and Material Commodities in the United States. Aluminum Statistics*. 2011, 4 p. Disponible sur Internet le 2 avril 2012 : <http://minerals.usgs.gov/ds/2005/140/ds140-alumi.pdf>
- [Ver 06] Vernède S, Jarry P., Rappaz M. A granular model of equiaxed mushy zone: Formation of a coherent solid and localization of feeding. *Acta Materiala*, 2006, vol. 54, p. 4023-4034.

Références bibliographiques

- [Ver 07] Vernède S. A granular model of solidification as applied to hot tearing. Ph.D. Thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne, 2007, 175 p.
- [Vil 11a] Ville L., Silva L., Coupez T. Convected Level Set method for the numerical simulation of Fluid Buckling. *International Journal for Numerical Methods in Fluids*, 2011, vol. 66, p. 324-344.
- [Vil 11b] Ville L. *Simulation Modélisation multiphasique et calcul d'interface dans les procédés de mise en œuvre des propergols*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2011, 154 p.
- [Vor 03] Van der Vorst H. A. *Iterative Methods for Large Linear Systems*, Cambridge University Press, 2003.
- [Vu 07] Vu T.-S. *Modélisation des microstructures et microségrégations en solidification dendritique d'alliages*. Rapport de stage, MINES ParisTech, 2007, 38 p.
- [Vui 93] Vuik C. *Some historical notes on the Stefan problem*. Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1993, 12 p.
- [Wan 09] Wang Z., Yang J., Stern F. An improved particle correction procedure for the particle level set method. *Journal of Computational Physics*, 2009, vol. 228, p. 5819-5837.
- [Win 07] Winick H. *The impact of synchrotron light sources on science and society in developing countries*. Contemporary Physics: Proceedings of the International Symposium, Islamabad, 2007, p. 35-40.
- [Woj 09] Wojtan C., Thürey N., Gross M., Turk G. *Deforming meshes that split and merge*, ACM Transactions on Graphics (TOG), 2009, vol. 28, No 3.
- [Yan 06] Yang X., James A. J., Lowengrub J., Zheng X., Cristini V. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics*, 2006, vol. 217, p. 364-394.
- [You 05] Youssef S., Maire E., Gaertner R. *Finite element modelling of the actual structure of cellular materials determined by X-ray tomography*. *Acta Materialia*, 2005, vol. 53, p. 719-730.
- [Zal 79] Zalesak S. T. Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids. *Journal of Computational Physics*, 1979, vol. 31, p. 335-362.
- [Zar 08] Zaragoci J.-F. *Modélisation de la microstructure dendritique par une méthode level set*. Rapport de stage, MINES ParisTech, 2008, 28 p.
- [Zar 09] Zaragoci J.-F. *Construction d'une représentations éléments finis à partir d'observations expérimentales par microtomographie*. Mémoire d'avancement, Sophia-Antipolis. MINES ParisTech, 2009, 21 p.
- [Zer 10] Zerguine W. *Adaptation de maillage anisotrope et écoulements multifluides. Applications en injection assistée eau*. Thèse de Doctorat spécialité Mécanique Numérique, Sophia-Antipolis. MINES ParisTech, 2010, 128 p.
- [Zha 96] Zhao H.-K., Chan T., Merriman B., Osher S. A Variational Level Set Approach to Multiphase Motion. *Journal of Computational Physics*, 1996, vol. 127, p. 179-195.

Simulation numérique directe multiphasique de la déformation d'un alliage Al-Cu à l'état pâteux – Comparaison avec des observations par tomographie aux rayons X in situ en temps réel.

RESUME : La fissuration à chaud est un défaut majeur rencontré en solidification des alliages d'aluminium. Elle est liée à l'incapacité du liquide de s'écouler dans les zones où des porosités sont présentes, ne permettant pas de les refermer avant qu'elles gagnent en volume. Pour comprendre la fissuration à chaud, il est crucial de développer nos connaissances du comportement mécanique de la zone pâteuse. Pour cela, il est très utile d'effectuer des expériences de microtomographie aux rayons X et des simulations mécaniques sur des volumes élémentaires représentatifs. Dans cette thèse, nous proposons de coupler les deux approches en initialisant une simulation par éléments finis grâce à des données de microtomographie issues d'un test de traction isotherme d'un alliage d'aluminium-cuivre à l'état pâteux. Cette approche originale nous donne directement accès à la réalité expérimentale et permet des comparaisons des évolutions numérique et expérimentale de l'éprouvette. Nous expliquons dans un premier temps comment obtenir la représentation numérique à l'aide de l'algorithme des marching cubes et de la méthode d'immersion de volume. Nous présentons ensuite notre modèle numérique qui s'appuie sur une résolution monolithique des équations de Stokes. Une fois le champ de vitesse obtenu dans l'ensemble des phases solide, liquide et gazeuse, nous utilisons une méthode level set dans un formalisme eulérien afin de faire évoluer la morphologie de notre échantillon numérique. Malgré la simplicité du modèle, les résultats expérimentaux et numériques montrent un accord raisonnable en ce qui concerne la propagation de l'air à l'intérieur de l'échantillon.

Mots clés : Test de traction, Microtomographies in situ 3D en temps réel, Marching cubes, Éléments finis, Méthode level set

Direct and multiphase numerical simulation of the Al-Cu alloy deformation in the mushy state – Comparison with in situ and real-time X-ray tomography observations.

ABSTRACT : Hot tearing is a major defect arising during solidification of aluminium alloys. This defect is associated with the inability of liquid to feed areas where voids have started to appear, not allowing to heal small defects before they grow bigger. To understand hot tearing, it is mandatory to develop a good knowledge of the semi-solid mechanical behaviour. It is thus very useful to carry out X-ray microtomography experiments and mechanical simulations on representative elementary volumes. In this work, we couple both approaches by initialising a finite element simulation with the help of microtomography data obtained during an isothermal tensile testing of an aluminium-copper alloy in the mushy state. This innovative approach gives a direct access to the experimental reality and allows comparisons of numerical and experimental evolutions of the sample. We explain in a first time how to get the numerical representation thanks to a marching cubes algorithm and the immersed volume method. Then, we present our numerical model for which we solve the Stokes equations in a monolithic way. Once the velocity computed in all the solid, liquid and gaseous phases, we use a level set method in a Eulerian formalism to obtain the morphological evolution of our numerical sample. Despite the model simplicity, numerical and experimental results show a reasonable agreement concerning the air propagation inside the sample.

Keywords : Tensile testing, In situ and real-time 3D Microtomography, Marching cubes, Finite elements, Level set method

