High-order fictitious domain method for the elliptic and Navier-Stokes equations. Application to the fluid-structure coupling.

Arthur Sarthou

N°d'ordre : 3867

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGÉNIEUR

par Arthur SARTHOU

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : MÉCANIQUE

****

Méthodes de domaines fictifs d'ordre élevé pour les équations elliptiques et de Navier-Stokes - Application au couplage fluide-structure

High-order fictitious-domain methods for the elliptic and Navier-Stokes equations. Application to fluid-structure coupling

****

Soutenue le : 3 Novembre 2009

Devant la commission d'examen formée de :

M.	Angelo IOLLO	Pr., IMB, Université Bordeaux 1	**Président**
M.	Georges-Henri COTTET	Pr., LJK, Université Joseph Fourier	**Rapporteur**
M.	Jean-Luc ESTIVALEZES	Pr., IMFT et Ing. de Rech., ONERA	**Rapporteur**
M.	Philippe ANGOT	Pr., LATP, Université de Provence	**Examinateur**
M.	Jean-Paul CALTAGIRONE	Pr., TREFLE, Université Bordeaux 1	**Directeur**
M.	Stéphane VINCENT	MdC, TREFLE, ENSCPB	**Directeur**
M.	Jean-Pierre LAMBELIN	Ing. de Recherche, CEA	**Invité**
M.	David MONTFORT	Ing. de Recherche, EDF R&D	**Invité**

- 2009 -

"Science is like sex. Sure, it may give some practical results, but that's not why we do it."

Richard Phillips Feynman

Remerciements

Je tiens tout d'abord à remercier Georges-Henri Cottet, Professeur à l'Université Joseph Fourier et Jean-Luc Estivalezes, Ingénieur de Recherche à l'ONERA et Professeur associé à l'IMFT pour avoir accepté de rapporter ce travail. Leurs commentaires ont mis en lumière les points importants à traiter par la suite. Un grand merci à Angello Iollo, Professeur à l'université de Bordeaux pour avoir accepté de participer au jury.

Je remercie Philippe Angot, Professeur à l'Université de Provence, pour avoir fait partie du jury, et surtout pour son aide sur les aspects les plus mathématiques du sujet. Ses apports concernant les idées et la rigueur ont fortement enrichi ce travail.

Un grand merci à Jean-Paul Caltagirone, Professeur à l'Université de Bordeaux, pour avoir eu confiance en moi et avoir dirigé cette thèse. Sa grande expérience et ses idées ont été un aiguillage crucial durant ces trois années.

Je remercie vivement Stéphane Vincent, Maître de Conférence à l'ENSCBP, pour m'avoir encadré durant toutes ces années et avoir fait de mon parcours ce qu'il a été. Je ne louerai jamais assez son ouverture d'esprit et la liberté qu'il m'a accordé, tout en étant présent quand il le fallait. Merci pour tout.

Concernant les partenaires de cette thèse, je remercie EDF R&D pour avoir permis à cette thèse d'exister, et tout particulièrement Marc Sakiz et David Montfort. Travailler avec eux a toujours été un plaisir. Je remercie ensuite l'entreprise Michelin pour son soutien ainsi que Pierre Février et Fabien Sonilhac. Cette collaboration fut l'occasion d'échanges scientifiques très intéressants. Merci enfin à Varel Europe et à Alfazazi Dourfaye. L'application de la thèse au domaine pétrolier fut très enrichissante.

Je remercie ensuite Jean-Pierre Lambelin, Ingénieur de Recherche au CEA, pour avoir accepté de faire partie du jury. J'en profite aussi pour remercier François Nadal, et Frédérique Delaurens. Leurs enseignements durant mon stage ont largement contribué à faire de ce travail ce qu'il est aujourd'hui.

Merci aussi à tous les jeunes docteurs dont les manuscrits m'ont souvent éclairé. Cela reste selon moi l'un des rôles les plus importants d'un manuscrit de thèse. Merci donc à Isabelle Ramière, Claire Bost, Aline Lefebvre et Sébastien Tanguy. Sur ce point je remercie particulièrement Frédéric Couderc. Les discussions critiques que nous avons eu permettront d'améliorer ce travail.

Du côté du TREFLE, je remercie tout d'abord Éric Arquis, directeur du laboratoire, pour son accueil et Mejdi Azaiez, chef de feu l'équipe MFEN, pour ses conseils et pour sa contribution financière à la thèse.

Toujours du côté du TREFLE, je remercie d'abord les anciens thésards dont les équations hantent encore le labo : Stéphanie, Aurélie le D. et Nirina. Mention spéciale pour Sylvain, ex-presque-voisin de Bergonié toujours de bonne humeur, et mention très spéciale pour le très sympathique Grégoire Pianet qui, grâce à sa culture de bon gout, doit être le seul à avoir compris toutes mes blagues (et vice-versa). Un grand merci à Delphine L.P. pour sa gentillesse et le

plaisir de travailler avec elle. Une larme me chatouille l'œil à chaque fois que je vois notre fils Bob s'exhiber dans les médias nationaux. Merci enfin à Etienne, flegmatique et sympathique basque, compère de cantine et conseiller en candidature CNRS/MdC.

Merci aussi à Nicolas pour son inébranlable gaieté et sa connoissance du bon françois. Je remercie aussi vivement Jérôme, aka la Rockeuse de Diamant pour son aide lors de mon stage et pour les sessions hippies sur les quais. Merci aussi à Aurélie M. W. pour les pauses-thés. Du côté des morts au combat, merci à toi Guillaume B. à l'humour terrifiant et à la sympathie affûtée, Erwan notre Tintin au pays de l'or noir, Houssen qui savait parler au rappeur qui est en moi, Matthieu le loup-de Mer et D'hav l'ancien collègue du DEA.

Du côté des permanents, merci à Pierre pour son soutien musical et à son favori Cédric pour ses explications motivées, Stéphane Glockner pour son infinie patience, Jean pour son soutien moral dans le tram du matin, Pascale pour son agréable voisinage et le Dies Irae, Arnaud pour ses fines analyses socioculturelles de l'Euskal Herria et David pour ses appareils rigolos. Merci aussi à notre chère secrétaire-gestionnaire modèle Valérie. Merci aussi à Marie-Paule pour son incroyable gentillesse. Enfin, merci à Jocelyn, je me sens moins seul dans le labo quand je parle de glVertex3f().

Du côté des plus jeunes, merci et bravo à Mathilde, seule survivante avec moi-même de la terrible promo 2009. C'est quand tu veux pour un nouveau battle de "Mais non c'est toi qui a plus bossé que moi" ou pour une soirée lors de laquelle "Non je n'ai presque pas bu". Passons maintenant à la plèbe. Je remercie Romain aka Jah Benny Rastafari Hailé Sélassia, spécialiste de musique qui fait monter et du all-in suicidaire. Ses exils à Metz nous rappellent que l'on est finalement peu de chose. Merci à Delphine V. pour sa joviale et entière compagnie, à Jérémie pour sa bonne humeur et pour m'avoir repris le titre de geek officiel du labo. Merci aux deux discrets ex-MATMECA, Céline la diablesse du Nord et Ludo le gossbo des îles. Et Merci à l'encore plus discret mais tout aussi sympathique Louis. Enfin, merci aux sbires de Pascale, Tarik le souriant et Honda l'éclair des terrains de foot.

Je me dois aussi de remercier mes propres stagiaires, Wided du lointain orient, Ludo "C'est bon j'ai fini", Arnaud "Ya un problème", Bastien "J'arrive pas a instancier le Node de la Scene", et tout particulièrement Simon aka Gros Kamatch' pour m'avoir traîné dans des soirées peu fréquentables.

Et bonne chance à Adrien E. qui récupère le bébé. Bonne chance petit.

En dehors du labo, un grand merci à Ju l'autiste et Romain le futur maître du monde pour toutes ces années en votre compagnie. Merci aux collègues Sylvain le "Bogoss" fassiste, Adam "Laden" l'homme à la salopette et à la cravate en bois et Hub' pour sa créativité capillaire. Enfin, un grand *trugarez* à Jean-Philippe Nicolas pour son enseignement et son amitié. Et pour en finir avec l'Université, merci à toute la chorale et particulièrement à Françoise Leroy pour ces mardi midis polyphoniques. Merci aussi à Roman Travé, lui aussi compromis dans les concerts métals familiaux et l'atelier BD. Enfin, je remercie mes anciens collègues de ce lieu où l'on travaillait à la fin du monde (mais chhuuutt), Cécile et Sylvain.

En dehors de l'université, je tiens à remercier tous ceux qui ont fait ces trois années de thèse. Merci aux petits gars du fanzine Zymaze et de l'atelier, et aux membres des Écoles Bushido.

Un grand merci aux membres de Jeunes-Science, passé et présent : Christelle, Greg et toute sa famille, Mathieu et Eric, Bastien, Antoine M., Vanessa, Baptiste, Guillaume, Florence et Ludo. Merci aussi à tous les adhérents à mèche que j'ai fait souffrir. Merci aux camarades d'études de près ou de loin, Peter, Manu, Piwel et les collocs, David et Romeo.
Merci aussi à tout ceux que la musique m'a fait connaître, tout particulièrement Ben et Mounir.

Un remerciement spécial aux amis de longue date : Vincent, Yana, Ludo, Ambre, Sandra, Niafron, Simon, Antoine S. et Zelimir.

Merci beaucoup à Gwendy pour son soutien dans la difficile période que fut la fin de thèse et aussi pour tout le reste.

De tout mon coeur, merci à ma famille qui plus que tout autre a fait de moi le nouveau Docteur Sarthou. Je remercie ma mère et ma soeur Marie, mon père qui est loin, ma belle-mère et mes autres frères, soeurs et beaux-frères, mes oncles et cousins, de Bordeaux, Berson et des Landes.

A mes grands-parents et à ma chère tante Annette, merci pour tout ce que vous m'avez donné, ce document vous est dédié.

Résumé

La plupart des applications industrielles de la simulation numérique mettent en oeuvre des objets, des frontières ou de façon plus générique des interfaces de formes complexes. Concrètement, ces interfaces correspondent à des discontinuités des variables physiques telles que la masse volumique ou la viscosité des fluides, la conductivité thermique de deux matériaux ou encore les propriétés de la matière dans le cas d'une interface fluide-solide. En plus de séparer deux milieux, ces interfaces font apparaître des phénomènes physiques spécifiques comme les tensions de surface ou des propriétés thermodynamiques particulières dans le cadre d'un changement de phase. Ainsi, l'importance des interfaces exige une attention particulière quand à leur modélisation et leur discrétisation. De ce point de vue, deux difficultés principales sont à surmonter. D'une part, une interface a une épaisseur que l'on peut la plupart du temps considérer comme nulle, alors que la discrétisation spatiale standard des méthodes de simulation repose sur un découpage en volume. On pourra s'accommoder de cela dans certains cas, quand l'interface passe exactement entre deux volumes discrets, ou encore au milieu des volumes de discrétisation. Toutefois, cette configuration ne peut pas toujours être obtenue ce qui est le second problème lié à la discrétisation des interfaces. Les cas industriels que l'on souhaite simuler mettent souvent en jeu des interfaces de formes complexes, comme des pneus, des véhicules ou tout simplement des surfaces libres. Il est souvent difficile d'obtenir une discrétisation spatiale conforme à ces interfaces. La méthode des éléments finis par exemple, donne naturellement une grand liberté dans la discrétisation spatiale mais nécessite un effort de maillage conséquent voir pénalisant en terme de performances si les interfaces sont mobiles et exigent donc la création d'un nouveau maillage à chaque itération. Une autre approche consiste à utiliser un maillage fixe non-conforme aux interfaces. La discrétisation spatiale est bien plus simple mais ne correspond pas aux interfaces, et ce au détriment de la précision.

Les méthodes de domaines fictifs proposent d'améliorer la précision de la discrétisation aux interfaces en modifiant de façon plus ou moins directe la discrétisation des équations au voisinage de l'interface. On peut ainsi retrouver l'ordre des schémas initiaux malgré la présence de discontinuités sur des interfaces non-conformes complexes et mobiles.

Toutefois, la simulation de cas industriels complexes ne requiert pas uniquement une discrétisation précise des équations à proximité des interfaces. Une première étape consiste tout d'abord à détecter l'interface et les sous-domaines qu'elle délimite. Ces opérations requièrent la plupart du temps des algorithmes tirés de l'informatique graphique dont la vitesse d'exécution peut aller du simple au centuple selon l'implémentation ou la méthode utilisée. L'interprétation spatiale des interfaces permet alors une application des méthodes de domaines fictifs. Il se peut qu'une physique particulière soit nécessaire à l'intérieur d'un sous-domaine, un objet immergé par exemple. On parle dans ce cas de couplage fluide-structure et une modélisation du mouvement solide, qu'il soit rigide ou déformable, doit être mise en oeuvre. Au final, la multitude des méthodes numériques peut nécessiter un effort de post-traitement particulier à des fins de visualisation spécifique ou pour produite des films de vulgarisation. Le présent document traite de tous ces domaines et propose ainsi une démarche globale pour la simulation des interactions fluides-structures et des transferts thermiques. Les méthodes proposées ont pour objectif l'obtention d'une précision spatiale générale à l'ordre deux.

Méthodes existantes de domaines fictifs

La première partie du document traite de différentes méthodes de la littérature. Trois catégories sont considérées :

- Les méthodes de pénalisation. De nombreuses méthodes de pénalisation ont été développées

au laboratoire TREFLE. Elles font ainsi l'objet d'un chapitre. Le principe de ces méthodes est de conserver une même équation dans tous le domaine et d'obtenir divers comportements locaux (frontière, interface, mouvement rigide...) en modifiant localement certains termes des équations.

- Les méthodes de frontières immergées. Sont traitées ici toutes les méthodes qui ne prennent en compte la solution que d'un côté d'une interface. Elles permettent d'imposer des conditions de Dirichlet, Neumann ou Stefan sur une interface immergée qui devient une nouvelle frontière du domaine résolu. Nous décrivons ainsi les méthodes IBM (Immersed Boundary Method) de type Peskin et Direct-Forcing, DLM (Distributed Lagrange Multiplier), les méthodes Cartesian Cell ainsi que la méthode Ghost-Fluid pour les frontières.

- Les méthode d'interfaces immergées. Ces méthodes permettent d'imposer précisément des conditions de saut et de transmission sur une interface immergée. Nous décrivons ici les méthodes IIM (Immersed Interface Method), Ghost-Fluid et MIB (Matched Interface and Boundary).

Gestion de formes lagrangiennes sur maillages curvilignes

Cette partie décrit une nouvelles méthodologie de projection de formes sur maillage curviligne. La première étape consiste à "déplier" le maillage eulerien curviligne vers un maillage cartésien à pas unitaire. Les interfaces discrétisées à l'aide de surfaces triangularisées sont projetées dans le nouveau repère en utilisant les mêmes facteurs de transformations que les cellules du maillage eulerien. Ce nouveau maillage permet l'utilisation d'une méthode rapide de Ray-Casting qui fournit ligne par ligne l'appartenance ou non des points euleriens au domaine défini par l'objet immergée. D'une façon générale, le maillage cartésien à pas unitaires permet l'accélération de nombreux calculs comme celui des efforts sur la surface d'un objet immergé. De nombreuses propriétés étant conservées d'un maillage à un autre, l'application de méthodes de domaines fictifs peut parfois se faire de façon transparente dans l'ancien repère curviligne ou le nouveau repère cartésien. Dans le cas des méthodes de suivi d'interface, une transformation simple du champ de vitesse est nécessaire. La précision et la rapidité de ces méthodes sont aussi étudiées

Nouvelles méthodes de domaines fictifs

Deux nouvelles méthodes de domaines fictifs sont décrites dans cette partie. La méthode de pénalisation de sous-maille (PSM, ou SMP pour Sub-Mesh Penalty) permet d'imposer une condition de Dirichlet à l'ordre deux pour la norme L^2 sur une frontière immergée non-conforme. C'est la première discrétisation de la pénalisation L^2 à atteindre un ordre deux en espace. La méthode permet aussi d'imposer des conditions de Neumann à l'ordre un. Elle remplace le terme de pénalisation volumique habituel [Ango 99] $\beta_i(\mathbf{u}_i - \mathbf{u}_D)$ par $\beta_i \sum_{\mathbf{x}_j \in Vois(\mathbf{x}_i)} (\alpha_j \mathbf{u}_j - \mathbf{u}_D)$. En conséquence, la contrainte de pénalisation pour un noeud $\mathbf{x}_i$ prend en compte les valeurs de la solution au voisinage de ce point. Les coefficients α_i sont construits à partir d'interpolations de Lagrange. L'imposition de la contrainte est totalement implicite et ne nécessite pas d'inversion de matrices supplémentaire. Cette méthode est applicable directement aux équations scalaires elliptiques. Pour les équations de Navier-Stokes, la mise en place de la méthode est directe si le couplage vitesse-pression s'effectue avec une méthode de lagrangien augmenté. Dans le cas d'une méthode de projection scalaire, une correction de l'étape de projection de pression et de correction de vitesse doit être mise en place.

La seconde méthode, dite Algebraic Immersed Interface and Boundary (AIIB) étend le principe de la PSM aux interfaces. Au noeud qui était précédemment pénalisé afin de porter la contrainte de frontière est ajouté une inconnue dite auxiliaire car elle cohabite avec l'inconnue

d'origine qui est physique. Ainsi, une équation physique et une contrainte d'interface cohabitent en un même nœud du maillage. L'imposition de conditions de Dirichlet ou Neumann sur une interface fine est directe. Une manipulation des contraintes de pénalisation et des conditions d'interface permet de traiter des cas de transmission et de saut au travers de cette dernière. Ces cas sont simulés à l'ordre deux en espace pour un cas de conduction thermique. L'objectif à terme est d'utiliser cette méthode dans le cadre des équations de Navier-Stokes pour traiter les problèmes à surface libre ou des cas de couplage fluide-structure implicite.

Le dernier chapitre de cette partie est dédié à la validation des deux précédentes méthodes. Les validations pour les cas frontière sont en partie communes aux deux méthodes.

Mécanique du solide et couplage fluide-structure

La première partie est dédiée à la modélisation du mouvement des solides isolés par une application classique du principe fondamental de la dynamique. Un modèle simple de collision est implémenté et couplé à une visualisation OpenGL.

Nous présentons ensuite une méthode de couplage fluide-structure. La discrétisation en temps est une marche alternée classique. La nouveauté consiste en l'utilisation de la méthode de pénalisation d'ordre deux pour prendre en compte les frontières objets lors de la résolution fluide. Les méthodes de projection de maillage présentées dans ce document sont utilisées à chaque pas de temps.

Applications industrielles

Afin d'illustrer notre démarche, nous présentons trois applications complexes qui ont été traitées durant la thèse.

Nous montrons d'abord les résultats du projet mené avec Varel Europe. Cette entreprise conçoit et fabrique des têtes de forage pour le milieu pétrolier. La durée de vie de ces têtes dépend entre autre de la bonne évacuation des copeaux de roche. A cet effet, de la boue est en permanence injectée à haut débit dans la tête. La bonne évacuation des copeaux dépend donc énormément des caractéristiques de l'écoulement. Ainsi, la présence de grandes recirculations favorise la création d'agglomérats de copeaux qui rendent la tête inutilisable. Le but du projet était de concevoir à partir du code de calcul Thétis un simulateur prenant en compte l'hydrodynamique de la boue ainsi que la génération et l'évacuation des copeaux. Une méthode de suivi lagrangien des copeaux a été développée spécialement à cet effet.

L'application suivante concerne l'étude de l'hydroplanage d'un pneu. L'objectif était de fournir à l'entreprise Michelin un simulateur pouvant caractériser la force verticale induite par une masse d'eau impactant un pneu en roulement. Ce cas a nécessité la prise en compte d'obstacles en rotation de forme complexe et détaillée. La géométrie du pneu étant modifiée à chaque pas de temps (rotation et déformation), il a été nécessaire d'avoir des méthodes de projection de maillage rapides et robustes. Une méthode surfacique de calculs d'efforts a de plus été mise au point.

Le troisième cas est la simulation des écoulements de convection naturelle dans la grotte de Lascaux. Le but est de prédire ou d'expliquer l'impact des différents choix de conservation ou d'une présence humaine dans la grotte. La simulation prend en compte l'écoulement fluide, les échanges thermiques ainsi qu'un modèle d'humidité.

Annexes

Une première annexe décrit les différentes équations et méthodes numériques utilisées dans ce document. On y décrit en particulier l'approche volume finie sur grille décalée et le couplage vitesse-pression par les méthodes de projection scalaire et de lagrangien augmenté.

La seconde annexe décrit l'application des résultats aux images de synthèse en expliquant la

méthode de couplage et en présentant quelques cas.

Enfin, nous expliquons la construction de diverses interpolations, notamment les fonctions Kernel.

Contents

Notations

Geometry

$\mathbf{n}$	Normal unit outward vector to Ω_0 on Σ
d	Dimension of space
$\mathcal{E}$	Set of index of the Eulerian structured mesh
h	Eulerian mesh step defined as $h = \sup_{I\in\mathcal{E}} h_I$
h_I	Maximum length of a $\mathcal{V}_I$
$\mathcal{I}$	Set of index of the intersection points between σ_e and the faces σ_l
$\mathcal{L}_f$	Set of index of the Lagrangian mesh
$\mathcal{L}_v$	Set of index of vertices
$\mathcal{N}_0, \mathcal{N}_1$	Set of index of the Eulerian interface points
$\mathcal{O}$	Set of index of immersed objects
Ω	Main physical domain
Ω_0	The fluid domain
Ω_i, $i > 0$	The solid domain corresponding to the ith object
σ_e	Faces of the Eulerian dual mesh
σ_l	Faces of the Lagrangian mesh
Σ	Immersed interface
Σ_h	Piecewise linear approximation of Σ such as $\Sigma_h = \{\sigma_l \in \mathbb{P}_1^{d-1}, l \in \mathcal{L}_f\}$
$(\mathcal{V}_I)$	Set of cell-centered finite volume
$(\mathcal{V}'_I)$	Set of dual finite volume
$x_{l,i}$	Vertices of face σ_l for $i = 1, d$

Eulerian volume functions

C	Color Phase function
ϕ	Level-set function
χ	Heaviside function

Physical variables

$\mathbf{D}$	rate-of-strain tensor	$N.m^{-2}$
F	Wall forces applied to the solid domain	N
H_0	Angular momentum	$kg.m^2.s^{-1}$
I	Inertia matrix of the solid	$kg.m^2$
m	Mass	kg
M	Torque	$N.m$
μ	Dynamic viscosity	$Pa.s$
ω	Rotation vector of the solid	$rad.s^{-1}$
p	Fluid pressure	Pa
ρ	Fluid density	$kg.m^{-3}$
$\boldsymbol{\sigma}$	stress tensor	$N.m^{-2}$
t	Time	s
T	Temperature	K
$\mathbf{u}$	Fluid velocity	$m.s^{-1}$

Superscripts

$*$ Auxiliary entities

Parameters

ε Penalty parameter

Abreviations

AIIB	Algebraic immersed interface and boundary
AL	Augmented Lagrangian
BC	Boundary condition
DF	Direct-forcing
DLM	Distributed Lagrange multipliers
FT	Front-tracking
FV	Finite volume
IB	Immersed boundary
IBM	Immersed boundary method
II	Immersed interface
IIM	Immersed interface method
LS	Level-set
MIB	Matched interface and boundary
SMP	Sub-mesh penalty
SMPM	Sub-mesh penalty method
TVD	Total variation diminushing
VOF	Volume of fluid

Function spaces, norms

The following function spaces and normes are defined on an open $\Omega \subset \mathbb{R}^d$ for a measurable function v.

$\|\mathbf{x}\|$	$=$	the Euclidian norm of $\mathbf{x} \in \mathbb{R}^d$
$a.e.$	$=$	almost everywhere
dx	:	Lebesgue measure on $\mathbb{R}^d$
$L^p(\Omega)$	$=$	$\{ v : \Omega \to \mathbb{R};\ v$ measurable and $\|v\|_{L^p(\Omega)} < \infty\}$
$\|v\|_{L^p(\Omega)}$	$=$	$\left(\int_\Omega |v(x)|^p dx\right)^{1/p},\ 1 \leq p < \infty$
$\|v\|_{L^\infty(\Omega)}$	$=$	$Inf\{C;\ |v(x)| \leq C\ a.e.\ on\ \Omega\}$
$H^p(\Omega)$	$=$	$\{v \in L^2(\Omega),\ \partial^\alpha v \in L^2(\Omega),\ \alpha \in \mathbb{N}^d,\ \alpha \leq p\}$, Hilbert space
$H_0^p(\Omega)$	$=$	$\{v \in H^p(\Omega), v_{\|\partial\Omega} = 0\}$

Error norms

We define u_h the computed solution such as

$$\forall \mathcal{V}_I \in \mathcal{T}_h, \forall x \in \mathcal{V}_I, u_h(x) = u_I \tag{1}$$

The discrete relative L^2 error is defined as:

$$\|u_h - \tilde{u}\|_{L^2_{rel}(\Omega)} = \frac{\|u_h - \tilde{u}\|_{L^2(\Omega)}}{\|\tilde{u}\|_{L^2(\Omega)}} = \Big(\sum_{x_I \in \Omega} meas(\mathcal{V}_I) |u_I - \tilde{u}(x_I)|^2 \Big)^{\frac{1}{2}} / \Big(\sum_{x_I \in \Omega} meas(\mathcal{V}_I) |\tilde{u}(x_I)|^2 \Big)^{\frac{1}{2}} \tag{2}$$

with $\tilde{u}$ the analytical solution.
The discrete L^∞ error is defined as:

$$\|u_h - \tilde{u}\|_{L^\infty(\Omega)} = max_{I \in \mathcal{N}} |u_I - \tilde{u}(x_I)| \tag{3}$$

List of Figures

Part I

Introduction

La valeur de la simulation numérique en tant qu'aide à l'ingénierie ou à la compréhension des phénomènes physiques n'est plus à démontrer. Cet outil relativement jeune possède bien sûr ses limitations et ses difficultés d'emploi. Le premier pas dans la conduite d'une simulation complexe est l'obtention d'un résultat a priori plausible. Ce dernier point étant toutefois très subjectif, les indicateurs numériques et les études de convergence permettent souvent de tendre vers l'objectivité. Celle-ci peut toutefois être illusoire si les phénomènes physiques mis en jeux sont mal compris ou trop fins pour être captés par la simulation. On prendra l'exemple de cas de réservoirs remplis de fluide en rotation-precession [Lamb 09] dont la déstabilisation dépend de modes résonnants totalement masqués, dans le cas de géométries trop complexes, par la diffusion numérique. On citera aussi les simulations en 6D des équations de Vlasov-Maxwell dont la croissance en maillage laisse peu de latitudes dans le raffinement et donc dans l'étude de convergence. Ainsi, tout ne peut pas être simulé avec pertinence. Les possibilités de la simulation dépendent fortement de la puissance de calcul disponible, mais on ne peut que naïvement se reposer sur celle-ci pour espérer résoudre à plus ou moins brève échéance des problèmes de plus en plus complexes. L'évolution de la structure des machines qui voit poindre la fin des machines monoprocesseur (et donc du calcul séquentielles) et l'ordre de complexité des algorithmes montrent l'importance du développement de méthodes et de modèles toujours mieux pensés. Des résultats certes impressionnants sont obtenus à l'aide de super calculateurs mais peu de structures disposent de telles machines. Dans beaucoup d'entreprises, petites ou mêmes parfois très grandes, la simulation n'est pas la priorité et les bureaux d'études ne disposent parfois que d'ordinateurs personnels pour réaliser des calculs.

Ce travail parle de méthodes numériques plus que de modélisation. Les équations considérées ici sont bien connues de même que les phénomènes physiques traités. Notre objectif est de développer des méthodes plus rapides, robustes et précises afin de mieux simuler ce que l'on simule déjà de façon limitée et biaisée. Le champ d'application de ces méthodes est résolument la simulation de cas industriels. Encore une fois, les phénomènes mis en jeux dans l'hydroplanage d'un pneu, le vol d'un planeur ou la convection naturelle dans une grotte sont assez bien connus, voir totalement appréhendés dans le cadre d'études de cas académiques simples. La difficulté à prévoir précisément le comportement fluide pour les cas industriels est tout autre et trouve en grande partie sa source dans la complexité des formes en présence. Ce saut de difficulté se transpose en simulation numérique, où reproduire avec précision l'écoulement autour d'un avion demande bien plus de développements et de puissance de calcul que la simulation de l'écoulement autour d'une sphère. D'une part, la complexité accrue de l'écoulement impose l'utilisation de méthodes plus précises et/ou d'un maillage plus fin. D'autre part, la gestion de formes complexes non-triviales (avion, éolienne, pneu) nécessite de nombreux développements informatiques.

Nous proposons ici un ensemble de méthodes permettant le passage de la sphère à l'avion. Le choix fondamental qui sous-tend ce travail est celui de l'utilisation d'un maillage de calcul structuré simple et fixe. Notre approche globale sera celle des domaines fictifs. Ce parti-pris s'oppose à l'approche classique de la simulation industrielle qui consiste à utiliser un maillage non-structuré permettant une gestion directe des formes complexes. Nous tacherons de montrer que notre approche se montre viable sur de tels cas même sans l'utilisation d'une grande puissance de calcul.

Objectifs et financement du travail

Ce travail a été financé par divers biais.

- Un tiers vient d'une collaboration avec EDF R&D (Châtou, en collaboration avec Marc

Sakiz et David Montfort) qui consistait à étendre à l'ordre 2 la méthode de pénalisation L^2 et à l'implémenter dans *Code_ Saturne*.

- L'équipe de recherche Mécanique des Fluides et Énergétique Numérique du laboratoire TREFLE a financé un tiers supplémentaire.

- La dernière partie a été assurée par la réalisation de contrats industriels pour Michelin (Pierre Février et Fabien Sonilhac), et Varel Europe (Alfazazi Dourfaye) en collaboration avec Laurent Gerbau (Armines) et la Région Aquitaine.

Ce travail a aussi mené à une collaboration avec Delphine Lacanette et Philippe Malaurent portant sur la grotte de Lascaux. Enfin, une collaboration avec Philippe Angot (LATP) a permis de traiter des aspects plus théoriques.

L'objectif initial de ce travail était donc le développement d'une méthode de pénalisation d'ordre élevée pour des obstacles fixes puis son extension aux objets mobiles. L'intérêt limité d'une telle méthode de frontière pour l'équation de l'énergie a motivé son évolution vers une méthode d'interface immergée. Afin de pouvoir traiter les cas industriels, notamment ceux qui ont co-financé cette thèse, il s'est vite révélé indispensable de développer des méthodes de projection de maillages performantes et cet aspect est devenu un des sujets principaux de ce travail.

Concernant la partie visualisation et images de synthèse, nous menons ce travail depuis un certain temps déjà et il s'enrichira vite des possibilités qu'offrent les travaux de cette thèse. Notons aussi une collaboration avec SVG sur l'aspect visualisation.

Mis à part l'introduction et la conclusion, ce mémoire a été rédigé en anglais.

Organisation du mémoire

Le but de ce document est de présenter une méthodologie globale de simulation d'écoulements et de transferts thermiques mettant en jeu des objets ou interfaces de formes complexes. Les principales étapes de la gestion des objets sont les suivantes:

- Les objets représentés par des maillages triangularisés surfaciques sont projetés sur le maillage eulérien à l'aide de diverses méthodes.

- Une méthode de domaines fictifs permet ensuite de modifier la discrétisation initiale des opérateurs spatiaux pour permettre une meilleur prise en compte des objets.

- Les objets sont mis en mouvement par une fonction analytique ou sous l'effet de l'effort fluide.

Une dernière partie illustre cette démarche par la présentation de cas industriels traités durant cette thèse.

Part II: Overview of the fictitious domain methods La première partie compare les approches à maillages structurés et non-structurés, et présente les méthodes de domaines fictifs les plus couramment utilisées. Nous présentons aussi les deux grandes classes de problèmes rencontrés, les problèmes de frontières immergées et ceux d'interfaces immergées.

Part III: Management of Lagrangian shapes on curvilinear grids Cette partie décrit une nouvelle méthode globale de traitement d'objets qui trouve son originalité dans l'utilisation systématique d'une grille cartésienne duale à pas unitaires. Cette approche est validée sur des problèmes d'advection diphasique et de frontières immergées.

Part IV: High-order fictitious domain methods La troisième partie constitue le cœur du document et présente deux méthodes originales de domaines fictifs. La méthode de pénalisation de sous-maille est la première méthode de pénalisation d'ordre élevé en espace et permet d'imposer des conditions de Dirichlet ou de Neumann sur une frontière. La méthode d'interface et de frontière immergée algébrique est quand à elle une extension de la pénalisation de sous-maille aux cas des interfaces. Ce n'est plus une méthode de pénalisation mais une méthode algébrique se basant sur la création d'inconnues auxiliaires. De nombreuses validations pour des équations elliptiques et les équations de Navier-Stokes sont présentées.

Part V: Solid mechanics and fluid-structure coupling Nous décrivons la modélisation du mouvement d'un objet solide et du couplage fluide structure. Un code temps réel d'interaction de particules est présenté et la méthode de couplage est appliquée à la sédimentation d'une particule.

Part VI: Industrial applications Cette partie présente différents cas complexes traités avec notre méthodologie. Nous présentons d'abord deux cas industriels concernent l'hydroplanage d'un pneu avec Michelin, et les écoulements dans des têtes de forage avec Varel Europe. Le troisième cas est la simulation de la convection naturelle dans la grotte de Lascaux. Quelques autres illustrations sont données.

Appendix A: Conservation equations and related numerical context Cette annexe présente les modèles et méthodes numériques utilisées.

Appendix B: Application to the image synthesis Nous exposons ici notre méthodologie de création d'images de synthèse.

Appendix C: Analytical solutions La construction de quelques solutions analytiques utilisées est expliquée.

Appendix D: Interpolation Nous décrivons les différentes interpolations de type polynomiales ou Kernel utilisées.

Part II

Overview of the fictitious domain methods

Table of Contents

Chapter 1

Base principle and motivations

1.1 A short story of mesh

THE resolution of a complex physical problem by the numerical simulation leads generally to a discretization in space and time of the initial modeling. Phenomena changing along the time will be solved step by step. When the solution is required on a non singular domain or when the problem cannot be solved by considering an averaging in space of the physical quantities, the space is discretized, that is to said cut into pieces, boxes, squares, rectangle, polyhedra etc... The spatial discretization generates a mesh, composed of cells, faces, vertices and nodes. The discretization of the operators, and consequently the accuracy of the solution, will depends on the discretization of the space and time.

A first classification of a calculation volume mesh is based on the number of neighbors of each cell and node (their *valence*). If this number is constant over the whole domain, the mesh is defined as *structured*, otherwise *unstructured*. Each class of mesh can be divided into many sub classes.
Let us now introduce some definitions :

Element An element is the elementary volume in which the conservation equations are discretized. The set of elements K_i of a domain Ω_h are such that

$$\bigcup_{K_i \in \mathcal{T}_h} K_i = \Omega_h \tag{1.1}$$

with $\mathcal{T}_h$ a given tessellation of Ω_h. Furthermore,

$$\bigcap_{K_i \in \mathcal{T}_h} \overset{\circ}{K_i} = \emptyset. \tag{1.2}$$

Hence, an element has the same dimension as Ω_h and is generally a polygon.

Node The nodes are the vertices of the elements K_i. They are generally the centers of the control volumes when a finite volume discretization is considered.

Face The faces are elements of dimension $d-1$ and are the boundaries of the elements. When the elements are polygons, the faces are segments for a 2D tessellation and 2D polygons for a 3D tessellation.

Orthogonal A tessellation is denoted as orthogonal if all faces which have a common node are orthogonal or parallel in a sufficiently close vicinity of the node.

1.1.1 Structured grid

A regular grid is a tessellation of the Euclidian space $\mathbb{R}^d$ by rectangles in 2D and boxes in 3D. Each cell can be indexed by a coefficient, or (i, j) in 2D and (i, j, k) in 3D. Structured grids can be classified in sub categories:

Regular grid A regular grid is composed of rectangles or parallelepipeds that all have the same proportions. One can find a translation of the grid such as each vertex indexed (i, j, k) has coordinates $(i.\Delta x, j.\Delta y, k.\Delta z)$, where Δx, Δy and Δz are the grid spacing, or the space steps.

Cartesian grid A Cartesian grid is composed of unit squares or unit cubes, *i.e.* $\Delta x = \Delta y = \Delta z = 1$. One can find a translation of the grid such as each vertex indexed (i, j, k) has coordinates (i, j, k). Hence, it is straightforward to determine which cell a Lagrangian point belongs to.

Rectilinear grid A rectilinear grid is composed of rectangles or parallelepipeds that can have different dimensions.

Curvilinear grid A curvilinear grid is a grid with the same combinatorial structure as a regular grid, in which the cells are quadrilaterals or cuboids rather than rectangles or rectangular parallelepipeds.

1.1.2 Unstructured grids

Although the valence of each elements of an unstructured mesh is not constrained, it is common to use a unique kind of elements to perform computations. In 3D, tetrahedral and hexahedral elements are generally used. One advantage of these elements is their convexity, so the discretization of the equations is easier and more robust than with non-convex elements.

1.1.3 Structured and unstructured grids

The following aspects have to be considered:

- The discretization of the space is not always trivial, especially if the shape of the domain (*i.e.* its boundaries) is complex. The construction of a mesh from a complex shape requires complex algorithms and has often to be performed on external meshers such as Gambit (ANSYS) or Gridgen (Pointwise).
- The discretization of the spatial operators will be easier with a simple discretization. Centered second-order operators are trivial to build and accurate on Cartesian meshes. Conversely, on unstructured meshes of poor quality (if, for instance, the perpendicular bisectors of each faces do not intersect in a single point), complex operations are required to build discrete operators.
- As the grid is composed of simple geometrical entities, the boundary of the discretized domain does not generally conform with the boundary of the original domain. As will be exposed many times in this document, if the segment size of the reconstruction depends on the size of the discrete elements, a stair-step reconstruction (Fig. 1.1.left) is of first

order only while a linear piecewise reconstruction (Fig. 1.1.right) provides a second order of approximation. More complex reconstructions with splines are not considered here.

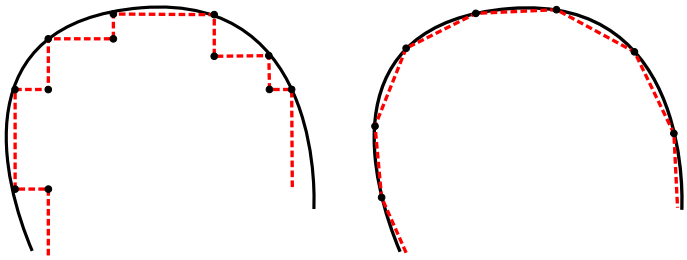


Figure 1.1: First order (left) and second order (right) reconstruction (dashed line) of an interface (solid line)

- The accurate discretization of the space using particular rules of cutting is not always possible. This points is *a priori* the greatest weakness of the structured meshes. For instance, a sphere cannot be meshed with Cartesian structured mesh without exaggerated distortions.

- The mesh generated with a given discretization has to be stored or has to be easy to deduce. Simple structured meshes, such as Cartesian grids, are easy to create and store. In this case, the generation of the mesh is a bijection from the space step and the number of cell in each directions, and the coordinates of a first point. Hence, the storage of a Cartesian mesh can be reduced to some values. Conversely, unstructured meshes are often uneasy to store and operate on, as the position of a cell and its sub elements have to be stored, as well as the connectivities.

Thereby, the structured meshes are generally easy to generate and offer an accurate and simple discretization of spatial operators but cannot accurately approximate complex domains. With unstructured grids, the discretization of the spatial operators is more complex and often produces a less accurate result for a given size of element. However such grids are able to discretize complex domains and thus to treat industrial problems. Furthermore, the loss of accuracy can be counterbalanced by an adapted mesh refinement. Conversely, the ability of structured grids to be refined in the critical parts of the domain (boundary layers, interfaces,...) is very limited. One may distinguish two different needs of refinement. A first one is the consequence of the non-conformity between the boundaries of the domain or an immersed interface and the Eulerian mesh. The aim of the fictitious domain method presented in the present work is to treat this non-conformity. The second need of refinement is due to the general irregularity of the solution and its gradient. Ideally, the mesh has to be refined according to the magnitude of the gradients of the solution. This last point can be treated on structured grids with Adaptive Mesh Refinement methods (AMR) [Dela 06] but they need a great implementation effort and lead to a locally structured grid but a globally unstructured grid.

An important point is that an additional problem occurs when the simulation implies moving objects and moving interfaces. In this case, an initially well-adapted unstructured mesh is no more relevant at the next time steps. In case of multiphase flows, it happens almost systematically. The mesh can be reconstructed at each time step, but such a procedure is very expensive in computational time and hard to implement.

As a conclusion, the adaptivity of unstructured grids is limited in practical terms by the movements of the discontinuities of the solution and the physical quantities. Its initial advantage on the structured grids concerning the fixed boundaries is counterbalanced by the use of fictitious domain methods. Next to it, the interest of the fictitious domain methods is extended to the unstructured grids when a moving obstacles is involved.

1.2 The fictitious domains

1.2.1 Immersed boundaries and interfaces

Immersed boundary problems Let us consider the following model Dirichlet problem:

$$\begin{cases} -\nabla \cdot (\mathbf{a} \nabla u) = f & \text{in } \Omega_0 \\ u = u_{|\Sigma} & \text{on } \Sigma \end{cases} \tag{1.3}$$

and the model Neumann problem:

$$\begin{cases} -\nabla \cdot (\mathbf{a} \nabla u) = f & \text{in } \Omega_0 \\ (\mathbf{a} \cdot \nabla u).\mathbf{n} = g & \text{on } \Sigma \end{cases} \tag{1.4}$$

As they only consider one side of the interface $\Sigma = \partial\Omega_0$, both problems are qualified as boundary problems. Let us now consider a domain discretized with a Cartesian structured grid. As no particular constraint is imposed on the shape of Σ (only being continuous), one can suppose that Σ is not conform to the Cartesian grid. The computational domain is denoted as Ω, and the *supplementary* domain is Ω_1 such as $\Omega = \Omega_0 \cup \Sigma \cup \Omega_1$. The present problems are called immersed boundary problems. Even if the problem is numerically solved in Ω_0 and Ω_1, the solution is only required in the first one. Hence, the initial equations can be partially or totally removed in Ω_1. Many methods change the discretization in Ω_1 to increase the accuracy in Ω_0.

Immersed interface problems Let us consider the following interface problem:

$$\begin{cases} -\nabla \cdot (\mathbf{a} \nabla u) = f & \text{in } \Omega \\ u^-_{|\Sigma} = u_D & \text{on } \Sigma \\ u^+_{|\Sigma} = u_G & \text{on } \Sigma \end{cases} \tag{1.5}$$

The solution is now required in the whole numerical domain. However, the solutions on both sides of the interface are independent, and two immersed boundary sub-problems in Ω_0 and Ω_1 can be solved independently. We consider now an other problem:

$$\begin{cases} -\nabla \cdot (\mathbf{a} \nabla u) = f & \text{in } \Omega \\ + \text{ Interface condition} & \text{on } \Sigma \end{cases} \tag{1.6}$$

where the interface conditions are :

$$[\![u]\!]_\Sigma = \varphi \quad \text{on } \Sigma \tag{1.7}$$

$$[\![(\mathbf{a} \cdot \nabla u) \cdot \mathbf{n}]\!]_\Sigma = \psi \quad \text{on } \Sigma \tag{1.8}$$

The notation $[\![u]\!]_\Sigma = (u^+ - u^-)$ denotes the jump of a quantity over the interface Σ. These transmission and jump conditions are called the *immersed interface conditions* as they are used for the IIM methods[1] (see section 4.1).

[1] According to [Li 06] a method cannot be designated as an IIM if an other set of interface conditions is used.

A more general formulation has been proposed by Angot[Ango 03, Ango 05]:

$$[\![\mathbf{a}\cdot\nabla u)\cdot\mathbf{n}]\!]_\Sigma = \alpha\overline{u}_{|\Sigma} - h \quad \text{on } \Sigma \tag{1.9}$$

$$\overline{(\mathbf{a}\cdot\nabla u)\cdot\mathbf{n}}_\Sigma = \beta[\![u]\!] - g \quad \text{on } \Sigma \tag{1.10}$$

where $\overline{u}_\Sigma = (u^+ - u^-)/2$ denotes the arithmetic mean of the traces of a quantity on both sides of the interface, and α, β, h and g scalar values which can be chosen to obtain various types of immersed BC such as Dirichlet, Neumann, Stefan or Fourier conditions on the immersed interface. These conditions are called the *embedded boundary conditions*

1.2.2 The fictitious domain approach

Let us consider a mesh $\mathcal{T}_h$ composed of elements K_i. The previous problems, *e.g.* (2.11) or (1.6) are considered. Ideally, we have

$$\Omega_0 = \bigcup_{i\in\mathcal{I}} K_i \tag{1.11}$$

with $\mathcal{I}$ the set of indices of the elements of $\mathcal{T}_h$. One can find two sets of indices $\mathcal{I}_0$ and $\mathcal{I}_1$ such as

$$\Omega_0 = \bigcup_{i\in\mathcal{I}_0} K_i \tag{1.12}$$

and

$$\Omega_1 = \bigcup_{i\in\mathcal{I}_1} K_i. \tag{1.13}$$

In this case, $\mathcal{I}_0 \cup \mathcal{I}_1 = \emptyset$. Generally, the elements of the meshes are polygons, and except if the boundaries of the original domains are piecewise linear, the conditions (1.11), (1.12) and (1.13) cannot be fulfilled. If an unstructured mesh is used, the discrete boundary $\partial\Omega_h$ of the discretization Ω_h of Ω and the discretization Σ_h of the interface Σ can be approximated with a second-order piecewise reconstruction (see Fig. 1.2 right). In this case, the nodes of the discrete interfaces are generally on the original interfaces and no particular attention is paid to correct this approximation.

For structured meshes, the spatial approximation of the boundary is generally of first order only (see Fig. 1.2 left). Due to the constraints of the meshing, the spatial discretization does not take the shapes of the interfaces and boundaries into account. The only requirement is $\Omega \subset \Omega_h$. As the domains of the original problems are not "visible" on the mesh, such domains are qualified as *fictitious* when considered toward the mesh. Consequently, all the methods allowing such domains to exist during the numerical resolution are called *fictitious domain methods*.

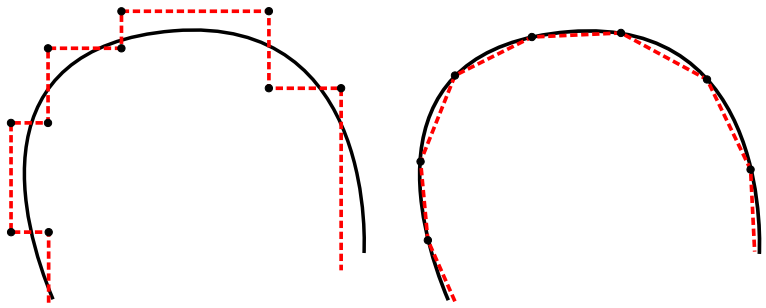


Figure 1.2: Approximation of a continuous interface. First (left) and second order (right)

Chapter 2

The penalty methods

THE penalty methods for conservation equations are a set of method originally presented in [Arqu 84, Calt 86] for the Navier-Stokes equations and in [Ango 89] for Poisson equation. The basic idea is to use a unique equation in the whole numerical domain which is divided in sub-domains of different kinds. A *penalty term* describing a *penalty constraint* (or a *penalty equation*) is added to the original physical equation. The penalty term is multiplied by a *penalty parameter* which may vary according to the subdomain. In the subdomains where the physical equation is relevant, the penalty parameter is equal to zero and the penalty term has no influence. In the subdomains where the penalty equation is relevant, the penalty term tends to infinity and the physical equation becomes negligible. The penalization of the solution is called the L^2 penalization, whereas the penalization of the gradient is called the H^1 penalization [Ango 99].

2.1 Darcy penalty method

The Darcy penalty method (DPM) was first presented in [Arqu 84, Calt 86] to treat porous media in fluid flows and extended to solid obstacles in [Ango 90, Khad 00]. The principle is to add a Darcy term $\frac{\mu}{K}\mathbf{u}$ into the Navier-Stokes equations. Hence, original equations become:

$$\begin{cases} \rho\left(\dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u}\right) + \dfrac{\mu}{K}\mathbf{u} = -\nabla p + \mu\Delta\mathbf{u} \\ \nabla\cdot\mathbf{u} = 0 \end{cases} \tag{2.1}$$

One can notice that contrary to some other fictitious domain methods, the penalty term is derived from the Darcy equation and then has a physical meaning. Practically, K is the local permeability and tends to infinity in the fluid media. A solid permeability $K \to 0$ is chosen in the solid media to impose $\mathbf{u} = 0$. Other values can be used to impose a porous media and an interest of the method lies in the possible modeling of fluid-porous flow interaction. Even if practically the behavior of the Darcy equation is obtained in this case [Calt 86, Laca 09], the presence of the convective and diffusive terms in the NS equations makes a formal difference between these equations.

If discretized such as the permeability K is piecewise constant per control volumes, the method reaches a first-order accuracy for the L^2 norm. Moreover, the DPM allows only static obstacles to be modeled as only a null velocity can be imposed.

2.2 Volume penalty method

The volume penalty method (VPM) was first presented in [Ango 89] and applied to heat transfers in electronic components. The penalty term is $Bi(T-T_\infty)$ and is derived from the Biot number so the penalty parameter is sometimes denoted as Bi. For an elliptic equation, the penalized problem is

$$\begin{cases} -\nabla\cdot(a\nabla u)+Bi(u-u_D)=f \text{ in } \Omega \\ \text{with } Bi_{|\Omega_0}=0, Bi_{|\Omega_1}=\frac{1}{\varepsilon}, \text{ for } 0<\varepsilon\ll 1 \end{cases} \tag{2.2}$$

where ε denotes the penalty parameter which tends to 0. As for the DPM, the VPM has a physical meaning when applied to heat transfers. The method can be applied to different equations but looses its physical meaning so the penalty parameter has to be almost zero or almost infinity (the physical and the non-physical penalty equations cannot be used at the same time for a same location). Applied to the Navier-Stokes equations, the VPM allows moving objects to be immersed. If $u_D=0$, the method is equivalent to the DPM. If the discretization considers a piecewise constant penalty term per control volume, the method is of first order too.

The present document shows applications of the method to the heat equation and to the incompressible Navier-Stokes equations. The method has been applied to large Mach number flows [Boir 09], the wave equation [Pacc 05] or to pseudo-spectral methods [Keet 07].

2.3 Implicit tensorial penalty method

This method is devoted to the resolution of the Navier-Stokes equations. It is based on a new formulation of the stress tensor $\boldsymbol{\sigma}$ which reads for a Newtonian fluid (see [Ryhm 85] and [Happ 63]):

$$\boldsymbol{\sigma}_{ij}=-p\,\delta_{ij}+\lambda\nabla\cdot\mathbf{u}\,\delta_{ij}+2\mu\,\mathbf{D}_{ij} \tag{2.3}$$

where λ et μ are respectively the compression and shearing viscosities and $\mathbf{D}$ is the tensor of deforming rate.
The introduction of the volumic viscosity ξ allows to highlight the spherical contribution of the viscous stresses. It can be expressed according to the compression and the shearing viscosities assuming $\xi=0$ for an imcompressible flow:

$$\begin{aligned} \xi &= \lambda+\frac{2\mu}{3} \\ \boldsymbol{\sigma}_{ij} &= -p\delta_{ij}-\frac{2}{3}\mu\nabla\cdot\mathbf{u}\delta_{ij}+2\mu D_{ij} \end{aligned}$$

Usually, this formulation of the stress tensor assumes that the fluid is homogeneous and the components of the stress tensor are null for a constant flow or a uniform rotation flow $\mathbf{u}=\boldsymbol{\Omega}\times\mathbf{r}$. Moreover, the components of $\boldsymbol{\sigma}_{ij}$ are linearly expressed according to velocity derivatives and are exactly equal and of opposite sign to the hydrostatic pressure when the fluid is at rest. This supposes that no direction is favoured in the fluid. In this way, the velocity and the corresponding viscous stresses are bind through an isotropic relation.
Starting from the Navier-Stokes equations in their conservative and compressible formulation dedicated to newtonian fluids, we can write:

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{u}) = 0 \tag{2.4}$$

$$\frac{\partial\rho\mathbf{u}}{\partial t}+\nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) = \rho\,\mathbf{g}+\nabla\cdot\boldsymbol{\sigma} \tag{2.5}$$

The aim is here to reformulate the problem so as to make appear several natural contributions of the stress tensor dealing with compression, tearing, shearing and rotation. The interest of this decomposition is then to distinctly penalize each term in order to strongly impose the associated stress (see[Calt 01]). If we assume that Navier-Stokes equations for a Newtonian fluid contain all physical contributions traducing compressibility effects, shearing or rotation, their splitting permits to act differentially on their effects by modifying the orders of magnitude of each term directly in the motion equations.
We first break up the second-order tensor ∇_{ij}, which corresponds to the gradient of a vectorial variable, in a symmetrical part D_{ij} and in an antisymmetrical part Ω_{ij} (Ryhming [Ryhm 85])

$$\nabla_{ij} = \frac{1}{2}\left(\nabla_{ij} + \nabla_{ji}\right) + \frac{1}{2}\left(\nabla_{ij} - \nabla_{ji}\right) = D_{ij} + \Omega_{ij} \tag{2.6}$$

Then the stress tensor (2.3) can be rewritten as follows

$$\boldsymbol{\sigma}_{ij} = -p\,\delta_{ij} + \lambda\,\nabla\cdot\mathbf{u}\,\delta_{ij} + 2\,\mu D_{ij} = -p\,\delta_{ij} + \lambda\,\nabla\cdot\mathbf{u}\,\delta_{ij} + 2\,\mu\,\left(\nabla_{ij} - \Omega_{ij}\right)$$

Decomposing $\boldsymbol{\sigma}_{ij}$ according to the partial derivative of the velocity in Cartesian coordinates

$$\begin{aligned}
\boldsymbol{\sigma} \;=\; & \begin{bmatrix} -p+\lambda\,\nabla\cdot\mathbf{u} & 0 & 0 \\ 0 & -p+\lambda\,\nabla\cdot\mathbf{u} & 0 \\ 0 & 0 & -p+\lambda\,\nabla\cdot\mathbf{u} \end{bmatrix} + \kappa \begin{bmatrix} \frac{\partial u}{\partial x} & 0 & 0 \\ 0 & \frac{\partial v}{\partial y} & 0 \\ 0 & 0 & \frac{\partial w}{\partial z} \end{bmatrix} \\
+ \;\; & \zeta \begin{bmatrix} 0 & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & 0 & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & 0 \end{bmatrix} - \eta \begin{bmatrix} 0 & \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} & \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} & 0 & \frac{\partial v}{\partial z} - \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} - \frac{\partial u}{\partial z} & \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} & 0 \end{bmatrix}
\end{aligned} \tag{2.7}$$

we thus obtain an original decomposition of the stress tensor in which new viscosity coefficients appear artificially

$$\boldsymbol{\sigma}_{ij} = \left(-p + \lambda\,\nabla\cdot\mathbf{u}\right)\delta_{ij} + \kappa\,\Lambda_{ij} + \zeta\,\Theta_{ij} - \eta\,\Gamma_{ij} \tag{2.8}$$

where

λ is the compression viscosity

κ is the tearing viscosity

ζ is the shearing viscosity

η is the rotation viscosity

The usual form of $\boldsymbol{\sigma}$ can be recovered allocating the following values to the new viscosities:

$$\lambda = -2/3\mu,\ \kappa = 2\mu,\ \zeta = 2\mu,\ \eta = \mu. \tag{2.9}$$

Several values of the viscosity coefficients appearing in the new formulation (2.8) can be verified experimentally or theoretically in the works of Bird et al [Bird 77] or Ryhming [Ryhm 85].
Four different terms appear in the divergence of the stress tensor $\boldsymbol{\sigma}$, corresponding to the compression tensor $\nabla\cdot\mathbf{u}$, the tearing tensor $\boldsymbol{\Lambda}$, the pure shearing tensor $\boldsymbol{\Theta}$ and the rotation tensor $\boldsymbol{\Gamma}$, which are associated to four characteristic phenomena of a flow:

$$\nabla\cdot\boldsymbol{\sigma} = -\nabla\left(p - \lambda\nabla\cdot\mathbf{u}\right) + \nabla\cdot\left(\kappa\,\boldsymbol{\Lambda}\right) + \nabla\cdot\left(\zeta\,\boldsymbol{\Theta}\right) - \nabla\cdot\left(\eta\,\boldsymbol{\Gamma}\right) \tag{2.10}$$

The main interest of the formulation (2.10) is to dissociate stresses operating in a Newtonian viscous flow and then to make the implementation of a numerical penalty method easier. The use of the viscosities λ, κ, ζ et η permits to satisfy accurately each kind of stress for both compressible and incompressible flows.
The new decomposition of the viscous stress tensor must be integrated in the energy equation for a coherent formulation. The terms $-\frac{\beta}{\chi T}T\nabla\cdot\mathbf{u}+\mu\Phi(\mathbf{u})$ are replaced by the generic tensor $\boldsymbol{\sigma}:\nabla\mathbf{u}$ written according to the previous theory.

This method has been applied to particle sedimentation [Pian 05, Rand 05, Vinc 07] and coupled to phase change [Maun 08].

2.4 Jump Embedded Boundray Condition methods

Angot and Ramière have proposed two methods to deal with various type of BC such as Dirichlet, Neumann, Stefan or Robin.

In [Rami 07b], the authors use a spread interface representation. We consider a closed domain $\tilde{\Omega}$ of boundary Σ embedded in a domain Ω. The shape of Ω is supposed to be simple. We define the complementary domain Ω_e such as $\Omega=\tilde{\Omega}\cup\omega_\Sigma\cup\Omega_e$ (see Fig. (2.1) left). The following problems is considered:

$$\begin{cases} -\nabla\cdot(\tilde{\mathbf{a}}\nabla u)+\tilde{b}u=\tilde{f} & \text{in } \tilde{\Omega} \\ BC & \text{on } \Sigma \end{cases} \tag{2.11}$$

The discretization $\tilde{\Omega}_h$ of $\tilde{\Omega}$ is embedded in the computational domain Ω_h such that $\Omega_h=\tilde{\Omega}_h\cup\omega_{h,\Sigma}\cup\Omega_{e,h}$. If a cell K is cut by the interface Σ, then $K\in\omega_{h,\Sigma}$. If $K\subset\tilde{\Omega}$, then $K\subset\tilde{\Omega}_h$. The other cells are in $\Omega_{e,h}$ (see Fig. 2.1.right). The following equation is considered in Ω:

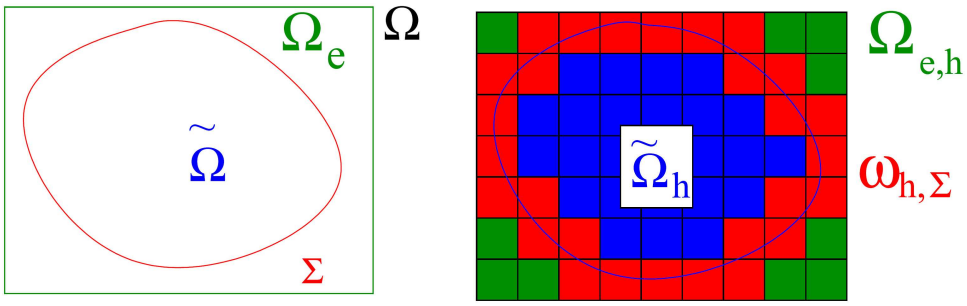


Figure 2.1: Definition of the domains for the spread interface method [Rami 07b]

$$-\nabla\cdot(\mathbf{a}\nabla u)+bu=f \tag{2.12}$$

and the various parameters $\mathbf{a}$, b and f are chosen according to the desired BC (see Tab. 2.1). The constants α_R and g_R are related to the Robin BC. The parameter ϵ_h is a local correcting factor which depends on the cell size.

The Neumann BC can be retrieved with $\alpha_R\equiv 0$. Concerning the Dirichlet BC, two methods are considered depending on the location of the penalty terms (exterior domain $\Omega_{e,h}$ or spread interface $\omega_{h,\Sigma}$).

	Parameters in $\Omega_{h,\Sigma}$	Parameters in $\Omega_{e,h}$
Dirichlet BC Spread interface penalization	$\mathbf{a} = \begin{cases} \tilde{\mathbf{a}}\,(L^2\,penalty) \\ \dfrac{1}{\eta}\mathbb{I}_d\ (H^1\,penalty) \end{cases}$, $b = \dfrac{1}{\eta},\ f = \dfrac{1}{\eta}u_D$	$\mathbf{a} = \mathbb{I}_d,$ $b = 0,\ f = 0$
Dirichlet BC Exterior penalization	$\mathbf{a} = \tilde{\mathbf{a}},$ $b = \tilde{b},\ f = \tilde{f}$	$\mathbf{a} = \begin{cases} \mathbb{I}_d\,(L^2\,penalty) \\ \dfrac{1}{\eta}\mathbb{I}_d\,(H^1\,penalty) \end{cases}$, $b = \dfrac{1}{\eta},\ f = \dfrac{1}{\eta}u_D$
Robin BC with different approximations of ϵ_h	$\mathbf{a} = \tilde{\mathbf{a}},$ $b = \tilde{b} + \dfrac{\alpha_R}{\epsilon_h},\ f = \tilde{f} - \dfrac{g_R}{\epsilon_h}$	$\mathbf{a} = \eta\mathbb{I}_d,$ $b = 0,\ f = 0$

Table 2.1: Parameters in $\omega_{h,\Sigma}$ and $\Omega_{e,h}$ for [Rami 07b]

A second method has been developed in [Rami 07c] which is a sharp interface approach combined with the JEBC of [Ango 03, Ango 05]:

$$[\![\mathbf{a}\cdot\nabla u)\cdot\mathbf{n}]\!]_\Sigma = \alpha\overline{u}_{|\Sigma} - h \quad \text{on } \Sigma \tag{2.13}$$

$$\overline{(\mathbf{a}\cdot\nabla u)\cdot\mathbf{n}}_\Sigma = \beta[\![u]\!] - g \quad \text{on } \Sigma \tag{2.14}$$

The different parameters $\mathbf{a}$, b and f in Ω_e and on Σ are chosen according to Tab. 2.2 to imposed the desired interface condition.

Two approaches are considered for the Dirichlet BC. For the surface penalty, the condition is imposed with the interface parameters whereas the condition is imposed in Ω_e for the volume penalty. Since this approach considers a sharp interface, the parameters are defined on Σ instead of in $\omega_{h,\Sigma}$. This approach is more difficult to implement than for the spread interface approach since the penalization on Σ requires a modification of the discretization of the spatial operators. The interface actually taken into account is a stairstep reconstruction from the faces of the cells crossed by the interfaces (however, the correcting factor ϵ considers a piecewise linear reconstruction of Σ). Hence, this method, as well as the first one have only a first order of spatial accuracy for the L^2 norm and a 1/2 order for the H^1 norm [Rami 07a]. The JEBC (2.13)-(2.14) can be also generalized for vector elliptic problems such as Stokes-Brinkman problems.

	Parameters in Ω_e	Parameters in Σ
Dirichlet BC Surface penalty	$\mathbf{a}_{\|\Omega_e} = 1,\ \mathbf{v}_{\|\Omega_e} = 0,$ $b_{\|\Omega_e} = f_{\|\Omega_e} = 0$	$\alpha = 4\beta = \frac{2}{\eta},$ $\frac{q}{2} - g = \frac{1}{\eta} u_D$
Dirichlet BC Exterior penalization	$\mathbf{a} = \begin{cases} \mathbb{I}_d\,(L^2\,penalty) \\ \frac{1}{\eta}\mathbb{I}_d\,(H^1\,penalty) \end{cases},\ \mathbf{v}_{\|\Omega_e} = 0,$ $b_{\|\Omega_e} = \frac{1}{\eta},\ f_{\|\Omega_e} = \frac{1}{\eta} u_e$	$\beta = \frac{1}{\eta},$ $\alpha = q = g = 0$
Robin BC no exterior control	$\mathbf{a}_{\|\Omega_e} = 1,\ \mathbf{v}_{\|\Omega_e} = 0,$ $b_{\|\Omega_e} = f_{\|\Omega_e} = 0$	$\alpha = 4\beta = 2\alpha_R,$ $g - \frac{q}{2} = g_R$

Table 2.2: Parameters in Ω_e and on Σ for [Rami 07a]

Chapter 3

Other methods for immersed boundary problems

IMMERSED boundary method is a generic appellation for the methods dealing with immersed obstacles. However, this term often specifically designates the method designed by Charles S. Peskin in 1972. In the present document, the acronym IBM will always denotes the method itself while the immersed boundary problems will be designated as IB problems. The IBM Direct-forcing (DF) approaches are sometimes totaly different from the original Peskin method but are also designated as a particular class of the IBM. The common principle of the IBM is to use a forcing term and a Cartesian grid. A good review can be found in [Mitt 05].

3.1 The immersed boundary method

3.1.1 Continuous forcing IBM

In [Pesk 72], Peskin simulated blood flows in beating heart valves. The method was originally designed for moving flexible boundaries and has been adapted later to rigid boundaries. The principal idea is to use forces to account for the presence of immersed boundaries or interfaces. As the immersed Lagrangian mesh discretizing interfaces does not generally match the Eulerian grid, the interface forces are spread from the Lagrangian point to the neighbor Eulerian nodes using discrete Dirac functions, or *Peskin functions*. The discrete Dirac function can be discretized as triangle, trigonometric or more complex functions, such as:

$$\delta_h = \begin{cases} \frac{1}{8\Delta x}(3 - 2|r|/h + \sqrt{1 + 4|r|/h - 4(|r|/h)^2} & |r| \leq \Delta x \\ \frac{1}{8\Delta x}(5 - 2|r|/h + \sqrt{-7 + 12|r|/h - 4(|r|/h)^2} & \Delta x \leq |r| \leq 2\Delta x \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

This approach is quite simple to implement and robust, and has been used to treat many applied problems, especially in biomechanics [Mill 05, Hopp 02]. However, the method is generally first-order accurate in space only due to the smooth representation of the interface. Higher orders can only be reached for the particular case of sufficiently smooth problems [Lai 00, Grif 05].

A formulation of the Peskin approach for the Navier-Stokes equations can be written as

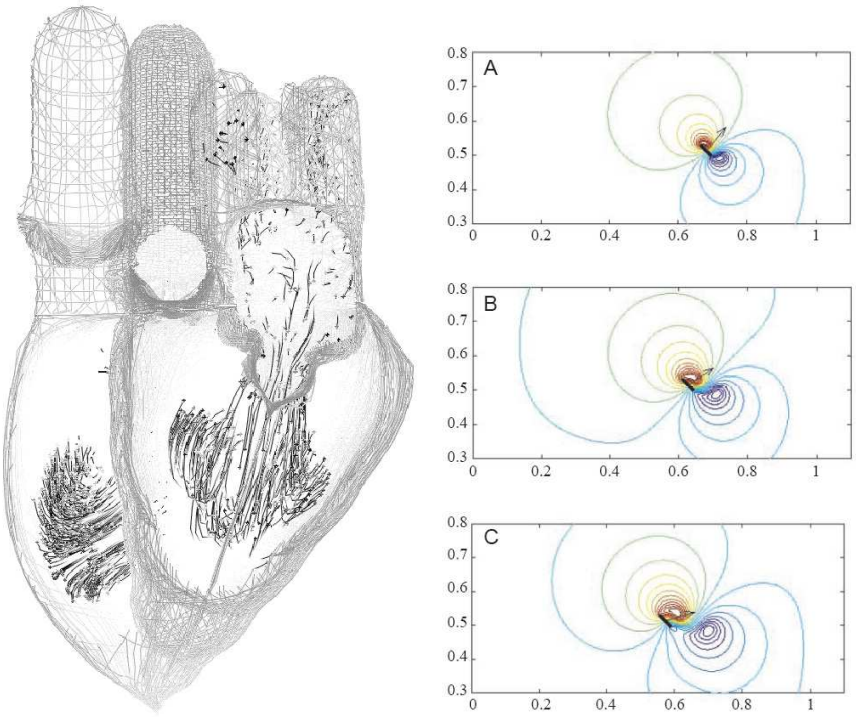


Figure 3.1: Simulation of heart flows using the Peskin immersed boundary method - Study of vortices near insect wings

follows:

$$\rho\Big(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}.\nabla)\,\mathbf{u}\Big) = -\nabla p + \mu\nabla^2\mathbf{u} + \mathbf{f} \text{ in } \Omega \tag{3.2}$$

$$\nabla.\mathbf{u} = 0 \text{ in } \Omega \tag{3.3}$$

$$\mathbf{f}(\mathbf{x},t) = \int_0^{L_b} \mathbf{F}(s,t)\delta(\mathbf{x} - \mathbf{X}(s,t))\,\mathrm{d}s \tag{3.4}$$

$$\frac{\partial \mathbf{X}(s,t)}{\partial t} = \mathbf{u}(\mathbf{X}(s,t),t) = \int_\Omega \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(s,t))\,\mathrm{d}\mathbf{x} \tag{3.5}$$

$$\mathbf{F}(s,t) = \mathbf{S}(\mathbf{X}(\cdot,t),t) \tag{3.6}$$

Here, the immersed boundary Σ is given in parametric form: $\mathbf{X}(s,t), 0 \leq s \leq L_b$, $\mathbf{X}(0,t) = \mathbf{X}(L_b,t)$, where s tracks a material point of Σ. The force density (with respect to $\mathrm{d}\mathbf{x} = \mathrm{d}x\,\mathrm{d}y$) acting on the fluid is $\mathbf{f}(\mathbf{x},t)$ while the boundary force density (with respect to $\mathrm{d}s$) is $\mathbf{F}(s,t)$. Eqs. (3.2)-(3.3) are the Navier-Stokes equations for a viscous incompressible fluid, Eqs. (3.4)-(3.5) represent the interaction between the immersed boundary and the fluid. In Eq. (3.4), the force density is applied to the fluid by the immersed boundary. In Eq. (3.5), the immersed boundary is carried along with the fluid. The last Eq. (3.6) states that the boundary force on a particular interface element at time t is determined by the boundary configuration at time t, where the function $\mathbf{S}$ satisfies the generalized Hooke's law if the boundary is elastic.

This formulation is very attractive for flows with elastic boundaries and can be easily adapted to rigid boundaries by considering the boundary as elastic but extremely stiff. However, this approach implies many stability constraints [Mitt 05] while lower values of stiffness lead to spurious elastic effects [Lai 00].

3.1.2 Direct-forcing IBM

In these methods, the physical forcing term of the Peskin approach is replaced by a non-physical term. The numerical instabilities encountered with the Peskin method when a rigid boundary is considered are no more present. The first direct forcing IBM has been presented in [Mohd 97] for pseudo-spectral elements and extended in [Fadl 00] to a 3D FD method on a Cartesian staggered grid. The following Navier-Stokes momentum equation is solved:

$$\rho\Big(\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t}+\left(\mathbf{u}^{n+1}.\nabla\right)\mathbf{u}^{n+1}\Big)=-\nabla p^{n}+\mu\nabla^{2}\mathbf{u}^{n+1}+\delta(\mathbf{x}-\mathbf{X}(s,t))\mathbf{f} \tag{3.7}$$

Contrary to the Peskin approach, the forcing term is not physical and there is no constant to set:

$$\mathbf{f}=\rho\Big(\frac{\mathbf{V}-\mathbf{u}^{n}}{\Delta}+\left(\mathbf{u}^{n+1}.\nabla\right)\mathbf{u}^{n+1}\Big)+\nabla p^{n}-\mu\nabla^{2}\mathbf{u}^{n+1} \tag{3.8}$$

where $\mathbf{V}$ is typically the Dirichlet value. The discrete Dirac function $\delta(\mathbf{x}-\mathbf{X}(s,t))$ forcing term is only activated for nodes near the boundary. The implementation of this method in Thétis shown that as the forcing term is not physical, it is hazardous to mix it with the a physical equation for a same node (a factor has to be determined). Hence, the δ is generally not smooth so its value is binary. If $\mathbf{V}$ is the desired velocity at the boundary, these approaches have a first-order accuracy in space. In [Fadl 00, Tsen 03], $\mathbf{V}$ is no more a constant but a linear combination of the solution near the interface, and $\mathbf{V}=\sum\omega_i u_i$ with w_i weighting coefficients depending on the interpolation. [Fadl 00] activates the forcing term in the fluid region while [Tsen 03, Mark 08] uses the nodes in the solid region. The advantage of this last approach is to conserve the original equation one node closer to the interface, increasing the accuracy of the simulation. The direct forcing approach has been widely applied to turbulent flows at the Center for Turbulence Research, Stanford (see, *e.g.* [Verz 01, Maju 01]). Concerning the velocity-pressure coupling, these methods are generally used with a pressure projection method. As will be demonstrated later [Iken 07, Dome 08], the standard projection is not well-suited to the forcing term.

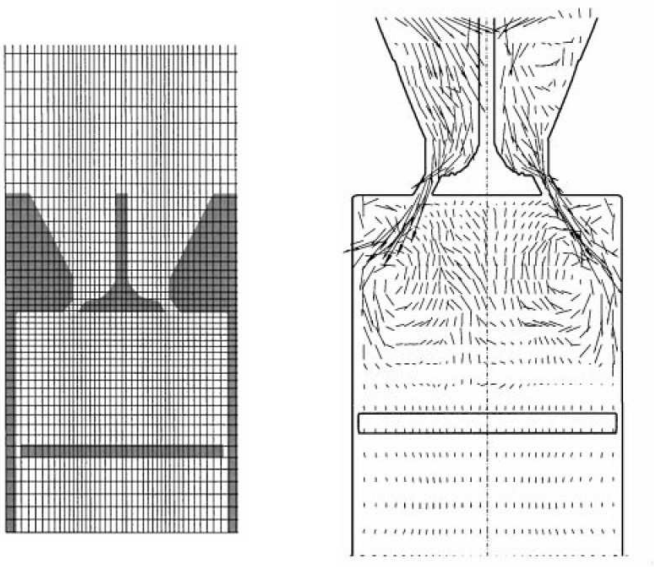


Figure 3.2: Flow simulation in a non-conforming piston [Fadl 00]

3.2 The Cartesian grid FV methods

The aim of these methods is to strictly keep the conservation laws at the close vicinity of the interface. The idea is to reshape the cells crossed by the interface and to build *ad hoc* FV schemes in them. The interface is approximated as a line or a plane in each cut cell.

3.2.1 The cut-cell method

First designed to treat inviscid flows around airfoils [Clar 86], the Cut-cell approach modifies the shape of the control volumes near the interface by cutting and merging them. Quantities such as mass, convective and diffusive flux integrals and pressure gradients have to be estimated on each face of these new cells. Ye et al. [Ye 99] proposed to express a given flow variable in terms of polynomial interpolating function in an appropriate region and evaluate the fluxes based on this interpolating functions. In this approach, the solution inside the obstacle is not used. Fig. 3.3 from [Ye 99] shows how the cells are cut and merged to define new cells which are conform to the immersed interface.

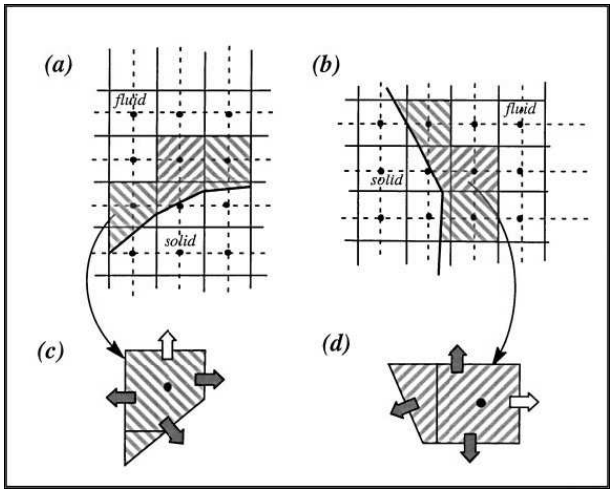


Figure 3.3: Cutting of the CV with the cut cell method and associated new face fluxes[Ye 99]

3.2.2 The embedded boundary method

First presented by Johansen and Colella [Joha 98], the Cartesian Grid Embedded Boundary method is based on a FV formulation and a modification of the control volumes to take into account the immersed interface. Contrary to the Cut-cell method, the solution is computed at the center of the cells, even the ones out of the physical domain. On the other hand, the fluxes are calculated at the new faces of the mesh, *i. e.* the position of the immersed interface is taken into account. Initially presented for the Poisson equation, the method has been used to solve the heat equation in 2D [Schw 06] and 3D [McCo 01]. An application to vessel segmentation and blood flow simulation can be found in [Desc 04].

3.3 The distributed Lagrange multipliers

The Distributed Lagrange Multipliers method has been formulated by Glowinski and Pan to simulate the sedimentation of a large amount of particles in 2D [Glow 99] and 3D [Glow 01]

(see Fig. 3.4) and has been applied to particle motion in non-newtonian flows [Sing 00, Hao 09] or to ellipsoidal particles [Pan 02, Pan 06]. The method uses Lagrange multipliers to enforce solid-body motion in fluid in order to simulate the presence of immersed objects.

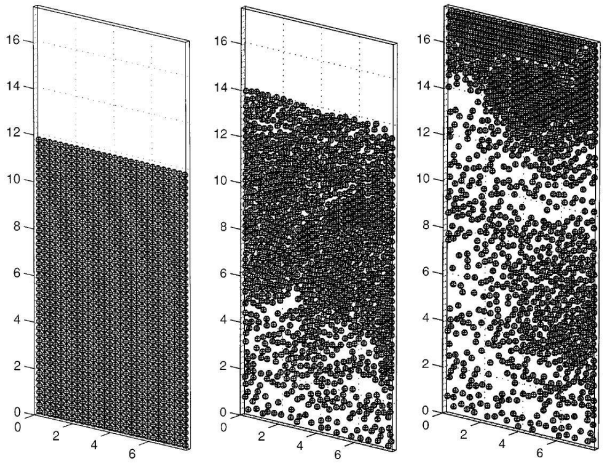


Figure 3.4: Fluidization of 1024 spherical particles

The method is presented for an immersed particle represented by a domain Ω_1. The principle of the DLM method is to couple the fluid equations:

$$\rho_L\left(\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}\right) = \rho_L\mathbf{g} + \nabla\cdot\boldsymbol{\sigma} \text{ in } \Omega_0 \tag{3.9}$$

$$\nabla.\mathbf{u} = 0 \text{ in } \Omega_0 \tag{3.10}$$

$$\tag{3.11}$$

and the rigid body equations:

$$m\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{F} \tag{3.12}$$

$$I\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} + \boldsymbol{\omega}\wedge I\boldsymbol{\omega} = \mathbf{M} \tag{3.13}$$

with ρ_L the fluid density, m the body mass, $\boldsymbol{\omega}$ its rotation vector, $\mathbf{F}$ the external resulting force applied to the solid, $\mathbf{M}$ the resulting torque. The force $\mathbf{F}$ is composed of physical forces and a repulsive force related to the other particles. The stress tensor $\boldsymbol{\sigma}$ takes the form:

$$\boldsymbol{\sigma} = -p\mathcal{I}_d + 2\mu\mathbf{D}(\mathbf{u}) \tag{3.14}$$

The velocity of a particle i is such that

$$\mathbf{u} = \mathbf{U}_i + \boldsymbol{\omega}\times\mathbf{r} \tag{3.15}$$

with $\mathbf{U}$ the translational velocity, and $\mathbf{r}$ the distance to the center of the concerned particles. Equation (3.15) is combined to the weak form of (3.9), (3.12) and (3.13):

$$\int_\Omega \rho\left(\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} - \mathbf{g}\right)\cdot\mathbf{v}\ \mathrm{d}\mathbf{x} + \left(1-\frac{\rho_L}{\rho_d}\right)\left(\mathbf{M}\left(\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} - \mathbf{g}\right)\cdot\mathbf{V} + \mathbf{I}\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t}\xi\right) - F\cdot V = -\int_\Omega \boldsymbol{\sigma} : \mathbf{D}(\mathbf{v})\ \mathrm{d}\mathbf{x} \tag{3.16}$$

with ρ_d the particle density and where $(\mathbf{v}, \mathbf{V}, \xi)$ is a combined variation from the *combined variation space*

$$\begin{aligned}\mathbb{V}_0(t) = \{(\mathbf{v}, \mathbf{V}, \xi) \,|\mathbf{v} \in H^1(\Omega)^2, \mathbf{V} \in \mathbb{R}^d, \xi \in \mathbb{R},\\ \mathbf{v} = \mathbf{V} + \xi \times \mathbf{r} \text{ in } \Omega_1, \text{ and } \mathbf{v} = 0 \text{ on } \partial\Omega\}\end{aligned} \tag{3.17}$$

while $(\mathbf{u}, \mathbf{U}, \omega)$ is taken in the *combined velocity space*

$$\begin{aligned}\mathbb{V}_{\mathbf{u}_\Sigma}(t) = \{(\mathbf{v}, \mathbf{V}, \xi) \,|\mathbf{v} \in H^1(\Omega)^2, \mathbf{V} \in \mathbb{R}^d, \xi \in \mathbb{R},\\ \mathbf{v} = \mathbf{V} + \xi \times \mathbf{r} \text{ in } \Omega_1, \text{ and } \mathbf{v} = \mathbf{u}_{\partial\Omega} \text{ on } \partial\Omega\}\end{aligned} \tag{3.18}$$

The fluid pressure p is required to lie in the space

$$L^2_0(\Omega_0) = \{q \in L^2(\Omega_0) | \int_{\Omega_0} q \, \mathrm{d}\mathbf{x} = 0\} \tag{3.19}$$

Concerning the equation (3.10), its weak form gives

$$\int_\Omega q \nabla \cdot \mathbf{u} \, \mathrm{d}\mathbf{x} = 0 \text{ for all } q \in L^2(\Omega) \tag{3.20}$$

The rigid-body motion constraint is enforced via the definition of the combined velocity spaces. Such a formulation is not well-adapted to the methods used to solve the standard Navier-Stokes equations. As for the augmented Lagrangian method (see section A.3.4), this constraint is relaxed thanks to a side constraint using an appropriate distributed Lagrange multiplier. The resulting formulation is the following one:
For a.e. $t > 0$, find $\mathbf{u} \in \mathbb{W}_{\mathbf{u}_\Sigma}$, $p \in L^2_0(\Omega)$, $\boldsymbol{\lambda} \in \Lambda(t)$, $\mathbf{U} \in \mathbb{R}$ satisfying

$$\int_\Omega \rho_L \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \mathbf{g} \right) \cdot \mathbf{v} \, \mathrm{d}\mathbf{x} - \int_\Omega p \nabla \cdot \mathbf{v} \, \mathrm{d}\mathbf{x} + \int_\Omega 2\eta \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) \, \mathrm{d}\mathbf{x} \tag{3.21}$$

$$+ \left(1 - \frac{\rho_L}{\rho_d}\right) \left(\mathbf{M} \left(\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} - \mathbf{g} \right) \cdot \mathbf{V} + \mathbf{I} \frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \xi \right) - F \cdot V = \langle \boldsymbol{\lambda}, \mathbf{v} - (\mathbf{V} + \xi \times \mathbf{r}) \rangle_{\Omega_1} \tag{3.22}$$

$$\text{for all } \mathbf{v} \in \mathbb{W}_0, \ \mathbf{V} \in \mathbb{R}^d, \text{ and } \xi \in \mathbb{R}, \tag{3.23}$$

$$\int_\Omega q \nabla \cdot \mathbf{u} \, \mathrm{d}\mathbf{x} = 0 \text{ for all } q \in L^2(\Omega) \tag{3.24}$$

$$\langle \boldsymbol{\mu}, \mathbf{u} - (\mathbf{U} + \boldsymbol{\omega} \times \mathbf{r}) \rangle_{\Omega_1} = 0 \text{ for all } \boldsymbol{\mu} \in \Lambda(t) \tag{3.25}$$

$$\tag{3.26}$$

where

$$\mathbb{W}_{\mathbf{u}_\Sigma} = \{\mathbf{v} \in H^1_0(\Omega)^2 | \mathbf{v} = \mathbf{u}_{\partial\Omega} \text{ on } \partial\Omega\} \tag{3.27}$$

$$\mathbb{W}_0 = H^1_0(\Omega)^2 \tag{3.28}$$

and $\Lambda(t)$ is $H^1(\Omega_1)^2$ with $\langle ., . \rangle$ an appropriate inner product. An alternative formulation consists in replacing the volume forcing term $\langle \boldsymbol{\lambda}, \mathbf{v} - (\mathbf{V} + \xi \times \mathbf{r}) \rangle_{\Omega_1}$ by a surface forcing term $\langle \boldsymbol{\lambda}, \mathbf{v} - (\mathbf{V} + \xi \times \mathbf{r}) \rangle_{\partial\Omega_1}$

3.4 The Boundary Ghost Fluid method

In [Gibo 02, Gibo 05], Gibou *et al.* uses the same principles as the Ghost Fluid method (see section 4.2.1) to impose boundary conditions for elliptic equations.

The 1D principle is described. The model Dirichlet problem is solved in a domain Ω_0 of boundary $\partial\Omega$:

$$\begin{cases} -\nabla\cdot(a\nabla u) = f & \text{in } \Omega_0 \\ u = u_{|\partial\Omega} & \text{on } \partial\Omega_0 \end{cases} \tag{3.29}$$

Let us consider the location of three regular grid nodes x_{i-1}, x_i and x_{i+1}. The interface location is α such as $x_{i-1} < x_i < \alpha < x_{i+1}$. We choose $x_{i+1} \notin \Omega_0$ so the solution at this point is written u^G_{i+1}. It does not exist in the initial problem and is then denoted as a *ghost node* with the superscript G.

The standard three-point discretization of the Laplacian yields

$$\{\nabla\cdot(a\nabla u)\}_i = \frac{a_{i+\frac{1}{2}}\left(\dfrac{u^G_{i+1}-u_i}{\Delta x}\right) - a_{i-\frac{1}{2}}\left(\dfrac{u_i-u_{i-1}}{\Delta x}\right)}{\Delta x}. \tag{3.30}$$

The value u^G_{i+1} is then considered as the linear extrapolation of the solution past the interface. One can write

$$u_{|\Gamma} = \frac{(\alpha-x_i)u^G_{i+1} + (x_{i+1}-\alpha)u_i}{\Delta x} \Leftrightarrow u^G_{i+1} = \frac{\Delta x}{\alpha-x_i}u_{|\Gamma} - \frac{x_{i+1}-\alpha}{\alpha-x_i}u_i. \tag{3.31}$$

Finally, the following discretization of the operator near the interface is obtained:

$$\{\nabla\cdot(a\nabla u)\}_i = \frac{1}{\Delta x}\left(a_{i+\frac{1}{2}}\left(\frac{\frac{\Delta x}{\alpha-x_i}u_{|\Gamma} - \left(\dfrac{x_{i+1}-\alpha}{\alpha-x_i}-1\right)u_i}{\Delta x}\right) - a_{i-\frac{1}{2}}\left(\frac{u_i-u_{i-1}}{\Delta x}\right)\right). \tag{3.32}$$

The same principle is applied direction by direction in higher dimensions. As showed latter, the discretized operators obtained with this method is quite similar to the SMP and AIIB methods for Dirichlet problems. However, the two latter methods do not require a by-hand re-discretization of the operators as this modification is performed algebraically.

Chapter 4

Other methods for immersed interface problems

4.1 The immersed interface method

4.1.1 Standard approach

The original IIM has been proposed by Leveque and Li in [Leve 94, Li 94]. Let us consider the following 1D problem:

$$\begin{cases} -\nabla \cdot (\mathbf{a}\nabla u) = f & \text{in } \Omega \\ [\![u]\!]_\Sigma = \varphi & \text{on } \Sigma \\ [\![(\mathbf{a}\cdot\nabla u)\cdot\mathbf{n}]\!]_\Sigma = \psi & \text{on } \Sigma \end{cases} \tag{4.1}$$

The equation is discretized using a three-point finite difference scheme. The criterion in determining the finite difference coefficients is to minimize the magnitude of the local truncation error for a given location x_i

$$T_i = \gamma_{i,1}u(x_{i-1}) + \gamma_{i,2}u(x_i) + \gamma_{i,3}u(x_{i+1}) - f(x_i) + C_i \tag{4.2}$$

where $\gamma_{i,k}$ are linear coefficients and C_j a source term. The main idea is to expand the solution $u(x_{i-1})$, $u(x_i)$, $u(x_{i+1})$, $f(x_i)$ at the interface for $x = \alpha$ from each side of the interface and then use the interface relations to express $u^{\pm}(\alpha)$, $u_x^{\pm}(\alpha)$ and $u_{xx}^{\pm}(\alpha)$ in terms of quantities from one particular side. The three locations x_{i-1}, x_i and x_{i+1} are such that $x_{i-1} < x_i < \alpha < x_{i+1}$. Finally, we match the expansion against the differential equation to the leading terms to get a system of equations for the finite difference coefficients. Using the Taylor expansion for $u(x_{i+1})$ at α, we have

$$u(x_{i+1}) = u^+(\alpha) + (x_{i+1}-\alpha)u_x^+(\alpha) + \frac{1}{2}(x_{i+1}-\alpha)^2 u_{xx}^+(\alpha) + \mathcal{O}(h^3). \tag{4.3}$$

Using the jump relations, the expression above can be written as

$$u(x_{i+1}) = u^-(\alpha) + \varphi + (x_{i+1}-\alpha)\left(\frac{a^-}{a^+}u_x^-(\alpha) + \frac{\psi}{a^+}\right) + \frac{1}{2}(x_{i+1}-\alpha)^2\frac{a^-}{a^+}u_{xx}^-(\alpha) + \mathcal{O}(h^3) \tag{4.4}$$

The Taylor expansion is then written for u_{i-1} and u_i. Thus, the local truncation error can be written at $x = x_i$

$$\begin{aligned} T_i &= \gamma_{i,1}u(x_{i-1}) + \gamma_{i,2}u(x_i) + \gamma_{i,3}u(x_{i+1}) - f(x_i) + C_i \\ &= \varphi + (\gamma_{i,1} + \gamma_{i,2} + \gamma_{i,3})u^-(\alpha) + \gamma_{i,3}(x_{i+1} - \alpha)\frac{\psi}{a^+} \\ &\quad + \left((x_{i-1} - \alpha)\gamma_{i,1} + (x_i - \alpha)\gamma_{i,2} + \frac{\beta^-}{\beta^+}(x_{i-1} - \alpha)\gamma_{i,3}\right) u_x^-(\alpha) \\ &\quad + \frac{1}{2}\left((x_{i-1} - \alpha)^2\gamma_{i,1} + (x_i - \alpha)^2\gamma_{i,2} + \frac{a^-}{a^+}(x_{i-1} - \alpha)^2\gamma_{i,3}\right) u_{xx}^-(\alpha) \\ &\quad - f(\alpha) - \mathcal{O}(h) + \mathcal{O}\left(\max_{1\le l\le 3}|\gamma_{i,l}|h^3\right). \end{aligned} \tag{4.5}$$

The system of equations for the $\gamma_{i,k}$ is then obtained by minimizing each terms of the previous equation:

$$\begin{cases} (\gamma_{i,1} + \gamma_{i,2} + \gamma_{i,3}) &= 0 \\ (x_{i-1} - \alpha)\gamma_{i,1} + (x_i - \alpha)\gamma_{i,2} + \dfrac{a^-}{a^+}(x_{i-1} - \alpha)\gamma_{i,3} &= 0 \\ \dfrac{1}{2}(x_{i-1} - \alpha)^2\gamma_{i,1} + \dfrac{1}{2}(x_i - \alpha)^2\gamma_{i,2} + \dfrac{1}{2}\dfrac{a^-}{a^+}(x_{i-1} - \alpha)^2\gamma_{i,3} &= a^- \end{cases} \tag{4.6}$$

and the correcting term C_i yields

$$C_i = \varphi + \gamma_{i,3}(x_{i+1} - \alpha)\frac{\psi}{a^+}. \tag{4.7}$$

As can be seen, the construction of such a finite difference scheme is not easy and is strongly dependent on the initial equation. The method has been extended to the Stokes [Leve 97, Li 04] and Navier-Stokes equations [Li 01, Li 03, Tan 09].

4.1.2 Augmented strategy

First introduced by Li in [Li 98], this approach considers a set of interface points $\mathbf{X}_i$. The jump conditions are written for these points and expressed from the Eulerian points thanks to a least square interpolation and Taylor series expansions. An augmented equation system is then obtained and solved using the Schur complement method. The augmented strategy has been extended to the generalized Helmoltz equations [Li 99] and the incompressible Stokes equations [Li 04]. An advantage of the augmented approach is to provide an algorithm which allows using fast Poisson solvers such as the FFT [Adam 99].

4.2 Ghost node methods

4.2.1 The ghost fluid method

The Ghost Fluid (GF) method has been originally designed to deal with fluid-fluid interfaces for the Euler equations [Fedk 99] and has been adapted to the elliptic equations [Liu 00, Liu 03]. The GF method is simpler than the IIM as it decomposes the flux jump in each axis direction so the problem can be treated dimension by dimension. The result is a lower accuracy than the IIM and a generally first-order accurate on the maximum norm only.

The 1D principle is described. The model interface problem is solved in a domain Ω. The transmission conditions are considered

$$\begin{cases} -\nabla \cdot (a \nabla u) = f & \text{in } \Omega \\ [\![u]\!]_\Sigma = \varphi & \text{on } \Sigma \\ [\![(\mathbf{a} \cdot \nabla u) \cdot \mathbf{n}]\!]_\Sigma = \psi & \text{on } \Sigma \end{cases} \tag{4.8}$$

Let us consider an Eulerian point $\mathbf{x}_i$. The interface passes between x_i and its neighbor x_{i+1}. One can consider that x_i is in the $-$ side and x_{i+1} is in the $+$ side. To discretize a quantity on the interface on a given $+$ or $-$ side, a combination of physical and fictitious nodes are used (as for the boundary GFM of Gibou *et al.*, see section 3.4). The interface coordinate is α and $x_{i-1} < x_i < \alpha < x_{i+1} < x_{i+2}$. The physical solution in Ω_1 is denoted u^- and u^+ in Ω_0. The solution $u^\pm$ is extended from its subdomain to the other subdomain. From a discrete point of view, this extension defines new unknowns which collocates with the existing solution. Hence, two solutions u_i^+ and $u^- i u_i$ coexist at x_i. The classic discretization of $\nabla \cdot (a \nabla u)$ is not designed for a discontinuous solution and using a combination of u^+ and u^- leads to numerical troubles. Hence, each operator is written with only $+$ or only $-$ unknowns. At x_i we have

$$\{\nabla \cdot (a \nabla u)\}_i = \frac{a_{i+\frac{1}{2}} \left(\frac{u_{i+1}^+ - u_i^+}{\Delta x} \right) - a_{i-\frac{1}{2}} \left(\frac{u_i^+ - u_{i-1}^+}{\Delta x} \right)}{\Delta x}. \tag{4.9}$$

Here, u_{i+1}^+ is the ghost node and is not a physical value. The solution jump condition yields

$$u_{i+1}^+ = u_{i+1}^- + \varphi \tag{4.10}$$

and the discretization of $\nabla \cdot (a \nabla u)$ can be rewritten

$$\{\nabla \cdot (a \nabla u)\}_i = \frac{a_{i+\frac{1}{2}} \left(\frac{u_{i+1}^- + \varphi - u_i^+}{\Delta x} \right) - a_{i-\frac{1}{2}} \left(\frac{u_i^+ - u_{i-1}^+}{\Delta x} \right)}{\Delta x} \tag{4.11}$$

and only physical unknowns are used. As can be seen, this method is easier to formulate than the IIM. However, the discretization of the operator has still to be rewritten by-hand.

The GF method has been used for many practical applications such as the fragmentation of a liquid jet [Coud 07], the implosion of bubbles under ultrasonic waves [Coud 09] (see Fig. 4.1) or the turbulent atomization [Desj 08].

4.2.2 The matched interface and boundary method

As for the augmented IIM, the Matched Interface and Boundary (MIB) method [Zhou 06b, Zhou 06a, Yu 07] formulates an augmented system. Instead of using Taylor extension, authors consider as in [Tsen 03, Gibo 02, Fedk 99, Sart 08b] fictitious nodes to discretize the jump conditions. The standard IIM set of jump equations on an interface Γ

$$[\![u]\!]_\Gamma = \varphi \tag{4.12}$$

$$[\![a u_n]\!]_\Gamma = \psi \tag{4.13}$$

are not used as they are and the authors obtain a third jump conditions by differentiating Eq. (4.13) along the tangential direction of the interface. Thus, the three jump conditions are

$$[\![u]\!]_\Gamma = u^+ - u^- = \varphi \tag{4.14}$$

$$[\![u_\tau]\!]_\Gamma = u_\tau^+ - u_\tau^- = \psi_\tau \tag{4.15}$$

$$[\![a u_n]\!]_\Gamma = a^+ u_n^+ - a^- u_n^- = \psi. \tag{4.16}$$

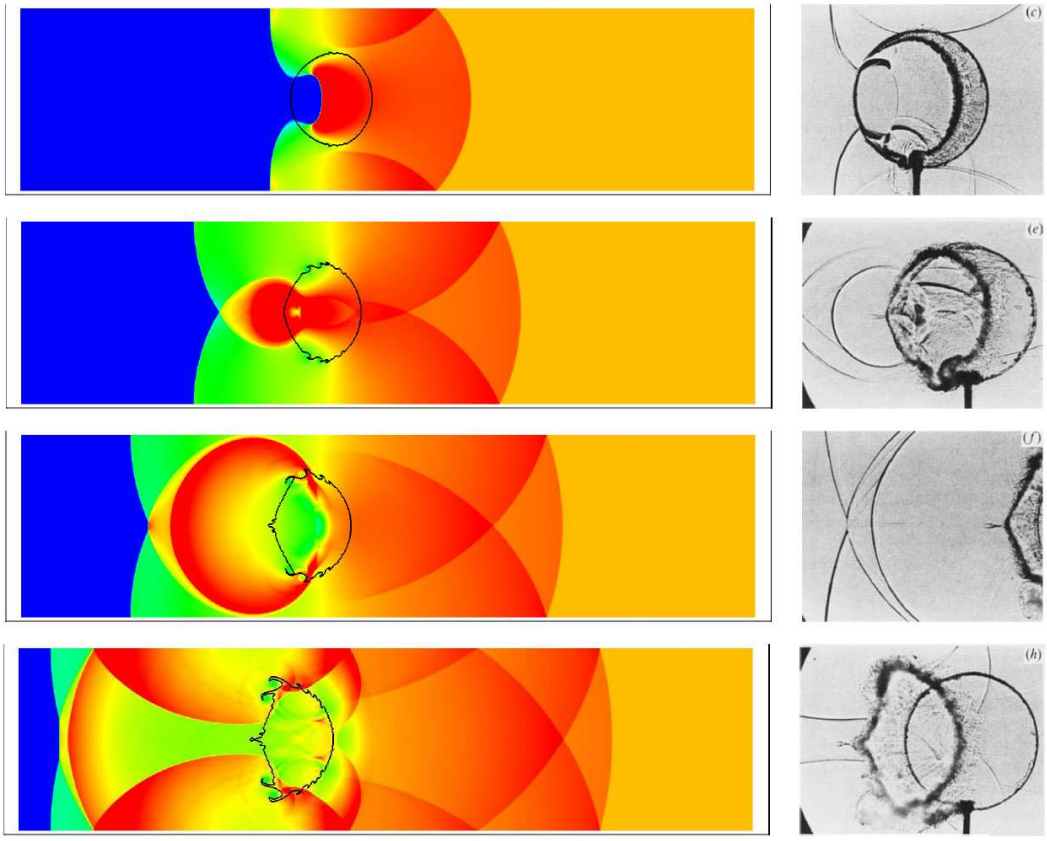


Figure 4.1: Interaction between a wave and a bubble with a Level-set and Ghost Fluid approach [Coud 09]. Comparison with experimentation

Let us consider θ, the orientation of the interface such as the normal vector of the interface is $\mathbf{n} = (\cos\theta, \sin\theta)$ and its tangent vector is $\boldsymbol{\tau} = (-\sin\theta, \cos\theta)$. The three interface conditions above can be reformulated as

$$[\![u]\!]_\Gamma = u^+ - u^- = \varphi \tag{4.17}$$

$$[\![u_\tau]\!]_\Gamma = (-u_x^+ \sin\theta + u_y^+ \cos\theta) - (u_x^- \sin\theta + u_y^- \cos\theta) = \psi_\tau \tag{4.18}$$

$$[\![au_n]\!]_\Gamma = a^+(u_x^+ \cos\theta + u_y^+ \sin\theta) - a^-(u_-^+ \cos\theta + u_y^- \sin\theta) = \psi. \tag{4.19}$$

These relations are considered at each intersection between the interface and the calculation grid. Let us consider an Eulerian point $\mathbf{x}_{i,j}$. The interface passes between $\mathbf{x}_{i,j}$ and its neighbors in the x-direction $\mathbf{x}_{i+1,j}$. One can consider that $\mathbf{x}_{i,j}$ is in the $+$ side and $\mathbf{x}_{i+1,j}$ is in the $-$ side. To discretize a quantity on the interface in a given $+$ or $-$ side, a combination of physical and fictitious nodes is used (see section 3.4). In our case, the two considered nodes are in a x-direction grid line. Hence, u^+, u^-, u_x^+ and u_x^- are easily obtained while u_y^+ and u_y^- are not naturally calculated. However, one can combine Eqs. (4.18) and (4.19) and avoid the computation of u_y^+ or u_y^-. Thus, if u_y^+ is easier to evalute, one will cancel u_y^+ from (4.18) and (4.19) to obtain

$$[\![u]\!]_\Gamma = u^+ - u^- \tag{4.20}$$

$$[\![au_n]\!]_\Gamma - a^- \tan\theta [\![u_\tau]\!]_\Gamma = C_x^+ u_x^+ - C_x^- u_x^- + C_y^+ u_y^+ \tag{4.21}$$

with

$$C_x^+ = a^+ \cos\theta + a^- \tan\theta \sin\theta$$
$$C_x^- = a^- \cos\theta + a^- \tan\theta \sin\theta$$
$$C_y^+ = a^+ \sin\theta - a^- \sin\theta.$$

These two jump conditions are used in the new augmented system. Practically, the corresponding matrix is never built as the additional constraints are related by peers to two fictitious nodes, allowing the local small equation system to be solved (the fictitious solutions are the unknowns). This easy reduction is a consequence of the dissociation of the components of the interface quantities. Contrary to the GF method and the IIM, the MIB approach dissociates the interface constraints and the discretization of the conservation equations.

Another advantage of the method is its ability to reach higher orders. The standard method is first performed. The expressions of the solution for the fictitious nodes are then combined with an higher-order discretization of the jump conditions using additional fictitious nodes. A second order is reached for the standard method (two fictitious nodes, one on each side of the interface), a fourth order is obtained with four auxiliary nodes etc... The only limitation of this approach is topological. A 16th-order scheme has been for instance build in $2D$ for a straight interface [Zhou 06b].

One of its drawback seems to be the size of the discretization stencil. Fig. 4.2 from [Zhou 06b] shows the stencil considered for a 2D case and the standard second-order method. All the nodes are not considered but eight physical nodes are required.

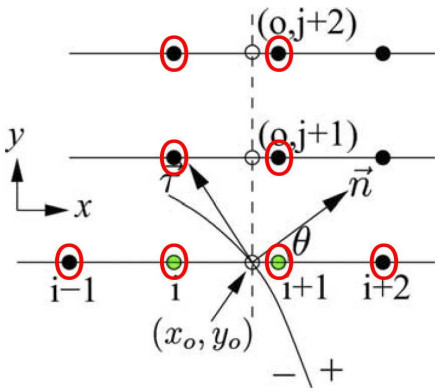


Figure 4.2: Illustration of the discretization of the MIB method from [Zhou 06b]. The four nodes in the x-direction, and the four other nodes around $x_{0,j+1}$ and $x_{0,j+2}$ are used

Discussion and conclusion of Part I

Immersed boundary problems

As can be seen, the forcing term of the IBM and the penalty constraint of the penalty methods can be quite similar. The fondamental difference lies in the way the boundary constraint is imposed. Let us consider an equation $\mathcal{L}(\mathbf{u}) = f$. The discretization of this equation on a Cartesian grid leads to the following equation:

$$LU = F \tag{4.22}$$

with L the discrete linear operator, U the discrete solution and F the discrete source term. The penalty method consists in adding a term $1/\varepsilon(PU - F_p)$ where PU is the linear part of a boundary constraint and F_p its source term. The final equation is

$$LU = F + \frac{1}{\varepsilon}(PU - F_p) \tag{4.23}$$

and ε varies according to the domain. Here, the penalty term does not depend on the discrete equation. For the IBM, one can dissociate two approaches. The continuous approach of the IBM adds a source term F_b

$$LU = F + F_b. \tag{4.24}$$

As the additional term is only a source term, this approach suffers from many limitations and cannot generally reach a second order in space.

In the Direct-forcing approach, a term $D(BU - F_b)$ is added where D is a discrete Dirac function, $B = L + P$ and $F_b = F_p - F$. The factor D is equal to zero everywhere except at the nodes close to the interface. For such nodes, the discrete equation is

$$LU = F + (BU - F_b) \Leftrightarrow PU = F_p \tag{4.25}$$

The terms F_p and P can be the same as for the penalty method and will have the same effect. The fondamental difference is that here the original equation is canceled while the penalty methods make it negligible.

One can notice that the IBM have been almost systematically applied to the fluid flows simulation. For some methods, especially the direct forcing class, the application to the elliptic equations seems to be straightforward. Concerning the DLM method, its initial formulation is strongly linked to a fluid-structure modeling, but an application of the Lagrange multipliers to the elliptic equation could be possible. To our knowledge the Boundary GFM has only been applied as it is to the elliptic equations (while the GFM has been applied to many equations). The Cartesian grid methods have been applied to elliptic and Navier-Stokes equations.

All these methods reach high orders in space accuracy, except for the Peskin IBM (which is very robust and well adapted to elastic boundaries) and the DLM method (well suited to rigid moving obstacles). The order of the penalty method depends on the penalty constraint. More generally, a first order only is obtained for non-smooth problems if the precise location of the boundary is not taken into account or if the influence of the boundary is smoothed. More precisely, the distance and the direction has to be considered by the model.

Among all these approaches, the Cartesian Cells method seems to be the hardest to implement as the Eulerian grid cells have to be modified. Furthermore, unstructured FV schemes has to be built. At a smaller scale, finding the intersection point between the Eulerian and the Lagrangian grids (for high-order Direct-forcing IBM, the Boundary GFM and the Sub-Mesh Penalty method) requires too an implementation effort which can be avoided with the Peskin IBM or the DLM

methods. Concerning the computational performances of the methods, the high-order methods generally push down the matrix conditioning and require more solver iterations. However, these methods are almost always fully implicit and allow higher time steps to be used. The DLM method requires additional sub time steps to impose a solid behavior in the object through a minimization procedure.

Immersed interface problems

The IIM and GFM have been applied to many equations, while the recent MIB method has only been applied to elliptic and Maxwell equations. The IIM and GFM are quite complicated to formulate as the discretization of the operators has to be modified by-hand. The formulation of the GFM using ghost nodes is nevertheless simpler than the IIM formulation using Taylor series expansions. Contrary to the IIM and GFM, the MIB method proposes an automatic correction of the discretization of the space operator. However, such a correction need an implementation effort itself. The same thing occurs with the Augmented IIM method. To our knowledge, the IIM and MIB methods systematically reach high orders in accuracy. That is not always the case for the GF method (especially for the maximum norm). However, the MIB method has only been applied to a short range of application.

As for the IB methods, higher orders are reached if the interface position is accurately accounted for. The Continuum Surface Force (CSF) method of Brackbill *et al.* [Brac 92] is an extension of the Peskin IBM to the interface. As the interface properties are smoothed with discrete Dirac functions, only a first-order accuracy is generally obtained.

Conclusion

As can be seen, the literature related to the fictitious domains is wide, and our presentation does not pretend to be exhaustive (one can cite the Fat Boundary Method [Maur 01] of Maury). Many methods still propose to reach high-order accuracy and the actual challenge is rather to obtain robust schemes, when complex geometries with singular points are involved. One can cite the work of [Yu 07] for the MIB method. However, the immersed interface schemes are generally quite complex to formulate and to implement, and are generally non-conservative at the interface. Hence, finding simple, robust and conservative schemes is the interesting objective, especially for the Stokes and the Navier-Stokes problems.

Building an efficient fictitious domain method requires one of the previous methods, but also a set of algorithms to pre-treat the considered boundary/interface. This part is not negligible in term of implementation effort and computational time. The next part presents a complete treatment of the Lagrangian shape and its projection onto the Eulerian grid.

Part III

Management of Lagrangian shapes on curvilinear grid

Table of Contents

Introduction

The simulation of realistic fluid mechanics and thermal transfers problems always involves varying physical values, such as density, viscosity or thermal conductivity. Generally, discontinuities are present on interfaces which can sometimes be considered as boundaries. Depending on the way the interface is defined and numerically stored, the different steps of the calculation requiring interface informations will be more or less easy to perform. The most common input format for an interface, a $d-1$ manifold in a space of dimension d, is the explicit Lagrangian representation. In 2D, the interface is a curve discretized as a set of segments. In 3D, the interface is a surface discretized as a set of triangles.

The first issue for the shape management is to couple the surface information of the interface to the volume information of the Eulerian discretization grid. This operation allows the physical quantities to be initialized in the physical Eulerian calculation grid, *e.g.* the different viscosities and densities for a two-phase flow.

Moreover, this exemple contraries the classic assumption of the basic discretization of the conservation equations which is that the physical quantities, if not constant, are smoothly varying. This assumption is false in many other cases: heat transfers between two materials, fluid-structure interactions, jump conditions, surface tension on an interface... The treatment of these discontinuities is a major issue when the discretization is based on an Eulerian structured grid, where the irregularities of the physical quantities are rarely matching the grid. The fictitious domain methods propose to deal with discontinuities on structured grids. The two last decades have been particularly creative in this domain and many methods have been invented, such as the Immersed Boundary methods, the Penalty methods, the Ghost-Fluid methods or the Immersed Interface methods. The accuracy of such approach requires many operations related to the interface position and its representation.

Hence, complex problems involve complex interfaces which have to be accurately and quickly managed. The present work proposes a global methodology to manage interfaces of complex shapes on Eulerian grids. The present part first explains different methods performing the Eulerian-Lagrangian grid coupling. The aim is to project Lagrangian surface informations on an Eulerian grid. This step allows to construct the common implicit volume functions: the binary Heaviside, the level-set and the volume of fluid (VOF) functions. A fast thread ray-casting method is presented to build the Heaviside function. As this method works only for Cartesian grids, a curvilinear to Cartesian transformation is required to generalize the approach to orthogonal curvilinear grids. This method unfold the curvilinear grid to a Cartesian grid. Many operations are then quickly performed on this new grid.

The fourth section is composed of tests and validation of the algorithms and the overall method. The last section summarizes the current methodology and describes some possible optimizations .

Chapter 5

Global methodology

5.1 Surface representation

A surface in a space of dimension d is a $d-1$ topological manifold. Its practical storage and representation depends on the application. An explicit representation gives directly the position of the points of the surface. Triangularized meshes or parametric surfaces are explicit surface representations. With an implicit representation, the position of the points of the interface are deduced as the iso-surface of a volume field or as the location of the solution of an equation. Our aim here is to simulate physics using complex surfaces (boundary or interface).

5.1.1 Explicit surface: the triangularized mesh

The natural and intuitive representation of a surface is explicit, and the easiest way to create a complex surface "by hand" is to work explicitly with it. Furthermore, a digital tool (a software) is needed to build such virtual entities, and the most used of them uses explicit representation. One can define two classes of softwares used to create explicit surfaces:

- **Computed assisted design (CAD) softwares**: Such softwares are designed for industrial applications: CATIA (Dassault System), Solidworks (SolidWorks corp.), ProEngineer (Parametric Technology Corporation)...
- **Computer graphics (CG) softwares**: Computer graphics tools are image oriented. They are less precise than CAD tools, but more intuitive. The most used are 3D Studio Max, Maya, Softimage (Autodesk), Lighwave 3D (NewTek) and Blender (Blender Foundation). One can notice that Blender is a free software under GPL licence.

The shape which defines an immersed boundary or an immersed interface is a key point in the fictitious domain approach. Methods can be built and studied using analytical interfaces, such as circles, spheres or boxes. However, a more general description of the interface is needed to treat more complex problems. The Lagrangian meshes are often used to explicitly define generic discrete shapes. Such meshes are composed of segments in 2D and triangles in 3D (Fig. 5.1).

Generally, the Lagrangian mesh is defined with the following constraints:

- The shape has to be closed. Each segment of the mesh must be side of two triangles.
- Two elements of the shapes cannot intersect themselves

Fig. 5.2 shows invalid meshes.

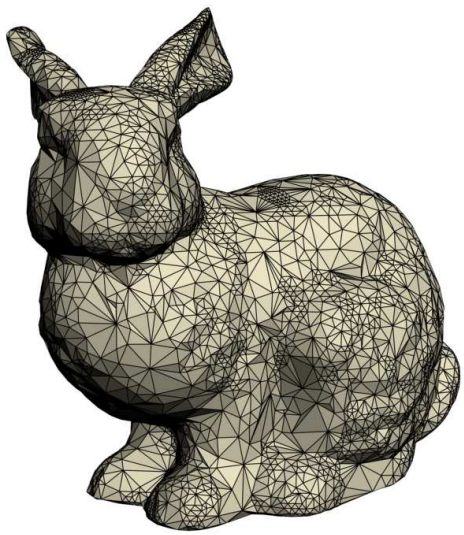


Figure 5.1: A triangularized Lagrangian mesh of the Stanford bunny

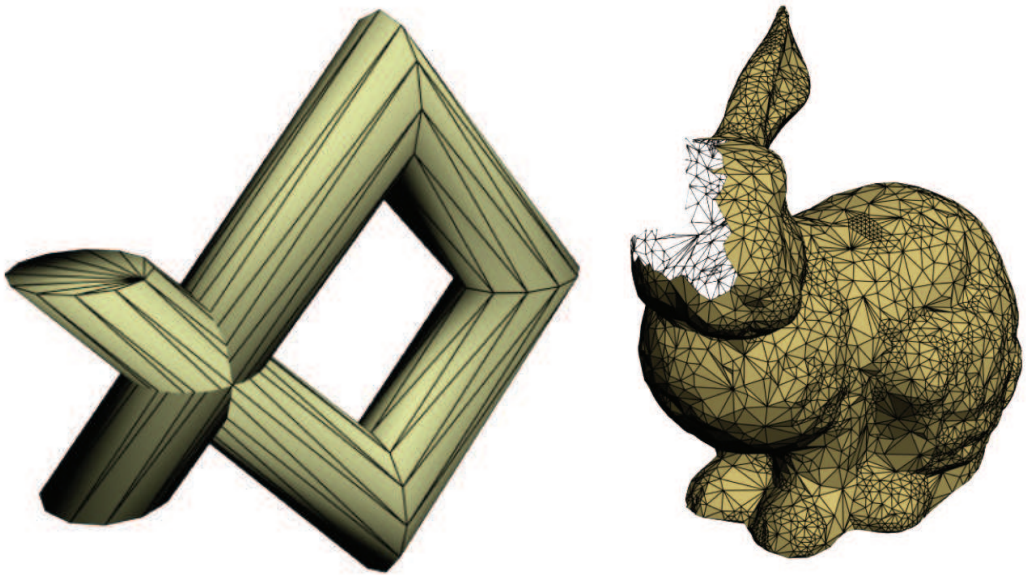


Figure 5.2: Invalid Lagrangian meshes. Overlapping triangles (left) and non closed shape (right)

5.1.2 Implicit surfaces with Heaviside, Level-Set and Color functions

Once the discrete shape is defined, the Lagrangian surface information has to be coupled with the volume Eulerian information. To know which part of the Eulerian mesh is inside the Lagrangian mesh is generally the most important information to obtain. The Fig. 5.3 shows the initial Lagrangian and Eulerian grids and the basic projection of the first on the second. This projection define a first implicit representation of the initially explicit surface. A adequate implicit representation can be accurate enough and can avoid to use the Lagrangian mesh during the rest of the simulation. An implicit representation is defined by a volume function such as

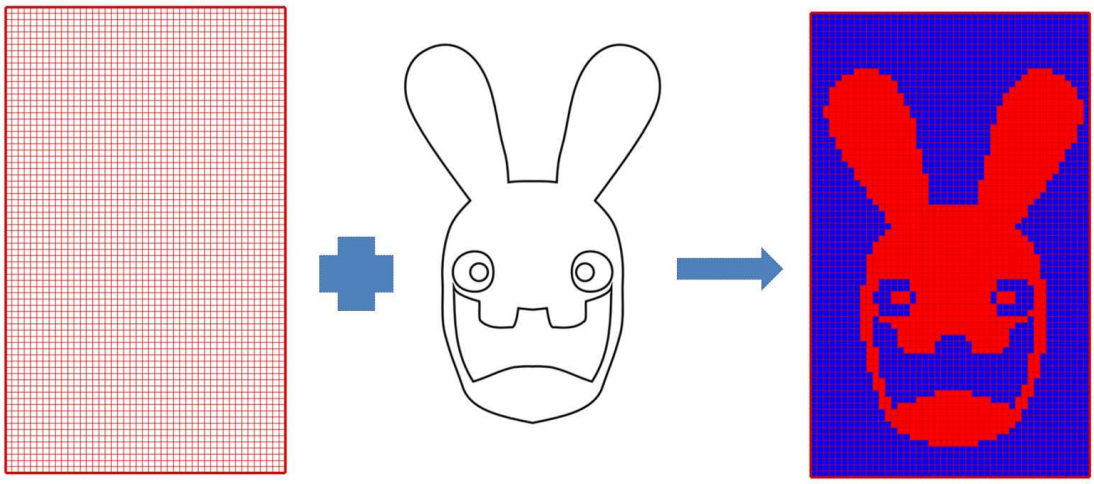


Figure 5.3: The Eulerian and Lagrangian grids and the resulting projection

$\chi : \mathbb{R}^d \mapsto \mathbb{R}$. Three volume functions are defined according to their return value:

- The discrete binary Heaviside function χ, defined as:

$$\chi(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_1 \\ 0.5 & \text{if } \mathbf{x} \in \Sigma \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

  This function is the basic indicator of the presence of an Eulerian point in Ω_1 and is build with a point in solid method presented below. An Eulerian node is near the interface if one of its neighbor has a χ function different to its value. However, a precise location of the interface cannot be retrieved with the χ function only.

- The level-set function ϕ, with :

$$\phi(\mathbf{x}) = \begin{cases} -\text{dist}_\Sigma(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega_1 \\ \text{dist}_\Sigma(\mathbf{x}) & \text{otherwise} \end{cases} \tag{5.2}$$

  and $\text{dist}_\Sigma(\mathbf{x}) = \inf_{\mathbf{p} \in \Sigma} \|\mathbf{x} - \mathbf{p}\|$. The unsigned distance is computed geometrically. The sign is directly obtained from the discrete Heaviside function χ. In 1D, the level-set function gives the exact location of the interface. For higher dimensions, a good average location of the interface can be found in $\phi = 0$. A major advantage of the Level-set approach is to allow the normal $\mathbf{n}$ and the curvature κ to be easily computed:

$$\mathbf{n} = \frac{\nabla \phi}{\|\nabla \phi\|} \tag{5.3}$$

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) \tag{5.4}$$

- The volume of fluid (VOF) function C, also called the color or phase function, is the volume ratio of a given phase in an elementary volume $\mathcal{V}_\mathbf{x}$ centered in $\mathbf{x}$ of boundary $\partial \mathcal{V}_\mathbf{x}$.

For $\mathcal{V}_\mathbf{x} \supset \mathbf{x}$ we denote $C(\mathbf{x})$ the phase ratio in $\mathcal{V}_\mathbf{x}$. This function is obtained from the normalized integration of the Heaviside function on $\mathcal{V}_\mathbf{x}$

$$C(\mathbf{x}) = \frac{1}{\text{meas}(\mathcal{V}_\mathbf{x})} \int_{\mathcal{V}_\mathbf{x}} \chi \, \mathrm{d}V \tag{5.5}$$

One can notice that

$$\text{meas}(\mathcal{V}_\mathbf{x} \cup \Omega_0) = 0 \Leftrightarrow C(\mathbf{x}) = 1 \tag{5.6}$$

$$\text{meas}(\mathcal{V}_\mathbf{x} \cup \Omega_1) = 0 \Leftrightarrow C(\mathbf{x}) = 0. \tag{5.7}$$

The local value $C(\mathbf{x})$ of the volume function is a filtered heaviside function representing the volume average of χ over $\mathcal{V}_\mathbf{x}$. This function is typically used to localize a fluid phase in multiphase-flows and is the base of the 1-fluid model.

The Fig. 5.4 summarizes the three functions. The VOF functions can be seen as a smoothed Heaviside function and implicitly defines the interface with more accuracy.

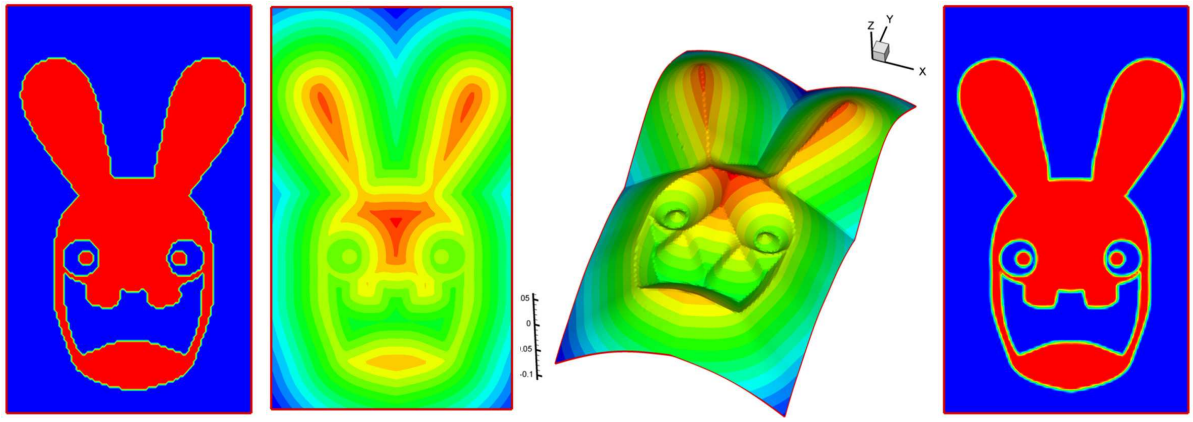


Figure 5.4: Heaviside (left), level-set (middle) and VOF (right) functions for a same geometry

5.2 The global methodology

The key point of the present methodology is to work as much as possible in a Cartesian framework instead of a curvilinear one. Many optimisations can be performed when the grid lines of the Eulerian mesh are straight and when the space step is unit. Hence, it is very easy to know in which cell a Lagrangian point is when such a grid is used. Furthermore, the Thread Ray-casting, a fast way to obtain the Heaviside function, works only on Cartesian grids.

The main idea is to first unfold the orthogonal curvilinear grid T_h to obtain a dual Cartesian grid $\hat{T}_h$. Then, the interfaces are projected onto this new grid. The Cartesian grid and the deformed interfaces are used as much as possible to perform various steps of the calculation, including the shape initialization, the fictitious domain methods and the interface tracking.

The 2D shape of the rabbit is projected on a curvilinear grid. The Fig. 5.5 shows the interface and the grid. The Fig. 5.6 shows now the initial interface in the curvilinear workframe and its transformed into a Cartesian workframe. As can be seen, the shape of the interface in the Cartesian frame is displaced, scaled and deformed.

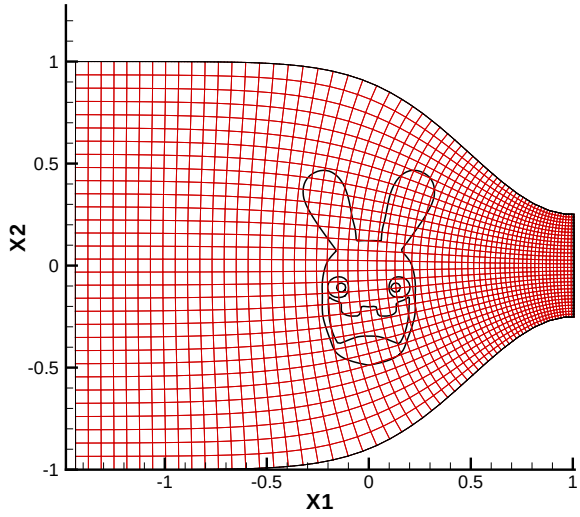


Figure 5.5: Original interface and the curvilinear mesh

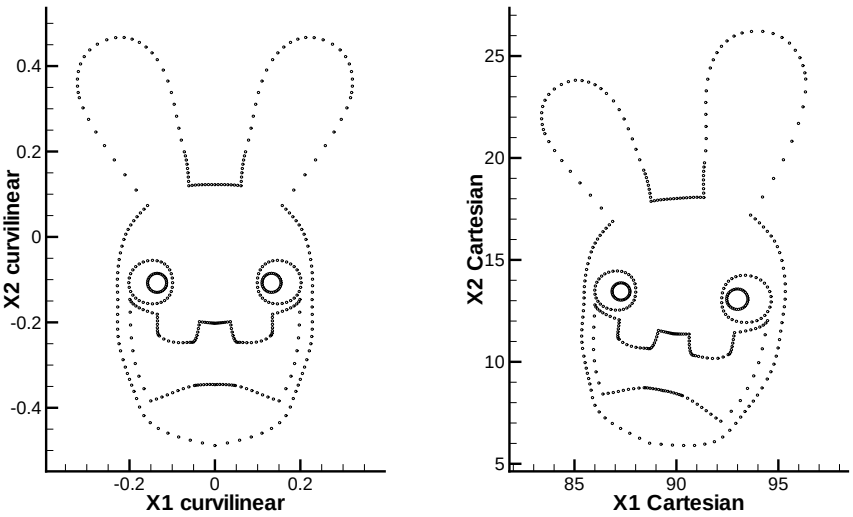


Figure 5.6: Original interface and its transformed onto the Cartesian framework

5.3 Interface tracking on curvilinear grids

The numerical simulation of interface motion and two-phase flows on fixed Cartesian grids requires an interface tracking with specific numerical methods such as the volume of fluid (VOF) approach [Hirt 81, Youn 82], the Level-set method [Suss 94] or the Front-tracking techniques [Unve 92, Shin 02b]. Among the wide variety of methods and articles published the last fifteen years, very few works were devoted to the extension of the previous methods to fixed orthogonal curvilinear grids. An adaptation of the VOF-PLIC method on curvilinear grids can be found in

[Jang 08]. In [Mura 06], Muradoglu and Kayaalp use an auxiliary Cartesian grid superimposed to the curvilinear one to manage the front tracking operations. Kernel functions are used to interpolate the velocity from one grid to another. In [Huan 07], Huang *et al.* extend the ghost fluid method [Fedk 99, Kang 00, Liu 00] to curvilinear grids. The jump conditions are enforced on the pressure and velocity and on the pressure gradient. To be complete, such a ghost fluid method would have to be extended to velocity gradient, viscosity and turbulence quantities. The approximation of the interface tracking on curvilinear grids for two-phase flows is conside*red in this section. The extension and the generalization of the curvilinear features for more than two fluids is straightforward. It is considered that the curvilinear velocity field representing the fluid motion is known. One can notice that in the present work, the Cartesian grid is not superimposed to the curvilinear grid as in [Mura 06], but is an unfolding of the curvilinear grid.

5.3.1 The VOF-PLIC method

The VOF is by definition associated to the use of a volume characteristic function C which is equal to 1 in one phase and 0 in the other phase. The interface is classically located by the iso-surface $C = 0.5$. A material equation on C, which correspond to the Lagrangian trajectory of fictitious particles placed on the interface, is added to the standard conservation equations, *i.e.* the Navier-Stokes and energy equations [Scar 99a], in order to follow the phase evolutions during time:

$$\frac{\mathrm{d}C}{\mathrm{d}t} = \frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = 0 \tag{5.8}$$

where $\mathbf{u}$ is the fluid velocity when no phase change occurs and t is the time.

Equation (5.8) is correct for every orthogonal coordinate system. However, discretizing the gradient operator in a curvilinear orthogonal grid G and simulating the corresponding velocity field in such a grid is a complex task, in particular in three dimensions. Moreover, as soon as (5.8) is approximated by means of a geometrical approach such as the VOF-PLIC method [Youn 82], the curvilinear extension of the approach becomes impossible in the real coordinate system as it requires to estimate the intersection between a segment in 2D or a plan in 3D with the curvilinear control volumes of the grid. Our idea lies in the use of a transformed auxiliary grid $\hat{T}_h$, as previously explained for object shape projection, to solve the advection equation on C with the standard VOF-PLIC technique.

A staggered Cartesian grid of constant space step is considered with space steps such as $\Delta x = \Delta y = \Delta z = 1$. In this auxiliary grid, Eq. (5.8) can be written in a new form

$$\frac{\partial C}{\partial t} + \hat{\mathbf{u}} \cdot \hat{\nabla} C = 0 \tag{5.9}$$

where $\hat{\mathbf{u}}$ and $\hat{\nabla}$ are the velocity and the gradient operator in the auxiliary coordinate system. The curvilinear metrics applied on C through ∇ in G disappear in $\hat{G}$ and $\hat{\nabla}$ is the standard Cartesian operator $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^T$.

Eq. (5.8) is then written as

$$\frac{\partial C}{\partial t} + u_x \frac{\partial C}{\partial x}\frac{\partial x}{\partial \xi} + u_y \frac{\partial C}{\partial y}\frac{\partial y}{\partial \eta} + u_z \frac{\partial C}{\partial z}\frac{\partial z}{\partial \zeta} = 0 \tag{5.10}$$

where ξ, η, ζ are the curvilinear coordinates. For each cells, $\frac{\partial x}{\partial \xi}$, $\frac{\partial y}{\partial \eta}$ and $\frac{\partial z}{\partial \zeta}$ are replaced by the

ratios of the local space steps :

$$\frac{\partial C}{\partial t} + \frac{u_x}{\Delta\xi}\frac{\partial C}{\partial x} + \frac{u_y}{\Delta\eta}\frac{\partial C}{\partial y} + \frac{u_z}{\Delta\zeta}\frac{\partial C}{\partial z} = 0. \tag{5.11}$$

Eq. (5.9) is then recovered with

$$\hat{\mathbf{u}}_i = \begin{pmatrix} \frac{u_{\xi i}}{\Delta\xi_i} \\ \frac{u_{\eta i}}{\Delta\eta_i} \\ \frac{u_{\zeta i}}{\Delta\zeta_i} \end{pmatrix} \tag{5.12}$$

By solving Eq. (5.9), the VOF-PLIC method can be used in $\hat{T}_h$, the Lagrangian advection of linear interface construction being achieved with velocity (5.12) without any modification of the implementation of the method. It has to be noticed that the initial values of C are obtained with the Lagrangian-Eulerian transformation on curvilinear grids as the physics and so the real interface shape are known in the curvilinear coordinate system. The discrete value $C(i,j,k)$ of C, as a binary scalar are the same in the two spaces T_h and $\hat{T}_h$.

5.3.2 The LCR Front Tracking method

Among the various variant of the Front Tracking approaches, the Level Contour Reconstruction (LCR) method of Shin and Juric [Shin 02b] is interesting as it allows the coalescence and break-up of interfacial structures to be automatically managed by using a Heaviside function χ to reseed the marker over time by estimating the intersections between each Lagrangian elements of the interface $\Gamma_i(t)$ and the control volumes of the Eulerian grid. The function χ is built with the position of the markers located on $\Gamma_i(t)$. The χ function is a pseudo VOF function which is equal to 1 in the interior phase bounded by the interface and 0 elsewhere. The resolution of a Poisson equation is required to obtain χ such that

$$\nabla^2\chi = \nabla \cdot \int_{\Gamma_i(t)} \mathbf{n}\delta(\mathbf{x} - \mathbf{x}_i)d\gamma \tag{5.13}$$

In equation (5.13), $\mathbf{n}$ is the unit normal to the interface, δ a Dirac function centered on the interface, $\mathbf{x}$ is a position on the Eulerian grid and $\mathbf{x}_i$ a marker position on $\Gamma_i(t)$.

In the LCR approach, the interface evolutions are estimated in a Lagrangian manner as follows:

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} \cdot \mathbf{n} = \mathbf{u} \cdot \mathbf{n} \tag{5.14}$$

The interpolation of the Eulerian velocity field $\mathbf{u}$ at the Lagrangian positions and the approximation of the Dirac function appearing in (5.13) correspond to Eulerian to Lagrangian projection of variable, or conversely. These operations require smooth distribution functions [Shin 02b] as the Lagrangian positions $\mathbf{x}_i$ do not generally coincide with the Eulerian grid nodes $\mathbf{x}$.
The main interest of LCR method is linked to the automatic management of interface merging and rupture by using the intersection of the $\chi = 0.5$ iso-surface with the control volumes of the calculation grid. To our knowledge, no curvilinear version of LCR exists in the literature, for the same reason as for the VOF-PLIC method: the estimate of geometrical intersections on curvilinear grids is complex. The idea of using an auxiliary Cartesian grid $\hat{G}$ is also a valid idea for the Front-Tracking method.
The curvilinear extension of LCR lies in the use of velocity $\hat{\mathbf{u}}$(5.12) for interpolating $\hat{\mathbf{u}}_i$ in $\hat{G}$ with

the discrete Dirac function proposed by Shin and Juric. This velocity field is used for advecting the markers and in the transformation of the Lagrangian interface shape $\Gamma_i(t)$, through its coordinate vectors $\mathbf{x}_i$, as proposed in the section devoted to interface initialization on the auxiliary grid $\hat{G}$. The coordinates of the markers so obtained are called $\hat{\mathbf{x}}_i$. The new equation describing the Lagrangian interface evolution reads

$$\frac{\mathrm{d}\hat{\mathbf{x}}_i}{\mathrm{d}t} \cdot \hat{\mathbf{n}} = \hat{\mathbf{u}}_i \cdot \hat{\mathbf{n}} \tag{5.15}$$

where $\hat{\mathbf{n}}$ is the local unit normal to the transformed interface $\hat{\Gamma}_i(t)$. Once $\hat{\mathbf{x}}_i$ and so $\hat{\Gamma}_i(t)$, is known, the Heaviside function χ is obtained by solving the following equation

$$\nabla^2 \chi = \nabla \cdot \int_{\hat{\Gamma}_i(t)} \hat{\mathbf{n}} \delta(\hat{\mathbf{x}} - \hat{\mathbf{x}}_i) d\gamma \tag{5.16}$$

As for the VOF function C, χ is the same in T_h and $\hat{T}_h$.

5.3.3 The Level-set method

As for the other volume functions which consider an implicit representation of the interface, the initialization and the time evolution of the level-set function is performed in the Cartesian frame. The level-set function is denoted $\hat{\phi}$ in the Cartesian grid. As for the VOF function, $\hat{\phi}$ is advected according to

$$\frac{\partial \hat{\phi}}{\partial t} + \hat{\mathbf{u}} \cdot \hat{\nabla} \hat{\phi} = 0 \tag{5.17}$$

The VOF-PLIC method solves Eq. (5.9) geometrically. For the Level-set method, the hyperbolic equation (5.17) is solved explicitly with a finite-volume method. In [Tang 04], Tanguy shows that high orders schemes have to be used to obtain acceptable results. Here, the time advancement is performed with a Runke-Kutta 3 scheme [Shu 98] while the spatial derivatives are discretized with a WENO 5 scheme [Shu 96]. The fonction thus obtained is no more a distance function as its fundamental properties are not conserved. Hence, a reinitialization procedure is needed. Proposed by Sussman *et al* [Suss 94], the idea is to use the only valid iso $\hat{\phi} = 0$ to rebuild the function. Amongst the numerous method to reinitialize the Level-Set, the one used here solved the following PDE:

$$\begin{cases} \dfrac{\partial \hat{\phi}}{\partial t'} = sign(\hat{\phi}(\mathbf{x}, t)(1 - \|\nabla \hat{\phi}\| \\ \hat{\phi}(\mathbf{x}, t' = 0) = \hat{\phi}(\mathbf{x}, t) \end{cases} \tag{5.18}$$

Details on the numerical discretization can be found in [Jian 00, Coud 07]. One can notice that some geometric properties of the level-set function in the Cartesian frame are no more verified in the curvilinear frame. However, the interface normal as well as the curvature can be retrieved with a suitable modification of the gradient:

$$\frac{\nabla \phi}{\|\nabla \phi\|} = \begin{pmatrix} \dfrac{1}{\Delta\xi} \\ \dfrac{1}{\Delta\eta} \\ \dfrac{1}{\Delta\zeta} \end{pmatrix} \frac{\hat{\nabla} \hat{\phi}}{\|\hat{\nabla} \hat{\phi}\|}. \tag{5.19}$$

Thus, the curvature κ can then be obtained with

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right). \tag{5.20}$$

5.4 The fictitious domain methods

Generally, the fictitious domain methods modify the original discretization of the conservation equations near the interface or in a given sub-domain. The spatial accuracy of these methods depends on the accuracy of the interface localization. If only the discrete Heaviside χ is used, one cannot obtain more than a first order of accuracy in space. Higher orders are obtained when the accurate location of the interface is used. A second order can be reached both using the explicit surface (*i.e.* the Lagrangian mesh) or the implicit surface provided by the level-set function. Various results are obtained with the different VOF functions.

Chapter 6

Detailed algorithms of the methodology

THE algorithms used to performed our strategy are presented in this chapter. Point-in-solid algorithms, a Level-set construction and a Cartesian to curvilinear transformation are described as well as their optimization using geometric arguments and advanced data-structures. Physical problems implying interfaces are then simulated to evaluate the accuracy of our approach.

The computational domain Ω is approximated with a curvilinear mesh T_h composed of $N \times M$ ($\times L$ in 3D) cell-centered finite volumes ($\mathcal{V}_I$).

6.1 Point in solid algorithm

Some algorithms used to initialize implicit representation of the interface are presented here. A huge literature can be found in the computer graphics community. A good review can be found in [Ogay 05].

6.1.1 A continuous method

The method presented here has been first introduced by Shin and Juric in [Shin 02b]. As usual, we are looking for χ such as χ is the indicator of the sub-domain bounded by Σ. Contrary to many point-in-solid methods where geometrical properties are used, this approach deals with the resolution of a Poisson equation:

$$\begin{cases} \nabla^2 \chi = \nabla \cdot \mathbf{F}_\Sigma \\ \chi = 0 \text{ on } \partial\Omega \end{cases} \tag{6.1}$$

with $\mathbf{F}_\Sigma = \int_\Sigma \mathbf{n}\delta_i \left(\mathbf{x} - \mathbf{x}_f\right) \, \mathrm{d}s$ a vector normalized interface contribution, $\mathbf{x}_f$ the interface position and δ_i the Dirac function. This equation can be seen as the modeling of a heat transfer problem where χ is the temperature. The source term of (6.1) is the divergence of a normal flux which is discretized on the staggered grid. The term $\mathbf{F}_\Sigma$ is located on the velocity nodes and $\nabla \cdot \mathbf{F}_\Sigma$ is naturally obtained with a centered scheme on the scalar grid.

Concerning the calculation of $\mathbf{F}_\Sigma$ itself, it has to be integrated over the whole interface for each Eulerian point. Practically, the interface contribution is considered as constant for each element σ_l, $l \in \mathcal{L}_f$ which defines the discrete interface Σ_h. The center of σ_l is denoted as $\mathbf{x}_l$ and $\Delta\sigma_l$ is the measure of σ_l in $\mathbb{R}^{d-1}$. For an Eulerian point $\mathbf{x}_i$, the following approximation is used:

$$\int_{\sigma_l} \mathbf{n}\delta_i \left(\mathbf{x} - \mathbf{x}_f\right) \, \mathrm{d}s \approx \mathbf{n}_l D_i(\mathbf{x}_l) \Delta\sigma_l \tag{6.2}$$

The Dirac function δ_i is approximated using the distribution functions D_i introduced by Peskin [Pesk 02]. In 2D, the distribution function is:

$$D_i(\mathbf{x}_l) = \frac{\delta_h\left((x_l - x_i)/h_x\right)\delta_h\left((y_l - y_i)/h_y\right)}{h_x h_y} \tag{6.3}$$

where (x_i, y_i) are the coordinates of the Eulerian point $\mathbf{x}_i$, (x_l, y_l) are the coordinates of the Lagrangian point $\mathbf{x}_l$, and δ_h is a discrete Dirac function. In our case, we use the same function as Shin and Juric [Shin 02b]:

$$\delta_h(r) = \begin{cases} \delta_1(r), & |r| \leqslant h \\ 1/2 - \delta_1(2 - |r|), & h < |r| \leqslant 2h \\ 0, & |r| \geqslant 2h \end{cases} \tag{6.4}$$

with

$$\delta_1(r) = \frac{3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}}{8} \tag{6.5}$$

where h is generally the local space step of the discrete grid. Hence, we finally obtain

$$\int_\Sigma \mathbf{n}\delta_i\left(\mathbf{x} - \mathbf{x}_f\right)\, \mathrm{d}s \approx \sum_{\sigma_l,\, l \in \mathcal{L}_f} \mathbf{n}_l D_i(\mathbf{x}_l)\Delta\sigma_l \tag{6.6}$$

and the divergence of $\mathbf{F}_{\Sigma_h}$ can be obtained. As for the IBM of Peskin [Pesk 72], the Dirac discrete function δ_h spreads the interface contribution to the neighbor Eulerian nodes. With the present discretization (6.4), the support of δ_h is two cells width on each side of the interface in the normal direction. As the Dirac function has a limited support, each contribution is computed only if the Eulerian point is in the vicinity of Σ_h.

Once the source term is discretized, the equation can be solved using a fast FISHPACK Poisson solver [Adam 99] based on spectral methods, or a standard finite volume discretization with a second-order centered scheme. In this case, the matrix inversion is performed with a BiCG-Stab II solver [Vors 92] and an ILU preconditioning [Gust 78a].

The approach of Shin and Juric suffers from some drawbacks:

- As only the Lagrangian points generates the source term, they have to be dense enough compared to the Eulerian node. Practically, the Lagrangian mesh has to be refined adaptively such that $\Delta\sigma_l \leqslant h$.
- The method does not perceive the inner holes, so the surface must separate the space in only two connected sub-spaces. In 2D, a circle is a valid interface, a ring is not (see Fig. 6.1).
- The method does not work properly when the Lagrangian mesh is not entirely in the computational domain. The method requires reconstruction of the Lagrangian interface near the boundary to work properly.

Even if the computational drawbacks can be solved thanks to an implementation effort, the limitation on the topology of the surface are very restrictive. As will be demonstrated later, the accuracy of the method in term of implicit surface representation is not as good as with a level-set function.

However, the intrinsic approximations of the method provide an advantage for some cases. If the curvature of the interface is too high compared to the Eulerian mesh, this continuous method smoothes the interface and gives an appropriate representation while other methods could produce a non-valid implicit surface (*e.g.* a non-closed surface).

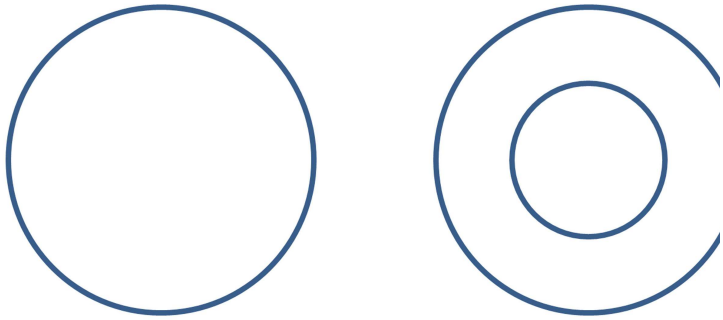


Figure 6.1: Surfaces for 2D problems. The left one is valid, the right one is not

6.1.2 Geometrical methods

6.1.2.1 Ray-casting method

A first issue is to determine which Eulerian points are inside the object defined by the Lagrangian surface. We use a Ray-casting method based on the Jordan Curve theorem. The principle is to cast a ray from each Eulerian point to infinity and to test the number of intersections between the ray and the Lagrangian mesh. If the number of intersections is odd, the Eulerian point is inside the object, either outside. Ray-casting methods can be enhanced by classifying elements of the Lagrangian mesh with an octree sub-structure (see section 6.5.2). If a ray does not intersect a cube, it does not intersect the triangles inside. More generally, a fast test to classify a point as outside or inside is to see if the point is in a box bounding the Lagrangian mesh. If the point is outside, one can be certain that the point is not inside the object. Some details of the implementation of the method can be found in [Ogay 05].

Algorithm 1 describes a pseudo-code performing a basic computation of the color function C. To avoid numerical errors due to the presence of great numbers to simulate the infinity, the ray is only cast to a point $x_{\infty i}$ which is far enough to be outside of the object and the grid. To optimize the intersections calculation, $x_{\infty i}$ is different for all x_i and parallel to a grid line. One recall that M and K are the number of Eulerian cells in the second and third directions.

Algorithm 1 Ray-casting algorithm

```
for i = 1, M do
  nsect := 0
  for k = 1, K do
    if Segment [x_i; x_∞i] intersects σ_k then
      nsect := nsect + 1
    end if
  end for
  if nsect is even then
    C(x_i) := 0
  else
    C(x_i) := 1
  end if
end for
```

Concerning the ray-triangle intersection, [Ogay 05] announces that Feito-Torres [Feit 97] algorithm seems to be the faster.

6.1.2.2 Thread Ray-casting method

We propose now an optimization of the Jordan-based method on orthogonal structured grids that greatly improves the performances of the algorithm. This optimization seems to be known in the computer graphics community but to our knowledge have never been applied to numerical simulation. In Jordan based-method, the direction of the ray is indifferent. If all rays are launched in the same direction, for instance Ox, many intersection tests are done more than one time for a set of point in a same Eulerian mesh row in the Ox direction. Hence, only one ray can be cast for a row. If rays are casted in the more refined direction, computational cost is simply divided by the number of cells in this direction. This method is called the Thread Ray-Casting (TRC).

Alg. 2 describes our TRC algorithm. Rays are cast from points x_i included in a boundary slice $\mathcal{S}_{xy}$ of the Eulerian mesh. For each starting point x_i, the intersections are stored and sorted according to their z component in a two entry structure $PTZ(i, nsect_i)$. For each $x_i \in \mathcal{S}_{xy}$, *nsect* is not known *a priori*. If PTZ is an array, a first pass has to be performed to determine the size of PTZ. A best choice is to use chained lists.

Algorithm 2 Thread Ray-casting algorithm

```
for i = 1, M with x_i ∈ S_xy do
  nsect := 0
  for k = 1, K do
    if Segment [x_i; x_∞i] intersects σ_k then
      Store the intersection in PTZ(i, nsect)
      nsect := nsect + 1
    end if
  end for
  if nsect is even then
    C(x_i) := 0
  else
    C(x_i) := 1
  end if
  In_state := C(x_i)
  nsect_tmp := 0
  for j = 1, m_z do
    while nsect_tmp < nsect and x_j(3) > PTZ(i, nsect_tmp) do
      Switch In_state
      nsect_tmp := nsect_tmp + 1
    end while
    C(x_j) := In_state
  end for
end for
```

For the sake of clarity, the two algorithms (Ray-casting and TRC) are not fully optimized (no bounding box test for instance).

The binary C_i function so obtained can be used to build an Eulerian Level-set function near the interface by estimating the Eulerian distance between the Eulerian points and the neighbor Lagrangian points.

6.2 Cartesian to curvilinear transformation

The key point of our methodology is to work in a Cartesian framework instead of a curvilinear one. Many optimisation in computing Eulerian functions can be performed when the grid lines of the Eulerian mesh are straight.
The main idea is to first unfold the orthogonal curvilinear grid to obtain a dual Cartesian grid. Then, the objects are projected onto this new grid. The method is presented in 2D but can be generalized in 3D without difficulties.

Let $\hat{T}_h$ be the Cartesian structured mesh composed of elements $\hat{\mathcal{V}}_i' = K_i$, T_h being the initial primal orthogonal curvilinear grid (in the finite volume sense). Let P be the projector from T_h to $\hat{T}_h$. Hence, the discrete interface $\hat{\Sigma}_h$ is the projected interface such as $\hat{\Sigma}_h = P(\Sigma_h)$. Each element $\hat{\mathcal{V}}_i'$ is an unit square, such as $\hat{\Omega}_h = [0, N] \times [0, M]$. The transformation of Σ_h is performed by displacing each node of elements σ_i, denoted by $\sigma_{ij}, j = 1, 2$. Let (x_l, y_l) be the position of a node σ_{ij} and $(x_k, y_k), k = 1, .., 4$ the position of each node K_{ij} of the element K_i containing σ_{ij} (see Fig. 6.2 for notations). Two $\mathbb{Q}_1$ interpolations Q_x and Q_y are defined such as $Q_x(x_k, y_k) = \hat{x}_k$ and $Q_y(x_k, y_k) = \hat{y}_k$. The determination of the coefficients requires to solve two linear systems. The analytical solution is used in 2D and a BiCG-Stab method is used in 3D to obtain the projector coefficients. At last, $(Q_x(x_l), Q_y(y_l))$ gives the position of $\hat{\sigma}_{ij}$.

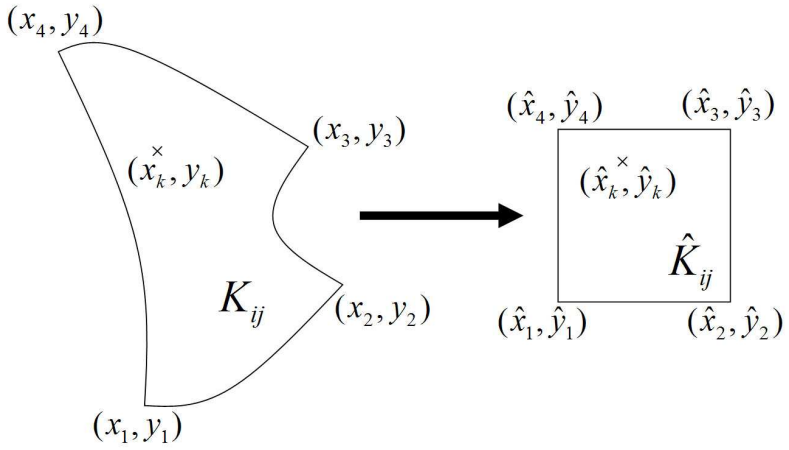


Figure 6.2: Notations and principle of the curvilinear to Cartesian transformation. Original element K_{ij} and projected element $\hat{K}_{ij}$ are described

6.3 Level-set function

6.3.1 Computing the Level-Set function

A good review of the methodologies used to compute the distance function has been performed by Jones et. al. [Jone 06]. A global method can be found in [Baer 05]. The Level-set function results from the calculation of the local sign and the calculation of the unsigned distance to the interface.

In the present work, the sign is obtained with the Heaviside function χ. If the purpose of the Level-Set is only to build a VOF function, the Level-Set function is only required at the nodes close to the interface. To locate such nodes, an explicit Laplacian is applied several times to the Heaviside function χ_Δ. Nodes where $0 < \chi_\Delta < 1$ are at the vicinity of the interface.

The calculation of the unsigned distance between a point $\mathbf{x}$ and a triangle T is now described. The idea is globally the same as [Jone 05]. The main difficulty of the computation of the distance from a point to a segment or a triangle is to determine which part of the element (vertex, edge, face) is the closest.

2D computation The segment σ_l is defined by two vertices $\mathbf{p}_1$ and $\mathbf{p}_2$. We define two vectors $\mathbf{v}_1 = \mathbf{p}_2 - \mathbf{p}_1$ and $\mathbf{v}_\mathbf{x} = \mathbf{x} - \mathbf{p}_1$. The position of the orthogonal projection of $\mathbf{x}$ on σ_l is deduced from the quantity $E = \mathbf{v}_\mathbf{x} \cdot \mathbf{v}_1$:

- $E < 0 \Rightarrow$ the closest part is the point $\mathbf{p}_1$
- $0 < E < 1 \Rightarrow$ the closest part is the segment σ_l
- $E > 1 \Rightarrow$ the closest part is the point $\mathbf{p}_2$

and the distance to the closest element is computed.

3D computation Thee triangles σ_l are defined by three vertices $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. We define two vectors $\mathbf{v}_2 = \mathbf{p}_2 - \mathbf{p}_1$ and $\mathbf{v}_3 = \mathbf{p}_3 - \mathbf{p}_1$. The normal vector $\mathbf{n}$ of σ_l is

$$\mathbf{n} = \frac{\mathbf{v}_2 \times \mathbf{v}_3}{\|\mathbf{v}_2 \times \mathbf{v}_3\|} \tag{6.7}$$

The point $\mathbf{x}'$ is the orthogonal projection of $\mathbf{x}$ on the plane containing σ_l. The problem is now reduced to a 2D problem. The next step is to determine which part of the triangle is the closest. If $\mathbf{x}'$ is in σ_l, the closest part is the face. If $\mathbf{x}'$ is exterior, six other cases appears (Fig. 6.3). To

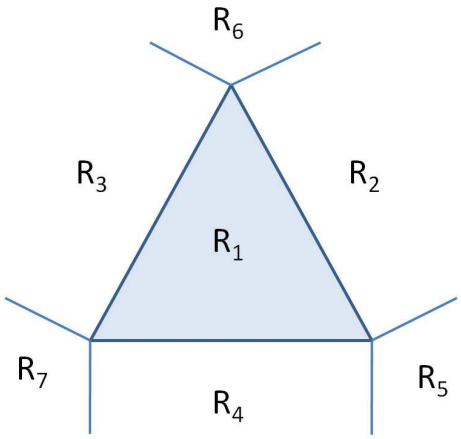


Figure 6.3: The seven different regions delimiting the closest element to a point

localize the region, we define oriented lines overlapping the triangle edges and the lines orthogonal to the edges passing by the vertices (see Fig. 6.4).

The edge equation [Pine 88] is used to determine in which side of the line the point is. The edge equation is

$$E(x, y) = (x - X)dY - (y - Y)dX \tag{6.8}$$

for a line passing through $(X; Y)$ with gradient $\frac{dX}{dY}$ with respect to a point (x, y). If $E < 0$, the point is to the left of the line, if $E > 0$ to the right, and if $E = 0$ it is on the line. Knowing the planar distance to the triangle, the distance in 3D is easily retrieved.

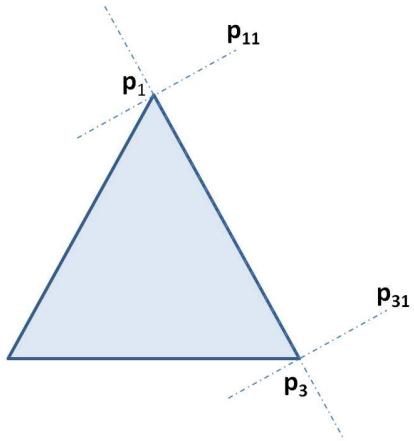


Figure 6.4: Lines used to find the region of a point

6.3.2 Optimizations

As for the other geometrical algorithms, many optimizations can be implemented:

- The search for the minimal distance has to be done with squared distances. Hence, a square root is only applied to the final distance.
- For points far enough from the interface, one can compute the distance to the points of the interface only. Such points are located using the smoothed Heaviside function χ_Δ. If all the elements are smaller than the Eulerian cells, such a method gives a good approximation. New points can be temporarily created and the elements refined for the need of this computation.
- An octree data-structure (see section 6.5.2) is used to sort the elements spatially. If a given leaf of the octree (*i.e.* the closest box containing elements of the interface) is the closest to $\mathbf{x}$, the closest element of the interface is in this leaf or in a neighbor leaf.
- For each element, a sphere containing the element is computed. Its center is $\mathbf{p}_c$ and its radius is r. If $\|\mathbf{x} - \mathbf{p}_c\| - r$ is greater than the actual closest computed distance, there is non need to compute the real distance from $\mathbf{x}$ to the element.

The formula commonly used to retrieve the location $\mathbf{x}_\Sigma$ of the interface between two nodes $\mathbf{x}_1$ and $\mathbf{x}_2$ from $\phi_1 = \phi(\mathbf{x}_1)$ and $\phi_2 = \phi(\mathbf{x}_2)$ is

$$\mathbf{x}_\Sigma = \frac{\mathbf{x}_1|\phi_2| + \mathbf{x}_2|\phi_1|}{|\phi_1| + |\phi_2|}. \tag{6.9}$$

6.4 VOF function

A first approximation of the VOF function is the Heaviside function χ. In our approach, the VOF function is generally build from the level-set function ϕ. In [Suss 98], Sussman and coauthors propose the following function:

$$C(\mathbf{x}) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} - \frac{1}{\pi}\sin(\pi\phi/\epsilon)\right) & \text{if } |\phi| \leqslant \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases} \tag{6.10}$$

where ϵ is a characteristic distance, *e.g* the cell size. The resulting interface has a thickness (the nature of this thickness will be discussed later) of about

$$\frac{2\epsilon}{|\nabla\phi|}. \tag{6.11}$$

If the level-set is the distance function, the thickness is then 2ϵ. This smooth function has good regularity properties.

The formula commonly used to retrieve the location $\mathbf{x}_\Sigma$ of the interface between two nodes $\mathbf{x}_1$ and $\mathbf{x}_2$ from $C_1 = C(\mathbf{x}_1)$ and $C_2 = C(\mathbf{x}_2)$ is

$$\mathbf{x}_\Sigma = \frac{\mathbf{x}_1|0.5 - C_2| + \mathbf{x}_2|0.5 - C_1|}{|0.5 - C_1| + |0.5 - C_2|}. \tag{6.12}$$

However, this interpolation is designed for a VOF function which is linear for $0 \leqslant C \leqslant 1$ and a loss of accuracy is encountered when the interface is described by a VOF function build with the Sussman function (6.10). If only a $\mathcal{C}^0$ the regularity of the VOF function is required, the following function is more desirable:

$$C(\mathbf{x}) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \dfrac{1}{2}\left(1 + \dfrac{\phi}{\epsilon}\right) & \text{if } |\phi| \leqslant \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases} \tag{6.13}$$

With this function, the same surface as for the level-set function can be retrieved. For the level-set function, the iso $\phi = 0$ is the location of the interface. For the VOF function, two approaches are generally used. A first approach considers that the interface is the zone where $0 < C < 1$, so the interface has a thickness. When diffusive advection schemes are used, the thickness of the interface will eventually grow.

Another point of view considers that the interface is located in $C = 0.5$. The numerical thickness of the interface is a way to increase the accuracy of the implicit representation of the surface (the Heaviside function which is binary does not allow the sub-mesh position of the interface to be retrieved). In this way, the interface position is as accurately defined as for a level-set function.

6.5 Validation and global convergence

6.5.1 Accuracy of the method

6.5.1.1 Interface location for a circle with an immersed boundary method

An accuracy test of the Lagrangian to Eulerian projections coupled with the curvilinear to Eulerian projection is performed here. The resolution of the Laplace equation with an immersed boundary is considered. The IB is accounted for using the Sub-Mesh Penalty (SMP) method (see 7). It allows a Dirichlet boundary condition on a complex interface to be imposed with a second order of spatial accuracy. This accuracy is directly bind to the accuracy of the interface localization. The SMP method can consider implicit and explicit representations of the interface and the different approaches are compared. As usual, computational time is saved by performing all the related calculations of the SMP method in the Cartesian transformed frame. Validations for the IB problem are performed on two curvilinear grids (see Fig. 6.5). Grid A is an orthogonal mesh with exponential periodic steps. Grid B is a converging pipe. The homogenous Laplace

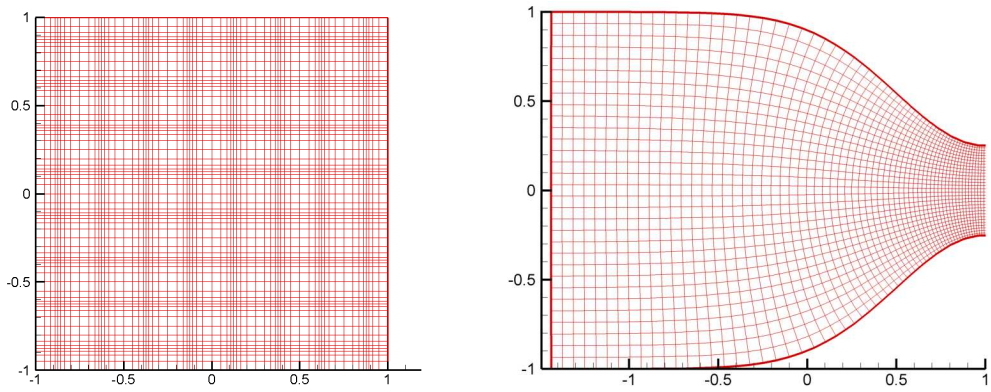


Figure 6.5: Curvilinear grids used for validation : Grid A (left) - Grid B (right)

equation between two circles of radius $R_1 = 0.5$ and $R_2 = 4$ is solved. The solution is $u_1 = 10$ on the first circle and $u_2 = 0$ on the second circle. The boundary condition on the inner circle is imposed with the SMP method and the analytical solution which account for the exterior circle is imposed on the boundary of the Eulerian grid. The position is obtained with the following representations:

- GI: The geometric intersection between the projected interface and the Cartesian grid is used
- LS-CUR: The level-set function for the curvilinear grid is used to locate the interface
- LS: The level-set function for the Cartesian grid is used to locate the interface
- SUS: The Sussman Heaviside function for Cartesian grid is used to locate the interface
- FT: The Front-Tracking projection algorithm for Cartesian grid is used to locate the interface

The Tables (6.7) and (6.7) shows the convergence results for grids A and B. As expected, the more accurate method is the GI. The LS-CUR and LS have a quite similar accuracy. The three functions almost reach a second order in space accuracy for the L^2 and L^∞ error norms. The level-set function calculated on the curvilinear grid is more accurate than the one calculated on the transformed Cartesian grid. However, the level of error is quite similar.

The FT method, which solves an elliptic equation to obtain the color function reach an order 2 in L^2 norm for the first meshes. Then, the performances go down. The same phenomenon occurs for the L^∞ norm where an order of 1.5 is found for the first meshes. An implementation error is perhaps involved. Nevertheless, for the meshes for which the method has a good convergence rate, the level of error is much more higher than for the GI or the LS.

For the SUS method, which uses a smooth Heaviside (6.10) constructed from the level-set function, an average order of 1.55 is found for the grid A and 1.36 for the grid B for the L^2 norm. For the L^∞ norm, the convergence orders are about 0.9 If the Sussman function is replaced by a linear function (6.13), the same results as for the level-set function are retrieved. Hence, the regularity of the Sussman function impacts on the order of convergence[1].

[1] A solution to retrieve a second order of convergence would be to build an *ad hoc* interpolation for the Sussman function.

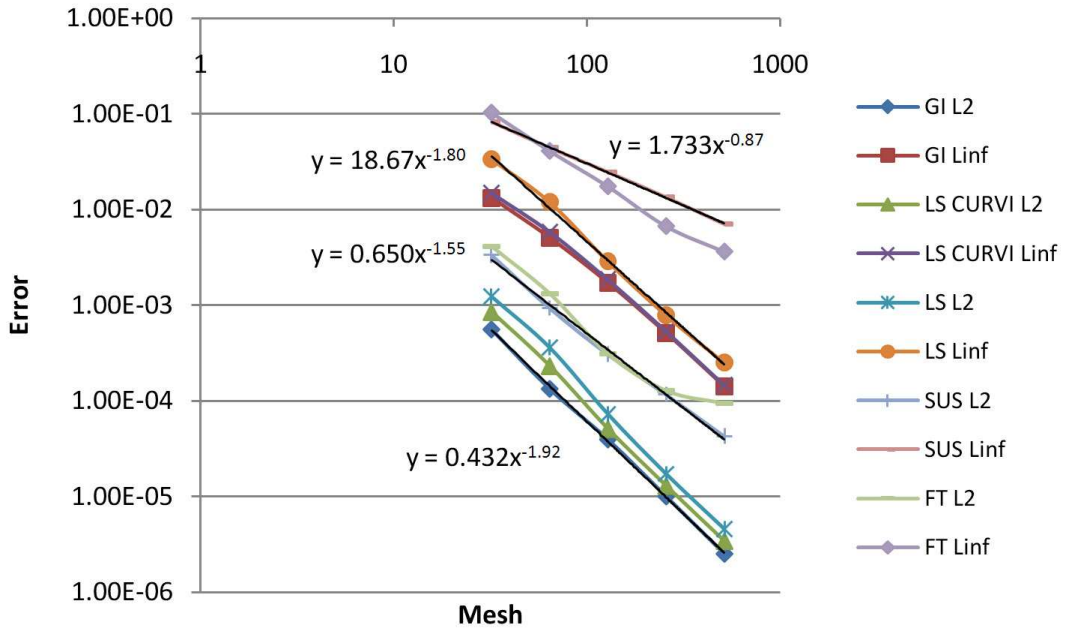


Figure 6.6: Relative L^2 and L^∞ errors for some implicit representations of the interfaces on the Grid A

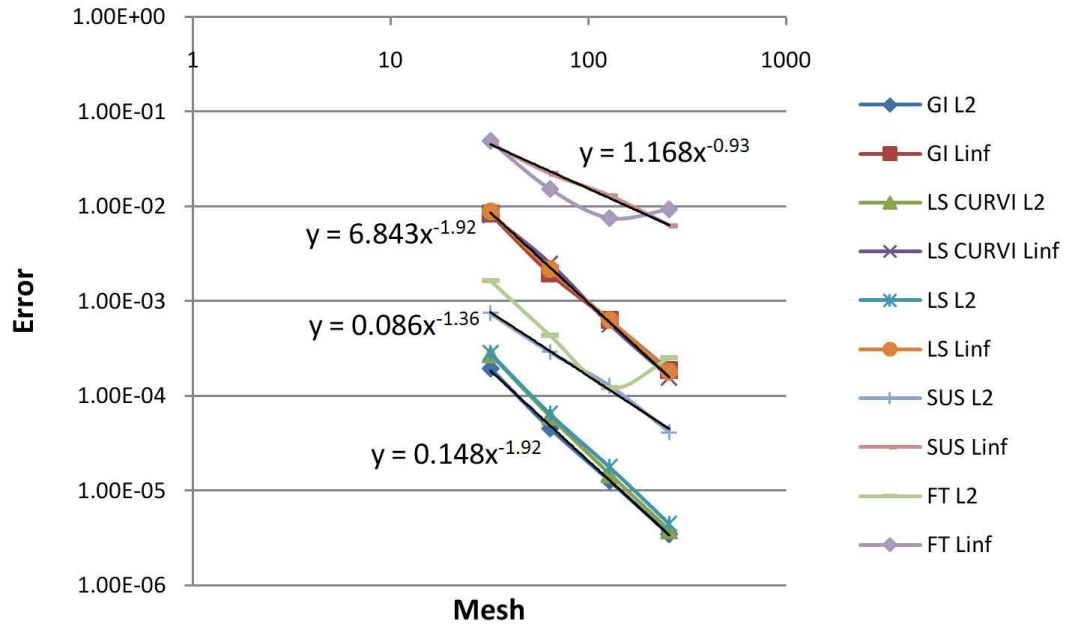


Figure 6.7: Relative L^2 and L^∞ errors for some implicit representations of the interfaces on the Grid B

6.5.1.2 Interface tracking

Description of the curvilinear grid The curvilinear grid used in the two following sections is a converging-diverging or contracted channel used by Friess *et al.* [Frie 04] in their test case number 27 on interface tracking in complex geometries. The grid has been generated with the Computational Fluid Dynamics CFD meshing Gridgen [Chaw 92] from Pointwise which enables to manage the orthogonality of curvilinear grids in an accurate manner. An example of contracted channel is provided in figure 6.8. Its dimensions are $[2.5\,;\,2.5] \times [-1\,;\,1]$ The ability

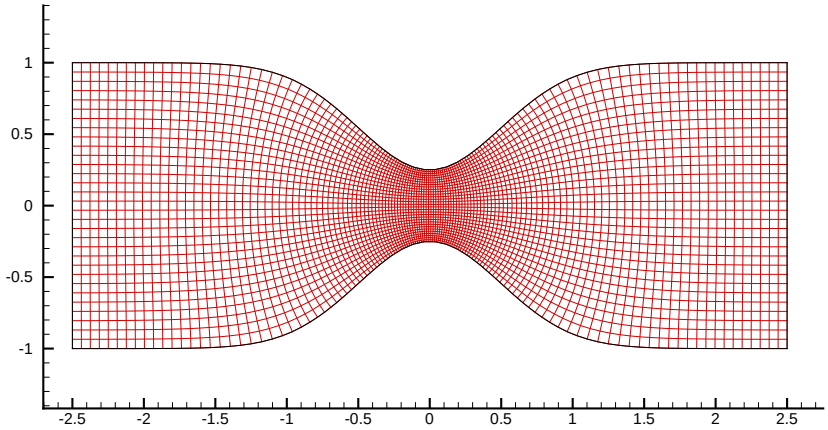


Figure 6.8: The 128×32 contracted channel mesh for the interface tracking cases

of our method to deal with multiphase flow methods is studied. A phase is initialized in the domain. Two different shapes are considered, a circle and a cross (see Fig. 6.9. Convergence study on the volume conservation are first performed. The difference between the initialized and the final volume is calculated. We do not use the analytical volume but the volume initialize as described previously. For each cases, a velocity field is initialised. A certain number of iterations is performed, then the velocity field is inverted and the same number of iterations is performed.

Advection of a shape in a horizontal velocity field (Field A)
The phase is advected with a constant velocity in the x-direction. In the transformed space, the velocity is not null for its two components. A circle of radius $0.225m$ centered in $(-1.45, 0)$ is first considered. For the second case, a cross of width $0.225m$ centered in $(-1.45, 0)$ is used.

The Fig. 6.12 shows the convergence in mesh of the error on the volume conservation for the four advection methods on the field A for the circle and the cross cases. The Fig. 6.11 shows the implicit surfaces reconstructed at the end of the simulations.

Advection of a shape in a parabolic velocity field (Field B)
For this case, the streamlines follows the mesh lines so the velocity field is null in the y-direction in the transformed Cartesian frame and the field is sheared. A circle of radius $0.5m$ centered in $(-1.95, 0)$ is first considered. For the second case, a cross of width $0.5m$ centered in $(-1.95, 0)$ is used.

The Fig. 6.13 shows the convergence in mesh of the error on the volume conservation for the four advection methods on the field B for the circle and the cross cases. The Fig. 6.11 shows the

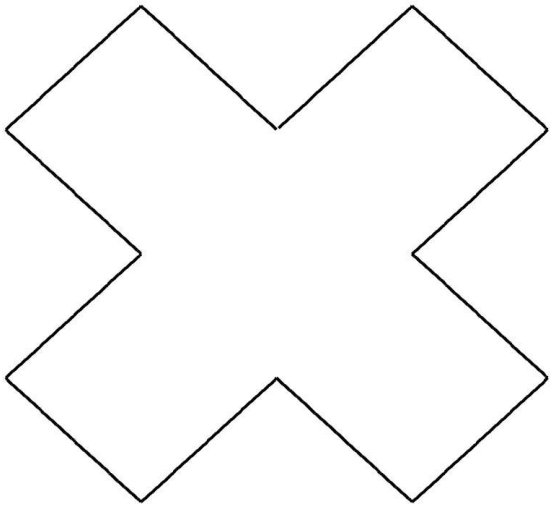


Figure 6.9: The Lagrangian shape of the cross for the curvilinear advection tsets

implicit surfaces reconstructed at the end of the simulations.

Discussion

The convergence of the error for the Front-Tracking is regular except for the case of the cross for the B field and reach a second order. The convergence of the Level-Set is less regular, especially for the circle. The LS always reach a second order, and more with the circle. For the field A, the LS is less accurate than the FT. The inverse is observed with the field B. The convergence of the VOF-PLIC and the VOF-TVD methods is more irregular. For the field A, the error is always decreasing except for one value for the TVD) and globally shows good performances compared to the other methods, especially the Level-Set. For the field B, even if the error levels are quite good compared to the other methods, the error is almost never decreasing with the mesh. Concerning the computational time for these cases, the VOF-PLIC and VOF-TVD methods are about five time faster than the Front-tracking and the Level-set. Concerning the Front-tracking, its computational cost depends directly on the number of Lagrangian elements.

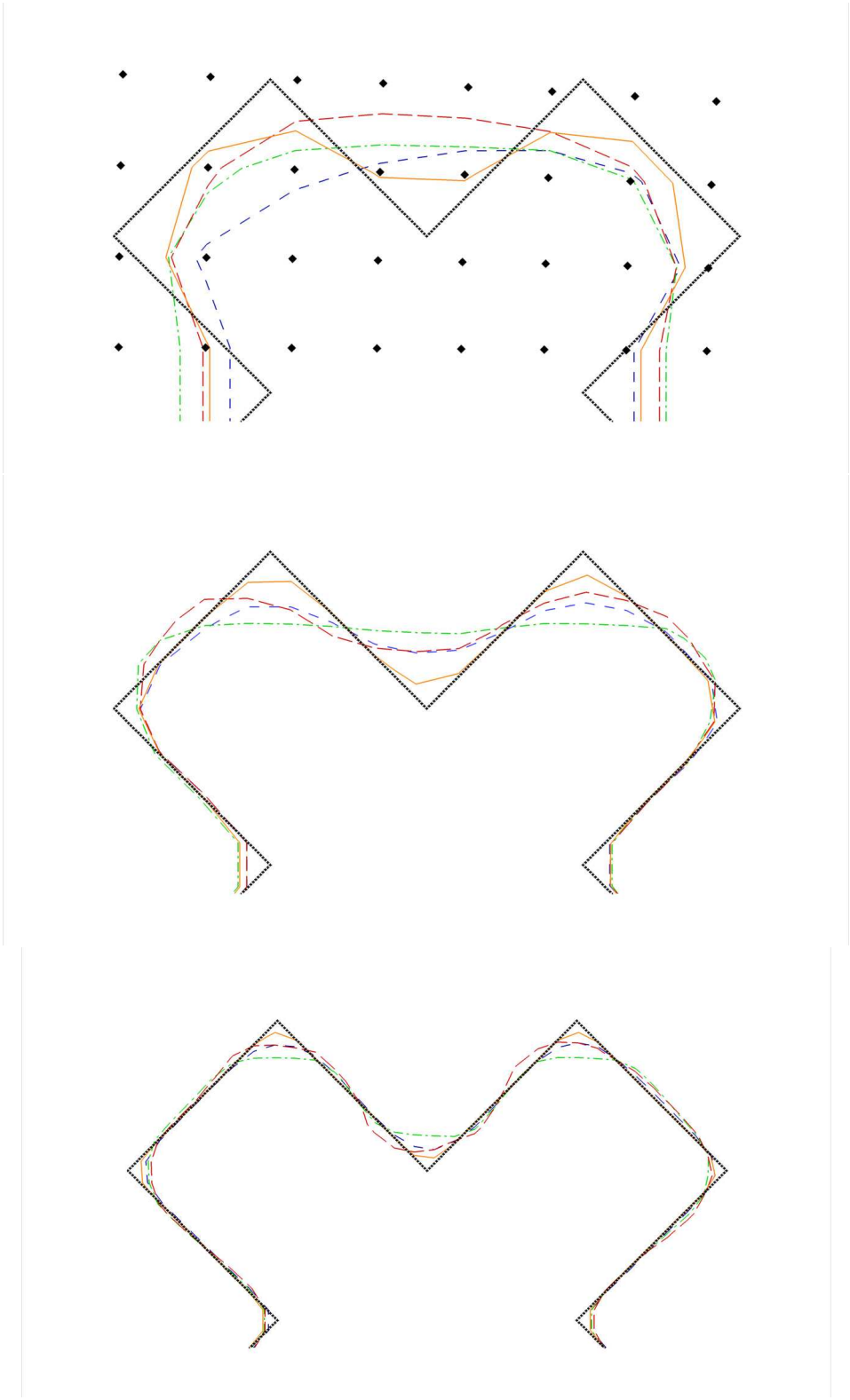


Figure 6.10: Iso-lines of the final position of the phase for the FT (solid), LS (dashed), TVD(dotted) and PLIC (long-dashed) for 128×32 (up, with the Eulerian mesh nodes), 256×64 (middle) and 512×128 (down, details) meshes on the field A

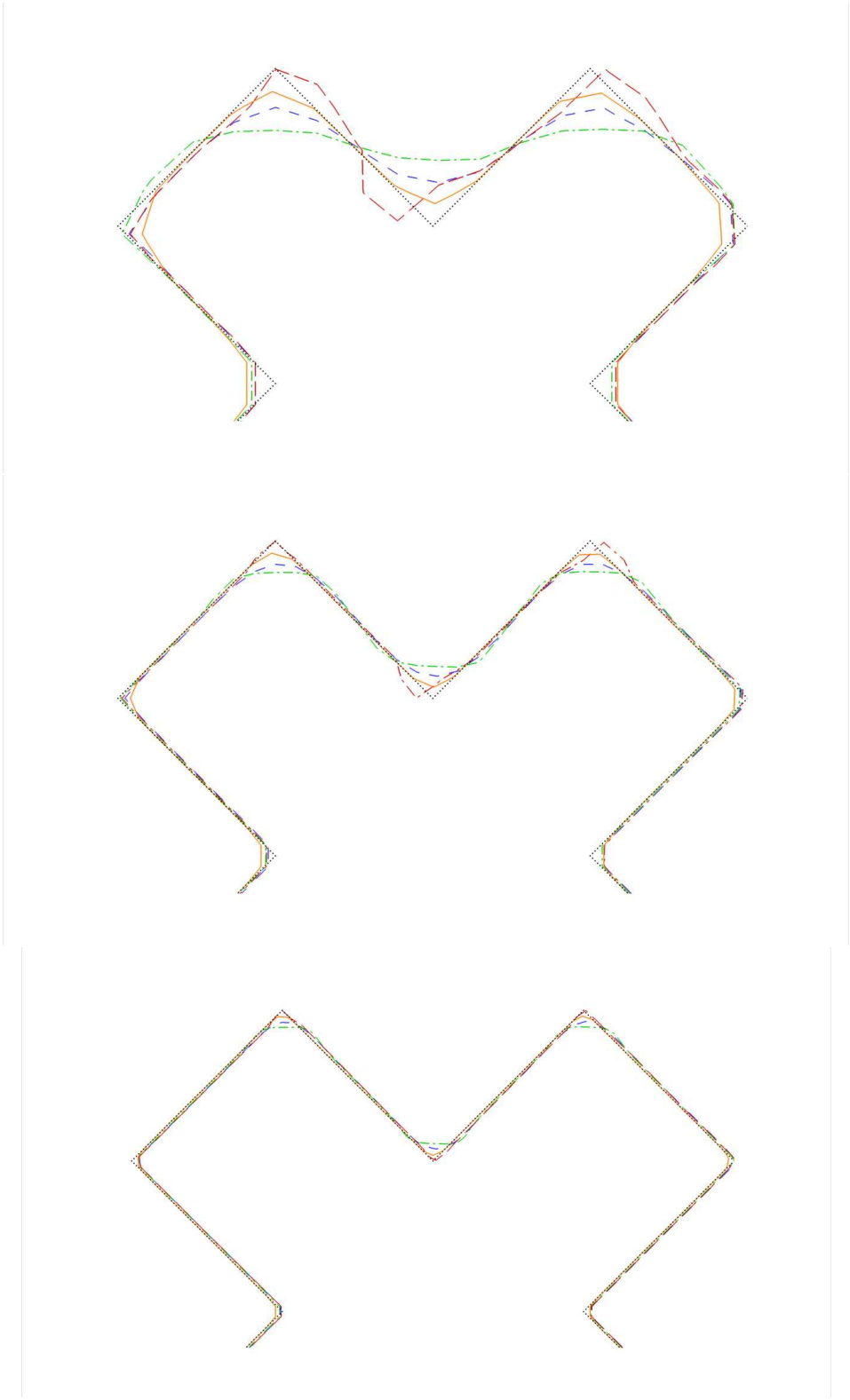


Figure 6.11: Iso-lines of the final position of the phase for the FT (solid), LS (dashed), TVD(dotted) and PLIC (long-dashed) for 128×32 (up), 256×64 (middle) and 512×128 (down, details) meshes on the field B

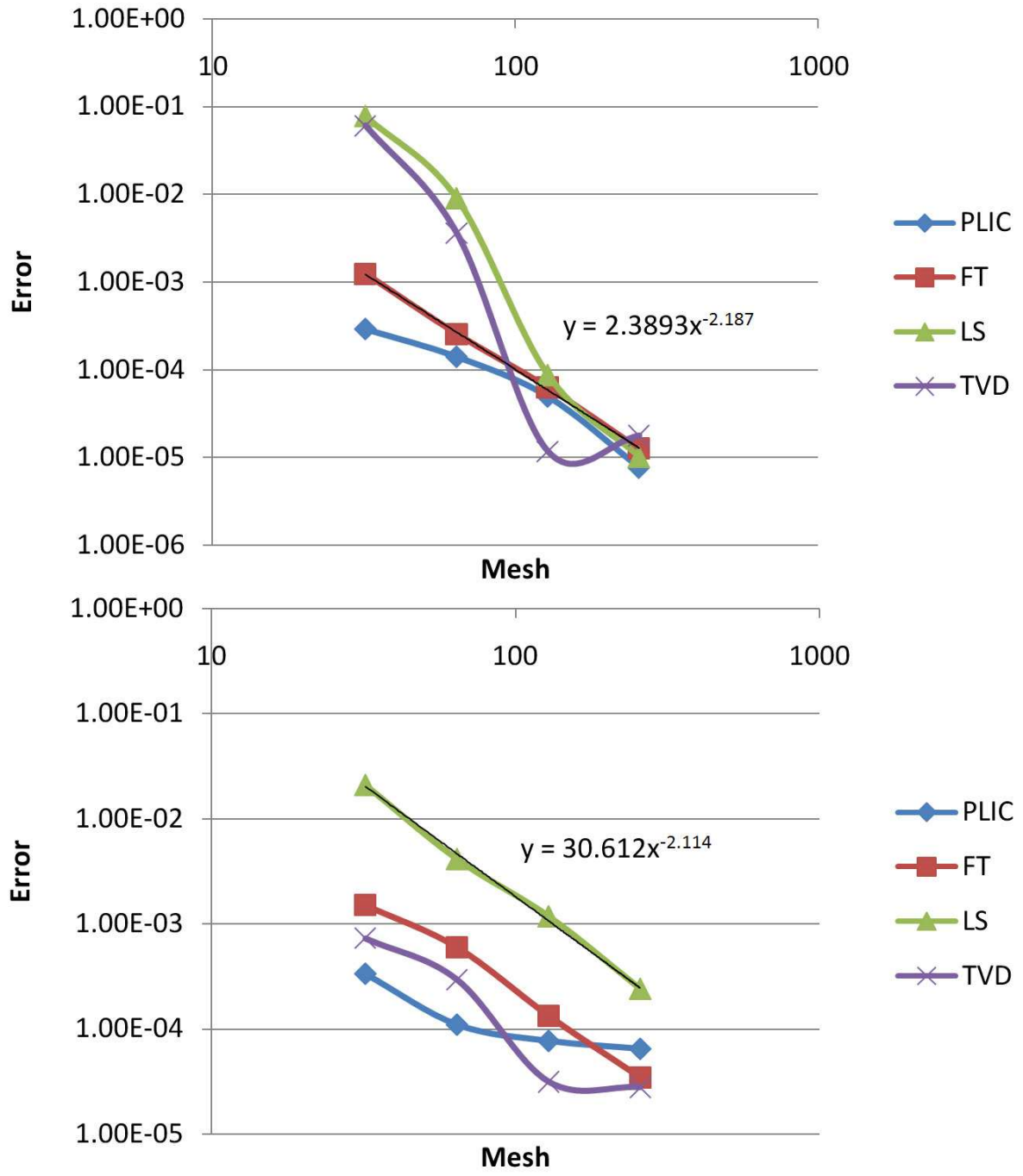


Figure 6.12: Convergence of the error on the volume conservation for the four advection methods on the field A for the circle case (up) and the cross case (down)

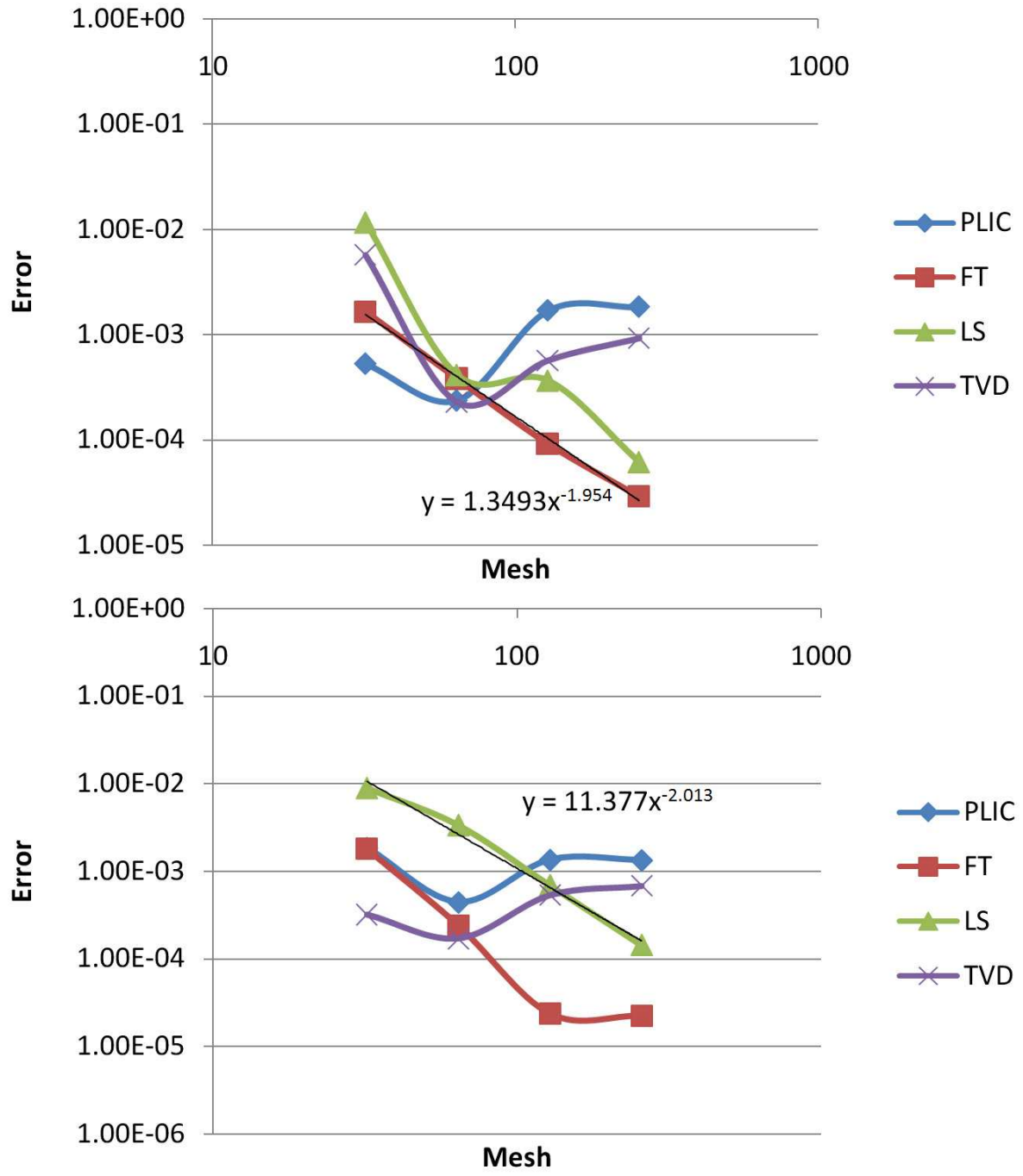


Figure 6.13: Convergence of the error on the volume conservation for the four advection methods on the field B for the circle case (up) and the cross case (down)

Concerning the iso-lines, the first mesh (128×32) with the field A shows that the cross is not accurately retrieved by the four methods, but as the Lagrangian mesh takes a small part of the domain, the Eulerian mesh is comparatively very coarse in this case. However, one can see that the FT method produces the best result. The Fig. 6.14 shows the Lagrangian shape managed by the FT method in the transformed space (explaining the slight twisting of the cross). As can be seen, the cross is very accurately retrieved, and the lack of precision of the final result is only

due to the projection [Shin 02b]. It can be observed on Fig. 6.15 which shows the horizontal advection of the cross on a 256×64 with the FT method. Five field are superimposed. The extreme-left cross is obtained for the first time while the extreme-right one is obtained after. The two extreme fields seems to be identical. Due to the refinement of the Eulerian mesh in the central part, the middle cross (intermediate time) describes the interface with more accuracy. It would be interesting to calculate the volume conservation using the Lagrangian mesh instead of the Eulerian projection.

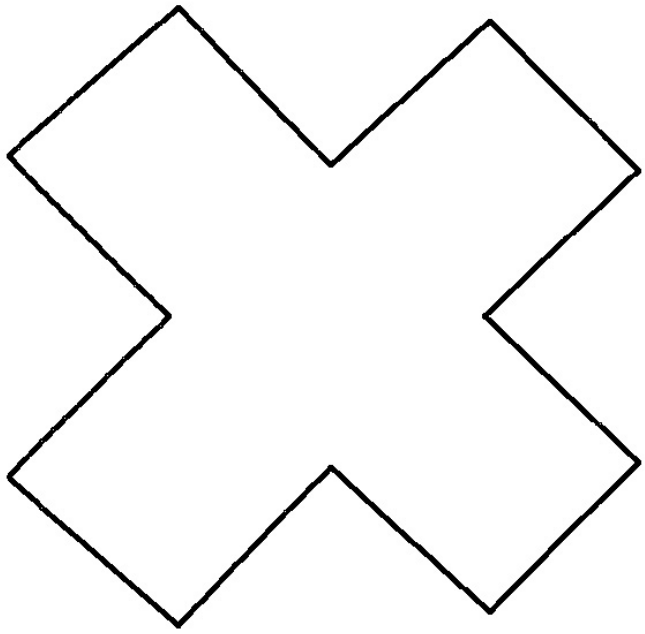


Figure 6.14: Lagrangian mesh managed by the FT method in the transformed space for a 128×32 mesh

On finest meshes, the Front-tracking gives a very regular result which is slightly smoothed by the projection. The LS gives a less regular result while the VOF-TVD smoothes a lot the lower and upper corners. For the second case on the filed B, the initial size of the cross is two times larger. The simulations show the good performances of the VOF-PLIC method. The VOF-TVD and Level-set methods show quite good results. However, one can see on the 512×128 mesh that the LS is less efficient on some corners. Concerning the FT, the method produces the more regular results which is one time again smoothed by the projection. As this latter produce a less accurate implicit interface than the Level-set, the VOF method retrieves the shape of the interface more accurately. However, the FT method shows a better volume conservation than the other methods for this case (Fig. 6.13).

6.5.2 Optimisation using a octree data structure

An octree is a tree data structure in which each internal node has up to eight children. Octrees are used in the present work to partition the 3D space by recursively subdividing it into eight octants. In 2D, the equivalent is a quadtree where the 2D space is recursively subdivided into quadrants.

To build the octree, the space is first divided in eight octants, generally eight boxes. Each

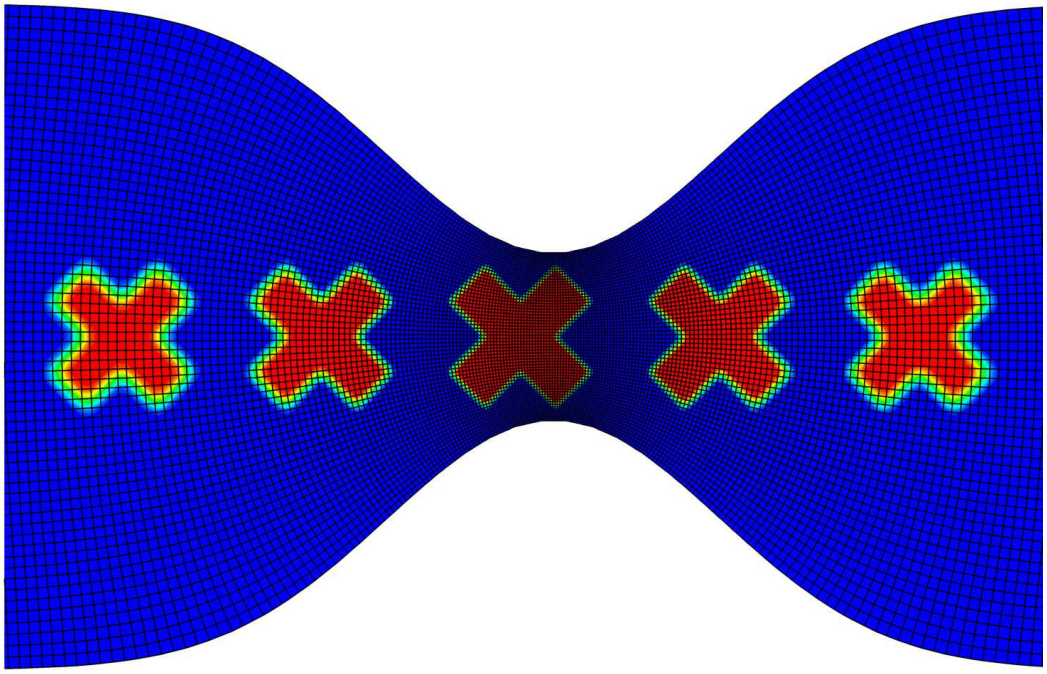


Figure 6.15: Advected phase for the field A on a 256×64 mesh for various time steps

triangle of the mesh is classified according to its belonging to one octant or more. Then, for each octant, the same process is repeated recursively. The depth of subdivision depends on the complexity of the mesh and is not the same for each part of the mesh (each leaf of the octree have not necessarily the same depth). Generally, the construction is stopped for a given octant if it contains a low limit number of elements. If an octant of the initial bounding box does not contain any elements, it is *a priori* useless to subdivide again this octant. However, some implementations are based on well-balanced trees and all leaves must have the same depth.

To sort a cloud of points, a kD-tree structure is generally used and allows to find quickly the closest point to an other . Here, the space is recursively cut into two sub-spaces by a median plan. Once again, the number of points in the final subdivisions is crucial for the performance. The Fig. 6.16 shows, for a random field of points, the average time to find the closest neighbor of a random point with respect to the size of the field and the number of points in the smallest sub-divisions. As can be seen, the optimal size for the smallest sub-divisions is about 20 points while the performances fall dramatically if too few elements are present in the smallest sub-divisions. These results can of-course vary according to the implementation choices.

This space subdivisions allows many optimisation of the previous algorithms to be performed and generally modify the complexity of a spatial search from $\mathcal{O}(n)$ to $\mathcal{O}(\log n)$.

6.5.2.1 Application to the curvilinear to Cartesian algorithm

In this algorithm, one have to find inside which Eulerian cell each vertex of the Lagrangian mesh is. The nodes of the Eulerian mesh are sorted with a kD-tree. Then, the closest pair of Eulerian-Lagrangian nodes is found and the belonging of the Lagrangian node to the cells containing the Eulerian cell is performed.

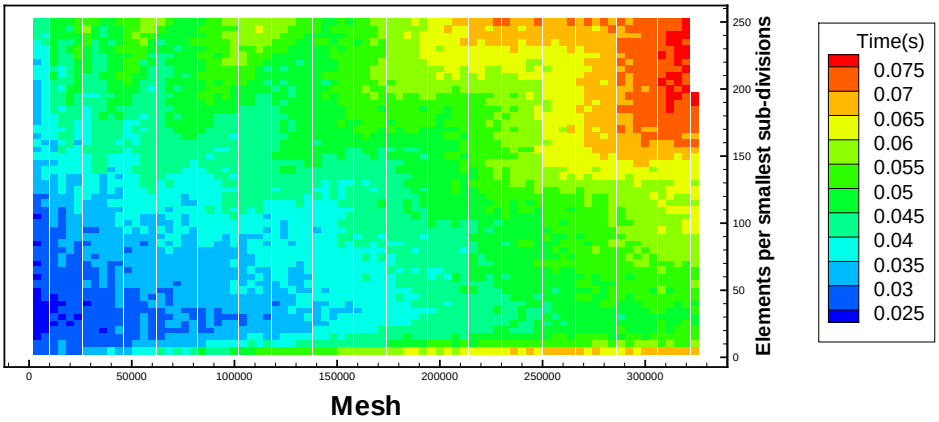


Figure 6.16: Time in second to find a closest point with respect to the size of the field and the number of points in the smallest sub-divisions for a kD-tree

6.5.2.2 Application to the Ray-casting algorithm

The basic optimisation of the Ray-casting algorithm is to test the intersection between a ray and a box bounding the object. One can use the octree to perform recursively this test. The octree gives boxes bounding a set of elements of the object, so one can disqualified such a set of elements by testing the intersection between a ray and their bounding box. If an intersection is detected, the ray will possibly intersect an element of the set, and intersection tests are performed with the eight octants which compose the previous octant.

6.5.2.3 Application to the Level-set algorithm

As the location of the elements inside a given octant is generally irregular, the closest element to a node in an octant is not necessarily inside this octant. Hence, for an Eulerian node in a given octant, the closest element to the node is searched in the octant and its neighbors.

6.5.3 Performance tests

Speed tests have been performed on a P4 2.4 GHz for several meshes with and without the octree. Three Lagrangian meshes are used, a sphere (18000 triangles, regular), the Stanford bunny (10122 triangles, irregular) and the Lascaux cave (271136 triangles, irregular. See section 14). The Tab. 6.1 shows the performances of the octree for these three meshes and a 100^3 Eulerian mesh. As expected, all the routines are faster with the octree. The gain for the curvilinear-Cartesian projection is between 1.35 and 1.73 only. However, the computational cost of this algorithm is negligible. For the Ray-casting, the gain is smaller than 2 on small meshes, and more than 10 for the Lascaux cave, where a deeper octree is used. For the computation of the Level-set, the gain ratio is from 35 to 75.

One can notice that as the size of the Eulerian grid is 100^3, the time ratio between the standard Ray-casting (not tested here) and the Thread Ray-casting would be about 100.

The most important result is that the maximum computational time for the whole method is shorter than a minute on a P4 2.4 GHz. This time is negligible against the computational time required to solve the linear system resulting from the discretization of a conservation equation in a 100^3 mesh.

Mesh	Method	curvi-Cart projection	Ray-casting	Level-set
Sphere	Standard	2.06	0.536	270
	Optimized	1.37	0.332	5.79
	Ratio	1.50	1.61	46.6
Bunny	Standard	1.19	0.172	123
	Optimized	0.88	0.132	1.63
	Ratio	1.35	1.30	75.5
Lascaux	Standard	45.4	35.9	1970
	Optimized	26.1	3.32	56.1
	Ratio	1.73	10.8	35.1

Table 6.1: Duration in second and performance ratio for three different meshes with and without octree

Discussion and conclusion of Part II

As can be seen, dealing with immersed interfaces or boundaries is not trivial when the considered shapes are not analytical. An efficient strategy for the Eulerian-Lagrangian grid coupling has been devised here. This methodology works for curvilinear grids thanks to a curvilinear to Cartesian projection, and the computational cost of this phase is negligible and is fully counterbalanced by the possibility to use the Thread Ray-casting method. For a 3D mesh composed by M^3 elements, the computational cost of the Ray-casting is divided by M.

The construction of some implicit representations of a surface has been presented. The accuracy of these methods has been compared with a fictitious domain test case. For the multiphase flows, the classic methods (VOF-PLIC, VOF-TVD, Level-Set, Front-Tracking) can be used without modification (for the FT, an implementation error seems to be involved). However, the unusual convergency results suggests that a more deep study has to be performed.

For moving objects and fluid-structure coupling, the initial mesh of the object can be used for each time step if the methods are optimized enough. For such objects, several approaches use VOF or Level-set functions. As our approach is based on the use of Lagrangian meshes, it is *de facto* the more accurate as the exact shape is always conserved. The case of moving object will be detailed in the Part IV. Concerning the computational cost of our approach, it would be interesting to compare it with the VOF and Level-set methods. However, the objective of any method is to reach a computational cost which is negligible in comparison to the solver cost. This aim can be reached with our Lagrangian mesh approach if enough efforts are putted into the optimization of the involved algorithms. The Thread Ray-casting as well as the octree and kD-tree data structures have shown their ability to highly decrease the cost of such operations. Furthermore, these algorithms are highly and simply parallelizable and a huge further gain can be obtained using GPUs or multicore CPUs.

This methodology is a prerequisite for many immersed boundary and interface methods which require to know the exterior and the interior of an object. As demonstrated in this part, the implicit representation as well as the explicit representation of a surface can be used to build fictitious domain method. The next part, which deal with such methods, will use the present methodology to enhance the resolution of the conservation equations near interfaces and boundaries.

Part IV

High-order fictitious domain methods

Table of Contents

Introduction

In this part, two high-order fictitious domain methods are presented. The first, the sub-mesh penalty method (SMP), is the extension to higher orders of the VPM [Ango 89]. The second, the algebraic immersed interface and boundary (AIIB) method, can be seen as an extension of the SMP method to the augmented system approach (see section 4.1.2). Algebraically, the penalty methods, more than penalizing equations, penalizes matrix lines. The AIIB method proposes to add new lines to the inverted matrix. This approach allows to treat immersed interface problems, when SMP method can treat immersed boundary problems only. These two methods are designated as high-order method as they are of second order in space for various problems and they can be extended to higher order for these problems straightforwardly.

Definitions and notations

Let us consider the original domain of interest denoted by Ω_0, typically the fluid domain, which is embedded inside a simple computational domain $\Omega \subset \mathbb{R}^d$, d being the spatial dimension of the problem. The auxiliary domain Ω_1, typically a solid particle or an obstacle, is such that : $\Omega = \Omega_0 \cup \Sigma \cup \Omega_1$ where Σ is an immersed interface (see Fig. 6.17). Let $\mathbf{n}$ be the unit outward normal vector to Ω_0 on Σ. Our objective is to numerically impose the adequate boundary conditions on the interface Σ. These conditions will be discretized in space on an Eulerian structured mesh covering Ω.

The computational domain Ω is approximated with a curvilinear mesh T_h composed of $N \times M$

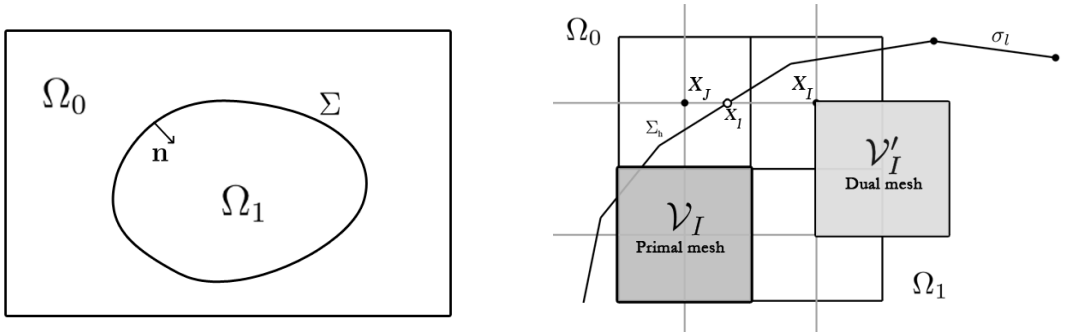


Figure 6.17: Definition of the domains and discretization kernels

($\times L$ in 3D) cell-centered finite volumes ($\mathcal{V}_I$) for $I \in \mathcal{E}$, $\mathcal{E}$ being the set of index of the Eulerian orthogonal curvilinear structured mesh. Let x_I be the vector coordinates of the center of each volume $\mathcal{V}_I$. In 2D, the horizontal and vertical mesh steps are respectively h_x and h_y This grid is used to discretized the conservation equations. A dual grid is introduced for the management of the AIIB method. The grid lines of this dual cell-vertex mesh are defined by the network of the cell centers x_I. The volumes of the dual mesh are denoted by ($\mathcal{V}'_I$). The Eulerian unknowns are noted u_I which are the approximated values of $u(x_I)$, i.e. the solution at the cell centers x_I. The discrete interface Σ_h, hereafter called the Lagrangian mesh, is given by a discretization of the original interface Σ. It is described by a piecewise linear approximation of Σ : $\Sigma_h = \{\sigma_l \subset \mathbb{R}^{d-1}, l \in \mathcal{L}_f\}$, $\mathcal{L}_f$ being the set of index of the Lagrangian mesh and K being the cardinal of $\mathcal{L}_f$. Typically, σ_l are segments in 2D and triangles in 3D. The vertices of each face σ_l are denoted by $x_{l,i}$ for $i = 1, d$ and the set of all vertices is $\{x_l, l \in \mathcal{L}_v\}$. The intersection points between the grid lines of the Eulerian dual mesh and the faces σ_l of the Lagrangian mesh are denoted by

$\{x_i, i \in \mathcal{I}\}$ (see Fig. 6.17). Our objective is to discretize Dirichlet, Neumann, transmission and jump conditions at these interface points to build a general fictitious domain approach. This method is expected to reach a global second-order spatial accuracy.
New sets of Eulerian points x_I are defined near the interface so that each one has a neighbor x_J verifying $\chi_J \neq \chi_I$ (with $\chi_I = \chi(x_I)$ and $\chi_J = \chi(x_J)$), *i. e.* the segment $[x_I; x_J]$ is cut by Σ_h. These Eulerian "interface" points are also sorted according to their location inside Ω_0 or Ω_1. Two sets $\{x_I, I \in \mathcal{N}_0\}$ and $\{x_I, I \in \mathcal{N}_1\}$ are thus obtained, where $\mathcal{N}_0 = \{I, x_I \in \Omega_0, \chi_I \neq \chi_J, x_J \in \Omega_1\}$ and $\mathcal{N}_1 = \{I, x_I \in \Omega_1, \chi_I \neq \chi_J, x_J \in \Omega_0\}$.
For each x_I, $I \in \mathcal{N}^0$ or $I \in \mathcal{N}^1$, we associate two unknowns : the *physical* one denoted as u_I and the *auxiliary* one u_I^*.

The various interpolations used in the present document, $\mathbb{L}_1^2$, $\mathbb{P}_1^2$ and $\mathbb{Q}_1^2$, are described in section C.

Chapter 7

The Sub-Mesh Penalty method

7.1 Principle of the method for a scalar equation

7.1.1 1D method for a Dirichlet boundary condition

The application field of the sub-mesh penalty (SMP) method concerns the problems with a Dirichlet or Neumann BC on an immersed interface. As exposed before, such problems can be treated with many existing methods. The SMP method is the first discretization of the L^2 penalization to reach a general second order of accuracy in space. The lack of accuracy of the first-order discretizations is primarily due to their approximative treatment of the interface. Such methods impose the interface solution u_l on whole control volumes, so the solution across the interface is piecewise constant. Hence, the shape of the interface Σ is rasterized, *i.e.* its shape is composed of segments which are oriented in the principal directions (Ox, Oy and Oz) only. The term rasterization comes from computer graphics, where any picture is generally composed of squared pixel and is therefore approximated.

To retrieve a more accurate shape of the interface, one had to impose the correct interface value u_l at the location x_l of a piecewise linear reconstruction of the original interface. The generic first-order penalty term for a node x_I in the immersed domain is $\frac{\chi}{\varepsilon}(u - u_l)$ with χ the indicator of Ω_1. To reach higher orders, the interpolation of the solution has to be considered instead of the solution at the discrete points. To impose that the interpolated approximation of the solution had to take the value u_l in x_l, the penalty term $\frac{\chi}{\varepsilon}(u - u_l)$ is discretized at x_I by

$$\frac{\chi_I}{\varepsilon}\Big(\sum_{k\in\mathcal{N}_0\cup\mathcal{N}_1}\alpha_k u_k - u_l\Big). \tag{7.1}$$

As can be seen, the constraint is a linear combination of u_I and of the solution near the interface. Practically, the nodes x_k are in the discrete neighborhood of x_I.

First, let us study the 1D case for the following scalar model problem:

$$\begin{cases} -\nabla\cdot(a\nabla u) = f & \text{in } \Omega_0 \\ u_{|\Sigma} = u_D & \text{on } \Sigma \end{cases} \tag{7.2}$$

Let $\Omega = [0;1]$ be the computational domain discretized in N control volumes of measure h_x. Let Σ be an immersed interface located at x_l separating the domain in two subdomains Ω_0 and Ω_1. The solution in Ω_1 is a constant u_D, and an associated Dirichlet BC is imposed on Σ. Let us consider two nodes I and J located at x_I and x_J, with $x_I < x_l < x_J$, $x_I \in \Omega_0$ and $x_J \in \Omega_1$. The first-order penalty constraint which can be used to impose the solution in Ω_1 is simply $u_I = u_D$.

This constant constraint is replaced by a linear one:

$$\frac{|x_J - x_l|u_I + |x_I - x_l|u_J}{h_x} = u_D \tag{7.3}$$

The solutions u_I and u_J are not directly imposed, but the linear interpolation of the solution between x_I and x_J will be equal to u_D, no matter what values u_I and u_J take. We can now define the high-order penalty term in x_J and modify the original Laplace equation:

$$\{\nabla^2 u\}_J + \frac{\chi_J}{\varepsilon}(\frac{|x_J - x_l|u_I + |x_I - x_l|u_J}{h_x} - u_D) = 0 \tag{7.4}$$

where $\{.\}_I$ denotes the discretization of a quantity at the location x_I. The resolution of a first problem is now detailed to simply expose some properties and implications of the method. Let us consider the following problem in $\Omega = [0;1]$:

$$\begin{cases} \nabla^2 u(x) = 0 \text{ on } \Omega \\ u(0) = T_1 \\ u(x_l) = T_2 \\ u(1) = T_3 \end{cases} \tag{7.5}$$

with its solution:

$$u(x) = \begin{cases} \dfrac{T_2 - T_1}{x_l}x + T_1 - (T_2 - T_1)\dfrac{R_1}{x_l} \text{ if } 0 \leq x \leq x_l \\ \dfrac{T_3 - T_2}{1 - x_l}x + T_2 - (T_3 - T_2)\dfrac{R_2}{1 - x_l} \text{ if } x_l \leq x \leq 1 \end{cases} \tag{7.6}$$

The solution is piecewise linear. As x_J is the node of Ω_1 which is the closest to Σ, the penalty term is activated for the discretization of the conservation equation at this node. The constraint is defined as:

$$\alpha_I u_I + \alpha_{I+1} u_{I+1} = T_2 \tag{7.7}$$

and the initial matrix used to discretize the Laplace problem is now:

$$A = \begin{pmatrix} 1 & & & & & & & & \\ 1/h_x & -2/h_x & 1/h_x & & & & & & \\ & & \ddots & & & & & 0 & \\ & & 1/h_x & -2/h_x & 1/h_x & & & & \\ & & & 1/h_x + \frac{1}{\varepsilon}\alpha_I & 1/h_x + \frac{1}{\varepsilon}\alpha_{I+1} & 1/h_x & & & \\ & & & & 1/h_x & \frac{1}{\varepsilon} - 2/h_x & 1/h_x & & \\ & 0 & & & & & \ddots & & \\ & & & & & & 1/h_x & \frac{1}{\varepsilon} - 2/h_x & 1/h_x \\ & & & & & & & & 1 \end{pmatrix} \tag{7.8}$$

with the following second member: $\mathbf{b} = \left(T_1, 0 \cdots 0, \frac{1}{\varepsilon}T_2, \frac{1}{\varepsilon}T_2 \cdots \frac{1}{\varepsilon}T_2, \frac{1}{\varepsilon}T_2\right)^T$. Except in x_J, the first-order discretization is still used inside Ω_1. The parameter ε is chosen such as $\varepsilon \ll 1$ so the limit penalized problem is considered. The penalty term is always active for the nodes in Ω_1 and the computer accuracy totally erase the terms of the initial equation. Hence for these nodes, the

terms $1/h_x$ are negligible. Consequently, an independent sub-matrix can be extracted. Let us consider this upper submatrix denoted as A_1 and its associated second member:

$$A_1 = \begin{pmatrix} 1 & & & & \\ 1/h_x & -2/h_x & 1/h_x & 0 & \\ & & \ddots & & \\ & 0 & 1/h_x & -2/h_x & 1/h_x \\ & & & \frac{1}{\varepsilon}\alpha_I & \frac{1}{\varepsilon}\alpha_{I+1} \end{pmatrix} \quad \mathbf{b}_1 = \begin{pmatrix} T_1 \\ 0 \\ \vdots \\ 0 \\ \frac{1}{\varepsilon}T_2 \end{pmatrix} \tag{7.9}$$

where $\mathbf{u}_1$ is the part of $\mathbf{u}$ related to Ω_0 and the penalized node of Ω_1. The new problem $A_1\mathbf{u}_1 = \mathbf{b}_1$ can be solved independently. Hence, the solution in Ω_1, far from the interface, does not impact on the solution in Ω_0 where the solution is not known *a priori*.

Concerning the properties of the matrix, one can notice that if $|\alpha_I| \geq |\alpha_{I+1}|$ the matrix looses its diagonal dominance, and its invertibility is no more easy to deduce. Practically, any matrix with a penalization term built with the SMP method can be inverted with an appropriate solver such as a BiCG-Stab with an ILU preconditioner.

7.1.2 General method for a Dirichlet boundary condition

Let us now describe the 2D SMP method for the model scalar problem (7.2) with a Dirichlet boundary condition on the interface Σ. Let us consider a point $x_I, I \in \mathcal{N}_1$. We first describe the case when x_I has only one neighbor x_J in Ω_0. The Lagrangian point x_l is the intersection between $[x_I; x_J]$ and Σ_h (Fig. 6.17 right). Then, the solution $u_l = u_D(x_l)$ at the interface is approximated by the $\mathbb{P}_1^1$ interpolation between the Eulerian unknowns u_I and u_J :

$$u_l = \alpha_I u_I + \alpha_J u_J \text{ with } 0 < \alpha_I, \alpha_J < 1 \text{ and } \alpha_I + \alpha_J = 1 \tag{7.10}$$

As noticed in [Tsen 03, Gibo 05], a linear interpolation only is required to reach a second order of accuracy. If now x_I has a second neighbor x_K in Ω_0, the intersection x_m between $[x_I; x_K]$ and Σ_h is considered with $u_m = u_D(x_m)$. We choose x_p, a new point of Σ_h between x_l and x_m (see Fig. 8.1 left). The solution $u_p = u_D(x_p)$ is then imposed using a $\mathbb{P}_1^2$-interpolation of the values u_I, u_J and u_K :

$$u_p = \alpha_I u_I + \alpha_J u_J + \alpha_K u_K \ , \ 0 < \alpha_I, \alpha_J, \alpha_K < 1 \ , \ \alpha_I + \alpha_J + \alpha_K = 1 \tag{7.11}$$

A $\mathbb{Q}_1^2$ interpolation of u_I, u_J, u_K and u_L can be also used by extending the interpolation stencil with the point x_L which is the fourth point of the cell of the dual mesh defined by x_I, x_J and x_K (see Fig. 8.1 left). As a third choice, two independent linear 1D interpolations are first considered (one for each direction). It produces :

$$\begin{cases} u_l = \alpha_I u_I^* + \alpha_J u_J \text{ with } 0 < \alpha_I, \alpha_J < 1 \text{ and } \alpha_I + \alpha_J = 1 \\ u_m = \alpha_I' u_I^{*'} + \alpha_K u_K \text{ with } 0 < \alpha_I', \alpha_K < 1 \text{ and } \alpha_I' + \alpha_J = 1 \end{cases} \tag{7.12}$$

Then, a simple choice for x_p is the barycenter between x_l and x_m where $u_p = (u_l + u_m)/2$. This particular case enables an easy implementation since we have :

$$\alpha_I u_I + \alpha_J u_J = u_l \tag{7.13}$$

$$\alpha_I' u_I + \alpha_K u_K = u_m \tag{7.14}$$

A summation of these two constraints gives :

$$\alpha_I u_I + \alpha_J u_J + \alpha_I' u_I + \alpha_K u_K = u_l + u_m \tag{7.15}$$

what is equivalent to build a constraint imposing u_p at x_p with a $\mathbb{P}_1^2$ interpolation :

$$\frac{(\alpha_I+\alpha'_I)u_I+\alpha_J u_J+\alpha_K u_K}{2}=u_p \ ,$$
$$\text{with } 0<\frac{\alpha_I+\alpha'_I}{2},\frac{\alpha_J}{2},\frac{\alpha_K}{2}<1 \ , \ \frac{\alpha_I+\alpha'_I}{2}+\frac{\alpha_J}{2}+\frac{\alpha_K}{2}=1 \tag{7.16}$$

Hence, an easy general implementation consists in summing the constraints corresponding to each direction, no matter the number of neighbors of x_I. If the elements σ_l of Σ_h used to define x_l and x_m are not the same, the barycenter x_p of these two points is not necessarily on Σ_h, especially for interfaces of strong curvature. However, the distance $d(x_p,\Sigma_h)$ between x_p and Σ_h varies like $\mathcal{O}(h^2)$ and so this additional error does not spoil the second-order precision of our discretization. The convergence of this additional error is numerically tested in section (9.1.2). If the curvature of Σ_h is small enough relatively to the Eulerian mesh, *i.e.* if the Eulerian mesh is sufficiently fine, x_I almost never has a third or a fourth neighbor in Ω_0. However, if this case appears, a simple constraint $u_I=u_B$ is used with u_B being an average of u_D at the neighbor intersection points. In any case, by decreasing the Eulerian mesh step h, the number of points x_I having more than two neighbors in Ω_0 also decreases.
Hence, the present method is suitable to impose a Dirichlet boundary condition on Σ for Ω_0, when the solution in Ω_1 has no interest.

7.1.3 General method for a Neumann boundary condition

Let us now consider the following model scalar problem with a Neumann BC on the interface Σ :

$$\begin{cases} -\nabla\cdot(a\nabla u)=f & \text{in } \Omega_0 \\ (a\cdot\nabla u)\cdot\mathbf{n}=g_N & \text{on } \Sigma \end{cases} \tag{7.17}$$

The principle is about the same as for Dirichlet BC, and the same interpolations, once derived, can be used to approximate the quantity $(a\cdot\nabla u)\cdot\mathbf{n}$. Hence, at any point $x_l, l\in\mathcal{I}$ on Σ_h we use

$$(a\cdot\nabla u_l)\cdot\mathbf{n}\approx(a\cdot\nabla p(x_l)\cdot\mathbf{n}). \tag{7.18}$$

For $p\in\mathbb{Q}_1^2$, we get $\nabla p(x,y)\cdot\mathbf{n}=(p_3y+p_2)n_x+(p_3x+p_1)n_y$ whereas for $p\in\mathbb{P}_1^2$, $\nabla p(x,y)\cdot\mathbf{n}=p_2n_x+p_1n_y$ is obtained which means that the normal gradient is approximated by a constant over the whole support. For example, in the configuration of Fig. 8.1.left, with $p\in\mathbb{P}_1^2$, we have :

$$\nabla p(x,y)\cdot\mathbf{n}=\frac{u_I-u_J}{h_x}n_x+\frac{u_K-u_I}{h_y}n_y=u_I(\frac{n_x}{h_x}-\frac{n_y}{h_y})+u_J\frac{n_x}{h_x}+u_K\frac{n_y}{h_y} \tag{7.19}$$

The diagonal coefficient of the constraint for the matrix raw related to u_I is $(\frac{n_x}{h_x}-\frac{n_y}{h_y})$. The case when $\frac{n_x}{h_x}\approx\frac{n_y}{h_y}$ leads to numerical instabilities. If we consider the configuration of Fig. 8.1.left, using the normal vector of the segment $[x_l,x_m]$ implies that the signs of n_x and n_y are always different so the diagonal coefficient is always dominant. The same property occurs for the other cases. When x_I has only one neighbor x_J in Ω_0, the $\mathbb{Q}_1^2$ and $\mathbb{P}_1^2$ interpolations degenerate to $\mathbb{L}_1^1$ interpolations which suit for Dirichlet BC. For Neumann BC, this loss of dimension no longer allows the interface orientation to be accurately taken into account, as one of the components of the normal unit vector disappears from the interfacial constraint. Hence, a third point x_K in Ω_0 is caught to build $\mathbb{P}_1^2$ interpolations (see Fig. 8.1 right). This point is a neighbor of x_J and is taken as $[x_I,x_J]\perp[x_J,x_K]$. As in 2D two choices generally appear, the point being so that the angle $(\mathbf{n},x_K-x_J)$ is in $[-\pi/2;\pi/2]$ is taken.

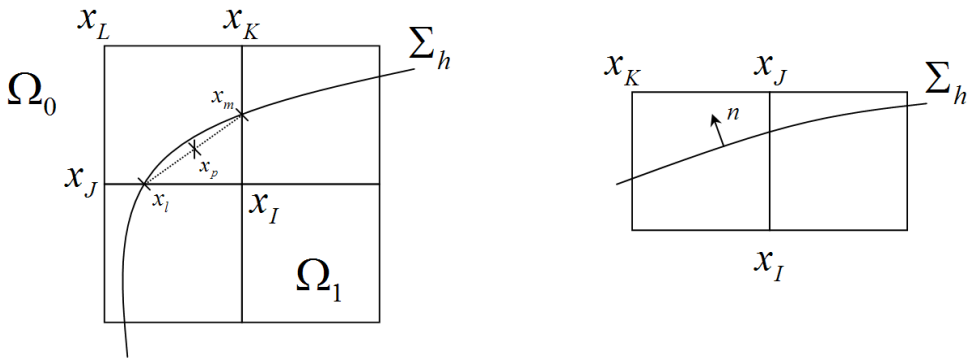


Figure 7.1: Example of selection of points for Dirichlet (left) and Neumann (right) constraints

7.1.4 Treatment of the solution in Ω_1

As shown in the 1D example, the solution u_I for $I \in \mathcal{N}_1$ is an extrapolation of the solution in Ω_0 in order to satisfy the boundary condition on Σ and thus is non-physical. Hence, the solution at the nodes of Ω_1 far from the interface does not impact on the solution in Ω_0. Nevertheless, the fictitious domain approach computes a non-physical solution in Ω_1. It is naturally obtained with the initial set of equations together with a volume penalty method such as VPM. The imposed solution can be analytical when possible, or an arbitrary constant value. The computational cost of this approach can be reduced by switching the solving of $u_I, x_I \in \Omega_1$ off, or by totally removing these nodes in the solving matrix.
One can scan the initial penalized matrix and remove the useless lines. The criterion can be deduced from the VOF function C or from an analysis of the matrix structure. A line I can be removed if it has no link with the nodes impacting on the solution of interest, *i.e.* if $x_I \in \Omega_1$ and $I \notin \mathcal{N}_1$. This have been demonstrated with the 1D example above where a matrix A_1 can be extracted from the initial matrix A. A non negligible gain of speed can be obtained if only A_1 is inverted instead of A. Furthermore, as the peak of memory is often reached during the matrix inversion, this method reduces the total memory requirement of the simulation for a given case. As the other parts of the code are not modified to take the reduction into account, a solution vector of initial size has to be retrieved and a solution has to be chosen where $x_I \in \Omega_1$ and $I \notin \mathcal{N}_1$. This complementary solution is easy to find using the initial penalized values (which is generally the analytical solution in Ω_1).

7.2 Application to the Navier-Stokes equations

7.2.1 The augmented Lagrangian method

The augmented Lagrangien (AL) method (see section A.3.4) consists in adding the term $\nabla(dr\nabla \cdot \mathbf{u})$ to the momentum equation of the NS equations so as to obtain the divergence free constraint. The parameter dr set the magnitude of the constraint and must be chosen according to the magnitude of the other terms of the equation. Iterative solvers can be very sensitive to the magnitude of dr and a high parameter implies an increase of the number of required internal iteration of the solver. A too high parameter penalizes the initial equation and leads to a strictly incompressible velocity field with no respect to the initial momentum equation. However, as one step only is required to solve the NS equations (contrary to the projection methods), the AL

methods allows large time steps to be used. The penalized momentum equation with AL yields:

$$\begin{aligned}&\rho\left(\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}+\mathbf{u}^n\cdot\nabla\mathbf{u}^{n+1}\right)-\nabla(dr\nabla\cdot\mathbf{u}^{n+1})\\&=-\nabla p^n+\rho\mathbf{g}+\nabla\cdot[\mu(\nabla\mathbf{u}^{n+1}+\nabla^T\mathbf{u}^{n+1})]+\frac{\chi_i}{\varepsilon}(\sum_{k\in\mathcal{N}_0\cup\mathcal{N}_1}\alpha_k\mathbf{u}_k^{n+1}-\mathbf{u}_D).\end{aligned}\tag{7.20}$$

The AL and the penalty terms are fully compatible in Ω_0 and no particular manipulation is required. The AL term is not active for the equations where the penalty term is activated ($u_I, I\in\mathcal{N}_1$). However, the stencil of the AL term is large enough to take all the penalized nodes into account.

7.2.2 The scalar projection method

When the AL method is used, an IB method designed for an elliptic equation can generally be applied directly. However, the AL method is not the most commonly used method to ensure the divergence free constraint. Most of the FV CFD codes on Eulerian grids use the scalar projection method (or fractional step method). The base method is described in section A.3.3.1. One of the key points is to solve the pressure projection

$$\nabla\cdot\mathbf{u}_i^*=\nabla\cdot\frac{\Delta t}{\rho}\nabla_i p'\tag{7.21}$$

where $\mathbf{u}^*$ is predicted field for which generally $\nabla\cdot\mathbf{u}^*\neq 0$. Once the pressure increment is obtained, velocity and pressure are updated:

$$p^{n+1}=p'+p^n\tag{7.22}$$

$$\mathbf{u}_i^{n+1}=\mathbf{u}_i^*-\frac{\Delta t}{\rho}\nabla_i p'\tag{7.23}$$

or, with the correction of [Timm 96] on the pressure increment

$$p^{n+1}=p'+p^n-\mu\nabla\cdot\mathbf{u}^*.\tag{7.24}$$

The IB methods for the NS equations are generally designed for the projector step only. As no modification of the corrector step is performed, the additional boundary constraint is not taken into account and is then violated. This problem is not frequently tackled in the literature, and satisfactory solutions have only appeared recently. In [Tair 07], authors modify the boundary force method of Peskin to correct the projection step. In [Dome 08], Domenichini analyzes in details the application of the DF-IBM to the fractional step solution of the Navier-Stokes equations. As can be expected, he notices that the boundary condition is not accurately imposed. In [Iken 07], authors propose a consistant correction for a second-order DF-IBM.

7.2.2.1 First-order correction

For a first-order penalty term, such a correction is easy to performed. Let us now write the momentum Navier-Stokes equation with a first-order penalty term:

$$\rho\frac{\partial\mathbf{u}}{\partial t}=RHS-\nabla p+\frac{\chi}{\varepsilon}(\mathbf{u}-\mathbf{u}_D)\tag{7.25}$$

The same process as for the equation without penalty term is performed. The equation (7.21) becomes:

$$\nabla\cdot\mathbf{u}_i^*=\nabla\cdot(\frac{\rho}{\Delta t}-\frac{\chi_i}{\varepsilon})^{-1}\nabla_i p'\tag{7.26}$$

One can notice that the source term of the penalty term has disappeared. The magnitude of the gradient inside the solid is reduced, but not vanished, with this formulation, and the classic velocity increment (7.23) gives a non-zero velocity inside the object. However, the consistant addition of the first-order penalty term in (7.23) gives:

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - (\frac{\rho}{\Delta t} - \frac{\chi_i}{\varepsilon})^{-1}\nabla_i p' \tag{7.27}$$

which induces a null velocity inside the solid. Hence, whatever the pressure gradient obtained during the pressure correction step, (7.27) cancels the velocity inside the solid and satisfy the penalty constraint as well as $\nabla \cdot \mathbf{u}^{n+1} = 0$ everywhere.

7.2.2.2 Higher-order correction

Correction A: A consistant correction following the precedent walkthrough when a penalty term of higher order is present is much more delicate. We consider the penalty term as always linear. The first-order term $\frac{\chi_i}{\varepsilon}(\mathbf{u}_i - \mathbf{u}_l)$ is replaced by $\frac{\chi_i}{\varepsilon}(P_i\mathbf{u} - \mathbf{u}_l)$ with $P_i\mathbf{u} = \sum_{j\in\mathcal{N}_0\cup\mathcal{N}_1} \alpha_j \mathbf{u}_j$. The pressure equation becomes:

$$\nabla \cdot \mathbf{u}_i^* = \nabla \cdot (\frac{\rho}{\Delta t} - \frac{\chi_i}{\varepsilon}P_i)^{-1}\nabla_i p' \tag{7.28}$$

and the velocity correction is then:

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - (\frac{\rho}{\Delta t} - \frac{\chi_i}{\varepsilon}P_i)^{-1}\nabla p' \tag{7.29}$$

which requires the calculation of the matrix $(\frac{\rho}{\Delta t}\mathbb{I}_d - \frac{\chi}{\varepsilon}P)^{-1}$ before the inversion of the resulting system.

Correction B: This approach has been proposed by [Iken 07] for an DF-IBM method for which the boundary term is expressed at the raw $i, i \in \mathcal{N}_0$ of the discretization matrix. We propose an adaptation of this method to the SMPM. Let us first rewrite (7.27) (RHS' is neglected) with the high-order discretization of the penalty term:

$$\left(\mathbf{u}_i^{n+1} - \mathbf{u}_i^*\right) = -\frac{\Delta t}{\rho}\nabla_i p' + \frac{\Delta t}{\rho}\frac{\chi_i}{\varepsilon}P_i(\mathbf{u}^{n+1} - \mathbf{u}^*). \tag{7.30}$$

The first idea is to keep $\frac{\chi_i}{\varepsilon}P_i$ in front of $\mathbf{u}^{n+1} - \mathbf{u}^*$. If the divergence operator is applied, we obtain

$$\nabla \cdot \mathbf{u}_i^* = \nabla \cdot \frac{\Delta t}{\rho}\nabla_i p' + \nabla \cdot \left(\frac{\Delta t}{\rho}\frac{\chi_i}{\varepsilon}P_i(\mathbf{u}^{n+1} - \mathbf{u}^*)\right). \tag{7.31}$$

As $\mathbf{u}^{n+1}$ is not known (and the primal variable is p'), (7.31) cannot be solved yet for $x_i \in \Omega_1$ (where $\chi_i \neq 0$). However, one can use Eq. (7.30) to write the corrections of the velocity ($\mathbf{u}'$) and the pressure (p'). First, we introduce P_i', a new interpolator such as:

$$P_i'\mathbf{u} = \sum_{j\in\mathcal{N}_0} \alpha_j \mathbf{u}_j \tag{7.32}$$

and $P_i\mathbf{u} = \alpha_i\mathbf{u}_i + P_i'\mathbf{u}$. Eq. (7.30) can be written as

$$\mathbf{u}_i^{n+1} - \mathbf{u}_i^* = -\frac{\Delta t}{\rho}\nabla_i p' - \frac{\Delta t}{\rho}\frac{\chi_i}{\varepsilon}(\alpha_i(\mathbf{u}_i^{n+1} - \mathbf{u}_i^*) - P_i'(\mathbf{u}^{n+1} - \mathbf{u}^*)). \tag{7.33}$$

As the limit penalized problem is considered, one can write the velocity correction: (7.33):

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{\Delta t}{\rho}\nabla_i p' \quad \text{if } x_i \in \Omega_0 \tag{7.34}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{P_i'(\mathbf{u}^{n+1} - \mathbf{u}^*)}{\alpha_i} \quad \text{if } x_i \in \Omega_1 \tag{7.35}$$

As $P_i'(\mathbf{u}_i^{n+1} - \mathbf{u}_i^*)$ is a linear combination of solutions at nodes in Ω_0 only, so one can use (7.34) in (7.35) to obtain

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{\Delta t}{\rho}\nabla_i p' \quad \text{if } x_i \in \Omega_0 \tag{7.36}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{P_i'\frac{\Delta t}{\rho}\nabla p'}{\alpha_i} \quad \text{if } x_i \in \Omega_1 \tag{7.37}$$

One can write a unified form of (7.36)-(7.37):

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \left((1-\chi_i)\frac{\Delta t}{\rho}\nabla_i p' + \chi_i \frac{P_i'\frac{\Delta t}{\rho}\nabla p'}{\alpha_i}\right). \tag{7.38}$$

The final pressure projection equation is obtained with the divergence of (7.38):

$$\nabla \cdot \mathbf{u}_i^* = \nabla \cdot \left((1-\chi_i)\frac{\Delta t}{\rho}\nabla_i p' + \chi_i \frac{P_i'\frac{\Delta t}{\rho}\nabla_i p'}{\alpha_i}\right). \tag{7.39}$$

The pressure is updated as in the standard method

$$p^{n+1} = p' + p^n. \tag{7.40}$$

Remark 7.2.1 *The velocity in Ω_1 is updated as*

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{P_i'\frac{\Delta t}{\rho}\nabla p'}{\alpha_i} \tag{7.41}$$

By construction of the interpolator $P_i\mathbf{u} = \alpha_i\mathbf{u}_i + P_i'\mathbf{u}$, no node of Ω_1 is involved in the stencil of P_i'. Hence, as the pressure correction in Ω_0 is

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{\Delta t}{\rho}\nabla_i p' \tag{7.42}$$

one can replace $\frac{\Delta t}{\rho}\nabla_i p'$ by $(\mathbf{u}_i^* - \mathbf{u}_i^{n+1})$ in (7.41)to obtain

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{P_i'(\mathbf{u}^* - \mathbf{u}^{n+1})}{\alpha_i}. \tag{7.43}$$

Using the initial interpolator P_i, we obtain

$$P_i\mathbf{u}^{n+1} = P_i\mathbf{u}^*. \tag{7.44}$$

which means that the boundary constraint obtained in the predictor step is conserved.

Remark 7.2.2 *For the solution at the nodes $x_i \in \Omega, i \notin \mathcal{N}_1$ (i.e. far from Σ). In this situation, $P_i' = 0$ and the first-order correction is retrieved.*

This method induces a more complex discretization with a larger stencil but seems preferable than a computation of $(\frac{\rho}{\Delta t}\mathbb{I}_d - \frac{\chi}{\varepsilon}P)^{-1}$ which leads to an extended stencil too.

Concerning the accuracy of the method, the same results as with the AL method are obtained for the velocity and the pressure if the correction of Timmermans *et al.* [Timm 96] is used for the projection method. However, the AL method generally allows higher time step to be used.

7.3 Application to *Code_Saturne* (EDF R&D)

One of the aim of this part is to show that the SMP method can be applied without fondamental modifications to an unstructured code.

7.3.1 *Code_Saturne*

Code_Saturne [Arch 04] is a CFD code principally developed by EDF R&D at Châtou. The basic capabilities of *Code_Saturne* enable the handling of either incompressible or expandable flows with or without heat transfer and turbulence. Dedicated modules are available for specific physics such as radiative heat transfer, combustion (gas, coal, heavy fuel oil, ...), magneto-hydrodynamics, compressible flows, two-phase flows (Euler-Lagrange approach with two-way coupling), extensions to specific applications (e.g. Mercure_Saturne for atmospheric environment). *Code_Saturne* is portable on Linux PCs and all UNIX platforms tested so far (HP-UX, Solaris, Cray, OSF1, ...). It runs in parallel with MPI on distributed memory machines (Origin 2000 and 3000, PC clusters, Cray XT-3, IBM Power PC p575, IBM Blue Gene, IBM Power PC 970 Marenostrum...). Developed since 1997 at EDF R&D, it is based on a co-located Finite Volume approach that accepts meshes with any type of cell (tetrahedral, hexahedral, prismatic, pyramidal, polyhedral...) and any type of grid structure (unstructured, block structured, hybrid, conforming or with hanging nodes,...). Compatible mesh generators include I-DEAS®, GMSH, Gambit®, Simail®, Salomé, Harpoon®, ICEM®,... Post-processing output is available in EnSight®, CGNS and MED_fichier formats, with advanced data management capabilities by the FVM library (EDF's "Finite Volume Mesh" library, under LGPL licence). Parallel code coupling capabilities are also provided by the FVM library. *Code_Saturne* is property of EDF and distributed under the GNU GPL licence. *Code_Saturne* can be coupled to EDF's thermal software SYRTHES (conjugate heat transfer). It can also be used jointly with EDF's structural analysis software Code_Aster, in particular in the Salomé platform. SYRTHES and Code_Aster are developed by EDF and distributed under GNU GPL licence.

7.3.2 Developments

7.3.2.1 Shape management

As *Code_Saturne* mainly works with unstructured meshes, the Thread Cay-casting method (6.1.2.2) has not been developed in this context and only the basic Ray-casting algorithm has been developed. Can the TRC be applied to the unstructured grids? One can considers nodes by peers instead of rows and use rays defined by peers of nodes. Theoretically, the cost of the base algorithm is divided by two if the cost of finding peers is negligible. However, the implementation of any of our Ray-casting method uses rays which are parallel to the main frame axes to avoid some computations. If the direction of the rays is random, many low-level optimisations cannot be used. One can imagine a frame transformation (following the same idea of the curvilinear to Cartesian projection) which produces a new frame where the nodes are as often as possible aligned.

7.3.2.2 SMP algorithm

The algorithm has been implemented. We choose to build the interpolations by summing the contributions of each neighbor of a penalized node. The main advantage here is that the method works directly, no matter the number of neighbors of a penalized node.

7.3.2.3 Heat equation

The heat equation is solved in structured grids in *Code_Saturne* with about the same discretization as for Thétis. The principal difference is that *Code_Saturne* increments progressively the solution by solving successive linear systems. The initial penalty term $P_i\mathbf{u}_i^{n+1} - \mathbf{u}_l$ becomes $P_i\mathbf{u}^{k+1,n+1} - \mathbf{u}_l + P_i\mathbf{u}_D^{n+1,k}$ where k is the number of the sub-iterations ($k+1$ being the current iteration) and P_i the interpolator.

The solvers available in *Code_Saturne* during the project was not able to solve a penalized matrix (which is not diagonal dominant). The solution has been to couple *Code_Saturne* with a solver of thétis (BiCG-Stab and ILU factorization). An alternate solution consists in replacing the iterative penalty term $P_i\mathbf{u}^{k+1,n+1} - \mathbf{u}_l + P_i\mathbf{u}_D^{n+1,k}$ by $P_i^*\mathbf{u}^{k+1,n+1} - \mathbf{u}_l + P_i\mathbf{u}_D^{n+1,k}$ where P_i^* is a new interpolator such as the resulting matrix line $P_i^*\mathbf{u}^{k+1,n+1}$ is diagonal dominant. As the source term $-\mathbf{u}_l + P_i\mathbf{u}_D^{n+1,k}$ is still defined with the original interpolator, the method converges to the desired solution.

On unstructured grids, the solution does not converge, even if the solution seems good at first sight. The reconstruction of the operators for the unstructured meshes seems to be involved.

7.3.2.4 Navier-Stokes equations

The method has been applied to the Navier-Stokes equations. The difficulty is to couple the SMPM with the scalar projection method. The correction proposed in the previous section works very well in thétis and has not been implemented yet in *Code_Saturne* . The Fig. 7.2.left shows the positive and the negative parts of u_x after the prediction step. One can see that the iso-line $u_x = 0$ matches the boundary of the particle. Concerning the pressure correction, the simulation diverge if the projection is not modified. A first try has been to cancel the source term of the projection equation in the solid. The Fig. 7.2.right shows the resulting streamlines and pressure field for the same case. An incorrect boundary layer is present around the particle.

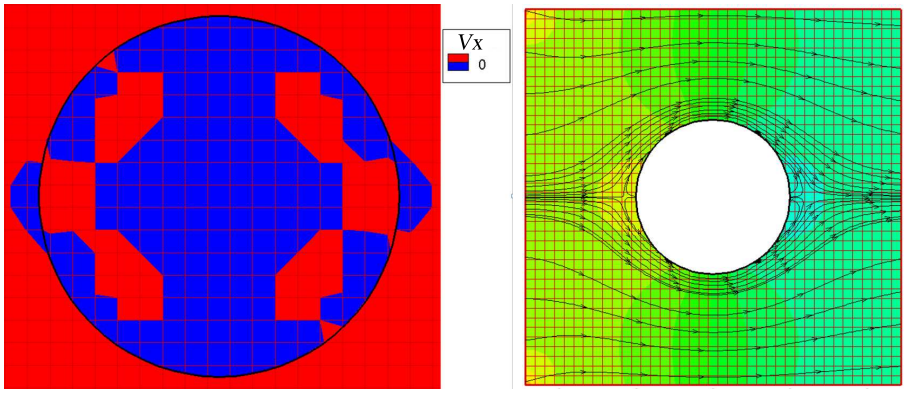


Figure 7.2: Positive and negative parts of u_x

7.3.3 Results

7.3.3.1 2D method on structured grid

The method is validated for a Poisson equation. We consider two disks of radii R_1 and R_2 with Dirichlet BC T_1 and T_2. The first circle is included in the numerical domain of dimensions

$[-0.5\,;\,0.5] \times [-0.5\,;\,0.5]$. The second bounds the domain and defines the boundary condition of the numerical domain. The Laplace equation is solved. The solution at a radius r with $R_1 \leq r \leq R_2$ is:

$$T(r) = \frac{T_2 - T_1}{ln(R_2) - ln(R_1)} ln(r) + T_1 - (T_2 - T_1) \frac{ln(R_1)}{ln(R_2) - ln(R_1)} \tag{7.45}$$

The calculations are performed with $R_1 = 0.2m$, $R_2 = 2m$, $T_1 = 20$, $T_2 = 0$. The results in Fig.

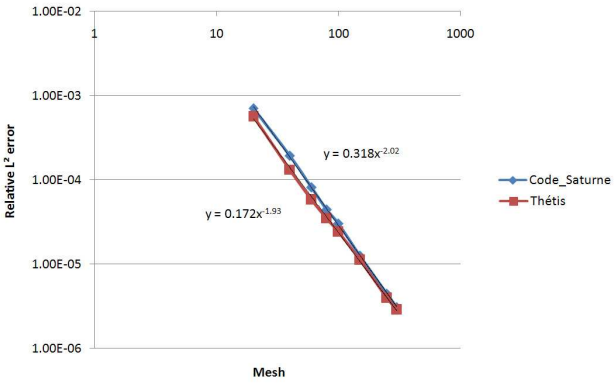


Figure 7.3: L^2 relative error for the SMP method with Thétis and *Code_Saturne* for a 2D case

(7.3) shows a second-order accuracy for Thétis and *Code_Saturne* .

7.3.3.2 3D method on structured grid

The equation $\Delta T = 6$ is solved on a 3D domain (a unit cube). The analytical solution is $T(x, y, z) = x^2 + y^2 + z^2$. The solution is penalized on a sphere of radius $0.2m$. The Fig. 7.4 shows a very satisfactory convergence for Thétis and *Code_Saturne* .

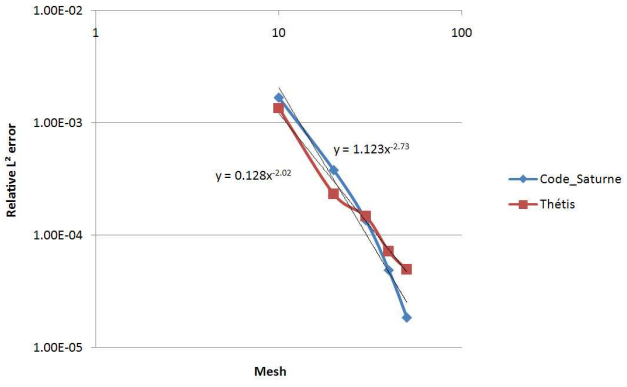


Figure 7.4: L^2 relative error for the SMP method with Thétis and *Code_Saturne* for a 3D case

Chapter 8

The Algebraic Immersed Interface method

8.1 General principle

Once the shape informations are available on the Eulerian grid, the problem discretization has to be modified to take into account the fictitious domain (an immersed boundary or an immersed interface). The sub-mesh penalty (SMP) method [Sart 08b, Sart 08a] was originally designed to treat immersed boundary problems. It could be extended to treat immersed interface problems by symmetrization of the algorithm with introduction of auxiliary unknowns as in the AIIB method presented here. This new method is an enhancement of the SMP method which is also able to solve immersed interface problems. The main idea of the AIIB method is to embed an interface into a given domain by modifying the final matrix only. As no modification of the discretization of the operators is required (contrary to [Gibo 02, Gibo 05] and the immersed interface methods [Leve 94]), the AIIB method is thus simple to implement.

Let $\mathcal{P}$ be a model problem discretized in the whole domain Ω as $Au = b$ where A is a square matrix of order m, u the solution vector and b a source term. The basic idea of the AIIB method is to add new unknowns and equations to the initial linear system so as to take into account additional interface constraints. The new unknowns, so-called the auxiliary or fictitious unknowns and labeled with $*$, are defined as being the extrapolation of the solution from one side of the interface to the other, and are used to discretize the interface conditions. Hence, the orignal problem $Au = b$ becomes $A'u' = b'$, with A' a square matrix of order $m + n$, with n the number of auxiliary constraints related to the interface conditions. The solution u' is decomposed such as $u' = (u, u^*)^T$ and the source term as $b' = (b, b^*)^T$. The interface constraints are discretized with a $(n, m + n)$ block matrix C and the source term b^*.

According to the interface conditions, the regularity of the solution on the interface is often lower than in the rest of the domain. Hence, the discretization of operators with a stencil cutting the interface can induce a great loss of accuracy. The first idea is to consider unknowns $u^*_I, I \in \mathcal{N}_1$ (resp. $u^*_I, I \in \mathcal{N}_0$) as the extension of the solution in Ω_0 (resp. Ω_1). The initial algebraic link between unknowns from both sides of the interface is cut, and the new link over the interface is obtained thanks to auxiliary unknowns. Practically, matrix coefficients must be modified to take into account the new connectivities. Let $\alpha_{I,J}$ be a coefficient of A at row I, column J and $\alpha'_{I,J}$ the new coefficient in A'. If $I \in \mathcal{N}_0$ and $J \in \mathcal{N}_1$, $\alpha'_{I,J} = 0$ and $\alpha'_{I,J^*} = \alpha_{I,J}$, where J^* is the index corresponding to u^*_J.

This is exactly the way how we proceed for the practical algorithm. However, this modification can be expressed algebraically with permutation and mask matrices as follows.

We define the two following mask matrices I_1 of dimensions $(m, m+n)$ and I_2 of size $(n, m+n)$:

$$I_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & & \vdots & \ddots & & \vdots \\ \vdots & & \ddots & 0 & & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}, \quad I_2 = \begin{pmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & & 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \tag{8.1}$$

The matrices A_0 and A_1 are defined such as $A_0 + A_1 = A$, $A_0(I,J) = A(I,J)$ if $I \in \mathcal{N}_0$, else $A_0(I,J) = 0$. Similarly $A_1(I,J) = A(I,J)$ if $I \in \mathcal{N}_1$ else $A_1(I,J) = 0$. Finally, the connectivities are changed using the permutation matrices P_0 and P_1: P_0 is defined to switch row I with row J if $I \in \mathcal{N}_0$, $J \in \mathcal{N}_1$ and P_1 to switch row I with row J if $I \in \mathcal{N}_1$, $J \in \mathcal{N}_0$. Hence, the new problem matrix is now defined by:

$$A' = I_1^T \left(P_0(A_0 I_1) + P_1(A_1 I_1)\right) + I_2^T C \tag{8.2}$$

The new problem is $A'u' = b'$ with A' written with 4 blocks of various sizes: $\tilde{A}(m,m)$, $B(m,n)$, $C_1(n,m)$, $C_2(n,n)$. The matrix $\tilde{A}$ is thus the modification of the initial matrix A by setting to zero the coefficient $\alpha_{I,J}$ if $\chi(x_I) \neq \chi(x_J)$, and C_1 and C_2 are the two sub-matrices of the matrix C. The problem can be written as:

$$\begin{pmatrix} \tilde{A} & B \\ C_1 & C_2 \end{pmatrix} \begin{pmatrix} u \\ u^* \end{pmatrix} = \begin{pmatrix} b \\ b^* \end{pmatrix} \tag{8.3}$$

The entire problem can be then solved to obtain $u' = (u, u^*)^T$. However, u^* being the auxiliary solution is not required to be computed explicitly . Hence, the Schur complement method can be used to calculate the solution for the physical unknowns only. The final problem is now:

$$(\tilde{A} - BC_2^{-1}C_1)u = b - BC_2^{-1}b^* \tag{8.4}$$

The opportunity of such a reduction will be discussed later.

8.2 AIIB method for immersed boundary problems

8.2.1 Scalar equation with Dirichlet boundary conditions

For sake of clarity, let us first describe in $2D$ the AIIB method for the model scalar problem $\mathcal{P}_b$ with a Dirichlet boundary condition on the interface Σ. For this version of the AIIB algorithm, Ω_0 is the domain of interest and auxiliary unknowns are created in Ω_1 only. Let us consider a point $x_I, I \in \mathcal{N}_1$. At location x_I, two unknowns coexist: a physical one u_I and an auxiliary one u_I^*. We first describe the case when x_I has only one neighbor x_J in Ω_0. The Lagrangian point x_l is the intersection between $[x_I; x_J]$ and Σ_h (Fig. 6.17 right). Then, the solution $u_l = u_D(x_l)$ at the interface is approximated by the $\mathbb{P}_1^1$ interpolation between the Eulerian unknowns u_I^* and u_J :

$$u_l = \alpha_I u_I^* + \alpha_J u_J \text{ with } 0 < \alpha_I, \alpha_J < 1 \text{ and } \alpha_I + \alpha_J = 1 \tag{8.5}$$

As noticed in [Tsen 03, Gibo 05], a linear interpolation only is required to reach a second order of accuracy. If now x_I has a second neighbor x_K in Ω_0, the intersection x_m between $[x_I; x_K]$ and Σ_h is considered with $u_m = u_D(x_m)$. We choose x_p, a new point of Σ_h between x_l and x_m (see

Fig. 8.1 left). The solution $u_p = u_D(x_p)$ is then imposed using a $\mathbb{P}_1^2$-interpolation of the values u_I^*, u_J and u_K :

$$u_p = \alpha_I u_I^* + \alpha_J u_J + \alpha_K u_K \ , \ 0 < \alpha_I, \alpha_J, \alpha_K < 1 \ , \ \alpha_I + \alpha_J + \alpha_K = 1 \tag{8.6}$$

A $\mathbb{Q}_1^2$ interpolation of u_I, u_J, u_K and u_L can be also used by extending the interpolation stencil with the point x_L which is the fourth point of the cell of the dual mesh defined by x_I, x_J and x_K (see Fig. 8.1 left). As a third choice, two independent linear 1D interpolations can be used (one for each direction) for an almost equivalent result. It produces :

$$\begin{cases} u_l = \alpha_I u_I^* + \alpha_J u_J \text{ with } 0 < \alpha_I, \alpha_J < 1 \text{ and } \alpha_I + \alpha_J = 1 \\ u_m = \alpha'_I u_I^{*'} + \alpha_K u_K \text{ with } 0 < \alpha'_I, \alpha_K < 1 \text{ and } \alpha'_I + \alpha_J = 1 \end{cases} \tag{8.7}$$

In this case, two auxiliary unknowns are created.

A simple choice for x_p is the barycenter between x_l and x_m where $u_p = (u_l + u_m)/2$. This particular case enables an easy implementation since we have :

$$\alpha_I u_I^* + \alpha_J u_J = u_l \tag{8.8}$$

$$\alpha'_I u_I^* + \alpha_K u_K = u_m \tag{8.9}$$

A summation of these two constraints gives :

$$\alpha_I u_I^* + \alpha_J u_J + \alpha'_I u_I^* + \alpha_K u_K = u_l + u_m \tag{8.10}$$

what is equivalent to build a constraint imposing u_p at x_p with a $\mathbb{P}_1^2$ interpolation :

$$\begin{gathered} \frac{(\alpha_I + \alpha'_I) u_I^* + \alpha_J u_J + \alpha_K u_K}{2} = u_p \ , \\ \text{with } 0 < \frac{\alpha_I + \alpha'_I}{2}, \frac{\alpha_J}{2}, \frac{\alpha_K}{2} < 1 \ , \ \frac{\alpha_I + \alpha'_I}{2} + \frac{\alpha_J}{2} + \frac{\alpha_K}{2} = 1 \end{gathered} \tag{8.11}$$

Hence, an easy general implementation consists in summing the constraints corresponding to each direction, no matter the number of neighbors of x_I. If the elements σ_l of Σ_h used to define x_l and x_m are not the same, the barycenter x_p of these two points is not necessarily on Σ_h, especially for interfaces of strong curvature. However, the distance $d(x_p, \Sigma_h)$ between x_p and Σ_h varies like $\mathcal{O}(h^2)$ and so this additional error does not spoil the second-order precision of our discretization. The convergence of this additional error is numerically tested in section (9.1.2). If the curvature of Σ_h is small enough relatively to the Eulerian mesh, *i.e.* if the Eulerian mesh is sufficiently fine, x_I almost never has a third or a fourth neighbor in Ω_0. However, if this case appears, a simple constraint $u_I^* = u_B$ is used with u_B being an average of u_D at the neighbor intersection points. In any case, by decreasing the Eulerian mesh step h, the number of points x_I having more than two neighbors in Ω_0 also decreases.

Hence, the present method is suitable to impose a Dirichlet boundary condition on Σ for Ω_0, when the solution in Ω_1 has no interest. The solution u_I^* for $I \in \mathcal{N}_1$ is an extrapolation of the solution in Ω_0 in order to satisfy the boundary condition on Σ and thus is non-physical. Hence, the solution at the nodes of Ω_1 far from the interface does not impact on the solution in Ω_0. Nevertheless, the fictitious domain approach computes a non-physical solution in Ω_1. It is naturally obtained with the initial set of equations together with a volume penalty method such as VPM [Khad 00]. The imposed solution can be analytical when possible, or an arbitrary constant value. The computational cost of this approach can be reduced by switching the solving of $u_I, x_I \in \Omega_1$ off, or by totally removing these nodes in the solving matrix.

8.2.1.1 Symmetric version for Dirichlet interface conditions

The next step is to allow for multiple Dirichlet boundary conditions on both sides of the immersed interface. Thin objects could be treated with this approach. The problem is now :

$$\begin{cases} -\nabla \cdot (a\nabla u) = f & \text{in } \Omega \\ u^-_{|\Sigma} = u_D & \text{on } \Sigma \\ u^+_{|\Sigma} = u_G & \text{on } \Sigma \end{cases} \tag{8.12}$$

The problem (8.12) requires for each point x_I a physical unknown u_I as well as an auxiliary unknown u_I^* on both sides of the interface.

Practically, the AIIB algorithm for a Dirichlet BC is applied a first time with Ω_0 as domain of interest, and auxiliary unknowns are created near Σ_h in Ω_1. As a second step, the Heaviside function is modified as $\chi := 1 - \chi$ and the algorithm is applied a second time. Now, Ω_1 is the domain of interest and auxiliary unknowns are created near Σ in Ω_0.

8.2.2 Scalar equation with Neumann boundary conditions

Let us now consider the following model scalar problem with a Neumann BC on the interface Σ :

$$\begin{cases} -\nabla \cdot (a\nabla u) = f & \text{in } \Omega_0 \\ (a \cdot \nabla u) \cdot \mathbf{n} = g_N & \text{on } \Sigma \end{cases} \tag{8.13}$$

The principle is about the same as for Dirichlet BC, and the same interpolations, once derived, can be used to approximate the quantity $(a \cdot \nabla u) \cdot \mathbf{n}$. Hence, at any point $x_l, l \in \mathcal{I}$ on Σ_h we use

$$(a \cdot \nabla u_l) \cdot \mathbf{n} \approx (a \cdot \nabla p(x_l) \cdot \mathbf{n}). \tag{8.14}$$

For $p \in \mathbb{Q}_1^2$, we get $\nabla p(x,y) \cdot \mathbf{n} = (p_3 y + p_2) n_x + (p_3 x + p_1) n_y$ whereas for $p \in \mathbb{P}_1^2$, $\nabla p(x,y) \cdot \mathbf{n} = p_2 n_x + p_1 n_y$ is obtained which means that the normal gradient is approximated by a constant over the whole support. For example, in the configuration of Fig. 8.1.left, with $p \in \mathbb{P}_1^2$, we have :

$$\nabla p(x,y) \cdot \mathbf{n} = \frac{u_I^* - u_J}{h_x} n_x + \frac{u_K - u_I^*}{h_y} n_y = u_I^* (\frac{n_x}{h_x} - \frac{n_y}{h_y}) + u_J \frac{n_x}{h_x} + u_K \frac{n_y}{h_y} \tag{8.15}$$

The diagonal coefficient of the raw related to u_I^* in C_2 is $(\frac{n_x}{h_x} - \frac{n_y}{h_y})$. The case when $\frac{n_x}{h_x} \approx \frac{n_y}{h_y}$ leads to numerical instabilities. If we consider the configuration of Fig. 8.1.left, using the normal vector of the segment $[x_l, x_m]$ implies that the signs of n_x and n_y are always different so the diagonal coefficient is always dominant. The same property occurs for the other cases. When x_I has only one neighbor x_J in Ω_0, the $\mathbb{Q}_1^2$ and $\mathbb{P}_1^2$ interpolations degenerate to $\mathbb{L}_1^1$ interpolations which suit for Dirichlet BC. For Neumann BC, this loss of dimension no longer allows the interface orientation to be accurately taken into account, as one of the components of the normal unit vector disappears from the interfacial constraint. Hence, a third point x_K in Ω_0 is caught to build $\mathbb{P}_1^2$ interpolations (see Fig. 8.1 right). This point is a neighbor of x_J and is taken as $[x_I, x_J] \perp [x_J, x_K]$. As in 2D two choices generally appear, the point being so that the angle $(\mathbf{n}, x_K - x_J)$ is in $[-\pi/2; \pi/2]$ is taken.

8.2.3 Algebraic elimination using the Schur complement

The Schur complement method allows an algebraic reduction to be performed. For a Dirichlet or Neumann BC, each constraint is written such as only one auxiliary unknown is needed:

$$u_I^* = \sum_{J \in \mathcal{N}} \alpha_J u_J + u_S \tag{8.16}$$

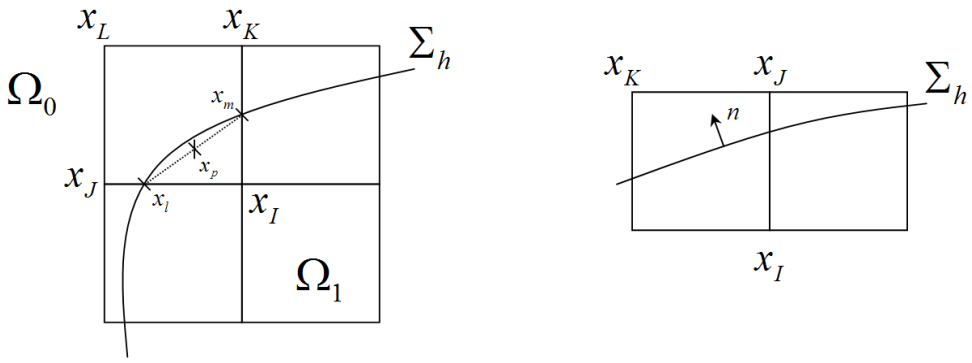


Figure 8.1: Example of selection of points for Dirichlet (left) and Neumann (right) constraints

where u_S is the source term. In this case, the matrix C_2 in (8.3) is diagonal and thus the Schur complement $(\tilde{A} - BC_2^{-1}C_1)$ is easy to calculate. Practically, when the algebraic reduction is made, $\tilde{A}$ is built directly by the suitable modification of A without considering the extended matrix A'. The part $-BC_2^{-1}C_1$ is then added to $\tilde{A}$ whereas $-BC_2^{-1}b^*$ is added to b. As will be subsequently demonstrated, the algebraic reduction decreases the computational cost of the solver by $10-20\%$.

If only $\mathbb{L}_1^1$ interpolations are used with the algebraic elimination, the matrix obtained with this method is similar to the one obtained in [Gibo 02] for a Dirichlet problem. However in this last paper the auxiliary unknowns are taken into account before the discretization of the operator which requires additional calculations for each discretization scheme.

If $\mathbb{P}_1^2$ interpolations are used, the computed solution in Ω_0 is the same as for the SMP [Sart 08a] method (when the penalty parameter tends to zero) and the DF-IB method [Tsen 03]. These methods change the discretization of the initial equation for the nodes $x_I, I \in \mathcal{N}_1$. The SMP method uses a penalty term and the DF-IB method uses terms of opposite signs to erase some part of the initial equation. The discretization matrix obtained with both methods is not equivalent to the one obtained with the AIIB method, with or without algebraic reduction. With algebraic reduction, the discretization for the nodes $x_I, I \in \mathcal{N}_0$ is modified, and without algebraic reduction, both auxiliary and physical unknowns coexist at $x_I, I \in \mathcal{N}_1$. The accuracy of these methods will be discussed in the next section.

The present algorithm seems simpler, as the standard discretization of the operators is automatically modified in an algebraic manner. So, various discretization schemes of the spatial operators can be used. However, the discretization of an operator at $x_I \in \Omega_0$ can only use in Ω_1 the fictitious unknowns and not the physical ones. Hence, the only limitation concerns the stencil of these operators which have to be limited, if centered, to three points by direction.

8.2.4 A word on the application to the Navier-Stokes equations

The SMP method has been applied to the Navier-Stokes equations in [Sart 08a]. For immersed boundary problems, the SMP and the AIIB methods give equivalent results and the AIIB method can be used to immerse obstacles in fluid flows. Both methods can be used for the scalar and the Navier-Stokes equations. In the latter, the procedure is done componentwise for the velocity vector. However, the AIIB method, with $\mathbb{L}_1^1$ interpolations only, cannot be applied to the Navier-Stokes equations on staggered grid (no tests have been performed for a collocated approach). An illustration is given Fig. 8.2. With such interpolations, two auxiliary unknowns u_I^* and

$u_I^{*'}, I \in \mathcal{N}_1$ can coexist at the same location x_I. Hence, u_I^* is the natural neighbor of u_J and $u_I^{*'}$ is the natural neighbor of u_K. So a problem occurs for the discretization of the inertial term since a node of a given velocity component has to use an auxiliary unknown of an other velocity component. In this case, neither u_I^* nor $u_I^{*'}$ are natural neighbors for v_l, a velocity unknown in the y direction. No matter which unknown is used, or an average of the two collocated unknowns, the simulation is instable outside 'the Stokes regime.

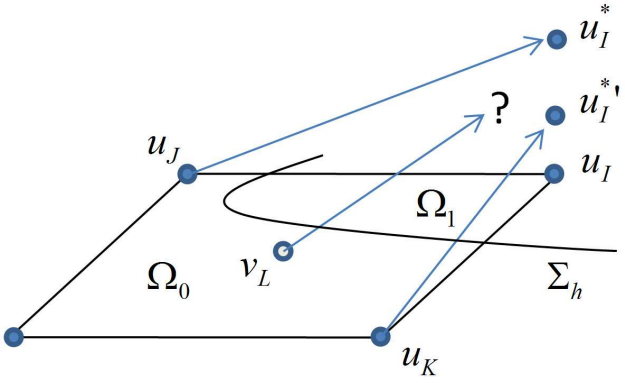


Figure 8.2: Illustration of the application to the Navier-Stokes equations on staggered grid

A particular attention has also to be given to the velocity pressure coupling. If a fractional step method is used, the prediction step is modified by any fictitious domain method to impose an immersed boundary condition for the velocity. Thus, the projection step has to be modified according to the prediction step to remain consistent with the overall problem.

However, some authors do not consider at all this modification of the correction step [Tsen 03] or have only made minor modifications. In fact, the projection step has to be rewritten considering the forcing term, as can be seen in [Iken 07, Dome 08]. In [Sart 08a], the authors use an iterative augmented Lagrangian method [Vinc] which adds a penalty term in the momentum equation to enforce the divergence free constraint.

8.3 AIIB for immersed interface problems

With the symmetric method described in (8.2.1.1), the problem can be solved on both sides of the interface when explicit Dirichlet BC are imposed. For many problems, the solution is not *a priori* known on the interface and some jump transmission conditions on the interface Σ are required. Let us now consider the problem :

$$(\mathcal{P}_i) \quad \begin{cases} -\nabla \cdot (a \nabla u) = f & \text{in } \Omega \\ + \text{ Interface conditions} & \text{on } \Sigma \end{cases}$$

where the interface conditions are :

$$[\![u]\!]_\Sigma = \varphi \quad \text{on } \Sigma \tag{8.17}$$

$$[\![(a \cdot \nabla u) \cdot \mathbf{n}]\!]_\Sigma = \psi \quad \text{on } \Sigma \tag{8.18}$$

The notation $[\![\ ]\!]_\Sigma$ denotes the jump of a quantity over the interface Σ. In the symmetric version of the AIIB method, a given intersection point $x_l, l \in \mathcal{I}$, is associated with two auxiliary unknowns on both sides of the interface. Hence, the interface constraints (8.17) and (8.18) of $(\mathcal{P}_i)$ can be

imposed at each intersection point x_l by using the two auxiliary unknowns. For example, the I^{nth} row of the matrix A' with $u_I^*, I \in \mathcal{N}_0$ can be used to impose the constraint (8.17) and the J^{nth} line of the matrix with $u_J^*, J \in \mathcal{N}_1$ is then used to impose the constraint (8.18).

8.3.1 The solution constraint

The symmetrized AIIB methods for Dirichlet BC reads :

$$\left\{ \begin{array}{l} u_\Sigma^+ = \alpha_1 u_I + \alpha_2 u_J^* \\ u_\Sigma^- = \alpha_1 u_I^* + \alpha_2 u_J \end{array} \right. \tag{8.19}$$

when $\mathbb{L}_1^1$ interpolations are used. With $[\![u]\!]_\Sigma = u_\Sigma^+ - u_\Sigma^- = \varphi$, we obtain :

$$\alpha_1 u_I + \alpha_2 u_J^* - \alpha_1 u_I^* - \alpha_2 u_J = \varphi \tag{8.20}$$

which is the first constraint to be imposed.

8.3.2 The flux constraint

Following the same idea and using $\mathbb{P}_1^2$ interpolations,

$$\left\{ \begin{array}{l} (a \cdot \nabla u_\Sigma^+) \cdot \mathbf{n} = a^+ (\frac{u_I - u_J^*}{h_x} n_x + \frac{u_K^* - u_I}{h_y} n_y) \\ (a \cdot \nabla u_\Sigma^-) \cdot \mathbf{n} = a^- (\frac{u_I^* - u_J}{h_x} n_x + \frac{u_K - u_I^*}{h_y} n_y) \end{array} \right. \tag{8.21}$$

for the case presented in Fig. 8.1.left. Using (8.18), we get:

$$a^+ \left(\frac{u_I - u_J^*}{h_x} n_x + \frac{u_K^* - u_I}{h_y} n_y \right) - a^- \left(\frac{u_I^* - u_J}{h_x} n_x - \frac{u_K - u_I^*}{h_y} n_y \right) = \psi \tag{8.22}$$

which is the second constraint to be imposed. With such an interpolation, the solution gradient is constant over the whole stencil. As demonstrated later, the second-order accuracy can be reached on Cartesian grids when $\psi = 0$.

Three auxiliary unknowns are thus involved in the discretizations (8.20) and (8.22). The auxiliary unknown u_K^* is also involved in the discretization of (8.17) and (8.18) at another intersection point on Σ_h. Hence, the whole system $A'u' = b'$ is closed.Since we need more than one auxiliary unknown to discretize each constraint, the matrix C_2 is not diagonal and a solver has to be used to compute C_2^{-1}.

For the matched interface and boundary (MIB) method, Zhou et al. [Zhou 06b] use a different discretization of the interface conditions which allows an easy algebraic reduction which is directly performed raw by raw.

The algebraic reduction for the immersed interface problems has not been yet implemented. However, the standard discretization of the AIIB method requires a more compact stencil than for the MIB method, and the additional computational time generated by the auxiliary nodes is small. Hence, the lack of algebraic reduction does not seem to be problematic.

Chapter 9

Validation

9.1 Elliptic equations

Elliptic equations are discretized using the standard second-order centered Laplacian. For all problems, similar results have been obtained with a PARDISO direct solver [Sche 04], and an iterative BiCGSTAB solver [Gust 78a], preconditioned under a ILUK method [Saad 86]. Unless otherwise mentioned, a numerical domain $[-1;1]\times[-1;1]$ is used for every simulation. Only Ω_0 is taken into account for the immersed boundary problems.

9.1.1 Immersed boundary problems

The immersed boundary problems are treated here with the SMP and the AIIB method. One can notice that for $\mathbb{P}_1^2$ and $\mathbb{Q}_1^2$ interpolations, both method are equivalent. The $\mathbb{L}_1^2$ interpolations cannot be used with the SMP method since two constraints have to coexist in a same node involving two auxiliary unknowns.

Problem 1 The homogenous 2D Laplace equation is solved. The interface Σ is a centered circle of radius $R_1 = 0.5$ with a Dirichlet condition of $U_1 = 10$. An analytical solution which accounts for the presence of a second circle with a radius $R_2 = 2$ and $U_2 = 0$ is imposed on the boundary conditions. The analytical solution is:

$$u(r) = \frac{U_2 - U_1}{ln(R_2) - ln(R_1)} ln(r) + U_1 - (U_2 - U_1)\frac{ln(R_1)}{ln(R_2) - ln(R_1)} \tag{9.1}$$

Accuracy tests are performed with $\mathbb{L}_1^1$, $\mathbb{P}_1^2$ and $\mathbb{Q}_1^2$ interpolations. Fig. 9.1 shows the solution and the error map for a 32×32 mesh with $\mathbb{P}_1^2$ interpolations. The same results are always obtained with and without algebraic reduction. Fig. 9.2 shows the convergence of the error for the L^2 and L^∞ norms. For all interpolations, the convergence slopes are approximatively 2 for the relative L^2 error. For the L^∞ error, the slopes are about 1.8. The $\mathbb{P}_1^2$ interpolation is the more accurate, followed by the $\mathbb{L}_1^1$ interpolation although it uses more auxiliary points (but a smaller stencil). However, the differences of accuracy between the different interpolations remain small. The same cases with algebraic reduction give the same accuracy. The performances of the ILUK-BiCG-Stab solver are now benchmarked for the three interpolations with and without algebraic reduction and for the SMP method. Tab. 9.1 shows the computational times of the matrix inversions (average time in seconds for 25 matrix inversions) and Tab. 9.2 shows the time ratio between the standard and the reduced matrix. Except for the $\mathbb{Q}_1^1$ interpolation on the 1024×1024 mesh, the differences between the two methods seem to decrease with the size of the matrix. In fact,

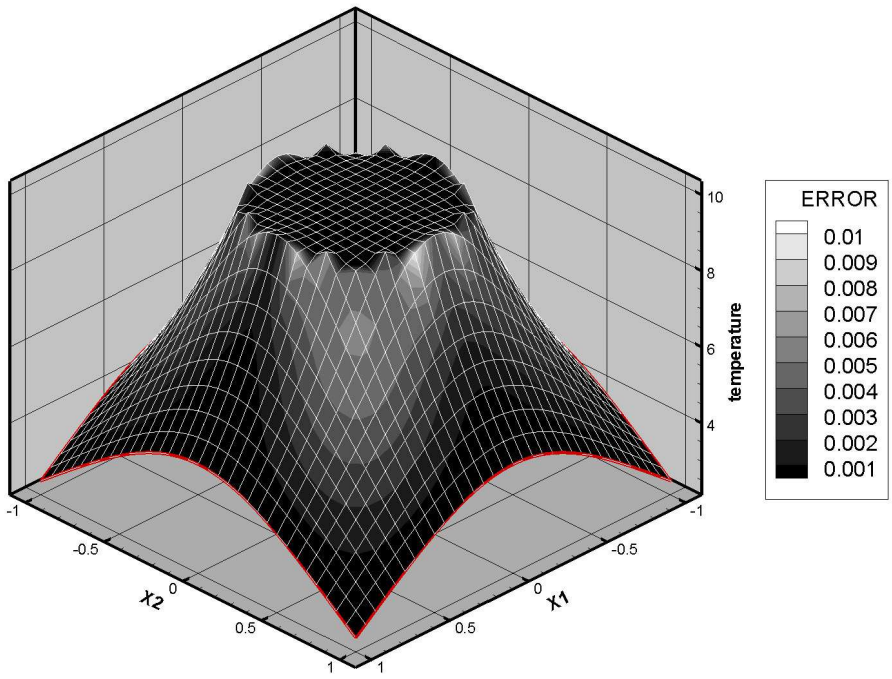


Figure 9.1: Solution and error map for problem 1

as interfaces are $d-1$ manifolds, the number of intersection points does not increase as fast as the Eulerian points. Hence, the ratio between the size of a reduced and a complete matrix tends to 1. The computational time for the SMP method is quite similar to the one obtained AIIB method with algebraic reduction. Figures 9.3, 9.4, 9.5, 9.6 shows the convergence of the ILUK-BiCG-Stab solver for the seven configurations. The type of interpolation does not significantly impact on solver performances.

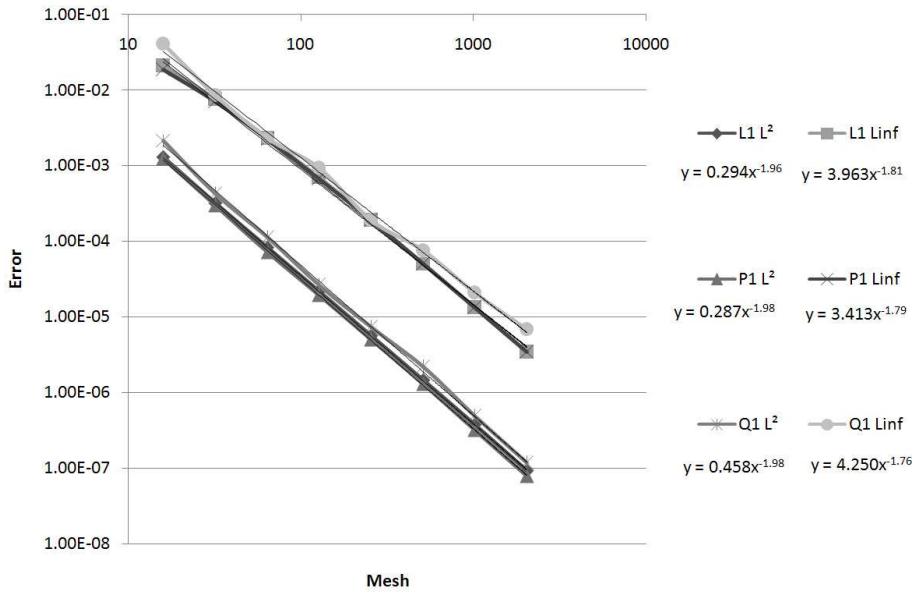


Figure 9.2: Curves of errors for section 9.1.1

Mesh	$\mathbb{L}_1^1$ std	$\mathbb{L}_1^1$ red	$\mathbb{P}_1^2$ std	$\mathbb{P}_1^2$ red	$\mathbb{Q}_1^2$ std	$\mathbb{Q}_1^2$ red	$\mathbb{P}_1^2$ SMP
128	0.215	0.189	0.216	0.182	0.208	0.181	0.181
256	2.18	1.89	2.14	1.83	2.14	1.88	1.88
512	19.7	17.6	19.5	17.1	20.3	18.4	16.9
1024	168	159	171	156	173	141	168

Table 9.1: Computational times in seconds for problem 1. Tests are performed with three different interpolations with (red) and without (std) algebraic reduction, and compared to the SMP method

Mesh	$\mathbb{L}_1^1$	$\mathbb{P}_1^2$	$\mathbb{Q}_1^2$ std
128	88.3%	84.5%	87.3%
256	86.9%	85.5%	88.2%
512	89.4%	87.5%	90.9%
1024	94.6%	91.2%	81.5%

Table 9.2: Ratio of computational times for reduced and standard matrices for section 9.1.1

	Mesh	$\mathbb{L}^1$ std	$\mathbb{L}^1$ red	$\mathbb{P}^1$ std	$\mathbb{P}^1$ red	$\mathbb{Q}^1$ std	$\mathbb{Q}^1$ red	$\mathbb{P}^1$ SMP
Dimension	128	16640	16384	16560	16384	16560	16384	16384
	256	66048	65536	65896	65536	65896	65536	65536
	512	328704	327168	328552	327472	328704	327472	326752
Non-zero elements	128	82432	81664	82352	81824	82432	81824	81472
	256	263168	262144	262864	262144	262864	262144	262144
	512	1312768	1309696	1312464	1310304	1312768	1310304	1308864

Table 9.3: Rank and number of non-zero coefficients of the computed matrix for section 9.1.1. Tests are performed with three different interpolations with and without algebraic reduction, and compared to the SMP method

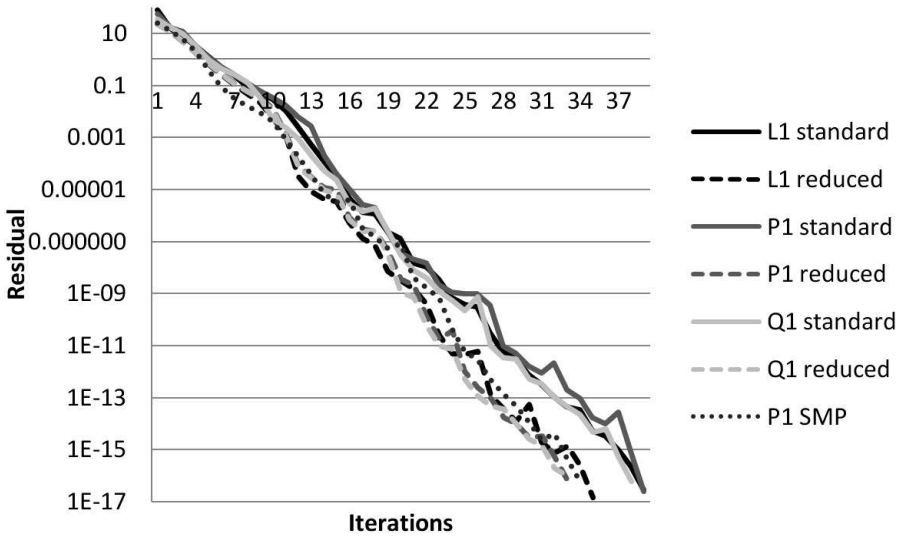


Figure 9.3: Residual against iterations of ILUK solver for problem 1 with a 128×128 mesh

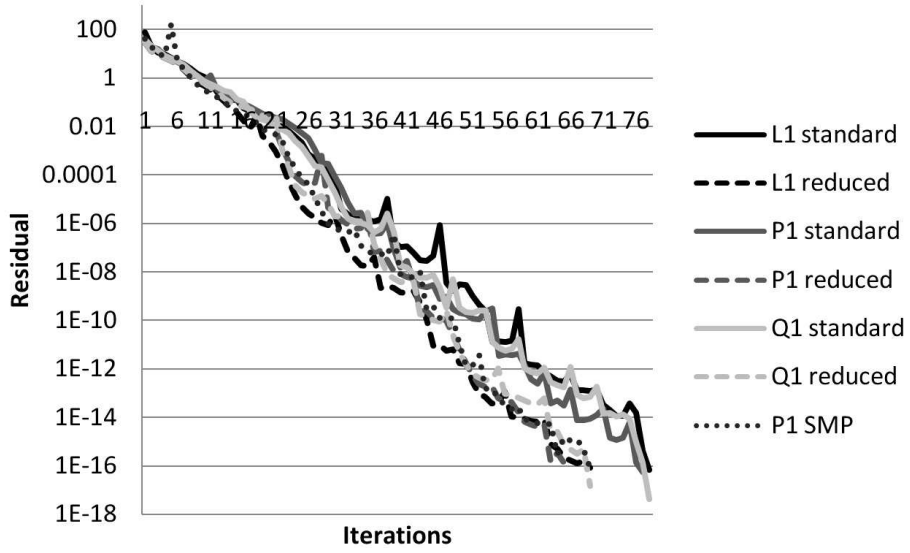


Figure 9.4: Residual against iterations of ILUK solver for problem 1 with a 256×256 mesh

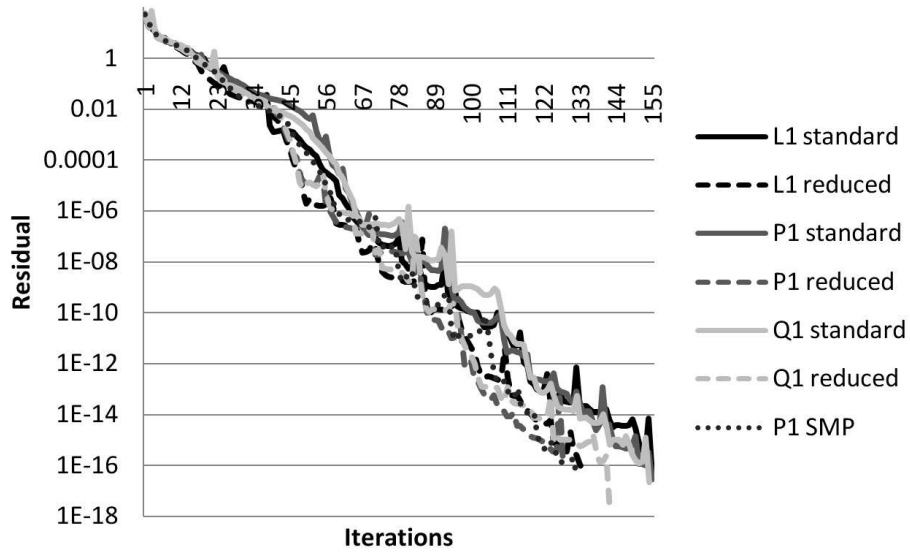


Figure 9.5: Residual against iterations of ILUK solver for problem 1 with a 512×512 mesh

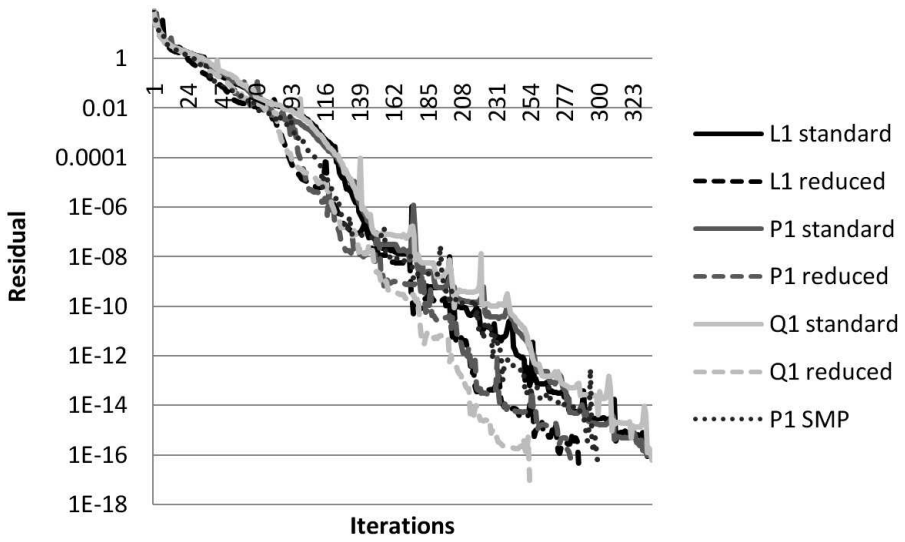


Figure 9.6: Residual against iterations of ILUK solver for problem 1 with a 1024×1024 mesh

Problem 2 The 3D equation $\Delta T = 6$ is solved. The solution is $T(r) = r^2$. The solution is imposed on an immersed centered sphere of radius 0.2. As expected, the second-order code gives the exact solution to almost computer-error accuracy without this inner boundary. Results of the numerical accuracy test with the spherical inner boundary are presented in Fig. 9.7. The results

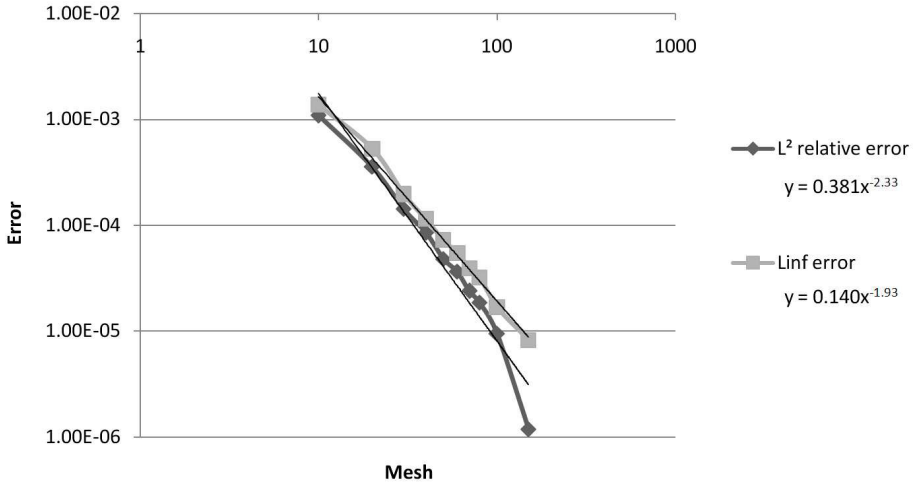


Figure 9.7: Curves of errors for problem 2

are presented in Fig. 9.8. For the L^∞ norm, the second order is regularly obtained. For the L^2 norm, the second order is not obtained for the coarsest meshes as the code has not reached its asymptotical convergence domain. As can be noticed by comparing results with and without the AIIB method, this last one does not spoil the convergence order of the code, and the presence of the immersed interface with an analytical solution imposed in Σ_h improves the accuracy. For both cases the numerical solution tends to a second order in space.

Problem 3 The 3D equation $\Delta T = 12r^2$ is solved in a unit box. The solution is $T(r) = x^4 + y^4 + z^4$. The results are presented in Fig. 9.8. For the L^∞ norm, the second order is

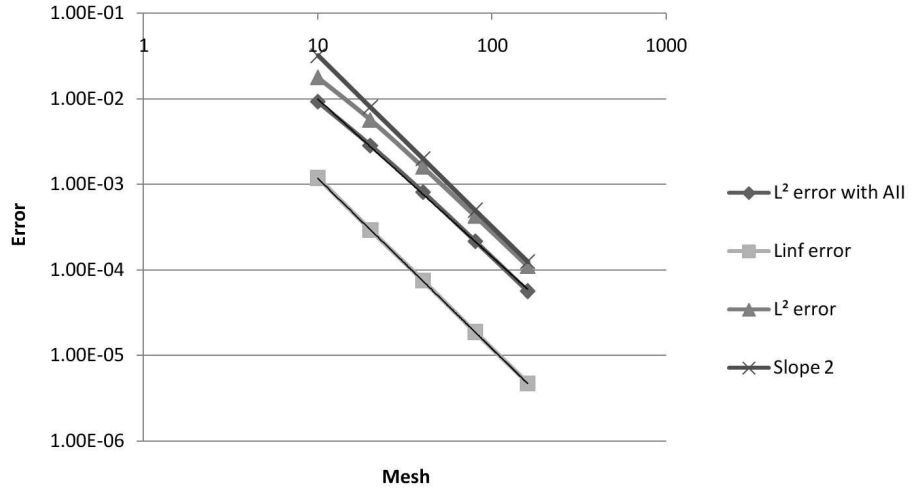


Figure 9.8: Curves of errors for problem 3

regularly obtained. For the L^2 norm, the second order is not obtained for the coarsest meshes as the code has not reached its asymptotical convergence domain. As can be noticed by comparing results with and without the AIIB method, this last method does not spoil the convergence order of the code, and the presence of the immersed interface with an analytical solution imposed in Σ_h improves the accuracy of the code. For both cases the numerical solution tends to an order two in space.

Problem 4 The 2D equation $\Delta T = 4$ is solved. The analytical solution is imposed on the boundaries of the domain and a Neuman BC is imposed on a centered circle of radius $R = 0.5$. As can be seen in Fig. 9.9, the global convergence has an average slope of 1.10. However, the convergence for the three biggest meshes reaches a slope of 2.

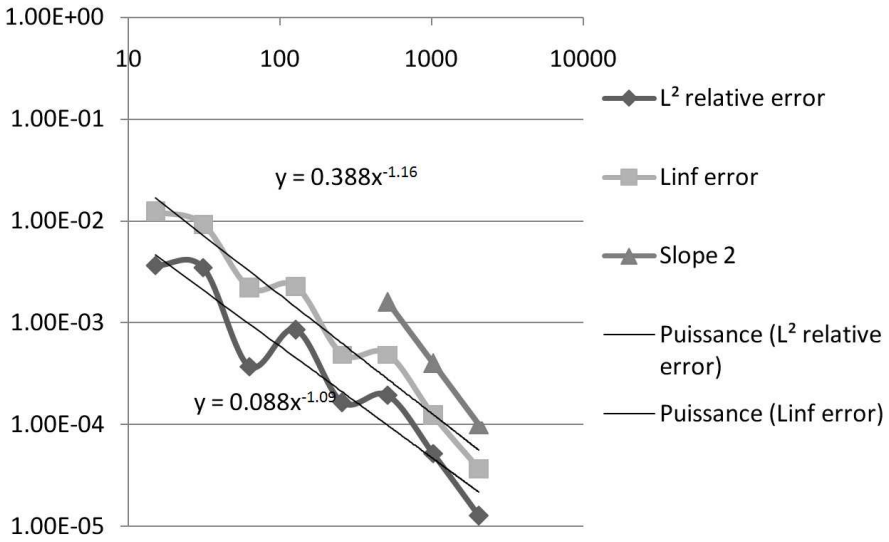


Figure 9.9: Curves of errors for problem 4

9.1.2 Convergence with the number of interface elements

Our aim here is to measure the sensibility of the method with the accuracy of the Lagrangian mesh discretizing the immersed interface. Problem 1 is solved on 32×32 and 128×128 meshes. Fig. 9.10 shows the accuracy of the solution with respect to the number of points used to discretized the interface which is here a circle. The reference solutions (Fig. 9.2) have been computed with an analytical circle. As can be seen, a second order in space is globally obtained. The reference numerical solutions for the 32×32 and 128×128 meshes are different but the sensitivity of the error to the number of points in the lagrangian mesh is almost the same.

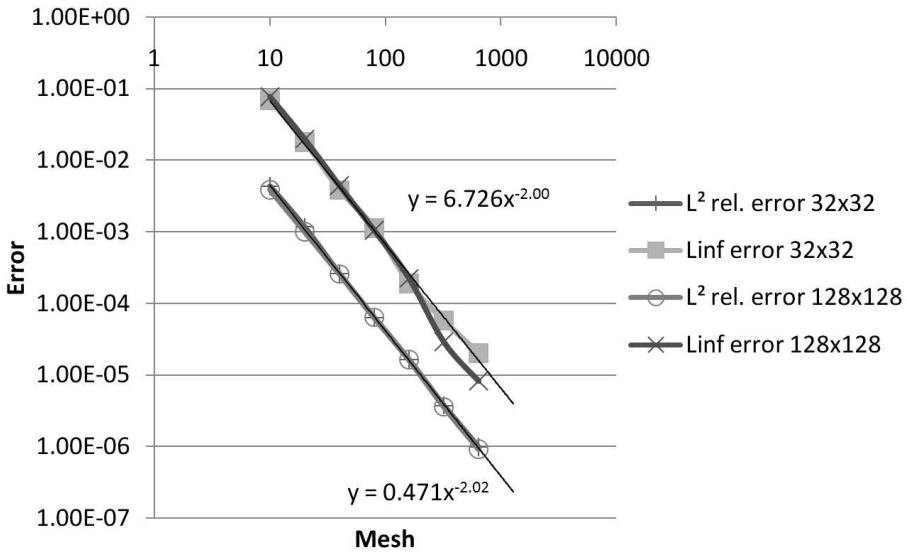


Figure 9.10: Convergence of the error with respect to the accuracy of the lagrangian shape problem 1

9.1.3 The Stanford bunny

This last case demonstrates how a second-order method enhances the representation of the boundary condition compared to a first-order method. The homogenous Laplace problem with a Dirichlet BC $T_\Sigma = 10$ is solved on a $60 \times 60 \times 50$ mesh bounding an obstacle of complex shape (the Stanford bunny). The extension of the solution in Ω_1 is used for the post treatment. Thus, all $u_J, J \in \mathcal{N}_1$ are replaced by u_J^*. Then, the iso-surface $T = T_\Sigma$ gives an idea of the approximation of the boundary condition. Fig. 9.11 shows the iso-surface for a first order method. As can be seen, the shape of the obstacle endures a rasterization effect as the solution is imposed in the entire control volumes. Fig. 9.12 shows the iso-surface for the second order AIIB method. Fig. 9.13 shows a slice of the solution passing through the bunny. As can be seen, overshoots are present inside the shape which corresponds to the auxiliary values allowing the correct solution at the Lagrangian interface points to be obtained.

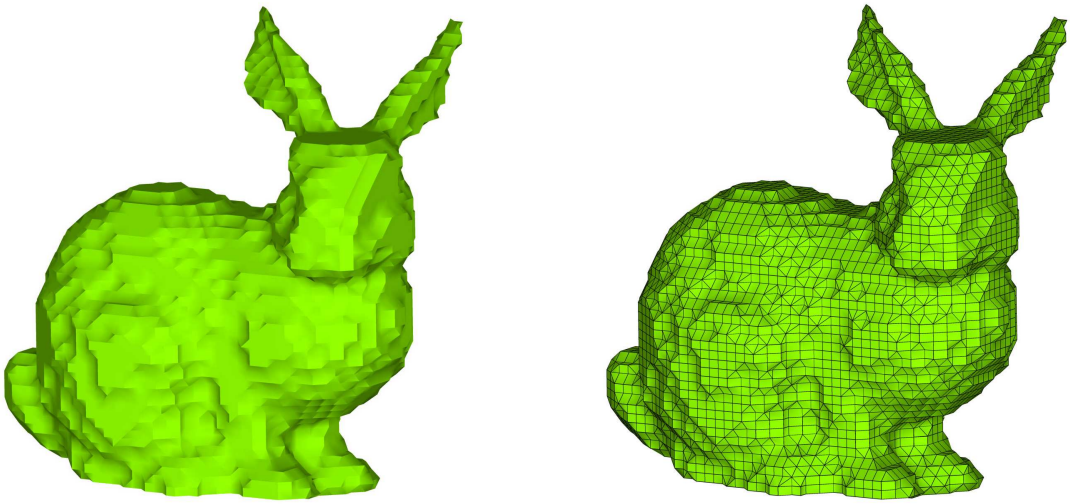


Figure 9.11: Iso-surface $T = 10$ for the Stanford bunny with a first-order method

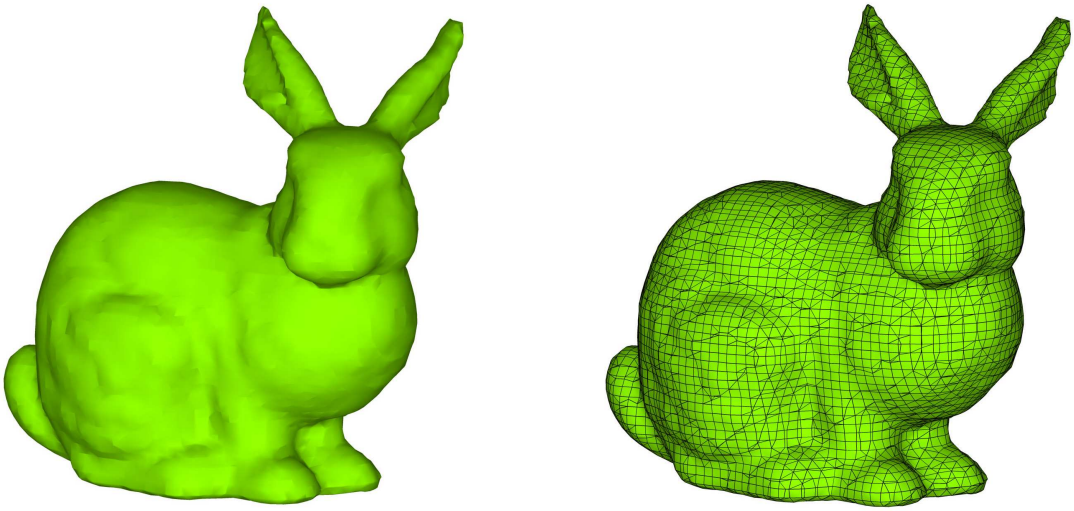


Figure 9.12: Iso-surface $T = 10$ for the Stanford bunny with a second-order method

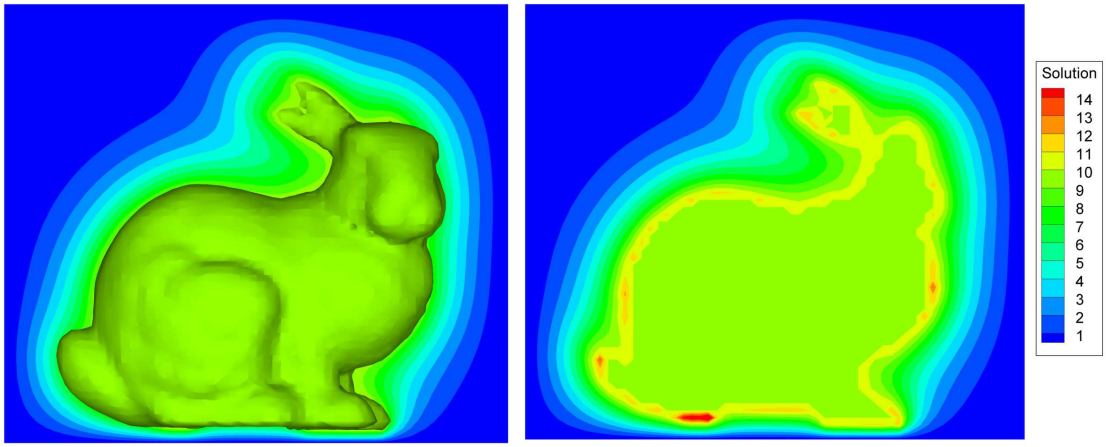


Figure 9.13: Iso-surface $T = 10$ and a slice of the solution

9.1.4 Immersed interface problems

Problem 5 The 2D problem $\mathcal{P}_{ii}$ with $f = -4$ and $a = 1$ is solved. As the equation remains the same in both domains, this problem can be solved without immersed interface method. The analytical solution is $u = r^2$. As can be expected with our second order code, computer error is reached for all meshes with or without AIIB method. The difference with problem 2, where the solution is a second-order polynomial too, is that the solution is not explicitly imposed at a given location. In the present case, the interface condition is still correct anywhere in the domain so the approximation of the interface position does not generate errors.

Fig. 9.14 shows that the same result is obtained with an interface jump such as $u = r^2$ for $r > 0.5$ and $u = r^2 + 1$ otherwise.

An equivalent quality of result is obtained with Σ such as:

$$\begin{cases} x(\alpha) = (.5 + .2\sin(5\alpha))\cos(\alpha) \\ y(\alpha) = (.5 + .2\sin(5\alpha))\sin(\alpha) \end{cases} \tag{9.2}$$

with $\alpha \in [0, 2\pi]$. The small stencil of the method allows interfaces with relatively strong

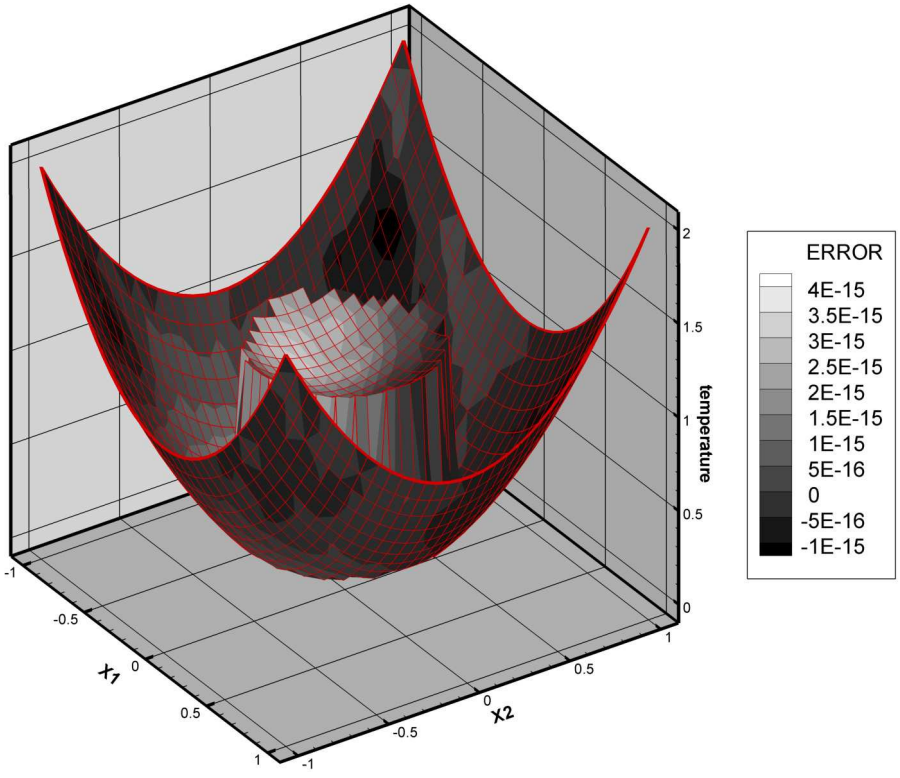


Figure 9.14: The solution and the error for problem 5 with a 32×32 mesh

curvatures to be used.

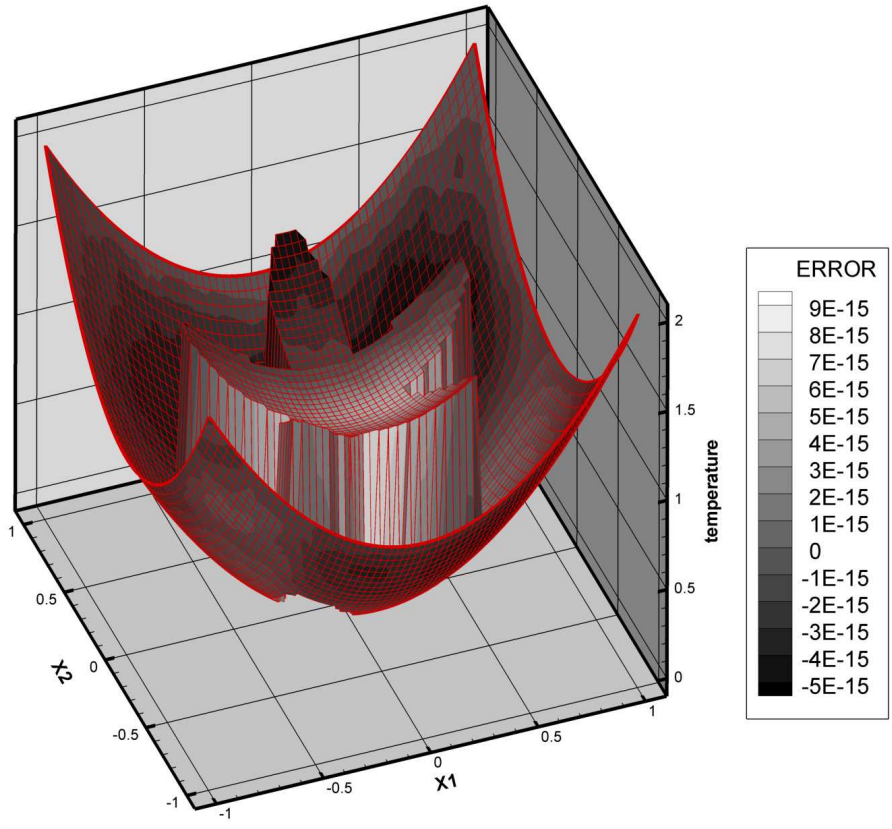


Figure 9.15: The solution and the error for problem 5 with a 64×64 mesh

Problem 6 The same problem as in 9.1.4 is now considered with a discontinuous coefficient a such as $a = 10$ in Ω_0 and $a = 1$ in Ω_1, involving the following analytical solution:

$$u(r) = \begin{cases} r^2 \text{ in } \Omega_0 \\ \frac{r^2}{10} + \frac{0.9}{4} \text{ in } \Omega_1 \end{cases} \tag{9.3}$$

Accuracy tests are first performed with the interface almost passing by some grid points (called odd mesh). The interface does not strictly lies on these points, as the shape is shifted by an ϵ. This configuration is difficult as the interpolations degenerates. Accuracy tests are then performed with a box of length 1.0001 (called even mesh). In this configuration, the interface never passes by a grid point. The results of the numerical accuracy test are presented in Fig. 9.16. For the odd series of test, the slope is 1.86 for the L^2 and L^∞ errors. For the even series, where no geometrical singularity is present, the slope for both errors is 2.04.

Figures 9.17 shows the solution and the L^2 relative error for a 32×32 mesh. As the analytical solution is imposed on the numerical boundary, the error is principally located in the interior subdomain.

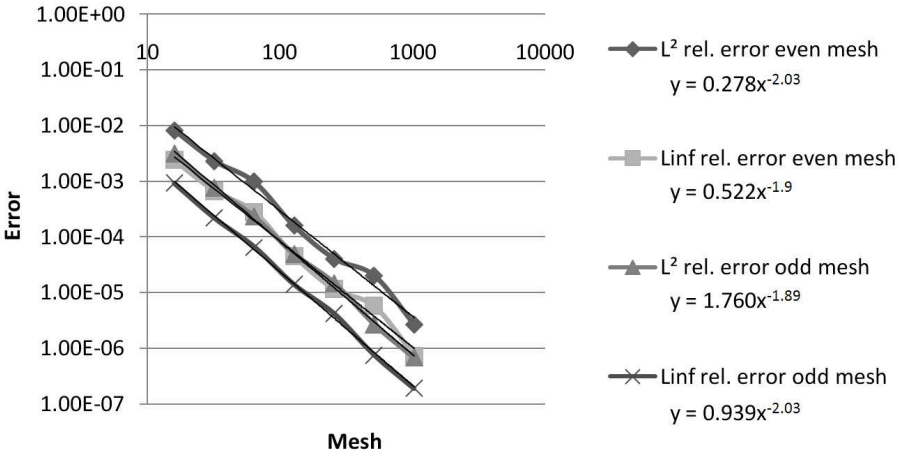


Figure 9.16: Curves of errors for odd and even meshes the problem 6

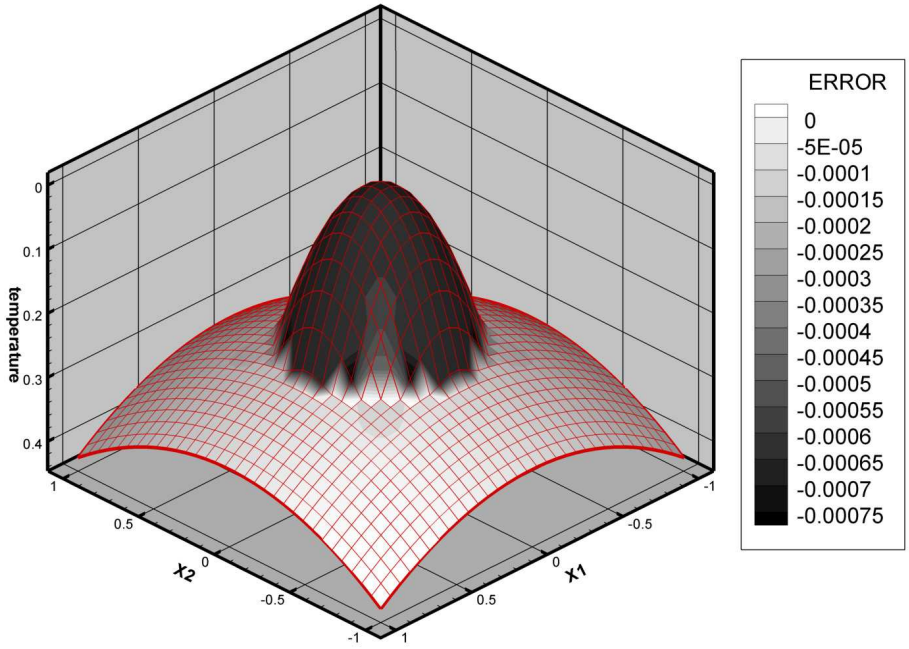


Figure 9.17: The solution and the error for problem 6 with a 33×33 mesh

Problem 7 The homogenous 2D Laplace equation is considered with the following analytical solution:

$$u(x,y) = \begin{cases} 0 \text{ in } \Omega_0 \\ e^x \cos(y) \text{ in } \Omega_1 \end{cases} \tag{9.4}$$

where Ω_0 and Ω_1 are delimited by Σ a centered circle of radius 0.5. Fig. 9.18 shows that the convergence for both L^2 and L^∞ error are of first order only. The Fig. 9.19 shows the numerical solution (which is not so different from the analytical solution) and the error map for a 32×32 mesh. In section (9.1.1), a first global order is observed too, even if a second order is reached for the three last meshes. Hence, the convergence is not as good as expected when a condition on the normal flux with a source term ($\psi \neq 0$) is imposed. Numerous trials implying interpolations of higher orders have lead to similar results, so, for now, we cannot explain the first order of convergence.

In [Tsen 03], the authors seems to have encountered the same difficulties as they explain how to impose Neumann BC with a quite similar method without performing showing a convergence test.

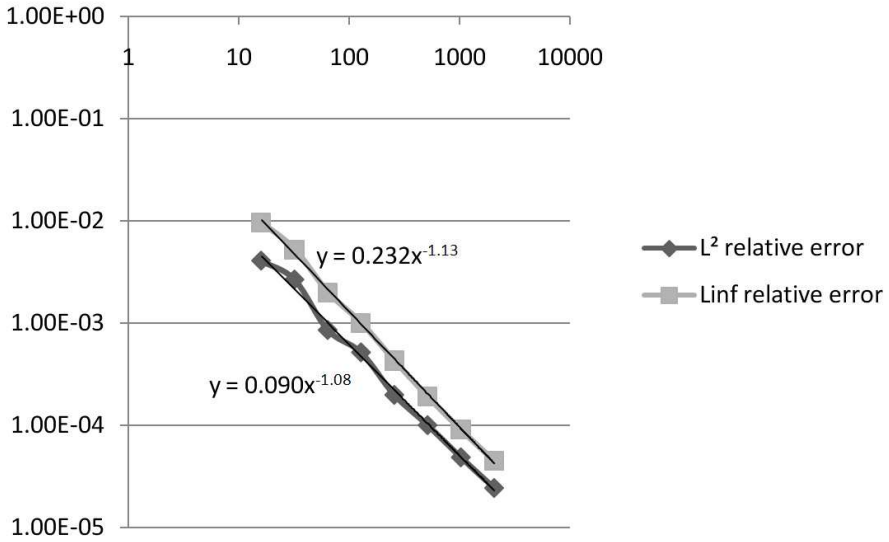


Figure 9.18: Convergence of the L^2 relative error and the L^∞ error for problem 7

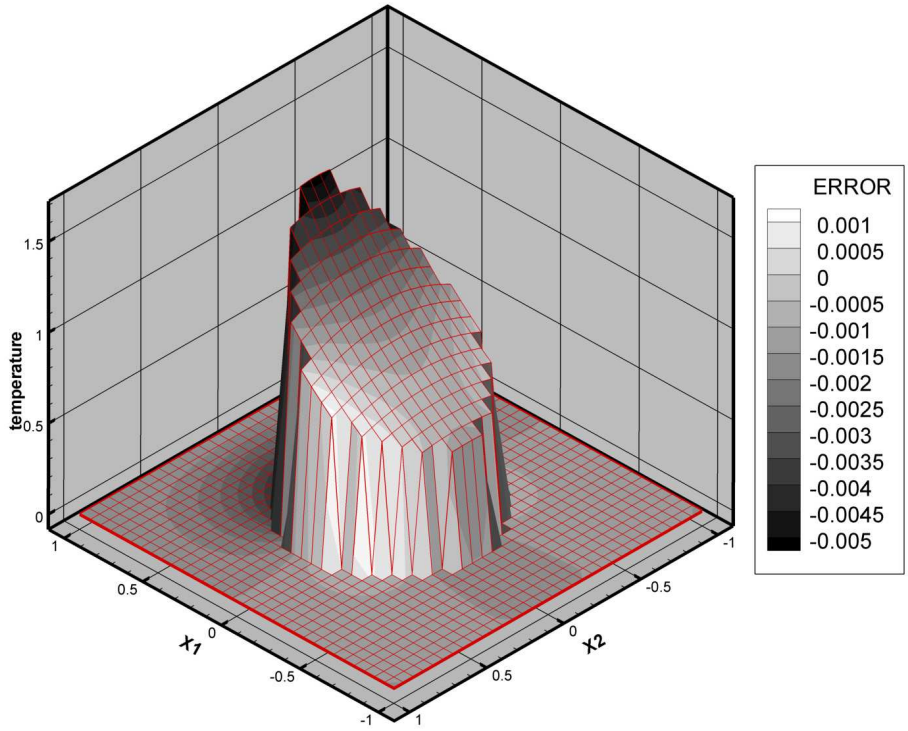


Figure 9.19: The solution and the L^2 relative error for problem 7 with a 32×32 mesh

9.1.5 Some remarks about the solvers

The kind of interpolation function used and the position of the interface have an impact on the final discretization matrix C', especially on its conditionning. Let us consider an intersection x_l of Σ_h between two points $x_J, J \in \mathcal{N}_0$ and $x_I, I \in \mathcal{N}_1$. A Dirichlet BC u_l is imposed on it. The constraint constructed with a $\mathbb{L}_1^1$ interpolation is $(1-\alpha)u_J + \alpha u_I^*$, with $\alpha = \frac{x_l - x_J}{x_I - x_J}$. Hence, $\frac{\alpha}{1-\alpha}$ tends to 0 when x_l tends to x_J. As the matrix loses its diagonal dominance, solver problems can be encountered. Tseng et al. [Tsen 03] proposed changing the interpolation by using a new node which is the image of x_I through the interface. In [Gibo 02, Gibo 05], authors pointed out this problem and suggest to slightly move the interface to a neighboring point (in our case x_J) if x_I is too close to Σ_h.

In this case, for the Dirichlet BC, an unknown u_J^* is created, and the equation in x_J is simply $u_J = u_l$. For the Neumann BC, the standard interpolation is written in x_J with u_J^* and its neighbor unknowns in Ω_0.

For the transmission conditions (8.17)-(8.18), if $\phi = 0$ and $\psi = 0$, no auxiliary unknown is created and the standard finite-volume centered discretization is used. However, for this case, or for $\phi \neq 0$ and $\psi \neq 0$, our implementation using ILUK preconditionner or a PARDISO direct solver does not necessarily require such methods, even if $\frac{\alpha}{1-\alpha} \approx 10^{-10}$.

9.2 Navier-Stokes equations

All these cases are treated with the SMP method. The related discretization and solvers are described in the Appendix A of the present document.

9.2.1 Cylindrical Couette flow

We consider a Couette flow between two cylinders of radius $R_1 = 0.5\,m$ and $R_2 = 3\,m$. Their angular velocities are $\omega_1 = 0\,rad.s^{-1}$ and $\omega_2 = 2\,rad.s^{-1}$. The solution is

$$v_\theta(r) = \frac{\omega_2 R_2^2 - \omega_1 R_1^2}{R_2^2 - R_1^2} r + \frac{(\omega_1 - \omega_2) R_2^2 R_1^2}{R_2^2 - R_1^2} \frac{1}{r} \tag{9.5}$$

The NS equations are solved in a domain $\Omega = [-0.15\,;\,0.15] \times [-0.15\,;\,0.15]$. The analytical solution is imposed on $\partial\Omega$. A penalty method is used to impose a Dirichlet BC on the inner circle. The solution for the SMPM and the VPM are compared in Fig. (9.20) with the same case simulated without IB method in polar coordinates. As expected, the SMPM reaches a second

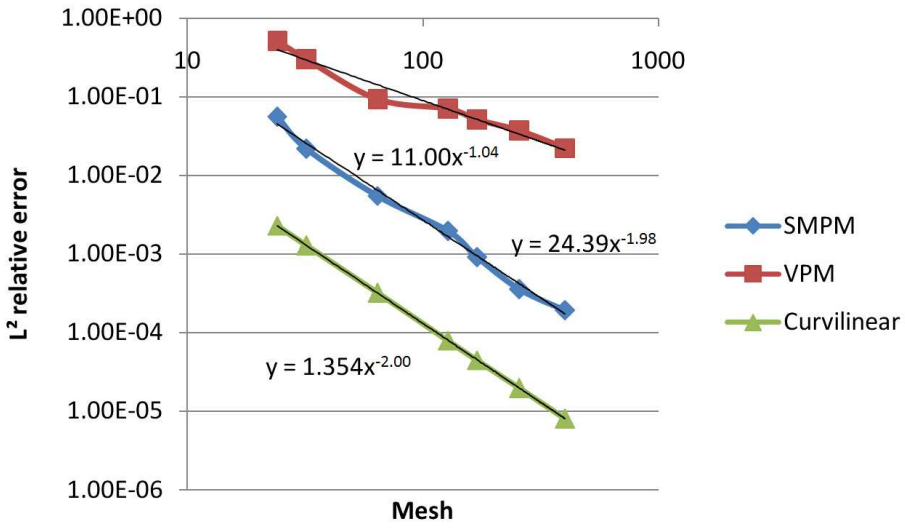


Figure 9.20: L^2 relative error on the velocity for the cylindrical Couette flow

order in space, the VPM a first order and the case in polar coordinates reach a second order. This case is 20 times more accurate than the Cartesian case with the SMPM. However, the polar mesh is boddy-fitted in this configuration and the mesh lines are colinear with the flow.

9.2.2 Flow past a cylinder

The case of the cylinder in an unbounded uniform flow is a common test for the IB methods. For very low Reynolds numbers, the flow is creepy and the streamlines are symmetric. For higher Reynolds numbers, (up to ~ 47) two symmetric steady vortices appears (see Fig. 9.21). For still higher Reynolds numbers, the vortices are no more symmetric and a wavy tail is observed (but the vortices are still attached to the particle). For even higher Reynolds numbers, an alternative shedding of the vortices, the Kàrmàn vortex street, occurs. A cylindrical particle of diameter $D = 0.1$ is immersed in a computational domain $\Omega = [0\,;\,10] \times [-4\,;\,4]$. An uniform flow $\mathbf{U} = (U_\infty, 0)$ is imposed at the left boundary. The center of the particle is located in $(1.5, 0)$.

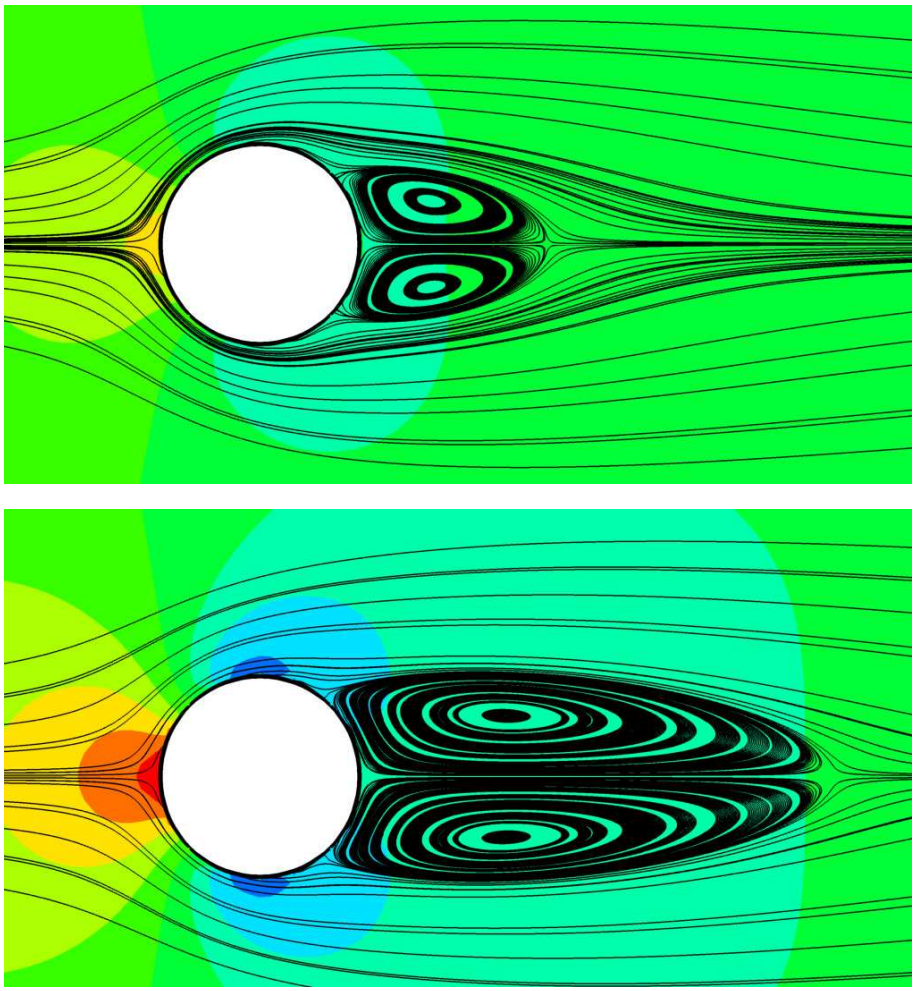


Figure 9.21: Streamlines and pressure field for a flow past a cylinder at $Re = 20$ (up) and $Re = 40$ (down)

For $Re = 20$ and $Re = 40$, a convergence study is presented in for the first (VPM) (Tab. (9.4)-(9.5)) and second-order (SMPM) (Tab. (9.6)-(9.7)) penalty method, where L is the recirculation length, a the horizontal distance between the particle and the center of the vortices, b the distance between the center of the vortices, θ_S the separation angle (see Fig. (9.22))and C_D the drag coefficient given by

$$C_D = \frac{2F_D}{\rho U_\infty^2 D} \tag{9.6}$$

with F_D the drag force.

The mesh has a constant space step in $[\,1.4\,;\,1.8\,] \times [\,-0.1\,;\,0.1\,]$. The number of cells for each directions in the constant zone is the power of 2 which is directly under half of the number of cells for this direction. For the 94×52 mesh the constant zone has 32×16 cells and the number of cells by directions follows a power of 2. For the other exterior zone, an exponential refinement is used.

The numerical error cannot be computed as there is no analytical solutions and the case theoretically requires an infinite domain. Hence, the results are generally compared with different works of the literature. The Fig. 9.4-9.7 shows a convergence study for $Re = 20$ and $Re = 40$.

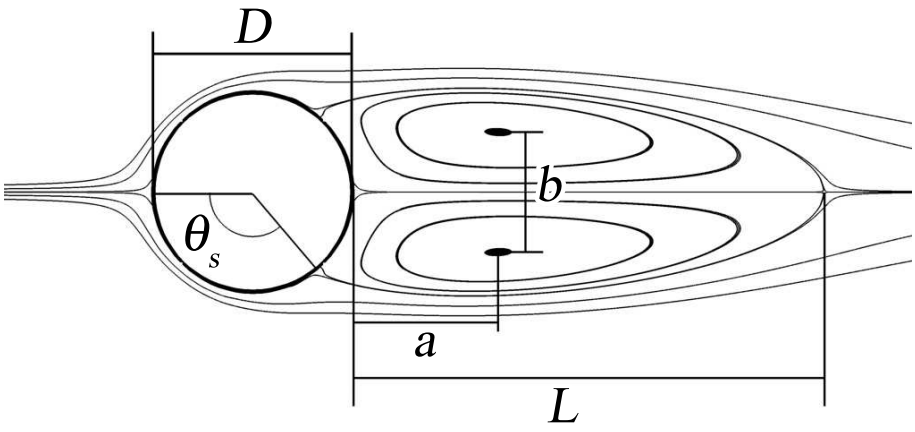


Figure 9.22: Notations for the case of the flow past a cylinder

The simulations with the SMPM seem to converge more quickly than for the VPM toward the asymptotic values.

Maillage	$\frac{L}{D}$	$\frac{a}{D}$	$\frac{b}{D}$	C_D	θ_S
94×52	0.7373	0.2758	0.3779	1.9076	140.8°
158×84	0.8540	0.3283	0.4052	2.0114	137.5°
286×148	0.9073	0.3495	0.4206	2.0371	136.8°
542×276	0.9317	0.3586	0.4281	2.0355	136.4°
1054×532	0.9454	0.3633	0.4314	2.0370	136.0°

Table 9.4: Convergence study for the flow past a cylinder at $Re = 20$ for the VPM

Maillage	$\frac{L}{D}$	$\frac{a}{D}$	$\frac{b}{D}$	C_D	θ_S
94×52	1.790	0.5729	0.5139	1.3817	138.1°
158×84	2.0874	0.6665	0.5644	1.4892	130.4°
286×148	2.2135	0.7043	0.5849	1.5141	127.8°
542×276	2.2777	0.7230	0.5957	1.5152	126.8°
1054×532	2.3089	0.7324	0.6011	1.5174	126.3°

Table 9.5: Convergence study for the flow past a cylinder at $Re = 40$ for the VPM

Maillage	$\frac{L}{D}$	$\frac{a}{D}$	$\frac{b}{D}$	C_D	θ_S
94×52	0.9397	0.358	0.4269	1.9868	141.2°
158×84	0.9423	0.3620	0.43063	2.038	137.6°
286×148	0.9473	0.3638	0.43215	2.04818	136.7°
542×276	0.9487	0.3646	0.43289	2.04788	136.5°
1054×532	0.9534	0.3661	0.4336	2.04875	135.9°

Table 9.6: Results for the flow past a cylinder at $Re = 20$ for the SMPM

Maillage	$\frac{L}{D}$	$\frac{a}{D}$	$\frac{b}{D}$	C_D	Θ_S
94×52	2.2647	0.7172	0.594097	1.45948	130.2°
158×84	2.2976	0.7275	0.5989	1.51499	126.3°
286×148	2.3122	0.7332	0.6014	1.52539	126.4°
542×276	2.3200	0.7349	0.60275	1.52790	126.3°
1054×532	2.3253	0.7371	0.6036	1.52366	126.1°

Table 9.7: Results for the flow past a cylinder at $Re = 40$ for the SMPM

The physical properties are compared with the literature in Tab. (9.8) for a mesh size of 1984×1152. All our results are in good agreement with the literature. The drag coefficient C_D is slightly upside the other values. The value of C_D is almost the same for the first and second-order methods. Practically, both methods are quite different but produce quite similar results, so the treatment of the flow near the boundary seems correct. The differences with the literature can come from the configuration of the case as the domain has to be theoretically unbounded. However, C_D is the only result of our study which is overestimated. As this parameters depends on the calculation of the wall forces our routine possibly overestimates the drag force.

	$\frac{L}{D}$	$\frac{a}{D}$	$\frac{b}{D}$	C_D	Θ_S
$Re = 20$					
Coutanceau et Bouard [Cout 77]	0.93	0.33	0.46	–	135°
Dennis et Chang [Denn 70]	0.94	–	–	2.05	136.3°
Le et al. [Le 06]	0.93	–	–	2.05	–
Ye et al. [Ye 99]	0.92	–	–	2.03	–
Russell and Wang [Russ 71]	0.94	–	–	2.13	–
Linnick et Fasel [Linn 05]	0.93	0.36	0.43	2.06	136.5°
He et al. [He 97]	0.921	–	–	2.152	137.04°
Patil et al. [Pati 09]	0.942	–	–	1.949	137.19°
Taira et al. [Tair 07]	0.94	0.37	0.43	2.06	136.7°
VPM	0.9454	0.3633	0.4314	2.037	136.0°
SMPM	0.9534	0.3661	0.4336	2.04875	135.9°
$Re = 40$					
Coutanceau et Bouard [Cout 77]	2.13	0.76	0.59	–	126.2°
Dennis et Chang [Denn 70]	2.35	–	–	1.522	126.2
Le et al. [Le 06]	2.22	–	–	1.56	–
Ye et al. [Ye 99]	2.27	–	–	1.52	–
Russell et Wang [Russ 71]	2.29	–	–	1.60	–
Linnick et Fasel [Linn 05]	2.28	0.72	0.60	1.54	126.4°
He et al. [He 97]	2.245	–	–	1.499	127.16°
Patil et al. [Pati 09]	2.142	–	–	1.558	127.26°
Taira et al. [Tair 07]	2.30	0.73	0.60	1.54	126.3°
Tseng et a. [Tsen 03]	2.21	–	–	1.53	–
VPM	2.309	0.732	0.601	1.517	126.3°
SMPM	2.325	0.737	0.604	1.524	126.1°

Table 9.8: Results for the flow past a cylinder at $Re = 20$ and $Re = 40$ and comparison with the literature

The next case is simulated for a Reynold number of 100. For this regime, the flow is periodic and its frequency is caracterized through the Strouhal number St:

$$St = \frac{fD}{U_\infty} \tag{9.7}$$

with f the vortex shedding frequency. A convergence study for the VPM and the SMPM is presented in Tab. (9.9)-(9.10)). The drag and lift coefficients converge more quickly with the SMPM. The convergence of the forces shows that the local convergence of the flow around the particle is faster with the SMPM. The convergence of the Strouhal number is quite the same for both methods but seems still far from its asymptotical value. The global flow (characterized by the Strouhal number) seems more influenced by the calculation mesh than by the treatment of the immersed boundary.

The results are compared with the literature in Tab. (9.8). The drag coefficient C_D varies not much for the literature and our approach underestimates this coefficient. For the lift coefficient C_L, the results are more varying and our results are in the same range as for the literature. Concerning the Strouhal number, it is underestimated in our approaches. As shows by the convergence study, the setting of the calculation mesh could be involved.

Maillage	C_D	C_L	S_t
124×72	0.96 ± 0.01	± 0.147	0.117
248×144	1.19 ± 0.01	± 0.264	0.134
496×288	1.27 ± 0.01	± 0.3081	0.147
992×576	1.31 ± 0.01	± 0.3271	0.155

Table 9.9: Convergence study for the flow past a cylinder at $Re = 100$ for the VPM

Maillage	C_D	C_L	S_t
124×72	1.08 ± 0.01	± 0.2413	0.113
248×144	1.24 ± 0.01	± 0.3185	0.132
496×288	1.30 ± 0.01	± 0.338	0.146
992×576	1.33 ± 0.01	± 0.340	0.155

Table 9.10: Convergence study for the flow past a cylinder at $Re = 100$ for the SMPM

		C_D	C_L	S_t
$Re = 100$	Braza et al. [Braz 86]	1.36 ± 0.015	± 0.25	–
	Kim et al. [Kim 01]	1.33	± 0.32	0.165
	Liu et al. [Liu 98]	1.35 ± 0.012	± 0.339	0.164
	Le et al. [Le 06]	1.37 ± 0.009	± 0.323	0.16
	VPM	1.31 ± 0.01	± 0.327	0.156
	SMPM	1.33 ± 0.01	± 0.340	0.155

Table 9.11: Results for the flow past a cylinder at $Re = 100$ and comparison with the literature

Discussion and conclusion of Part III

Two new fictitious domain methods have been proposed. The SMP method is the first second-order discretization of the L^2 penalty method and is fully implicit. The method can be applied to elliptic equations or to Navier-Stokes equations. Concerning this last case, a particular attention has been paid to the velocity-pressure coupling. No modification has been required for the standard augmented Lagrangian method as the divergence free constraint and the penalty constraint can coexist without interference in a unique discretization matrix. That is not enough for the case of a pressure projection approach where the velocity is corrected afterwards which removes the penalty constraint. However, a similar correction approach to [Iken 07] has been used and the same results have been obtained with both velocity-pressure coupling method. Concerning the application of the method to the curvilinear grids, the Part II of the present document has demonstrated that the second-order accuracy is retrieved on curvilinear grids for a Dirichlet problem.

It as been shown that the SMP method was not well suited to the interface problem. By replacing penalty terms with constraints on auxiliary nodes, a new simple immersed interface method, the AIIB method, using algebraic manipulations has been presented. This method is able to treat elliptic equations with discontinuous coefficients and solution jumps over complex interfaces. A second order in space is reached for several configurations with minor modifications of the original code. The interface conditions are dicretized with a compact stencil, so the AIIB approach is directly able to treat interfaces with strong curvatures even if a particular treatment of geometric singularities can be required. As the modified matrix looses its diagonal dominance, efficient solvers are required.

For the immersed boundary problems with a Dirichlet BC, the method has shown a second order of convergence in space for various kinds of interpolations. For some interpolations, the method is equivalent to the SMP method. An algebraic reduction has been applied to accelerate the convergence of the solver. For the Neumann BC, a second order seems to be reachable for the densest meshes.

For the immersed interfaces, a second order of convergence in space is obtained when the jump of the normal flux is null, even if the equation has discontinuous coefficients. An algebraic reduction is not possible, but compared to the MIB method [Zhou 06b] or to the IIM [Leve 94], this new method has a simplest formulation and uses a smaller stencil.

Future works will be devoted to extend the accuracy of the method when the jump of the normal flux is not zero, and to extend the method to the Navier-Stokes equations with immersed interfaces. Our general aim is to treat complex moving fluid/solid and fluid/fluid interfaces using both AIIB method and ITP method [Rand 05] to obtain an accurate two-way coupling.

The next part of this document will show an application of the SMP to the moving objects.

Part V

Solid mechanics and fluid-structure coupling

Table of Contents

Chapter 10

Solid mechanics

THE present work proposes a quite complete study of the cases of immobile interfaces. An extension to moving interfaces involves additional aspects:

- The fictitious domain method and its associated algorithms (Ray-Casting, Level-Set, etc...) used to treat the object have to be very robust. When thousands of time steps are involved, the slightest bug in the repeated algorithms has a good chance to appear through the time.
- Concerning the computational cost, the methodology can be quite slow for immobile obstacles as the algorithms are repeated one time only. For moving objects, the methodology has to be highly optimized.
- Depending on the methodology, modeling moving interfaces is more than repeating the algorithms at each time step. As will be shown latter, the nodes passing from one side of the interfaces to the other one have to be treated specifically.
- The interface movement at each time step has to be modeled. Analytical functions or physical laws can be used.

The first point is treated in details in the other parts of this document. This chapter deals with the physical laws involved in the displacement of interfaces, the related modeling and the adaptation of the initial static algorithms.

10.1 Base principles

Let us consider solid objects at a given time t. Their interior is denoted as $\Omega_i(t)$ and their frontier $\partial\Omega_i(t)$. The position of their center of mass is $\mathbf{X}_i(t)$, their velocity is $\mathbf{U}_i(t) = \dot{\mathbf{X}}_i(t)$, their acceleration $\mathbf{A}_i(t) = \dot{\mathbf{U}}_i(t)$. Their angular position is denoted $\boldsymbol{\theta}_i(t)$ and their angular velocity $\boldsymbol{\omega}_i(t) = \dot{\boldsymbol{\theta}}_i(t)$. External resulting force applied to the objects are denoted as $\mathbf{F}_i(t)$ and the external torque is denoted as $\mathbf{T}_i(t)$. Under the rigid-body motion assumption, the velocity $\mathbf{u}$ at a point $\mathbf{x} \in \Omega_i(t)$ is

$$\mathbf{u}(t,\mathbf{x}) = \mathbf{U}_i(t) + \boldsymbol{\omega}_i(t) \times \mathbf{r}(t,\mathbf{x}) \text{ in } \Omega_i(t) \tag{10.1}$$

with $\mathbf{r}(t,\mathbf{x}) = \mathbf{x} - \mathbf{X}_i(t)$ and $\mathbf{X}_i(t) = (X_i(t), Y_i(t), Z_i(t))$. The evolution of the ith object is described by the Euler-Newton equations:

$$\begin{cases} m_i \mathbf{A}_i(t) = \mathbf{F}_i(t) & \text{(10.2a)} \\ \dfrac{\mathrm{d}\mathbf{H}_i(t)}{\mathrm{d}t} = \mathbf{T}_i(t) & \text{(10.2b)} \end{cases}$$

with $\mathbf{H}_i(t) = \mathbf{I}_i(t)\boldsymbol{\omega}_i(t)$ and $\mathbf{I}_i(t)$ the inertia tensor defined as

$$\mathbf{I}_i(t) = \int_{\Omega_1} \rho_i(\mathbf{r}^2(t,\mathbf{x})\mathbb{I}_d - \mathbf{r}(t,\mathbf{x}) \otimes \mathbf{r}(t,\mathbf{x}))\ \mathrm{d}\mathbf{x}. \tag{10.3}$$

For the next formulation, the time dependance (t) is not notified. Practically, the following form of (10.3) is used:

$$\mathbf{I} = \int_{\Omega_i} \rho_i \begin{pmatrix} (y-Y_i)^2+(z-Z_i)^2 & -(y-Y_i)(x-X_i) & -(z-Z_i)(x-X_i) \\ -(x-X_i)(y-Y_i) & (x-X_i)^2+(z-Z_i)^2 & -(z-Z_i)(y-Y_i) \\ -(z-Z_i)(x-X_i) & -(y-Y_i)(x-X_i) & (x-X_i)^2+(y-Y_i)^2 \end{pmatrix} \mathrm{d}V. \tag{10.4}$$

Using

$$\frac{\mathrm{d}\mathbf{I}_i}{\mathrm{d}t}\boldsymbol{\omega}_i = \boldsymbol{\omega}_i \times (\mathbf{I}_i.\boldsymbol{\omega}_i) \tag{10.5}$$

the Eq. (10.2b)) can be expanded as

$$\mathbf{I}_i\frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = \mathbf{T}_i - \boldsymbol{\omega}_i \times (\mathbf{I}_i.\boldsymbol{\omega}_i). \tag{10.6}$$

The demonstration of (10.5)[Bost 08b] uses the following property for 3 vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$:

$$(\mathbf{a} \otimes \mathbf{b}).\mathbf{c} = (\mathbf{b}.\mathbf{c})\mathbf{a}. \tag{10.7}$$

$$\begin{aligned} \frac{\mathrm{d}\mathbf{I}_i(t)}{\mathrm{d}t}\boldsymbol{\omega}_i(t) &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\int_{\Omega_i} \rho_i(\mathbf{r}^2(t,\mathbf{x})\mathbb{I}_d - \mathbf{r}(t,\mathbf{x}) \otimes \mathbf{r}(t,\mathbf{x}))\ \mathrm{d}\mathbf{x}\right)\boldsymbol{\omega}_i(t) \\ &= \int_{\Omega_i} \rho_i \frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{r}^2(t,\mathbf{x})\mathbb{I}_d - \mathbf{r}(t,\mathbf{x}) \otimes \mathbf{r}(t,\mathbf{x}))\boldsymbol{\omega}_i(t)\ \mathrm{d}\mathbf{x} \\ &= \int_{\Omega_i} \rho_i \left(2\left(\mathbf{r}\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)\boldsymbol{\omega}_i - \left(\mathbf{r} \otimes \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)\boldsymbol{\omega}_i - \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} \otimes \mathbf{r}\right)\boldsymbol{\omega}_i\right)\ \mathrm{d}\mathbf{x} \\ &= \int_{\Omega_i} \rho_i \left(2\left(\mathbf{r}\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)\boldsymbol{\omega}_i - \left(\boldsymbol{\omega}_i\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)\mathbf{r} - (\boldsymbol{\omega}_i\mathbf{r})\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)\ \mathrm{d}\mathbf{x}. \end{aligned} \tag{10.8}$$

Yet,

$$\begin{aligned} \frac{\mathrm{d}\mathbf{r}(t,\mathbf{x})}{\mathrm{d}t} &= \frac{\mathrm{d}(\mathbf{x} - \mathbf{X}_i(t))}{\mathrm{d}t} \\ &= -\frac{\partial\mathbf{X}_i(t)}{\partial t} + (\dot{\mathbf{X}}_i(t) + \boldsymbol{\omega}_i(t) \times \mathbf{r}(t,x)).\nabla(\mathbf{x} - \mathbf{X}_i(t)) \\ &= -\dot{\mathbf{X}}_i(t) + (\dot{\mathbf{X}}_i(t) + \boldsymbol{\omega}_i \times \mathbf{r}(t,\mathbf{x})).\mathbb{I} \\ &= \boldsymbol{\omega}_i(t) \times \mathbf{r}(t,\mathbf{x}). \end{aligned} \tag{10.9}$$

Hence,

$$\begin{aligned} \frac{\mathrm{d}\mathbf{I}_i(t)}{\mathrm{d}t}\boldsymbol{\omega}_i(t) &= -\int_{\Omega_i(t)} \rho_i(\boldsymbol{\omega}_i(t).\mathbf{r}(t,\mathbf{x}))(\boldsymbol{\omega}_i(t) \times \mathbf{r}(t,\mathbf{x})\ \mathrm{d}x \\ &= -\boldsymbol{\omega}_i(t) \times \int_{\Omega_i(t)} \rho_i(\boldsymbol{\omega}_i(t).\mathbf{r}(t,\mathbf{x}))\mathbf{r}(t,\mathbf{x})\ \mathrm{d}x \\ &= -\boldsymbol{\omega}_i(t) \times \int_{\Omega_i(t)} \rho_i((\mathbf{r}(t,\mathbf{x}) \otimes \mathbf{r}(t,\mathbf{x})).\boldsymbol{\omega}_i(t))\ \mathrm{d}x \\ &= -\boldsymbol{\omega}_i(t) \times \left(\int_{\Omega_i(t)} \rho_i(\mathbf{r}(t,\mathbf{x}) \otimes \mathbf{r}(t,\mathbf{x}))\ \mathrm{d}x.\boldsymbol{\omega}_i(t)\right) \\ &= \boldsymbol{\omega}_i(t) \times (\mathbf{I}_i(t).\boldsymbol{\omega}_i(t)) - \boldsymbol{\omega}_i(t) \times \left(\int_{\Omega_i(t)} \rho_i(\mathbf{r}(t,\mathbf{x}))^2\mathbb{I}\ \mathrm{d}x.\boldsymbol{\omega}_i(t)\right) \\ &= \boldsymbol{\omega}_i(t) \times (\mathbf{I}_i(t).\boldsymbol{\omega}_i(t)) \end{aligned} \tag{10.10}$$

as the form of the last term is $\mathbf{a} \times (C\mathbf{a})$ and so is equal to zero. Hence,(10.5) is recovered.
For solids with three orthogonal symmetry planes (sphere, ellipsoid, cube, box...), one can find a frame for which only the diagonal terms of the inertia matrix are non-zero. These diagonal terms are denoted as I_1, I_2 and I_3. Hence, one can develop the $\boldsymbol{\omega} \times \mathbf{H}$ term:

$$\boldsymbol{\omega} \times \mathbf{H} = \begin{pmatrix} (I_3 - I_2)\omega_2\omega_3 \\ (I_1 - I_3)\omega_3\omega_1 \\ (I_2 - I_1)\omega_1\omega_2 \end{pmatrix} \tag{10.11}$$

Hence, $\boldsymbol{\omega}_i \times \mathbf{H}_i$ is null[1] for a solid with $I_1 = I_2 = I_3$, such as spheres or cubes. Eq. (10.2b) can be written as:

$$\begin{cases} \mathbf{A}_i = \dfrac{\mathbf{F}_i}{m_i} \\ \dot{\boldsymbol{\omega}}_i = \dfrac{\mathbf{T}_i}{\mathbf{I}_i} \end{cases} \tag{10.12}$$

For the other types of objects, the system (10.2) is written as:

$$\begin{cases} \mathbf{A}_i = \dfrac{\mathbf{F}_i}{m_i} \\ \dot{\boldsymbol{\omega}}_i = \dfrac{\mathbf{T}_i - \boldsymbol{\omega}_i \times (\mathbf{I}_i.\boldsymbol{\omega}_i)}{\mathbf{I}_i} \end{cases} \tag{10.13}$$

With such a formulation, the velocity $\mathbf{V}$ and rotation $\boldsymbol{\theta}$ are easily computed through thanks to a time integration scheme. From $\mathbf{V}$, the same scheme can be used to find $\mathbf{X}$. With a first-order Euler scheme, we obtain:

$$\begin{cases} \mathbf{V}^{n+1} = \mathbf{V}^n + \Delta t \dfrac{\mathbf{F}}{m}^{n+1} \\ \mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t \mathbf{V}^{n+1} \\ \boldsymbol{\omega}^{n+1} = \boldsymbol{\omega}^n + \Delta t \dfrac{\mathbf{T}^{n+1} - \boldsymbol{\omega}^n \times \mathbf{H}^n}{\mathbf{I}} \\ \boldsymbol{\theta}_i^{n+1} = \boldsymbol{\theta}_i^n + \Delta t \boldsymbol{\omega}^{n+1} \end{cases} \tag{10.14}$$

The external forces applied to an object are

- The gravity
- The air resistance
- The friction with the other moving objects
- The friction with the ground
- The impulsion generated by solid-solid contact

10.2 Numerical computation of the inertia matrix

A numerical computation of the inertia matrix and the volume of a polyhedron can be found in [Mirt 96] where the triangularized surface is used. As the purpose here is to perform a fluid-structure coupling, some volume fonctions of the objects are always available, especially the VOF

[1] In 2D, the third component only is considered and $\boldsymbol{\omega}_1 = \boldsymbol{\omega}_2 = 0$, so $\boldsymbol{\omega} \times \mathbf{H}$ is null too.

function C_i of the solid media. The mass of a solid i is calculated as follows:

$$m_i = \rho_i \sum_{K_j \in \mathcal{T}_h} meas(K_j) C_i(\mathbf{x}_j). \tag{10.15}$$

For the inertia matrix,

$$\mathbf{I} = \rho_i \sum_{K_j \in \mathcal{T}_h} meas(K_j) C_i(\mathbf{x}_j) \mathbf{I}_j \tag{10.16}$$

with

$$\mathbf{I}_j = \begin{pmatrix} (y_j - Y_i)^2 + (z_j - Z_i)^2 & -(y_j - Y_i)(x_j - X_i) & -(z_j - Z_i)(x_j - X_i) \\ -(x_j - X_i)(y_j - Y_i) & (x_j - X_i)^2 + (z_j - Z_i)^2 & -(z_j - Z_i)(y_j - Y_i) \\ -(z_j - Z_i)(x_j - X_i) & -(y_j - Y_i)(x_j - X_i) & (x_j - X_i)^2 + (y_j - Y_i)^2 \end{pmatrix}. \tag{10.17}$$

This matrix is symmetric, so invertible and one can find a frame for which $\mathbf{I}$ is the diagonal matrix $\mathbf{I}_d$ (its diagonal terms are denoted in this case I_1, I_2 and I_3). As the solid generally rotates, the inertia matrix has to be calculated at each time step. However, the full computation of the inertia matrix can be avoided using a transformation matrix $R(t)$ such that at any time, we have $\mathbf{I} = R\mathbf{I}_d R^T$.

10.3 Solid-solid collisions modeling

10.3.1 Model

10.3.1.1 Time advancement

The only moving object treated here are spheres so $\boldsymbol{\omega} \times \mathbf{H}$ is null. Hence, the time integration of the position and rotation for the ith particle with an Euler scheme is

$$\begin{cases} \mathbf{V}_i^{n+1} = \mathbf{V}_i^n + \Delta t \dfrac{\mathbf{F}_i}{m_i}^{n+1} \\ \mathbf{X}_i^{n+1} = \mathbf{X}_i^n + \Delta t \mathbf{V}_i^{n+1} \\ \boldsymbol{\omega}_i^{n+1} = \boldsymbol{\omega}_i^n + \Delta t \dfrac{\mathbf{T}_i^{n+1}}{\mathbf{I}_i} \\ \boldsymbol{\theta}_i^{n+1} = \boldsymbol{\theta}_i^n + \Delta t \boldsymbol{\omega}^{n+1} \end{cases} \tag{10.18}$$

10.3.1.2 External forces and torques

Gravity The gravity is a simple force term $\mathbf{F}_i^g = m_i \mathbf{g}$ where $\mathbf{g} = 9.81 m.s^{-1}$ is the standard gravity acceleration.

Friction with the other objects This force is produced by the differential of the tangential velocities of two objects at the contact point. Let us consider two particles P_i and P_j. Their contact point is $\mathbf{x}$. The differential tangential velocity is $\mathbf{V}_{ij}^t = (\mathbf{V}_j - \mathbf{V}_i)(\mathbb{I}_d - \mathbf{n})$ for the translation part. For the rotation part, $\mathbf{V}_{ij}^r = \boldsymbol{\omega}_j \times \mathbf{r}_j - \boldsymbol{\omega}_i \times \mathbf{r}_i$ where $\mathbf{r}_i = \mathbf{x} - \mathbf{X}_i$. Hence, the friction force is

$$\mathbf{F}_i^t = \mu_s (\mathbf{V}_{ij}^t + \mathbf{V}_{ij}^r) \tag{10.19}$$

with μ_s the friction coefficient between the two surfaces.

10.3.1.3 Collisions

A simple algorithm is used. At the beginning of each time step, a collision test is performed between all the objects. If two spheres of radii r_i and r_j collides, their center of mass $\mathbf{X}_i$ and $\mathbf{X}_j$ are such that $|\mathbf{X}_i - \mathbf{X}_j| < |(r_i + r_j)|/2$. Concerning the collisions between a sphere and the ground, the distance is calculated using the distance from a point to a triangle algorithm (*see* sction 6.3.1). In this first implementation, all particles have the same mass. When a collision occurs between two spheres, the velocity of each particles is modified by the normal component of the relative velocity, $(\mathbf{V}_j - \mathbf{V}_i)\mathbf{n}_{ij}$ and a bounce coefficient c_b. Hence,

$$\mathbf{V}_i := \mathbf{V}_i - \frac{1 + c_b}{2}(\mathbf{V}_j - \mathbf{V}_i)\mathbf{n}_{ij} \tag{10.20}$$

$$\mathbf{V}_j := \mathbf{V}_j + \frac{1 + c_b}{2}(\mathbf{V}_j - \mathbf{V}_i)\mathbf{n}_{ij} \tag{10.21}$$

and the distance between the center of mass of the sphere is increased such as $|\mathbf{X}_i - \mathbf{X}_j| = |(\mathbf{r}_i + \mathbf{r}_j)|/2$. For a sphere-ground collision, the resulting velocity of the sphere is

$$\mathbf{V}_i := \mathbf{V}_i - c_b(\mathbf{V}_i)\mathbf{n} \tag{10.22}$$

where $\mathbf{n}$ is the normal of the ground at the contact point.

10.3.2 Implementation of a solid mechanics code: *Dresden*

Defining an efficient model for the solid-solid collision is not trivial, even for simple solids such as spheres. The hardest point is the detection and the management of the collisions. Various methods can be found in [Guen 03]

Implementing a model directly in a complex CFD code such as Thétis is not the simplest way to build step-by-step a solid mechanics code. A new solid mechanics code *Dresden* has been developed with the following requirements:

1: management of objects of different kind (primitives or complex meshes)

2: easy testing of collision models between many objects of different kind,

3: easy visualization.

The first point implies the use of an oriented object programming language. The third requirement can be fulfilled thanks to the OpenGL library. Hence, the $C++$ language has been chosen.

10.4 Illustration

Two cases are presented. For each, particles with random positions are dropped over a triangularized mesh.

10.4.1 Gathering

The principle of this case is to drop particles in a bowl. The topography will gather them, and at the final time, all the particles would have to be immobile. As the particles are in horizontal stacking, forces are constantly exerted on them. As in the algorithm a force produce a movement, it is quite hard to converge to an immobile state. The Fig. 10.1 shows the evolution of the particles for this case. A quasi immobile state is obtained.

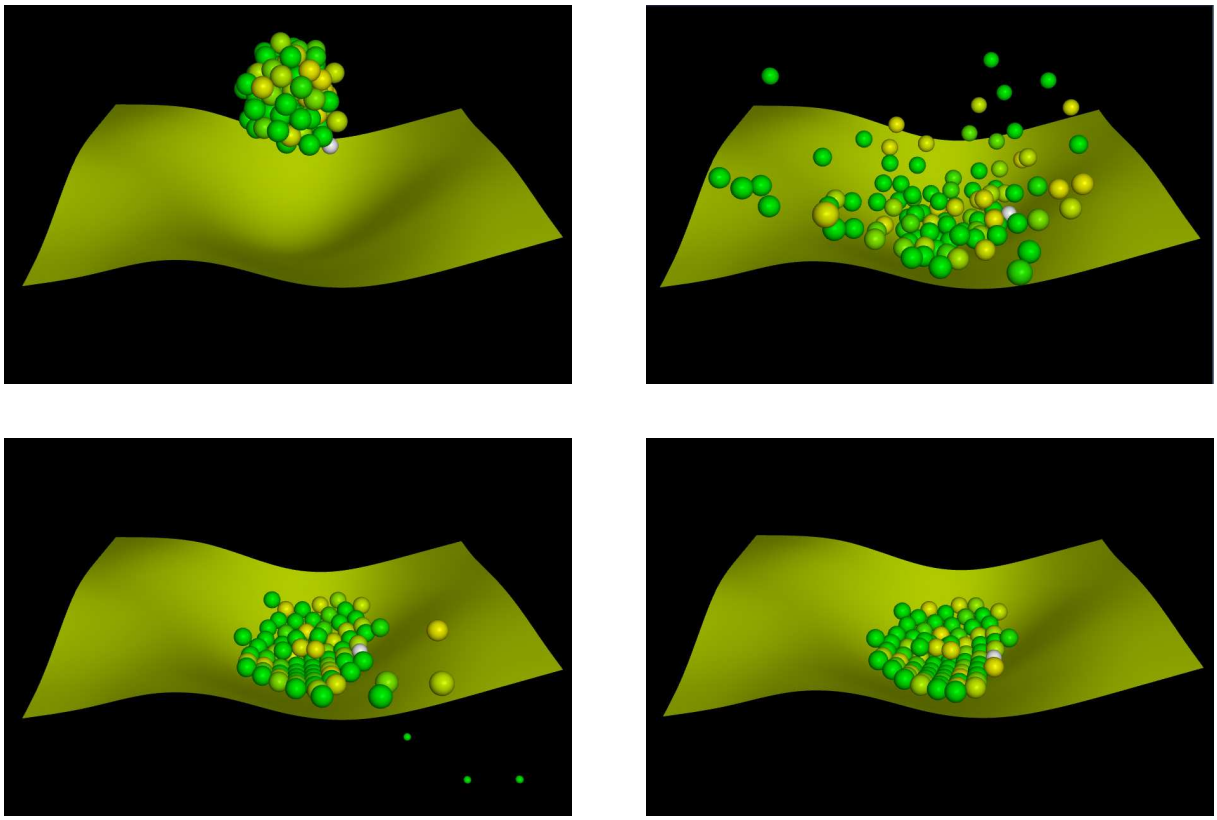


Figure 10.1: View of the gathering case for 100 particles

10.4.2 Stacking

One of the hardest things to obtain with solid simulation is a static stacking. The explicit collision algorithms moves the objects one by one. If an object is moved to correct an intersection with a second object, the first can then intersect a third object. With a stacking, many object are in contact with more than one other object. The Fig. (10.2) shows the evolution of the particles for this stacking case. The evolution of the kinetic energy is plotted in Fig. 10.4.2 for a case with 50 particles and two time steps $\Delta t = 0.005s$ and $\Delta t = 0.0025s$. As the bounce coefficient are set as quite small here, the kinetic energy has to reach quickly a null value. The parasitic movements are reduced with the smaller time step. The same kind of time step is used for fluid simulation, but the solid simulation performed here is almost at real-time, so if a coupling with fluid simulation is required, very smaller time steps can be used for the solid part. However, we have use a quite simple method and the algorithm has to be revised till an absolutely null energy is obtained. The algorithm of [Guen 03] could be considered. One of its slight drawback is that the transition between a slow motion and immobility is sharp.

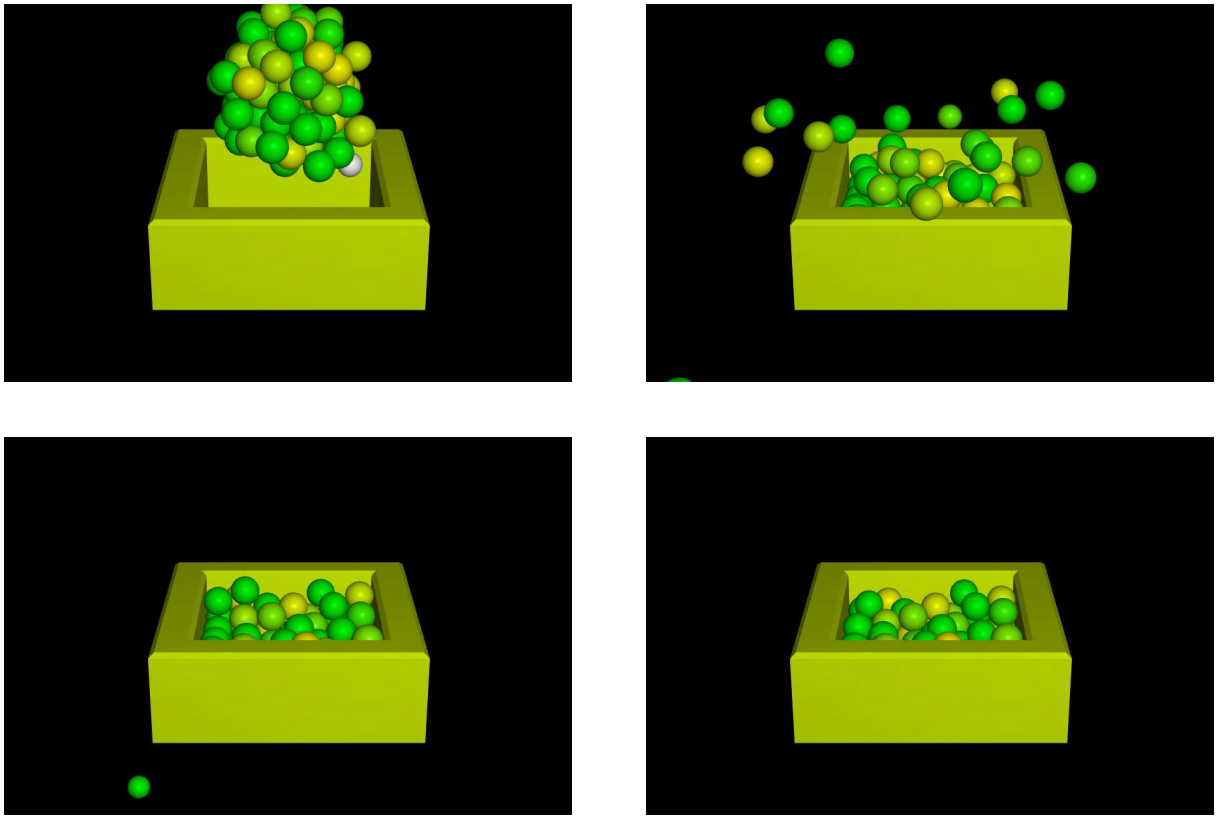


Figure 10.2: View of the stacking case for 100 particles

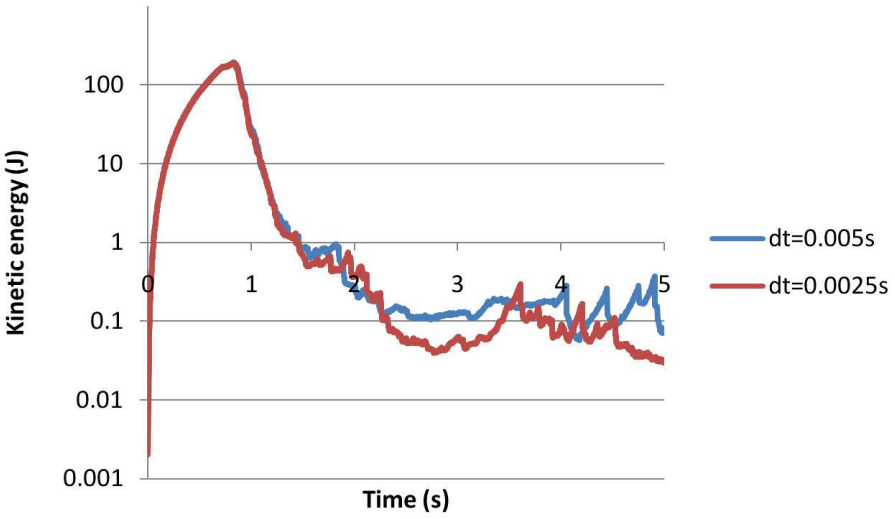


Figure 10.3: Time evolution of the kinetic energy (logarithmic axis) of a 50 particle system

Chapter 11

Fluid-structure interaction

11.1 Formulation and time coupling

The fluid-structure coupling is one of the major aims of the fictitious domain methods for the Navier-Stokes equations. The first difficulty lies in the amount and the diversity of the numerical methods required to treat the coupling. The coupling method itself is one of the key point and determines which complementary methods will be implied. In a one-way coupling, the solution is desired in one media only (the fluid or the solid). The case where an object has an analytical velocity is considered as a one-way coupling. Methods allowing a one-way coupling to be fully performed have been treated in the previous chapters of the present document. In a two-way coupling, both fluid and solid are dynamically interacting each other. The two media are governed by *a priori* different physical laws and a simultaneous resolution of all the equations involved is *a priori* a delicate issue. For instance, the incompressible Navier-Stokes equations with Dirichlet boundary conditions are used in the fluid domain Ω_0

$$\begin{cases} \rho\left(\dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = \rho\mathbf{g} + \nabla\cdot\boldsymbol{\sigma} \text{ in } \Omega_0 & \text{(11.1a)} \\ \nabla\cdot\mathbf{u} = 0 \text{ in } \Omega_0 & \text{(11.1b)} \\ \mathbf{u} = \mathbf{u}_{|\partial\Omega} \text{ on } \partial\Omega & \text{(11.1c)} \\ \mathbf{u} = \mathbf{u}_{|\partial\Omega_i} \text{ on } \partial\Omega_i & \text{(11.1d)} \end{cases}$$

with $\boldsymbol{\sigma} = 2\mu\mathbf{D} - p\mathbb{I}$ the stress tensor and $\mathbf{D} = \frac{1}{2}(\nabla\mathbf{u} + \nabla^T\mathbf{u})$ the deformation tensor. The constraint (11.1d) is a consequence of the non-slip condition required on the boundary $\partial\Omega_i$ of the objects.

In the solid domain, the equations of the solid dynamic can be used:

$$\begin{cases} m_i\dfrac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = \mathbf{F}_i \\ \mathbf{I}_i\dfrac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = \mathbf{T}_i - \boldsymbol{\omega}_i\times(\mathbf{I}_i.\boldsymbol{\omega}_i) \end{cases} \quad \text{(11.2a)}$$

where $\mathbf{F}_i$ and $\mathbf{T}_i$ are defined in the present case as

$$\mathbf{F}_i = \int_{\Omega_i}\mathbf{g}\,\mathrm{d}x + \int_{\partial\Omega_i}\boldsymbol{\sigma}.\mathbf{n}\,\mathrm{d}x \quad \text{(11.3)}$$

$$\mathbf{T}_i = \int_{\partial\Omega_i}(\mathbf{x}-\mathbf{X}_i)\times\boldsymbol{\sigma}.\mathbf{n}\,\mathrm{d}x. \quad \text{(11.4)}$$

This formulation requires two alternate resolution steps. The Eqs. (11.1) are solved with a Dirichlet condition on $\partial\Omega_i$ given by the velocity of the solids at the precedent time step. The resulting wall forces exerted on $\partial\Omega_i$ are then used to solve Eqs (11.2). This method is used in [John 96] and seems to have the worst temporal accuracy. However, an iterative process can be performed inside a same time step to increase the temporal accuracy [Hu 92]. Concerning the solution on the Eulerian grid inside the solid, one can impose the solid velocity for $\mathbf{u}$ in Ω_i with the rigid motion relation

$$\mathbf{u}(x) = \mathbf{U}_i + \boldsymbol{\omega}_i \times (\mathbf{x} - \mathbf{X}_i) \tag{11.5}$$

This approach is used in [Shar 05]. In [Coqu 08], authors use this method with a penalization formulation while the Navier-Stokes equations are solved with a vortex method [Cott 04]. A theoretical study of the method is presented in [Bost 08a, Bost 08b]. In [Lamb 09], authors simulate flows inside an isolated rotating tank and its destabilization. As a rotating frame is used, the walls are immobile while complementary terms are added to the Navier-Stokes equations.

The DLM method of Glowinski (see section 3.3) proposes a time implicit formulation of the coupling by using a variational formulation. However, a time-splitting method is used to solve the unique set of equations and the coupling cannot be considered as implicit.

A fully implicit method is obtained with the ITP method (see section 2.3) where the stress tensor is penalized while the divergence free constraint is ensured with an augmented Lagragian method.

For the spatial coupling, see the part II of the present document.

All these approaches use the following property:

Property 11.1.1 $\mathbf{u}(\mathbf{x}) = \mathbf{U}_i + \boldsymbol{\omega}_i \times (\mathbf{x} - \mathbf{X}_i) \Leftrightarrow \mathbf{D} = 0$

The proof of this classical result can be found in [Tema 01, Bost 08b, Lefe 07].

In the present work, the coupling approach of [Coqu 08] is combined with the SMP method (section 7)and the AL method (section A.3.4). The following walkthrough is performed

- The penalized NS equations are solved

$$\begin{aligned} &\rho\left(\frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n \cdot \nabla\tilde{\mathbf{u}}^{n+1}\right) - \nabla(dr\nabla\cdot\tilde{\mathbf{u}}^{n+1}) \\ &= -\nabla p^n + \rho\mathbf{g} + \nabla\cdot[\mu(\nabla\tilde{\mathbf{u}}^{n+1} + \nabla^T\tilde{\mathbf{u}}^{n+1})] + \beta(\sum_{k/\mathbf{x}_k\in\mathcal{N}_1^*}\alpha_k\tilde{\mathbf{u}}_k^{n+1} - \mathbf{u}_{|\partial\Omega_i}) \end{aligned} \tag{11.6}$$

  The pressure is then updated using

$$\tilde{p}^{n+1} = p^n - dr\nabla\cdot\tilde{\mathbf{u}}^{n+1} \tag{11.7}$$

  with $(\tilde{\mathbf{u}}^{n+1}, \tilde{p}^{n+1})$ the solution of this first step.

- The fluid wall forces $\mathbf{F}_i^f$ and torques $\mathbf{T}_i^f$ applied to the solids are computed

- The solid dynamic is updated with

$$\forall i \in \mathcal{O}, \begin{cases} \mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \Delta t\frac{\mathbf{F}_i}{m_i}^{n+1} \\ \mathbf{X}_i^{n+1} = \mathbf{X}_i^n + \Delta t\mathbf{U}_i^{n+1} \\ \boldsymbol{\omega}_i^{n+1} = \boldsymbol{\omega}_i^n + \Delta t\frac{\mathbf{T}_i^{n+1} - \boldsymbol{\omega}_i^n \times \mathbf{H}_i^n}{\mathbf{I}_i} \\ \boldsymbol{\theta}_i^{n+1} = \boldsymbol{\theta}_i^n + \Delta t\boldsymbol{\omega}_i^{n+1} \end{cases} \tag{11.8}$$

  with $\mathcal{O}$ the set of indexes of the solids.

- The Eulerian implicit surface functions of each objects (χ_i, ϕ_i, and C_i) are updated
- The Eulerian velocity and pressure in the objects are updated. At the Eulerian nodes $\mathbf{x}$ in $\Omega_i(t^{n+1})$, the new solution is

$$\bar{\mathbf{u}}^{n+1}(\mathbf{x}) = \mathbf{U}_i^{n+1} + \boldsymbol{\omega}_i^{n+1} \times (\mathbf{x} - \mathbf{X}_i^{n+1}) \tag{11.9}$$

$$p^{n+1} = 0 \tag{11.10}$$

An additional correction has to be done. Here comes the main difference with the first-order penalty algorithm. As explained in chapter 7, the solution $(\tilde{\mathbf{u}}^{n+1}, \tilde{p}^{n+1})$ is not physical for the nodes in Ω_i near $\partial\Omega_i$. At time t^{n+1} the previous step put a physical solution in the solids. However, the field $(\tilde{\mathbf{u}}^{n+1}, \tilde{p}^{n+1})$ is obtained for solids located at $\Omega_i(t)$ while the correction is applied in $\Omega_i(t^{n+1})$. Hence, the correction has to be extended to the nodes for which $|\chi_i(t^n)(\mathbf{x})|.|1 - \chi_i(t^{n+1})(\mathbf{x})| = 1$. We define a new Heaviside function $\bar{\chi}_i(t^{n+1}). = \max(\chi_i(t^{n+1}), |\chi_i(t^n)(\mathbf{x})|.|1 - \chi_i(t^{n+1})(\mathbf{x})|)$ The velocity $\bar{\mathbf{u}}^{n+1}(\mathbf{x})$ in the whole domain is

$$\bar{\mathbf{u}}^{n+1}(\mathbf{x}) = \sum_{j \in \mathcal{O}} \bar{\chi}_j \left(\mathbf{U}_j^{n+1} + \boldsymbol{\omega}_j^{n+1} \times (\mathbf{x} - \mathbf{X}_j^{n+1}) \right). \tag{11.11}$$

The final correction consists in solving

$$\frac{\partial \mathbf{u}}{\partial t} = \beta \bar{\chi} (\bar{\mathbf{u}} - \mathbf{u}) \tag{11.12}$$

with $\bar{\chi} = \sum_{j \in \mathcal{O}} \bar{\chi}_j$. As in [Coqu 08], we choose $\beta = 1/\Delta t$ so the following semi-discrete form is obtained:

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\Delta t} = \frac{\bar{\chi}^{n+1}(\bar{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^{n+1})}{\Delta t}. \tag{11.13}$$

Hence,

$$\mathbf{u}^{n+1} = \begin{cases} \bar{\mathbf{u}}^{n+1} \text{ where } \bar{\chi} = 1 & (11.14) \\ \tilde{\mathbf{u}}^{n+1} \text{ elsewhere} & (11.15) \end{cases}$$

One can notice that the algorithm for the first-order method is retrieved if for all objects $\bar{\chi}_i = \chi_i$.

Remark 11.1.1 *The solution is divergence-free inside each subdomains. However, the solution is not generally divergence-free for the whole domain as its derivatives are not continue at the solid-fluid interfaces.*

11.2 Wall force calculation

11.2.1 Theoretical definitions

The fluid force $\mathbf{F}_i^f$ applied to a particle P_i is

$$\mathbf{F}_i^f = \int_{\partial\Omega_i} \boldsymbol{\sigma}.\mathbf{n} \, dS = \int_{\partial\Omega_i} (2\mu\mathbf{D} - pId) \, .\mathbf{n} \, dS \tag{11.16}$$

The resulting force is decomposed in two parts:

$$\mathbf{F}_i^p = -\int_{\partial\Omega_i} (pId)\,.\mathbf{n}\; dS \tag{11.17}$$

$$\mathbf{F}_i^v = \int_{\partial\Omega_i} (2\mu\mathbf{D})\,.\mathbf{n}\; dS \tag{11.18}$$

with $\mathbf{F}_i^p$ the pressure force and $\mathbf{F}_i^v$ the viscous force.

11.2.2 Numerical method

The numerical computation of the wall forces are often used in the literature of the fictitious domain methods where the accuracy of the drag force of a particle is a common test case. Nonetheless, the details of such a calculation are rarely explained. For the first-order methods, the integral method proposed by Caltagirone [Calt 94] is simple and well suited. When higher orders fictitious methods are used, a more accurate calculation is desirable. The approach described here is close to the one explained in [Mark 08]. One can suppose that many authors use the same approach.

The theoretical force calculation requires to know the pressure and the deformation tensor of the velocity on the fluid-solid interface. The tire surface is discretized as a Lagrangian surface composed of triangular elements. The fluid field is only defined outside of the object (exceptions will be discussed later) and the interface does not generally match the location where the quantities are discreetly expressed (the Eulerian nodes). As a consequence, a quantity has to be extrapolated from the fluid domain to the interface and the first step is to define points in the fluid to build the interpolations. For each element σ_i of Σ_h (the discretization of Σ), we define $\mathbf{x}_i$ the barycenter of the element and $\tilde{\mathbf{n}}_i$ its outward local normal at $\mathbf{x}_i$. Fluid points are then defined at the locations

$$\mathbf{x}_{ik} = \mathbf{x}_i + k\frac{\tilde{\mathbf{n}}_i}{\max(\Delta x, \Delta y, \Delta z)},\; k = 1, .., 3 \tag{11.19}$$

The discrete values of a given field Φ (pressure, derivative...) are then interpolated on the Lagrangian points $\mathbf{x}_{ik}$. For each $\mathbf{x}_{ik}$, the Eulerian points used to interpolate Φ in $\mathbf{x}_{ik}$ are denoted $\mathbf{x}_{ik}^j$, $j = 1, ..., S_{max}$ with S_{max} the number of Eulerian points of the stencil of the considered interpolation function. Using two or three Lagrangian points and more related Eulerian points to interpolate Φ on σ_i produces an interpolation with a large stencil. Furthermore, each extrapolated value of a component of D is itself a centered derivative which enlarges again the stencil. Hence, two constraints opposed themselves:

- The calculation and the interpolation on the points x_{ik} of the stress tensor requires a large stencil, so the tensor has not to be taken too close to the considered element σ_i.
- A point inside the solid could be accidently used to compute the extrapolated values of D if the Lagrangian points x_{ik} is taken too far from σ_i.

To a smaller degree, these constraints remain valid for the computation of the pressure at the interface. The occurrences of such problems varies according to the complexity of the interface. A convex shape does not generally induces such effects and the study of the wall forces on a sphere is easily performed. For shapes with concavity such as a tire with complex patterns, the troubles increases with the curvature of the shape.

However, if the ratio of the magnitudes of the pressure forces and the viscous forces is such that the viscous forces are negligible, the wall forces calculation is easier to perform. The pressure field is directly available on the grid, and one can simply take its value at the closest fluid node. If a calculation of higher order of the pressure is required, two methods can be used to prevent the troubles induced by the curvature of the interface.

Here are some methods to increase the robustness and the accuracy of the method:

- A simple solution is to use adaptive interpolations such as the kernel functions (see appendix C.2). The kernel functions are commonly used to interpolate a quantity from a Lagrangian grid such as a particle field. Hence, they are well designed to take only valid nodes. The same effect can be obtained by using a combination of polynomial interpolations which requires an additional implementation effort.

- Each quantity cannot be directly interpolated from the Eulerian grid to the interface as the quantities does not numerically exists, or are not relevant, in the solid media. The idea is to extend the quantities from the fluid domain to the solid domain. Let us consider a quantity Φ. The field Φ^{n+1} is obtained from the resolution of the Navier-Stokes equations at time $n+1$. The fluid domain is Ω_0 and Ω_1 is the solid one. The following problem is solved:

$$\nabla^2\Phi' + \frac{1}{\varepsilon}(\Phi' - \Phi^{n+1}) = 0 \tag{11.20}$$

  with ε a penalty parameter such as

$$\begin{aligned} \varepsilon &\to 0 \text{ in } \Omega_0 \\ \varepsilon &\to +\infty \text{ in } \Omega_1 \end{aligned}$$

  The resulting property,

  Property 11.2.1 $\Phi^{n+1} \in \mathcal{C}^0(\Omega) \Rightarrow \Phi' \in \mathcal{C}^0(\Omega)$

  allows us to interpolate Φ' from the Eulerian grid to Σ_h. As a Neumann condition is not imposed on Σ for the problem (11.20), Φ' is not in $\mathcal{C}^1$. Hence, if Φ' is the velocity, one cannot use it to calculate $\mathbf{D}$.

- The number of points $\mathbf{x}_{ik}$ involved in the extrapolation has to be chosen according to the global topology. If two elements σ_l of Σ_h are face to face, taking a $\mathbf{x}_{ik}$ to compute a quantity for a first element is not ideal if $\mathbf{x}_{ik}$ is closer to the other element. The Level-Set function ϕ can be used to detect such a situation. For a given element, the points $\mathbf{x}_{ik}$ are taken on the ray $\mathcal{R}$ parametrized as:

$$\mathbf{x}_{\mathcal{R}}(t) = \mathbf{x}_l + t\mathbf{n},\ t \in \mathbb{R}^+ \tag{11.21}$$

  For a fixed element σ_l of Σ_h, let us consider two variable parameters t_1, $t_2 \in \mathbb{R}^+$ such as $0 < t_1 < t_2$, two constants t_{ab}, $t_{bc} \in \mathbb{R}^+$ such as $0 < t_1 < t_2$ and three subsets of $\mathbb{R}^+$: $\mathcal{R}_a$, $\mathcal{R}_b$ and $\mathcal{R}_c$ defined as follows:

$$\forall t \in \mathcal{R}_a, \forall t' \in \mathcal{R}_b, t < t' \tag{11.22}$$

$$\forall t \in \mathcal{R}_b, \forall t' \in \mathcal{R}_c, t < t' \tag{11.23}$$

$$\mathcal{R}_a = \{t \in \mathbb{R}^+, \phi(t_2) < \phi(t_1) \text{ and } \phi(t_2) - \phi(t_1) = |\mathbf{x}_{\mathcal{R}}(t_2) - \mathbf{x}_{\mathcal{R}}(t_1)|\} \tag{11.24}$$

$$\mathcal{R}_b = \{t \in \mathbb{R}^+, \phi(t_2) < \phi(t_1) \text{ and } \phi(t_2) - \phi(t_1) \neq |\mathbf{x}_{\mathcal{R}}(t_2) - \mathbf{x}_{\mathcal{R}}(t_1)|\} \tag{11.25}$$

$$\mathcal{R}_c = \{t \in \mathbb{R}^+, \phi(t_2) > \phi(t_1)\} \tag{11.26}$$

  the following properties are deduced:

Property 11.2.2 *$t \in \mathcal{R}_a \Rightarrow |\mathbf{x}_\mathcal{R}(t) - \mathbf{x}_l| = \phi(\mathbf{x}_\mathcal{R}(t))$, i.e. σ_l is the closest element to $\mathbf{x}_\mathcal{R}(t)$.*

Property 11.2.3 *ϕ is generally not differentiable in $\mathbf{x}_\mathcal{R}(t_{ab})$.*

Property 11.2.4 *ϕ is generally not differentiable in $\mathbf{x}_\mathcal{R}(t_{bc})$ and $\mathbf{x}_\mathcal{R}(t_{bc})$ is a local extremum of ϕ.*

Hence, using a point $\mathbf{x}_\mathcal{R}(t)$ when $t \in \mathcal{R}_a$ is relevant, and is irrelevant if $t \in \mathcal{R}_c$ as the physics in $\mathbf{x}_\mathcal{R}(t)$ cannot be considered as local from the point of view of σ_l. The property 11.2.3 can be used to build a surface $\mathcal{T}_b$. The property 11.2.4 can be used to build a surface $\mathcal{T}_c$. Then, a point $\mathbf{x}_\mathcal{R}(t)$ can be taken for the extrapolation of Φ while the ray $\mathcal{R}$ has not crossed $\mathcal{T}_c$ (*i.e.* $t < t_{bc}$) and eventually $\mathcal{T}_b$ (*i.e.* $t < t_{ab}$). This last point has to be studied further.

As a conclusion, if the field Φ is prolonged, the $\mathbb{Q}_1^d$ functions can be directly used. If the initial field Φ^{n+1} is used, the kernel functions are more designated.

The locations of the various requested quantities are not the same if a staggered grid is used (which is the case in the present work). Let us write the 2D deformation tensor:

$$\mathbf{D} = \begin{pmatrix} \dfrac{\partial u}{\partial x} & \dfrac{\partial v}{\partial x} \\ \dfrac{\partial u}{\partial y} & \dfrac{\partial v}{\partial y} \end{pmatrix} \tag{11.27}$$

On a staggered grid, the natural position (when centered derivative are used) of the diagonal terms are the pressure nodes. For the extra-diagonal terms, their natural locations is neither at the pressure nodes nor at the velocity nodes. The Fig. 11.1 shows the natural location of the components of the tensor. In Thétis, these nodes are called *viscosity nodes* (One can notice that the combination of the locations of the pressure, velocity and viscosity nodes defines a new regular grid). These components can be interpolated to the pressure nodes to obtain $\mathbf{D}$ in a unique node

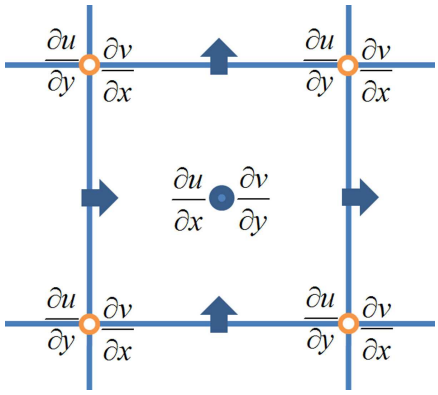


Figure 11.1: Location of the components of the deformation tensor

in order to use a unique extrapolation per elements to find all the physical quantities at the interface. However, as each quantity is interpolated then extrapolated, additional interpolation must be avoided. Hence, the extrapolation of $\mathbf{D}$ on Σ_h is done component by component from their natural location.

11.3 Validation

11.3.1 Settling of a cylinder

We first study the spatial convergence of the method which is expected to be of second order. A cylindrical particle is dropped in a tank of dimensions $l \times L = [-0.005\,;\,0.005] \times [-0.02\,;\,0.005]$ in meters. The initial velocity is zero and its coordinates are $\mathbf{X}_i = (0, 0)$. The cylinder has a radius $r = 0.001m$ so the confinement $k = 2r/l = 0.2$. The Faxen theory (see [Happ 63]) gives the resistance force of the fluid on a cylindrical particle with respect to the terminal velocity U_∞:

$$F = \frac{4\pi\mu_f U_\infty}{\ln(1/k) - 0.9157 + 1.7244(k)^2 - 1.7302(k)^4} \tag{11.28}$$

whereas the buoyancy force is $F = (\rho_f - \rho_p)\pi 4r^2 g$. The physical parameters are chosen such as $Re = 1.85 \times 10^{-4}$. The Faxen theory is defined for $Re < 1$ and for a tunnel of infinite length. Practically, a no-slip boundary condition is imposed for all the walls. As proven by the good and regular obtained results, the presence of an upper and a lower wall does not influence the behavior of the particle.

The Fig. 11.2 shows that the convergence of the terminal velocity U_∞ reach a second order in space. As can be seen, the ITPM is more accurate for the coarsest meshes but does not converge

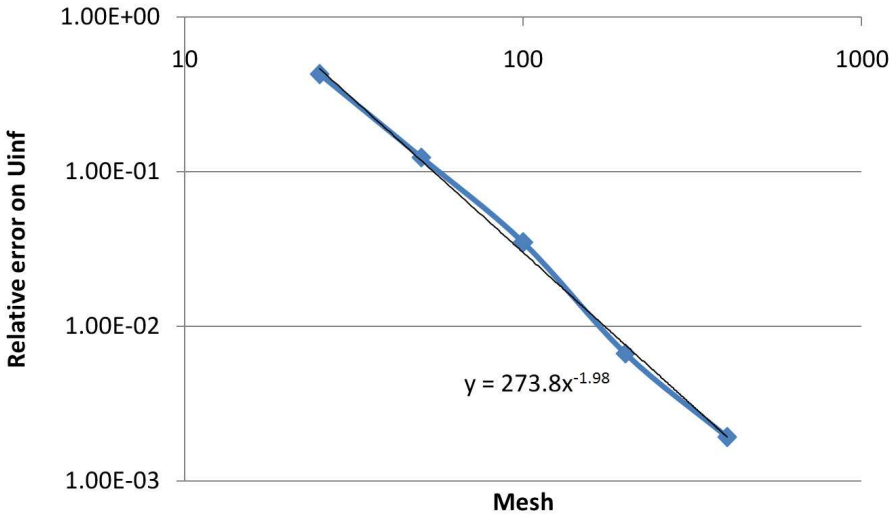


Figure 11.2: Relative error on the terminal velocity for the SMPM and the ITPM

to the desired solution (even if the difference is slight). In fact, the SMP methods can impose the penalty constraint to machine error accuracy. The ITPM penalizes the viscosity in the solid. Due to the numerical deterioration of the matrix conditioning, the ratio of magnitude between fluid and solid viscosities has to be limited. Even with a direct solver, the solid cannot have an infinite viscosity and the calculation cannot converge to the real value. The Fig. 11.3 shows the evolution of U_∞ through the time. For coarsest meshes, the displacement from a cell to an other produces a periodic noise on the terminal velocity which oscillates (for the convergence, the average value is considered). The Fig. 11.4 shows the velocity magnitude for the first four meshes at $t = 4s$. The velocity field near the particle is qualitatively quite similar for the four meshes. Globally, the position of the particle differs, especially for the two first meshes. By experience, we know that the force calculation for a given surface element is dependant to its

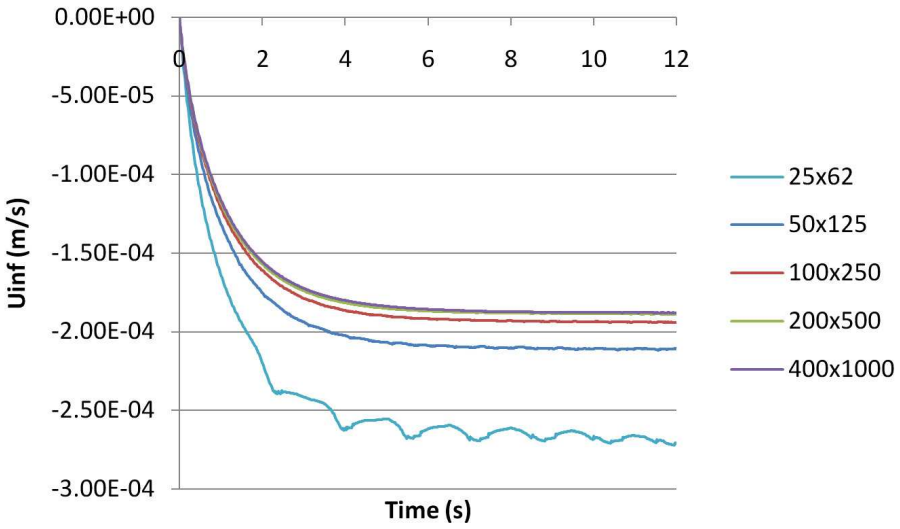


Figure 11.3: Time evolution of the terminal velocity of the particle for various meshes

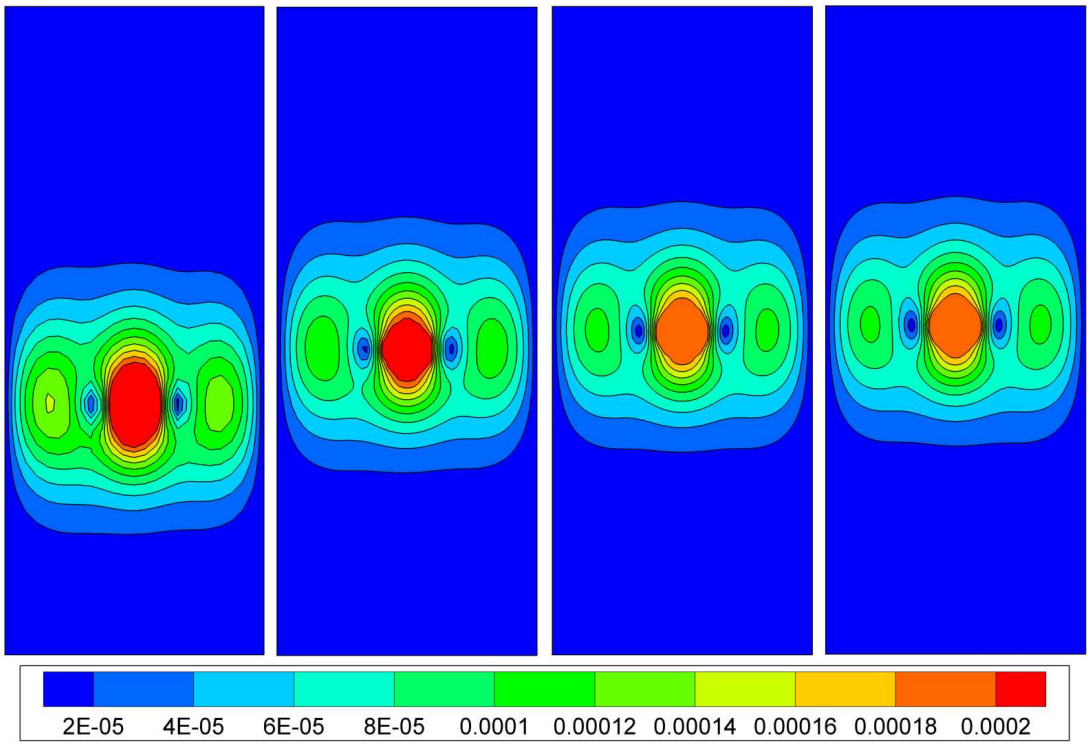


Figure 11.4: Velocity magnitude in $m.s^{-1}$ at $t = 2s$ for 25×62, 50×125, 100×250 and 200×500 meshes

relative position in a given grid cell. Fig. 11.5 shows the evolution of the settling velocity on a 25×62 mesh for SMPM and ITPM. As can be seen, the solution is very regular for the ITPM and closer to the analytical result.

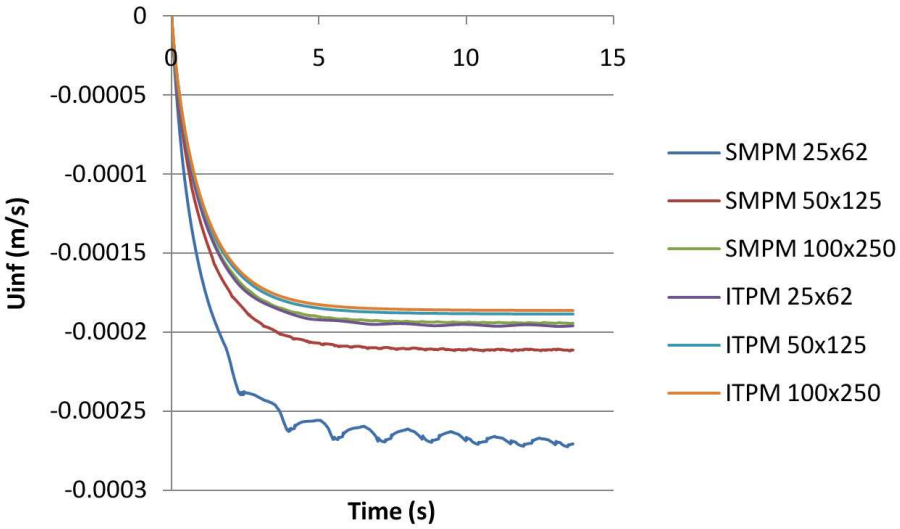


Figure 11.5: Comparison between the SMPM and the ITPM for the time evolution of the terminal velocity

11.3.2 Calculation of the wall force in a spherical particle

A convergence test for the calculation of the hydrodynamic force on a particle is performed. A sphere of radius $0.25m$ is immersed and centered in an unit box. The analytical solution of the unbounded flow for $Re = 0.01$ is imposed [Happ 63]. The Fig. 11.6 shows the convergence of the draft coefficient C_D. The global convergence order is 1.50. However, the order tends to 2 when the mesh step size decrease. One can see that the accuracy for the coarsest meshes is quite poor. However, the error goes under 10% quite quickly. Furthermore, a separate convergence for the viscous and the pressure forces would be desirable as the two quantities are not obtained in a same manner.

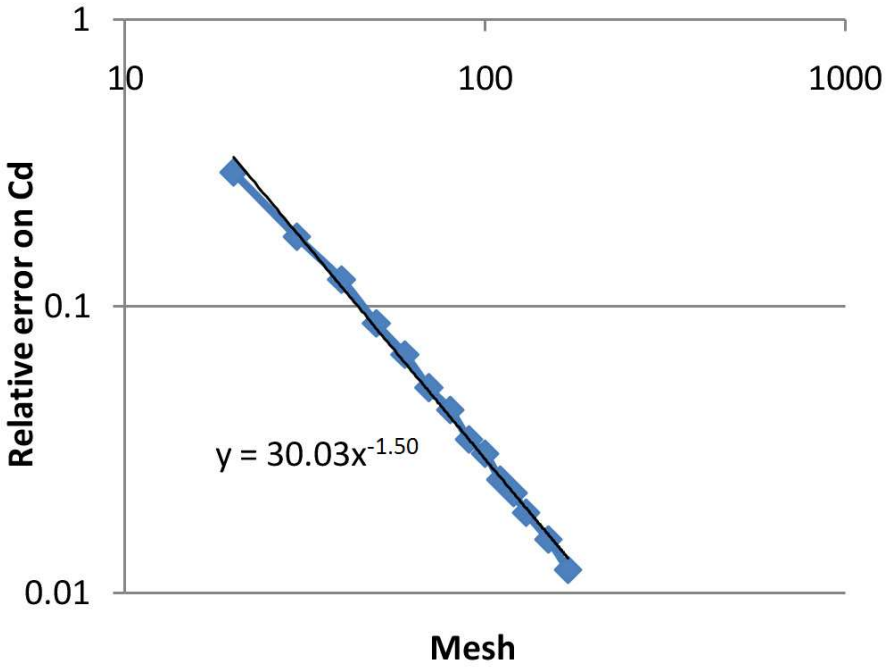


Figure 11.6: Convergence of the drag coefficient for a sphere at $Re = 0.02$

Discussion and conclusion of Part IV

A real-time code for the simulation of solid interaction, *Dresden*, has been created. The aim was to easily define a collision algorithm to implement it in our CFD code. *Dresden* provides promising results for the cases involving a large amount of objects. The algorithm has to be enhanced for the case of the static stacking. An additional model of viscous interactions for the case of fluid-solid coupling would be desirable [Lefe 07].

A fluid-structure coupling has been designed by combining the SMP method and a high-order computation of the hydrodynamical forces on objects. A second order of spatial accuracy has been reached. Concerning the time accuracy, a classical time-splitting approach coupled with a first Euler scheme has been considered. However, settling of the particle tends to a constant terminal velocity, so constant hydrodynamical forces, and a time scheme of higher order cannot enhanced a constant velocity. In [Vinc 07], the time splitting is avoided by using the ITP method (see section 2.3) so the solid velocity is implicitly computed simultaneously with the fluid velocity. The spatial accuracy is of first order only at the vicinity of the objects. The performances of both method has been compared on a simple case. The ITPM has clearly shown the best accuracy except for the finest meshes. In fact, the ITPM does not consider a perfectly rigid movement inside objects so the model does not converge to the physical solution. However, the difference between the physical solution and the converged numerical solution is very small. The SMPM method with its time-splitting is more accurate for the finest meshes on our simple case (the settling of a cylinder in a Stokes flow). However, such a ratio mesh-quality/complexity of flow is practically never reached in realistic cases.

Concerning the wall forces calculation, this high-order method suffers from many limitations, especially time oscillations of the calculated forces for the coarsest meshes. Its application to complex geometries is sometimes difficult, especially for convex meshes with high curvature. Some ideas to enhance this point have been proposed.

To conclude, the ITPM is clearly the most interesting method and the next step will be to couple our Eulerian-Lagrangian mesh projections with the ITPM so as to use it with complex meshes (only cylinders in 2D and spheres in 3D have been treated for now with the ITPM). The final step would be to increase the spatial accuracy of the ITPM thanks to the AIIB method. The Dirichlet BC imposed with the SMPM could be replaced by interface conditions to obtain an implicit resolution and an accurate imposition of the jump conditions. Furthermore, the ITP formulation implicitly transmit the forces from a media to an other and so does not require a complex calculation of the hydrodynamical forces on the objects (as explained before, this process suffers from robustness problems when the considered object is non-convex or is close to an other object).

Part VI

Industrial applications

Table of Contents

Introduction

The industrial application is one of the motivations of the present work. The step between the basic validation of a method on academic cases and its application to real case is sometimes huge. The fluid simulation often involves very high turbulent flows requiring dense calculation meshes. The immersed objects are sometimes composed of millions of elements. For this last point, it is critical to treat these Lagrangian meshes with fast algorithms and naive methods cannot be used practically. The ability of our methodology to treat complex cases is demonstrated through the following realistic cases. The first one, the flow inside a drilling bit, is a one-phase turbulent flow. The next case studies the hydroplaning of a tire. Two-phase turbulent flows are involved while a complex moving mesh is used for the tire. The last case details the simulation of the natural convection in the cave of Lascaux. In a last chapter, illustrative cases are presented. These simulations have been carried out by members of the laboratory TREFLE with Thétis and the present methodology.

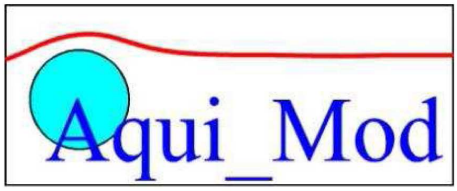
Chapter 12

Simulation of a drilling head

This work was a part of a project founded by the Aquitaine Region Council and leaded by Varel (Alfazazi Dourfaye). Varel manufactures drill bits for the global oil & gas drilling community as well as for the blasthole mining, industrial, construction and water well drilling communities. The other participants was the laboratory TREFLE (Arthur Sarthou, Stéphane Vincent and Jean-Paul Caltagirone) and Armines (Laurent Gerbau).

The aim of the project was to enhance the performances of the Varel drill bits used for the oil extraction. The part of the work related to this thesis was the simulation of the mud flows in the drill bits and the removal of the rock chips.

The final report is presented here. For the second part, where a parametric study is performed on five different drill bits, the figures with the considered drill bits have been removed for confidentiality reasons. The shape of these bits is about the same as for the three firsts presented in the first part.



Projet Varel – Aquilon

Rapport final

Présentation du projet

L'objectif de la partie TREFLE du projet est de simuler les écoulements de boues dans des têtes de forages, puis d'étudier l'évacuation de copeaux de roche. Ce projet nécessite l'utilisation d'un outil de mécanique des fluides basé sur des modélisations compressibles capables de gérer des obstacles de forme complexe (tête de forage) et leur mouvement. Nous avons décidé d'utiliser Aquilon (appelé maintenant Thétis depuis Novembre 2008), bibliothèque de calcul scientifique en mécanique des fluides et transferts développée dans l'équipe MFEN du TREFLE. Certains développements ont été nécessaires dans le code Aquilon pour mener à bien le projet :

- L'écoulement dans les têtes se fait à haute vitesse à des nombres de Mach pouvant atteindre 0.3. Ainsi, nous sommes partis d'un modèle de résolution incompressible existant dans Aquilon. Celui-ci doit cependant pouvoir prendre en compte les effets compressibles isothermes au travers de termes de dilatation.
- Aquilon utilise des maillages structurés fixes. Les têtes sont des objets complexes en mouvement qui nécessitent donc le développement d'outils et de méthodologies spécifiques de gestion de maillages pour pouvoir représenter les effets de la tête à chaque instant sur un maillage qui n'est pas adapté à la géométrie. Nous avons ainsi mis en œuvre des méthodes de domaines fictifs.
- Le suivi des copeaux dans l'écoulement à l'intérieur du forage près de la tête a été réalisé par une méthode lagrangienne. A partir du champ de vitesse eulérien on suit les copeaux dans leur mouvement grâce à une méthode mixte eulérienne/lagrangienne. Un module lagrangien a été développé pour traiter la génération et du suivi des copeaux

Hydrodynamique

Une boue aux caractéristiques proche de l'eau est injectée dans la tête par la zone rouge et ressort par la zone bleue (Figure 1). La tête de forage est en rotation à vitesse constante. Des copeaux de roches apparaissent au niveau des taillants et sont entraînés par le fluide. La vitesse d'injection est de l'ordre de la dizaine de m/s ce qui rend l'écoulement fortement turbulent dans certaines zones. Les Reynolds sont ainsi de l'ordre de la dizaine de milliers tandis que le nombre de Mach peut atteindre 0.3. Dans les zones de resserrement la vitesse peut être de l'ordre de la centaine de m/s. L'écoulement reste laminaire au niveau de l'injection avant d'être turbulent au niveau des taillants. L'intensité turbulente décroît généralement au niveau des canaux latéraux d'évacuation.

Figure 1 : Tête de forage – Zones d'injection et d'évacuation

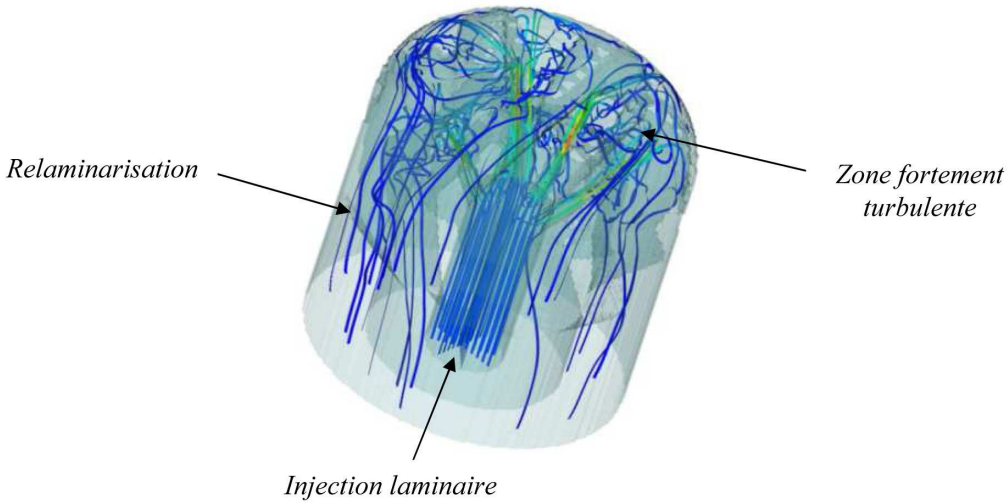


Figure 2 : Caractérisation de l'écoulement

Advection de copeaux

Modèles numériques

La bibliothèque de calcul scientifique Aquilon

L'outil Aquilon est un simulateur volumes finis sur maillages structurés. Ses caractéristiques principales sont les suivantes :

- Un schéma d'Euler du premier et du second ordre est utilisé pour les dérivées en temps et un schéma centré du second ordre pour les dérivées en espace

- La résolution du système linéaire est effectuée par divers solveurs :
  - Solveur itératif : Bi-Conjugate Gradient Stabilised (BiCGSTAB) [VAN 92] pour la résolution de la matrice preconditionnée avec une méthode LU incomplète modifiée
  - Solveurs directs : PARDISO, MUMPS
  - Solveur parallèle : HYPRE

- Le suivi d'interface est effectué principalement par une méthode VOF Piecewise Linear Interface Construction (PLIC) [SCA 99] pour la résolution de l'advection de fractions volumiques et implique une faible diffusion numérique
- La résolution du couplage vitesse-pression et de la contrainte d'incompressibilité des équations de Navier-Stokes est obtenue par une méthode de Lagrangien Augmenté [VIN 04]. La discrétisation des équations devient :

$$\rho\left(\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t}+\nabla\cdot\left(\mathbf{u}^{n}\otimes\mathbf{u}^{n+1}\right)\right)+b(\mathbf{u}-\mathbf{u}_{D})=-\nabla p^{n}+\nabla\cdot\left(\mu\ \left[\nabla\mathbf{u}^{n+1}+(\nabla\ \mathbf{u}^{n+1})^{T}\right]\right)+\nabla\left(r\nabla\cdot\mathbf{u}^{n+1}\right)$$

$$p^{n+1}=p^{n}-r\nabla\cdot\mathbf{u}^{n+1}$$

où $\mathbf{u}$ est la vitesse, t le temps, p la pression, ρ la masse volumique, μ la viscosité dynamique et n l'indice de l'itération temporelle correspondant au temps n Δt.

Le terme $\nabla\left(r\nabla\cdot\mathbf{u}^{n+1}\right)$ est un terme de pénalisation associé à un lagrangien (la pression) qui force l'incompressibilité de l'écoulement. La seconde équation permet d'accumuler une contrainte de divergence dans la pression par la méthode d'Uzawa. L'utilisation du Lagrangien Augmenté permet de s'affranchir d'une étape de correction de pression et rend ainsi totalement implicite la résolution fluide au cours d'un pas de temps. Son principal défaut est de rendre la matrice plus difficile à inverser. Un bon choix du paramètre r est crucial. Plus r est grand et plus l'incompressibilité est assurée, mais plus les autres termes de l'équation deviennent petits et donc leurs effets risquent de disparaître au cours des itérations si un solveur itératif est utilisé.

Le suivi lagrangien d'espèces est effectué avec une méthode VOF Sous Maille (VOF-SM) qui couple des aspects lagrangiens et eulériens et assure un très bon transport des espèces. [BAL 07]

La gestion des objets

Diverses méthodes de domaines fictifs dites de pénalisation permettent de prendre efficacement en compte les objets. La méthode VPM (Volumic Penalty Method) consiste à ajouter un terme $b(u-u_D)$ aux équations de Navier-Stokes qui deviennent :

$$\rho\left(\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t}+\nabla\cdot\left(\mathbf{u}^{n}\otimes\mathbf{u}^{n+1}\right)\right)+b(\mathbf{u}^{n+1}-\mathbf{u}_D)=-\nabla p^{n}+\nabla\cdot\left(\mu\left[\nabla\mathbf{u}^{n+1}+(\nabla\,\mathbf{u}^{n+1})^{T}\right]\right)+\nabla\left(r\nabla\cdot\mathbf{u}^{n+1}\right)$$

On pose $b=0$ dans le milieu fluide et $b=10^{40}$ dans le milieu solide. Ainsi, dans le solide, l'équation de conservation de quantité de mouvement devient simplement :

$$\mathbf{u}^{n+1}=\mathbf{u}_D$$

La vitesse est imposée à l'ensemble de la maille même si celle-ci n'appartient que partiellement au domaine solide. De par cette approximation, la méthode VPM est d'une précision en espace du premier ordre seulement. La méthode de pénalisation de sous-maille, ou SMPM [SAR 08], améliore la VPM en proposant une précision du second ordre. Toutefois, l'impact de la complexité de l'écoulement dans le cas qui nous intéresse sur la SMPM n'a pas encore été étudié. Nous utiliserons donc uniquement la méthode VPM pour le moment, sa relative imprécision étant compensée par la forte densité du maillage utilisé.

La méthode VPM nécessite une localisation des mailles appartenant au milieu solide. Elle est ainsi couplée à une méthode de Ray-Casting qui permet de projeter sur une grille eulérienne un objet défini par une surface lagrangienne, c'est-à-dire un maillage triangularisé de la peau de la tête de forage. La méthode de Ray-Casting nécessite le calcul de nombreuses intersections rayon/triangle et peut donc demander énormément de temps de calcul dans sa version naïve. Une méthode « colonne par colonne » a été développée afin d'accélérer le processus.

Modélisation

La rotation différentielle de la tête par rapport à la paroi pose localement des problèmes de divergence discrète. La vitesse de rotation étant petite par rapport à la vitesse de l'écoulement, on négligera la rotation différentielle tête/paroi. On utilisera les équations de Navier-Stokes en repère mobile pour prendre en compte une rotation globale du domaine de calcul à la vitesse de rotation de la tête.

La condition d'entrée est de type Dirichlet constante, la condition de sortie impose un gradient de vitesse normal nul et une vitesse tangentielle nulle.

Gestion des copeaux

Les copeaux sont de petits morceaux de roche de taille millimétrique. On fait les hypothèses réductrices suivantes :

- Les copeaux n'influencent pas l'écoulement et on travaille donc avec un champ de vitesse stationnaire pré-calculé
- Les copeaux sont advectés passivement sans prise en compte de leur inertie
- Les copeaux n'interagissent pas entre eux. Rien ne limite le nombre de copeaux dans une zone.
- Tous les copeaux sont de taille et de masse identiques

Ainsi, il n'est pas possible de modéliser le phénomène en temps long de « cake » engendré par une agglomération de copeaux. Le modèle défini ici permet toutefois d'étudier la dynamique instantanée des copeaux et de cibler les potentielles zones mortes ou d'accumulation. Il permet aussi d'utiliser un unique champ de vitesse calculé au préalable.

Les copeaux sont représentés par des particules auxquelles un volume égal à celui de la maille de calcul est associé. A chaque pas de temps, les particules sont advectées puis la quantité qu'elles transportent est projetée sur le maillage eulérien pour obtenir la concentration eulérienne. Cette projection facilite la visualisation des zones de forte concentration et offre un post traitement plus aisé.

Zone de génération des copeaux

La génération des copeaux est faite dans une zone située à proximité des taillants. Cette zone doit être définie explicitement à l'aide d'un maillage lagrangien surfacique triangularisé. La définition de cette zone comme étant un ensemble de tubes à proximité des taillants (Figure 3) a été initialement utilisée puis abandonnée car la définition des tubes était complexe. La méthode nouvellement retenue consiste à extruder la surface des taillants afin de générer un volume d'injection. (Figure 4)

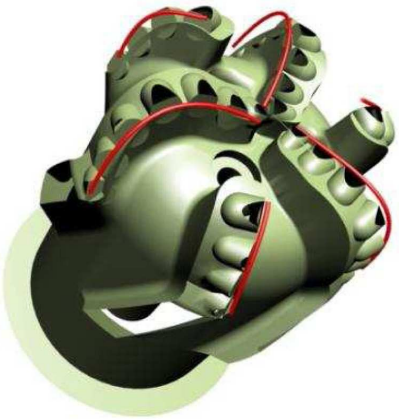


Figure 3 : Zone de génération de copeaux créée avec l'ancienne méthode

Figure 4 : Nouvelle méthode de définition de la zone de génération de copeaux

L'injection est effectuée en initialisant des particules dans les zones d'injection. Ces injections se font par salves. Les copeaux étant advectés passivement, le fait d'injecter en continu ou par salves n'a pas d'influence sur les résultats.

Résultats

Premiers tests hydrodynamiques

Les premiers tests sur l'hydrodynamique ont été menés sur 3 têtes de diamètre 6', 8.12', 12.14'.

Configuration tête 6'

Caractéristiques physiques du cas :

$\Omega = 12.56 - 20.9 rad.s^{-1}$

$\mu = 0.025 kg.m^{-1}.s^{-1}$

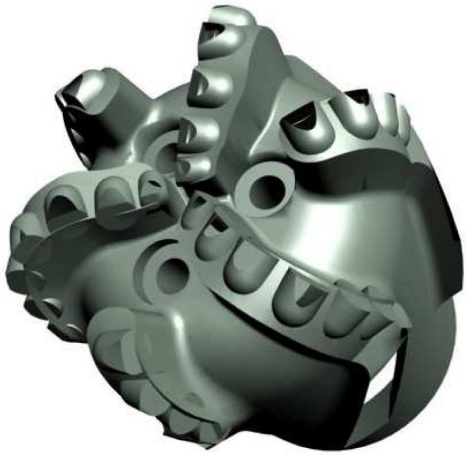
$\rho = 1300 kg.m^{-3}$

$d = 2000 l.\min^{-1} = 3.33.10^{-2} m^3.s^{-1}$

$V_{inj} = 29.4 m.s^{-1}$

$Ro_{inj} = V_{inj} / \Omega L_{inj} \approx 100$

$\mathrm{Re}_{inj} = 60000$



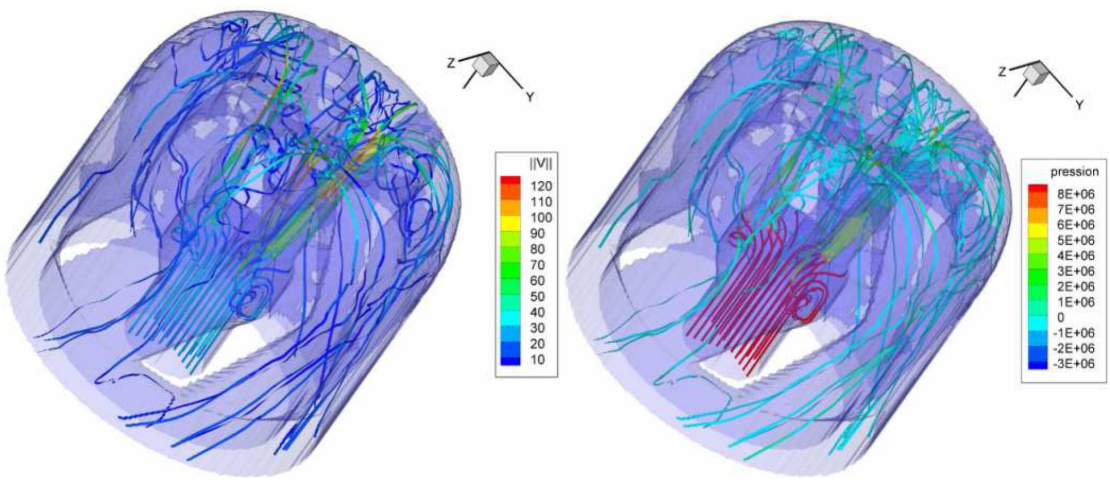


Figure 5 : Lignes de courant norme de vitesse et pression pour la tête 6'

Configuration tête 8.12'

Caractéristiques physiques du cas :

$\Omega = 12.56 - 20.9 rad.s^{-1}$

$\mu = 0.025 kg.m^{-1}.s^{-1}$

$\rho = 1250 kg.m^{-3}$

$d = 2800 l.\min^{-1} = 4.6.10^{-2} m^3.s^{-1}$

$V_{inj} = 18.9 m.s^{-1}$

$Ro_{inj} = V_{inj} / \Omega L_{inj} \approx 32.3$

$\mathrm{Re}_{inj} = 26400$

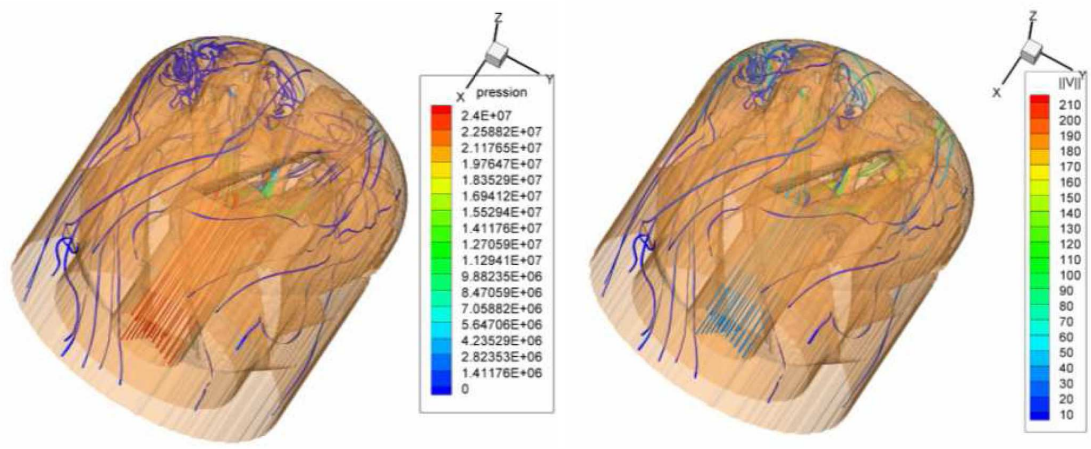


Figure 6 : Lignes de courant, norme de vitesse et pression pour la tête 8'12

Configuration tête 12.25'

Caractéristiques physiques du cas :

$\Omega = 8.37 - 20.9 rad.s^{-1}$

$\mu = 0.026 kg.m^{-1}.s^{-1}$

$\rho = 1150 kg.m^{-3}$

$d = 3500 l.\min^{-1} = 5.83.10^{-2} m^3.s^{-1}$

$V_{inj} = 13.5 m.s^{-1}$

$Ro_{inj} = V_{inj} / \Omega L_{inj} \approx 10$

$\mathrm{Re}_{inj} = 22000$

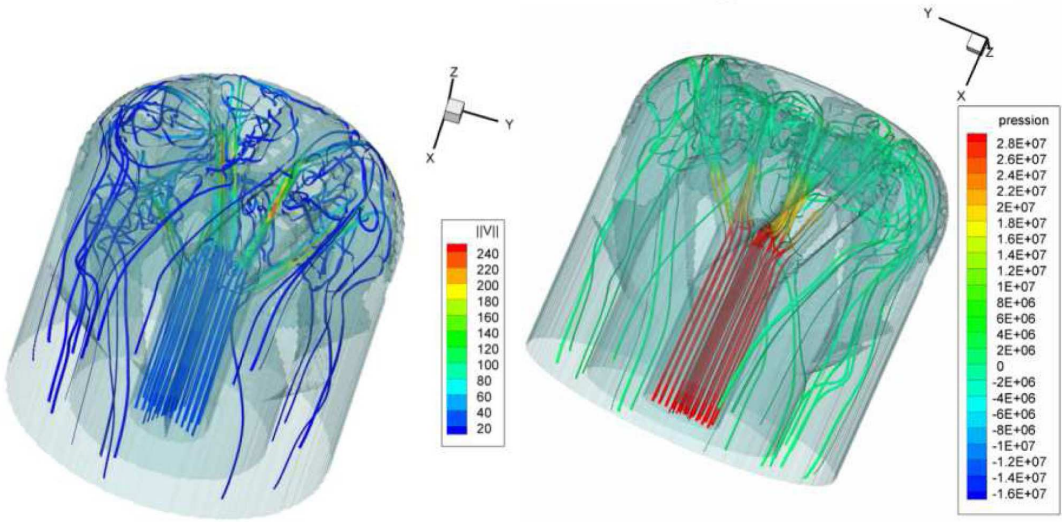


Figure 7 : Lignes de courant, norme de vitesse et pression pour la tête 12.25'

Ecoulement

Quelles que soient les têtes, on remarque plusieurs caractéristiques communes. L'injection se fait dans une première zone cylindrique dans laquelle l'écoulement reste laminaire. On observe dans certains cas une recirculation quand le rayon du tube augmente à une de ses extrémités (têtes 6' et 8.12'). L'écoulement se divise ensuite en trois ou six tubes de plus faible section ce qui induit une forte perte de charge et une augmentation de la vitesse. C'est dans cette zone que l'écoulement est le plus rapide. On observe une surpression sur la paroi supérieure du domaine là où impactent les jets sortants de ces tubes (Figure 8). La présence de ces jets à très forte vitesse dans la partie supérieure de la tête dont la forme est très irrégulière rend l'écoulement très turbulent. Les nombreuses recirculations peuvent générer des zones mortes qui perturbent la bonne évacuation des copeaux. Le fluide s'échappe ensuite le long des parois latérales de la tête. La régularité de cette zone engendre une relaminarisation de l'écoulement.

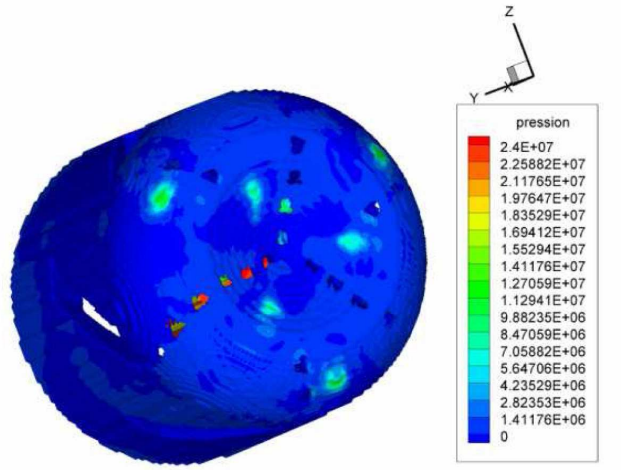


Figure 8 : Pression à la surface du domaine. Mis en évidence des surpressions dues aux jets impactants

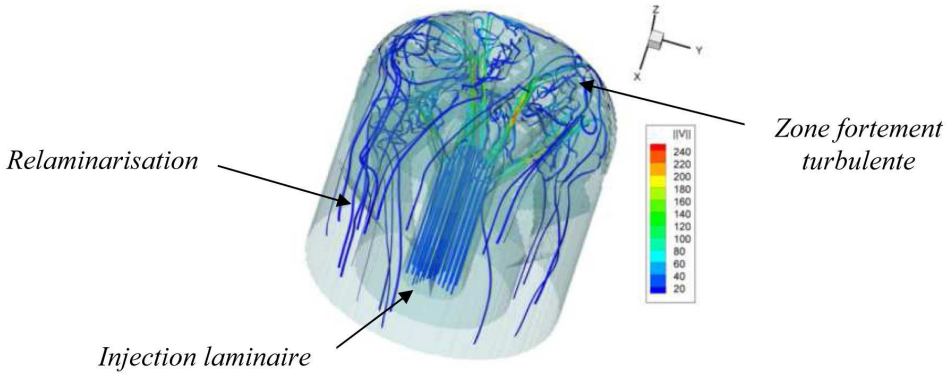


Figure 8 : Caractérisation de l'écoulement

Etude paramétrique de cinq têtes

Afin de valider notre modélisation, nous effectuons maintenant une étude complète comparée de cinq têtes différentes. Les comparaisons concernent l'hydrodynamique dans un premier temps, puis l'évacuation des copeaux. Les paramètres physiques sont très similaires à ceux des études précédentes :

$\Omega = 12.56 rad.s^{-1}$

$\mu = 0.026 kg.m^{-1}.s^{-1}$

$\rho = 1150 kg.m^{-3}$

$d = 2.8 l.\min^{-1}$

$V_{inj} = 18.947 m.s^{-1}$

$Ro_{inj} = V_{inj} / \Omega L_{inj} \approx 50$

$\mathrm{Re}_{inj} = 47000$

Le taux d'injection des copeaux est de 7%.

Présentation des têtes

Les paramètres d'injection, de rotation ou de rhéologie sont les mêmes pour les cinq têtes. La seule différence réside dans l'angle d'incidence des lames (Figures 9 et 10).

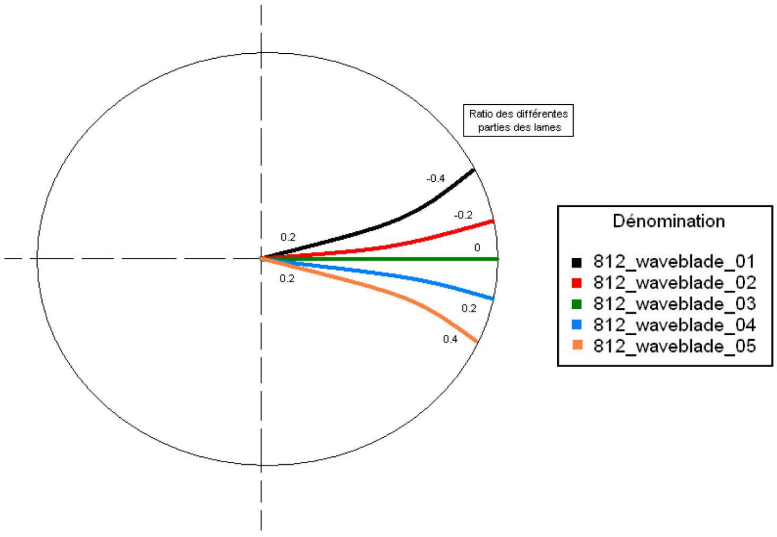


Figure 9 : Schéma des angles d'incidence des lames pour les cinq têtes

Hydrodynamique

Les conditions de simulation sont similaires aux cas précédent de tête 8'. Les simulations sont effectuées sur deux maillages 60×60×70 (dit maillage 60) et 120×120×140 (dit maillage 120) avec des pas temps respectivement de 2×10^{-5}s et 10^{-5}s.

La dynamique globale est la même pour les cinq têtes et reste très similaire à celle observée avec le premier lot de trois têtes. Le tableau 1 montre les moyennes spatiales de la valeur absolue de la vitesse ainsi que de la vorticité pour les cinq têtes :

Tête	1	2	3	4	5
Vitesse (m/s)	10.0	9.95	9.98	9.72	9.91
Vorticité (1/s)	232	227	229	223	230

Tableau 1 : Moyenne spatiale de vitesse et de vorticité pour le maillage 120 et t=0.017s

Les différences entre les têtes sont minimes, que ce soit pour la vitesse ou la vorticité. Le tracé des iso-surfaces de vorticité montre que la vorticité maximale est toujours située dans les canaux d'injections et se propage jusqu'aux zones d'impact des jets. On peut relever que la différence de vorticité au niveau des zones d'impact entre les injecteurs primaires et secondaires est plus grande pour les têtes 1,4 et 5.

Convergence

Nous étudions la convergence de la simulation en temps et en espace en utilisant les critères de vitesse moyenne et de vorticité moyenne pour la tête 1. Les figures 10 et 11 montrent la convergence au cours du temps de ces critères pour ces deux maillages.

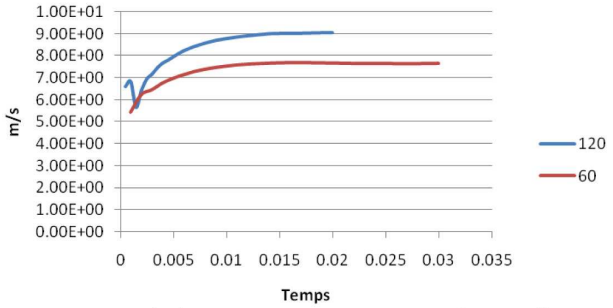


Figure 10 : Convergence de la vitesse moyenne pour des maillages 60×60×70 et 120×120×140 pour la tête 1

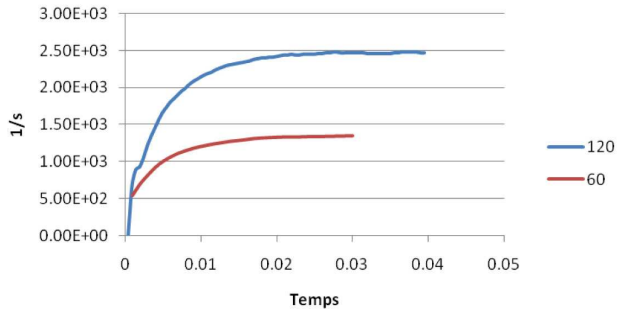


Figure 11 : Convergence de la vorticité moyenne pour des maillages 60×60×70 et 120×120×140 pour la tête 1

La convergence de la vitesse est atteinte à t=0.015s pour le maillage 120 et à t=0.012s environ pour le maillage 60. Le temps de convergence pour la vorticité est plus grand, de l'ordre de t=0.027s pour le maillage 120 et de t=0.017s pour le maillage 60. Concernant la magnitude

des données physiques, on remarque d'assez fortes différences entre les deux maillages. Alors que la vitesse moyenne est plus élevée de 15% environ sur le maillage 120, la vorticité est presque deux fois plus grande sur le maillage 120 que sur le maillage 60. Les cas simulés sont à très haut Reynolds donc les plus petites échelles de la turbulence ne sont pas directement simulées quelque soit le maillage (un modèle particulier de turbulence prend toutefois le relais et permet de capter dans une certaine mesure le reste de la turbulence. Il s'agit d'une modélisation des grande échelles, SGE ou LES en anglais, qui se base sur une séparation d'échelles entre les grandes structures turbulentes qui sont résolues et les petites qui sont modélisées). Le maillage 120 capte donc des échelles de turbulence que le maillage 60 ne capte pas ce qui réduit d'autant la vorticité sur ce dernier. Le but ici est de simuler l'écoulement de copeaux et c'est justement cette turbulence qui provoque des zones mortes et pénalise l'évacuation des copeaux. On ne peut donc pas se satisfaire d'un point de vue physique d'un maillage trop grossier captant correctement la dynamique générale mais ne produisant pas une vorticité suffisante. Malheureusement, la convergence de l'écoulement nécessite pour le maillage 120 près de deux semaines de calcul en mono processeur ce qui est trop long à l'échelle des temps d'étude mis en place chez Varel Europe. Idéalement le temps de calcul ne devrait pas dépasser 10 heures.

Copeaux

Une simulation de génération et d'évacuation de copeaux est menée sur les 5 têtes à partir du champ de vitesse calculé précédemment. Ce champ étant stationnaire et les particules n'interagissant pas entre elles, il n'est pas nécessaire de simuler les mouvements de copeaux sur un temps long. Toutefois, la position des particules ajoutées dans les zones de création est aléatoire, et l'augmentation du temps de simulation permet d'obtenir des moyennes statistiques mieux convergées

Les quantités obtenues en sortie de calcul sont la vitesse et la position de chaque particule ainsi que son âge. Cette dernière donnée permet de repérer les zones où les particules sont piégées. Le tableau 2 montre le volume total de particules présentes dans les têtes au bout de 0.5s de calcul.

Tête	1	2	3	4	5
Volume (m^3)	1.40×10^{-4}	1.22×10^{-4}	1.21×10^{-4}	1.41×10^{-4}	1.77×10^{-4}

Tableau 2 : Volume de copeaux présent par têtes pour t=0.5s

Plus le volume est important et moins l'évacuation est bonne. La différence de volume de copeaux est au maximum de 30% entre la tête 3 et la 5. On constate aussi que l'évacuation diminue avec l'angle des lames.

La figure 15 montre la répartition des particules et leur âge pour la tête 3 au bout de 0.5s. Dans les deux cas, on remarque la présence de zones mortes au niveau de l'évacuation ainsi que dans la partie supérieure périphérique de la tête. Cette vue ne permet pas de discerner les qualités d'évacuation des têtes et sert seulement à repérer les zones mortes.

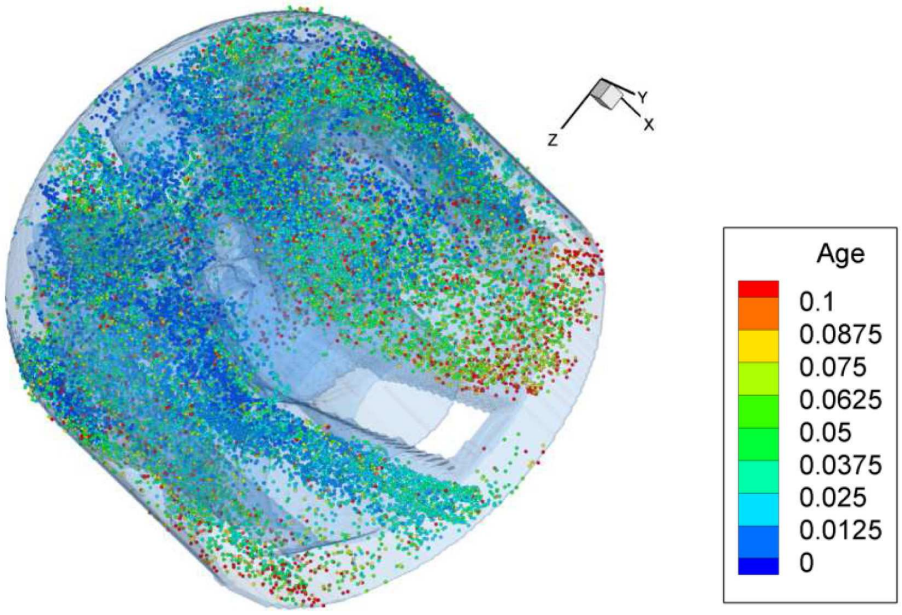


Figure 12 : Position et âge des particules au bout de 0.5s pour la tête 3

Conclusion de la partie simulation

L'objectif était d'obtenir un simulateur d'hydrodynamique dans les têtes de forage avec une prise en compte de la dynamique et de l'évacuation des copeaux de roche.

La partie simulation hydrodynamique s'est appuyée sur le simulateur Aquilon/Thétis qui est depuis longtemps utilisé pour traiter des cas industriels. Le projet a tout de même nécessité le développement d'outils de gestion d'objets complexes qui sont maintenant intégrés à la version standard du code de calcul. Les résultats des simulations hydrodynamiques obtenues sont ainsi totalement satisfaisants. Seule la durée des simulations en calcul mono-processeur semble encore destiner l'outil à une utilisation non R&D.

Pour ce qui est de la dynamique des copeaux, la marge d'amélioration du modèle est assez grande pour les phénomènes émergeants aux temps longs, comme les effets d'agglomération. En l'état, le simulateur donne toutefois une bonne évaluation des qualités d'évacuation en temps court des têtes et permet en particulier de cibler les zones mortes.

Une nouvelle piste d'amélioration serait le développement d'un modèle d'usure de la tête. On peut en effet calculer l'effort fluide sur la tête. Ce dernier peut se traduire par une érosion de la tête au cours du temps et donc une modification de la dynamique de l'écoulement. Cette partie demande toutefois des développements supplémentaires assez techniques.

Bibliographie

[BAL 07] G. Balmigere, S. Vincent, J.-P. Caltagirone, E. Meillot, 2007
Utilisation d'une méthode de suivi d'interface mixte eulérienne/lagrangienne pour les écoulements diphasiques.
Comptes rendus du Congrès Français de Mécanique.

[SAR 08] A. Sarthou, S.Vincent, J.-P. Caltagirone, P. Angot, 2008
Eulerian-Lagrangian grid coupling and penalty methods for the simulation of multiphase flows interacting with complex objects.
International Journal of Numerical Methods in Fluids, 56 :8:1093-1099.

[SCA 99] R. Scardovelli, S. Zaleski, 1999
Direct numerical simulation of free surface and interfacial flow.
Annual Review of Fluid Mechanics 31, 567-603.

[VAN 92] H.A. van der Vorst, BiCGSTAB, 1992
A fast and smoothly converging variant of BiCG for the solution of non-symmetric linear systems.
SIAM Journal on Scientific and Statistical Computing 13.

[VIN 00] S. Vincent, J.-P. Caltagirone, 2000
A One-Cell Local Multigrid method for solving unsteady incompressible multiphase flows.
Journal of Computational Physics 163:172-215.

[VIN 04] S. Vincent, J.-P. Caltagirone, 2004
An adaptative augmented Lagrangian method for three-dimensional multimaterial flows.
Computers and Fluids 33:1273-1289.

Chapter 13

Aquaplaning of a tire

13.1 Introduction

The hydroplaning is a phenomenon resulting from the lost of contact between a tire and the road when a vehicle is moving at a certain speed on a wet road. For a given velocity, the interaction between the water laying on the road and the tire generates a water reserve in front of the tire which is larger than the initial water depth. A resulting pressure is generated at the tire surface around the contact area between the tire and the water reserve. When the vertical effort generated by this pressure becomes superior to the weight of the vehicle, the contact between this vehicle and the road is no more maintained and the hydroplaning occurs: the adherence between the vehicle and the road is lost and the trajectory of the vehicle is no more controled. From the literature, it is well known that the link between the vehicle velocity and the hydroplaning pressure follows at first order [Tunn 06]

$$P_h = KV_v^2 \tag{13.1}$$

where P_h is the hydroplaning pressure, V_v is the velocity of the vehicle and K is a constant equal to half the density of water. This law is obtained under Bernouilli's assumptions of perfect fluid behavior. From the tire manufacturer experience, it is well known that the more efficient way to increase P_h for given wetting conditions of roads is to incorporate specific structures at the tire surface. In this way, more incoming water is evacuated laterally, the water reserve generated in front the tire is reduced for a given velocity and the hydroplaning effect appears for higher vehicle velocities. This effect of the tire structure has been studied for example by Masataka and Toshihiko [Masa] .

No existing experimental or theoretical studies are able to predict qualitatively the improvement brought by the choice of a tire structure on the hydroplaning compared to a reference flat tire whereas building a tire is very expansive. The aim of the present article is to propose a numerical modeling dedicated to the prediction of hydroplaning effects and to the classification of tire structures. This objective requires to account for three-dimensional turbulent free surface air-water flows interacting with complex tire geometries.

Among the rare existing literature works in the field of the numerical simulation of hydroplaning, two studies are of interest. The first one concerns the three-dimension simulation of the interaction between a free surface flow and a tire [Akse 96] . The second interesting work [Cho 06]

The numerical simulation of unsteady and incompressible isothermal multi-phase flows involving macroscopic interfaces is classically achieved thanks to the single-fluid Navier-Stokes equations [Kata 86, Scar 99b] and to Eulerian interface tracking methods such as the Volume Of Fluid (VOF) method [Youn 82] , the Level-Set technique [Oshe 01] or the Front Tracking approaches [Shin 02a] . These methods have been extensively compared and evaluated in the last ten years and have demonstrated their qualities and drawbacks [Ride 95] . Once an interface tracking method has been chosen, the major difficulty consists in solving the motion equations for high density or viscosity ratios and large interface distortions. Near the interface, parasitic currents or unphysical flow behavior occur when using, for example, time splitting projection methods for simulating air-water or particulate unsteady flows. In these problems, the resolution of the coupling between the incompressibility constraint and the Navier-Stokes equations is not ensured in one of the phases due to the ill conditionning of the linear system or to the boundary condition treatment. Consequently, this gives a wrong flow solution.

13.2 Numerical modeling of two-phase flows interacting with obstacles of complex shape

13.2.1 The 1-fluid model

The modeling of incompressible two-phase flows involving separated phases can be achieved by convolving the incompressible Navier-Stokes equations with a phase function C. As explained by Kataoka [Kata 86] , the resulting model takes implicitly into account the jumps relations at the interface [Delh 74, Scar 99b] and the interface evolutions are described by an advection equation on function C:

$$\nabla \cdot \mathbf{u} = 0 \tag{13.2}$$

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}) = -\nabla p + \rho \mathbf{g} + \nabla \cdot ((\mu + \mu_t)(\nabla \mathbf{u} + \nabla^t \mathbf{u})) + \mathbf{F}_{st} \tag{13.3}$$

$$\frac{\partial C_i}{\partial t} + \mathbf{u} \cdot \nabla C_i = 0 \tag{13.4}$$

where $\mathbf{u}$ is the velocity, p the pressure, t the time, $\mathbf{g}$ the gravity vector, ρ and μ respectively the density and the viscosity of the equivalent fluid. A mixed scale turbulence model is added through the turbulent viscosity μ_t [Saga 98, Laro 08]. The surface tension forces are taken into account thanks to a volume force $F_{st} = \sigma \kappa \mathbf{n}_i \delta_i$. The surface tension coefficient σ is assumed constant. The local curvature of the interface is κ whereas the normal to the interface is $\mathbf{n}_i$ and δ_i is a Dirac function indicating interface.

The 1-fluid model is almost identical to the classical incompressible Navier-Stokes equations, except that the local properties of the equivalent fluid (ρ and μ) depends on C, that the interface location requires the solving of an additional equation and that a specific volume force is added at the interface to account of capillary effects.

13.2.2 Discretization and solvers

The 1-fluid Navier-Stokes equations are discretized with finite volumes on an irregular staggered Cartesian grid. The coupling between velocity and pressure is ensured with an implicit algebraic adaptive augmented Lagrangian 3AL method (see section A.3.4.2). The augmented Lagrangian

methods presented in this work are independent on the chosen discretization and could be implemented for example in a finite element framework [Bert 97]. The linear system is inverted with a BiCGSTAB II solver [Gust 78b], preconditionned under a Modified and incomplete LU method [Vors 92].

Concerning the interface tracking, a Volume Of Fluid (VOF) approach is used with a Piecewise Linear Interface construction (PLIC) [Youn 82]. This approach ensures the mass conservation while maintaining the interface width on one grid cell.

The numerical methods and the 1-fluid model have been widely validated by the authors concerning jet flows [Vinc 99, Laro 09], capillary flows [Vinc 00, Leba 04, Tron 08] , wave breaking [Lubi 06, Vinc 04, Vinc 07], material processes [Laca 06], plasma to water jet interaction [Vinc 09] and more generally turbulent two-phase flows [Labo 07, Vinc 08].

13.3 Three-dimensional simulation of hydroplaning flows

13.3.1 Description of the problem

The three-dimensional air-water flow interacting with a tire is considered for a road and tire rotation velocity of $50 km.h^{-1}$. The tire is a fictitious domain of imposed velocity which is accounted for into the calculation grid by means of penalty terms imposing the velocity in all the tire zone. As presented in figure 13.1, three tire geometries, called tires $T1$, $T2$ and $T3$, are considered, in order to evaluate the effect of the tire structures on the flow-structure interaction and resulting forces exerted on on tire during hydroplaning. The tires are shown as they are really projected onto the simulation grid. It is observed that only their bottom part is considered in the simulation. On the road, the tire is in contact with the bottom boundary of the simulation domain (empty parts of the tire surface in figure 13.1). It can be pointed that the tire structured is not accurately described in the upper part of the calculation domain, due to grid coarsening in these zones. However, the two-phase flow does not provide important features in these part of the simulation, so their influence on the hydroplaning motion can be solved in a coarse manner. At each time step, the movement of the tire structures, as well as the deformation of the tire, are calculated thanks to a home-made software by MICHELIN, which provides us the triangular surface elements defining the tire topology. The effect of the flow on the tire deformation are not currently taken into account. The tire deformations are only due to the force exerted by the vehicle.
The fluid characteristics are $1000 kg.m^{-3}$ and $1.1768 kg.m^{-3}$ for the densities in water and air and $10^{-3} Pa.s$ and $1.8510^{-5} Pa.s$ for the dynamic viscosities in water and air respectively. The surface tension coefficient σ between water and air is assumed to be equal to $0.075 N.m^{-1}$. Initially, a water layer lays upward onto the road with a height of $8mm$ on the total width of the road.

The simulation grid is exponential, with a refined area in the zone where the tire is in contact with a road. The total number of cells in each directions is 270 x 110 x 80. The size of the cells in the refined zone is $1mm$ in each direction, while the macroscopic dimensions of the simulation domain are $1.4m$ x $0.291m$ x $0.6m$. The grid structure in vertical and horizontal slice views is presented in figure 13.2. All the simulations are computed on the same grid, which provides the required cell density to obtain results independent on the numerical parameters. The calculation time step is chosen constant and equal to $10^{-4} s$.

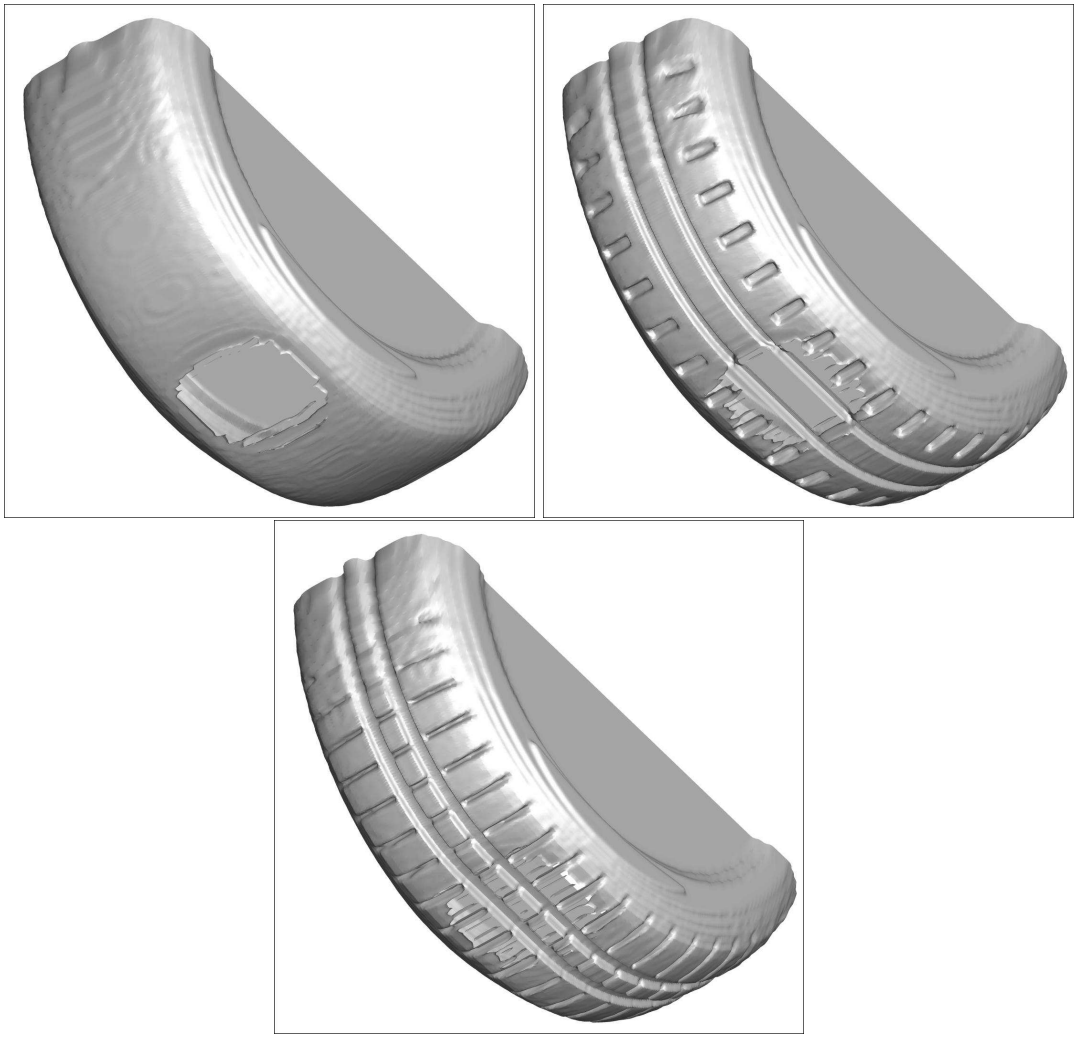


Figure 13.1: Topology of the three considered tires called $T1$, $T2$ and $T3$ (from left to right and top to bottom) - the tire structures are provided by MICHELIN.

13.3.2 Study of three-dimensional flows

The three-dimensional two-phase unsteady flow occurring when a water-air free surface hits and interacts with a tire has never been studied numerically with a full unsteady description of the two-phase motion and the corresponding efforts exerted on the tire. The typical flow structures that are observed when the tire geometry changes are presented in figure 13.3. The free-surface is represented by iso-surface $C = 0.5$. The simulations are considered after the flow has reached a stabilized state, *i.e.* the global shape of the free surface does not evolve during time. Whatever the type of tire topology, the macroscopic flow structure is almost the same. A V-like free surface form develops downstream the tire, a lot of water droplets are generated in the vicinity of the rotating obstacle and a pressure peak is created on the forward face of the tire near the road. The pressure projected onto the tire surface is described in figure 13.4. The flow is observed with a bottom view perpendicular to the road. The simulated values of the maximum pressure on the tire are in the range 80000 to $100000Pa$. These values are in good agreement with the experimental measurements of MICHELIN and the theoretical predictions provided by equation

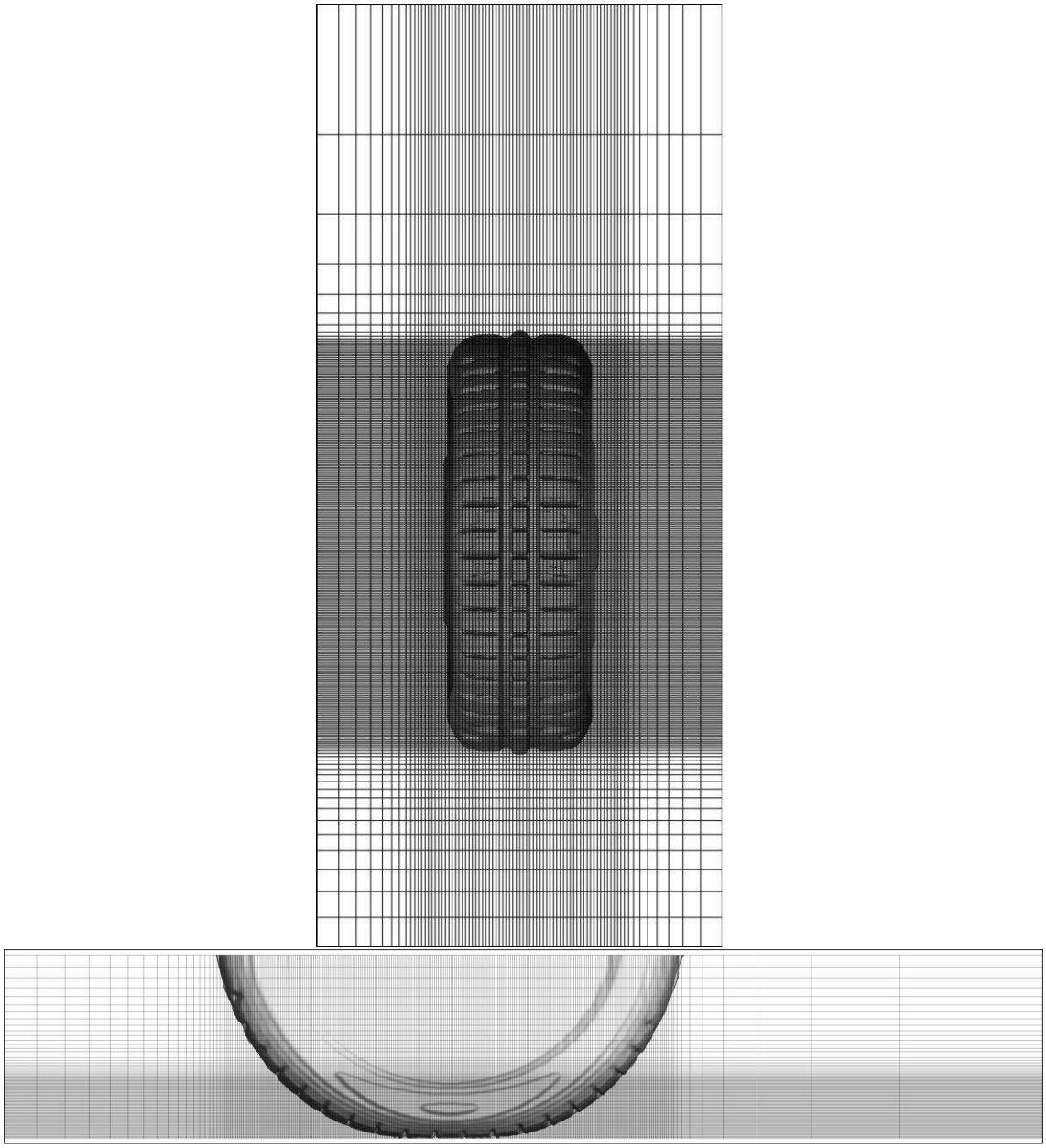


Figure 13.2: Structure of the calculation grid in an horizontal (top) and a vertical slice (bottom).

13.1:

$$P_h = KV_v^2 = 96466Pa \tag{13.5}$$

with the constant K is equal to 500 and the velocity of the vehicle V_v is $13.89m.s^{-1}$. On a two-phase point of view, it is observed that air tubes are generated when the water touches the tire on the side and in the wake of the obstacle. Under surface tension and shearing effects, these gas tubes break and generate bubbles and droplets, as can be observed in hydroplaning experiments.

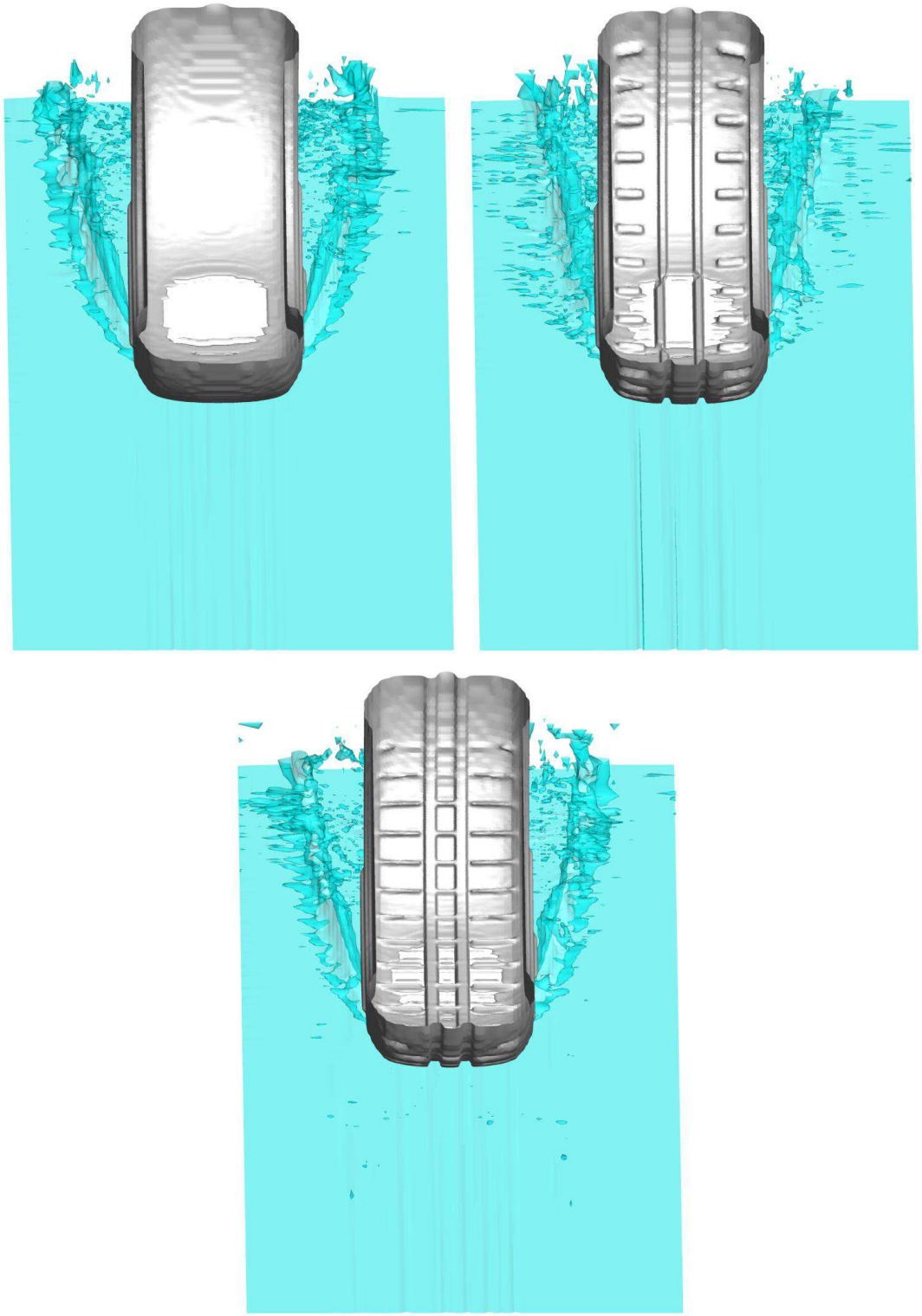


Figure 13.3: Two-phase flow interacting with tires $T1$, $T2$ and $T3$ - the tire iso-surface is plotted in grey whereas the free surface is represented in blue.

13.3.3 Analysis of forces exerted on a tire by water

In this section, the vertical component of the normal force F_n exerted by the two-phase flow on the tire is first studied. The positive and negative contributions of F_n, defined as F_n^+ and F_n^-, are introduced in order to estimate and discriminate the tire structure effect on the flow-structure interaction. The behavior of F_n^+ and F_n^- is proposed in figure 13.5. For the three tires, the positive contribution of the vertical component of force F_n is 7 to 8 orders of magnitude higher than the negative part F_n^-. This observation illustrates the hydroplaning character of the fluid-structure interaction considered in this work.

The time evolution of F_n^+ admits a similar characteristic behavior for $T1$, $T2$ and $T3$. For $0s \leq t \leq 0.01s$, F_n^+ increases until a maximum value between $800N$ and $900N$. This time interval correspond to time required for the incident water layer to wet the tire surface near the road. For $0.01s \leq t \leq 0.02s$, the positive part of the vertical force component decreases, corresponding to the obtention of an almost equilibrium state between the incident water flow and the tire and road dynamics. After this time, the effort reaches an average asymptotic value included in the range $750N$ to $850N$. The main difference between the flat and structured tires is observed in the asymptotic region, for which the vertical force exerted by the two-phase flow is constant for $T1$ (flat tire) whereas regular oscillations are numerically measured for $T2$ and $T3$, characteristic of the tire structure.

The discrimination of the tire is now investigated by considering the evolution of the total vertical force F_n during time. These results are presented in figure 13.6. It is observed that building a structure on a tire reduces by 20% the effort exerted by water on the tire, as observed experimentally by tire manufacturers. As for F_n^+ and F_n^+, after the fluid-structure interaction has reach a stabilized state for $t \geq 0.02s$, F_n linearly increases over time. This results in the correlated increase of the ambient pressure in the calculation domain. The real effort exerted on the tire after $t = 0.02s$ no more increases. This could be verified numerically by subtracting the ambient pressure to the pressure calculated locally. The difficulty lies in the choice of the ambient pressure in our simulations. However, the important feature here lies in the classification of the forces resulting from the fluid-structure interaction according to the tire structure, which is nicely established by the simulations. Contrary to the classification brought by F_n^+, the F_n curve demonstrates that the tire $T3$ involves a 5 to 10% in average lower effort than $T2$. For these to tire geometries, the main difference lies in the opportunity provided by $T3$, due to its shape design, to exert a negative vertical effort which compensate the value of F_n^+ and allows F_n to be lower for $T3$ than for $T2$.

A last interesting parameter can be extracted from the simulated efforts: the characteristic frequencies arising when the tires are structured. It is recalled and observed in the simulations that no typical periodic variations are observed for a flat tire, as expected. The best variable allowing to measure the characteristic time variations of efforts is F_n^+, as observed in figure 13.5. A zoom of the positive vertical contributions of F_n is presented in figure 13.7. The analysis of the F_n^+ signals allows to extract the characteristic frequencies f_{T2} and f_{T3} of tires $T2$ and $T3$:

$$\begin{cases} f_{T2} = 25/0.0892545 = 280Hz \\ f_{T3} = 32/0.0910765 = 351Hz \end{cases} \tag{13.6}$$

It can be tried to correlate f_{T2} and f_{T3} to the typical tire structures presented in figure 13.8. A frequency can be estimated by dividing the velocity of the road $V_v = 13.89m/s$ by a characteristic

distance. If the vertical distance between the tire structures is used, it can be demonstrated that

$$\begin{cases} f_{T2} = 13.89/0.048 = 289Hz \\ f_{T3} = 13.89/0.04 = 347Hz \end{cases} \tag{13.7}$$

As a conclusion, it has been demonstrated that the fluctuations observed on the time evolution of the efforts are directly dependent on the size of the larger structure of the considered tire.

13.4 Concluding remarks

The three-dimensional two-phase flow structure interacting with the tire are clearly simulated. The corresponding efforts exerted on the tire are compared for three different tires. A classification of the tire topologies is proposed with respect to the magnitude of the total vertical normal forces, *i.e.* the flat tire involves a 20% higher effort than the structure tire. This demonstrates that using structure tire clearly reduces the vertical force exerted on the tire and that in this case, the hydroplaning will occur for higher road velocities. To finish with, characteristic frequencies of structured tires are clearly observed. They are related to the size of the larger tire structure.

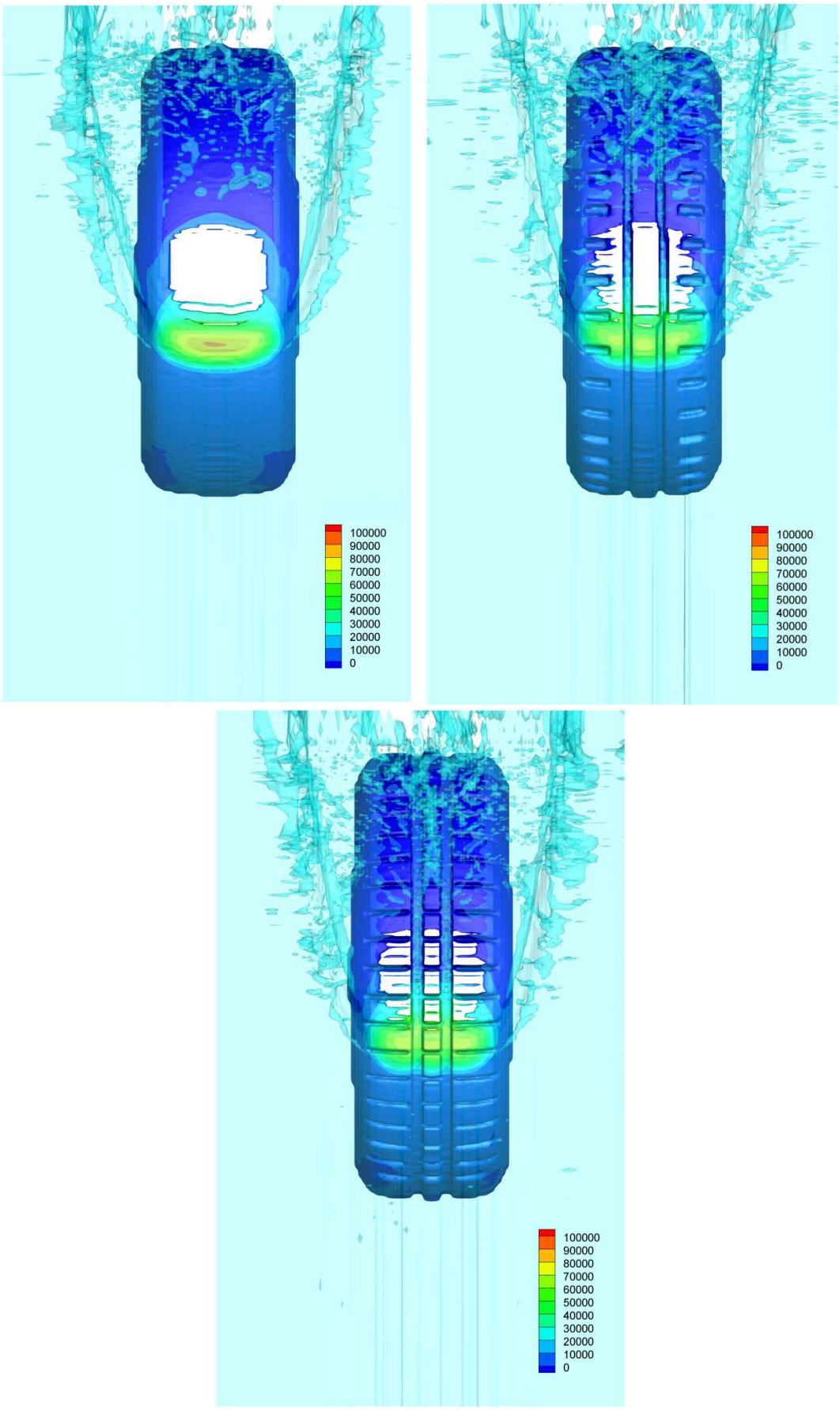


Figure 13.4: Two-phase flow interacting with tires $T1$, $T2$ and $T3$ - the free surface is represented in blue with translucency whereas the iso-colors describe the pressure projected onto the tire surface.

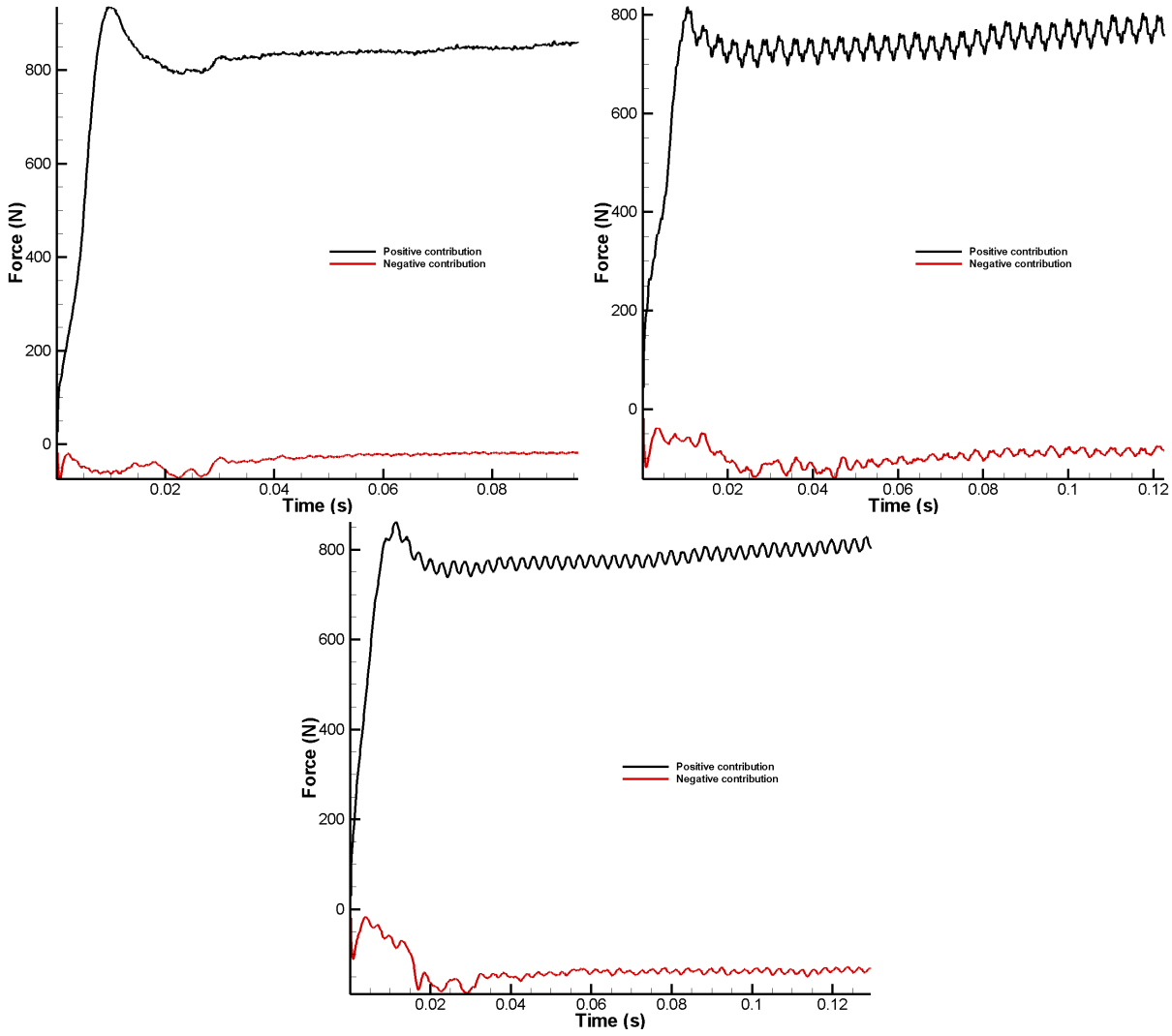


Figure 13.5: Time evolution of F_n^- and F_n^+ for tires $T1$, $T2$ and $T3$ (from left to right and top to bottom

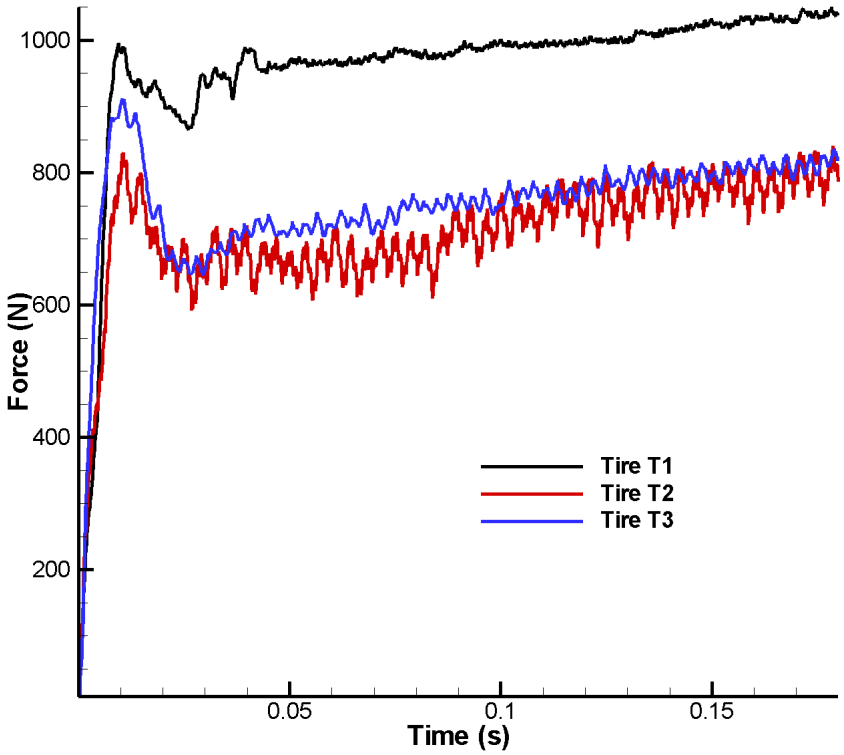


Figure 13.6: Time evolution of F_n for tires $T1$, $T2$ and $T3$

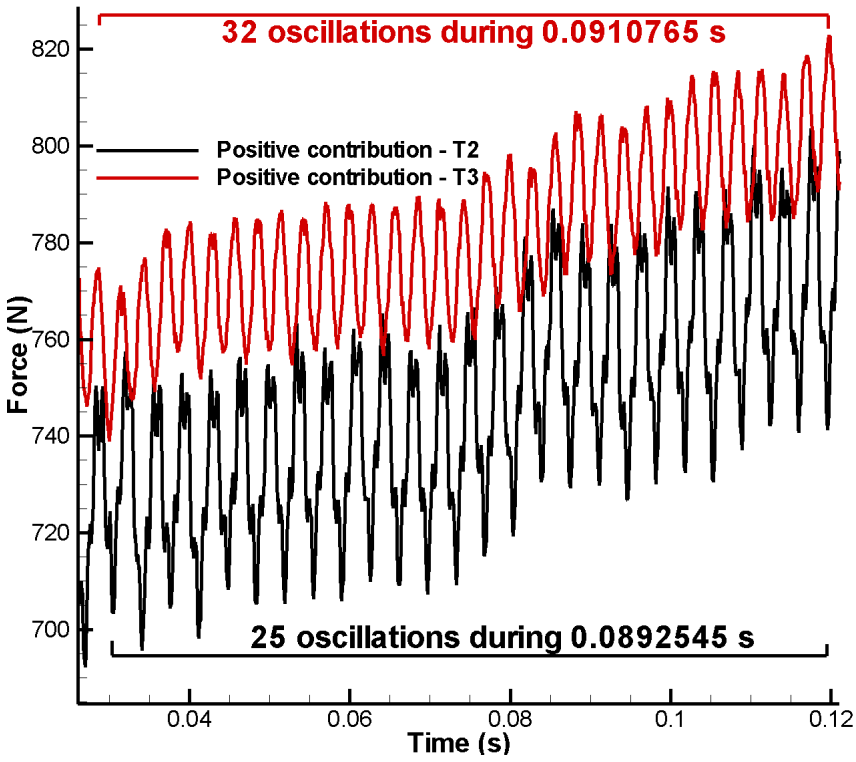


Figure 13.7: Zoom on the time evolution of F_n^+ for tires $T2$ and $T3$

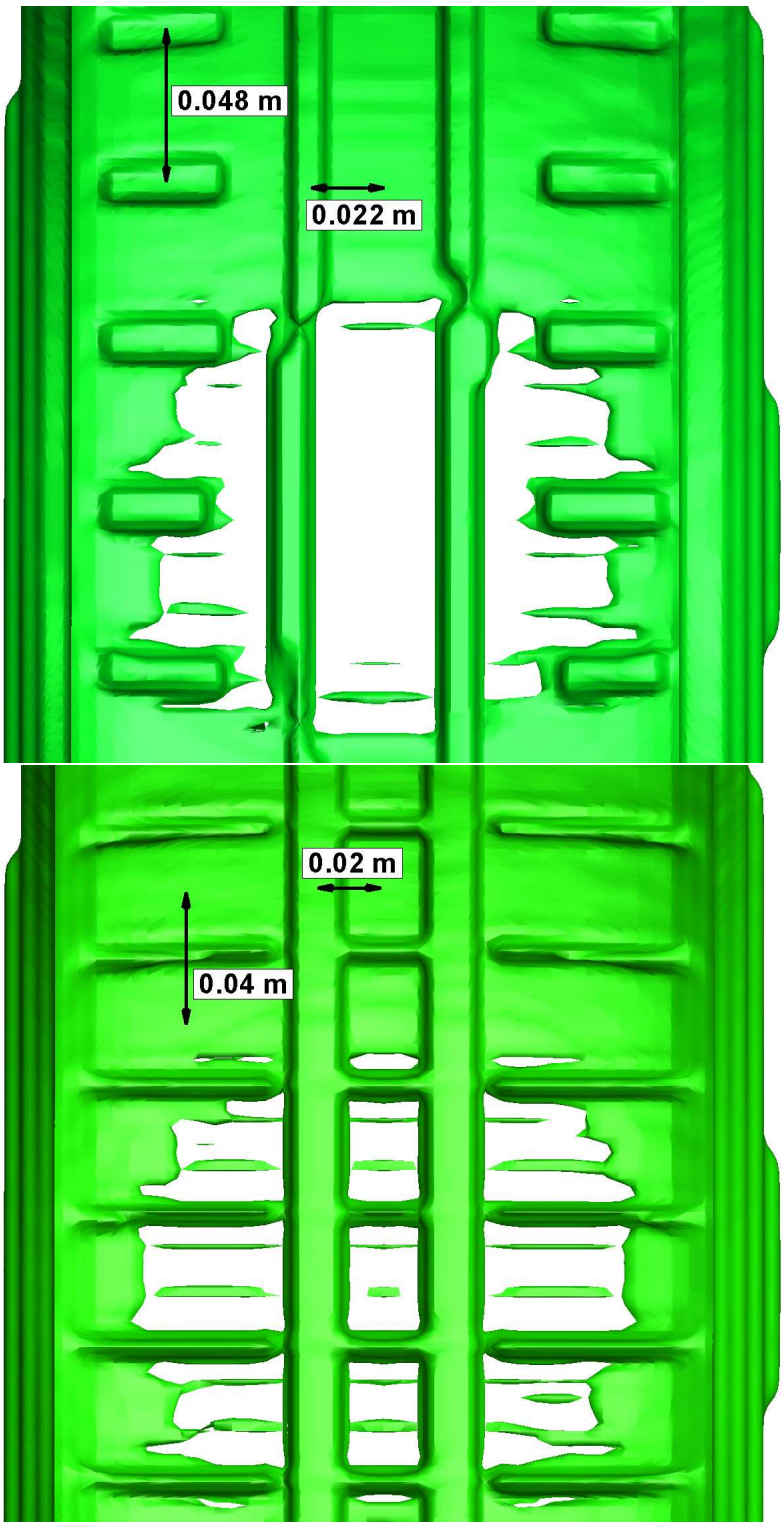


Figure 13.8: Typical structure of tires $T2$ (top) and $T3$ (bottom) - a downside view of the tires is presented at the contact area with the road.

Chapter 14

The Lascaux cave

14.1 Context

LASCAUX is the setting of a complex of caves in southwestern France famous for its Paleolithic cave paintings (Fig. 14.1). The original caves are located near the village of Montignac, in the Dordogne département. They contain some of the most well-known Upper Paleolithic art. These paintings are estimated to be 16,000 years old. They primarily consist of realistic images of large animals, most of which are known from fossil evidence to have lived in the area at the time. In 1979, Lascaux was added to the UNESCO World Heritage Sites list along with other prehistoric sites in the Vézère valley. Since its discovery, several problems have occurred, due to the huge amount of visitors, and their release of vapor and carbon dioxide by their breath, causing the formation of calcite and the apparition of green algae and mosses. The Minister of Cultural Affairs (André Malraux) had the cave closed in 1963. The closure solved some of the problems for a while and the Lascaux cave art returned to the state it was in the day of the discovery. Since then, prehistorians, archeologists, geologists, hydrogeologists, have tried hard to maintain the cavity in the most stable state possible, using remote metering to record the variations in temperature, hygrometry, and carbon dioxide gas pressure. The biological equilibrium remained fragile and in 2001 colonies of micro- organisms, fungi and bacteria developed on the rock edges and on the floor. This attack made the authorities and the Minister of Culture and Communication create an international committee of the Lascaux cave. This multidisciplinary committee is composed of archeologists, physicists, geologists, hydrogeologists and conservators working altogether to understand the mechanisms of apparition of the micro-organisms in order to stop their propagation. Since then, biologists have developed treatments and complex processes to eradicate these micro-organisms.

In the process of time the temperatures and hydric conditions have often changed. Under the influence of exchanges and energy transfers with the outside, the system formed by the Lascaux cave evolved and its state variables have been modified. Climate change had consequences which occurred before its discovery which can be observed in the paintings on various places of the cave.

Among the measures taken by the committee, a better understanding of the flows in the cave was deemed a paramount importance, and has induced the creation of a simulation tool, the "Lascaux Simulator". The non intrusive character of the simulation is one of the major assets of this method. Thus, the numerical simulation in fluid mechanics is here dedicated to the conservation of the Lascaux cave. The project is conducted by Delphine Lacanette (TREFLE).

Two articles are now presented. The first one explain the methodology employed to simulate

Figure 14.1: Paintings in the Lascaux cave - Room of the Bulls, first and second bulls

the natural convection in the Lascaux cave on Cartesian grid. The moisture is taken into account. The second article propose to carry out the same simulations on curvilinear grids. Both approaches are validated on academic cases. The Sierpinski carpet case is presented in each articles with the methodology employed therein.

14.2 The article in International Journal of Heat and Mass Transfer

International Journal of Heat and Mass Transfer 52 (2009) 2528–2542

Contents lists available at ScienceDirect

International Journal of Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/ijhmt

An Eulerian/Lagrangian method for the numerical simulation of incompressible convection flows interacting with complex obstacles: Application to the natural convection in the Lascaux cave

Delphine Lacanette [a,c,*], Stéphane Vincent [a,c], Arthur Sarthou [a,c], Philippe Malaurent [b], Jean-Paul Caltagirone [a,c]

[a] University of Bordeaux, Transferts, Écoulements, Fluides, Énergétique (TREFLE), UMR CNRS 8508 Site ENSCPB, 16 Avenue Pey-Berland, 33607 PESSAC Cedex, France
[b] University of Bordeaux, Géosciences, Hydrosciences, Matériaux, Construction (GHYMAC), Avenue des Facultés, 33405 TALENCE Cedex, France
[c] CNRS, Laboratoire TREFLE, Pessac Cedex, F33607, France

ARTICLE INFO

Article history:
Received 18 March 2008
Received in revised form 8 December 2008
Available online 9 March 2009

Keywords:
Incompressible flows
Natural convection
Penalty methods
Lascaux cave
Fictitious domains
Eulerian/Lagrangian grid coupling

ABSTRACT

An Eulerian/Lagrangian method for the numerical simulation of incompressible convection flows interacting with complex obstacles is developed in this article. This method is successfully validated on natural convection cases, a porous medium and the Sierpinski carpet. The ability of the model to take accurately into account complex topologies allows its application to natural convection in the Lascaux cave, in order to give information on velocities, temperature and moisture content values and to provide helpful details to the conservators of the cave.

1. Introduction

Designing a numerical model to deal with natural convection in cavities as well as the conduction into the rock surrounding the cave is of major importance for the conservation of confined areas. Moreover, the complex topology of the cave has to be accurately taken into account by the numerical tool. Two approaches are currently encountered. On the one hand, using body-fitted unstructured grid [1,2] the method consists in considering two subdomains with their own grids. They are connected by a boundary condition corresponding to the interface between the solid and the fluid media. The solutions in each subdomain are connected thanks to jump conditions on mass, momentum and energy. The main advantages of unstructured methods are that they naturally take into account the complex shape of the objects and they provide an explicit description of the interface between the media, in order to apply the real physical jump or transmission conditions. Concerning the drawbacks of this method, the grid generation is complex and even impossible due to the strong irregularities of the fluid–solid interface. On the other hand, another numerical methodology for treating the fluid–solid interaction is the fictitious domain approach developed during the last 15 years by many authors [3–6] or [7]. This technique is based on the concept of using a structured grid for dealing with conservation equations such as the Navier–Stokes or energy equations. The obstacles or solids are drawn into the structured simulation grid and specific terms are added to the conservation equations in order to account for the presence of obstacles. The major advantages of this method are its readiness to implement, even in three dimensions and its ability to be integrated to existing CFD tools. Furthermore, it can deal with moving solids and several approaches have been extended to high order [7] or [8]. The main drawback is the relative lack of accuracy in the description of the boundary layers as the grid is not *a priori* adapted to the shape of the obstacles.

The management of complex shaped objects, such as those involved in the last section of this paper, requires months of work with grid generators in order to build unstructured meshes of good quality in the fluid and solid media. In addition, if moving solid objects interact with the flow motion, the 3D unstructured grid must be updated by means of an automatic procedure. This operation is not possible in certain situations. For all these reasons, we have chosen to simulate the natural convection in a complex shape cave using the fictitious domain approach and penalty methods, following the works [3,4,9,8].

The Lascaux cave, discovered in 1940 and located in the Dordogne area in France, is inscribed on the Unesco World Heritage List. It is considered as one world's major prehistoric caves. Since its

* Corresponding author. Address: CNRS, Laboratoire TREFLE, Pessac Cedex, F33607, France. Tel.: +33 5 40 00 28 31; fax: +33 5 40 00 66 68.
E-mail address: lacanette@enscpb.fr (D. Lacanette).

0017-9310/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijheatmasstransfer.2008.12.028

Nomenclature

Latin letters

a	thermal diffusivity (m^2/s)
C_P	specific heat (J/kg K)
dr	augmented Lagrangian parameter (Pa.s)
D	diffusion coefficient (m^2/s)
D_h	hydraulic diameter (m)
E	characteristic dimension (m)
F_C	moisture source term (g/kg dry air s)
$\mathbf{g}$	gravity vector (m/s^2)
h	exchange coefficient (W/m^2 K)
K	permeability (m^2)
p	pressure (Pa)
T	temperature (K)
t	time (s)
$\mathbf{u}$	velocity (m/s)

Greek symbols

β	expansion coefficient (K)
λ	conductivity (W/m K)
μ	dynamic viscosity (Pa s)
ϕ	absolute moisture content (g/kg dry air)
ρ	density (kg/m^3)
χ	color function

Subscripts and superscripts

f	fluid medium
s	solid medium
I	interface

Non-dimensional

$Ra = \frac{\rho^2 C_P g \beta \Delta T E^3}{\mu\lambda}$	Rayleigh number
$Da = \frac{hE}{\lambda}$	Darcy number
$Le = \frac{a}{D}$ with $a = \frac{\lambda}{\rho C_p}$	Lewis number
$Nu = \frac{hE}{\lambda}$	Nusselt number
$Pr = \frac{\mu C_p}{\lambda}$	Prandtl number
$Ra^* = Ra\ Da = \frac{\rho^2 C_P g \beta \Delta T E^3}{\mu\lambda} \frac{K}{E^2}$	filtration Rayleigh number
$Sc = \frac{\mu}{\rho D}$	Schmidt number

discovery, several problems have occurred, due to the huge amount of visitors [10], and their release of vapor and carbon dioxide by their breath, causing the formation of calcite and the apparition of green algae and mosses. The Minister of Cultural Affairs (André Malraux) had the cave closed in 1963.

The closure solved some of the problems for a while and the Lascaux cave art returned to the state it was in the day of the discovery. Since then, prehistorians, archeologists, geologists, hydrogeologists, have tried hard to maintain the cavity in the most stable state possible, using remote metering to record the variations in temperature, hygrometry, and carbon dioxide gas pressure. The biological equilibrium remained fragile and in 2001 colonies of micro-organisms, fungi and bacteria developed on the rock edges and on the floor. This attack made the authorities and the Minister of Culture and Communication create an international committee of the Lascaux cave. This multidisciplinary committee is composed of archeologists, physicists, geologists, hydrogeologists and conservators working altogether to understand the mechanisms of apparition of the micro-organisms in order to stop their propagation. Since then, biologists have developed treatments and complex processes to eradicate these micro-organisms [11,12].

In the process of time the temperatures and hydric conditions have often changed [13]. Under the influence of exchanges and energy transfers with the outside, the system formed by the Lascaux cave evolved and its state variables have been modified. Climate change had consequences which occurred before its discovery which can be observed in the paintings on various places of the cave.

Among the measures taken by the committee, a better understanding of the flows in the cave was deemed a paramount importance, and has induced the creation of a simulation tool, the "Lascaux Simulator". The nonintrusive character of the simulation is one of the major assets of this method. Thus, the numerical simulation in fluid mechanics is here dedicated to the conservation of the Lascaux cave. It has previously been studied by Ferchal [14,15] with the CFD code developed by EDF on an unstructured grid. The present work constitutes a different view of the problem, using a fictitious domain approach [16,17].

The numerical methodology used in this paper is first exposed. The governing equations and the numerical modeling of solid walls are detailed. Then the method is validated on two different cases, the natural convection in a porous medium, and the case of Sirpinski carpet. Finally, the method is applied to the study of the natural convection in the Lascaux cave, and information about the distribution of temperature and moisture contents considering different thermal configurations is provided.

2. Numerical methodology

2.1. Conservation equations

In the fluid medium (see Fig. 1) Ω_f, the conservation equations describing the unsteady incompressible convection flows of a

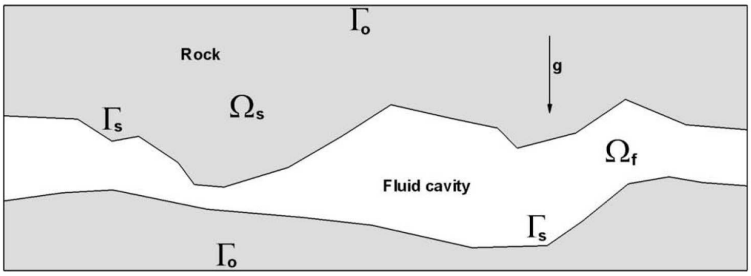


Fig. 1. Definition sketch.

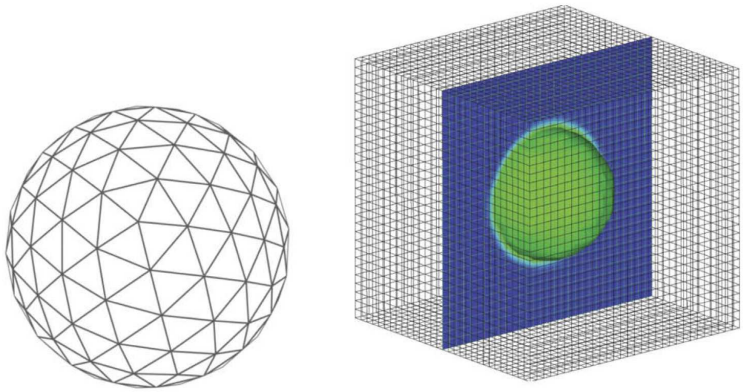


Fig. 2. Example of Lagrangian grid and corresponding projected solid fraction χ (the $\chi = 0.5$ isosurface and isocontours of χ in a slice are plotted).

Newtonian fluid and the evolution of the moisture concentration, under the Boussinesq approximation, are the Navier–Stokes, energy and transport of moisture concentration equations written in terms of velocity and temperature:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u})\right) = -\nabla p + \rho \mathbf{g} + \nabla \cdot (\mu[\nabla \mathbf{u} + \nabla^T \mathbf{u}]) \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

$$\rho C_P\left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T\right) = \nabla \cdot \lambda \nabla T \quad (3)$$

$$\left(\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi\right) = \nabla \cdot (D \nabla \phi) \quad (4)$$

It is assumed that λ, C_P, D and μ are constant with respect to T. The density variations are described by the following linear expression:

$$\rho(T) = \rho_0(1 - \beta[T - T_0]) \quad (5)$$

In the solid part Ω_s, the velocity is assumed to be negligible. Only conduction effects drive the thermal exchanges in this case. We have

$$\mathbf{u} = 0 \quad \text{and} \quad \phi = 0 \quad (6)$$

$$\rho C_P \frac{\partial T}{\partial t} = \nabla \cdot \lambda \nabla T \quad (7)$$

The boundary conditions for the velocity, temperature fields and moisture content concentration are the following:

$$\mathbf{u} = 0 \quad \text{and} \quad -\lambda_f \frac{\partial T_f}{\partial n} = -\lambda_s \frac{\partial T_s}{\partial n} \quad \text{and} \quad \phi = 0.5545 \times T + 1.8909 \text{ on } \Gamma_s \quad (8)$$

$$\mathbf{u} = 0 \quad \text{and} \quad \frac{\partial T_s}{\partial n} = 0 \quad \text{and} \quad \phi = 0 \text{ on } \Gamma_o \quad (9)$$

2.2. Numerical modeling of solid walls

2.2.1. Management of the fluid/solid interface

The present work aims at proposing a numerical method which is able to deal with fluid/solid interaction while using structured grids non conforming to the complex shape of the obstacles. The main idea is a continuation of the previous works of Caltagirone et al. [18,3] concerning fictitious domains. The method is structured as follows:

⊗ the simulation domain $\Omega = \Omega_s \cup \Omega_f$ which includes both the fluid and solid zones is discretized with a global structured grid *a priori* not adapted to Γ_s. On this Eulerian grid, standard numerical methods apply (see below).

⊗ the topology of the fluid/solid interfaces is not explicitly known on the Eulerian structured grid which doest not fit to Γ_s. As a consequence, the jump conditions cannot be explicitly implemented in Eqs. (5)–(9). The fictitious domain approach consists in introducing local volume effects in the conservation equations so as to account for the volume effect of the solid medium in Ω_s. In a first step, a triangular Lagrangian grid $\mathbf{x}_f$ of the solid zones is projected onto the Eulerian grid (see Fig. 2) by solving a diffusion equations as follows [19]:

$$\Delta\chi = \nabla \cdot \int_{\Gamma_s} \mathbf{n}_i(\mathbf{x} - \mathbf{x}_f)\delta_i ds \quad (10)$$

where χ is the local volume fraction of the solid, $\mathbf{n}_i$ is the normal to Γ_s, $\mathbf{x}$ is a location on the Eulerian grid and δ_i is the Dirac function indicating the interface. After solving Eq. (10), $\chi = 1$ in Ω_s and 0 elsewhere. The interface Γ_s between Ω_f and Ω_s is defined as $\chi = 0.5$.

⊗ function χ allows us to locate the solid and fluid part on the Eulerian grid. A Darcy term is added to the momentum equations in order to penalize the solid behavior through the conservation equations. The new penalty Navier–Stokes Brinkman model so obtained, which replaces Eq. (1), reads:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u})\right) + \frac{\mu}{K}\mathbf{u} = -\nabla p + \rho \mathbf{g} + \nabla \cdot (\mu[\nabla \mathbf{u} + \nabla^T \mathbf{u}]) \quad (11)$$

where $K = +\infty$ if $\chi < 0.5$ and $K = 0$ if $\chi \geqslant 0.5$. To sum up, our fictitious domain method uses penalty terms added to the momentum conservation equations to model the presence of solid obstacles.

2.2.2. Management of humidity transfer

The humidity transfer is estimated directly on the real Lagrangian surface rather than the projected Eulerian one. It is made of two steps.

1. The temperature gradient in the normal direction for each surface triangle $\mathbf{S}_l$ is first calculated as follows:
   - ⊗ a normal vector $\mathbf{N}_l$ to $\mathbf{S}_l$ is defined. It is oriented toward the fluid medium, starting from the barycenter of $\mathbf{S}_l$ and its length is the local grid space.
   - ⊗ the temperatures T_b and T_f, respectively, at the barycenter and in the fluid, i.e. at the edges of $\mathbf{N}_l$, are evaluated by linearly interpolating the Eulerian field.
   - ⊗ the temperature gradient is directly obtained by $\nabla_\perp T_l = \frac{T_f - T_b}{\|\mathbf{N}_l\|}$.

2. A specific penalty term is added in Eq. (4) in order to impose the moisture content in the cell cut by the air/rock interface:

$$\frac{\phi^{n+1}-\phi^{n}}{\Delta t}+\mathbf{u^{n+1}}\cdot\nabla\phi^{n+1}=\nabla\cdot(D\nabla\phi^{n+1})+B\left(\phi^{n+1}-f(\nabla_{\perp}T_{I}^{n})\right) \tag{12}$$

where B is a penalty coefficient equal to 10^{40} in the cells cut by the interface and 0 elsewhere, and f is a function based on an energetic balance between the heat driven by the conduction through each triangle and the energy necessary to evaporate or condensate all the vapor contained in an elementary volume of the Eulerian grid. The normal temperature gradient to the interface is $\nabla_{\perp}T_I$. It is calculated on the Lagrangian interface and is projected on the Eulerian grid to build f.

2.3. Discretization and solvers

The system of Eqs. (1)–(9) which describes the interaction between the natural convection in a cavity and the conduction in the rock is discretized thanks to implicit finite volumes [20] on a fixed staggered Cartesian Marker And Cell (MAC [21]) grid. In Ω_f, the coupling between pressure and velocity, as well as the incompressibility constraint, are fulfilled by using a minimization algorithm called augmented Lagrangian [22]. The time derivatives are discretized by first order Euler schemes, whereas the spatial derivatives are approximated by centered schemes.

The implicit discretization of the momentum and energy equations involves the solving of a linear system $A^nX^{n+1}=F^n$ by using an iterative BiCGSTAB solver [23], preconditioned under a Modified and Incomplete LU (MILU) method [24]. The index n is related to time $n\Delta t$ for which Δt is the time step. The mass and momentum equations are first solved by the augmented Lagrangian method in order to obtain $(\mathbf{u}^{n+1},p^{n+1})$ as follows:

$$\begin{aligned}
&k=0, p^0=p^n \quad \text{and} \quad \mathbf{u}^0=\mathbf{u}^n\\
&\text{Solve}\\
&k=k+1\\
&\rho^n\left(\frac{\mathbf{u}^k-\mathbf{u}^n}{\Delta t}+\nabla\cdot(\mathbf{u}^{k-1}\otimes\mathbf{u}^k)\right)\\
&\quad=-\nabla p^{k-1}+dr\nabla(\nabla\cdot\mathbf{u}^k)\\
&-\frac{\mu}{K}\mathbf{u}^k+\rho\mathbf{g}+\nabla\cdot(\mu[\nabla\mathbf{u}^k+\nabla^T\mathbf{u}^k])\\
&p^k=p^{k-1}-dr\nabla\cdot\mathbf{u}^k\\
&\text{While}\nabla\cdot\mathbf{u}^k\geqslant\epsilon|
\end{aligned} \tag{13}$$

where ϵ is chosen equal to almost zero computer error ($\epsilon=10^{-15}$ in double precision calculations). At the end of the minimization procedure, we assume that $\mathbf{u}^{n+1}=\mathbf{u}^k$ and $p^{n+1}=p^k$. The minimization parameter dr can be set constant or determined automatically by analyzing the physical or numerical parameters of the problem [25,26]. In our simulations, the augmented Lagrangian parameter $dr=1$ as the local variations of density and viscosity are small. In addition, the permeability is chosen equal to 10^{40} in the fluid and 10^{-40} in the solid. The magnitude of K has no effect on the efficiency of the iterative solver as the permeability only affects diagonal coefficients in the linear system. More details on the convergence order and behavior of the penalty method are given for example in [3].

Then, the temperature T^{n+1} is obtained by:

$$\rho^nC_P\left(\frac{T^{n+1}-T^n}{\Delta t}+\mathbf{u}^{n+1}\cdot\nabla T^{n+1}\right)=\nabla\cdot\lambda\nabla T^{n+1}$$

To finish with, the vapor concentration ϕ^{n+1} is obtained by:

$$\phi^*=\phi^n+\Delta t*B(\phi^*-f(\nabla_{\perp}T_I^n))$$

$$\left(\frac{\phi^{n+1}-\phi^*}{\Delta t}+\mathbf{u}^{n+1}\cdot\nabla\phi^{n+1}\right)=\nabla\cdot D\nabla\phi^{n+1}$$

More information, details and validations about the discretizations and solvers have been extensively investigated in previous works [3,27].

3. Validation

3.1. Natural convection in a porous medium

Following the works of Arquis [28], the interest and accuracy of the Brinkman penalty method can be illustrated by simulating on a local scale the natural convection in a square cavity, differentially heated between a cold temperature wall T_c and a hot one T_h, in which cylindrical inclusions are placed following a square shaped periodical network (see Fig. 3). It is assumed that the conductivity of the cylindrical obstacle is the same as the fluid. Under the action of gravity, natural convection flows develop in the cavity between the cylinders. The main dimensionless parameter of the problem is the Rayleigh number Ra. The Prandtl number of the problem is assumed to be equal to 1.

It is proposed to consider several configurations in which the porosity is constant and equal to 0.615 and the number of cylinders N is increased progressively from 4^2 to 32^2. Due to the presence of the obstacles, the flow velocities are decreased and the cavity behaves as a porous medium. In order to take into account the effects of the cylinders on the convection, a modified Rayleigh number, called the filtration Rayleigh number, is introduced. Ra^* is equal to the product between the classical Rayleigh and the Darcy number. In the following simulations, Ra^* is assumed constant and equal to 122.6.

We consider three convection cases where N of 4^2, 8^2 and 32^2, are associated to 64^2, 256^2 and 512^2 simulation grids, respectively. The stationary results of temperatures, streamlines and obstacles are presented in Fig. 5. For $N=4^2$, the isotherms are irregular, demonstrating a different thermal behavior in fluid and solid media. In this case, a macroscopic analysis of the results is not possible. On the contrary, for higher values of N, the flow and temperature isolines are smoother. A macroscopic study is suitable in these configurations.

A quantitative exploitation of the simulations is interesting by analyzing the evolution of the global thermal exchanges in the cavity in terms of Nusselt number Nu. The values of Nu according to $1/N$ are presented in Fig. 4. They all have been obtained with the penalty Brinkman method, except for $1/N=0$ where a Darcy model has been used to simulate this case in particular, assuming a full porous cavity. As a reference, the limit value of the Nusselt number is obtained by simulating with our model a porous medium throughout the whole cavity. This case can be compared to a cavity containing an infinity of cylinder, such as $N\to+\infty$. It can be observed that when N takes large values, the asymptotic value of the Nusselt number tends to the reference value of $Nu=3.574$ obtained with the Darcy penalty model ($Ra\,Da=122.6$).

To sum up, it has been demonstrated that the Darcy penalty method is able to provide a local description of the interaction between natural convection flows and obstacles in a closed cavity. In terms of thermal exchange, the numerical model is accurate and the reference value of 3.574 of a porous medium is recovered in terms of Nusselt number. This test case validates the choice of the Darcy penalty method for simulating the convection flows in the Lascaux cave which highly depend on the fluid–solid interactions.

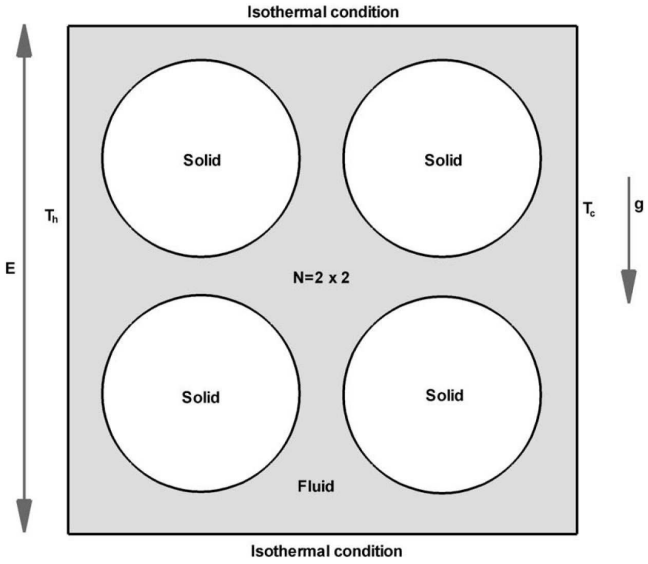


Fig. 3. Definition sketch of the natural convection in a cavity.

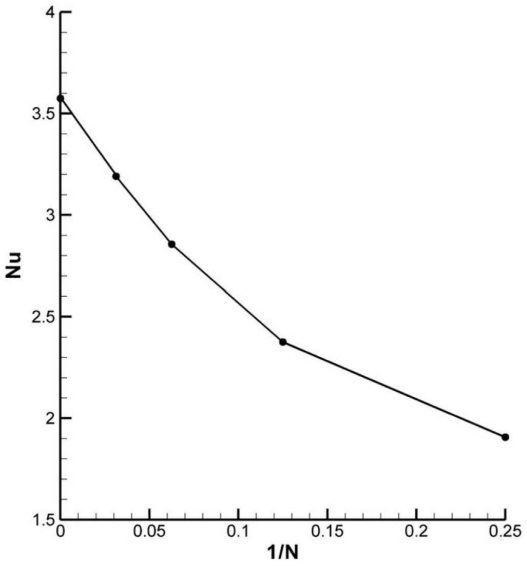


Fig. 4. Evolution of the Nusselt number according to the number N of obstacles in the cavity.

3.2. The Sierpinski carpet

Our purpose in this section is to validate the methodology used to simulate the thermal convection in a 3D cavity comparing simulations with experimental measurements. In [29], Amine et al. compare their numerical results for the Sierpinski carpet case obtained with both experiments and simulations. As the case is not exactly the same for the two methods, experimental and numerical results agree concerning the shape, temperature and velocity profiles. However, the magnitude of the maxima varies in the 20–30% range. Our aim here is to compare the results in [29] with our results in order to demonstrate the validity of our penalty approach.

The Sierpinski carpet is a fractal model composed of squares of various sizes. Fig. 6 illustrates the first three generations of the model. The empty cell is a $100 \times 100\ \text{mm}^2$ box. We impose adiabatic conditions on the upper and lower walls, and a Dirichlet condition of 20 °C and 25 °C on the left and right walls. For all simulations, the liquid used has the following properties: $\mu = 0.0815\ \text{kg}/(\text{m s})$, $\rho = 857\ \text{kg}/\text{m}^3$, $C_P = 1880\ \text{J}/(\text{kg K})$ and $\lambda_f = 0.132\ \text{W}/(\text{m K})$. Hence, the Prandtl number Pr is equal to 1160. Obstacles are in plexiglas for which the thermal conductivity is $\lambda_s = 0.19\ \text{W}/(\text{m K})$.

3.2.1. First generation

The obstacle is located between $33.3\ \text{mm} \leqslant x \leqslant 66.7\ \text{mm}$ and $33.3\ \text{mm} \leqslant y \leqslant 66.7\ \text{mm}$. 2D and 3D simulations are first performed. Fig. 7 shows the vertical velocity profile $V_z(x)$ from experiments [29] and our 2D (100^2 mesh) and 3D (70^3 mesh with $z = 50\ \text{mm}$) computations. Results for 2D configuration are very similar to the numerical simulations of Amine et al. (not represented) which are relatively far from experiments. In this paper, the authors have some suppositions about the differences between experimental and numerical results. First, contrary to their initial assumption, the lower and upper walls are not really adiabatic. Amine et al. have tried to change the upper and lower boundary conditions to the Stefan condition. The gain for V_x was counterbalanced by a loss of quality for V_z. Another difference between experiments and simulations lies in the real size of the obstacles used in experiments. Commercially available bars used for the experiment had dimensions changing by steps of 1 mm.

The box used for the experiment has a depth of only 85 mm. In our opinion the flow can have some 3D structures that cannot be represented by 2D simulations due to confinement effects. As can

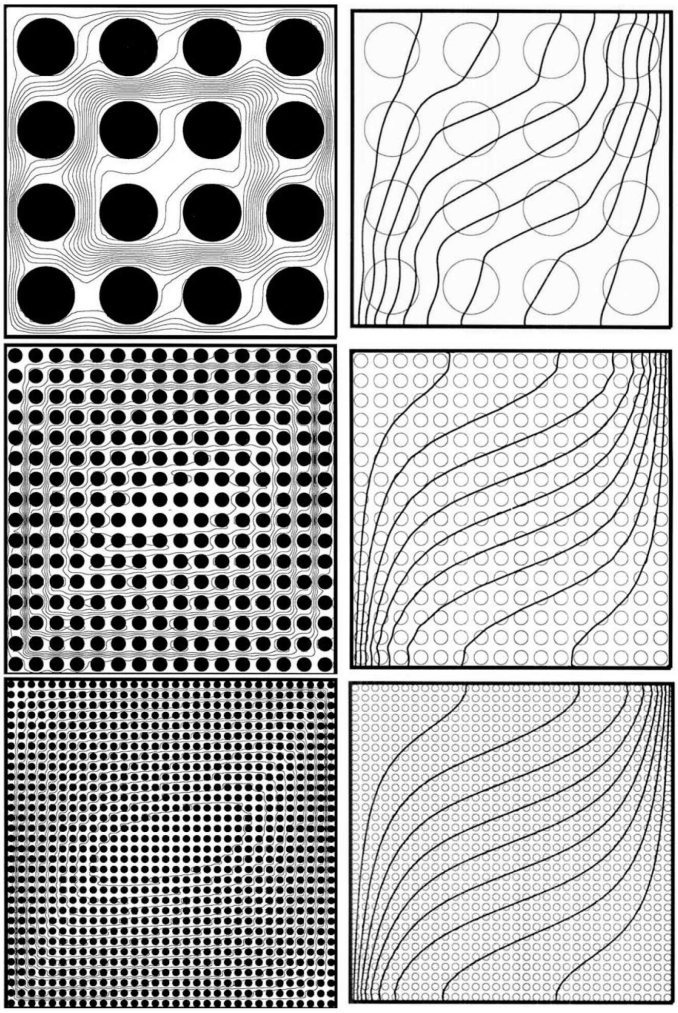


Fig. 5. 2D simulation of the natural convection in a cavity filled with various amounts N of cylinders-$N = 4 \times 4$ with a 64^2 grid, $N = 16 \times 16$ with a 256^2 grid, $N = 32 \times 32$ with a 512^2 grid-left column: streamlines, right column: temperature profiles.

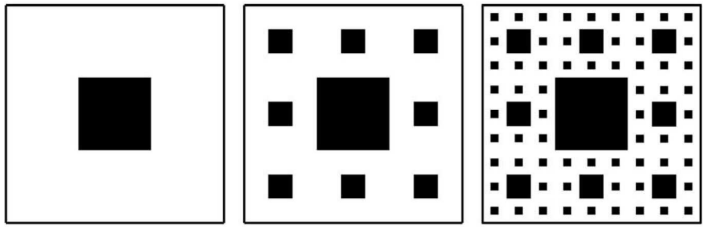


Fig. 6. Sierpinski carpet of 1st, 2nd and 3rd generation.

be seen on Fig. 7, the results obtained with our 3D simulation are closer to the experiment. The difference between the 2D and 3D flow is then studied. Fig. 7 shows the evolution of V_x, V_y and T along the y axis for $z = 10$ mm and $x = 50$ mm. The velocity V_x

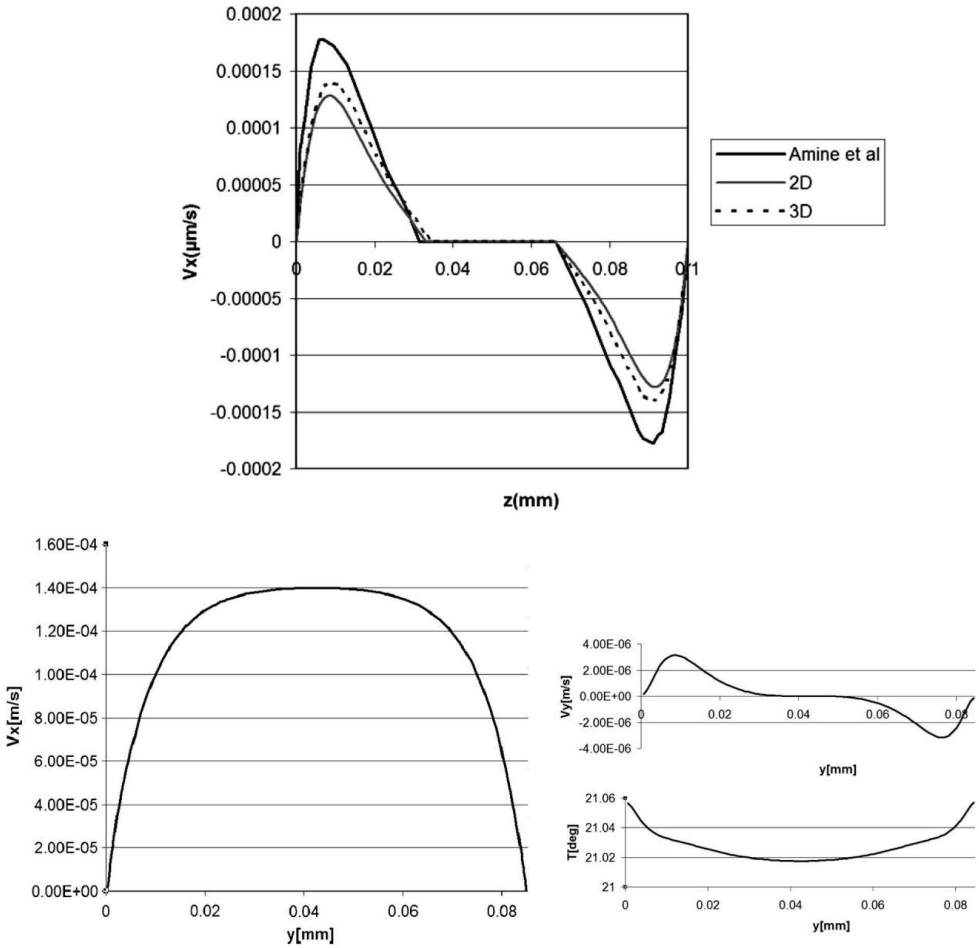


Fig. 7. 3D simulation for the Sierpinski carpet of 1st generation. Horizontal velocity profile $V_x(z)$ at $x = 50$ mm for experiments and numerical simulations (top), values $V_x(y)$, $V_y(y)$, $T(y)$ for $z = 10$ mm and $x = 50$ mm (bottom).

and the temperature T along the y axis for $y \approx 42.5$ mm are quite stable. This is observed in Fig. 9. The transverse velocity V_y is about two magnitudes smaller than V_x. Hence, the flow is mainly 2D but the effect of the 3D structures is not negligible.

3.2.2. Second generation

A new set of eight additional blocks is considered. The new obstacles are quite small, but the mesh is not chosen to match perfectly with obstacles. The purpose here is to demonstrate the interest of our penalty method even if obstacles do not match the grid. Fig. 8 compares the vertical velocity profile $V_z(x)$ at $z = 17$ mm obtained with experiment and simulation. The 2D (with a 150^2 mesh) and 3D (with a 70^3 mesh) simulations are close to the results obtained by experiment. However, the 3D calculation provides a better agreement.

Fig. 8 shows the velocity profile $V_x(z)$ at $x = 50$ mm. As for the first generation carpet, the correspondence between experiment and simulations is not very good for 2D, even if the results obtained with the 3D simulation on a relatively coarse grid are closer to experiment. The results with an additional 2D simulation with a 300^2 mesh are slightly improved.

Streamlines and sensors position is shown in Fig. 9. As in [29], we observe a negative velocity V_z between the 2nd and 3rd lower obstacles.

3.2.3. Conclusion for the Sierpinski carpet

The experiments of Amine et al. [29] and our numerical simulations have been compared. Even on coarse grids, the main structures of the experimental flows have been well reproduced by our methodology and it has been demonstrated that the 3D character of the flow improves the accuracy of the simulations, compared to the 2D computations of Amine et al. However, for various reasons, for instance the inaccurate modeling of the real boundary conditions, differences on velocities have been observed.

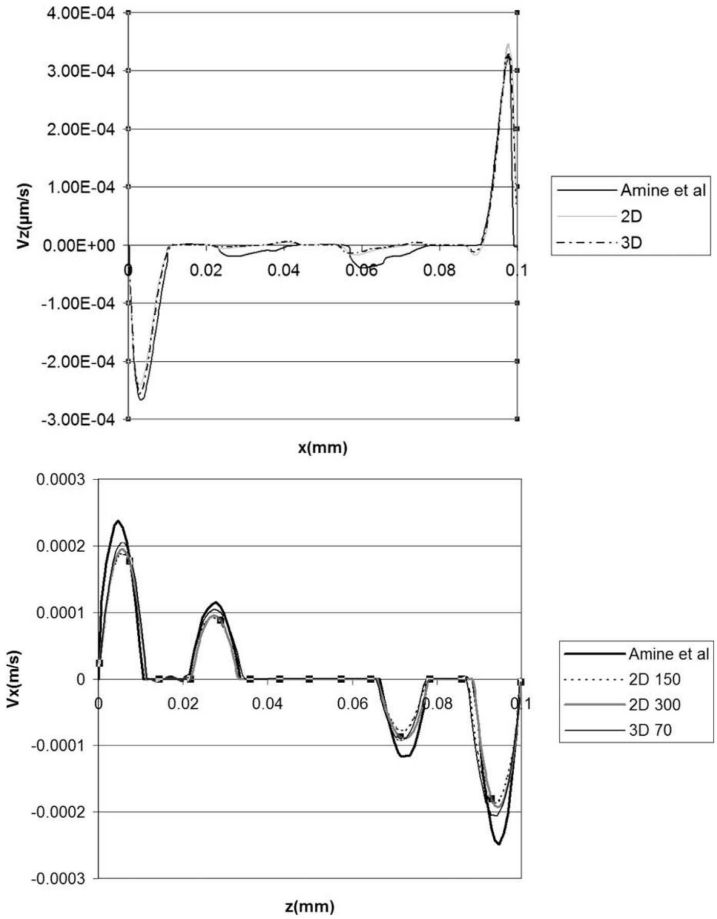


Fig. 8. Sierpinski carpet of 2nd generation. Horizontal velocity profile $V_z(x)$ at $z = 17$ mm (left), horizontal velocity profile $V_x(z)$ at $x = 50$ mm (right) for experiments and numerical simulations.

4. Application to the Lascaux cave

The numerical methodology is applied here to the study of the natural convection in the Lascaux cave. The accurate description of the cavity allows a fine analysis of the flow as well as the moisture content distribution in the cave.

4.1. Initial conditions

The reversal time has been evaluated to 1 h, thus the total simulation time has been fixed to 7 h. The boundary conditions are steady, considering the temperature in the cave is stable during 7 h. Nevertheless, the flow is unsteady, due to the complex geometry and the thermal gradients, as it can be checked in Fig. 10.

4.1.1. Geometry

A three-dimensional survey of the Lascaux cave was made by the land surveyor Perazio using laser scanning. Triangular surface elements of each object interacting with the flow motion are generated. A detail of this surface is shown in Fig. 11. In the following, the gravity acceleration is directed towards the *Y*-axis.

The Lagrangian description of the solid objects is projected onto the fixed Eulerian flow grid as detailed in Section 2.2. The Eulerian view of the geometry as well as reference marks of the Lascaux cave are given in Fig. 11. The global domain of computation is composed of 3.5 million points.

4.1.2. Thermal conditions

Concerning the Rayleigh number, simulations on a differentially heated square cavity are achieved in order to get the evolution of the Nusselt number with the Rayleigh number and to compare the values of velocities with the theoretical ones.

The order of magnitude of the velocity in the boundary layer for high Rayleigh numbers is given by $V_0 = \sqrt{g\beta\Delta T D_h}$. The Table 1 shows a comparison between simulated values of velocity and theoretical ones for a given Rayleigh number.

This comparison is valid for a Rayleigh higher or equal to 10^5, at this point the boundary layers are separated. The order of magnitude of the velocities is the same, the CFD code used as a basis for the simulation in the Lascaux cave gives classical results of natural convection.

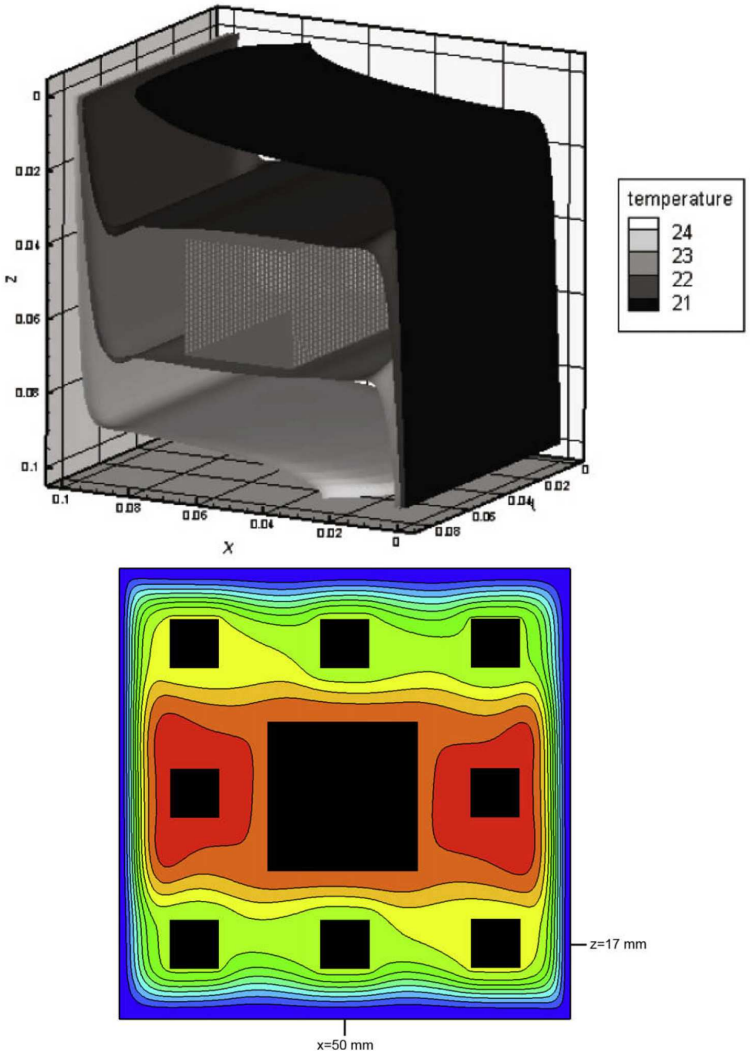


Fig. 9. Isosurfaces of temperature for 3D simulations (top), streamlines for the Sierpinski carpet of 2nd generation (bottom).

The Nusselt numbers corresponding to the simulated velocities have been calculated on a 256×256 Chebyshev grid and compared to reference spectral solutions [30] in Table 2. The calculated values are in good agreement with the benchmark results. Plotting its evolution with the Rayleigh number, the relation (14) is found for a differentially heated square cavity.

$$Nu = 0.17Ra^{0.2821} \tag{14}$$

For a vertical plate, the evolution of the Nusselt number follows the relation (15) [31]:

$$Nu = 0.59Ra^{0.25} \tag{15}$$

This expression is slightly different from the relation (14) due to the containment of the geometry, the boundary layer is finite in the cavity whereas it is considered as infinite in the vertical plate case. Nevertheless the expression (14) is characteristic of a separated boundary layer flow.

In the Lascaux cave, in the thermal configuration of 1981, and without human disturbances, the measured velocities are approximately 5×10^{-2} m/s. Referring to the previous relation, an equivalent Rayleigh number of 10^8 can be given, which corresponds to a differentially heated cavity. This number indicates a laminar flow regime.

The Lewis, Prandtl and Schmidt numbers are, respectively, equal to 1.015, 0.71 and 0.721. The heat, mass and momentum diffusion are of the same order. Thus, the characteristic time and space scales of the involved physical phenomena are compatible; the same time steps and the same grid can be used for all the equations. Our unsteady and deterministic modeling strategy is confirmed by the previous remarks.

The initial conditions in temperature are calculated on a one-dimensional heat conduction model in the floor, based on the temperature measured by Météo France [32] during more than 50

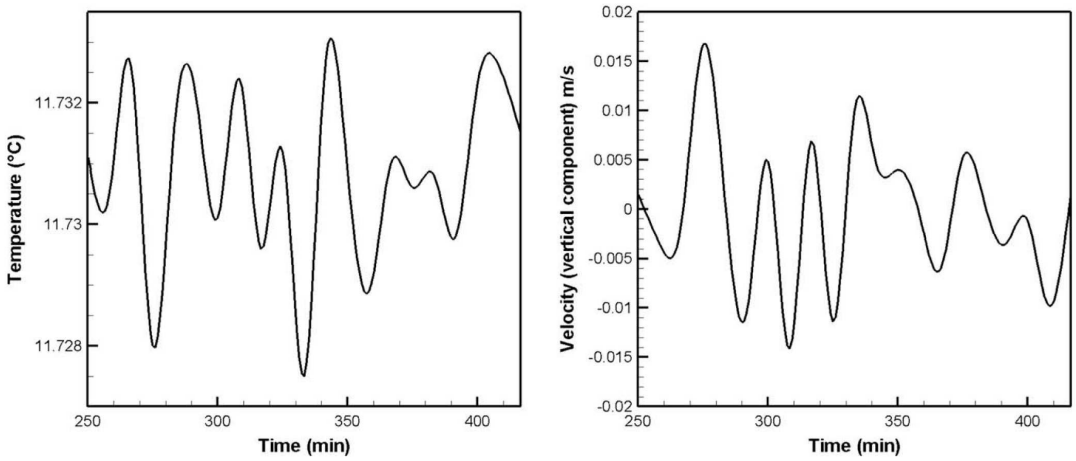


Fig. 10. Evolution of temperature (left) and vertical component of the velocity (right) as a function of time.

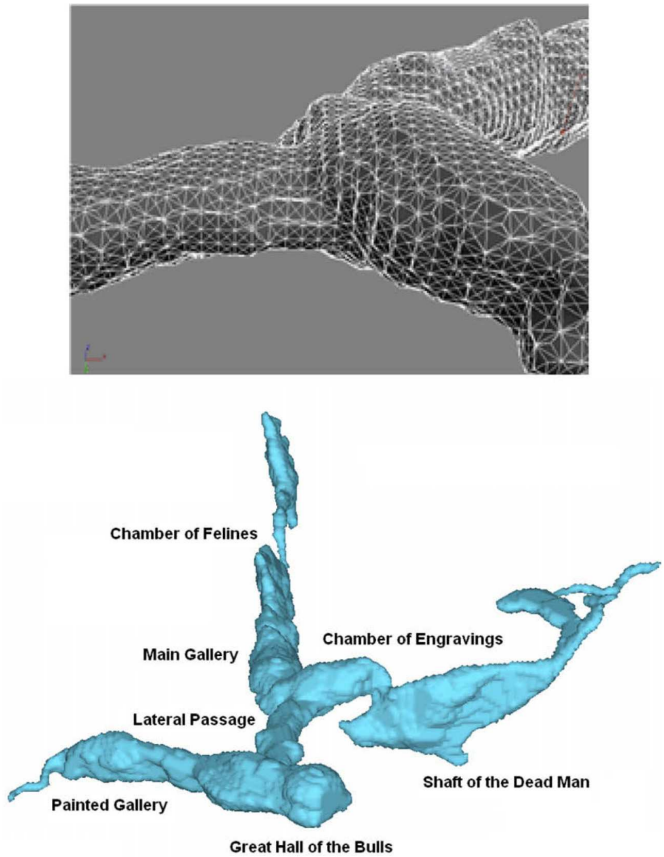


Fig. 11. Detail of the triangularized surface on the Lagrangian grid (top), topology of the Lascaux cave as considered in the simulation – several reference marks (bottom).

Table 1
Comparison between theoretical V_0 and simulated V values of velocities in the boundary layer of a differentially heated square cavity.

Ra	10^2	10^3	10^4	10^5	10^6	10^7	10^8
V_0 (m/s)	1.8610^{-2}	2.7410^{-2}	4.0210^{-2}	5.910^{-2}	8.610^{-2}	1.2710^{-1}	1.8710^{-1}
V (m/s)	8.710^{-4}	3.810^{-3}	9.410^{-3}	1.510^{-2}	2.210^{-2}	3.310^{-2}	4.910^{-2}

Table 2
Comparison between reference Nu_{ref} and our simulated values Nu of Nusselt numbers in the boundary layer of a differentially heated square cavity.

Ra	10^2	10^3	10^4	10^5	10^6	10^7	10^8
Nu_{ref} [30]				4.521	8.8252	16.523	30.225
Nu	1.0015	1.1178	2.2448	4.5217	8.8252	16.523	30.225

years above the cave, and those taken in the cave since 1963. A computerized system using a remote metering has been set up in the cave to record the variations in temperature, hygrometry and carbon dioxide gas pressure.

Two climatic configurations are chosen, corresponding to two different periods, September 1981, during which the cavity remains in a stable state, and December 1999, before the work of replacement of the air treatment machine [33]. Profiles of temperature are given in Fig. 12 as a function of depth. These two periods are representative of two very different configurations. September 1981 represents the typical behavior of the 1980s, while December 1999 corresponds to the 1990s and early 2000s, whatever is the season in the year.

Fig. 13 shows the different distribution of temperature depending on the climatic configuration along the Y direction.

In September 1981, the slope of temperature is positive, inside the cave the vaults are colder than the cave floor. In December 1999, the slope is negative, the cave floor being colder than the vaults.

Once the temperature gradient is introduced in the calculation domain (including the cave and the surrounding rock) the flow induced by natural convection is established in the cavity.

4.1.3. Physical characteristics

The humidity is initialized to a value of 98% of relative moisture content in the whole cave. The calculations are made on the absolute moisture content, related to the temperature at each point. The diffusion coefficient of the moisture in air is $D = 2.18 \times 10^{-5}$ m/s.

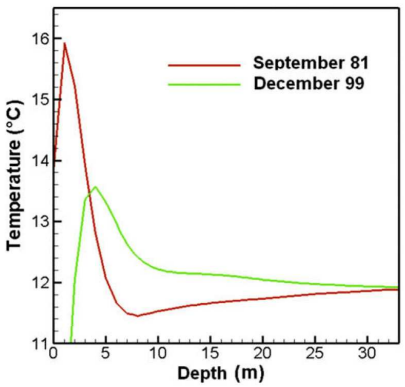


Fig. 12. Temperatures following a one-dimensional heat conduction model in the rock as a function of depth.

The characteristics of the rock and air are the following: $\rho_s = 1800$ kg/m^3, $C_{Ps} = 1000$ J/kg K, $\lambda_s = 1$ W/m K, $\rho_a = 1.1768$ kg/m^3, $\mu_a = 1.85 \times 10^{-5}$ Pa s, $C_{Pa} = 1006$ J/kg K, $\lambda_a = 0.0263$ W/m K.

4.2. Results and analysis

The simulation is dedicated here to measure the impact of the local outside climate change on the management of the climate in the cave. Between 1965 and 1981, the temperature gradient in the cave met the requirements of the operational plan as set out by the Scientific Commission. In winter, the air temperature increases from the surface to the lower areas of the cave, whereas in summer this order is maintained artificially by deliberately lowering the air in the Machine Room, located before the Great Hall of the Bulls. Since 1981, temperature distribution changed, temperatures in the lower parts of the cave became lower than the mean surface temperatures (the thermal inertia of the ground increases as its thickness increases). The natural temperature gradient is inverted. This phenomenon is independent of the artificial control system of the cave and is related only to the outside weather pattern.

The temperature distribution in the cave is homogeneous in the thermal configuration of September 1981, as it can be seen in Fig. 14, where are exhibited zones of convective currents. The mean velocity values are approximately 10^{-2} m/s. Whereas in December 1999, in Fig. 14, the temperature is stratified, and no major convection current is noticed, due to the inversion of the temperature gradient between the two dates (Fig. 13). In this case, velocities are 100 times lower, around 10^{-4} m/s.

One of the major problem concerning the conservation of the Lascaux cave is its evolving state. A porous rock in equilibrium with a humid atmosphere is classically more or less saturated with water. Under certain conditions, the water vapor contained in the air present in the network of rock pores can condense. This liquid water is fixed by capillary action in the smallest pores, the pores for which the radius is less than a function of the relative moisture of the air. The condensed water contained in the pores is aggressive and a chemical equilibrium is reached by dissolution of carbonate minerals in the rock. Once saturated, this water remains inert as long as the atmospheric pressure, the temperature and the partial pressure of carbon dioxide is unchanged. When this equilibrium is broken, a drop would evaporate and this process would then result in precipitation of calcite. The condensation–evaporation cycle can occur repeatedly and lead to a loss of carbonates from the porous matrix, causing a major issue of conservation. These observations have demonstrated the need to avoid creating conditions in which rock dissolution and calcite deposition are promoted and to strive to maintain the most stable possible air conditions, while taking into account the natural rhythms of the cave.

The aim of the simulation is to give information about the precise location of the condensation risk zones. The case presented here corresponds to the configuration without anthropogenic effects. It is meant to serve as a basis for further studies, as the introduction of a machinery, human presence, and hot and cold points.

The moisture content distribution on the walls of the cave for the two climatic conditions described before is given in Fig. 16. Its distribution in a view of the right gallery, from the lateral Passage towards the Chamber of Felines for the two previous climatic

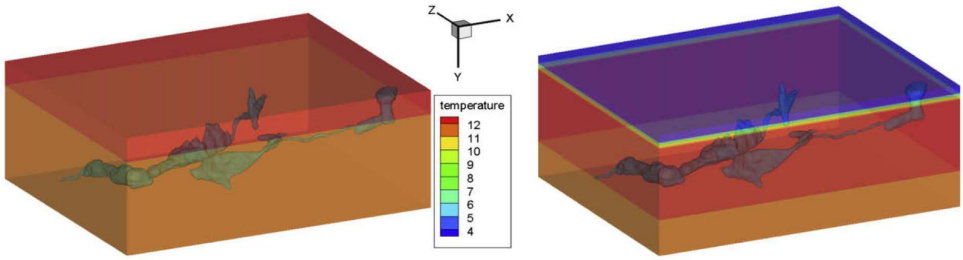


Fig. 13. Visualization of the temperature profiles for both climatic configurations, September 1981 (a) and December 1999 (b).

configurations is given in Fig. 17. In the case of September 1981 (Figs. 16 and 17) the absolute moisture content is higher in the vaults than on the floor, whereas in December 1999 (Figs. 16 and 17) the moisture is concentrated in the floor. This accurate description allows to know the places where the condensation risk is higher, before any introduction of external disturbance. The slice concerning the right gallery shows the spatial evolution of the absolute moisture content depending on the thermal configuration. In 1981, there are homogeneous zones, while in 1999, we found layers of different absolute moisture contents, due to the inversion of temperature.

Moreover, the climatic configuration of December 1999 corresponds to a higher global absolute moisture content than the one of September 1981.

The distribution of temperature and their values directly influence the moisture content field. In September 1981, the vaults are colder than the floor, inducing a concentration of moisture in the vaults with a higher value. In December 1999, the distribution of temperature is reversed, thus the distribution of moisture is also reversed, following the value of temperature. Nevertheless, inside the cave, concentration lines do not follow iso-temperatures, as it can be observed in Fig. 18. The moisture content is transported by the air.

Furthermore, the inversion of temperature implies a drastic modification of natural convection, and of the intensity of the velocities. Indeed, in September 1981, the air flows from the floor to the vaults increase the moisture concentration in this zone, whereas in December 1999 the very low velocities lead to a stagnation of the layers of moisture, on the floor of the cave.

The convective currents are closer studied in Fig. 15. A slice taken from the right part of the cave, in the Main Gallery, presents the normalized velocity and several streamlines. The velocity is higher on the walls, and distinct convective currents can be observed. A different view of the situation is also presented in this figure, virtual particles are released in the right part of the cave, and their trajectory is related to the temperature by the color of the ribbon. The natural convection occurs both from the ground to the vaults and from the Great Hall of the Bulls to the end of the Main Gallery. The resulting currents are complex and fully three-dimensional, with a general convection from the lateral Chamber of Felines towards the Lateral Passage, and several smaller ones, isolated and corresponding to a convection going from the ground to the vaults.

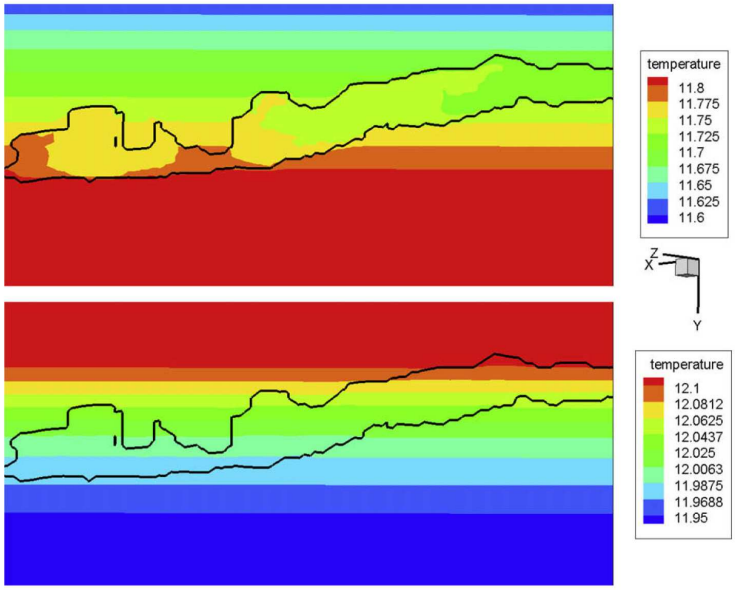


Fig. 14. Temperature distribution on a view of the right gallery, from the lateral Passage towards the Chamber of Felines for the climatic configuration of September 1981 in which $V \approx 10^{-2}$ m/s (top) and December 1999 in which $V \approx 10^{-4}$ m/s (bottom).

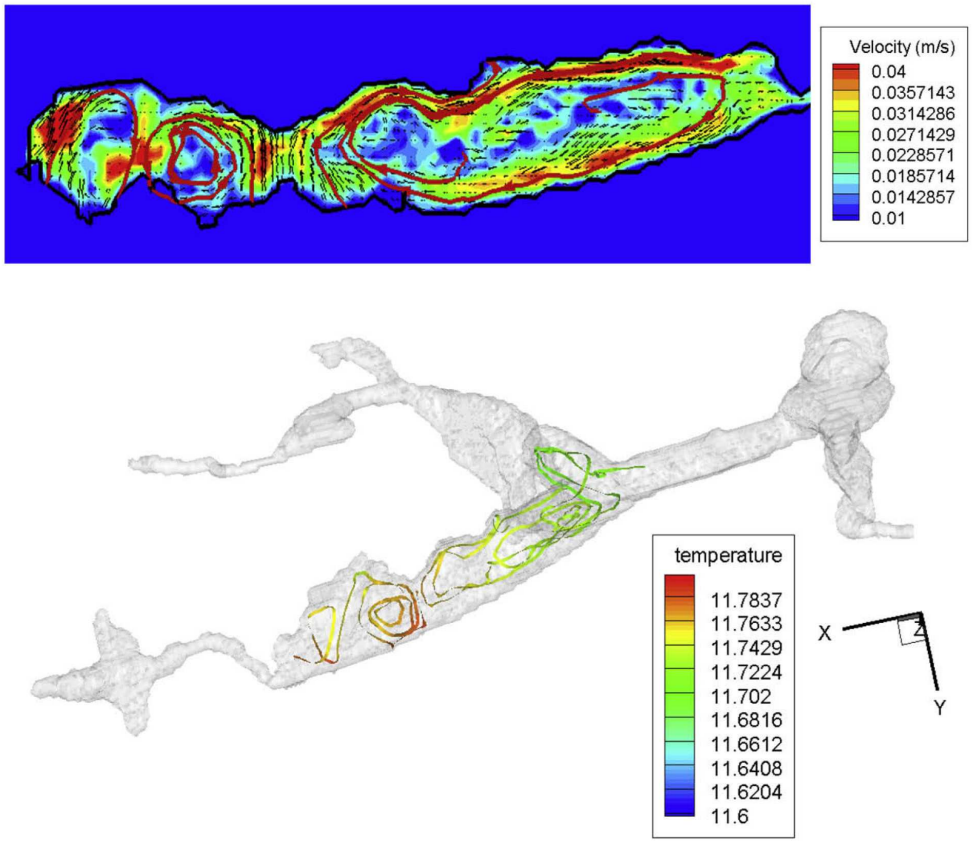


Fig. 15. Velocity distribution on the right gallery of Lascaux cave for the climatic configuration of September 1981, on a slice with the convective currents in red (top) and on a general view with the trajectory of a virtual particle on a ribbon colored with the temperature of the air crossed (bottom).

Finally, these numerical results have been validated by observations in the cave at the two climatic periods. In 1981, no major problem was noticed, whereas in the late 1990s and early 2000s, a spread of micro-organisms caused conservation issues. It can be assumed that the consequences of the increase and reverse of temperatures, i.e. drastic decrease of velocities and increase of

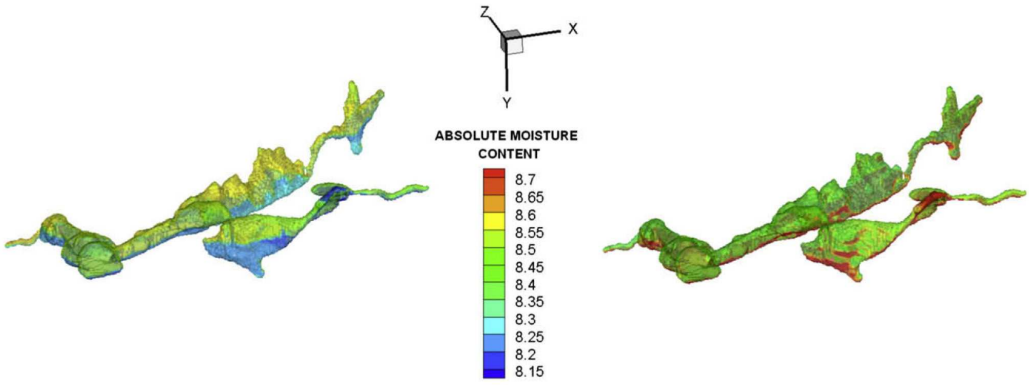


Fig. 16. Absolute moisture content on the walls of the cave for the climatic configuration of September 1981 (top) and December 1999 (bottom).

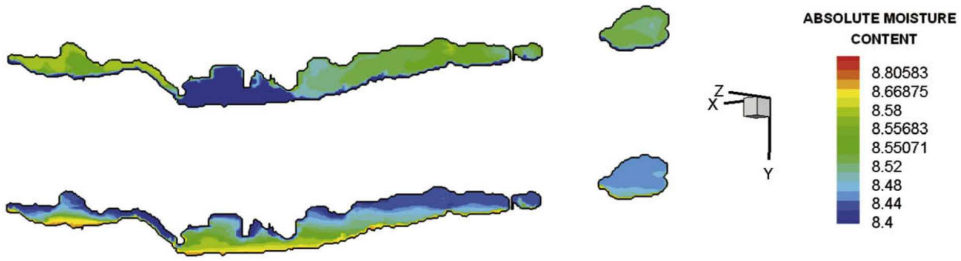


Fig. 17. Absolute moisture content on a view of the right gallery, from the lateral Passage towards the Chamber of Felines for the climatic configuration of September 1981 (top) and December 1999 (bottom).

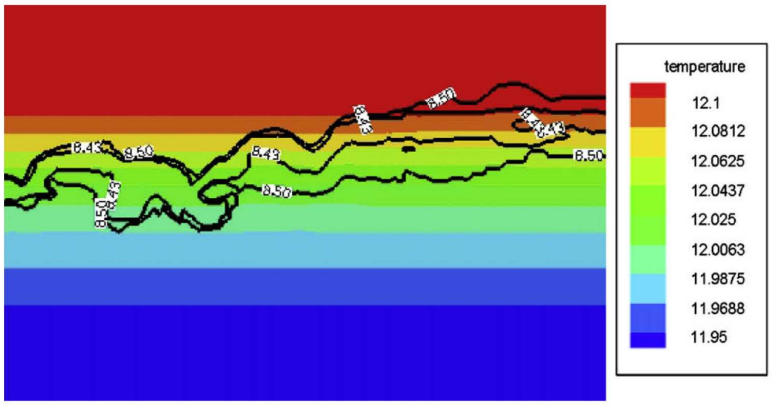


Fig. 18. Superposition of temperature and concentration distribution on a view of the right gallery, from the lateral Passage towards the Chamber of Felines for the climatic configuration of December 1999.

absolute moisture content, implied a stagnation of the air among the cavity and favored the development of micro-organisms.

5. Conclusions

An Eulerian/Lagrangian method for the numerical simulation of incompressible convection flows interacting with complex obstacles has been successfully validated on several natural convection cases, then applied here to the conservation of the Lascaux cave.

It has been shown in this article that a fictitious domain approach method coupled to a Lagrangian grid of obstacles allowed the correct description of the interaction between the natural convection flows and these obstacles. For example, a cavity filled with a large amount of cylinders shows the thermal comportment of a porous medium. Compared to experiments of natural convection (Sierpinski carpet) interacting with obstacles set according to fractal patterns, experimental measurements are found in good agreement with the penalization method with a precision lower than 20%. It can be due to the fact that the numerical boundary limits are slightly different than the experimental ones. Moreover, it has been pointed out that the two dimension hypothesis was not entirely valid and that three dimension simulations brought better results.

Concerning the application to the Lascaux cave, the article provided the first simulations of the entire geometry of the cavity, with a fictitious domain approach method. The results are confirmed by the observations made in the cave: it is more confined in the present thermal configuration than in the 1980s. The climate change made the cave more sensitive to disturbances. For example, the influence of humans entering the cave will be more devastating for the paintings in the present configuration than before.

Numerical perspectives are numerous. Higher order penalization will be implemented, by a technology currently under development, in order to better take into account the complex geometry of the objects at a scale lower than the grid. Our purpose is also to integrate moving obstacles in our simulations in order to take into account the impact of the moving of a human visiting the cave for example.

Acknowledgements

Electricité de France (EDF) within the framework of the scientific and technological sponsorship policy has provided the Ministry for the Arts the competence and the means of its own teams in partnership with the University of Bordeaux 1 and the Centre National de la Recherche Scientifique (CNRS) to understand and foresee the thermal and aeraulic effects inside the Lascaux cave. The authors also wish to thank the IDRIS and CINES (No. TER2237-2006) for their computer support.

References

[1] P. Chow, M. Cross, K. Pericleous, A natural extension of the conventional finite volume method into polygonal unstructured meshes for CFD application, Appl. Math. Model. 20 (2) (1996) 179–183.

[2] S.R. Mathur, J.Y. Murthy, A pressure-based method for unstructured meshes, Numer. Heat Transfer 31 (2) (1997) 195–215.
[3] K. Khadra, P. Angot, S. Parneix, J.P. Caltagirone, Fictitious domain approach for numerical modelling of Navier–Stokes equations, Int. J. Numer. Methods Fluids 34 (2000) 651–684.
[4] T.N. Randrianarivelo, G. Pianet, S. Vincent, J.P. Caltagirone, Numerical modelling of solid particle motion using a new penalty method, Int. J. Numer. Methods Fluids 47 (2005) 1245–1251.
[5] C. Peskin, The immersed boundary method, Acta Numer. 11 (2000) 479–517.
[6] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flows 25 (1999) 755–794.
[7] X. Zhong, A new high-order immersed interface method for solving elliptic equations with imbedded interface of discontinuity, J. Comput. Phys. 255 (2007) 1066–1099.
[8] A. Sarthou, S. Vincent, J.P. Caltagirone, P. Angot, Eulerian/Lagrangian grid coupling and penalty methods for the simulation of multiphase flows interacting with complex objects, Int. J. Numer. Methods Fluids 56 (8) (2008) 1093–1099.
[9] I. Ramiere, P. Angot, M. Belliard, A general fictitious domain method with immersed jumps and multilevel nested structured meshes, J. Comput. Phys. 225 (2007) 1347–1387.
[10] J. Brunet, P. Malaurent, J. Vouvé, Lascaux, histoire d'un difficile sauvetage, Archéologia 332 (1997) 24–35.
[11] M.A. Sire, Des restaurateurs au chevet des peintures de Lascaux, De l'élimination des champignons au constat d'tat, Les grottes ornées Monumental 2006 Monum Editions du Patrimoine 2 (2006) 68–75.
[12] G. Orial, J.D. Mertz, Lascaux: une grotte vivante, Étude et suivi des phénomènes microbiologiques, Les grottes ornées Monumental 2006 Monum Editions du Patrimoine 2 (2006) 76–78.
[13] P. Malaurent, R. Lastennet, J. Brunet, Une grotte sous influence: l'environnement hydrogéologique et climatique de la grotte de Lascaux Les grottes ornées Monumental 2006 Monum Editions du Patrimoine 2 (2006) 88–93.
[14] C. Ferchal, Modélisation des écoulements et des transferts de masse et de chaleur dans la grotte de Lascaux, Ph.D. Thesis, University of Bordeaux, Talence, France, 2003.
[15] C. Ferchal, J.-B. Ritz, J.P. Caltagirone, Ph. Malaurent, Simulation des écoulements et transferts de masse et de chaleur dans la Grotte de Lascaux, Congrès de la Société Francaise de Thermique, Grenoble, 3–6 juin, 2003.
[16] D. Lacanette, P. Malaurent, J.P. Caltagirone, S. Vincent, A model of thermal and aeraulic flows in the cave of Lascaux, International Association for Mathematical Geology XIth International Congress Université de Liège – Belgium, 2006.
[17] D. Lacanette, J.P. Caltagirone, Le simulateur Lascaux: un outil d'aide à la décision pour l'avenir de la préhistoire, Les grottes ornées Monumental 2006 Monum Editions du Patrimoine 2 (2006) 94–97.
[18] E. Arquis, J.-P. Caltagirone, On hydrodynamic conditions near a fluid-porous interface: application to the natural convection, Comptes Rendus de l'Académie des Sciences Série II b Mécanique 299 (1984) 1–4.
[19] S. Shin, D. Juric, Modeling three-dimensional multiphase flow using a level-contour reconstruction method for front tracking without connectivity, J. Comput. Phys. 180 (2002) 427–470.
[20] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation, New York, USA, 1980.
[21] F.H. Harlow, J.E. Welsh, Numerical calculation of time-dependent viscous incompressible flow with free surface, Phys. Fluids 8 (1965) 2182–2189.
[22] M. Fortin, R. Glowinski, Méthodes de lagrangien augmenté, Application à la résolution numérique de problèmes aux limites, Dunod, 1982.
[23] I. Gustafsson, On first and second order symmetric factorization methods for the solution of elliptic difference equations, Technical Report, Chalmers University of Technology, 1978.
[24] H.A. Van Der Vorst, Bi-cgstab: a fast and smoothly converging variant of bi-cg for the solution of non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (1992) 631–644.
[25] S. Vincent, J.P. Caltagirone, P. Lubin, T.N. Randrianarivelo, An adaptative augmented Lagrangian method for three-dimensional multimaterial flows, Comput. Fluids 33 (2004) 1273–1289.
[26] S. Vincent, T.N. Randrianarivelo, G. Pianet, J.P. Caltagirone, Local penalty methods for flows interacting with moving solids at high Reynolds numbers, Comput. Fluids 36 (2007) 902–913.
[27] S. Vincent, J.P. Caltagirone, A one cell local multigrid method for solving unsteady incompressible multi-phase flows, J. Comput. Phys. 163 (2000) 172–215.
[28] E. Arquis, Transferts en milieu poreux et à l'interface: de l'échelle microscopique à l'échelle macroscopique, Ph.D. Thesis, University of Bordeaux, Talence, France, 1994.
[29] A. Amine, J.K. Platten, M. Hasnaoui, Thermal convection around obstacles: the case of Sierpinski carpets, Exp. Fluids 36 (2004) 717–727.
[30] P. Le Quéré, Accurate solutions to the square thermally driven cavity at high Rayleigh numbers, Comput. Fluids 20 (1991) 29–41.
[31] J. Taine, J.P. Petit, Transferts thermiques: Mécanique des fluides anisothermes, Dunod, 1989.
[32] Website: www.meteo.fr.
[33] P. Malaurent, J. Brunet, D. Lacanette, J.P. Caltagirone, Contribution of numerical modelling of environmental parameters to the conservation of prehistoric cave paintings: the example of Lascaux Cave, Conserv. Manage. Archaeol. Sites 8 (2006) 1–11.

14.3 Proceeding of the Société Française de Thermique 2009 (in French)

Pénalisation sur des maillages curvilignes pour la simulation des transferts de masse et de chaleur dans la grotte de Lascaux.

Delphine LACANETTE[1*], Stéphane VINCENT[1], Arthur SARTHOU[1], Jean-Paul CALTAGIRONE[1], Philippe MALAURENT[2]

[1]Laboratoire Transferts, Ecoulements, Fluides, Energétique (TREFLE)
16 avenue Pey-Berland – 33607 Pessac Cedex
[2]Laboratoire Géosciences Hydrosciences Matériaux Constructions (GHYMAC)
Avenue des Facultés – 33405 Talence Cedex
[*](auteur correspondant : lacanette@enscpb.fr)

Résumé – Des techniques de pénalisation sur maillage curviligne sont présentées et validées sur un cas de convection naturelle bidimensionnelle en interaction avec des obstacles. Ces techniques sont utilisées pour simuler les transferts de masse et de chaleur dans la grotte de Lascaux, qui constitue un obstacle de forme complexe. L'outil de simulation permet de différencier des scenarios climatiques avec pour objectif l'aide à la décision dans le cadre de la conservation du patrimoine.

Nomenclature

C	fonction de présence
C_P	chaleur spécifique, $J.kg^{-1}.K$
$\mathbf{g}$	accélération de la gravité, $m.s^{-2}$
K	perméabilité, m^2
p	pression, Pa
t	temps, s
T	température, K
$\mathbf{u}$	vitesse, $m.s^{-1}$

Symboles grec

β	*coefficient de dilatation,* K^{-1}
λ	conductivité thermique, $W.m^{-1}.K$
μ	viscosité dynamique, $Pa.s$
ρ	masse volumique, $kg.m^{-3}$
Σ_h	maillage des objets

Indices et exposants

0	état thermodynamique de référence

1. Introduction

La réalisation d'un modèle numérique pour traiter de la convection naturelle dans les cavités et de la conduction dans la roche environnante est importante pour la conservation des milieux confinés. Pour être pertinent, l'outil numérique doit prendre en compte la géométrie souvent complexe de la grotte avec suffisamment de précision. Deux approches sont communément employées. La première consiste à considérer deux sous domaines avec leur propre maillage non structuré (body fitted grid) connectés par une condition limite à l'interface entre les deux milieux solide et fluide. L'avantage des méthodes non structurées est leur prise en compte naturelle de la forme complexe des objets, ainsi que leur description explicite à l'interface. Cependant la génération du maillage est complexe voire parfois impossible à cause des fortes irrégularités de l'interface. La seconde méthode numérique pour la gestion de l'interface est l'approche des domaines fictifs. Cette technique est basée sur le concept de l'utilisation d'un maillage structuré pour gérer les équations de conservation (Navier-Stokes, énergie). Les obstacles solides sont projetés sur le maillage structuré de la simulation et des termes spécifiques, dits de pénalisation, sont ajoutés aux équations de conservation pour tenir compte de la présence d'obstacles. Cette méthode possède l'avantage d'être facile à implémenter même en 3D et de pouvoir intégrer des outils CFD existants. En outre, elle peut s'appliquer à des objets mobiles et plusieurs approches ont été étendues à des ordres élevés.

Les travaux proposés ici présentent les premières pénalisations avec l'approche des domaines fictifs sur des maillages curvilignes en 3D, avec pour objectif la simulation des écoulements de convection naturelle dans la grotte de Lascaux. Après une validation sur un cas de convection naturelle sur un tapis de Sierpinski avec un maillage curviligne, l'outil numérique est utilisé afin de comprendre le comportement de la grotte de Lascaux lorsqu'elle est soumise à différentes conditions limites thermiques, avec un objectif d'aide à la conservation de ce patrimoine. L'aspect non intrusif et prédictif de cet outil, appelé Simulateur Lascaux, en fait une technique conservative originale.

2. Modèles et méthodes numériques

La modélisation des écoulements de convection naturelle en interaction avec des obstacles est basée sur les équations de Navier-Stokes Brinkman incompressibles [1] et l'hypothèse de Boussinesq :

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}.\nabla)\mathbf{u}\right) + \frac{\mu}{K}\mathbf{u} = \rho\mathbf{g} - \nabla p + \nabla.\left(\mu(\nabla\mathbf{u} + \nabla^T\mathbf{u})\right) \tag{2}$$

$$\rho C_p\left(\frac{\partial T}{\partial t} + (\mathbf{u}.\nabla)T\right) = \nabla.(\lambda\nabla T) \tag{3}$$

$$\rho = \rho_0\left(1 - \beta\left(T - T_0\right)\right) \tag{4}$$

Dans ce modèle, les obstacles sont vus comme un milieu poreux au travers de la perméabilité K. La propriété de non-déformabilité est imposée par un terme de Darcy qui est ajouté dans l'équation de conservation de la quantité de mouvement : il s'agit ici d'une méthode de domaines fictifs dans laquelle le milieu solide est vu comme un fluide de propriétés spécifiques. Concernant les méthodes numériques, les schémas et les solveurs, le détail est donné dans [2]. L'ensemble est basé sur des volumes finis et des maillages structurés.

La principale difficulté lorsque l'on souhaite traiter de l'interaction d'un fluide avec un obstacle de géométrie non triviale (grotte de Lascaux par exemple) est la définition *a priori* de la perméabilité en fonction de la géométrie de l'obstacle. Pour réaliser ceci, nous nous appuyons sur les techniques écrites dans [2] qui utilisent une fonction de présence C, égale à 1 dans le solide et 0 ailleurs. Connaissant cette fonction C, K est définie par $K=10^{40}$ si $C<0.5$ (totalement perméable), $K=10^{-40}$ sinon.

Dans [2], les auteurs proposent d'utiliser un maillage surfacique lagrangien Σ_h des objets et des méthodes issues des techniques de Immersed Boundary Method (IBM) de [3] pour projeter le maillage surfacique sur la grille de calcul des équations de conservation (1-4). Cette démarche est efficace mais comporte deux désavantages majeurs :

- On résout une équation de diffusion sur C pour obtenir cette fonction de phase en fonction du maillage Σ_h. Cette étape ne peut gérer les configurations où l'enveloppe de l'objet n'est pas totalement incluse dans le maillage de calcul.
- Les techniques IBM sont difficilement transposables sur des maillages de calcul curvilignes orthogonaux car elles requièrent des étapes d'intersection géométriques entre les éléments du maillage surfacique de l'objet Σ_h et le maillage de calcul.

Nous proposons ici de remplacer les deux étapes précédentes afin de pouvoir gérer des obstacles partiellement inclus dans le maillage de calcul quand celui-ci est curviligne. Nous avons basé notre démarche sur l'utilisation d'une méthode de Ray-Casting [4] qui permet de générer automatiquement C connaissant Σ_h.

Le principe est le suivant : on parcourt le maillage de calcul et on détermine l'appartenance de chaque nœud à l'obstacle ou non en lançant un rayon dont la source est située sur le point du maillage. On compte ensuite le nombre d'intersections entre ce rayon et les éléments de Σ_h. Si ce nombre est impair, le point du maillage de calcul est dans l'objet, si le nombre est pair, il est à l'extérieur. Connaissant C, on construit automatiquement K avec les formules introduites précédemment.

La méthode de Ray-Casting est très efficace, elle voit les objets même lorsqu'ils débordent du maillage de calcul. La technique fonctionne sur des maillages structurés ou non. Par contre, l'efficacité de la méthode est grandement améliorée sur des maillages structurés à pas constants car les nœuds du maillage de calcul et le rayon sont directement localisés par les indices I,J,K des cellules et non par leurs coordonnées. Pour continuer à utiliser le Ray-Casting sur des grilles à pas constant cartésiennes, même lorsque le maillage de calcul est curviligne orthogonal, nous avons opéré des transformations de coordonnées. Il existe une bijection f(x,y,z) qui permet de transformer un maillage curviligne orthogonal en un maillage à pas constant. Nous avons utilisé des approximations polynomiales bilinéaires de f pour réaliser les changements de grille. Cette fonction a été utilisée pour projeter les coordonnées de Σ_h d'un repère à l'autre. Ainsi, on obtient un objet d'enveloppe transformée, compatible avec le maillage à pas constant, lui-même étant la transformée du maillage curviligne orthogonal initial. On utilise ensuite le Ray-Casting sur le maillage à pas constant pour définir C. Il se trouve que les sommets des cellules de ce maillage gardent leurs valeurs lorsque l'on revient sur le maillage curviligne, il n'est donc pas utile d'utiliser f^{-1}.

3. Validation : le tapis de Sierpinski

Le cas dit du tapis de Sierpinski est un cas de convection naturelle dans une boîte contenant des obstacles carrés. La disposition des obstacles suit le motif fractal du tapis de Sierpinski dont quelques itérations sont présentées figures 1, 2 et 3.

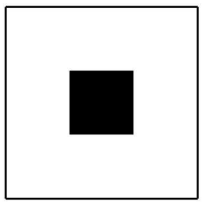


Figure 1 : Première itération du tapis de Sierpinski

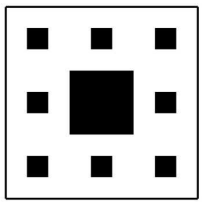


Figure 2 : Deuxième itération du tapis de Sierpinski

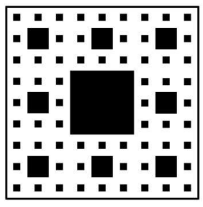


Figure 3 : Troisième itération du tapis de Sierpinski

Un différentiel de température de 5 degrés est imposé aux parois gauche et droite. Les parois supérieures et inférieures sont considérées comme adiabatiques. Le fluide est une huile aux propriétés suivantes :

$$\mu = 0.0815 kg \cdot m^{-1} s, \rho = 857 kg \cdot m^{-3}, C_p = 1880 J \cdot kg^{-1} K, \lambda = 0.132 W \cdot m^{-1} K$$

Les résultats obtenus avec notre approche sur un maillage curviligne sont comparés avec ceux obtenus avec le même code sur un maillage cartésien, ainsi qu'avec les résultats expérimentaux de Amine et al. [5]. Seule la seconde itération du motif est traitée. La boite de simulation est un carré de 100 mm de côté. L'obstacle central a un côté de 33 mm et les

obstacles périphériques un côté de 11 mm. La figure 4 montre le type de maillage curviligne utilisé. Afin d'obtenir un maillage curviligne non trivial, les côtés de la boite sont étirés et la nouvelle zone ainsi créée est pénalisée afin de retrouver un domaine de calcul carré.

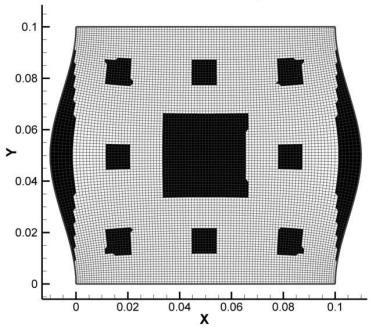


Figure 4 : Exemple de maillage curviligne pour le cas du tapis de Sierpinski. Les zones pénalisées sont coloriées en noir

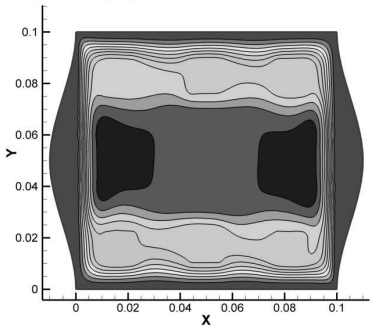


Figure 5 : Lignes de courant pour l'écoulement stationnaire

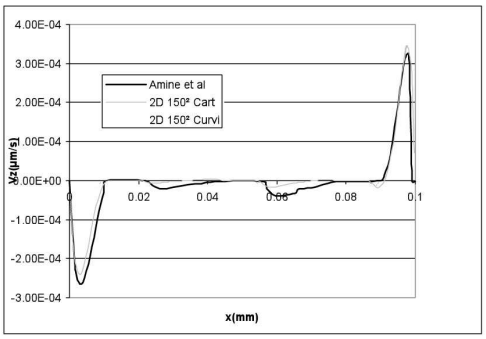


Figure 6 : Comparaison des résultats expérimentaux d'Amine et al et des résultats numériques sur maillages cartésiens et curvilignes Vz pour z=17mm

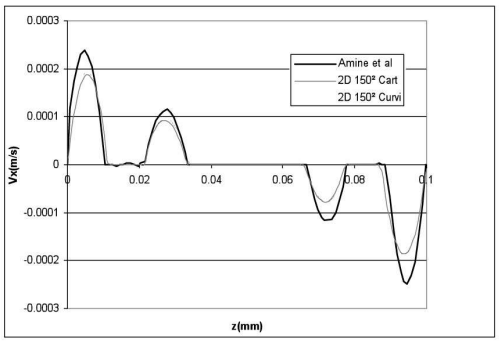


Figure 7 : Comparaison des résultats expérimentaux d'Amine et al et des résultats numériques sur maillages cartésiens et curvilignes Vx pour x=50mm

Les résultats numériques obtenus pour des maillages 150 x 150 sont présentés figures 5, 6 et 7. Les valeurs observées sont les vitesses Vz(x) pour z=17mm et Vx(z) pour x=50mm. On observe une corrélation assez moyenne entre les résultats numériques et les résultats expérimentaux. Toutefois, les auteurs exposent dans [2] de bien meilleurs résultats en menant un calcul 3D. Quoiqu'il en soit, les résultats sur maillage curviligne sont très proches des résultats obtenus avec un maillage cartésien.

4. Application à la convection naturelle dans la grotte de Lascaux

La méthode présentée dans le paragraphe 2. est utilisée pour prendre en compte la géométrie complexe de la grotte de Lascaux, et résoudre les écoulements de convection naturelle s'y tenant.

4.1. Position du problème

La méthode de pénalisation est utilisée sur un maillage curviligne, la grotte de Lascaux est ainsi présentée dans le massif environnant. L'utilisation d'un maillage curviligne présente plusieurs intérêts, celui d'économiser le nombre de points en ne résolvant pas des zones dans lesquelles seul le massif est présent, il permet en outre de se rapprocher de la géométrie réelle du massif.

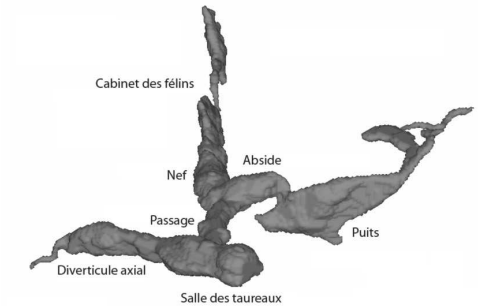


Figure 8 : Grotte de Lascaux telle qu'elle est prise en compte dans le simulateur

Le nombre de Rayleigh relatif à l'écoulement dans la grotte de Lascaux est de 10^8. Ce Rayleigh est transitionnel entre un écoulement laminaire et turbulent. Les gradients de température sont de 0,1°C. Ce nombre de Rayleigh élevé est lié aux grandes dimensions rencontrées dans la cavité, jusqu'à 10 m de hauteur.

4.2. Phénomène d'inversion des températures

L'un des intérêts du simulateur Lascaux réside dans son aspect prédictif des écoulements. Il permet de se situer à différentes époques en fonction de la configuration thermique correspondante. Les températures sont relevées au-dessus de la colline environnant la grotte depuis les années 1940, ce qui permet de déduire les températures dans la profondeur de la colline et dans la grotte par une loi de conduction. Ainsi on se place à différentes époques afin de retrouver les écoulements correspondants. Les figures 9 et 10 montrent les distributions de températures à deux époques bien distinctes, septembre 1981 et février 2008, sur une coupe longitudinale dans la nef (figure 8). Ces deux époques sont caractéristiques de deux régimes très différents, en septembre 1981 la température augmente lorsqu'on s'enfonce dans les profondeurs, tandis qu'en février 2008, c'est le contraire.

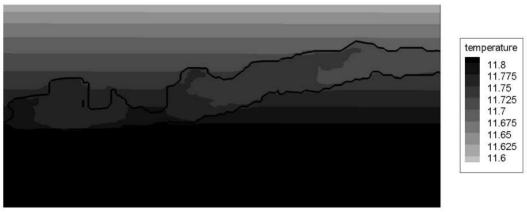


Figure 9 : Distribution des températures dans la configuration thermique de septembre 1981

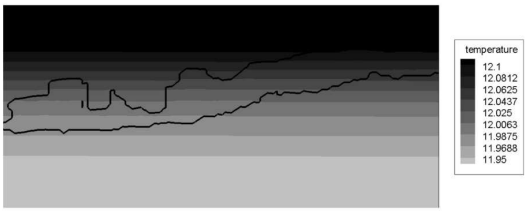


Figure 10 : Distribution des températures dans la configuration thermique de février 2008

Les gammes de vitesse sont différentes selon la configuration thermique. En septembre 1981, elles étaient de l'ordre du cm/s, tandis qu'en février 2008, elles sont 10 fois plus faibles, comme représenté sur les figures 11 et 12 sur une coupe longitudinale dans la nef.

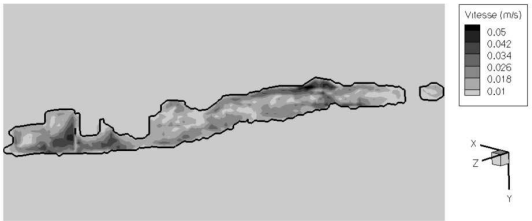


Figure 11 : Distribution des vitesses dans la configuration thermique de septembre 1981

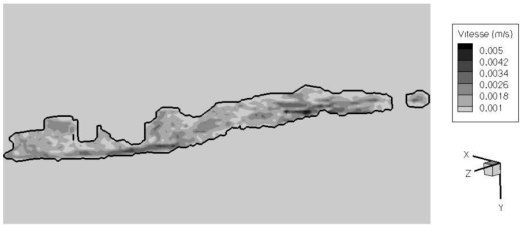


Figure 12 : Distribution des vitesses dans la configuration thermique de février 2008

La trajectoire d'un traceur dans la partie droite de la grotte de Lascaux pour la configuration climatique de septembre 1981 est présentée à la figure 13, elle montre l'aspect tridimensionnel de l'écoulement de convection naturelle.

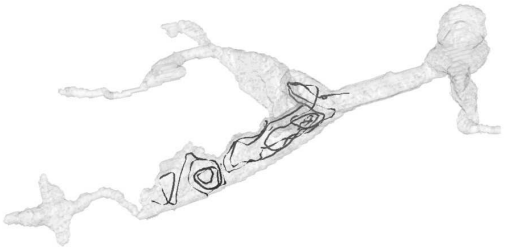


Figure 13 : Trajectoire d'un traceur dans la configuration thermique de septembre 1981

5. Conclusion

Les méthodes de domaines fictifs présentées dans [2] sont étendues pour la première fois aux maillages curvilignes grâce à des algorithmes de Ray-Casting. Une validation 2D a démontré le potentiel de la méthode sur un écoulement de convection naturelle interagissant avec un tapis de Sierpinski. Des simulations 3D ont montré l'intérêt de la méthodologie pour étudier les écoulements dans des milieux à géométrie complexe comme la grotte de Lascaux. Ces simulations s'inscrivent dans le projet de conservation préventive innovante qu'est le simulateur Lascaux [6].

Références

[1] K. Khadra, P. Angot, S. Parneix, J.P. Caltagirone, Fictitious domain approach for numerical modelling of Navier-Stokes equations, *Int. J. Num. Meth. in Fluids*, 34 (2000), 651-684.

[2] D. Lacanette, S. Vincent, A. Sarthou, P. Malaurent, J.P. Caltagirone, An Eulerian/Lagrangian method for the numerical simulation of incompressible convection flows interacting with complex obstacles: application to the natural convection in the Lascaux cave, *Int. J. Heat Mass Transfer*, *under press, 2009.*

[3] C. S. Peskin, The immersed boundary method, *Acta Numerica*, (2002), 1-39.

[4] C. J. Ogayar, R. J. Segura, F. R. Feito, Point in solid strategies, *Comp. Graph.*, 29 (2005), 616-624.

[5] A. Amine, J. K. Platten, M. Hasnaoui, Thermal convection around obstacles: the case of Sierpinski carpets, *Exp. In Fluids,* 36 (2004), 717-727.

[6] D. Lacanette, P. Malaurent, J.P. Caltagirone, J. Brunet, Étude des transferts de masse et de chaleur dans la grotte de Lascaux : Le suivi climatique et le simulateur, *Karstologia*, 50 (2007), 19-30.

Remerciements

Les auteurs remercient la DRAC Aquitaine pour son soutien financier. Ils remercient également la région Aquitaine, pour son support financier destiné à l'acquisition d'un cluster de 256 processeurs, situé au laboratoire TREFLE.

Discussion and conclusion of Part V

Complex simulations have been realized with our methodology. The simulation in the drill bits would have required more validations, but there is no accurate experimental results to compare with. However, it is generally know that a greater blade angles induces a less efficient evacuation of the coppers. Our parametric study with five different drill bits shows the same tendency. Concerning the simulation of the hydroplaning of tires, the classification obtained by the experiments of Michelin has been retrieved.

Hence, our methodology has been successfully on realistic simulation cases. The adaptation of our home-made code Thétis is currently used on personal computers by Michelin and Varel for their study.

Some illustrations are now quickly presented. Many of them have been realized in collaboration with members of our laboratory. In fact, another difficulty when implementing a new method is to make it usable by almost anyone.

Thermal cooling with a jet (with Ludovic Osmar)
The impact of an oil jet injected through a nozzle on a heating square of copper is simulated. The injection speed is $1m.s^{-1}$. The domain size is $32mm \times 24mm$ and the quare has a side of $8mm$. The properties of the oil are $\rho = 864.1kg.m^{-2}$ and $\mu = 0.1Pa.s$ while rest of the domain is filled by air. The mesh size is 240×180. The oil phase is managed with a VOF-PLIC method. The Fig. (14.2) shows the phase location and the temperature in the domain for a long time simulation. The flow is almost stationary. The SMP method is used to impose a Dirichlet

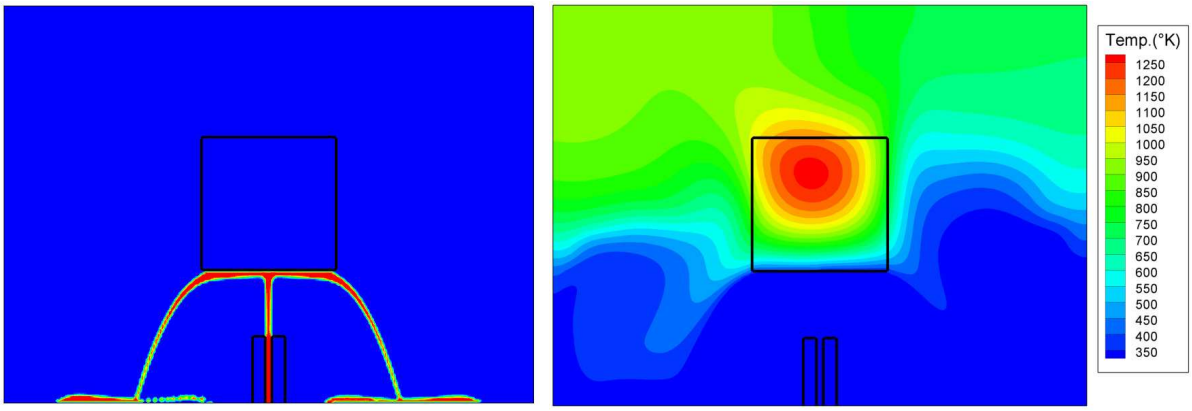


Figure 14.2: Position of the oil jet (left) and temperature field (right)

condition for the fluid flow on the solids. The AIIB method imposes the interface conditions on the surface of the cube. This case emphasize the need of an interface method for thermal transfer while many cases can be treated with only a boundary method for fluid flows. As can be seen, the maximum temperature is obtained near the middle of the heating square. The temperature is lower near the lowest side thanks to the cooling effect of the oil jet.

Subaquatic harrow (with Pierre Lubin)
The aim of these simulations is to study the impact of subaquatic harrows on the sedimentary transport. Fig. (14.3) shows the effect of a single harrow (of height $h = 1.80m$) on an oceanic flow. The recirculation created by the presence of the obstacle generates a dead zone and slows down the displacement of sediment. These images have been created with a computer graphics software from the results of the simulation.

Other illustrations

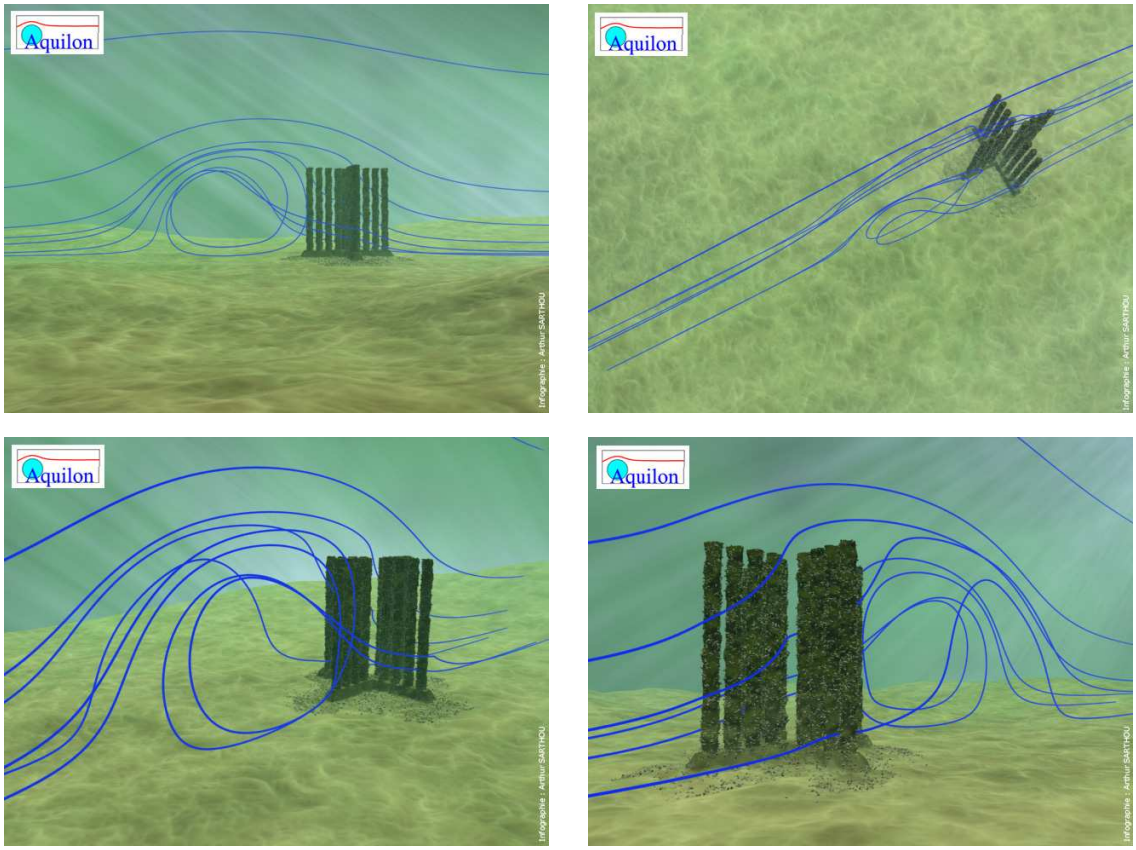


Figure 14.3: Streamlines of the oceanic flow around a subaquatic harrow

Some type cases are presented. The Fig. (14.4) show a simulation of the interaction of a ship hull and the water. The Fig. (14.5) is a case of thermal diffusion in a brain-like cavity.

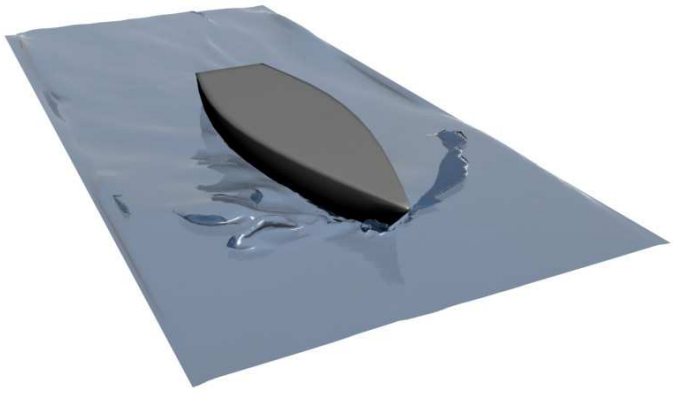


Figure 14.4: Preliminary simulation of fluid flows around a ship hull

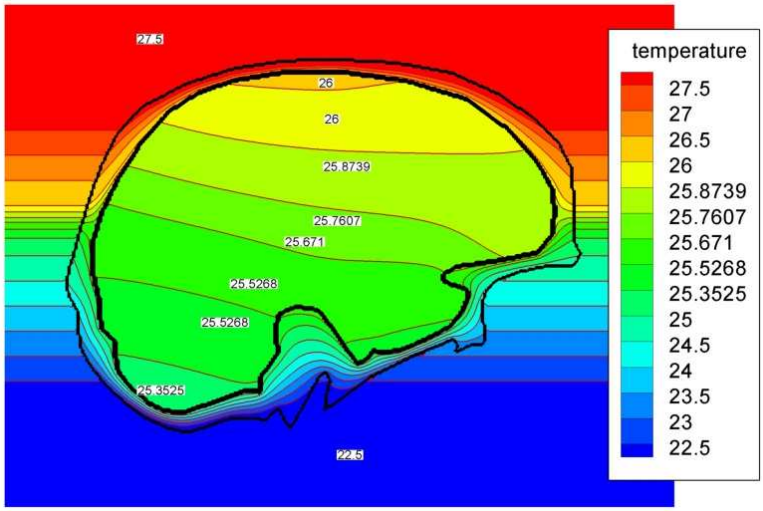


Figure 14.5: Thermal diffusion in a brain-like cavity

Part VII

Conclusion générale et perspectives

Au début de ce travail, le code Aquilon, devenu par la suite Thétis, permettait une utilisation limitée d'obstacles ou d'interfaces. Seuls des objets de formes simples et analytiques pouvait être utilisés, et avec une précision spatiale au premier ordre. Concernant les objets mobiles, un couplage fluide-structure efficace (la méthode ITP) existait déjà mais ne fonctionnait que pour des particules cylindriques en 2D et sphériques en 3D.

Par le présent travail, nous avons permis l'initialisation de phases, d'interfaces ou encore d'obstacles de formes quelconques à l'aide d'algorithmes robustes et rapides. La précision spatiale du traitement de ces éléments a été dans de nombreux cas portée à l'ordre deux. La gestion des objets mobiles a de plus été étendue aux formes quelconques.

Afin de mettre en valeur ces nouvelles possibilités, le code Thétis a été couplé à un logiciel d'image de synthèse professionnel permettant un rendu bien plus réaliste des simulations. Ce dernier est d'ailleurs couramment utilisé au laboratoire pour créer ou pré-traiter le maillage des objets et interfaces.

L'objectif premier de cette thèse était de développer une méthode de pénalisation au second ordre et son extension aux objets mobiles. Cette nouvelle méthode a été validée dans de nombreuses configurations et a montré sa robustesse. Elle reste d'après nos connaissances, la seule méthode d'ordre élevée a avoir été couplée au lagrangien augmenté. Nous avons d'ailleurs exposé les facilités qu'offre cette approche par rapport à une méthode de projection scalaire de pression. Le principal défaut de cette méthode de pénalisation est qu'elle exige des solveurs performants traitant des matrices à diagonales non-dominantes. Concernant son implémentation dans *Code_ Saturne*, les résultats en maillage structuré sont similaires à ceux obtenus avec le code Thétis. En non-structuré, les méthodes de correction d'opérateurs posent toujours problème.

Afin d'obtenir un couplage fluide-structure implicite, il était initialement prévu de coupler la méthode avec l'ITPM ([Rand 05]). Il s'est révélé impossible d'étendre la pénalisation de sous-mailles aux problèmes d'interfaces, ce qui a conduit à la conception d'une méthode de frontière-interface immergée. Cette nouvelle méthode a montré sa capacité a traiter des cas typiques de transferts thermiques avec un ordre deux en espace. Son principal avantage par rapport aux méthode concurrentes est sa formulation simple engendrant une implémentation facile, ainsi que son stencil de discrétisation très réduit. Cette méthode reste toutefois au premier ordre pour certains problèmes elliptiques, notamment pour une condition de flux d'interface ou de frontière inhomogène. Ce point devra rapidement être amélioré. L'objectif suivant sera d'étendre la méthode aux équations de Navier-Stokes afin de permettre une utilisation conjointe à l'ITPM offrant ainsi un couplage implicite en temps et un ordre deux en espace. A terme, un traitement d'interfaces fluide-fluide, bien que complexe de par la nature des conditions de sauts concernées, semble réalisable.

Afin d'utiliser ces méthodes pour des cas industriels complexes, nous avons développé des méthodes de projections de maillage. Notre nouvelle stratégie, qui consiste à ramener le maillage curviligne à un maillage cartésien, a montré son efficacité. La projection curviligne-eulerienne est quelque peu complexe à implémenter mais facilite énormément l'implémentation d'autres méthodes (projections surfacique-volumique, calcul d'efforts, advection...) ainsi que leur rapidité d'exécution (tout particulièrement pour la méthode de Ray-Casting). Sur ces aspects, les futurs développement porteront sur l'adaptation de ces méthodes à un environnement parallèle a mémoire partagée de type OpenMP ou GPU. Les diverses méthodes de projection de maillage effectuent beaucoup d'opérations indépendantes et sont donc faciles à paralléliser. Pour ce qui est du MPI, le code Thétis procède par décomposition de domaines ce qui n'influence pas les routines de projection qui fonctionnent ainsi naturellement en parallèle MPI. Un autre aspect important sera de trouver une correction adéquate des propriétés géométriques de la fonction

distance. Ces dernières sont un des principaux intérêts de la méthode Level-set mais ne sont pas valables dans l'espace réel si la fonction distance a été construite dans un espace transformé.

Concernant les objets mobiles, nous avons étendu l'approche de [Coqu 08] à la pénalisation de sous-maille et au lagrangien augmenté. Nous avons choisi d'advecter le maillage des objets plutôt qu'une fonction volumique. Cette approche permet de conserver exactement la forme de l'objet et reste très performante en terme de temps de calcul pour peu que l'effort d'implémentation nécessaire soit déployé (Thread Ray-Casting, Octree...). Il est toutefois apparu que l'ITPM était généralement plus précise. Elle est également plus robuste dans de nombreux cas. Cela renforce l'idée que la voie à suivre est de coupler cette méthode avec la méthode d'interface immergée algébrique. Pour ce qui est des interactions solide-solide, un code de calcul temps réel avec visualisation OpenGL a été construit afin de faciliter l'étude d'un algorithme de collision. Les résultats sont satisfaisants sauf pour les cas raides de type empilements.

Toutes ces méthodes ont permis de traiter des cas industriels complexes. Les simulations d'hydroplanage ont produit des résultats en cohérence avec des expérimentations. Nous avons aussi développé pour Varel un simulateur d'écoulements dans les têtes de forage avec une gestion de copeaux. Ce dernier point peut encore être grandement amélioré par l'ajout de propriétés physiques dans la dynamique des copeaux. Le troisième cas, la grotte de Lascaux, est un bon exemple d'aide à la conservation du patrimoine. De nombreux médias (Libération, JT TF1, France 2, France 3) se sont fait écho du simulateur et ont diffusé les images de synthèses réalisées par nos soins ce qui montre le potentiel de communication et de vulgarisation de cette approche. Concernant l'imagerie de synthèse, les cas présentés en annexe ont été réalisés avant que ce travail ne commence. Les différents outils et améliorations développées ici permettront d'obtenir rapidement de nouvelles séquences plus intéressantes.

Tout ceci démontre la capacité d'une approche "tout cartésien" a mener des simulations pour l'industrie ou l'environnement avec une puissance de calcul réduite, et malgré leur complexité, les cas industriels de Michelin et Varel ont été traités sur des machines standards. Des cas similaires continuent à être traités par ces entreprises avec notre simulateur, et ce toujours sur des machines de bureau. Quand à l'évolution de ces dernières, la tendance est résolument une augmentation soutenue du nombre de processeurs pour une évolution relativement moindre de la mémoire vive. Ce dernier point nous pousse –pour ce qui est des perspectives a plus long terme– à considérer sérieusement un portage complet de notre approche aux architectures multi-cœurs (CPU comme GPU) a mémoire partagée.

Ce travail a donné lieu a plusieurs publications et actes de congrés:

Publications avec comité de lecture

- A. Sarthou, S. Vincent, J.-P. Caltagirone, P. Angot, "Eulerian-Lagrangian grid coupling and penalty methods for the simulation of multiphase flows interacting with complex objects" International Journal of Numerical Methods in Fluids, 2008; 56-8:1093-1099.
- D. Lacanette, S. Vincent, A. Sarthou, P. Malaurent and J.-P. Caltagirone, "An Eulerian/Lagrangian method for the numerical simulation of incompressible convection flows interacting with complex obstacles: application to the natural convection in the Lascaux cave". International Journal on Heat en Mass Transfer 52 (2009) 2528-2542.
- F. Nadal, J.-P. Lambelin, R. Lagrange, A. Sarthou, "Non-resonant viscous theory for the stability of a fluid-filled gyroscope". Journal of Fluid Mechanics 639 (2009) 167-194.

Publications soumises

- A. Sarthou, S. Vincent, P. Angot, J.-P. Caltagirone, "The Algebraic Immersed Interface and Boundary method for elliptic equations with discontinuous coefficients", Hal : http://hal.archives-ouvertes.fr/hal-00390075/fr/.
- S. Vincent, A. Sarthou, J.-P. Caltagirone, F. Sonilhac, P. Février and G. Pianet "Augmented Lagrangian and penalty methods for the simulation of two-phase flows interacting with moving solids. Application to hydroplaning flows interacting with real tire structures", Hal : http://hal.archives-ouvertes.fr/hal-00424018/fr/

Actes de congrès internationaux avec comité de lecture

- A. Sarthou, S. Vincent, P. Angot, JP. Caltagirone, "The Sub-Mesh Penalty Method". 5th International Symposium on Finite Volumes for Complex Applications, Aussois, France, June 08-13, 2008.
- A. Sarthou, S. Vincent, JP. Caltagirone, P. Angot, "Eulerian-Lagrangian coupling grid technology and penalty methods for the simulation of multiphase flows interacting with complex objects". 9th International Conference on Numerical Methods for Fluid Dynamics, ICFD 2007, Reading, England,March 26-29, 2007.

Actes de congrès nationaux avec comité de lecture

- A. Sarthou, S. Vincent, J.-P. Caltagirone, P. Angot, La méthode d'interface immergée algébrique pour les équations elliptiques à coefficients discontinus, SMAI 2009.
- A. Sarthou, S. Vincent, J.-P. Caltagirone, P. Angot, La méthode d'interface immergée algébrique, CFM 2009.
- D. Lacanette, S. Vincent, A. Sarthou, J.P. Caltagirone, P. Malaurent, Pénalisation curviligne pour la simulation des transferts de masse et de chaleur dans la grotte de Lascaux, SFT 2009.
- A. Sarthou, S. Vincent, JP. Caltagirone, La méthode de pénalisation de sous-mailles. 3eme Journée Nationale des Doctorants Thermiciens.

Part VIII

Appendices

Appendix A

Conservation equations and related numerical context

A.1 Preamble

In this section, the numerical methods used in the home made code Thetis (formerly Aquilon) are presented. This document has been written at the laboratory TREFLE which is composed of both mathematicians and physicists. Hence, each element of the following chapter will possibly be considered as evidence by one of the two communities.

A.2 Equations

Let us consider first a second-order linear PDE:

$$au_{xx} + 2bu_{xy} + cu_{xx} + 2du_x + 2eu_y + fu = 0 \tag{A.1}$$

Three kinds of equations can be identified according to the value of $b^2 - ac$:

- $b^2 - ac > 0$, the equation is hyperbolic (wave equation)
- $b^2 - ac = 0$, the equation is parabolic (diffusion equation)
- $b^2 - ac < 0$, the equation is elliptic (Poisson equation)

A.2.1 Conservation equations

Conservation equations are all based on the consideration of the flux of some state variable flowing into and out of some region of the domain. In general the sources and sinks within this region are also considered in arriving at a conservation equation. These fluxes generally depend on position, but they may also depend on the state variable itself, or they may represent fluxes of state variable carried into the region by moving material. Let us consider a physical quantity Φ. The following PDE is a generic scalar transport equation :

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot \mathbf{f}(t, \mathbf{x}, \Phi, \nabla\Phi) = \mathbf{g}(t, \mathbf{x}, \Phi) \tag{A.2}$$

where $\mathbf{f}$ is called the flux and $\mathbf{g}$ the source.

A.2.2 General elliptic equations

An operator P is defined as elliptic is the equation $Pu = 0$ is elliptic. An important case of elliptic operator is the Laplacian, or Laplace operator, denoted as $\Delta = \nabla^2$. In non-Euclidian spaces, the Laplace operator can be generalized and is not necessarily elliptic (*e.g.* the Laplace operator becomes the d'Alembert operator $\Box = \partial_t^2 - \partial_x^2 - \partial_y^2 - \partial_z^2$ in the Minkowski space). The Laplace equation is $\Delta u = 0$. With a source term f, the Laplace equation becomes the Poisson equation $\Delta u = f$. One can notice that the heat equation, $u_t - a\Delta u = f$, is a parabolic equation but its steady state solution solves the corresponding elliptic equation. When no particular physical application is considered, the present work generally considers the following model elliptic equation:

$$-\nabla \cdot (\mathbf{a}\nabla u) + bu = f \tag{A.3}$$

Let us consider the following model problem:
For $\tilde{\mathbf{a}} \in (L^\infty(\tilde{\Omega}))^{d\times d}$, $\tilde{b} \in L^\infty(\tilde{\Omega})$ *and* $\tilde{f} \in L^2(\Omega)$, find a function $\tilde{u}$ defined on $\tilde{\Omega}$ such that:

$$\begin{cases} -\nabla \cdot (\tilde{\mathbf{a}}\nabla\tilde{u}) = \tilde{f} & \text{in } \tilde{\Omega} \\ B.C. & \text{on } \partial\tilde{\Omega} \end{cases} \tag{A.4}$$

where B.C. represent several types of boundary conditions:

- A Dirichlet condition $\tilde{u} = u_D$ with $u_D \in H^{1/2}(\partial\tilde{\Omega})$,
- A Robin (or Fourier) condition: $-(\tilde{\mathbf{a}}.\nabla).\mathbf{n} = \alpha_R\tilde{u} + g_R$, with $\alpha_R \in L^\infty(\partial\tilde{\Omega})$; $\alpha_R \geq 0$, and $g_R \in L^2(\partial\tilde{\Omega})$
- A Neumann condition, $-(\tilde{\mathbf{a}}.\nabla).\mathbf{n} = g$, considered as a particular case of the Robin condition where $\alpha_R \equiv 0$ and $g_R \equiv \tilde{g}$

Moreover, the tensor of diffusion $\tilde{\mathbf{a}} \equiv (\tilde{a}_{ij})_{1\leq i,j\leq d}$ and the reaction coefficient $\tilde{b}$ verify the classical ellipticity assumptions:

$$\exists a_0 > 0, \forall \xi \in \mathbb{R}^d,\ \tilde{\mathbf{a}}(x).\xi.\xi \geq a_0|\xi|^2 \ a.e.\ in\ \tilde{\Omega}$$
$$where\ |.|\ is\ the\ Euclidian\ norm\ in\ \mathbb{R}^d$$

$$\exists b_0 \geq 0,\ \tilde{b}(x) \geq b_0\ a.e.\ in\ \tilde{\Omega}$$

In this case, the classical variational techniques (e.g. [Ravi 82]) prove that the solution $\tilde{u}$ of the original problem exists and is the unique solution $\tilde{u}$ in the space $H^1(\tilde{\Omega})$ satisfying the weak formulation of the problem.

A.2.3 The incompressible Navier-Stokes equation

The Navier-Stokes (NS) equations are named after Claude-Louis Navier and George Gabriel Stokes. They describe the motion of fluid substances, *i.e* substances which can flow. Extremely useful, they can describe a large amount of phenomena, such as weather, flow around a car or a plane, or blood flows.
The unknowns of the NS equations are generally the velocity and the pressure. The nonlinear term due to convective acceleration provides a time dependent chaotic behavior called turbulence. The NS equations are often extremely difficult to solve numerically when turbulence appears. Mathematically, the existence and the smoothness of the 3D NS equations is not demonstrated.

Let us consider the domain of interest Ω of boundary $\partial\Omega$.

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \tag{A.5}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = \rho\mathbf{g} - \nabla p + \nabla\cdot[\mu(\nabla\mathbf{u} + \nabla^T\mathbf{u})] + \sigma k\mathbf{n}_i\delta_i \text{ in } \Omega \tag{A.6}$$

$$\frac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C = 0 \text{ in } \Omega \tag{A.7}$$

A.3 Numerical methods

A.3.1 The finite volume method

We consider the integral form of (A.2) for the whole domain Ω :

$$\int_\Omega \frac{\partial \Phi}{\partial t}\,\mathrm{d}V + \int_\Omega \nabla\cdot\mathbf{f}(t,\mathbf{x},\Phi,\nabla\Phi)\,\mathrm{d}V = \int_\Omega \mathbf{g}(t,\mathbf{x},\Phi)\,\mathrm{d}V. \tag{A.8}$$

One can notice that the equation (A.8) is the flux balance for the whole domain Ω. The integral form of the equation is now rewritten for all CVs $\mathcal{V}_i$ of measure v_i compounding the domain Ω:

$$\int_{\mathcal{V}_i} \frac{\partial \Phi}{\partial t}\,\mathrm{d}V + \int_{\mathcal{V}_i} \nabla\cdot\mathbf{f}(t,\mathbf{x},\Phi,\nabla\Phi)\,\mathrm{d}V = \int_{\mathcal{V}_i} \mathbf{g}(t,\mathbf{x},\Phi)\,\mathrm{d}V. \tag{A.9}$$

Numerous numerical schemes can be used to discretize each of the terms. On integrating the first term to get the volume average and applying the divergence theorem to the second, this yields

$$v_i\frac{\partial \Phi}{\partial t} + \oint_{\mathcal{S}_i} \mathbf{f}(t,\mathbf{x},\Phi,\nabla\Phi).\mathbf{n}\,\mathrm{d}S = \int_{\mathcal{V}_i} g(t,\mathbf{x},\Phi)\,\mathrm{d}V. \tag{A.10}$$

Hence, the discretization of the second term is based on the fluxes at the faces of the CVs. As the FV method uses the integral form of the conservation equations, the conservation of the physical properties is straightforwardly obtained.

A.3.2 The staggered grid

Our methodology uses four primary variables (u, v, w, p) to solve the incompressible Navier-Stokes equations. The intuitive discretization consists in putting the variables four by four at the same location. The calculation of derivatives in such a "colocative" grid leads to major difficulties. Let us consider a cell-centered pressure variable p_C and his left and right neighbors p_W and p_E. The location of the faces of the control volume are p_w and p_e. The calculation of the 1D pressure gradient $\frac{dp}{dx}$ gives the quantity $p_e - p_w$. Using the cell-centered variable, we obtain:

$$p_e - p_w = \frac{p_W - p_C}{2} - \frac{p_C - p_E}{2} = \frac{p_W - p_E}{2} \tag{A.11}$$

This means that the momentum equation will contain the pressure difference between two alternate grid points, and not between adjacent ones. First, the pressure is taken from a coarser grid than the one actually employed. The same problem occurs with discretization of $\nabla\cdot\mathbf{u}$ in the pressure correction equation or in the Uzawa algorithm. But this discretization has a far more serious problem. Let us consider a *zig-zag* pressure field alternating between two constant values p_1 and p_2. If $p_W = p_1$, $p_C = p_2$ and $p_E = p_1$, the calculated pressure gradient in p_C is

null. Hence, the gradient of a pressure field will be seen as constant so the pressure field will be perceived as constant. The same effect is obtained in 2D with a checkboard-like pressure field. And if a given pressure field is obtained as a solution, any number of additional solutions can be constructed by adding a checkboard pressure field to that solution.

Hence, an alternative interpolation method, such as the *Rhie and Chow* method [Rhie 83] is commonly used. This discretization uses a weighting factor to take into account the value of the pressure in p_C when calculating the pressure gradient at this location.

Harlow and Welch proposed, combined with their MAC method, to use a different grid for each variable [Harl 65]. In this staggered grid, the velocity components are calculated for the points that lie on the faces of the control volumes. Thus, the x-direction velocity is calculated at the faces that are normal to the x direction. In Fig. A.1, the location of the velocity variables is shown as arrows while circles shows the location of the pressure points. With such a grid,

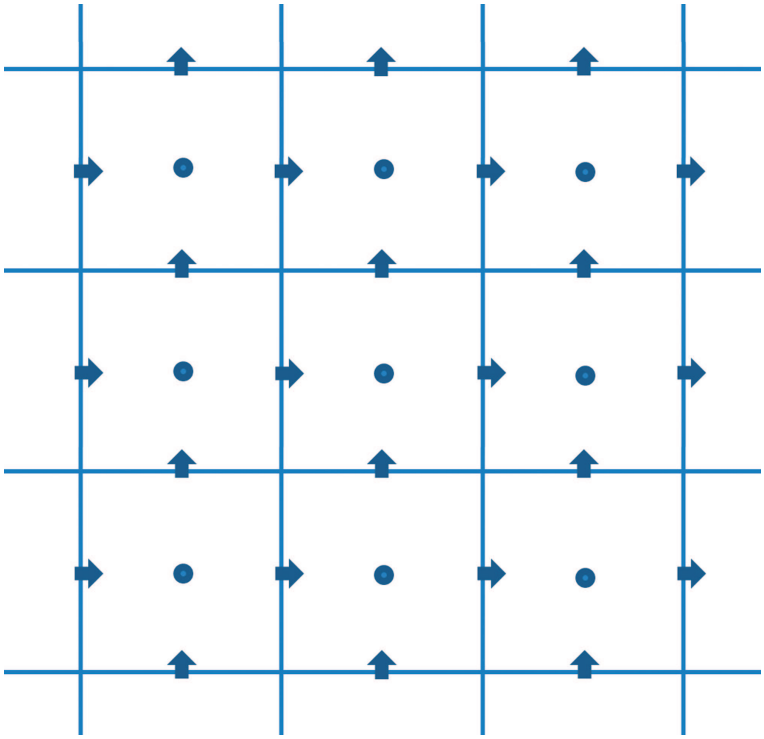


Figure A.1: Location of the unknowns for the staggered grid

the mass flow rates across the CVs can be calculated without any interpolations for the relevant velocity component. Moreover, the discretization of the pressure gradient in the momentum equation and the discretization of the divergence of the velocity for the pressure correction and for the Uzawa algorithm use adjacent grid points. Hence, the checkboard effect and its related problems observed with a colocative approach are no more. However, as $1 + d$ grids are used instead of a unique one, the staggered grid method require a larger amount of memory and some additional computation efforts to determine the location of the velocity nodes. Nevertheless, the benefits of such an approach are well worth the additional troubles.

A.3.3 The velocity-pressure coupling

A.3.3.1 The scalar projection method

A first common way to obtain the pressure when solving the Navier-Stokes equation is the SIMPLE algorithm of Patankar and Spalding [Pata 72].

The idea is to obtain first a predicted velocity from the momentum equation. This velocity is not divergence free. In a second step, the projection step, the pressure is risen with respect to the divergence of the velocity obtained in the prediction step. The third step consists in updating the velocity according to the pressure gradient obtained with the second step.
Let write the momentum Navier-Stokes equation:

$$\frac{\partial u}{\partial t} = +RHS - \nabla p \tag{A.12}$$

with RHS the convective, diffusive and source terms. The half discretization in time gives:

$$\rho\left(\frac{\mathbf{u}_i^{n+1} - \mathbf{u}_i^n}{\Delta t}\right) = RHS_i^{n+1} - \nabla_i p^{n+1} \tag{A.13}$$

This equation is solved, but as here $\nabla \cdot \mathbf{u}^{n+1} \neq 0$, a first predicted solution $\mathbf{u}^*$ can only be obtained. We define $\mathbf{u}'$ such as $\mathbf{u}^{n+1} = \mathbf{u}' + \mathbf{u}^*$ and p' such as $p^{n+1} = p' + p^*$. Hence, the predictor step solves:

$$\rho\left(\frac{\mathbf{u}_i^* - \mathbf{u}_i^n}{\Delta t}\right) = RHS_i^* - \nabla_i p^n \tag{A.14}$$

One can write now (A.13)-(A.14):

$$\rho\left(\frac{\mathbf{u}_i^{n+1} - \mathbf{u}^*}{\Delta t}\right) = RHS' - \nabla_i p' \tag{A.15}$$

$$\left(\mathbf{u}_i^{n+1} - \mathbf{u}^*\right) = \frac{\Delta t}{\rho} RHS' - \frac{\Delta t}{\rho}\nabla_i p' \tag{A.16}$$

The correction equations is then defined by writing the divergence of (A.16). The term $\nabla \cdot \frac{\Delta t}{\rho} RHS'$ is neglicted. We obtain :

$$\nabla \cdot \mathbf{u}_i^* = \nabla \cdot \frac{\Delta t}{\rho}\nabla_i p'. \tag{A.17}$$

Once the pressure increment is obtained, velocity and pressure are updated:

$$p^{n+1} = p' + p^n \tag{A.18}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{\Delta t}{\rho}\nabla_i p' \tag{A.19}$$

In [Timm 96], Timmermans *et al.* proposes a correction of this last step replacing (A.18 by

$$p^{n+1} = p' + p^n - \mu \nabla \cdot u^*. \tag{A.20}$$

This correction gives better results with the pressure Neumann BC. One can find an overview of the different projection methods in [Guer 06].

A.3.3.2 The Uzawa operator

The discrete Navier-Stokes equations are written as:

$$A\mathbf{u}^{n+1} + Gp^{n+1} = F \tag{A.21}$$

$$D\mathbf{u}^{n+1} = 0 \tag{A.22}$$

where D is the discrete divergence matrix and G the discrete gradient matrix. If A is inversible, one can rewrite (A.21) as

$$D\mathbf{u}^{n+1} + DA^{-1}Gp^{n+1} = DA^{-1}F \tag{A.23}$$

As $D\mathbf{u}^{n+1} = 0$, the pressure can be solved with

$$DA^{-1}Gp^{n+1} = DA^{-1}F \tag{A.24}$$

and $DA^{-1}G$ is the Uzawa operator. The system (A.24) can be solved with an iterative method. We choose $p_0^{n+1} = p^n$ and m is the current iteration of the iterative algorithm. The Richardson method gives:

$$p_{m+1}^{n+1} = p_m^{n+1} - dp(DA^{-1}F - DA^{-1}Gp_m^{n+1}) \tag{A.25}$$

and using $DA^{-1}F - DA^{-1}Gp^{n+1} = D\mathbf{u}^{n+1}$,

$$p_{m+1}^{n+1} = p_m^{n+1} - dpD\mathbf{u}_{m+1}^{n+1} \tag{A.26}$$

A.3.4 The augmented Lagrangian method

A.3.4.1 Theoretical formulation

Let us consider the incompressible Navier-Stokes equation on a domain $\Omega \in \mathbb{R}^d$:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = \rho\mathbf{g} - \nabla p + \nabla\cdot[\mu(\nabla\mathbf{u} + \nabla^T\mathbf{u})] + \sigma k\mathbf{n}_i\delta_i \text{ in } \Omega \tag{A.27}$$

$$\nabla\cdot\mathbf{u} = 0 \text{ in } \Omega \tag{A.28}$$

Contrary to scalar projection methods, the augmented Lagrangian (AL) [Fort 82] proposes to satisfy the two equations (A.28) and (A.27) at the same time, resulting a divergence free flow $\mathbf{u}$ after only one matrix inversion. The method uses at the same time a minimisation under constraint and a penalty term to accelerate the convergence. The pressure p is here a Lagrange multiplier which allow the constraint to be ensured.
For all $\mathbf{v} \in (H_0^1(\Omega))^d$, let $J(\mathbf{v})$ be a functional built from the weak formulation of the momentum equation (A.27). This functional has to be minimized under the constraint $\mathbf{u}, \mathbf{v} \in M = \{\mathbf{v} \in (H_0^1(\Omega))^d, \nabla\cdot\mathbf{v} = 0\}$. This problem is equivalent to the following one:

$$\begin{cases} J(\mathbf{u}) \leq J(\mathbf{v}), & \forall \mathbf{v} \in M \\ \mathbf{u} \in M \end{cases} \tag{A.29}$$

Practically, a solution in a constrained space such as M cannot be easily computed. This problem of minimization under constraint is transformed into a problem of minimization without constraint introducing a pressure q as a Lagrange multiplier. We define the following Lagrangian:

$$L(\mathbf{v}, q) = J(\mathbf{v}) - \int_\Omega q\nabla\cdot\mathbf{v}\ \mathrm{d}\Omega \tag{A.30}$$

The minimization problem (A.29) consists in finding a saddle-point $(\mathbf{u}, p) \in (H_0^1(\Omega))^d \times L^2(\Omega)$ of the Lagrangian (A.30):

$$\begin{cases} L(\mathbf{u}, q) \leq L(\mathbf{u}, p) \leq L(\mathbf{v}, p) \quad \forall v \in (H_0^1(\Omega))^d,\ \forall q \in L^2(\Omega), \\ \mathbf{u} \in (H_0^1(\Omega))^d,\ \forall p \in L^2(\Omega) \end{cases} \tag{A.31}$$

which implies

$$\begin{array}{rcl} L(\mathbf{u}, p) & = & \min_{\mathbf{v} \in (H_0^1(\Omega))^d} \max_{q \in L^2(\Omega)} L(\mathbf{v}, q) \\ & = & \max_{q \in L^2(\Omega)} \min_{\mathbf{v} \in (H_0^1(\Omega))^d} L(\mathbf{v}, q) \end{array} \tag{A.32}$$

In order to increase the convergence rate [Fort 82], the constraint is used to build a penalty term $\frac{1}{2} dr |\nabla \cdot \mathbf{v}|^2$, $dr \in \mathbb{R}$. The augmented Lagrangian is denoted as:

$$L_r(\mathbf{v}, q) = J(\mathbf{v}) - \int_\Omega q \nabla \cdot \mathbf{v} \ \mathrm{d}\Omega + \int_\Omega \frac{dr}{2} |\nabla \cdot \mathbf{v}|^2 \ \mathrm{d}\Omega \tag{A.33}$$

and the relation (A.31) can be applied to $\mathcal{L}_r$ too.
We admit that the solution of saddle-point problem for the weak formulation of the initial equations is the solution of the strong formulation of the problem (proven for the Stokes equations, admitted for the Navier-Stokes equation). The resulting Navier-Stokes equations are:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) = \rho \mathbf{g} - \nabla p + \nabla \cdot [\mu(\nabla \mathbf{u} + \nabla^T \mathbf{u})] + \sigma k \mathbf{n}_i \delta_i - dr \nabla(\nabla \cdot \mathbf{v}) \text{ in } \Omega \tag{A.34}$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \tag{A.35}$$

As the value of dr is for now arbitrary, $\frac{dr}{2}$ has been replaced by dr for the sake of simplicity.

A.3.4.2 Numerical application

The base methodology The numerical resolution is an iterative process. The equation (A.34) is solved with an explicite pressure which is then updated with (A.26). For the sake of simplicity, the method is first written for one iteration per time step only. Hence, the following implicit problem is solved:

$$\begin{array}{c} \rho \left(\dfrac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n \cdot \nabla \mathbf{u}^{n+1} \right) - dr \nabla(\nabla \cdot \mathbf{u}^{n+1}) \\ = -\nabla p^n + \rho \mathbf{g} + \nabla \cdot [\mu(\nabla \mathbf{u}^{n+1} + \nabla^T \mathbf{u}^{n+1})] + \sigma k \mathbf{n}_i \delta_i \end{array} \tag{A.36}$$

The pressure is then updated using

$$p^{n+1} = p^n - dp \nabla \cdot \mathbf{u}^{n+1} \tag{A.37}$$

The equations (A.36) and (A.37) can be rewritten as:

$$A\mathbf{u}^{n+1} - drDG\mathbf{u}^{n+1} + Gp^n = F \tag{A.38}$$

$$p^{n+1} = p^n - dpD\mathbf{u}_{m+1}^{n+1} \tag{A.39}$$

Ideally, one want to solve

$$A\mathbf{u}^{n+1} + Gp^{n+1} = F \tag{A.40}$$

Using (A.39) on (A.40), we obtain

$$A\mathbf{u}^{n+1} - GdpD\mathbf{u}^{n+1} + Gp^n = F \tag{A.41}$$

The augmented Lagrangian terms in (A.38) and (A.41) differs and are not equivalent, even if $dp = dr$. The term $drDG\mathbf{u}^{n+1}$ is inherited from the mathematical formulation and comes from the penalty term $\int_\Omega \frac{dr}{2}|\nabla\cdot\mathbf{v}|^2\ \mathrm{d}\Omega$ for the weak problem. The solution is to integrate by part the weak term considering $\frac{dr}{2}\nabla\cdot\mathbf{v}$ as the integrated term instead of $\nabla\cdot\mathbf{v}$ which is the standard choice in [Fort 82]. The resulting penalty term is $-\int_\Omega(\nabla\frac{dr}{2}\nabla\cdot\mathbf{v})\mathbf{v}\ \mathrm{d}\Omega$ and the consistant penalty term $-GdpD\mathbf{u}^{n+1}$ can be retrieved for a good choice of dr. In all our algorithms, we choose $dp = dr$.

Standard Augmented Lagrangian (SAL) Starting with $\mathbf{u}^{*,0} = \mathbf{u}^n$ and $p^{*,0} = p^n$, the predictor solution reads while $||\nabla\cdot\mathbf{u}^{*,\mathbf{m}}|| > \epsilon$, solve

$$\left\{\begin{array}{c}(\mathbf{u}^{*,0},p^{*,0}) = (\mathbf{u}^n,p^n)\\ \rho\left(\dfrac{\mathbf{u}^{*,m}-\mathbf{u}^{*,0}}{\Delta t}+\mathbf{u}^{*,m-1}\cdot\nabla\mathbf{u}^{*,m}\right)-r\nabla(\nabla\cdot\mathbf{u}^{*,m})\\ =-\nabla p^{*,m-1}+\rho\mathbf{g}+\nabla\cdot[\mu(\nabla\mathbf{u}^{*,m}+\nabla^T\mathbf{u}^{*,m})]+\sigma k\mathbf{n}_i\delta_i\\ p^{*,m}=p^{*,m-1}-r\nabla\cdot\mathbf{u}^{*,m}\end{array}\right. \tag{A.42}$$

where r is the augmented Lagrangian parameter used to impose the incompressibility constraint, m is an iterative convergence index and ϵ a numerical threshold controlling the constraint. Usually, a constant value of r is used. From numerical experiments, optimal values are found to be of the order of ρ_i and μ_i to accurately solve the motion equations in the related zone [Vinc 07] . The momentum, as well as the continuity equations are accurately described by the predictor solution $(\mathbf{u}^*,p^*)$ coming from (A.42) in the medium, where the value of r is adapted. However, high values of r in the other zones act as penalty terms inducing the numerical solution to satisfy the divergence free property only. Indeed, if we consider for example $\rho_1/\rho_0 = 1000$ (characteristic of water and air problems) and a constant $r = \rho_1$ to impose the divergence free property in the denser fluid, the asymptotic equation system solved in the the predictor step is:

$$\left\{\begin{array}{c}\rho\left(\dfrac{\mathbf{u}^*-\mathbf{u}^n}{\Delta t}+(\mathbf{u}^n\cdot\nabla)\mathbf{u}^*\right)-r\nabla(\nabla\cdot\mathbf{u}^*)\\ =\rho\mathbf{g}-\nabla p^n+\nabla\cdot[\mu(\nabla\mathbf{u}^*+\nabla^T\mathbf{u}^*)]+\sigma k\mathbf{n}_i\delta_i \text{ in } \Omega_1\\ \dfrac{\mathbf{u}^*-\mathbf{u}^n}{\Delta t}-r\nabla(\nabla\cdot\mathbf{u}^*)=0 \text{ in } \Omega_0\end{array}\right. \tag{A.43}$$

Our idea is to locally estimate the augmented Lagrangian parameter in order to obtain satisfactory equivalent models and solutions in all the media.

Adaptive Augmented Lagrangian (2AL) Instead of choosing an empirical constant value of r fixed at the beginning of the simulations, we propose at each time step to locally estimate the augmented Lagrangian parameter r. Then, $r(t,M)$ becomes a function of time t and space position M. It must be two to three orders of magnitude higher than the most important term in the conservation equations.
Let L_0, t_0, u_0 and p_0 be reference space length, time, velocity and pressure respectively. If we consider one iterative step of the augmented Lagrangian procedure (A.42), the non-dimensional

form of the momentum equations can be rewritten as

$$\begin{aligned}
&\rho\frac{u_0}{t_0}\frac{\mathbf{u}^{*,m}-\mathbf{u}^n}{\Delta t}+\rho\frac{u_0^2}{L_0}(\mathbf{u}^{*,m-1}\cdot\nabla)\mathbf{u}^{*,m}\\
&-\frac{u_0}{L_0^2}\nabla(r\nabla\cdot\mathbf{u}^{*,m})=\rho\mathbf{g}-\frac{p_0}{L_0}\nabla p^{*,m-1}\\
&+\frac{u_0}{L_0^2}\nabla\cdot[\mu(\nabla\mathbf{u}^{*,m}+\nabla^T\mathbf{u}^{*,m})]+\frac{\sigma}{L_0^2}k\mathbf{n}_i\delta_i
\end{aligned}\tag{A.44}$$

Multiplying the right and left parts of equation (A.44) by L_0^2/u_0, we can compare the augmented Lagrangian parameter r to all the contributions of the flow (inertia, gravity, pressure and viscosity). We obtain

$$\begin{aligned}
&\rho\frac{L_0^2}{t_0}\frac{\mathbf{u}^{*,m}-\mathbf{u}^n}{\Delta t}+\rho u_0L_0(\mathbf{u}^{*,m-1}\cdot\nabla)\mathbf{u}^{*,m}\\
&-\nabla(r\nabla\cdot\mathbf{u}^{*,m})=\rho\frac{L_0^2}{u_0}\mathbf{g}-\frac{p_0L_0}{u_0}\nabla p^{*,m-1}\\
&+\nabla\cdot[\mu(\nabla\mathbf{u}^{*,m}+\nabla^T\mathbf{u}^{*,m})]+\frac{\sigma}{u_0}k\mathbf{n}_i\delta_i
\end{aligned}\tag{A.45}$$

It can be noticed that r is comparable to a viscosity coefficient. It is then defined as

$$\begin{aligned}
r(t,M)=K\ max\Big(&\rho(t,M)\frac{L_0^2}{t_0},\rho(t,M)u_0L_0,\\
&\rho(t,M)\frac{L_0^2}{u_0}g,\frac{p_0L_0}{u_0},\mu(t,M),\frac{\sigma}{u_0}\Big)
\end{aligned}\tag{A.46}$$

If $\dfrac{\rho_1}{\rho_0}=1000$ and $\mu_0<\mu_1<<\rho_0<<\rho_1$ for example, the semi-discrete form of the momentum equations resulting from the new values of $r(t,M)$ given by (A.46) then becomes

$$\begin{aligned}
&\rho_1\left(\frac{\mathbf{u}^{*,m}-\mathbf{u}^n}{\Delta t}+(\mathbf{u}^{*,m-1}\cdot\nabla)\mathbf{u}^{*,m}\right)-\nabla(r\nabla\cdot\mathbf{u}^{*,m})\\
&=\rho_1\mathbf{g}-\nabla p^{*,m-1}+\nabla\cdot[\mu_1(\nabla\mathbf{u}^{*,m}+\nabla^T\mathbf{u}^{*,m})]+\sigma k\mathbf{n}_i\delta_i\\
&\qquad\text{in }\Omega_1\text{ with }r=K_1\rho_1L_0^2u_0\\
&\rho_0\left(\frac{\mathbf{u}^{*,m}-\mathbf{u}^n}{\Delta t}+(\mathbf{u}^{*,m-1}\cdot\nabla)\mathbf{u}^{*,m}\right)-\nabla(r\nabla\cdot\mathbf{u}^{*,m})\\
&=\rho_0\mathbf{g}-\nabla p^{*,m-1}+\nabla\cdot[\mu_0(\nabla\mathbf{u}^{*,m}+\nabla^T\mathbf{u}^{*,m})]+\sigma k\mathbf{n}_i\delta_i\\
&\qquad\text{in }\Omega_0\text{ with }r=K_0\rho_0L_0^2u_0
\end{aligned}\tag{A.47}$$

where K_0 and K_1 are in between 10 and 1000. In this way, thanks to expression (A.45), the adaptive Lagrangian parameter is at least 10 to 1000 times the order of magnitude of the most important term between inertia, viscosity, pressure or gravity in both Ω_1 and Ω_0 domains. Compared to the SAL approach (A.43), the 2AL is consistent with the Navier-Stokes equations in each phase [Vinc 07] . The new method (A.46) can be easily extended to other forces such as surface tension and Coriolis or specific source terms. Comparisons between standard and adaptive augmented Lagrangian (2AL) methods are presented in the next section. In particular, the

influence of the penalty parameter on the convergence speed of the BiCGSTAB solver and time and space variations of $r(M)$ are discussed.

As a summary, the complete time-marching procedure of the predictor-corrector algorithm including the 2AL method is the following:

$\otimes$ **Step 1**: defines initial values $\mathbf{u}^0$ and p^0 and boundary conditions on Γ,

$\otimes$ **Step 2**: knowing $\mathbf{u}^n$, p^n and a divergence threshold ϵ, estimates the predictor values $\mathbf{u}^*$ and p^* with the Uzawa algorithm (A.42) associated to the local estimate of $r(t, M)$ defined in expression (A.46), so that $\mathbf{u}^* = \mathbf{u}^{*,m}$ and $p^* = p^{*,m}$ when m verifies $||\nabla \cdot \mathbf{u}^{*,\mathbf{m}}|| < \epsilon$

$\otimes$ **Step 3**: projects the solution $(\mathbf{u}^*, p^*)$ on a divergence free subspace thanks, for example, to projection approaches [Goda 78, Calt 99] to get the correction solution $(\mathbf{u}', p')$. Then, the numerical solution at time $(n+1)\Delta t$ is $(\mathbf{u}^{n+1}, p^{n+1}) = (\mathbf{u}^* + \mathbf{u}', p^* + p')$,

$\otimes$ **Step 4**: iterates n in steps 3 and 4 until the physical time is reached.

Algebraic adaptive Augmented Lagrangian (3AL) It has been demonstrated that estimating a local and adapted augmented Lagrangian parameter is crucial for simulating multiphase flows [Vinc 07] . The main remaining drawback of the 2AL method is linked to the *a priori* definition of dimensionless parameters for defining $r(t, M)$. The augmented Lagrangian approach is based on the concept of a penalty method. As a consequence, the augmented Lagrangian parameter acts as an algebraic parameter which increases the magnitude of specific coefficients in the linear system in order to verify a specific constraint, while solving at same time the conservation equations. In this section, an estimate of $r(t, M)$ which is based on a scanning of the linear system is proposed. The main interests of the algebraic adaptive augmented Lagrangian method (3AL) are the following: it does not require any *a priori* physical information, it applies to any kind of geometry and grid and it takes into account the residual of the linear solver and the fulfilment of incompressible and solid constraints.

At each time step and in two-dimensions, the 3AL method determines $r(t, M)$ as follows:

$\otimes$ **Step 1**: Two matrix A and A^* are built corresponding respectively to the discretization of the momentum equations with $r(t, M) = 0$ and $r(t, M) = 1$. In order to optimize computer memory, a compressed storage raw (CSR) structure is chosen to store only the non null coefficients of each matrix,

$\otimes$ **Step 2**: On the fixed staggered Cartesian grid, $r(t)_{i,j}$ is evaluated according to the discretization coefficients of the surrounding velocity $u_{x,i-\frac{1}{2},j}$, $u_{x,i+\frac{1}{2},j}$, $u_{y,i,j-\frac{1}{2}}$ and $u_{y,i,j+\frac{1}{2}}$ components, as presented on figure A.2.

The discretization of each velocity component, $u_{x,i-\frac{1}{2},j}$ for example, requires the use of 9 neighboring velocity nodes, i.e. $u_{x,i-\frac{1}{2},j}$, $u_{x,i+\frac{1}{2},j}$,$u_{x,i-\frac{3}{2},j}$, $u_{x,i-\frac{1}{2},j+1}$ and $u_{x,i-\frac{1}{2},j-1}$ for the discretization of the inertial and viscous terms and $u_{y,i-1,j-\frac{1}{2}}$, $u_{y,i-1,j+\frac{1}{2}}$, $u_{y,i,j-\frac{1}{2}}$ and $u_{y,i,j+\frac{1}{2}}$ for the viscous and augmented Lagrangian terms. In this way, we estimate the maximum values of the discretization coefficients $A_I(u)$, $1 \leq I \leq 9$ associated to velocities $u_{x,i-\frac{1}{2},j}$, $u_{x,i+\frac{1}{2},j}$, $u_{y,i,j-\frac{1}{2}}$ and $u_{y,i,j+\frac{1}{2}}$. We define:

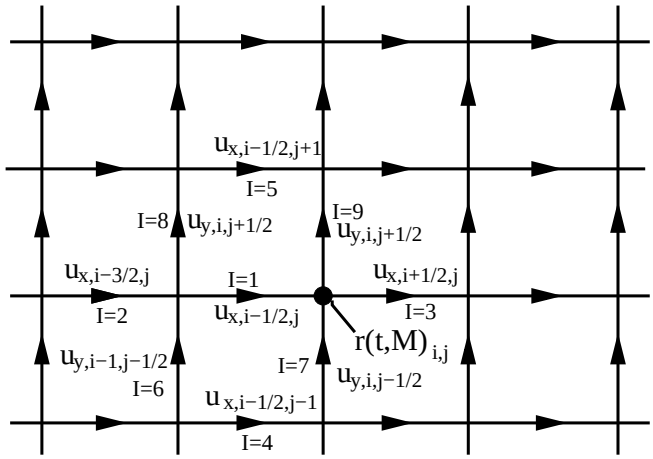


Figure A.2: Typical distribution of discretization variables on a 2D fixed staggered Cartesian grid

$$\begin{cases} C_{i-\frac{1}{2},j} = max_{I=1...9}\left[A_I(u_{x,i-\frac{1}{2},j})\right] \\ C_{i+\frac{1}{2},j} = max_{I=1...9}\left[A_I(u_{x,i+\frac{1}{2},j})\right] \\ C_{i,j-\frac{1}{2}} = max_{I=1...9}\left[A_I(u_{y,i,j-\frac{1}{2}})\right] \\ C_{i,j+\frac{1}{2}} = max_{I=1...9}\left[A_I(u_{y,i,j+\frac{1}{2}})\right] \end{cases}$$

$\otimes$ **Step 3**: In the same way, the coefficient $C^*_{i-\frac{1}{2},j}$, $C^*_{i+\frac{1}{2},j}$, $C^*_{i,j-\frac{1}{2}}$ and $C^*_{i,j+\frac{1}{2}}$, corresponding to A^*, are estimated,

$\otimes$ **Step 4**: The minimum and maximum discrete coefficient $Rmin_{i,j}$ and $Rmax_{i,j}$ at the scalar position of $r(t)_{i,j}$ are then defined as:

$$\begin{cases} Rmin_{i,j} = min(\|\frac{C_{i-\frac{1}{2},j}}{C^*_{i-\frac{1}{2},j}}\|, \|\frac{C_{i+\frac{1}{2},j}}{C^*_{i+\frac{1}{2},j}}\|, \|\frac{C_{i,j-\frac{1}{2}}}{C^*_{i,j-\frac{1}{2}}}\|, \|\frac{C_{i,j+\frac{1}{2}}}{C^*_{i,j+\frac{1}{2}}}\|) \\ Rmax_{i,j} = max(\|\frac{C_{i-\frac{1}{2},j}}{C^*_{i-\frac{1}{2},j}}\|, \|\frac{C_{i+\frac{1}{2},j}}{C^*_{i+\frac{1}{2},j}}\|, \|\frac{C_{i,j-\frac{1}{2}}}{C^*_{i,j-\frac{1}{2}}}\|, \|\frac{C_{i,j+\frac{1}{2}}}{C^*_{i,j+\frac{1}{2}}}\|) \end{cases}$$

$\otimes$ **Step 5**: If $\frac{Rmax_{i,j}}{Rmin_{i,j}} \leq 10^{10}$ and $\frac{max(\Delta_{x,i,j},\Delta_{y,i,j})}{min(\Delta_{x,i,j},\Delta_{y,i,j})} \leq 1000$, $r(t)_{i,j} = Rmin_{i,j}$. Else, $r(t)_{i,j} = Rmax_{i,j}$ unless the penalty of the incompressible or solid constraint is not ensured due to grid irregularity or strong local variation of physical or penalty parameters (at the interface between fluid and solid media for example),

$\otimes$ **Step 6**: Once $r(t)_{i,j}$ has been estimated for all i and j, $1 \leq i \leq n_x$ and $1 \leq j \leq n_y$, we normalize the local values of the algebraic penalty parameters by $r(t)_{i,j} = \frac{Kr(t)_{i,j}}{r_{min}+10^{-40}}$, where $r_{min} = min_{i=1..n_x,j=1..n_y}(r(t)_{i,j})$. The constant K is equal to 1 except at the first calculation step where $K = 100$ if $\frac{Rmax_{i,j}}{Rmin_{i,j}} \leq 10^{10}$ or if during the five first calculation steps the norm of the residual of the linear solver divided by the norm of the divergence, obtained at the previous calculation step, is grower than 10^6. This last particular case can be obtained when a flow simulation is initialized without any imposed velocity field.

Steps 1 to 6 are repeated at each calculation step corresponding to a physical time incremented of Δt seconds.

A.3.5 Solvers

A.3.5.1 Direct solvers

For 2D simulations, we have generally used the PARDISO solver of [Sche 04]. The advantage of a direct solver is to always solve a linear system with a computer error residual (of course, the matrix has to be inversible matrix). PARDISO can use more than one core for one inversion on multicore CPU. For linear systems describing a 3D problem, the direct solvers required a too large amount of memory and are generally not used.

A.3.5.2 Iterative solvers

The iterative solvers employed here is a Bi-Conjugate Gradient Stabilized [Vors 92]. Let us consider a linear system $A.x = b$ of rank N. From an initial guess x^0, iterations are performed until the residual r^k is below a chosen value

Initialization :

$$\left\| \begin{array}{l} x^0 \in \mathbb{R}^N, \\ r^0 = b - Ax^0, \\ \hat{r}^0 = r^0, \\ p^0 = r^0 \end{array} \right. \tag{A.48}$$

Itérations : For $k = 1$ to K,

$$\left\| \begin{array}{l} \alpha^{k-1} = \frac{(\hat{r}^0, r^{k-1})}{(\hat{r}^0, Ap^{k-1})}, \\ s^{k-1} = r^{k-1} - \alpha^{k-1} Ap^{k-1}, \\ \lambda^{k-1} = \frac{(As^{k-1}, s^{k-1})}{(As^{k-1}, As^{k-1})}, \\ x^k = x^{k-1} + \alpha^{k-1} p^{k-1} + \lambda^{k-1} s^{k-1}, \\ r^k = s^{k-1} - \lambda^{k-1} As^{k-1}, \\ \beta^k = \frac{\alpha^{k-1}}{\lambda^{k-1}} \frac{(\hat{r}^0, r^k)}{(\hat{r}^0, r^{k-1})}, \\ p^k = r^k + \beta^k (p^{k-1} - \lambda^{k-1} Ap^{k-1}) \end{array} \right. \tag{A.49}$$

The convergence rate of the system generally depends on the conditionner. When not precised, an incomplete LU (ILU) factorisation [Gust 78a] has been used. An ILUK factorization [Saad 86] has been generally used for the AIIB method for immersed interface cases.

A.3.6 Multiphase flows

labeltvd A wide litterature is devoted to interface tracking on Eulerian grids: the front tracking method of [Unve 92], the Volume Of Fluid method (VOF) of [Hirt 81], improved for example by [Guey 99], the level set method of [Oshe 88] and [Fedk 99] or the explicit Total Variation Decreasing (TVD) Lax-Wendroff (LW) scheme of [Vinc 99]. In our work, the interface advection is directly handled by a Piecewise Linear Interface Reconstruction PLIC VOF method of [Youn 82], which lies on a Lagrangian advection of planar pieces of interfaces. The main advantages of the PLIC-VOF approach is to be accurate for tracking tearing and stretching interfaces and to avoid numerical diffusion.

The surface tension force $\mathbf{F}_{ST} = \sigma\kappa\mathbf{n}_i\delta_i$ is modelled by a volume force proposed by [Brac 92]. The Continuous Surface Force (CSF) evaluates $\mathbf{F}_{ST}$ according to the variations of the phase function C as follows:

$$\mathbf{F}_{ST} = \sigma\nabla \cdot \left(\frac{\nabla C}{\|\nabla C\|}\right)\nabla C \tag{A.50}$$

In the CSF approach, the interface is spread on several mesh cells. The discretization of the surface tension force can be found for example in [Brac 92] or [Vinc 00].

Appendix B

Application to the image synthesis

DURING the twenty past years, image synthesis have invaded our every day life. The first step was to draw objects. First, single frames were rendered. Then, many methods such as the inverse cinematics and the keyframing have been developed to give life to more or less deformable objects. Objects were first rigid by parts, and then were fully deformable.

The image synthesis of flows is a quite recent thing. The graphical rendering is not the biggest problem since the well-known ray-tracing methods allow to render translucent materials. Animation is the big issue. The first occurrence of CG flow appears in the James Cameron blockbuster movie "Abyss" (1989). However, the fluid motion was not performed thanks to CFD but animated "by hand". Ten years after, animation of fluid was still a big deal. If we look at the recent SIGGRAPH publications, basic realistic non real-time animation of fluid flow has stopped to be itself a subject of publications only a few years ago. The new challenges concern complex fluid properties such as foams [Losa 08], multiphase flows [Losa 06], real-time optimization [Treu 06] or motion control.

Many studies have been devoted to the fluid-structure coupling for CG. In one-way methods, the interaction between fluid and solid domain is not mutual. In solid-to-fluid methods, the motions of solids are predetermined [Fost 97] and [Fost 01]. In fluid-to-solid method, the fluid can move objects, but the object do not change the motion of the fluid [Fost 96]. These

methods can be used to simulate phenomenons for which the force generated by the two domains is disproportionate.
Two way-methods consider the mutual interaction between the two domains. Some recent works have created methods for specific cases. Coupling between infinitesimally thin objects and fluid or smoke [Guen 05], coupling between breaking solids and compressible fluid in explosion [Yngv 00]. [Feld 05] have used hybrid meshes to animate gases around irregularly shaped obstacles. In [Carl 04], Carlson *et al.* have developed a new coupling method using DLM called Rigid Fluid. The objects are treated as fluids, but their velocities are constrained to be rigid motion.

The first aim of the CFD for special effects is to provide a motion that seels realistic to the viewer. The physical rightness is a good way to obtain a good aspect but is not a priority. However, we believe that a realistic flow, from a CG point of view is as well a realistic flow from a physical point of view.

Hence, the coupling between the CFD code Thétis and the CG software 3D Studio Max (Autodesk) has been performed to fulfill two objectives. First, the visual aspect of the physical simulations is not always suitable for general public. The enhancement of the graphic quality of the results is a good way to impress an audience and can make a difference. It is a good way to lead the student to the CFD. Secondly, the graphical simulation of fluid flows has both an academic and economic interest. The real-time simulation (especially on GPUs) is currently in fast expansion [Cran 07].

B.1 Global methodology

According to the type of flow, different properties or entities has to be displayed.

B.1.1 Case setting

The surface data are then read by the CFD code which uses it as interfaces for fictitious domain methods.

The flow is then computed with a constant time step to allow future image-per-image animations. According to the physical case simulated, different informations can be extracted such as velocity field, phase function, free surface, concentration etc...

B.1.2 Free surface animation

B.1.3 Strategy of Eulerian/Lagrangian grid coupling

B.1.3.1 Volume and surface data

Fundamentally, the numerical simulation codes uses volume date while CG softwares use surfacic data. The two approaches are well suited to define an interface, *e.g.* a fluid free surface. The volume approach consider an implicit representation of the interface considering a volume function, Heaviside(χ), VOF(C), level-set(ϕ), etc...The surface approach uses an explicit representation of the interface which is generally a triangularized mesh.

The volumic approach is well-suited for CFD codes while the surfacic approach is more natural in a CG software context. In fact, the rendering techniques used by the CG softwares use generally surfaces only. However, medical, geological or thermal applications such as MRI requires the representation of volume data. Hence, volume shaders are currently developed in some visualization softwares such as Avizo(SVG). The recent graphic cards have a hardware

acceleration of volume shaders. However, the CG softwares used for non-real time realistic rendering are not initially designed to treat volume data, so the volume shaders are not currently present in the CG software we use.

Hence, coupling a CFD code and a CG software implies a coupling between the two approach. The projection from a volume data to a surface data is done with the extraction of an iso-surface. If the VOF function C is considered, we consider the iso surface such as $C = 0.5$ as being the most relevant as $0 \leq C \leq 1$. The extraction can be performed directly in the CFD code. However, the implementation of a robust extractor is not simple, and the parameters of the extraction have to be set one time for all while the simulation is running. To increase the flexibility of the processus, we choose to extract the iso surfaces after the simulation using output data. The extraction is performed in the scientific visualization software Tecplot.

The algorithms used to obtain a volume data from a surface data are described in section 6.1 and are directly implemented in the CFD code Thétis.

B.1.3.2 Walkthrough

The proposed method consists in 6 distinct steps :
⊗ A scene is first designed with 3D Studio Max (Autodesk) or Blender (see Fig. B.1)

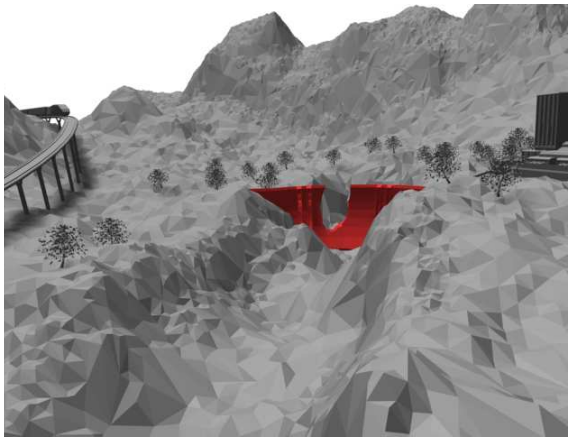


Figure B.1: Design of a 3D dam break scene.

⊗ The triangular surface elements of each object which interacts with the flow motion are then generated (see figure B.2),

⊗ The lagrangian description of the solid objects is projected onto the fixed Eulerian flow grid. Figure B.3 shows the Eulerian projection of the topography and the dam.

⊗ The characteristics of the whole fluid/solid medium are defined, such as the density and the viscosity, according to C and the unsteady flow motion (see figure B.4) is calculated with a single fluid model coupled to penalty methods for the treatment of solids.

⊗ The lagrangian iso-surfaces describing the free surface (see figure B.5) $C = 0.5$ for each fluid are generated.

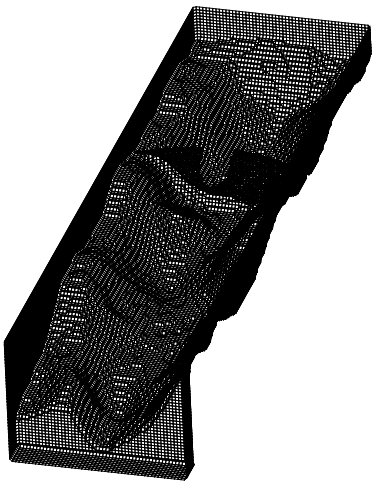


Figure B.2: Sketch of a dam topography on the Lagrangian grid extracted thanks to the 3D software.

$\otimes$ Finally, the lagrangian iso-surfaces of fluid interfaces are loaded into the 3D CG software to perform scientific visualisation or computer design. An example is provided in figure B.6.

B.1.3.3 Animation

In CG softwares, animations are created image per image. The movement of objects can be defined at each time step, or key positions can be defined for some time steps only. In this last case, the global movement is interpolated from the key where the position is explicitly defined. This method is called the keyframing. Once the animation is defined, the CG software render the animation for all the time steps. Hence, the animation of a triangularized surface can be performed by moving its elements (vertices, polygons...), a constraint being that the connectivities cannot be modified. During the simulation of a two phase simulation, the interface between the two fluids can undergoes topological modification such as coalescence or formation of droplets. For this reason, the interface cannot be considered as an unique object through the whole simulation time. As consequence, the rendering process itself has to be modified. The interface has to be reloaded at each time step as a new object. For technical reasons, the ability of the software to render an entire animation cannot be used if a new object is loaded. Instead of launching the rendering of each frame by hand, a script recreating the rendering process of an animation has been written and coupled to the script loading the iso surface.

B.1.3.4 Rendering

Many rendering methods can be used to recreate realistic fluids. The two main rendering engines of 3D Studio Max are the Scanline Renderer which is the original default renderer and Mental Ray. The first one is fast, easy to configure and give good results. Refraction, reflection and radiosity are available. Mental ray is harder to configure and slower but can simulate caustic phenomena and gives a more realistic rendering. However, problems have been encountered with mental ray when rendering iso surfaces of poor quality (self intersecting or non closed surfaces). In this case, the light rays trajectory can change drastically from one frame to an other generating

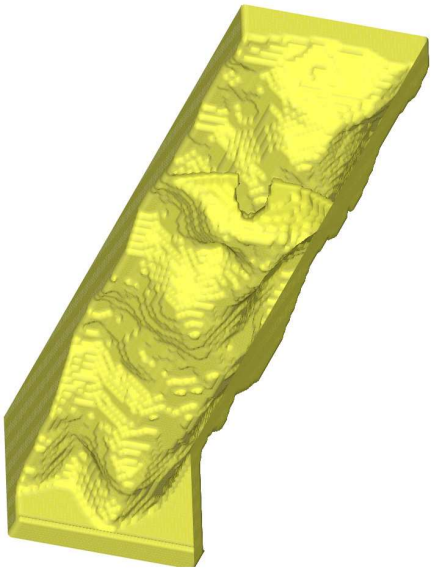


Figure B.3: Sketch of a dam topography on the eulerian grid after equation (6.1) solving.

a blinking or the refracted image.

B.1.4 Moving complex objects

Contrary to free surface flow, the moving mesh is not deformable, so only the position and rotation angle of each objects is written for some regular time step in the exchange file. Each object is then created in the CG software. Its global trajectory is interpolated from the informations of the exchange file.

B.1.5 Passively advected particles

The main difference with the last case is that the particle rotation is not considered, the number of particles can be huge and particles can be created during the time. Two approaches have been experimented :

- The movement of the particles is computed in the CFD code. At each time step, the position of all particles is written in a file. Once the simulation is complete, the rendering of the animation is performed in a quite similar way as in (B.1.2). For each frame, the file containing the position of the particles is loaded and the particles are generated, then rendered and destroyed. The main drawback of this method is the size of the files containing the particles positions.

- The particles are directly created in the CG software from the velocity field. This approach is more flexible as the injector position or the particle amount can be set in the CG software. However, a particle generation and advection system must be entirely rewritten. 3D Studio Max has a particle system which can be highly scripted. Unfortunately, the scriptable particle system was highly unstable in the version 7 of 3D Studio Max. The other drawback is again the size of the output required files if the velocity field is unsteady and needed at each time step.

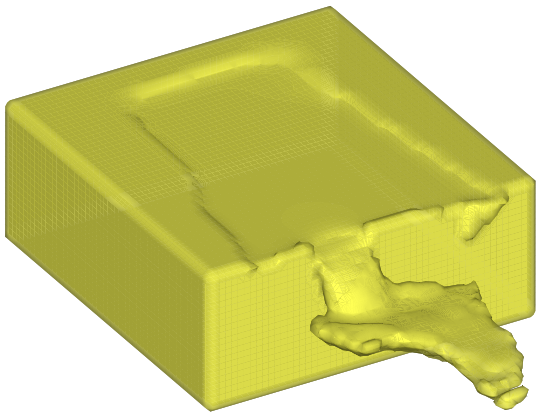


Figure B.4: Topology of the Eulerian water-air surface during dam break.

B.2 Results

B.2.1 Dam break over letters

In a 3D tank, a column of viscous water ($\mu = 0.01Pa.s$ and $\rho = 1000kg.m^{-3}$), initially at rest, breaks over a fixed obstacle of 'Aquilon' letters shape in an air medium ($\mu = 1.85 \cdot 10^{-5}Pa.s$ and $\rho = 1.1768kg.m^{-3}$). As demonstrated in figure B.7, the flow is turbulent and the surface tension ($\sigma = 0.075N.m^{-1}$) is taken into account. This still academic problem is interesting as it involves complex interfacial structures as well as strong interactions between the free surface and the obstacle. In particular, the dynamics of gaz pocket rupture and coalescence can be observed.

B.2.2 Dam break over a realistic topologie

As last illustration test case, a real dam break flow is considered in order to show the simplicity and power of the Aulerian/Lagrangian coupling associated to penalty methods. The scene presented in figures B.1 and B.6 is calculated considering real water and air and a dam of almost 30 meter height. Figure B.8 illustrates the potential of the DNS simulation associated to 3D softwares for movie or video games design.

B.2.3 Particulate flows

The first illustration of the interest of the coupled Eulerian/lagrangian grid technique associated to penalty methods is related to the sedimentation of 147 particles in a liquid tank. All the particle/particle and particle/wall interactions are solved, with an explicit modeling if the local mesh refinement is not enough. A serie of particle motion pictures is proposed in figure B.9.

B.2.4 Flow around a tire

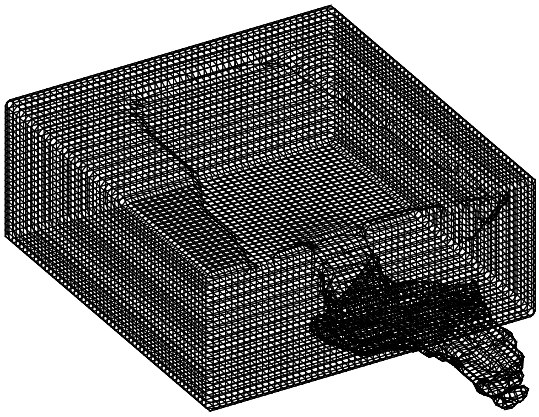


Figure B.5: Topology of the Lagrangian water-air surface during dam break after Level Contour Reconstruction.

Figure B.6: Final vue after the rendering process of the dam break.

Figure B.7: Dam break flow over a complex obstacle.

Figure B.8: Real dam break flow over a complex topography.

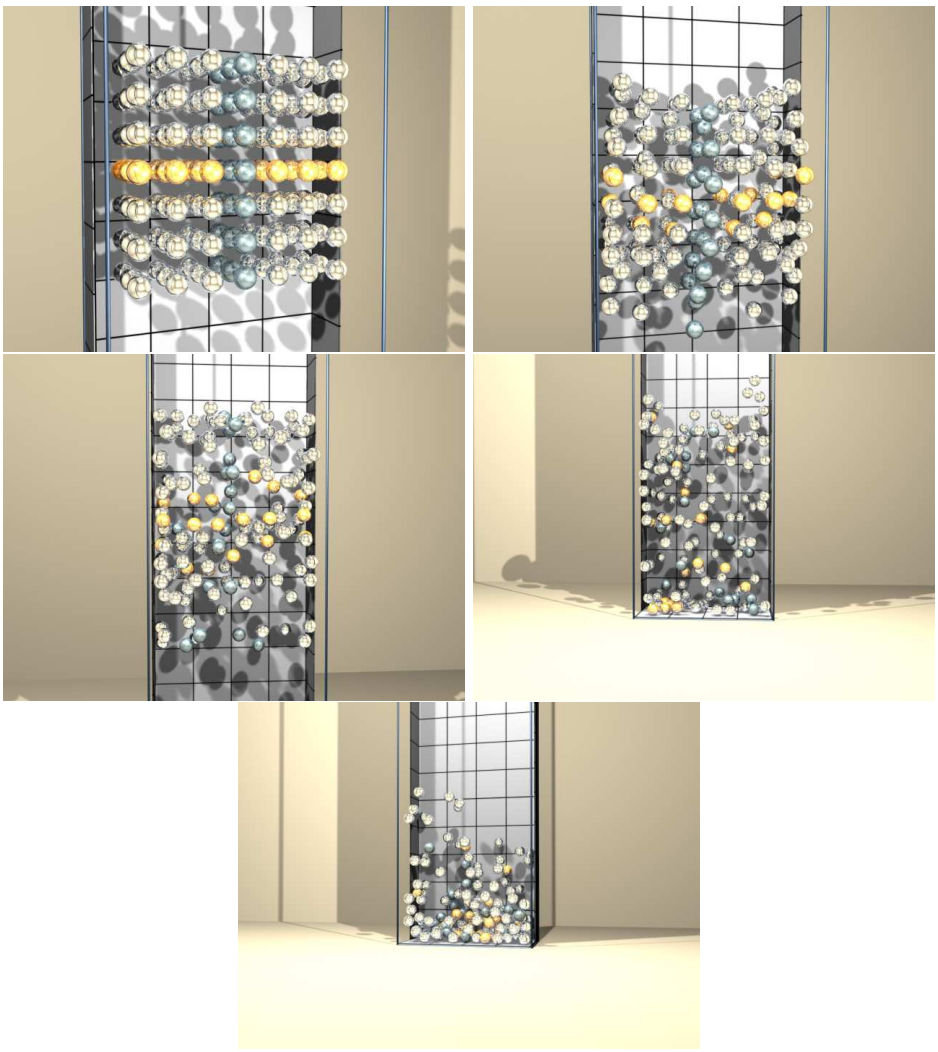


Figure B.9: Rigid particle sedimentation in water.

Appendix C

Interpolations

The interpolations used for the present work are described.

C.1 Polynomial interpolations

The interpolations polynomials are first denoted according to their support. In 2D, the support of a $\mathbb{L}$ interpolation is a line, the support of $\mathbb{P}$ is a triangle and the support of $\mathbb{Q}$ is a rectangle. $\mathbb{L}_D^d$ indicates that the interpolation is in dimension d and of order D. For instance:

$$L_1(x) = ax + b \tag{C.1}$$

$$P_1^2(x) = ax + by + c \tag{C.2}$$

$$Q_1^2(x) = axy + bx + cy + d \tag{C.3}$$

C.1.1 Construction of the $\mathbb{Q}_1^2$ element

The considered grid cell is mapped to a $D = [0,1] \times [0,1]$ cell. For the penalty methods, the penalized point is the origin. The $\mathbb{Q}_1^2$ element is defined by

$$Q_1(x,y) = ax + by + cxy + d \tag{C.4}$$

with a, b, c and d scalar coefficient determined with the following constraints:

$$\begin{cases} Q_1(0,0) = u_C \\ Q_1(1,0) = u_E \\ Q_1(0,1) = u_N \\ Q_1(1,1) = u_{NE} \end{cases} \tag{C.5}$$

where the indices of u indicate the position of a node according to its direction (E is east, N is north, etc...). We obtain the following expression for the Q_1 :

$$Q_1(x,y) = (-u_C + u_E)x + (-u_C + u_N)y + (-u_C + u_E + u_N - u_{NE})xy + u_C \tag{C.6}$$

one can write

$$Q_1(x,y) = u_1(1 - x - y - xy) + u_2(x + xy) + u_3(y + xy) + u_4(-xy) \tag{C.7}$$

and the coefficient for each u_i is then deducted.

C.1.2 Construction of the $\mathbb{Q}_2^2$ element

For a Q_2^2 interpolation, the function is

$$Q_2(x,y) = ax^2y^2 + bx^2y + cxy^2 + dx^2 + ey^2 + fxy + gx + hy + k. \tag{C.8}$$

The following constraints are considered:

$$\begin{cases} Q_2(0,0) = u_C \\ Q_2(-1,0) = u_W \\ Q_2(1,0) = u_E \\ Q_2(0,-1) = u_S \\ Q_2(0,1) = u_N \\ Q_2(-1,-1) = u_{SW} \\ Q_2(1,-1) = u_{SE} \\ Q_2(-1,1) = u_{NW} \\ Q_2(1,1) = u_{NE} \end{cases} \tag{C.9}$$

We obtain a linear combination such as

$$coef_i = \alpha_i \sum C_i u_i \tag{C.10}$$

with the following coefficients:

$Coef$	facteur	u_C	u_W	u_E	u_S	u_N	u_{SW}	u_{SE}	u_{NW}	u_{NE}
a	1/4	4	−2	−2	−2	−2	1	1	1	1
b	1/4	0	0	0	−2	−2	−1	1	−1	1
c	1/4	0	−2	−2	0	0	−1	−1	1	1
d	1/2	−2	1	1	0	0	0	0	0	0
e	1/2	−2	0	0	1	1	0	0	0	0
f	1/4	0	0	0	0	0	1	−1	−1	1
g	1/2	0	−1	1	0	0	0	0	0	0
h	1/2	0	0	0	−1	1	0	0	0	0
k	1	1	0	0	0	0	0	0	0	0

Table C.1: Coefficient for the Q_2 interpolation

The resulting function is

$$\begin{aligned}
Q_2(x_0,y_0) &= u_c\big(x_0^2y_0^2 - x_0^2 - y_0^2 + 1\big)\\
&+u_W\frac{1}{2}\big(-x_0^2y_0^2 - x_0y_0^2 + x_0^2 - x_0\big)\\
&+u_E\frac{1}{2}\big(-x_0^2y_0^2 - x_0y_0^2 + x_0^2 + x_0\big)\\
&+u_S\frac{1}{2}\big(-x_0^2y_0^2 - x_0^2y_0 + y_0^2 - y_0\big)\\
&+u_N\frac{1}{2}\big(-x_0^2y_0^2 - x_0^2y_0 + y_0^2 + y_0\big)\\
&+u_{SW}\frac{1}{4}\big(x_0^2y_0^2 - x_0^2y_0 - x_0y_0^2 + x_0y_0\big)\\
&+u_{SE}\frac{1}{4}\big(x_0^2y_0^2 + x_0^2y_0 - x_0y_0^2 - x_0y_0\big)\\
&+u_{NW}\frac{1}{4}\big(x_0^2y_0^2 - x_0^2y_0 + x_0y_0^2 - x_0y_0\big)\\
&+u_{NE}\frac{1}{4}\big(x_0^2y_0^2 + x_0^2y_0 + x_0y_0^2 + x_0y_0\big).
\end{aligned} \tag{C.11}$$

C.2 Kernel functions

The kernel function is a weighting function used in nonparametric function estimation. It gives the weights of the nearby data points in making an estimate. They can be probability density functions.

Compared to classical interpolations, the formulation of the kernel functions allows any stencil to be used. Let us consider a quantity ϕ which has to be interpolated from nodes $\mathbf{x}_j$ to a node $\mathbf{x}_i$. The kernel function method gives

$$\phi_i = \frac{1}{n_i}\sum_{i\neq j} w(|r_i - r_j|)\phi_j \tag{C.12}$$

with

$$n_i = \sum_{i\neq j} w(|r_i - r_j|) \tag{C.13}$$

and w a weighting function. These functions are generally piecewise continuous, bounded, symmetric around zero, concave at zero, real valued, and for convenience often integrate to one. For instance, we have

$$w(r) = \begin{cases} 1 - 6\dfrac{r}{r_e}^2 + 8\dfrac{r}{r_e}^3 + 3\dfrac{r}{r_e}^4 & 0 \leqslant r \leqslant r_e \\ 0 & r_e \leqslant r \end{cases} \tag{C.14}$$

with r_e a cutoff radius. Many weighting functions are presented in [Atai 06, RM 08] where authors applied kernel functions to a smoothed particle hydrodynamics (SPH) method.

Bibliography

[Adam 99] J. Adams, P. Swartztrauber, and R. Sweet. "FISHPACK: efficient FORTRAN subprograms for the solution of separable eliptic partial differential equations". 1999.

[Akse 96] A. Aksenov, A. Dyadkin, and A. Gudzovsky. "Numerical Simulation of Car Tire Aquaplaning". *ECCOMAS 96, Paris*, Vol. , p. , 1996.

[Ango 03] P. Angot. "A model of fracture for elliptic problems with flux and solution jumps". *C. R. Math. Acad. Sci. Paris*, Vol. 337, No. 6, pp. 425 – 430, 2003.

[Ango 05] P. Angot. "A unified fictitious domain model for general embedded boundary conditions". *C. R. Math. Acad. Sci. Paris*, Vol. 341, No. 11, pp. 683 – 688, 2005.

[Ango 89] P. Angot. *Contribution à l'étude des transferts thermiques dans des systèmes complexes aux composants électroniques.* PhD thesis, Université Bordeaux I, 1989.

[Ango 90] P. Angot and J. Caltagirone. "". In: *Proceedings of the 2nd World Congress on Computational Mechanics, Stuttgart*, pp. 973–970, 1990.

[Ango 99] P. Angot, C. Bruneau, and P. Fabrie. "A penalization method to take into account obstacles in incompressible viscous flows". *Numerische Mathematik*, Vol. 81, pp. 497–520, 1999.

[Arch 04] F. Archambeau, N. Méchitoua, and M. Sakiz. "Code_Saturne : a Finite Volume Code for the Computation of Turbulent Incompressible Flows & Industrial Applications". *International Journal on Finite Volumes*, Vol. 1, 2004.

[Arqu 84] E. Arquis. *Convection mixte dans une couche poreuse verticale non confinée. Application à l'isolation perméodynamique.* PhD thesis, Université Bordeaux I, 1984.

[Atai 06] B. Ataie-Ashtiani and L. Farhadi. "A stable moving-particle semi-implicit method for free surface flows". *Fluid Dynamics Research*, Vol. 38, No. 4, pp. 241 – 256, 2006.

[Baer 05] J. A. Bærentzen and H. Aanæs. "Generating Signed Distance Fields From Triangle Meshes". 2005. IMM-TECHNICAL REPORT-2002-21.

[Bert 97] F. Bertrand, P. Tanguy, and F. Thibault. "A three-dimensional fictious domain method for incompressible fluid flow problems". *Int. J. Numer. Meth. Fluids*, Vol. 25, pp. 716–736, 1997.

[Bird 77] R. B. Bird, R. C. Armstrong, and O. Hassager. *Dynamics of Polymeric liquids : Volume I Fluid Mechanics.* John Wiley and Sons, 1977.

[Boir 09] O. Boiron, G. Chiavassa, and R. Donat. "A high-resolution penalization method for large Mach number flows in the presence of obstacles". *Computers and Fluids*, Vol. 38, No. 3, pp. 703 – 714, 2009.

[Bost 08a] C. Bost, G.-H. Cottet, and E. Maitre. "Numerical analysis of a penalization method for the two-way coupling of an incompressible flow with a rigid body". 2008. Submitted.

[Bost 08b] C. Bost. *Mt́hodes Level-Set et pénalisation pour le calcul d'interactions fluide-structure*. PhD thesis, Université Joseph Fourier - Grenoble I, 2008.

[Brac 92] J. U. Brackbill, D. B. Kothe, and C. Zemach. "A continuum method for modeling surface tension". *Journal of Computational Physics*, Vol. 100, No. 2, pp. 335 – 354, 1992.

[Braz 86] M. Braza, P. Chassaing, and H. Minh. "Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder". *Journal of Fluid Mechanics*, Vol. 165, pp. 79–130, 1986.

[Calt 01] J. Caltagirone and S. Vincent. "Sur une méthode de pénalisation tensorielle pour la résolution des équations de Navier-Stokes". *C. R. Acad. Sci. Paris*, Vol. 329, Série IIb, pp. 607–613, 2001.

[Calt 86] J. Caltagirone and E. Arquis. "Recirculating flow in porous media". *C. R. Acad. Sci. Paris*, Vol. 302, Série IIb, No. 14, pp. 843–846, 1986.

[Calt 94] J. Caltagirone. "Sur l'interaction fluide-milieu poreux, application au calcul des efforts exercés sur un obstacle par un fluide visqueux". *Comptes Rendus de l'Académie des Sciences de Paris. Série II b*, Vol. 318, pp. 571–577, 1994.

[Calt 99] J. Caltagirone and J. Breil. "Sur une méthode de projection vectorielle pour la résolution des équations de Navier-Stokes". *C. R. Acad. Sci. Paris*, Vol. 327, pp. 1179 – 1184, 1999.

[Carl 04] M. Carlson, P. J. Mucha, and G. Turk. "Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid". In: *ACM SIGGRAPH 2004*, pp. 377–384, 2004.

[Chaw 92] J. Chawner and J. Steinbrenner. "Recent enhancements for the GRIDGEN structured grid generation system". *it NASA CP-3143 - Software Systems for Surface Modeling and Grid Generation, Langley*, 1992.

[Cho 06] R. Cho, H. Lee, J. Sohn, G. Kim, and J. Woo. "Numerical investigation of hydroplaning characteristics of three dimensional patterned tire". *Eur. J. Mech. A Solid*, Vol. 25, pp. 914–926, 2006.

[Clar 86] D. K. Clarke, H. Hassan, and M. Salas. "Euler calculations for multielement airfoils using Cartesian grids". *AIAA Journal*, Vol. 24, pp. 353 – 358, 1986.

[Coqu 08] M. Coquerelle and G.-H. Cottet. "A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies". *J. Comput. Phys.*, Vol. 227, No. 21, pp. 9121–9137, Nov. 2008.

[Cott 04] G. H. Cottet and P. Poncet. "Advances in direct numerical simulations of 3D wall-bounded flows by Vortex-in-Cell methods". *Journal of Computational Physics*, Vol. 193, No. 1, pp. 136 – 158, 2004.

[Coud 07] F. Couderc. *Développement d'un code de calcul pour la simulation d'écoulements de fluides non miscibles. Application à la dśintǵration assisté d'un jet liquide par un courant gazeux.* PhD thesis, ENSAE - Toulouse, 2007.

[Coud 09] F. Couderc, D. Legendre, J. Bera, and B. Gilles. "Simulation numérique du collapse et de l'implosion ultrasonor de bulles de cavitation". *Proceeding du 19éme Congrès Français de Mécanique, Marseille*, 2009.

[Cout 77] M. Coutanceau and R. Bouard. "Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow". *Journal of Fluid Mechanic*, Vol. 79, pp. 231–256, 1977.

[Cran 07] K. Crane, I. Llamas, and S. Tariq. "Real time simulation and rendering of 3D fluids". *GPU Gems*, Vol. 3, pp. 633–675, 2007.

[Dela 06] S. Delage Santacreu. *Méthode de raffinement de maillage adaptatif pour la mécanique des fluides numérique.* PhD thesis, Universit'e Bordeaux I, 2006.

[Delh 74] J. Delhaye. "Jump conditions and entropy sources in two-phase systems. Local instant formulation". *Int. J. Multiphase Flow*, Vol. 1, No. 3, pp. 395–409, 1974.

[Denn 70] S. Dennis and G. Chang. "Numerical solutions for steady flow post a circular cylinder at Reynolds number up to 100". *Journal of Fluid Mechanics*, Vol. 42, p. 471, 1970.

[Desc 04] T. Deschamps, P. Schwartz, D. Trebotich, P. Colella, D. Saloner, and R. Malladi. "Vessel segmentation and blood flow simulation using Level-Sets and Embedded Boundary methods". *International Congress Series*, Vol. 1268, No. , pp. 75 – 80, 2004. CARS 2004 - Computer Assisted Radiology and Surgery. Proceedings of the 18th International Congress and Exhibition.

[Desj 08] O. Desjardins, V. Moureau, and H. Pitsch. "An accurate conservative level set/ghost fluid method for simulating turbulent atomization". *Journal of Computational Physics*, Vol. 227, No. 18, pp. 8395 – 8416, 2008.

[Dome 08] F. Domenichini. "On the consistency of the direct forcing method in the fractional step solution of the Navier-Stokes equations". *Journal of Computational Physics*, Vol. 227, pp. 6372–6384, 2008.

[Fadl 00] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations". *Journal of Computational Physics*, Vol. 161, No. 1, pp. 35 – 60, 2000.

[Fedk 99] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. "A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method)". *Journal of Computational Physics*, Vol. 152, No. 2, pp. 457 – 492, 1999.

[Feit 97] F. R. Feito and J. C. Torres. "Inclusion test for general polyhedra". *Computers & Graphics*, Vol. 21, No. 1, pp. 23–30, 1997.

[Feld 05] B. Feldman, J. O. Brien, and B. Klingner. "Animating Gases with Hybrid Meshes". In: *Proceedings of ACM SIGGRAPH 2005*, 2005.

[Fort 82] M. Fortin and R. Glowinski. *Méthodes de lagrangien augmenté. Application à la résolution numérique de problèmes aux limites.* Dunod Paris, 1982.

[Fost 01] N. Foster and R. Fedkiw. "Practical Animation of Liquids". In: *Proceedings of ACM SIGGRAPH 2001*, pp. 15–22, 2001.

[Fost 96] N. Foster and D. Metaxas. "Realistic animation of liquids". In: *PGraphical Models And Image Processing 58*, pp. 471–483, 1996.

[Fost 97] N. Foster and D. Metaxas. "Controlling fluid animation". In: *Proceedings CGI97*, pp. 178–188, 1997.

[Frie 04] H. Friess, D. Lakehal, and S. Vincent. "Test case n^o 27: interface tracking based on an imposed velocity field in a convergent-divergent channel (PN)". *Multiphase Science and Technology*, Vol. 16, pp. 163–168, 2004.

[Gibo 02] F. Gibou, R. P. Fedkiw, L. Cheng, and M. Kang. "A Second-Order-Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains". *Journal of Computational Physics*, Vol. 176, No. 1, pp. 205–227, Feb. 2002.

[Gibo 05] F. Gibou and R. Fedkiw. "A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem". *Journal of Computational Physics*, Vol. 202, No. 2, pp. 577–601, 2005.

[Glow 01] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Périaux. "A Fictitious Domain Approach to the Direct Numerical Simulation of Incompressible Viscous Flow past Moving Rigid Bodies: Application to Particulate Flow". *Journal of Computational Physics*, Vol. 169, No. 2, pp. 363 – 426, 2001.

[Glow 99] R. Glowinski, T. W. Pan, T. I. Hesla, and D. D. Joseph. "A distributed Lagrange multiplier/fictitious domain method for particulate flows". *International Journal of Multiphase Flow*, Vol. 25, No. 5, pp. 755 – 794, 1999.

[Goda 78] K. Goda. "A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows". *Journal of Computational Physics*, Vol. 30, pp. 76 – 95, 1978.

[Grif 05] B. E. Griffith and C. S. Peskin. "On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems". *Journal of Computational Physics*, Vol. 208, No. 1, pp. 75 – 105, 2005.

[Guen 03] E. Guendelman, R. Bridson, and R. Fedkiw. "Nonconvex rigid bodies with stacking". *ACM SIGGRAPH*, Vol. 22, pp. 871–878, 2003.

[Guen 05] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw. "Coupling Water and Smoke to Thin Deformable and Rigid Shells". In: *Proceedings of ACM SIGGRAPH 2005*, pp. 973–981, 2005.

[Guer 06] J. Guermond, P. Minev, and J. Shen. "An overview of projection methods for incompressible flows". *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 44-47, pp. 6011 – 6045, 2006.

[Guey 99] D. Gueyffier, J. Li, A. Nadim, S. Scardovelli, and S. Zaleski. "Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows". *Journal of Computational Physics*, Vol. 152, pp. 423–456, 1999.

[Gust 78a] I. Gustafsson. *On First and Second Order Symmetric Factorization Methods for the Solution of Elliptic Difference Equations. Research report, Department of Computer Sciences, ChalmersUniversity of Technology and the University of Gateborg*, la, 1978.

[Gust 78b] I. Gustafsson. *On First and Second Order Symmetric Factorization Methods for the Solution of Elliptic Difference Equations. Research report, Department of Computer Sciences, ChalmersUniversity of Technology and the University of Gateborg*, 1978.

[Hao 09] J. Hao, T.-W. Pan, R. Glowinski, and D. D. Joseph. "A fictitious domain/distributed Lagrange multiplier method for the particulate flow of Oldroyd-B fluids: A positive definiteness preserving approach". *Journal of Non-Newtonian Fluid Mechanics*, Vol. 156, No. 1-2, pp. 95 – 111, 2009.

[Happ 63] J. Happel and H. Brenner. *Low Reynolds Number Hydrodynamics*. Kluwer Academic Publishers, 1963.

[Harl 65] F. H. Harlow and J. E. Welch. "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface". *The Physics of Fluids*, Vol. 8(12), pp. 2182 – 2189, 1965.

[He 97] X. He and G. Doolen. "Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder". *Journal of Computational Physics*, Vol. 134, pp. 306–315, 1997.

[Hirt 81] C. Hirt and B. Nichols. "Volume of fluid (vof) methods for the dynamics of free boundaries". *Journal of Computational Physics*, Vol. 39, pp. 201–225, 1981.

[Hopp 02] F. C. Hoppensteadt and C. S. Peskin. *Modeling and simulation in medicine and the life sciences*. Springer-Verlag New-York, 2002.

[Hu 92] H. Hu, D. Joseph, and M. Crochet. "Direct simulation of fluid particle motion". *Theoretical Computational Fluid Dynamics*, Vol. 3, pp. 285–306, 1992.

[Huan 07] J. Huang, P. M. Carrica, and F. Stern. "Coupled ghost fluid/two-pahse level set method for curvilinear body-fitter grids". *International Journal for Numerical Methods in Fluids*, Vol. 55, pp. 867–897, 2007.

[Iken 07] T. Ikeno and T. Kajishima. "Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations". *Journal of Computational Physics*, Vol. 226, No. 2, pp. 1485–1508, 2007.

[Jang 08] W. Jang, J. Jilesen, F. S. Lien, and H. Ji. "A study on the extension of a VOF/PLIC based method to a curvilinear co-ordinate system". *International Journal of Computational Fluid Dynamics*, Vol. 22, pp. 241–257, 2008.

[Jian 00] G.-S. Jiang and D. Peng. "Weighted ENO schemes for hamilton-jacobi equations". *Journal of Scientific Computing*, Vol. 21, pp. 2126–2143, 2000.

[Joha 98] H. Johansen and P. Colella. "A Cartesian Grid Embedded Boundary Method for Poisson's Equation on Irregular Domains". *Journal of Computational Physics*, Vol. 147, No. 1, pp. 60 – 85, 1998.

[John 96] A. Johnson and T. Tezduyar. "Simulation of multiple spheres falling in a liquid filled tube". *Computer Methods in Applied Mechanics and Engineering*, Vol. 134, pp. 351–373, 1996.

[Jone 05] M. W. Jones. "3D Distance from a Point to a Triangle". 2005.

[Jone 06] M. W. Jones, J. A. Bærentzen, and M. Sramek. "3D Distance Fields: A Survey of Techniques and Applications". *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, No. 4, pp. 581–599, 2006.

[Kang 00] M. Kang, R. P. Fedkiw, and X.-D. Liu. "A Boundary Condition Capturing Method for Multiphase Incompressible Flow". *Journal of Scientific Computing*, Vol. 15, pp. 323–360, 2000.

[Kata 86] I. Kataoka. "Local instant formulation of two-phase flow". *Int. J. Multiphase Flow*, Vol. 12, No. 5, pp. 745–758, 1986.

[Keet 07] G. Keetels, U. D'Ortona, W. Kramer, H. Clercx, K. Schneider, and G. van Heijst. "Fourier spectral and wavelet solvers for the incompressible Navier-Stokes equations with volume-penalization: Convergence of a dipole-wall collision". *Journal of Computational Physics*, Vol. 227, No. 2, pp. 919 – 945, 2007.

[Khad 00] K. Khadra, P. Angot, S. Parneix, and J. Caltagirone. "Fictitious domain approach for numerical modelling of Navier-Stokes equations". *International Journal for Numerical Methods in Fluids*, Vol. 34, pp. 651–684, 2000.

[Kim 01] J. Kim, D. Kim, and H. Choi. "An immersed boundary finite volume method for simulations of flow in complex geometries". *Journal of Computational Physics*, Vol. 171, pp. 132–150, 2001.

[Labo 07] E. Labourasse, D. Lacanette, A. Toutant, P. Lubin, S. Vincent, O. Lebaigue, J. Caltagirone, and P. Sagaut. "Detailed comparisons of front-capturing methods for turbulent two-phase flow simulations". *Int. J. Mult. Flow*, Vol. 33, pp. 1–39, 2007.

[Laca 06] D. Lacanette, A. Gosset, S. Vincent, J. Buchlin, and E. Arquis. "Macroscopic analysis of gas-jet wiping: Numerical simulation and experimental approach". *Phys. Fluid*, Vol. 18, pp. 1–15, 2006.

[Laca 09] D. Lacanette, S. Vincent, A. Sarthou, P. Malaurent, and J. Caltagirone. "An Eulerian-Lagrangian method for the numerical simulation of incompressible convection flows interacting with complex obstacles: Application to the natural convection in the Lascaux cave". *International Journal of Heat and Mass Transfer*, Vol. 52, pp. 2528–2542, 2009.

[Lai 00] M. Lai and C. S. Peskin. "An immersed boundary method with formal second order accuracy and reduced numerical viscosity". *Journal of Computational Physics*, Vol. 160, pp. 705 – 719, 2000.

[Lamb 09] J.-P. Lambelin, F. Nadal, R. Lagrange, and A. Sarthou. "Non-resonant viscous theory for the stability of a fluid-filled gyroscope". *Journal of Fluid Mechanics*, Vol. 639, pp. 167–194, 2009.

[Laro 08] J. Larocque. *Modélisation et simulation numérique dâĂŹécoulements incompressibles turbulents diphasiques à phases non miscibles : application à lâĂŹinteraction dâĂŹun jet turbulent avec une surface libre dans une cavité.* PhD thesis, Université Bordeaux I, 2008.

[Laro 09] J. Larocque, N. Rivière, S. Vincent, D. Reungoat, J. Fauré, J. Héliot, and J. Caltagirone. "Macroscopic analysis of a turbulent round liquid jet impinging on an air/water interface in a confined medium". *accepted in Phys. Fluid*, Vol. , p. , 2009.

[Le 06] D. Le, B. Khoo, and J. Peraire. "An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries". *Journal of Computational Physics*, Vol. 220, pp. 109–138, 2006.

[Leba 04] O. Lebaigue, C. Ducquennoy, and S. Vincent. "TEST-CASE NO 1: RISE OF A SPHERICAL CAP BUBBLE IN A STAGNANT LIQUID (PN)". *Mult. Phase sci. Tech.*, Vol. 6, No. , pp. 1–4, 2004.

[Lefe 07] A. Lefebvre. *Modélisation numérique d'écoulements fluide/particules.* PhD thesis, Université Paris XI, 2007.

[Leve 94] R. J. Leveque and Z. Li. "The immersed interface method for elliptic equations with discontinuous coefficients and singular sources". *SIAM Journal of Numerical Analysis*, Vol. 31, pp. 1001–1025, 1994.

[Leve 97] R. Leveque and Z. Li. "Immersed interface methods for Stokes flow with elastic boundaries of surface tension". *SIAM Journal of Scientific Computing*, Vol. 18, pp. 709–735, 1997.

[Li 01] Z. Li and M. Lai. "The immersed interface method for the Navier-Stokes equations with singular forces". *Journal of Computational Physics*, Vol. 171, pp. 822–842, 2001.

[Li 03] Z. Li and C. Wang. "A fast finite difference method for solving Navier-Stokes equations on irregular domains". *Communications in Mathematical Sciences*, Vol. 1, pp. 180–196, 2003.

[Li 04] Z. Li, K. Ito, and M. Lai. "An augmented approach for Stokes equations with a discontinuous viscosity and singular forces". NCSU-CRSC Tech. Report: CRSC-TR04-23, North Carolina State University, 2004.

[Li 06] Z. Li and K. Ito. *The immersed interface method. Numerical solutions of PDEs involving interfaces and irregular domains.* SIAM - Frontiers in Applied Mathematics, 2006.

[Li 94] Z. Li. *The immersed interface method - A numerical approach for partial differential equations with interfaces.* PhD thesis, University of Washington, 1994.

[Li 98] Z. Li. "A fast iterative algorithmfor elliptic interfaces problems". *SIAM Journal of Numerical Analysis*, Vol. 35, pp. 230–254, 1998.

[Li 99] Z. Li, H. Zhao, and H. Gao. "A numerical study of electro-migration voiding by evolving level set functions on a fixed Cartesian grid". *Journal of Computational Physics*, Vol. 152, pp. 281–304, 1999.

[Linn 05] M. Linnick and H. Fasel. "A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains". *Journal of Computational Physics*, Vol. 204, pp. 157–192, 2005.

[Liu 00] X.-D. Liu, R. P. Fedkiw, and M. Kang. "A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains". *Journal of Computational Physics*, Vol. 160, No. 1, pp. 151 – 178, 2000.

[Liu 03] X. Liu and T. Sideris. "Convergence of the ghost fluid method for elliptic equations with interfaces". *Mathematics of computation*, Vol. 72, pp. 1731–1746, 2003.

[Liu 98] C. Liu, X. Sheng, and C. Sung. "Preconditioned multigrid methods for unsteady incompressible flows". *Journal of Computational Physics*, Vol. 139, pp. 35–57, 1998.

[Losa 06] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw. "Multiple interacting liquids". *ACM SIGGRAPH*, Vol. 25, pp. 812–819, 2006.

[Losa 08] F. Losasso, J. Talton, N. Kwatra, and R. Fedkiw. "Two-way Coupled SPH and Particle Level Set Fluid Simulation". *IEEE TVCG*, Vol. 14, pp. 797–804, 2008.

[Lubi 06] P. Lubin, S. Vincent, S. Abadie, and J. Caltagirone. "Three-dimensional Large Eddy Simulation of air entrainment under plunging breaking waves". *Coastal Eng.*, Vol. 53, pp. 631–655, 2006.

[Maju 01] S. Majumdar, G. Iaccarino, and P. Durbin. "RANS solvers with adaptive structured boundary non-conforming grids". Annual research brief, NASE Center for Turbulence Reseach, 2001.

[Mark 08] A. Mark and B. G. Wachem. "Derivation and validation of a novel implicit second-order accurate immersed boundary method". *Journal of Computational Physics*, Vol. 227, pp. 6660–6680, 2008.

[Masa] K. Masataka and O. Toshihiko. "Hydroplaning simulation using msc.dytran". *MSC Publication*, Vol. , p. .

[Maun 08] A. Maunoury. *Simulations numériques de lâĂŹascension dâĂŹune particule évolutive sous lâĂŹeffet du changement dâĂŹétat liquide-solide.* PhD thesis, Universit'e Bordeaux I, 2008.

[Maur 01] B. Maury. "A Fat Boundary Method for the Poisson Equation in a Domain with Holes". *Journal of Scientific Computing*, Vol. 16, pp. 319–339, 2001.

[McCo 01] P. McCorquodale, P. Colella, and H. Johansen. "A Cartesian Grid Embedded Boundary Method for the Heat Equation on Irregular Domains". *Journal of Computational Physics*, Vol. 173, No. 2, pp. 620 – 635, 2001.

[Mill 05] L. A. Miller and C. S. Peskin. "A computational fluid dynamics study of 'clap and fling' in the smallest insects". *Journal of Experimental Biology*, Vol. 208-2, pp. 195 – 212, 2005.

[Mirt 96] B. Mirtich. "Fast and accurate computation of polyhedral mass properties". *Journal of Graphics, GPU and Game Tools*, Vol. 1, pp. 31–50, 1996.

[Mitt 05] R. Mittal and G. Iaccarino. "Immersed boundary methods". *Annual Review in Fluid Mechanics*, Vol. 37, pp. 239 – 261, 2005.

[Mohd 97] J. Mohd-Yusof. "Combined immersed boundary/B-spline methods for simulations of flows in complex geometries". Tech. Rep., NASA ARS/Stanford University CTR, 1997.

[Mura 06] M. Muradoglu and A. D. Kayaalp. "An auxiliary grid method for computations of multiphase flows in complex geometries". *Journal of Computational Physics*, Vol. 214, No. 2, pp. 858 – 877, 2006.

[Ogay 05] C. J. Ogayar, R. J. Segura, and F. R. Feito. "Point in solid strategies". *Computers & Graphics*, Vol. 29, No. 4, pp. 616–624, 2005.

[Oshe 01] S. Osher and R. Fedkiw. "Level Set Methods: An Overview and Some Recent Results". *J. Comput. Phys.*, Vol. 169, pp. 463–502, 2001.

[Oshe 88] S. Osher and J. A. Sethian. "Front propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations". *Journal of Computational Physics*, Vol. 79, pp. 12–49, 1988.

[Pacc 05] A. Paccou, G. Chiavassa, J. Liandrat, and K. Schneider. "A penalization method applied to the wave equation". *Comptes Rendus MÃľcanique*, Vol. 333, No. 1, pp. 79 – 85, 2005. High-order methods for the numerical simulation of vortical and turbulent flows.

[Pan 02] T. Pan, R. Glowinski, and G. Galdi. "Direct simulation of the motion of a settling ellipsoid in Newtonian fluid". *Journal of Computational and Applied Mathematics*, Vol. 149, pp. 71–92, 2002.

[Pan 06] T. Pan, C. Chang, and R. Glowinski. "On the motion of a neutrally buoyant ellipsoid in a three-dimensional Poiseuille flow". *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, pp. 2198 – 2209, 2006.

[Pata 72] S. V. Patankar and D. B. Spalding. "A Calculation Procedure for Heat". *Mass and Momentum Transfer in Three-Dimensional Parabolic Flows*, Vol. 15, pp. 1787 – 1806, 1972.

[Pati 09] D. Patil and K. Lakshmisha. "Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh". *Journal of Computational Physics*, Vol. 228, pp. 5262–5279, 2009.

[Pesk 02] C. S. Peskin. "The immersed boundary method". *Acta Numerica*, Vol. 11, pp. 479–517, 2002.

[Pesk 72] C. S. Peskin. "Flow patterns around heart valves: A numerical method". *Journal of Computational Physics*, Vol. 10, No. 2, pp. 252–271, 1972.

[Pian 05] G. Pianet. *Simulations 3D non-stationnaires dédiées à lâĂŹinvestigation de processus de sédimentation à forte dynamique*. PhD thesis, Universit'e Bordeaux I, 2005.

[Pine 88] J. Pineda. "A parallel algorithm for polygon rasterization". *Proceedings of SIGGRAPH '88*, Vol. 22, pp. 17 – 20, 1988.

[Rami 07a] I. Ramière. "Convergence analysis of the Q_1-finite element method for elliptic problems with non-boundary-fitted meshes". *International Journal for Numerical Methods in Engineering*, Vol. 75, pp. 1007–1052, 2007.

[Rami 07b] I. Ramière, P. Angot, and M. Belliard. "A fictitious domain approach with spread interface for elliptic problems with general boundary conditions". *Computer Methods in Applied Mechanics and Engineering*, Vol. 196, pp. 766–781, 2007.

[Rami 07c] I. Ramière, P. Angot, and M. Belliard. "A general fictitious domain method with immersed jumps and multilevel nested structured meshes". *Journal of Computational Physics*, Vol. 225, No. 2, pp. 1347–1387, 2007.

[Rand 05] T. Randrianarivelo, G. Pianet, S. Vincent, and J. Caltagirone. "Numerical modelling of solid particle motion using a new penalty method". *International Journal for Numerical Methods in Fluids*, Vol. 47, pp. 1245–1251, 2005.

[Ravi 82] P. Raviart and J. Thomas. *Introduction à l'analyse numérique des équations aux dérivées partielles*. Masson, 1982.

[Rhie 83] C. M. Rhie and W. L. Chow. "Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation". *AIAA Journal*, Vol. 21(11), pp. 1525 – 1532, 1983.

[Ride 95] W. Rider and D. Kothe. "Stretching and tearing interface tracking methods". *in the 12^{th}, AIAA CFD Conference, San Diego, June 20*, Vol. , p. 1717, 1995.

[RM 08] C. R.M., G.-S. D., and R. A. "A one-parameter family of interpolating kernels for smoothed particle hydrodynamics studies". *J. Comput. Phys.*, Vol. 227, No. , pp. 8523–8540, 2008.

[Russ 71] D. Russell and Z. Wang. "A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow". *Journal of Computational Physics*, Vol. 191, pp. 177–205, 1971.

[Ryhm 85] I. Ryhmng. *Dynamique des Fluides*. Presses Polytechniques Romanes, 1985.

[Saad 86] Y. Saad and M. Schultz. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems". *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, pp. 856–869, 1986.

[Saga 98] P. Sagaut. *Large Eddy Simulation for incompressible flows - An introduction*. Springer Verlag, 1998.

[Sart 08a] A. Sarthou, S. Vincent, P. Angot, and J. Caltagirone. *Finite Volumes for Complex Applications V*, Chap. The sub-mesh-penalty method, pp. 633–640. Wiley, 2008.

[Sart 08b] A. Sarthou, S. Vincent, J. Caltagirone, and P. Angot. "Eulerian-Lagrangian grid coupling and penalty methods for the simulation of multiphase flows interacting with complex objects". *International Journal for Numerical Methods in Fluids*, Vol. 56, No. 8, pp. 1093–1099, 2008.

[Scar 99a] R. Scardovelli and S. Zaleski. "Direct numerical simulation of free-surface and interfacial flow". *Ann. Review Fluid Mech.*, Vol. 31, pp. 567–603, 1999.

[Scar 99b] R. Scardovelli and S. Zaleski. "Direct numerical simulation of free surface and interfacial flows". *Ann. Rev. Fluid Mech.*, Vol. 31, pp. 567–603, 1999.

[Sche 04] O. Schenk and K. Gärtner. "Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO". *Journal of Future Generation Computing Systems*, Vol. 20, pp. 475–487, 2004.

[Schw 06] P. Schwartz, M. Barad, P. Colella, and T. Ligocki. "A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions". *Journal of Computational Physics*, Vol. 211, No. 2, pp. 531 – 550, 2006.

[Shar 05] N. Sharma and N. A. Patankar. "A fast computation technique for the direct numerical simulation of rigid particulate flows". *Journal of Computational Physics*, Vol. 205, No. 2, pp. 439 – 457, 2005.

[Shin 02a] S. Shin and D. Juric. "Modeling Three-Dimensional Multiphase Flow Using a Level Contour Reconstruction Method for Front Tracking without Connectivity". *J. Comput. Phys.*, Vol. 202, pp. 427–470, 2002.

[Shin 02b] S. Shin and D. Juric. "Modelling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity". *Journal of Computational Physics*, Vol. 180, pp. 427–470, 2002.

[Shu 96] C. Shu and G. Jiang. "Efficient implementation of weighted eno schemes". *Journal of Computational Physics*, Vol. 126, pp. 202–228, 1996.

[Shu 98] C. Shu and S. Gottlieb. "Total variation diminishing runge-kutta schemes". *Math. Comput.*, Vol. 67, pp. 73–85, 1998.

[Sing 00] P. Singh, D. D. Joseph, T. I. Hesla, R. Glowinski, and T. Pan. "A distributed Lagrange multiplier/fictitious domain method for viscoelastic particulate flows". *Journal of Non-Newtonian Fluid Mechanics*, Vol. 91, pp. 165 – 188, 2000.

[Suss 94] M. Sussman, P. Smereka, and S. Osher. "A level set approach for computing solutions to incompressible two-phase flow". *Journal of Computational Physics*, Vol. 114, pp. 146–159, 1994.

[Suss 98] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. "An improved level set method for incompressible two-phase flows". *Computers & Fluids*, Vol. 27, No. 5-6, pp. 663 – 680, 1998.

[Tair 07] K. Taira and T. Colonius. "The immersed boundary method: A projection approach". *Journal of Computational Physics*, Vol. 225, pp. 2118–2137, 2007.

[Tan 09] Z. Tan, D. Le, K. LIM, and B. Khoo. "An immersed interface method for the incompressible Navier-Stokes equations with discontinuous viscosity across the interface". *SIAM Journal of Scientific Computing*, Vol. 31, pp. 1798–1819, 2009.

[Tang 04] S. Tanguy. *Développement d'une méthode de suivi d'interface. Applications aux écoulements diphasiques.* PhD thesis, Université de Rouen - CORIA, 2004.

[Tema 01] R. Temam and A. Miranville. *Mathematical modeling in continuum mechanics*. Cambrige University Press, 2001.

[Timm 96] L. Timmermans, P. Minev, and F. V. D. Vosse. "An approximate projection scheme for incompressible flow using spectral elements". *International Journal of Numerical Methods in Fluid*, Vol. 22, pp. 673–688, 1996.

[Treu 06] A. Treuille, A. Lewis, and Z. Popović. "Model reduction for real-time fluids". *ACM Transactions on Graphics*, Vol. 25, No. 3, pp. 826–834, July 2006.

[Tron 08] P. Trontin, S. Vincent, J.-L. Estivalezes, and J.-P. Caltagirone. "Detailed comparisons of front-capturing methods for turbulent two-phase flow simulations". *Int. J. Numer. Meth. Fluids*, Vol. 56, pp. 1543–1549, 2008.

[Tsen 03] Y. Tseng and J. H. Ferziger. "A ghost-cell immersed boundary method for flow in complex geometry". *Journal of Computational Physics*, Vol. 192, No. 2, pp. 593–623, Dec. 2003.

[Tunn 06] N. Tunnicliffe, J. Chesterton, and N. Nancekivell. "The use of the gallaway formula for aquaplaning evaluation in new Zealand". *NZIHT Transit NZ 8 Annual Conference*, Vol. , p. , 2006.

[Unve 92] S. O. Unverdi and G. Tryggvason. "A front-tracking method for viscous, incompressible, multi-fluid flows". *Journal of Computational Physics*, Vol. 100, pp. 25A front–tracking method for viscous, incompressible, multi–fluid flows–37, 1992.

[Verz 01] R. Verzicco, G. Iaccarino, M. Fatica, and P. Orlandi. "Flow in a impeller stirred tank using an immersed boundary method". Annual Research Brief, NASA Center for Turbulence Research, 2001.

[Vinc] S. Vincent, A. Sarthou, J.-P. Caltagirone, F. Sonilhac, P. Février, C. Mignot, and G. Pianet. "Augmented Lagrangian and penalty methods for the simulation of two-phase flows interacting with moving solids. Application to hydroplaning flows interacting with real tire tread patterns". Submitted to Journal of Computational Physics.

[Vinc 00] S. Vincent and J. Caltagirone. "A One-Cell Local Multigrid Method for Solving Unsteady Incompressible Multiphase Flows". *J. Comput. Phys.*, Vol. 163, No. 1, pp. 172–215, Sep. 2000.

[Vinc 04] S. Vincent, J. Caltagirone, P. Lubin, and T. Randrianarivelo. "An adaptative augmented Lagrangian method for three-dimensional multimaterial flows". *Comput. Fluids*, Vol. 33, No. 10, pp. 1273–1289, Dec. 2004.

[Vinc 07] S. Vincent, T. N. Randrianarivelo, G. Pianet, and J.-P. Caltagirone. "Local penalty methods for flows interacting with moving solids at high Reynolds numbers". *Computers & Fluids*, Vol. 36, No. 5, pp. 902–913, 2007.

[Vinc 08] S. Vincent, J. Larocque, D. Lacanette, A. Toutant, P. Lubin, and P. Sagaut. "Numerical simulation of phase separation and a priori two-phase LES filtering". *Comput. Fluids*, Vol. 37, No. 7, pp. 898–906, Aug. 2008.

[Vinc 09] S. Vincent, G. Balmigère, C. Caruyer, E. Maillot, and J. Caltagirone. "Contribution to the modeling of the interaction between a plasma flow and a liquid jet". *Surf. Coating Tech.*, Vol. 203, No. , pp. 2162–2171, 2009.

[Vinc 99] S. Vincent and J. Caltagirone. "Efficient solving method for unsteady incompressible interfacial flow problems". *Int. J. Numer. Meth. Fluids*, Vol. 30, No. 6, pp. 795–811, 1999.

[Vors 92] H. V. D. Vorst. "BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems". *SIAM Journal on Scientific and Statistical Computing*, Vol. 13, No. 2, pp. 631–644, 1992.

[Ye 99] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy. "An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries". *Journal of Computational Physics*, Vol. 156, No. 2, pp. 209 – 240, 1999.

[Yngv 00] G. Yngve, J. O'Brien, and J. Hodgins. "Animating explosions". In: *Proceedings of SIGGRAPH 2000*, pp. 29–36, 2000.

[Youn 82] D. Youngs. *Time-dependent multimaterial flow with large fluid distortion.* K. W. Morton and M. J. Baines (eds), Academic Press, New-York, 1982.

[Yu 07] S. Yu, Y. Zhou, and G. Wei. "Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces". *Journal of Computational Physics*, Vol. 224, No. 2, pp. 729 – 756, 2007.

[Zhou 06a] Y. Zhou and G. Wei. "On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method". *Journal of Computational Physics*, Vol. 219, No. 1, pp. 228–246, 2006.

[Zhou 06b] Y. Zhou, S. Zhao, M. Feig, and G. Wei. "High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources". *Journal of Computational Physics*, Vol. 213, No. 1, pp. 1 – 30, 2006.