Modélisation numérique d'un limiteur de débit en cuve de réacteur à eau pressurisée à l'aide d'une approche de type frontière immergée.

Georis Billo

NNT/NL : 2021AIXM0338/020ED353

THÈSE DE DOCTORAT

Soutenue à l'École Centrale de Marseille
dans le cadre d'une cotutelle avec le CEA
le 20 juillet 2021 par

Georis BILLO

Numerical modeling of an in-vessel flow limiter:
A second order finite element penalized direct forcing immersed boundary method for infinitely thin obstacles in a dilatable flow

Discipline
Sciences pour l'Ingénieur

Spécialité
Mécanique et physique des fluides

École doctorale
ED 353 Sciences pour l'ingénieur

Laboratoire/Partenaires de recherche
Laboratoire M2P2
CEA/DES/IRESNE/DER/SESI/LEMS

Composition du jury

Lisl WEYNANS INRIA Bordeaux Sud-Ouest	Rapporteure
Stéphane VINCENT Université Gustave Eiffel	Rapporteur
Barbara BIGOT CEA	Examinatrice
Cédric GALUSINSKI Université de Toulon	Président du jury
Pierre SAGAUT Aix-Marseille Université	Directeur de thèse
Michel BELLIARD CEA	Encadrant de thèse

Je soussigné, Georis Billo, déclare par la présente que le travail présenté dans ce manuscrit est mon propre travail, réalisé sous la direction scientifique de Pierre Sagaut et l'encadrement de Michel Belliard, dans le respect des principes d'honnêteté, d'intégrité et de responsabilité inhérents à la mission de recherche. Les travaux de recherche et la rédaction de ce manuscrit ont été réalisés dans le respect à la fois de la charte nationale de déontologie des métiers de la recherche et de la charte d'Aix-Marseille Université relative à la lutte contre le plagiat.

Ce travail n'a pas été précédemment soumis en France ou à l'étranger dans une version identique ou similaire à un organisme examinateur.

Fait à Cadarache le 20 juillet 2021

“*All models are wrong, but some are useful.*”

George BOX

Je dédie cette thèse à Michel BILLO et Fabienne GRABOWSKI.

Résumé en français

Dans le cadre du développement de nouveaux systèmes passifs de sûreté pour la seconde et la troisième génération de réacteurs nucléaires à eau pressurisée, les simulations numériques d'écoulements diphasiques turbulents autour de géométries complexes [Bel18; Shi11] sont des outils privilégiés pour modéliser, évaluer et optimiser de nouvelles formes ou de nouveaux designs. Afin de satisfaire les exigences de l'industrie, les outils de mécanique des fluides numérique doivent être le plus rapide, robuste et précis possible. Le but de mon projet de thèse est de développer un outil de cet acabit.

Les contraintes susmentionnées tendent à écarter une approche de type "body-fitted". Nous avons en effet choisi une approche de type domaine fictif pour traiter ce problème [Mar82]. Plus précisément, l'outil développé résout les équations de Navier-Stokes pour un mélange équivalent dilatable à l'aide d'un schéma de projection, d'une méthode de frontière immergée appelée Penalized Direct Forcing – une technique qui hérite à la fois de la pénalisation [ABF99] et du Direct Forcing [Moh97] – adaptée à des obstacles infiniment fins et d'une formulation éléments finis. Différentes conditions limites immergées (adhérence, glissement, Neumann) peuvent être modélisées en imposant des valeurs Dirichlet pour le champ de vitesse au voisinage de la paroi immergée. Pour imposer ces valeurs Dirichlet, deux variantes ont été étudiées : la première consiste à directement utiliser la vitesse de l'obstacle comme valeur Dirichlet et la seconde à interpoler linéairement le champ de vitesse en proche paroi (cette dernière variante est motivée par une augmentation de l'ordre de convergence en espace) [BBS20]. Plusieurs méthodes d'interpolation (directionnelle, multidirectionnelle, hybride) ont été développées mais, dans tous les cas, ces méthodes nécessitaient de nouvelles données concernant la géométrie de l'obstacle. Ainsi, le traitement des données géométriques, provenant généralement de maillages créés par des outils de conception assistée par ordinateur, est une question centrale et, encore une fois, différentes approches ont été testées.

Deux autres problèmes clés, en simulation numérique, sont la vérification et la validation. Dans un premier temps, des cas tests académiques (tels que l'écoulement de Poiseuille, de Taylor-Couette, autour d'un cylindre) ont été utilisés pour mener des études de convergence. Les résultats obtenus sont en accord avec les solutions analytiques et les données expérimentales. De plus, conformément à la théorie, l'ordre 2 est atteint en espace lorsque l'interpolation linéaire est utilisée. Ensuite, des cas quasi-industriels (dont un en 3D) ont été utilisés, d'une part, pour démontrer la capacité de notre méthode à traiter des géométries complexes et, d'autre part, pour dresser ses avantages et ses inconvénients.

En termes de perspectives à venir, dans le but de réaliser des simulations d'écoulements diphasiques et turbulents, des modifications de la méthode sont envisagées, telles que l'adaptation du schéma de projection à un modèle faiblement compressible et l'ajout de loi de paroi immergée [WJS18] (*i.e.* interpolation via des lois de paroi turbulentes). En octobre 2020, un autre projet de thèse a démarré sur ces thématiques.

Mots-clés : équations de Navier-Stokes, Mécanique des Fluides Numérique, Éléments Finis, Domaine Fictif, Frontière Immergée, Direct Forcing, Pénalisation, schéma de projection, reconstruction de données, interpolation directionnelle et multidirectionnelle

Abstract

In the framework of the development of new passive safety systems for the second and third generations of nuclear reactors, the numerical simulations, involving complex turbulent two-phase flows around thin or massive inflow obstacles [Bel18; Shi11], are privileged tools to model, optimize and assess new design shapes. In order to match industrial demands, computational fluid dynamics tools must be the fastest, most accurate and most robust possible. The purpose of my PhD was to design and develop such a tool.

The aforementioned constraints tend to rule out a "body-fitted". Indeed, we chose a Fictitious Domain approach to deal with this problem [Mar82]. More precisely, the developed tool involves solving the Navier-Stokes equations using a projection scheme for a dilatable mixture fluid coupled with an Immersed Boundary (IB) approach: the penalized direct forcing method [BF10; IBF14] – a technique whose characteristics inherit from both penalty [ABF99] and immersed boundary methods [Pes02] – adapted to infinitely thin obstacles and to a Finite Element (FE) formulation. Various IB conditions (slip, no-slip or Neumann) for the velocity on the IB can be managed by imposing Dirichlet values in the vicinity of the thin obstacles. To deal with these imposed Dirichlet velocities, we investigated two variants: one in which we directly use the obstacle velocity and another one in which we use linear interpolation (this last variant being motivated by an increase of the space order of convergence) [BBS20]. Several approaches were investigated (directional, multi-directional and hybrid) for the linear interpolation of the velocity near the obstacle but, in any case, geometrical data coming from the obstacle are needed. Thus, retrieving geometrical data, generally from a Computer Assisted Design (CAD) object, is a key issue and, once again, several methods were studied and compared.

Another major issue, when dealing with numerical simulations, is validation. First, studies involving various one-phase academic test cases (such as Poiseuille, Taylor-Couette and the flow around a circular cylinder) were carried out. The results obtained were in good agreement with analytical and experimental data. As expected, second order accuracy (in space) was numerically assessed when using linear interpolation. Then, studies involving industrial or quasi-industrial test cases were carried out to illustrate the advantages and drawbacks of this approach.

In a shortcoming second step, to face two-phase turbulent fluid simulations, some methodology modifications will be considered such as adapting the projection scheme to low-compressible fluid and immersed wall-law boundary conditions [WJS18] (another PhD project has begun in October 2020).

Keywords: Navier-Stokes equations, Computational Fluid Dynamics, Finite Element Method, Fictitious Domain Method, Immersed Boundary Method, Direct Forcing Method, Penalty, Projection scheme, Data reconstruction, Directional and multi-directional interpolation

Remerciements

Dans un premier temps, j'aimerais remercier Michel BELLIARD pour ces quatre années de travaux passionnants. À aucun moment je n'ai senti de pression hiérarchique entre nous, au contraire, j'ai toujours eu l'impression que nous travaillions ensemble, dans une coopération enrichissante avec des échanges mutuels. J'admire ta personne car, à mon sens, tu incarnes à merveille les qualités que chaque scientifique devrait cultiver : calme, écoute, humilité, remise en question et rigueur. Cette thèse sous ton encadrement a été un honneur et un immense plaisir. Je te remercie également pour ton soutien pendant la période difficile que j'ai eu à surmonter – mon expérience personnelle m'a montré que tous les encadrants ne réagissaient pas ainsi, il me paraît donc important de le souligner.

Ensuite, je remercie Pierre SAGAUT pour sa supervision. Ton impressionnante connaissance de la physique, des méthodes numériques mais aussi des enjeux industriels te permet d'avoir une vision d'ensemble éclairée et de prodiguer des conseils très pertinentes. J'ai également été impressionné par ta sagacité : tu as facilement et rapidement compris mes travaux tout en identifiant les difficultés quasi-immédiatement. Merci pour cette opportunité de recherche.

Je remercie également chacun des membres du jury :

- Merci Lisl WEYNANS et Stéphane VINCENT pour le temps qu'ils ont accordé à la lecture critique de mon mémoire de thèse. Grâce à vos remarques bienvenues, je peux être fier de ce travail. Merci également à Lisl d'avoir fait le déplacement malgré les aléas des transports.

- Merci Cédric GALUSINSKI d'avoir présidé mon jury. Je suis vraiment heureux et fier que tu aies pu constater mon évolution depuis SEATECH. C'est en grande partie grâce à la qualité de la formation dispensée par l'IMATH que j'ai pu réaliser ces travaux – à ce propos je remercie chaleureusement Frédéric Golay et Mehmet Ersoy.

- Merci à Barbara BIGOT pour sa réactivité, son implication et ses questions pertinentes. Merci d'avoir fait le déplacement jusqu'à Marseille.

Au CEA, dans le cadre de mes travaux, j'aimerais notamment remercier Adrien BRUNETON, Pierre LEDAC et Anida KHIZAR. Vous vous êtes toujours montrés disponibles malgré votre charge de travail. Vous m'avez sorti du pétrin un nombre incalculable de fois ; sans vous ce projet n'aurait jamais été aussi avancé et pérennisé. Je remercie également Christophe BOURCIER pour son aide concernant la génération de maillage

body-fitted avec SALOME. Merci à Isabelle RAMIÈRE pour son expertise scientifique et son aide de manière plus générale.

Par ailleurs, je remercie sincèrement toute l'équipe du SESI pour son accueil chaleureux. Grâce à vous, ces quatre années ont été très agréables. Dans ces remerciements, je n'oublie pas la nouvelle génération de doctorants pour leur soutien psychologique, en particulier Marlène JEANNIN qui a eu le courage et la gentillesse de relire ce mémoire – j'en profite également pour adresser un message aux doctorants : ça va bien se passer, force à vous camarades. Je remercie également mon « jumeau » de thèse, Jean-Marc LABIT pour les conversations passionnantes qu'on a partagées, autant sur le plan scientifique que philosophique. J'espère sincèrement que tout se goupillera bien pour toi.

Je profite également de cette section pour remercier personnellement Manuel SAEZ pour son professionnalisme mais surtout pour soutien en dépit des difficultés rencontrées. Merci également pour ton aide concernant mon insertion professionnelle. J'ai apprécié travailler avec toi.

Je souhaite remercier Edwige RICHEBOIS et Gérard MIGNOT pour leur honnêteté, leur disponibilité ainsi que leur sollicitude à mon égard. Merci également à Jean-Michel Ruggieri pour son accessibilité et son soutien.

Dans un registre moins solennel, j'aimerais remercier Nathalie CHAUVIN et Jean-Pierre CHAUVIN pour leur aide. C'est en partie grâce à vous que j'en suis là aujourd'hui – désolé pour les cafouillages dans la diffusion de la date de ma pré-soutenance à Cadarache, je me rattraperai.

Un grand merci à toute ma famille, à commencer par ma mère. Je sais que tu t'es beaucoup inquiétée pendant cette thèse, en grande partie malgré moi, mais finalement tout s'est bien passé, grâce à toi. Toute ma vie, et même récemment, tu t'es littéralement sacrifiée pour m'offrir le meilleur. J'espère que maintenant tu vas pouvoir profiter de ta retraite et enfin vivre un peu pour toi-même. En tout cas, tu peux être fière : ce doctorat c'est un peu le tien car sans toi et ton abnégation, je ne serais jamais parvenu jusque-là. Merci également à ma tante, pour l'émotion, le soutien, l'amour, ça m'a énormément touché !

J'aimerais remercier mes deux grandes sœurs, pour leur amour inconditionnel et leur présence lors de ma soutenance. Ça me touche profondément, je ne sais pas si je pourrai un jour vous rendre la pareille. Sachez que je dédie ce diplôme à notre père. Mon oncle, si tu lis ça, j'espère que tu es fier de moi, en tout cas je pense fort à toi.

Merci aussi à tous mes neveux et nièces, en particulier Maëlle pour avoir fait le déplacement – j'espère que tu ne t'es pas trop ennuyée. Je vous souhaite de la réussite dans tous vos projets. Je suis fier de vous, de votre détermination malgré les circonstances actuelles. Courage !

Merci aux cousines et aux cousins, notamment Ève, Delphine et Robin qui se sont perdus dans les dédales de l'université de Saint-Jérôme pour venir me voir. Votre soutien m'a vraiment touché, ça fait très plaisir, à bientôt pour de grandes fêtes ! Tony et la petite famille, j'espère que vous êtes fiers aussi ; merci d'être toujours là quand on a besoin.

Enfin, merci à mes amies et amis, sans qui je ne serais tout simplement pas là aujourd'hui ; vous savez pourquoi.

Kévin et Romain, vous êtes vraiment géniaux – vous étiez là pendant la période difficile que j'ai traversé, je n'ai pas oublié.

Simon et Manon, est-ce que j'aurais vraiment survécu sans parties de jeu de rôle, sans grandes discussions politiques, sans game jam ? En l'absence de preuve, je me permets de douter raisonnablement.

J'ai la chance d'avoir la bande de pote la plus soudée, solidaire et chaleureuse du monde. Merci aux jouquards et à sieur Bohic pour les franches rigolades et autres poilades. Merci Jimmy, pour nos conversations toujours passionnantes et sur des sujets qui nous tiennent à cœur – courage pour la suite. Merci Joël, j'espère qu'on pourra fêter ça autour d'une forge ou d'un tour. Merci Justine d'avoir fait le déplacement, ça m'a fait trop plaisir ! Toujours entière, pleine de réconfort et d'admiration, ça fait vraiment du bien au moral mais n'oublie pas que tu es une personne géniale qu'on ne mérite pas ! Chère Alaïs, merci d'avoir affronté ton dégoût épidermique pour les maths en venant à ma soutenance, ça me fait vraiment très plaisir, j'espère que ça en valait la peine ! Merci aussi pour toutes les créations et œuvres partagées ! Alice, ton amitié et ton soutien comptent vraiment pour moi, j'espère que je parviens à te rendre la pareille. Merci pour tout – arrête de te sous-estimer et dépêche-toi de te mettre à Dark Souls. Olivier, on dit souvent que tu parles peu ou que tu es un peu lymphatique, mais c'est toi qui en fais le plus. Tu es un gars en or, ne change rien. Mathieu, le bon Mathieu alias l'homme qui a bravé tous les dangers du monde moderne pour venir à ma soutenance ! Merci pour ta profonde dévotion aux autres et ton abnégation (pense un peu à toi, tu veux), on en fait peu des amis comme toi. Léo, tu n'étais pas là, mais on va fêter ça dignement, peut-être à base d'instruments ? En tout cas, merci pour ton amitié durant toutes ces années et en particulier les découvertes musicales qui en ont découlé. Alban, le bro, toujours là quand il faut, en dépit des années, de la distance ; tu t'es toujours soucié de moi et m'a toujours offert ta gentillesse ainsi que ton écoute. Éric, frérot, que dire ? Est-ce que ce ne serait pas le pinacle de notre amitié ? Du CP à la thèse ? Je dis ça, mais je suis sûr que non. Tout ne fait que nous rapprocher et je pense que ça va durer encore longtemps. Merci d'exister, tout simplement. Je dédie cette thèse à nos défunts parents, j'aurais aimé qu'ils soient là pour nous voir.

Le meilleur pour la fin, ma chérie. Depuis que tu es dans ma vie j'ai la force d'entreprendre, d'avancer dans des projets sans abandonner. Grâce à toi, je fais preuve de plus d'ouverture. Grâce à toi, je suis quelqu'un de meilleur. Grâce à toi, j'ai trouvé un but dans ma vie et cette thèse va peut-être me permettre d'atteindre : m'installer et vieillir à tes côtés. Je ferai tout pour nous offrir la meilleure vie possible. J'espère que cela sera à la hauteur de tout ce que tu as fait pour moi, en particulier pendant cette période très difficile de ma vie. Je t'aime de tout mon cœur.

Glossary

Acronym	Definition
AMR	Adaptative Mesh Refinement
ASN	French Nuclear Safety Authority
BC	Boundary Conditions
BTD	Balancing Tensor Diffusivity
CAD	Computer Aided Design
CEA	French Alternative Energies and Atomic Energy Commission
CFD	Computational Fluid Dynamics
C-FEM	Cut Finite Element Method
CGEM	Cartesian Grid Embedded Method
EOS	Equation Of State
FB(M)	Fictitious Boundary (Method)
FCM	Finite Cell Method
FD(M)	Fictitious Domain (Method)
FE(M)	Finite Element (Method)
GFE(M)	Generalized Finite Element (Method)
GCIBM	Ghost-Cell Immersed Boundary Method
HEM	Homogeneous Equilibrium Model
IB(M)	Immersed Boundary (Method)
IBVP	Initial-Boundary Value Problem
IIM	Immersed Interface Method
ISI	Immersed Spread Interface method
IC	Initial Conditions
JEBC	Jump Embedded Boundary Condition
LES	Large Eddy Simulation
LM	Lagrange Multipliers
PDE	Partial Derivative Equation
PDF	Penalized Direct Forcing
PWR	Pressurized Water Reactor
LOCA	Loss Of Coolant Accident
RANS	Reynolds Averaged Navier-Stokes
X-FE(M)	eXtended Finite Element (Method)

Notations

The following notations apply to the whole document. Some notations are not described as they are only used once, in specific sections. Double bars denote tensors. Vectors are usually denoted with bold small letters while matrices are usually denoted with bold capital letters.

Sets and spaces

Symbol	Definition
$[\![n,m]\!] \subset \mathbb{N}$	The set of integers superior or equal to $n \in \mathbb{N}$ and inferior or equal to $m \in \mathbb{N}$
$K \times E$	Cartesian product of two sets
$K \setminus E$	Set-theoretic difference of K and E
K^n	Cartesian power of a set with $n \in \mathbb{N}$
$\mathbb{R}^n$	The space of real vectors with $n \in \mathbb{N}^*$ components
$\mathbb{R}^{m \times n}$	The space of real matrices with $m \in \mathbb{N}^*$ rows and $n \in \mathbb{N}^*$ columns
$d \in [\![1,3]\!]$	The space dimension
$\Omega \subseteq \mathbb{R}^d$	A generic open compact set (fictitious domain)
$\partial\Omega$	The piecewise regular boundary of Ω
$\Omega_f \subset \Omega$	The fluid or actual computational domain
$\Omega_s = \Omega \setminus \Omega_f$	The solid or non-fluid part of the fictitious domain
Γ_i	The surface associated to the i-th distinct interface between Ω_f and Ω_s
$\Gamma = \bigcup_{i=1}^{n} \Gamma_i$	The union (collection) of surfaces constituting the frontier between Ω_f and Ω_s or the boundary of the actual computational domain with n the number of distinct interfaces between Ω_f and Ω_s

Vector and function spaces

Symbol	Definition
$\mathbf{0}_V$	The neutral element of a vector space V
V'	The dual space of a vector space V
$\mathscr{L}^2(\Omega)$	The set of square-integrable functions on Ω
$\mathscr{L}^1_{\text{loc}}(\Omega)$	The set of locally integrable functions on Ω
$\mathscr{L}^\infty(\Omega)$	The set bounded functions on Ω
$\mathscr{H}^1(\Omega)$	The Sobolev space of order 1 on Ω
$\mathscr{V}^h(\Omega)$	A discrete approximation of the vector space $\mathscr{V}(\Omega)$
$\mathscr{D}(\Omega)$	The set of smooth compactly supported functions (bump or test functions) on Ω
$\mathbb{Q}_k$	The set of polynomial of partial degree less or equal to k

Vector operations and norms

Symbol	Definition
$.$	An operand
$. \otimes . : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n\times m}$	Tensor product
$. \cdot . : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$	Scalar product
$.^2 : \mathbb{R}^n \to \mathbb{R}$	The scalar product of a vector with itself
$. \circ . : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$	The Hadamard product
$\lvert . \rvert : \mathbb{R}^n \to \mathbb{R}$	The Euclidian norm of a vector (absolute value for $n = 1$)
$\lvert . \rvert_2 : \mathscr{L}^2(\Omega) \to \mathbb{R}$	The $\mathscr{L}^2$ norm
$\lvert . \rvert_\infty : \mathscr{L}^\infty(\Omega) \to \mathbb{R}$	The $\mathscr{L}^\infty$
$\lvert . \rvert_2^h : \mathbb{R}^n \to \mathbb{R}$	Approximate or discrete $\mathscr{L}^2$ norm
$\lvert . \rvert_\infty^h : \mathbb{R}^n \to \mathbb{R}$	Approximate or discrete $\mathscr{L}^\infty$
$\lVert . \rVert_1 : \mathbb{R}^{m\times n} \to \mathbb{R}$	1-norm for matrices, which is equivalent to the maximum absolute column sum
$\lVert . \rVert_\infty : \mathbb{R}^{m\times n} \to \mathbb{R}$	∞-norm for matrices, which is equivalent to the maximum absolute row sum

Differential operators

Symbol	Definition
$\partial_x y$	Partial derivative of function y with respect to the variable x
$\nabla \cdot y$	Divergence of vector or tensor y
∇y	Gradient of scalar or vector y
Δy	Laplacian of scalar or vector y

Functions and variables

Symbol	Definition
$\chi : \Omega \longrightarrow \{0,1\}$	The characteristic function associated to Γ
$\rho : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}$	Mixture fluid density (kg.m^{-3})
$\rho_G : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}$	Density of the gas phase(kg.m^{-3})
$\rho_L : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}$	Density of the liquid phase(kg.m^{-3})
$\mathbf{u} : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}^d$	Mixture velocity (m.s^{-1})
$\mathbf{u}^* : \Omega \longrightarrow \mathbb{R}^d$	Predicted mixture velocity (m.s^{-1})
$\mathbf{u}_G : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}^d$	Velocity of the gas phase (m.s^{-1})
$\mathbf{u}_L : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}^d$	Velocity of the liquid phase (m.s^{-1})
$H : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}$	Mixture enthalpy (J.kg^{-1})
$p : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}$	Mixture pressure (Pa)
$\bar{\bar{\sigma}} : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}^{d \times d}$	Mixture viscous stress tensor (Pa)
$\bar{\bar{\varepsilon}} : \mathbb{R}^+ \times \Omega \longrightarrow \mathbb{R}^{d \times d}$	Mixture strain rate tensor

Notations and parameters related to numerical methods

Symbol	Definition
x^n	Notation for a time-discrete quantity x (which can be $\rho, \mathbf{u}, p$ or H) at time step number $n \in \mathbb{N}$
δt	Time step
η	Penalty parameter

Physical properties, indicators and parameters

Symbol	Definition
ν	Cinematic viscosity of the fluid ($m^2.s^{-1}$)
μ	Dynamic viscosity of the fluid (Pa.s)
H^L_{SAT}	Enthalpy at saturation of the liquid phase ($J.kg^{-1}$)
$\mathfrak{L}$	Latent heat at saturation ($J.kg^{-1}$)
$\mathfrak{v}$	Void fraction
$\mathfrak{x}$	Thermodynamic title
C_d	Drag coefficient
C_l	Lift coefficient
L_w	Recirculation length (m)
Re	Reynolds number
St	Strouhal number
Ta	Taylor number

Contents

List of Figures

List of Tables

1. Introduction and generalities

Summary

1.1. A bit of semantics

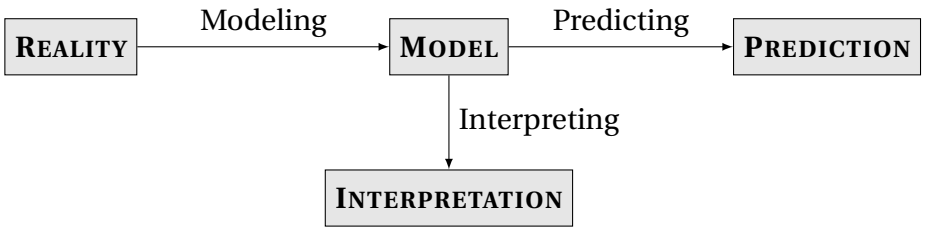


Figure 1.1. Diagram representing the general process of modeling.

The idea of this section is to define the terms present in the title of this PhD thesis, or at least provide a framework. Modeling refers to the action of building a model (which

can be of very various kinds) based on reality (provided we agree about the existence of reality). The interesting property of a model is that it is easier to handle than reality itself. Indeed, models are usually used to interpret (deduce properties or information, isolate patterns or mechanisms, etc.) and even, for the most sophisticated ones, to predict some aspects of reality (*cf.* Figure 1.1). As an example of model, we can consider the map of Marseille's subway line 1 (*cf.* Figure 1.2). Using this map, we can deduce properties of the subway network – for instance: station Baille is next to station La Timone – and even make predictions – for instance: if a person take the line 1 at station Malpassé, direction La Fourragère, and comes out of the train at the third stop, this person will stand at the station Cinq Avenues. Of course, there is no unique model for the entire reality. Every model is a partial representation of the world (*i.e.* the subway map does not explain how the locomotive moves from one station to another) and is limited by its scope of validity (for instance, Figure 1.2 does not provide the geographical positions of the stations).

Figure 1.2. Marseille's subway line 1.

Now that we went through the general concepts, let us discuss numerical modeling, which is, in fact, just a particular case of the above-mentioned process. Regarding numerical modeling, we are concerned by a specific part of reality for which mathematical models exist (what I call physical systems). Those mathematical models can be of various types but our cases of interested are Initial-Boundary Value Problems (IBVP) involving a set of Partial Derivative Equations (PDE). Roughly, studying a system and building mathematical models are aims of physics. However, some PDE coming from mathematical models cannot be solved "manually" (*i.e.* there can be proof of existence and uniqueness of a solution but we are not able to write it). Here intervene applied mathematics and more specifically numerical simulation (or numerical solving). Numerical simulation encompass legions of numerical methods to construct approximate solutions using computer science. Finally, numerical modeling refers to the whole process of mathematical modeling and numerical solving (*cf.* Figures 1.3).

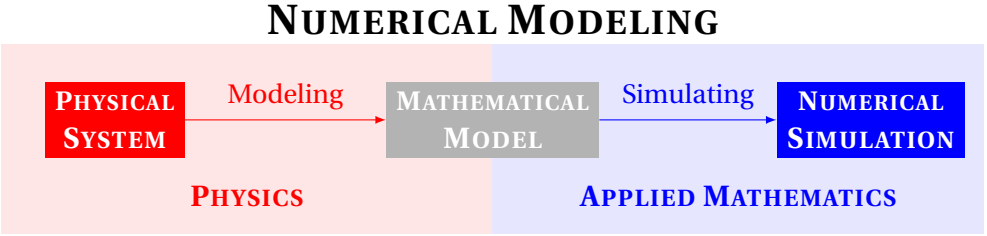


Figure 1.3. Diagram representing the process of numerical modeling.

Thus, the title of this PhD now becomes clearer:

- "Numerical modeling" refers to the numerical solving of an Initial-Boundary Value problem representative of a physical system,
- "In-vessel flow limiter" refers to the physical system we aim to study,
- "Second order finite element penalized direction forcing immersed boundary method" refers to the set of numerical methods used to construct approximate solutions.

This gives us some kind of schedule. In this Chapter, we first discuss generality about the physical systems in question and their mathematical modeling in Sections 1.2 and 1.3. Then, we give a brief history of the class of numerical methods that is central in this PhD thesis: the Fictitious Domain (FD) methods.

Nota: Of course, those themes (modeling and simulating) are extensively studied in philosophy of science and this little introduction is unpretentious, non-exhaustive and does not depict all the subtleties of the subject. To deal with this issue in depth, I recommend the collective French book "Modéliser et Simuler : Epistémologies et pratiques de la modélisation et de la simulation." [VSD21] and the work of Franck VARENNE, a French expert of the field.

1.2. Context

In practice, designing a nuclear reactor implies carrying out numerous preliminary safety studies in order to match the criteria provided by independent nuclear safety authorities – in France: Nuclear Safety Authority (ASN). Some of those studies consist in examining the behavior of the reactor under specific hypothetical incidental or accidental configurations – not only for safety purpose, but also for system design or redesign. There are various scenarios and one of them is called Large-Break Loss Of Coolant Accident (LB-LOCA). It considers a large break happening in the cold-water inlet section (*cf.* Figure 1.7). As the pressure of the primary circuit is about 150 bar, a brutal depressurization happens, coupled with a recirculation within the core and a fast (within seconds) core dewatering. Without proper countermeasures, this

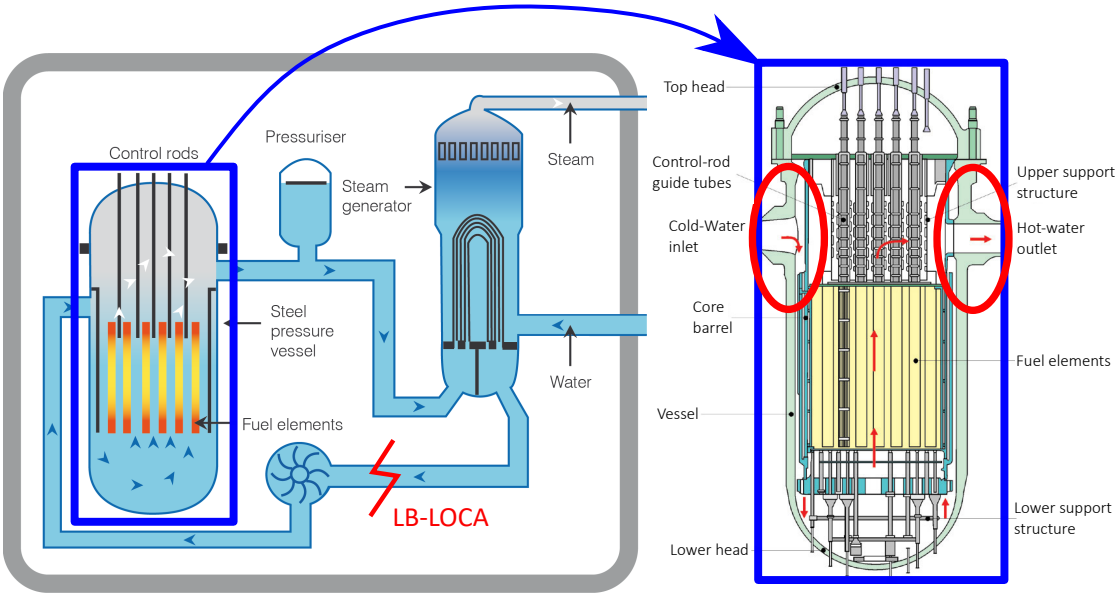


Figure 1.4. Schematic representation of a Pressurized Water Reactor (PWR) and detailed view of its core.

situation can lead to the melting of the core (severe accident). For more details about the phenomenology of the LB-LOCA, the interested reader can refer to [SB17; Bel18].

Several ways are considered by the CEA and its partners to prevent or lower the risk of severe accident (occurring in the case of a LB-LOCA for instance). One of them, driven by the increasing performance and safety requirements for the third generation of nuclear reactors , consists in adding a passive safety systems (*i.e.* a system that activates itself without the need of mechanical or electrical actuation) within the reactor. Some of them are based on the concept of hydraulic diodes, *i.e.* devices allowing the flow to cross them in only one direction, by analogy with electric diodes. In terms of fluid dynamics, that means finding a shape inducing a pressure drop that is minimal in nominal conditions and maximal in reverse flow. Engineers came up with many different designs of hydraulic diodes (*cf.* Figure 1.5 for some examples). In the nuclear industry, at least two passive safety systems are based on hydraulic diodes:

- **"Advanced" accumulator:** In the case of a LB-LOCA, due to the brutal depressurization, a huge flowrate is needed to refill the circuit (to prevent the dewatering of the core). This flowrate comes from a safety water accumulator that rapidly discharges its content in the primary circuit. However, once the circuit is refilled, the flowrate needed to compensate the leak is much lower. Contrary to the standard accumulator, the design of the "advanced" accumulator allows a double-regime (a huge flowrate to refill the primary circuit at first, and then a low sustained flowrate to compensate the leak of the break) thanks to a vortex diode (*cf.* Figure 1.6 for a schematic drawing that illustrates the double-regime). There are different designs of "advanced accumulator", notably Japanese [Shi11]

and Korean [Chu+08].

- **Flow limiter**: This system is located between the core barrel and the vessel at the height of the cold-water inlets and hot-water outlets (*cf.* red ellipses in Figure 1.4) and is composed of thin fins (*cf.* Figure 1.7a). During normal operation, those fins do not significantly disturb the flow heading into the vessel via the cold-water inlet. In accidental behavior (LB-LOCA leading to a recirculation within the core), force the creation of a large vortex (*cf.* Figure 1.7b), which dissipates kinetic energy due to turbulence (*i.e.* reduces the flowrate going outflow of the vessel and so postpones the core dewatering). Once again, for more details about this device, the interested reader can refer to [SB17; Bel18].

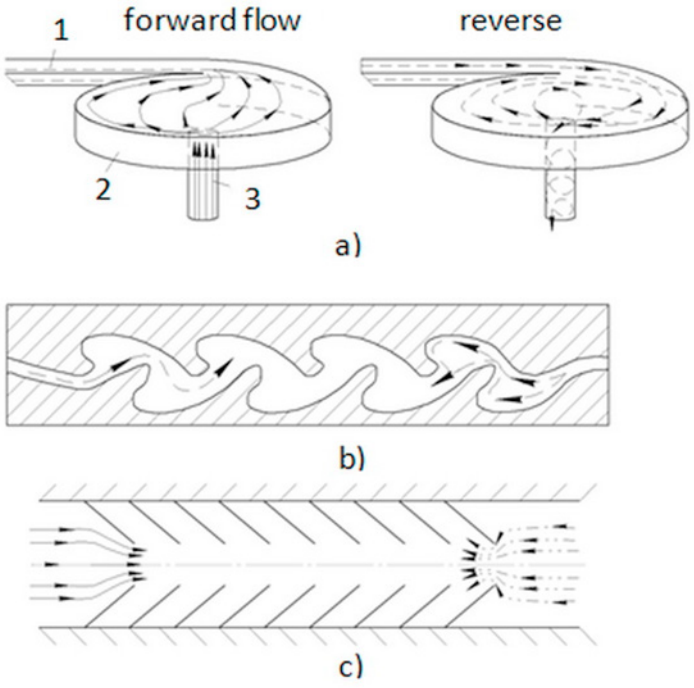


Figure 1.5. Several designs of hydraulic diodes: a) vortex diode, b) Tesla diode, c) diaphragm diode (source: [KPS17]).

Thus, developing a numerical tool capable of simulating the flow between thin obstacles of arbitrary shapes with accuracy and rapidness could be very useful to assess optimal design of hydraulic diodes (via topological optimization for instance, as in [Shi+18]). This document focuses on simulations of the flow limiter, but efforts to simulate the flow in an advanced accumulator using the same methodology are currently carried out.

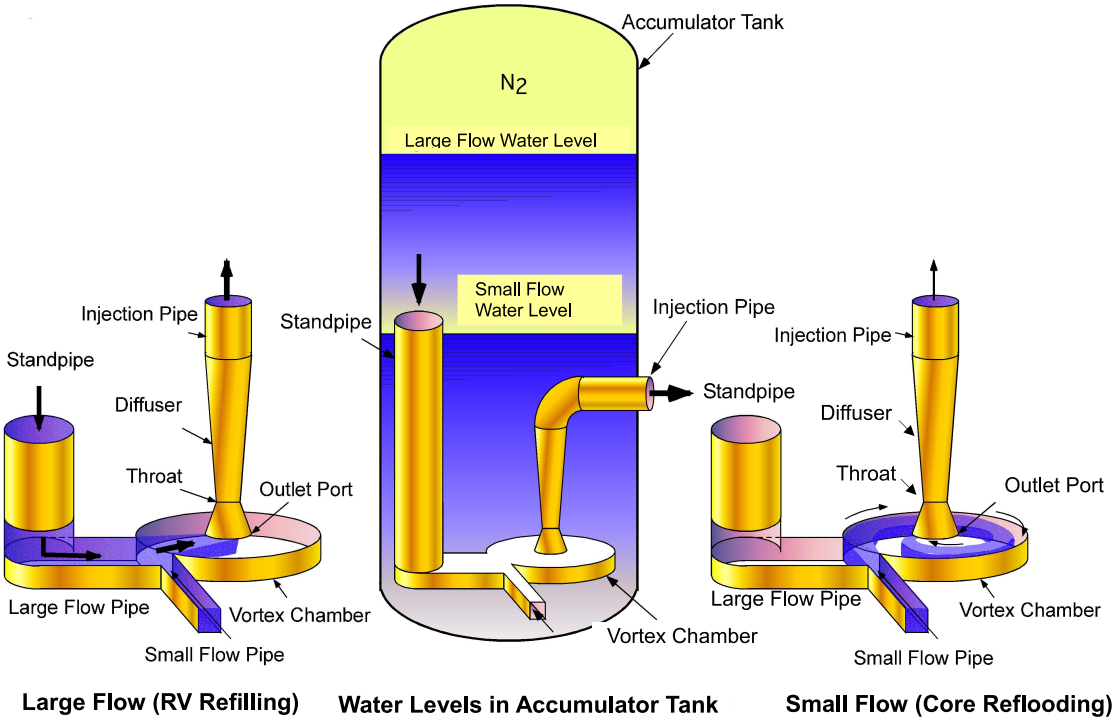


Figure 1.6. Schematic drawing of the Japanese design of the "advanced" accumulator (source: [Shi11]).

1.3. Motivation and objective

In physics, the modeling of problems, such as the one presented in the previous section, is usually based on Initial-Boundary Value Problems (IBVP) involving Partial Derivative Equations (PDE) ensuring the conservation of given quantities (momentum, internal energy and mass in the case of thermal hydraulics) as well as closure models for physical properties or parameters. In engineering, the idea is to solve the IBVP for initial or boundary values representative of a system (or device) in order assess its design or even redesign it.

However, solving the IBVP representative of an industrial case can be difficult in practice as, sometimes, solutions, if they exist, are unknown and/or impossible to compute manually. A common way to approximate those solutions is numerical solving. More precisely, in fields such as thermodynamics, mechanics or electromagnetism, grid-based resolutions are widely used to produce approximations and simulations.

In the field of fluid dynamics and thermal hydraulics, many engineering works concern numerical flow simulations around (or through) obstacles. A straightforward way to take into account inflow obstacles is making grids that conforms to their shapes, what we call the "body-fitted" approach (*cf.* Figure 1.8a). Nevertheless, when we consider moving or deformable obstacles (fluid-structure interaction, flows induced by a stirrer, *etc.*) the fluid computation mesh must be rebuilt at each modification (in

(a) Behavior during nominal operation.

(b) Behavior during accidental situation.

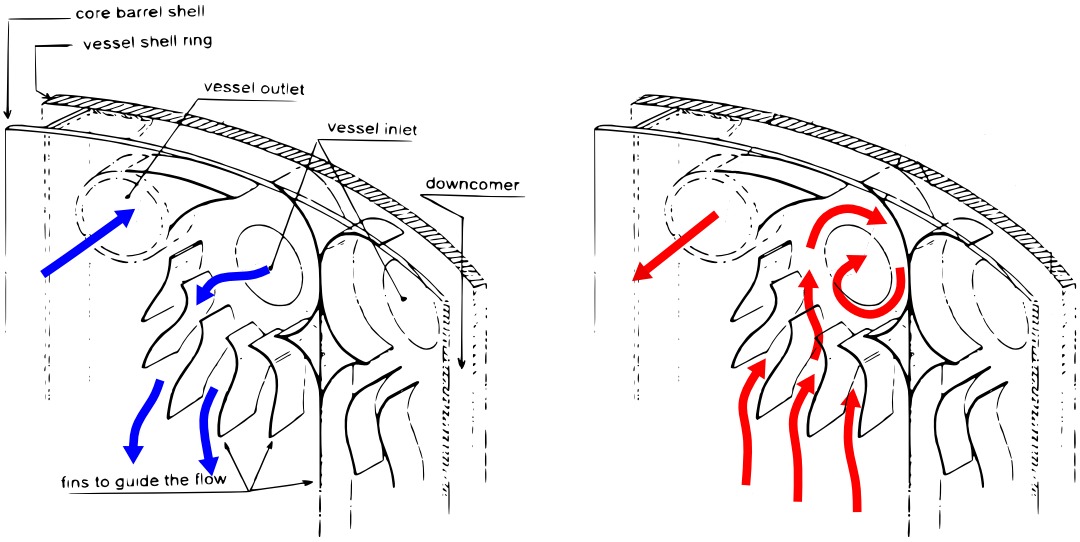


Figure 1.7. Schematic drawing of the flow limiter system patented by the CEA.

shape or position) of the body. Mesh rebuilding is also required when considering the topic of topological shape optimization of rigid fixed obstacles (which is our topic of interest as mentioned in Section 1.2) because many shapes need to be assessed in order to build a statistical answer surface and determine the optimal geometry. Any of the above-mentioned cases can lead to costly (in both time and computing resources) simulations. To overtake this issue, methods allowing the decoupling of the computation grid from the physical domain can be very useful, notably the fictitious domain approach (*cf.* Figure 1.8b), originally introduced by V.K. SAUL'EV [Sau63]. It allows to consider simple meshes (often Cartesian grids) and fast methods (such as i, j, k Finite Difference). Since the 60's, the Fictitious Domain have been declined in many variants and has been investigated in various domains of industrial interest, such as tire design [Vin+11], fluid-transported solid particles [Glo+99] or nuclear-waste vitrification [BI15].

Hence, given the framework presented in Section 1.2, we chose to investigate a Fictitious Domain approach to develop a numerical tool whose ultimate goal is to carry out shape optimization studies of thin inflow obstacles. The following section gives an overview of the major families of Fictitious Domain methods, as well as their pros and cons regarding the aimed application. We conclude with the chosen method and explain our choice.

1.4. Fictitious Domain Method

As mentioned in the previous section, the numerical solving of PDE generally involves building a spatial discretization of the computational domain. The common way to

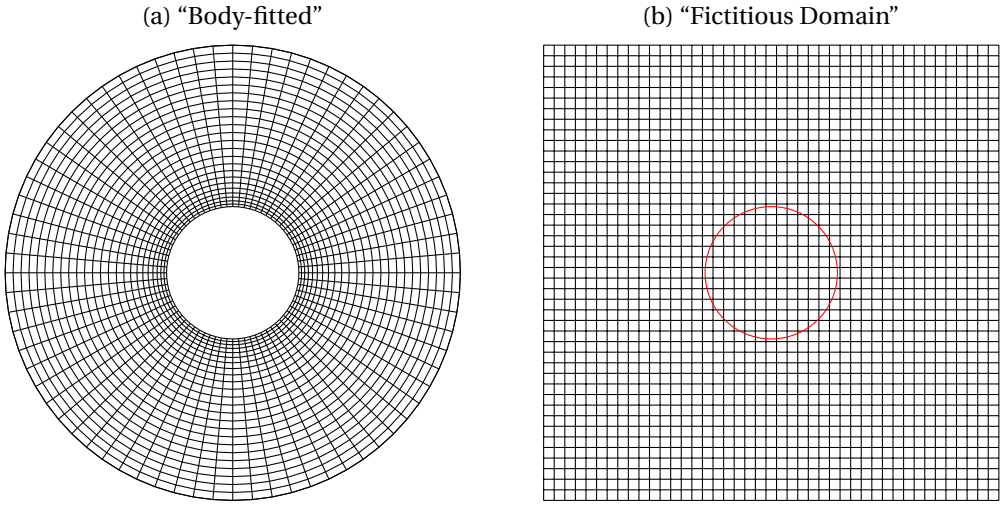


Figure 1.8. Schematic representation of an Eulerian grid used to compute the flow around a circular cylinder in both “body-fitted” and “immersed boundary” cases.

achieve this discretization consists in creating a discrete domain, or mesh, whose boundaries fit those of the physical domain. However, we also saw that, when the geometry involved is rather complex (which is the case in many industrial applications, notably when considering the flow limiter presented in section 1.2), this approach can be very expensive in terms of computational resources – especially for problems involving moving (or deformable) boundaries or topological shape optimization because many different meshes (or grids) need to be generated. An alternative to “body-fitted” grids resides in the Fictitious Domain (FD) approach which is a class of numerical methods belonging to the domain decomposition methods.

The main idea, which is pretty well summarized by K. KHADRA *et al.* in the context of the Navier-Stokes equations [Kha+00], is to solve the PDE constitutive of a given problem on a much simpler computation domain Ω (for instance a parallelepipedal box in 3D), called “fictitious domain”, in which the physical domain Ω_f is embedded (*cf.* Figure 1.9a). Here, Ω_f and Ω_s are two subsets embedded in the FD Ω such as $\overline{\Omega} = \overline{\Omega}_f \cup \overline{\Omega}_s$ and $\dim(\overline{\Omega}_f \cap \overline{\Omega}_s) = \dim(\Omega) - 1$. The interface between $\overline{\Omega}_f$ and $\overline{\Omega}_s$ – which can be a union of interfaces – is denoted $\Gamma = \partial\Omega_f \cap \partial\Omega_s \setminus \partial\Omega$. This approach allows using simple mesh types and fast methods such as, for instance, Cartesian grid with an ijk finite difference scheme. Since this approach was introduced, in 1963 by V.K. SAUL'EV [Sau63], many techniques have been proposed in different fields of physics – I. RAMIÈRE wrote a quite complete overview of those techniques as an introduction in her PhD thesis [Ram06]. A way to discriminate those techniques is by considering how they take the shape of the physical domain into account:

1. via modifying the operators,

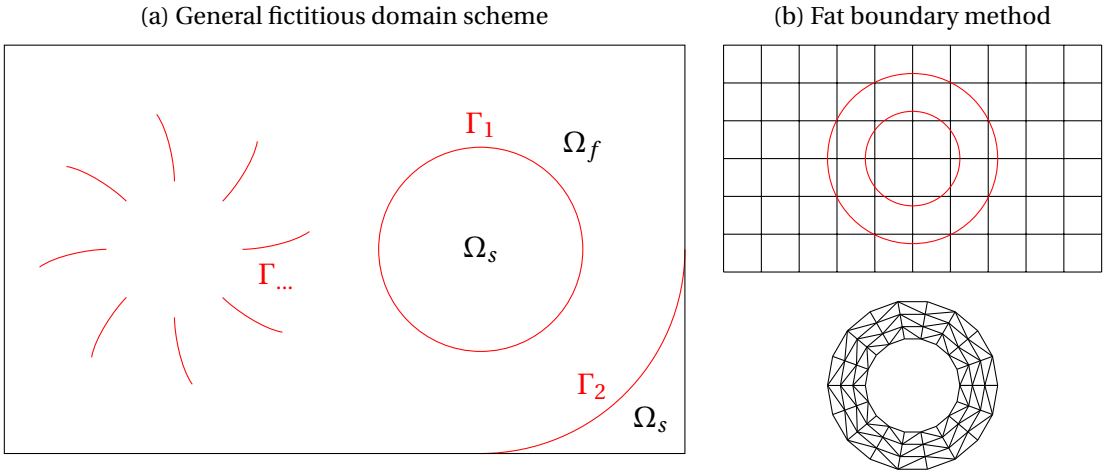


Figure 1.9. Schematic representations of some fictitious domain approaches.

2. via adding new terms in the balance equations.

***Remark* 1** *Of course, there are other ways to classify the various FDM such as:*

1. *When the interface is taken into account at a continuous level – and so is not so much adhesive to the discretization – such as penalty, Fat Boundary Method (FBM), Lagrangian Multipliers (LM), Immersed Interface Method (IIM), Immersed Spread Interface (ISI), Immersed Boundary Method (IBM), Ghost-Cell IBM (GCIBM), etc.*

2. *When the interface is taken into account at a discrete level – which means it is adhesive to the discretization – such as Cartesian Grid Embedded Method (CGEM), Cut Finite Element Method (C-FEM), Finite Cell Method (FCM) etc.*

or, as proposed by I. RAMIÈRE *in her PhD [Ram06]:*

1. *When the interface is spread such as FBM, ISI, IBM, GCIBM, FCM, etc.*

2. *When the interface is thin such as penalty, CGEM, LM, IIM, C-FEM, etc.*

Another key issue in the scope of FDM is data reconstruction on a boundary/interface or in its vicinity because it is the most common way to reach higher orders in space. Thus, various techniques have been developed in order to reconstruct fields coming from a boundary on the computation domain (*i.e.* the computation of the velocity imposed by the IB in our case) or the contrary (reconstruction of stress on the envelope of an airfoil to compute aerodynamic forces for instance [MWD20]) such as:

- Mollifier functions, used in the original IBM to approximate the Dirac delta functions [Pes72; RPB99].

- Extrapolation outside the computation domain, used in ghost cells techniques [TF03; BMZ16; CAT20].
- Interpolation, widely used in all kind of fictitious domain methods [Fad+00; IBF14; IK07; IV03; BG20].
- In the case of infinitely thin obstacles, the eXtended (or Generalized) Finite Element Method (X-FEM), which is often used in the field of fracture mechanics [Wag+01; YAA09], can give some answers: it is capable to deal with discontinuous quantities (typically tangential velocities on each side of an infinitely thin obstacle) while preserving the standard finite element properties elsewhere.

Let us notice that extrapolation and interpolation techniques, aside from involving polynomials or spline functions, can rather be directional (1D) [Fad+00] or spatial (multi-D) [IBF14]. Those two variants are not used in the same frameworks and do not require the same type of information about the boundary/interface.

As an illustrative example for the non-exhaustive list of fictitious domain methods presented in this chapter, let us consider a generic initial and boundary value problem with a Dirichlet Boundary Condition (BC):

$$\begin{cases} \partial_t \mathbf{y} + \theta(\mathbf{y}) = \mathbf{g} & \text{on } \Omega_f \\ \mathbf{y} = \mathbf{w} & \text{on } \partial\Omega_f \\ \mathbf{y}(t=0) = \mathbf{y}_0 & \text{on } \Omega_f \end{cases} \tag{1.1}$$

where:

- Ω_f is the physical (or fluid in the context of fluid dynamics) domain,
- $\mathbf{y} : \mathbb{R}^+ \times \Omega_f \to \mathbb{R}^d$ is the (vectorial) unknown,
- $\mathbf{g} : \mathbb{R}^+ \times \Omega_f \to \mathbb{R}^d$ is the source term,
- $\mathbf{w} : \mathbb{R}^+ \times \partial\Omega_f \to \mathbb{R}^d$ is the Dirichlet BC on $\partial\Omega_f$,
- $\mathbf{y}_0 : \Omega_f \to \mathbb{R}^d$ is the initial condition on Ω_f,
- θ is an operator only involving spatial derivatives of $\mathbf{y}$.

Ω_f is then embedded in a fictitious domain Ω on which, as mentioned before, the problem (1.1) is solved and the part of the BC lying on Γ are imposed by modifying the governing equations. Consequently, a new problem is defined on Ω:

$$\begin{cases} \partial_t \mathbf{u} + \Theta(\mathbf{u}) = \mathbf{f} & \text{on } \Omega \\ \mathbf{u} = \mathbf{a} & \text{on } \partial\Omega \\ \mathbf{u}(t=0) = \mathbf{u}_0 & \text{on } \Omega \end{cases} \tag{1.2}$$

where Θ (resp. $\mathbf{u}$, $\mathbf{f}$, $\mathbf{a}$ and $\mathbf{u}_0$) is the extension in Ω of θ (resp. $\mathbf{y}$, $\mathbf{g}$, $\mathbf{w}$ and $\mathbf{y}_0$). The way of modifying the balance equations (*i.e.* the form of Θ and $\mathbf{f}$) leads to different techniques which, as said previously, can be separated in two categories: the ones adding new terms and the ones modifying the operators (cf. sections 1.4.1 and 1.4.2).

1.4.1. Methods modifying operators

1.4.1.1. Immersed Interface Method

This method was introduced by R.J. LEVEQUE and Z. LI [LL94] in order to enhance the Immersed Boundary Method developed by C.S. PESKIN [Pes72] (cf. section 1.4.2.6). The idea is to solve PDE with discontinuous or singular coefficients or source terms by taking into account potential jumps of the computed quantities – and their derivatives – through one or several interfaces while preserving a uniform Cartesian grid. The jumps are then considered as input data of the problem.

Z. LI wrote a quite complete review about this method and its application with more than eighty references within [Li03]. Originally, the method was based on a finite difference scheme but it can be – and has been – adapted to a finite element formulation [Li98]. Indeed, the jumps are modeled via source terms in the equations – directly in the normal direction, with respect to the interface(s), and with discrete delta functions in the tangential direction. Those source terms are non-zero only in the cells for which the stencil extend to a cell cut by the interface. In their article [LL03], L. LEE and R. J. LEVEQUE tend to show that only considering a pressure jump at the interface can significantly enhance the fidelity of the results when compared to a classical IBM. Moreover, adding this correction does not seem extremely complex in terms of implementation.

In the same idea, P. ANGOT developed an augmented IIM that he called Jump Embedded Boundary Condition (JEBC) [Ang10; RAB07b]. It is augmented in the sense that the problem (Brinkman in [Ang10]) is enriched with jump equations. Those equations link the stress vector and velocity jumps. Later on, he proposed, together with Z. LI, an enhancement of his method in which a preconditioner was added. Numerical results are provided in [AL17].

We can also mention the Matched Interface and Boundary (MIB) method by Y. C. ZHOU *et al.* [ZW06] in which the solution on fictitious points is extrapolated from one side to the other. The force of this method rely on the fact that it dissociates the discretization of the equations and the enforcement of interface jump conditions. Based on this approach, we can also cite the work of M. CISTERNINO and L. WEYNANS [CW12] (more details in the open-access INRIA research report [Wey19]) in which the additional unknowns are not located on grid points on the other side of the interface but on the interface itself.

This way of dealing with interfaces is interesting in the framework of my PhD because it can apply to thin obstacles and can be adapted whether for FEM or FV. Adding this pressure jump correction can even be implemented in the context of IBM

as in [Ria+18] for instance.

1.4.1.2. Cartesian Grid Embedded Method

The Cartesian Grid Embedded method (CGEM) is adhesive to the spatial discretization. Indeed, it relies on a Finite Volumes (FV) scheme with the values of the unknown computed at the center of the cells. The boundary of the computational domain is interpolated linearly within the cells. Then, using the normal of the interpolated boundary, the boundary flux are also interpolated – quadratically [JC98] and, more recently, with a fourth-order scheme [Dev+17] – but only using values which are actually in the computational domain (*i.e.* interpolation and not extrapolation).

Unlike the IIM, the CGEM does not add any local unknowns to the problem. Moreover, because it preserves a Cartesian grid, it can easily be combined with multi-level Adaptative Mesh Refinement (AMR). This technique has been used to solve the Poisson equations for both Neumann and Dirichlet BC in 2D and 3D (cf. [JC98], [Sch+06] and references within) with moving and non-moving irregular boundaries.

However, as said in [JC98] on page 8, this method, by its construction of the interface, is not adapted for thin boundaries (*i.e.* smaller than the Cartesian grid step) which are under-resolved. This does not fit the framework of my PhD.

1.4.1.3. Finite Cell Method

The Finite Cell Method can be seen as an "extension" of the standard FEM. In [PDR07], linear elasticity problems and their extension to an embedding simpler domain Ω, similarly to problems (1.1) and (1.2), are solved using the FEM – material properties are extrapolated outside of the physical domain and the extrapolation value depends on the type of immersed boundary condition considered (Dirichlet or Neumann). However, the elements that are crossed by Γ are called "finite cells" and are treated differently (*i.e.* operators are modified): variables are smoothly extended using high order Ansatz functions. Numerical, theoretical aspects and error estimate have been studied in [DDR15; DS21].

This method is mainly used in the context of solid mechanics [Düs+08] and considers only massive obstacles which is not interesting in the framework of this PhD.

1.4.1.4. Cut Finite Element Method

The Nitsche's method, [Nit71] which consist in weakly enforcing essential boundary conditions via the addition of jump terms in the variational formulation of the problem, gave birth to the unfitted finite element method [HH02] and, later, to the cut finite element method [HLZ14] – the idea was to propose an alternative to the eXtended Finite Element Method (X-FEM) which usually leads to ill conditioning. The specificities of the method rely on a weighting of the Nitsche numerical fluxes, a change in the variational formulation to avoid spurious oscillations and the addition of stabilization terms.

This method has been successfully applied to multifluid Stokes equations in [HLZ14] as well as solid mechanics (fractured media in [BHL20] and elastic wave equations in [SLK20]).

1.4.1.5. Generalized and eXtended Finite Elements Methods

The Generalized Finite Element Method (GFEM) is a numerical method based on the FEM and the partition of unity. It consists in taking into account some knowledge we can have about the solution of a problem via locally enriching the solution space (*i.e.* add special basis functions where we have knowledge of the solution). This method gave birth to another variant called the eXtended Finite Element Method (X-FEM) which allows enriching with discontinuous functions. Those methods are widely used in material mechanics [BGV09], especially fracture propagation [YAA09], as they are capable of building the approximation on meshes independent of the geometry of the computation domain.

This approach could be interesting in order to take into account knowledge about turbulent wall laws, as well as infinitely thin boundaries using Heaviside-like basis functions.

1.4.2. Methods adding new terms

1.4.2.1. Ghost Cells

As mentioned in [TF03], the ghost-Cell Immersed Boundary Approach (GCIBM) is inspired by the work of E. A. FADLUN regarding IBM [Fad+00] and Ghost Fluid Method (GFM) developed by R. P. FEDKIW to model multimaterial flows [Fed02]. Indeed, it relies on a new type of forcing which includes an extrapolation of the fluid velocity within the "ghost-cells" (*i.e.* solid domain).

This technique, for instance, has been successfully applied to the Euler equations to model inviscid gas dynamics in [BMZ16] and to the incompressible Navier-Stokes to model blood flows in [CAT20]. This approach only considers massive and not infinitely thin obstacles.

1.4.2.2. Lagrange multipliers

Probably one of the most known, the Lagrangian multipliers (or distributed frontier) method was originally developed by R. GLOWINSKI – [GPP94] and references within. In order to keep a Cartesian grid for the fictitious domain, another mesh, non-structured, is used to compute the Lagrange multipliers. This second mesh can either be a discretization of the immersed boundary (i.e. a line in 2D and a surface in 3D) or a discretization of the computation domain – fluid domain in the context of my PhD – (i.e. a volume in 3D). The immersed boundary conditions are taken into account via a constraint, which is weakly imposed. This leads to an optimization problem consisting in solving a linear system.

This method has been used for the Dirichlet (cf. [GPP94] and references within) and Neumann (cf. [Glo+99] and references within) problems, mainly in the framework of particles flow. V. GIRAULT and R. GLOWINSKI also published about the error analysis of the method in [GG95].

1.4.2.3. Fat Boundary Method

The original method proposed by B. MAURY [Mau01] lays on a bi-grid formulation. A Cartesian grid covers the whole computation domain while a local finer body-fitted grid adapted to the immersed boundary (cf. Fig. 1.9b). Those grids induce two problems – coupled by a mesh interpolation and a jump condition of the normal derivative of the variables at the interface (as in the IIM [Li03]) –, which are solved thanks to a fixed-point algorithm.

This method was first applied for a domain containing circular holes. As the geometry of such holes is quite simple, it provided good results. Later on, it has been used to solve the Navier-Stokes equations in the context of particular flow [Ism04].

However, in the framework of my PhD, this method is not applicable. Indeed, as the Lagrange multipliers method, it adds many unknowns – due to the bi-grid approach – which seems incompatible with the principle of a fast tool.

1.4.2.4. Penalty

Those techniques have mainly been used in the context of fluid dynamics, in order to impose Dirichlet type conditions on Γ or in Ω_S (to impose a no-slip condition for instance). There are mainly two ways to penalize a fluid dynamics problem (cf. [ABF99] and [Ang99] for a mathematical approach and application to fluid-porous flows):

- $\mathscr{L}^2$ penalty: it consists in adding a reaction term to the Right-Hand-Side (RHS) of the PDE, solved on the fictitious domain (1.2), so that $\partial_t \mathbf{u} + \Theta(\mathbf{u}) = \mathbf{f} + \mathbf{b}$ where $\mathbf{b} = \chi_\Gamma \eta^{-1} (\mathbf{u} - \mathbf{u}_\Gamma)$ is the penalty term, χ_Γ the characteristic function of Ω_Γ, $\mathbf{u}_\Gamma$ the variable value imposed by the Dirichlet BC on Γ and $\eta \ll 1$ the penalty parameter.
- $\mathscr{H}^1$ penalty: it consists in, like the $\mathscr{L}^2$ penalty, adding a reaction term but also penalizing the diffusion and/or time derivative term (i.e. multiplying by the inverse of η).

Even if this method is filed in the thin interface approach, volume penalty, or Sub-Mesh Penalty (SBM) [Sar+08], can also be used to inforce Dirichlet BC (notably in [Sar09] and [RAB07a]). Concerning this matter, we can also cite the Implicit Tensorial Penalty Method by S. VINCENT and J.P. CALTAGIRONE [Vin+07; CV01]. As the name indicates, it consists in applying penalty directly in the stress tensor by introducing new viscosity coefficients in order to impose either incompressibility, shearing or the solid rotation of the flow.

The estimation of the error induced by the penalty parameter are also discussed in [ABF99] and [Ang99]. Indeed convergence properties and theorem are given for the Navier-Stokes and Brinkman models.

1.4.2.5. Immersed Spread Interface

This fictitious domain approach developed by I. RAMIÈRE *et al.* [RAB07a] is inspired from the penalty techniques. In this formulation, the approximate interface is composed of all the cells crossed by the interface. Embedded BC are enforced by either $\mathscr{H}^1$ or $\mathscr{L}^2$ volume penalty according to their type (Dirichlet or Robin). In addition to this, jump of the convective and diffusive flux across the interface are considered as source terms in the governing equations and determined by the BC. This method was applied for diffusion and convection-diffusion problems in a FE framework.

The spread interface approach provided very interesting results knowing that it does not add any unknown and that the order 1 representation of the interface is balanced by the modeling of the flux jumps at the interface. This method has been a source of inspiration for the Penalized Direct Forcing (cf. section 1.4.2.6).

1.4.2.6. Immersed Boundary Method

The Immersed Boundary Method (IBM) was originally introduced by C. S. PESKIN to model cardiac mechanics and blood flow [Pes72]. The original idea was to take into account elastic boundaries via a backmoving force imposed at the interface by the mean of Dirac delta functions. This force is considered as a source term added to the governing equations at a continuous level. At a discrete level, this approach implies two meshes:

- A volumetric mesh – often Cartesian – on which the Eulerian variables are computed (fluid velocity, pressure, density, *etc.*)
- A curvilinear mesh – corresponding to the immersed boundary – on which Lagrangian variables are computed (velocity and position of the immersed boundary, backmoving force, *etc.*)

The backmoving force computed at the interface is then distributed on several cells on each side of the boundary by the mean of discrete Dirac delta functions. The interested reader can refer to [Pes02] for a more precise explanation, a review of different techniques and many references about this topic.

This method is quite simple to implement and it preserves the use of a Cartesian Eulerian grid. However, the main disadvantage occurs when rigid boundaries or moving deformable boundaries are considered. With rigid boundaries (instead of elastic boundaries), numerical instabilities may appear and, in the case of deformable moving boundaries, the position and number of Lagrangian markers needed to describe the boundary must be computed at each time step and spurious oscillations

can appear. However, the issue of moving immersed boundaries is not addressed in my PhD.

Then J. MOHD-YUSOF developed a new IBM, based on the one of C.S. PESKIN, often referred as "Direct Forcing" [Moh97]. Let us consider the following time discretization of problem (1.2):

$$\begin{cases} \dfrac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\delta t} + \Theta_h(\mathbf{u}^{n+1},\mathbf{u}^n) = \mathbf{f}^{n+1} & \text{on } \Omega \\ \mathbf{u}^n = \mathbf{a}(t=t_n) & \text{on } \partial\Omega \\ \mathbf{u}^0 = \mathbf{u}_0 & \text{on } \Omega \end{cases} \tag{1.3}$$

where:

- δt is the time step,
- n is the time index,
- $t_n = n\,\delta t$ the time at index n,
- Θ_h is a time-discretization of the operator Θ,
- $\mathbf{u}^n : \Omega \to \mathbb{R}^d$ is the unknown at t_n.
- $\mathbf{f}^n : \Omega \to \mathbb{R}^d$ is the source term at t_n.

Similarly to a projection scheme[BCM01], the "Direct Forcing" method adds an intermediate step (with an intermediate variable $\widehat{\mathbf{u}}$) to the algorithm in order to compute its forcing term:

$$\begin{cases} \dfrac{1}{\delta t}(\widehat{\mathbf{u}}-\mathbf{u}^n) + \Theta_h(\widehat{\mathbf{u}},\mathbf{u}^n) = \mathbf{f}^{n+1} & \text{on } \Omega \\ \dfrac{1}{\delta t}(\mathbf{u}^{n+1}-\mathbf{u}^n) + \Theta_h(\mathbf{u}^{n+1},\mathbf{u}^n) = \mathbf{f}^{n+1}+\mathbf{b}^{n+1} & \text{on } \Omega \\ \mathbf{u}^n = \mathbf{a}(t=t_n) & \text{on } \partial\Omega \\ \mathbf{u}^0 = \mathbf{u}_0 & \text{on } \Omega \end{cases} \tag{1.4}$$

with:

$$\mathbf{b}^{n+1} = \chi\left[\Theta_h(\widehat{\mathbf{u}},\mathbf{u}^n) - \mathbf{f}^{n+1} + \frac{1}{\delta t}(\mathbf{u}_\Gamma^{n+1}-\mathbf{u}^n)\right] \tag{1.5}$$

and:

- χ is the characteristic function Γ,
- $\mathbf{u}_\Gamma^{n+1} : \Gamma \to \mathbb{R}^d$ is the Dirichlet BC imposed on Γ at time index $n+1$.

This approach is explicit, in the sense that $\mathbf{b}^{n+1}$ does not depends on $\mathbf{u}^{n+1}$. An implicit version can be built with:

$$\mathbf{b}^{n+1} = \chi \left[\Theta_h(\mathbf{u}^{n+1}, \mathbf{u}^n) - \mathbf{f}^{n+1} + \frac{1}{\delta t}(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^n) \right] \tag{1.6}$$

This way, the terms relative to the operator Θ_h and the RHS are cancelled in Γ, imposing $\mathbf{u}^{n+1} = \mathbf{u}_\Gamma^{n+1}$. However, in practice, this generally imply an iterative scheme for each time step (with $\mathbf{u}_k^{n+1}$ the variable at iteration k, k_{fin} the maximum number of iteration and $\epsilon \ll 1$ the stop parameter):

1. **Initialization** ($k = 0$): Solve:

$$\frac{1}{\delta t}(\mathbf{u}_0^{n+1} - \mathbf{u}^n) + \Theta_h(\mathbf{u}_0^{n+1}, \mathbf{u}^n) = \mathbf{f}^{n+1} \tag{1.7}$$

2. **Iteration** ($k > 0$): Solve:

$$\frac{1}{\delta t}(\mathbf{u}_k^{n+1} - \mathbf{u}^n) + \Theta_h(\mathbf{u}_k^{n+1}, \mathbf{u}^n) = \mathbf{f}^{n+1} + \mathbf{b}^{n+1} \tag{1.8}$$

   with:

$$\mathbf{b}^{n+1} = \chi \left[\Theta_h(\mathbf{u}_{k-1}^{n+1}, \mathbf{u}^n) - \mathbf{f}^{n+1} + \frac{1}{\delta t}(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^n) \right] \tag{1.9}$$

3. **Finalization** ($k > k_{\text{fin}}$ or $|\mathbf{u}_k^{n+1} - \mathbf{u}_{k-1}^{n+1}| < \epsilon$):

$$\mathbf{u}^{n+1} = \mathbf{u}_k^{n+1} \tag{1.10}$$

To prevent the use of an iterative scheme, M. BELLIARD *et al.* developed a new technique, called Penalized Direct Forcing (PDF), inspired by the penalty and Direct Forcing methods [BF10]. The PDF method has been applied to the incompressible Navier-Stokes equations and, in this case, the forcing term is written as follows:

$$\mathbf{b} = \frac{\chi_\Gamma}{\eta \delta t}\left(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^{n+1}\right) \tag{1.11}$$

with $\eta \in \mathbb{R}^{+*}$, such as $\eta \ll 1$. This formulation provides several advantages:

- Computing the forcing term does not need extra steps (contrary to DF),
- The formulation has good mathematical properties and adaption to fractional-step schemes due to the Direct Forcing formulation.

This method has been applied together with a finite difference formulation involving a Markers And Cells (MAC) grid (cf. [BF10] and [IBF14]). To reach order 2, in the cells cut by the immersed boundary, the imposed velocities at the degrees of freedom (*i.e.* at the faces of the cells in case of a MAC grid) are linearly interpolated from the immersed

boundary. However, the penalty parameter η can imply very badly conditioned matrix. Diagonal preconditioning easily cures this.

Otherwise, IBM are often combined with fractional-step methods. T. IKENO *et al.* showed that increasing the order of the interpolation scheme does not significantly improve the fidelity of the results if the forcing term is only added during the prediction step [IK07]. Indeed, they proposed to define two different forcing terms for the prediction and projection equations, which seems to enhance the accuracy of their method. For more recent work on this subject, the interested reader can refer to F. DOMENICHINI [Dom08] or R.D. GUY *et al.* [GH10]. C. INTROÏNI *et al.* applied this technique to the PDF method and came to the same conclusion, reaching a second order IBM scheme [IBF14].

1.5. Summary and choice of the method

In Section 1.2, we discussed the aimed application and clarified the kind problematic in which the CEA is interested: modeling turbulent two-phase around or through complex thin geometries in order to find the optimal design of some innovative nuclear passive safety systems. An efficient way to achieve this goal would be topological shape optimization. However, to build a statistical answer surface, a numerical modeling tool is required to produce a large number of simulations in various configurations. The purpose of this PhD project is to design and develop this tool.

The typical way of achieving the numerical modeling of this kind of problems is via grid-based numerical solving of IBVP (*cf.* Section 1.3). However, to match the aimed application, we cannot afford a "body-fitted" approach (too costly in terms of computational resources). Thus, we opted for the Fictitious Domain approach in order to decouple the geometry of the computation domain from the geometry of the physical domain (less costly and easier grid generation).

In addition to the above-mentioned constraints, we want to remain in a Finite Element (FE) framework for two main reasons:

- A CEA FE house-code at component scale, called GENEPI, contains interesting models for the aimed application:
  - Homogeneous Equilibrium Model that can handle two-phase flow with only three equations (*cf.* Section 3.1.1),
  - Tabulated Equation Of State (EOS) of water,
  - Weak compressibility model,
  - *etc.*
- The FE framework provides a natural vector basis to achieve interpolation (*cf.* Chapter 4) or extrapolation. As we ultimately want our tool to handle turbulence modeling, this could be very useful to implement wall-law-based interpolation

(this work is currently in progress in the new PhD project that has been launched in October 2020).

Finally, we chose the Penalized Direct Forcing method for the following reasons:

- It does not depend strongly on the type of space discretization and can therefore be adapted easily. As we are using the CEA industrial codes, we cannot afford to change the space discretization because it would mean restarting the validation process from the beginning.
- It does not involves additional unknowns (we want the numerical simulation tool to remain the fastest possible).
- As it does not need extrapolation techniques to reach order 2, it can easily be adapted to infinitely thin obstacles with fluid on both sides.
- It has already been adapted to a projection scheme (GENEPI uses a projection scheme to decouple velocity and pressure).
- It does not need extra iterative scheme.
- It can reach order 2 in space via linear interpolation.
- It is simple to implement.
- IBM have been widely used for turbulent flows, as evidenced by the extensive review of G. IACCARINO and R. VERZICCO [IV03].

Hence, to summarize briefly, our work in this PhD project mainly consisted in adapting the Penalized Direct Forcing method to a Finite Element formulation. We also proposed and developed adapted ways to achieve order 2 in space via linear interpolation. In Chapter 2, we focus on the different methods that we developed in order to recover geometrical data from the approximate immersed obstacles. To achieve order 1 in space, only the approximation of the characteristic function χ is needed whereas, to achieve order 2 via linear interpolation, the approximate normal projections of the nodes in the vicinity of the obstacles are also needed. Chapter 3 presents the physical modeling and the numerical methods involved in the resolution of the governing PDE (time discretization, space discretization and general principle of the PDF method). Then, Chapter 4 discusses the order of the PDF method and provides detailed explanations about the different methods proposed to achieve order 2 in space via linear interpolation. Finally, Chapter 5 compiles all simulation results among academic and industrial cases. It provides valuable validation data as well as proof of the capability of the method regarding a complex 3D geometry.

2. Geometrical data reconstruction

Summary

2.1. Preamble

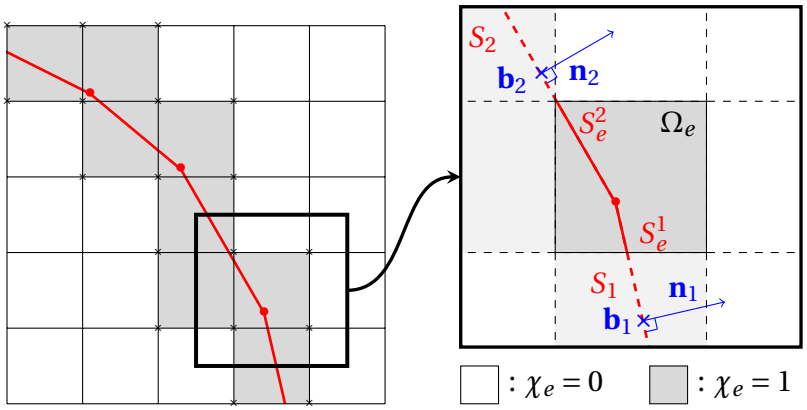


Figure 2.1. Schematic representation of the computation grid and the collection of facets associated to the immersed boundary.

The object of this chapter is the to explain the different way used to recover useful geometrical data from the immersed boundary input data. Those input data, representative of the shape of the immersed obstacle, can come from various sources and take various forms. Without any claim to be exhaustive, we formally define two categories:

- Analytical equations (simple shapes like circles, squares, *etc.*),
- Discrete values (measurements, experiments, Computer Aided Design, *etc.*).

Obviously, the second category is wider than the first one (discrete values can be generated from analytical equations) so, in order to remain as general as possible, we consider the approximate obstacle Γ_h (representative of Γ) as a union of discrete elements, here plane convex polygons called facets:

$$\Gamma_h = \bigcup_{i=1}^{N_S} S_i \tag{2.1}$$

with N_S the total number of facets, $(S_i)_{i\in[\![1,N_S]\!]}$ the collection of facets and their respective normal vectors $(\mathbf{n}_i)_{i\in[\![1,N_S]\!]}$ and barycenters $(\mathbf{b}_i)_{i\in[\![1,N_S]\!]}$. Let us also consider a volume mesh for the fictitious domain Ω, composed of N_E "elements", or "cells" to remain general, denoted $(K_e)_{e\in[\![1,N_E]\!]}$ with $\Omega_e \subset \Omega$ the portion of the FD included in K_e – as an illustrative example, without loss of generality, we consider a 2D Cartesian grid for purpose of pedagogy (*cf.* Figure 2.1).

In order to achieve a first order approximation (or "staircase" approximation) of the immersed boundary, we only need to define a discretization for the characteristic function (*cf.* Section 2.2). However, if we want to achieve a second order approximation via linear interpolation, the normal projection on Γ_h of each node of the grid in its vicinity is also needed (*cf.* Section 2.3). To achieve this normal projection on Γ_h, we propose two approaches: weighting (*cf.* Section 2.3.1) and optimization (*cf.* Section 2.3.2) – each including several variants. Then, those approaches are compared over two cases: one involving a regular geometry and the other involving a non-regular one (*cf.* Section 2.3.3).

2.2. Characteristic function of the IB

As mentioned in the introduction (*cf.* chapter 1), the first issue to address is the spatial discretization of the characteristic function χ. To achieve it, we can define S_e^i, the portion of facet S_i contained in Ω_e (*cf.* Figure 2.1), as follows:

$$\forall e \in [\![1,N_E]\!], \forall i \in [\![1,N_S]\!], \quad S_e^i = S_i \cap \Omega_e \tag{2.2}$$

Then we can define the elemental (or cellular) discretization of χ, denoted χ_e:

$$\forall e \in [\![1,N_E]\!], \begin{cases} \chi_e = 1 & \text{if} \quad \exists i\, [\![1,N_S]\!] \ / \ S_e^i \neq \emptyset \\ \chi_e = 0 & \text{else} \end{cases} \tag{2.3}$$

2.3. Normal projection of a node on the IB

As mentioned in the preamble of this chapter, to reach order 2 in space, we use linear interpolation (*cf.* Chapter 4). Hence, for the positions of degrees of freedom in the vicinity of the immersed boundary, we need an approximate normal projection on the obstacle. In the following sections, we proposed two ways to achieve the projection:

1. an approach based on the weighting of the facets normal vectors (*cf.* section 2.3.1). In this case, we use Γ_h to compute elemental data (intersected area, barycenter, normal vector) and then assemble those data in a FE fashion to get nodal data (projection on Γ_h).

2. an approach based on a minimization problem, inspired from [IBF14] (*cf.* section 2.3.2). In this case we directly use Γ_h to compute nodal data, without using intermediary elemental data.

Before explaining those two approaches, let us introduce some notations:

- $J_e = \left\{ j \in [\![1, N_N]\!] \,\middle|\, \mathbf{x}_j \in \Omega_e \right\}$: the set of node indexes included in element K_e,
- $E_j^0 = \left\{ e \in [\![1, N_E]\!] \,\middle|\, j \in J_e \right\}$: the set of element indexes which share the node j,
- $E_j^\chi = \left\{ e \in E_j^0 \,\middle|\, \chi_e = 1 \right\}$: the set of element crossed by Γ_h which share the node j,
- $J = \{ j \in [\![1, N_N]\!] \,|\, \exists e \in E_j^0 / \chi_e = 1 \}$: the set of node indexes for which a projection point need to be computed (points marked with a cross on Figure 2.1).

2.3.1. Weighting approach

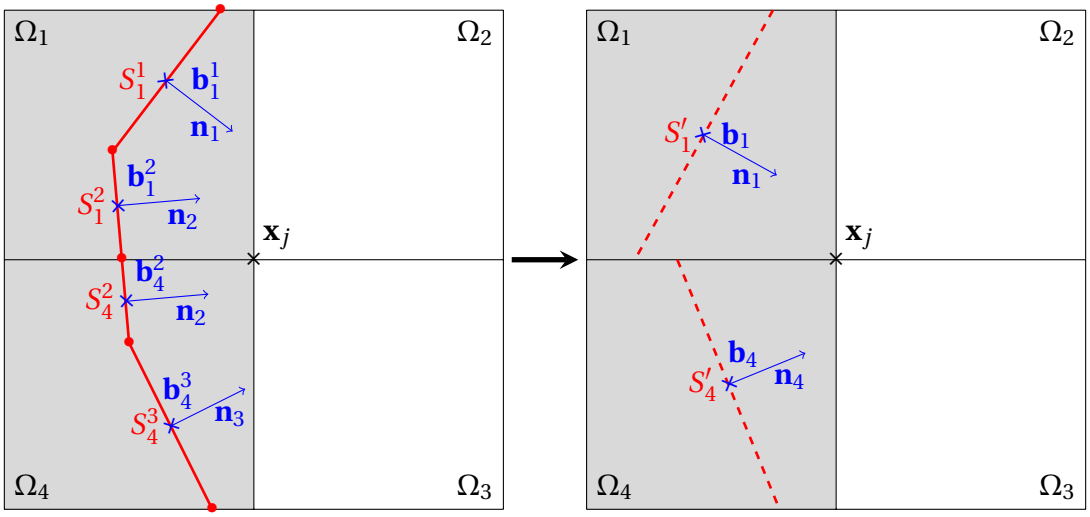


Figure 2.2. Schematic representation of the weighting approach for a node j. In this particular example, we have $i \in \{1,2,3\}$ and $e \in E_j^0 = \{1,2,3,4\}$. Gray elements are element in which $\chi_e = 1$.

For each S_e^i, we can compute a barycenter, denoted $\mathbf{b}_e^i$ (*cf.* Figure 2.1), and an area, denoted $\mathscr{A}_e^i$:

For each S_e^i, we can compute an area, denoted $\mathscr{A}_e^i$, and a barycenter, denoted $\mathbf{b}_e^i$ – $\mathbf{n}_i$ (*cf.* Figure 2.1):

$$\forall i \in [\![1, N_S]\!] \ , \ \forall e \in [\![1, N_E]\!] \ , \ \mathscr{A}_e^i = \kappa(S_e^i) \tag{2.4}$$

with $\kappa(S_e^i)$ the measure of S_e^i ($\kappa(\emptyset) = 0$). Note that S_e^i and S_i share the same normal vector $\mathbf{n}_i$

Now, using area weighting, we can compute elemental data:

$$\forall e \in [\![1, N_E]\!] \ / \ \chi_e = 1, \begin{cases} \mathscr{A}_e = \displaystyle\sum_{i=1}^{N_S} \mathscr{A}_e^i \\ \mathbf{n}_e = \dfrac{1}{\mathscr{A}_e} \displaystyle\sum_{i=1}^{N_S} \mathscr{A}_e^i \mathbf{n}_i \\ \mathbf{b}_e = \dfrac{1}{\mathscr{A}_e} \displaystyle\sum_{i=1}^{N_S} \mathscr{A}_e^i \mathbf{b}_e^i \end{cases} \tag{2.5}$$

In a way, the triplet $(\mathscr{A}_e, \mathbf{n}_e, \mathbf{b}_e)$ represents a plane facet S'_e that is considered "equivalent" to the collection $(S_i)_{i \in [\![1, N_S]\!]}$ in the element K_e (*cf.* Figure 2.2). Then, each node j of an element e can be projected on this equivalent plane facet, which gives:

$$\forall e \in [\![1, N_E]\!] \ / \ \chi_e := 1, \forall j \in J_e, \quad \mathbf{p}_e^j := \mathbf{x}_j + l_e^j \mathbf{n}_e \tag{2.6}$$

with $\mathbf{x}_j$ the coordinates of node number j, $\mathbf{p}_e^j$ the coordinates of the projection of node number j on the plane facet associated to K_e and $l_e^j = (\mathbf{b}_e - \mathbf{x}_j) \cdot \mathbf{n}_e$ the oriented normal distance between the node j and the plane facet associated to K_e – note that, in practice, we choose the orientation of $\mathbf{n}_e$ such as $l_e^j < 0$. Finally, $\mathbf{p}_e^j$ is assembled in a FE way to obtain:

$$\forall j \in J, \quad \alpha_j = \sum_{e \in E_j^\chi} \alpha_e^j, \quad \mathbf{x}_j^p = \frac{1}{\alpha_j} \sum_{e \in E_j^\chi} \alpha_e^j \mathbf{p}_e^j \tag{2.7}$$

with $\mathbf{x}_j^p$ the coordinates of the actual discrete projection of node j and α_e^j: the weight. Several variants, depending on the value of α_e^j, have been developed: arithmetic mean, area weighting, invert distance weighting and area over distance weighting.

1. $\alpha_e^j = 1$, arithmetic mean
2. $\alpha_e^j = \mathscr{A}_e$, area weighting
3. $\alpha_e^j = \dfrac{1}{|l_e^j|}$, invert distance weighting

4. $\alpha_e^j = \dfrac{\mathscr{A}_e}{|l_e^j|}$, area over distance weighting

2.3.2. Optimization approach

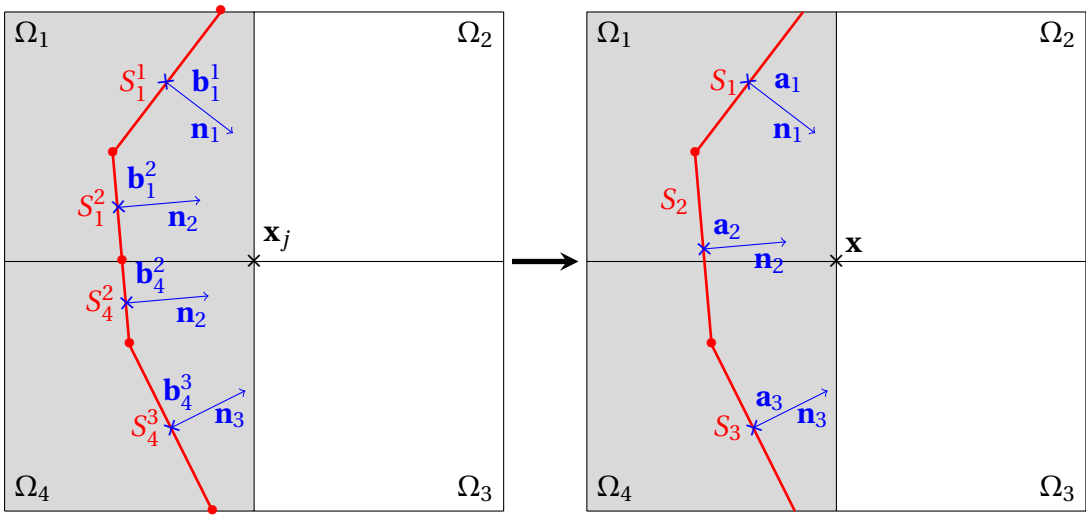


Figure 2.3. Schematic representation of the optimization approach for a node j. In this particular example, we have $I_j = \{1,2,3\}$. Gray elements are element in which $\chi_e = 1$.

In this approach, for a given node, we search for the point, located on Γ_h, which minimizes the distance between itself and the considered node – method inspired by C. Introïni *et al.* [IBF14]. The optimization problem can be written as follows:

$$\forall j \in J, \quad \text{Find} \quad \mathbf{x}_j^p \in \Gamma_h \quad \text{such as} \quad (\mathbf{x}_j - \mathbf{x}_j^p)^2 = \inf_{\mathbf{y}\in\Gamma_h} (\mathbf{x}_j - \mathbf{y})^2 \tag{2.8}$$

To solve this problem for each $j \in J$, we use the algorithm proposed in [IBF14]. First, the immersed boundary is partially reconstructed in the vicinity of $\mathbf{x}_j$ by collecting the facets of Γ_h which intersect at least one element sharing the node number j (the indexes of those facets form a set denoted I_j, *cf.* Figure 2.3). Then, for each of those facets, we write the equation of the plane passing through it:

$$\forall i \in I_j, \quad \mathbf{n}_i \cdot \mathbf{y} = \mathbf{n}_i \cdot \mathbf{a}_i \tag{2.9}$$

with:

$$\forall j \in J, I_j = \left\{ k \in [\![1, N_S]\!] \,\middle|\, \exists e \in E_j^0 \;/\; S_e^k \neq \emptyset \right\} \tag{2.10}$$

and $\mathbf{a}_i \in \mathbb{R}^d$ the coordinates of a point belonging to S_i (barycenter or any other point). Gathering all the collected facets for a node j, the following linear system is obtained:

$$\mathbf{C}^j \mathbf{y} = \mathbf{r}^j \tag{2.11}$$

with:

$$\forall \in j \in J, \forall i \in I_j, \forall k \in [\![1,d]\!], \quad C^j_{ik} = n_{i,k} \tag{2.12}$$

and:

$$\forall \in j \in J, \forall i \in I_j, r^j_i = \sum_{k=1}^{d} n_{i,k} a_{i,k} \tag{2.13}$$

where:

- C^j_{ik} is the general term of $\mathbf{C}^j \in \mathbb{R}^{s\times d}$ with $s = \text{card}\left(I_j\right)$
- $n_{i,k}$ the kth component of vector $\mathbf{n}_i$
- $a_{i,k}$ the kth component of vector $\mathbf{a}_i$

Finally, we can consider that a point of coordinates $\mathbf{y}$ which verifies, at least approximately, the system (2.11) is located on Γ_h, or at least in its vicinity – *i.e.* solving this linear system is a way to impose the constraint of the minimization problem (2.8). If we formally denote $\mathcal{V}^j_{\Gamma_h} = \left\{\mathbf{y} \in \mathbb{R}^d \,\middle|\, \mathbf{C}^j\mathbf{y} \approx \mathbf{r}^j\right\}$, the set of the points which approximately verify the system (2.11), the minimization problem (2.8) becomes:

$$\forall j \in J, \quad \text{Find} \quad \mathbf{x}^p_j \in \mathcal{V}^j_{\Gamma_h} \quad \text{such as} \quad (\mathbf{x}_j - \mathbf{x}^p_j)^2 = \inf_{\mathbf{y}\in\mathcal{V}^j_{\Gamma_h}} (\mathbf{x}_j - \mathbf{y})^2 \tag{2.14}$$

This problem is solve using an Uzawa algorithm for each node j, with $\epsilon \ll 1$ the shutoff parameter, z_j a real parameter that indicates on which side of Γ_h the point $\mathbf{x}_j$ is located (with respect to the orientation of the normal vectors $(\mathbf{n}_i)_{i\in I_j}$), $x_{j,k}$ the kth component of vector $\mathbf{x}_j$ and Λ^m the Lagrange multiplier vector at iteration m:

1. *Initialization* $(m = 0)$:
   - $\Lambda^0 = \mathbf{0}_{\mathbb{R}^s}$
   - $z_j = \sum\limits_{i\in I_j} \left(\sum\limits_{k=1}^{d} \left(C_{ik} x_{j,k}\right) - r^j_i \right)$
2. *Iterations* $(m > 0)$:
   - $\mathbf{y}^m = \mathbf{x}_j - \dfrac{1}{2}\left(\mathbf{C}^j\right)^T \Lambda^{m-1}$
   - $\mathbf{i}^m = \dfrac{\mathbf{C}^j\mathbf{y}^m - \mathbf{r}^j}{\left\|\mathbf{C}^j\right\|_1 \left\|\mathbf{C}^j\right\|_\infty}$
   - if $z_j > 0$:

- $\Lambda^m = \max\left(\Lambda^{m-1} + \mathbf{i}^m, 0\right)$
- else:
  - $\Lambda^m = \min\left(\Lambda^{m-1} + \mathbf{i}^m, 0\right)$

3. *Finalization*:
   - if $(\mathbf{y}^{m-1} - \mathbf{y}^m)^2 < \epsilon$ then $\mathbf{x}_j^p = \mathbf{y}^m$

Another variant of this approach has also been developed. It relies on a different definition of I_j: instead of considering only the elements sharing the node j to collect the facets of Γ_h, we also consider the neighbors of those elements. Formally, it can be written as follows:

$$\forall j \in J, \quad I_j = \left\{ k \in [\![1, N_S]\!] \,\middle|\, \exists e \in E_j^1 \ / \ S_e^k \neq \emptyset \right\} \tag{2.15}$$

with:

$$\forall j \in J, \quad E_j^1 = \left\{ e \in [\![1, N_E]\!] \,\middle|\, \exists k \in J_e \ / \ \exists g \in E_j^0 \ / \ k \in J_g \right\} \tag{2.16}$$

Those two variants are designated by what we call their level l which refers to the exponent of E_j^l in the definition of I_j. Thus, the level 0 approach considers only direct neighbor elements of a given node to collect facets whereas the level 1 also considers neighbors of direct neighbor elements (*cf.* Figure 2.4).

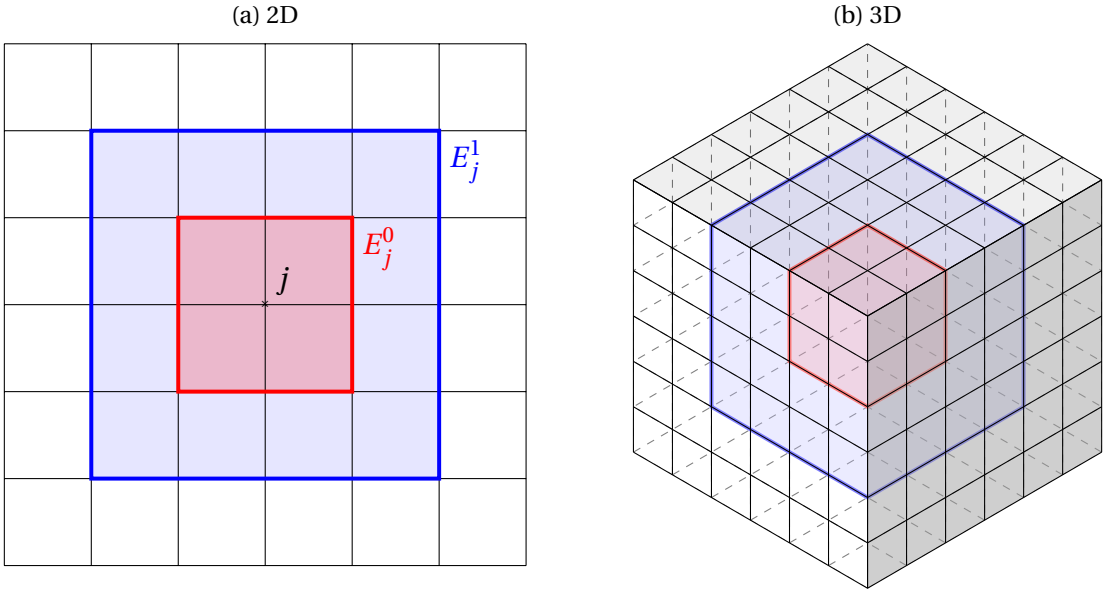


Figure 2.4. Representation of the sets E_j^0 and E_j^1 for a Cartesian grid.

2.3.3. Comparison between the two approaches

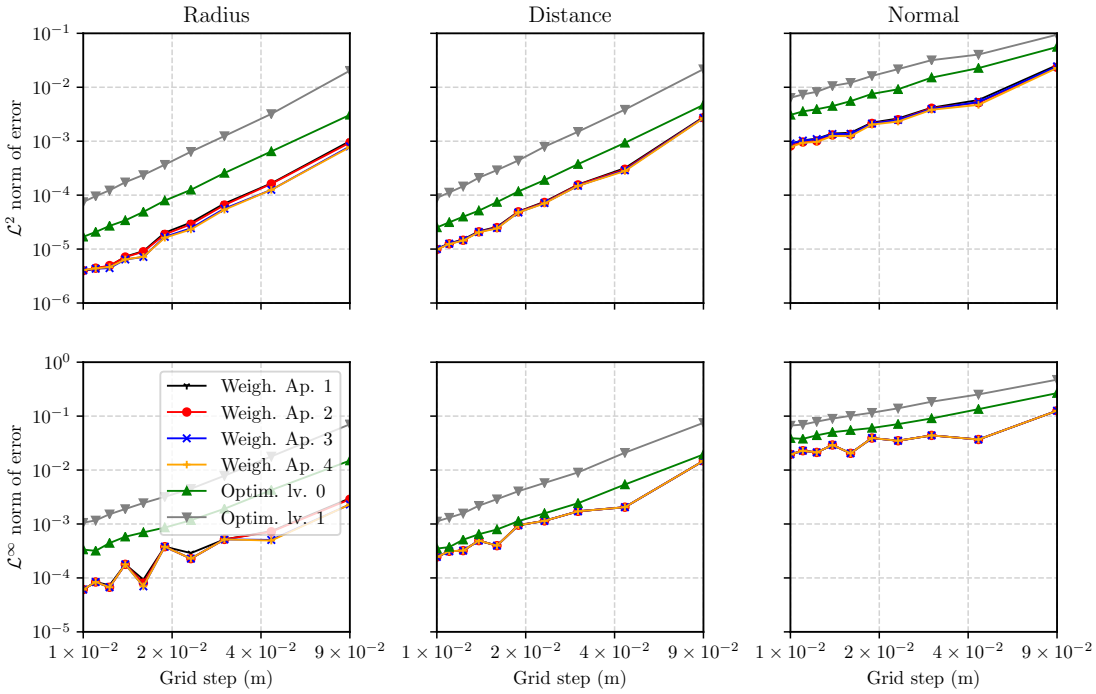


Figure 2.5. Intersection of an Eulerian volume mesh and a cylindrical Lagrangian surface mesh: evolution of different error indicators with respect to the space grid of the Eulerian meshes (250 Lagrangian facets) in log/log scale.

Table 2.1. Values of convergence order, with respect to the Eulerian grid step, of the different indicators computed in the case of the cylinder.

Method	Radius		Distance		Normal	
	$\mathscr{L}^2$	$\mathscr{L}^\infty$	$\mathscr{L}^2$	$\mathscr{L}^\infty$	$\mathscr{L}^2$	$\mathscr{L}^\infty$
Weighting 1	2.54	1.67	2.44	1.70	1.42	0.70
Weighting 2	2.51	1.67	2.44	1.70	1.42	0.70
Weighting 3	2.43	1.55	2.42	1.70	1.39	0.70
Weighting 4	2.41	1.54	2.42	1.70	1.41	0.70
Optimization 1	2.34	1.70	2.34	1.78	1.30	0.85
Optimization 2	2.47	1.86	2.44	1.87	1.20	0.87

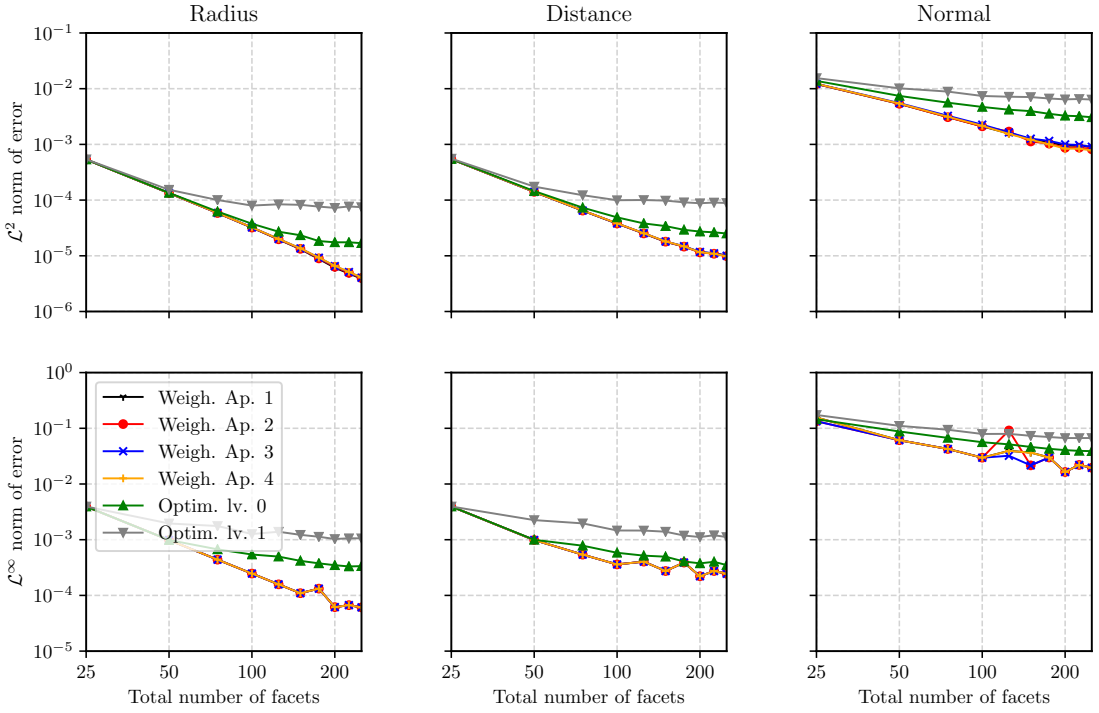


Figure 2.6. Intersection of an Eulerian volume mesh and a cylindrical Lagrangian surface mesh: evolution of different error indicators with respect to the number of facets of the Lagrangian mesh (151 Eulerian elements) in log/log scale.

Table 2.2. Convergence orders, with respect to the Lagrangian grid step, of the different indicators computed in the case of the cylinder.

	Radius		Distance		Normal	
Method	$\mathscr{L}^2$	$\mathscr{L}^\infty$	$\mathscr{L}^2$	$\mathscr{L}^\infty$	$\mathscr{L}^2$	$\mathscr{L}^\infty$
Weighting 1	2.16	1.84	1.77	1.12	1.18	0.83
Weighting 2	2.14	1.84	1.78	1.12	1.23	0.79
Weighting 3	2.13	1.84	1.77	1.12	1.18	0.82
Weighting 4	2.13	1.84	1.77	1.12	1.25	0.83
Optimization 1	1.52	0.99	1.31	0.94	0.63	0.57
Optimization 2	0.76	0.55	0.71	0.53	0.37	0.40

Table 2.3. Computation times, in seconds, of the pre-processor for the intersection between a hexahedral Eulerian mesh and a cylindrical Lagrangian mesh.

Eulerian		Lagrangian		Execution time (s)	
Elements	Grid step	Facets	Grid step	Weighting	Optimization
16	6.25×10^{-1}	250	1.26×10^{-2}	0.7	335.5
31	3.23×10^{-1}	250	1.26×10^{-2}	1.2	467.0
46	2.17×10^{-1}	250	1.26×10^{-2}	1.5	414.0
61	1.64×10^{-1}	250	1.26×10^{-2}	2.3	496.0
76	1.32×10^{-1}	250	1.26×10^{-2}	4.1	394.5
91	1.10×10^{-1}	250	1.26×10^{-2}	5.7	433.1
106	9.43×10^{-2}	250	1.26×10^{-2}	7.9	363.0
121	8.26×10^{-2}	250	1.26×10^{-2}	11.3	391.1
136	7.35×10^{-2}	250	1.26×10^{-2}	13.5	334.7
151	6.62×10^{-2}	250	1.26×10^{-2}	19.1	338.8
151	6.62×10^{-2}	25	1.26×10^{-1}	18.4	30.3
151	6.62×10^{-2}	50	6.28×10^{-2}	18.4	41.7
151	6.62×10^{-2}	75	4.19×10^{-2}	18.2	52.9
151	6.62×10^{-2}	100	3.14×10^{-2}	18.3	75.0
151	6.62×10^{-2}	125	2.51×10^{-2}	21.0	117.1
151	6.62×10^{-2}	150	2.09×10^{-2}	21.0	117.0
151	6.62×10^{-2}	175	1.80×10^{-2}	20.9	213.0
151	6.62×10^{-2}	200	1.57×10^{-2}	18.9	230.0
151	6.62×10^{-2}	225	1.40×10^{-2}	19.0	289.1

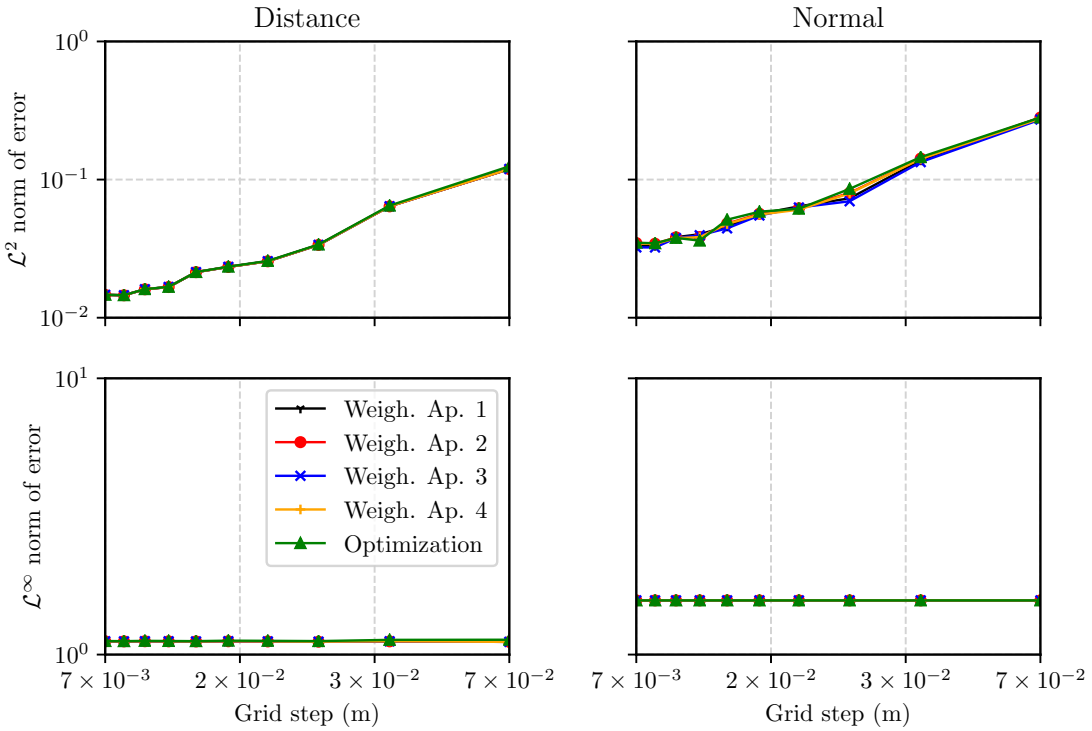


Figure 2.7. Intersection of a hexahedral Eulerian volume mesh and a NACA 0012 airfoil Lagrangian surface mesh: evolution of different error indicators with respect to the space grid of the Eulerian meshes (250 Lagrangian facets) in log/log scale.

Table 2.4. Convergence orders of the different indicators computed in the case of a NACA0012 airfoil considering only $\mathscr{L}^2$ norm.

	Distance		Normal	
Method	Eulerian	Lagrangian	Eulerian	Lagrangian
Weighting 1	0.95	0.24	0.95	0.21
Weighting 2	0.95	0.24	0.95	0.19
Weighting 3	0.95	0.24	0.94	0.21
Weighting 4	0.95	0.24	0.94	0.19
Optimization 1	0.97	0.24	0.96	0.17
Optimization 2	0.97	0.24	0.96	0.17

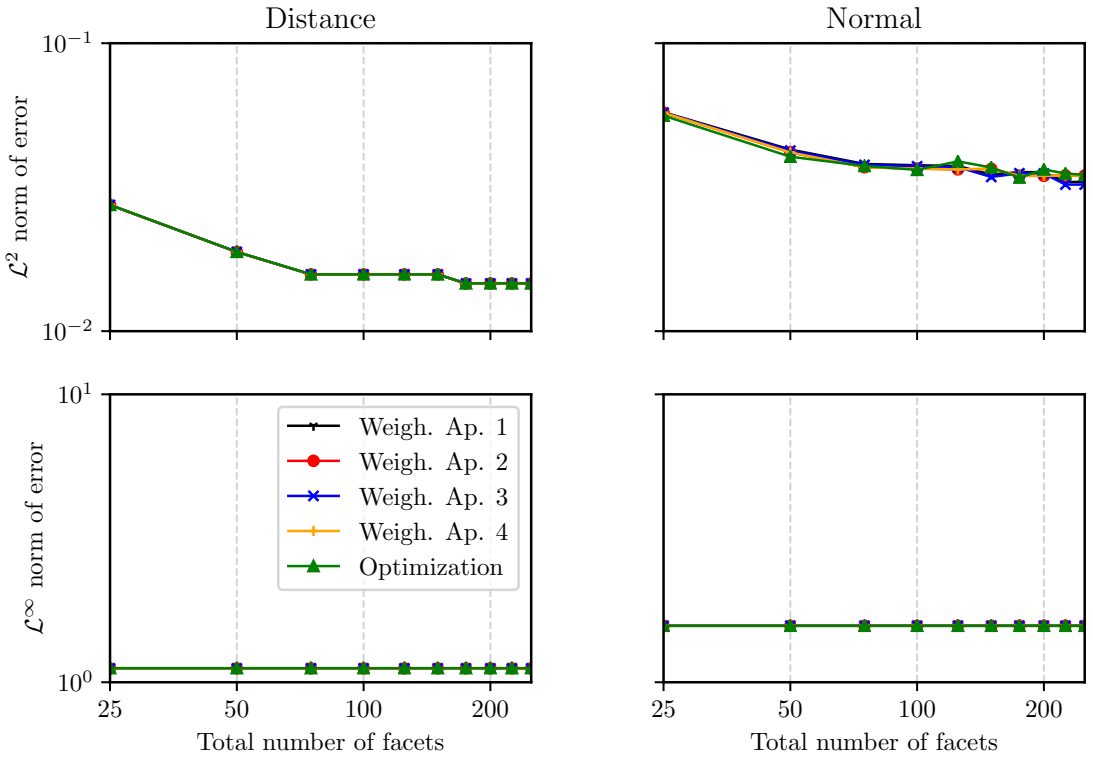


Figure 2.8. Intersection of a hexahedral Eulerian volume mesh and a NACA 0012 airfoil Lagrangian surface mesh: evolution of different error indicators with respect to the number of facets of the Lagrangian mesh (151 Eulerian elements) in log/log scale.

First, the different geometrical methods are compared using a simple 2D case with regular boundary: the circular cylinder. We consider a square domain of side 1.5 with a circular cylinder of radius $r = 0.5$ and center $\mathbf{c} = (0,0)$. We compute some error indicators and observe their evolution when we refine the square hexahedral volume

Table 2.5. Computation times, in seconds, of the pre-processor for the intersection between a hexahedral Eulerian mesh and a NACA 0012 airfoil Lagrangian surface mesh.

Elements	Facets	Weighting (s)	Optimization (s)
21	250	0.6	157.7
41	250	1.0	361.8
61	250	1.5	311.2
81	250	2.6	211.8
101	250	4.2	173.4
121	250	6.5	147.7
141	250	9.2	179.4
161	250	15.2	153.5
181	250	19.8	159.6
201	250	19.1	140.1
201	25	22.2	53.0
201	50	22.2	85.4
201	75	22.5	111.9
201	100	23.3	115.7
201	125	23.5	137.8
201	150	23.5	138.1
201	175	23.3	138.3
201	200	23.4	135.4
201	225	23.9	152.4

mesh or the cylindrical surface mesh (*i.e.* increase the number of facets) in order to compare accuracy and convergence rate. We use three different error indicators:

1. The radius related error, which indicates how close to the cylinder is the approximated projection:

$$\forall j \in J, \quad e_j^r = \left| r_j^p - r \right| \quad \text{with} \quad r_j^p = \left| \mathbf{x}_j^p - \mathbf{c} \right| \tag{2.17}$$

2. The distance related error, which indicates how close from the exact projection $\mathbf{x}_j^e$ is the approximated projection.

$$\forall j \in J, \quad e_j^d = \left| \mathbf{x}_j^p - \mathbf{x}_j^e \right| \tag{2.18}$$

3. The normal vector related error which corresponds to the angle between the approximated ($\mathbf{n}_j$) and exact ($\mathbf{n}_j^e$)) normal vectors:

$$\forall j \in J, \quad e_j^n = \left| \cos^{-1} \left| \mathbf{n}_j \cdot \mathbf{n}_j^e \right| \right| \quad \text{with} \quad \mathbf{n}_j = \frac{\mathbf{x}_j - \mathbf{x}_j^p}{\left| \mathbf{x}_j - \mathbf{x}_j^p \right|} \tag{2.19}$$

and, for each of them, we compute an approximated $\mathscr{L}^2$ and $\mathscr{L}^\infty$ norms as follows:

$$\left| e^w \right|_2^h := \left(\frac{1}{\text{card}(J)} \sum_{j \in J} \left(e_j^w \right)^2 \right)^{\frac{1}{2}} \quad \text{and} \quad \left| e^w \right|_\infty^h := \max_{j \in J} \left| e_j^w \right| \tag{2.20}$$

with $w \in \{r, d, n\}$. Figure 2.5 (resp. Figure 2.6) shows the evolution of those error indicators when the volume (resp. surface) mesh is refined. Table 2.1 (resp. 2.2) gives the convergence order, with respect to the Eulerian (resp. Lagrangian) grid step, computed for all the different approaches. As a whole, when considering convergence with respect to the Eulerian grid step, all approaches seem to provide a roughly quadratic rate for the radius and the distance related errors and linear for the normal related error, in both $\mathscr{L}^2$ and $\mathscr{L}^\infty$ norms. Concerning convergence with respect to the Lagrangian, the conclusion for the weighting approach remains the same (except for the convergence of the distance indicator in $\mathscr{L}^\infty$ norm which becomes roughly linear) but optimizations approaches systematically provide lower convergence rates.

It can also be noted that the optimization "level 1" systematically provides the largest error – probably because it takes into account data coming farther away from the considered point, making the approach formally non-local. On the other hand, when considering the weighting approach, the way of defining α_e^j (*cf.* section 2.3.1) seems to have little impact on the results except for the $\mathscr{L}^\infty$ norm of the normal vector related error when refining the surface mesh. Indeed, when $\alpha_e^j = 1$ or $\alpha_e^j = \mathscr{A}_e$ (*i.e.*Weigh. Ap. 1 and 2), we can see a jump at $N_S = 125$ which is soften if an invert

distance weighting is used (*i.e.* Weigh. Ap. 3 and 4). From Figure 2.8, we notice that a plateau minimal error is reached (or almost reached) for the finest Lagrangian space discretization. The level of this minimal error is driven by the Eulerian space step. It is not the case for the Eulerian space-step convergence study of Figure 2.7, where the minimal error driven by the Lagrangian space step is not reached. This could be explained by the fact that the minimum space step for the Lagrangian mesh (1.26×10^{-2}) is sligthly lower than the one of the Eulerian mesh (6.62×10^{-2}) – factor 5, *cf.* Table 2.3). Finally, if we compare the weighting and optimization approaches, we can see that the optimization provides slightly larger errors but also a steadier convergence (less jagged), especially in $\mathscr{L}^\infty$ norm.

At this point, it is not clear which approach best fits our problem; another comparison element is needed. Therefore, we monitored the execution time of the intersection pipeline – other CPU tasks can pollute execution time but the accuracy of the results is about 0.1 s, which is clearly sufficient for the comparison. The results are gathered in Table 2.3 (all different weighting approaches provide similar execution times and we consider only optimization "level 0"). The first obvious assessment is that the weighting approach is greatly faster – this is a key finding because the application of the proposed modeling is nuclear component design and shape optimization. Another interesting finding is that the execution time seems to be more dependent on the number of volume elements (*i.e.* grid step) when considering the weighting approach but more dependent on the number of facets when considering the optimization approach (in fact it depends on the number of collected facets per node). These tendencies were expected. On one side, the weighting approach is rather local and the number of operations is, to some extent, constant for one given volume. Therefore, the computation cost mainly depends on the volume number. On the other side, the optimization approach is global and the number of operations, for a given node, strongly depends on the collected-facet number in the adjacent volumes. Therefore, when decreasing the volume space step, the number of nodes to project is increasing, but the number of Uzawa operations is decreasing, leading to the observed behavior.

To check those assessments, we also considered a more complex geometry: the NACA0012 airfoil. The same error indicators are considered (except the radius one for obvious reasons) and their evolution with respect to the grid step (resp. number of facets) is show in Figure 2.7 (resp. Figure 2.8). In that case, we can see that all approaches provide equivalent error values and equivalent convergence orders (*cf.* Table 2.4). Moreover, convergence, for all approaches, is roughly linear with respect to the Eulerian grid step and really degraded when considering the Lagrangian mesh. Thus, all approaches fail to handle the trailing edge ($\mathscr{L}^\infty$ norm of error close to 1 and located around the trailing edge). However, when we look at the execution times gathered in Table 2.5 we can assess that the weighting approach is still largely faster for the same accuracy in the results.

3. Numerical methods and Penalized Direct Forcing

Summary

3.1. Original problem

3.1.1. Governing equations

3.1.1.1. Homogeneous Equilibrium Model

As mentioned in the introduction, the aimed application induces the modeling of compressible turbulent two-phase flow w ith thermodynamic and mechanical disequilibrium at low Mach number (*i.e.* the break is considered far enough from the flow limiter for the flow to remain subsonic). Keeping in mind the topological shape optimization goal, a Homogeneous Equilibrium Model (HEM) [Cle00], with disequilibrium closure laws [GOch], is considered to preserve low computational times (only three balance equations: mass, momentum and internal energy) while taking into account two-phase aspects (two-phase flow illustration can be found in [Bel18]). The assumptions of the HEM are the following:

i The same pressure is considered for the two phases,

ii An eddy viscosity model is used to take turbulence into account,

iii Surface tension and turbulent dissipation are neglected,

iv We neglect wave effect and eliminate the time term in the mass balance equation ($\partial_t \rho \equiv 0$) .

the Equations Of State of the mixture fluid are written as follow:

$$\mathfrak{x} = \frac{H - H_L^{SAT}}{\mathfrak{L}} \tag{3.1}$$

$$\mathfrak{v} = \frac{\mathfrak{x}\rho_L}{\mathfrak{x}\rho_L + (1-\mathfrak{x})\rho_G} \tag{3.2}$$

$$\rho = \mathfrak{v}\rho_G + (1-\mathfrak{v})\rho_L \tag{3.3}$$

with:

- $\mathfrak{x}$: the thermodynamic title,
- $\mathfrak{L}$: latent heat at saturation,
- H: the enthalpy of the mixture fluid,
- H_{SAT}^L: enthalpy at saturation of the liquid,
- $\mathfrak{v}$: the void fraction,
- ρ (resp. ρ_L and ρ_G): the density of the mixture fluid (resp. liquid and gas phase).

The thermodynamic properties of each phase are given by tabulated EOS (polynomial interpolation between experimental values). Then many physicalclosure models (semi-empirical correlations) are used [GOch] , notably for the gas and liquid velocities (resp. $\mathbf{u}_G$ and $\mathbf{u}_L$) in the kinetic disequilibrium:

$$\mathbf{u} = \mathfrak{x}\mathbf{u}_G + (1-\mathfrak{x})\mathbf{u}_L \tag{3.4}$$

where $\mathbf{u}$ is the mixture velocity. In terms of turbulence modeling, a scalar Schlichting's model [Sch68], described in Section 3.1.1.3 is considered – a second PhD project, launched in October 2020, focuses especially on enhancing the turbulence modeling.

In this document, as only fluid dynamic cases are treated (*cf.* Section 5), the internal energy balance will not be considered (not solved). In addition, as all cases are monophasic, the momentum balance and mass balance equations of the HEM boil down to the dilatable Navier-Stokes equations (*cf.* the next section). Without loss of generality and for purpose of clarity, we will describe our method using this set of equations.

3.1.1.2. Dilatable Navier-Stokes equations

Given $\Omega \subseteq \mathbb{R}^d$ a d-dimensional open compact domain with a piecewise regular boundary denoted $\partial\Omega$ such as $\overline{\Omega} = \Omega \cup \partial\Omega$, given some Boundary Conditions (BC) on $\partial\Omega$ an Initial Conditions (IC) on Ω, the governing system of equation, written in conservative form, is:

$$\begin{cases} \partial_t(\rho\mathbf{u}) + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u} - \bar{\bar{\sigma}} + \bar{\bar{I}}p) = \mathbf{s} & \text{on } \Omega \\ \nabla\cdot(\rho\mathbf{u}) = 0 & \text{on } \Omega \\ +\text{BC} & \text{on } \partial\Omega \\ +\text{IC} & \text{on } \Omega \end{cases} \tag{3.5}$$

with $\bar{\bar{I}} \in \mathbb{R}^{d\times d}$ the identity matrix, $\rho : \Omega \times \mathbb{R}^+ \to \mathbb{R}$ the fluid density given by tabulated EOS of water (*cf.* Section 3.1.1.3), $p : \Omega \times \mathbb{R}^+ \to \mathbb{R}$ the pressure, $\mathbf{u} : \Omega \times \mathbb{R}^+ \to \mathbb{R}^d$ the fluid velocity, $\bar{\bar{\sigma}} : \Omega \times \mathbb{R}^+ \to \mathbb{R}^{d\times d}$ the viscous stress tensor and $\mathbf{s}$ the source terms coming from physical modeling (for instance gravity, interface friction, *etc.*).

3.1.1.3. Viscous stress tensor

In this paper, we assume that the viscous stress $\bar{\bar{\sigma}}$ only depends on the dynamic viscosity μ and on the strain rate tensor $\bar{\bar{\varepsilon}}$, which is defined as follows:

$$\bar{\bar{\varepsilon}} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right) \tag{3.6}$$

Hence:

$$\bar{\bar{\sigma}} = 2\mu\bar{\bar{\varepsilon}} \tag{3.7}$$

with $\mu = \mu_f + \mu_T$ and:

- μ_f the equivalent mixture dynamic viscosity that can be given by the tabulated EOS of water – in practice, those EOS use the values of the fluid pressure and enthalpy at the previous time step to interpolate the fluid dynamic viscosity and density between experimental points – or considered constant.
- μ_T the eddy viscosity given by a Schlichting model [Sch68]:

$$\mu_T = \begin{cases} 0 & \text{if the flow is laminar} \\ a_S\rho|\mathbf{u}|L_T & \text{if the flow is turbulent} \end{cases} \tag{3.8}$$

  where $a_S \in \mathbb{R}$ is a parameter (its value is determined empirically) and $L_T \in \mathbb{R}$ a characteristic turbulence length.

3.1.2. Time discretization

The adapted semi-implicit time scheme and the fractional-step algorithm are conjointly presented here.

3.1.2.1. Time scheme

Let us denote $\delta t \in \mathbb{R}^{+*}$ the time step and $N_T \in \mathbb{N}$ the number of time steps. The sequence associated to the discrete time steps $(t_n)_{n<N_T+1}$ is defined such as:

$$\forall n \in [\![0, N_T]\!], \quad t_n = n\delta t \tag{3.9}$$

Then, we define $\mathbf{u}^n : \Omega \to \mathbb{R}^d$, $p^n : \Omega \to \mathbb{R}$, $\rho^n : \Omega \to \mathbb{R}$ as approximations of, respectively, the velocity, the pressure and the density at time t_n. The viscous stress tensor at time t_n is approximated as follows:

$$\bar{\bar{\sigma}}^n = \mu^n \left(\nabla \mathbf{u}^n + \left(\nabla \mathbf{u}^n \right)^T \right) \tag{3.10}$$

with $\mu^n = \mu_f^n + \mu_T^n$ where:

- μ_f^n is computed with the tabulated EOS of water using the values of pressure and enthalpy at time t_{n-1},
- μ_T^n is given by:

$$\mu_T^n = \begin{cases} 0 & \text{if the flow is laminar} \\ a_S \rho |\mathbf{u}^{n-1}| L_T & \text{if the flow is turbulent} \end{cases} \tag{3.11}$$

In order to enhance the readability of this document, the inertia and viscous terms are gathered together by introducing the notations:

$$\begin{cases} \bar{\bar{\Xi}}^{n,n+1} = \rho^n \mathbf{u}^n \otimes \mathbf{u}^{n+1} - \bar{\bar{\sigma}}^{n+1} \\ \bar{\bar{\Sigma}}^{n,n+1} = \bar{\bar{\Xi}}^{n,n+1} + \bar{\bar{I}} p^n \end{cases} \tag{3.12}$$

Considering a semi-implicit scheme with time-implicit diffusive and (linearized) advective terms, the semi-discrete system can be written as follows:

$$\begin{cases} \delta t^{-1} \rho^n (\mathbf{u}^{n+1} - \mathbf{u}^n) + \nabla \cdot \bar{\bar{\Xi}}^{n,n+1} + \nabla p^{n+1} = \mathbf{s}^n & \text{on } \Omega \\ \nabla \cdot (\rho^n \mathbf{u}^{n+1}) = 0 & \text{on } \Omega \\ +\text{BC} & \text{on } \partial\Omega \\ +\text{IC} & \text{on } \Omega \end{cases} \tag{3.13}$$

In this system of equations, we have assumed that the physical quantities as the density, the dynamic viscosity, *etc.* are computed with EOS involving the main variables $\mathbf{u}$ and p at the previous time step and are kept constant during the current time step.

Remark 2 *The time derivative of the momentum is not written in a conservative way in system (3.13). Indeed, $\rho^{n+1}\mathbf{u}^{n+1}$ should appear but, as the compressibility is assumed to be very low, the time variation of the density is supposed negligible during a time step.*

3.1.2.2. Fractional-step algorithm

Projection schemes [BCM01] are often used to solve the incompressible Navier-Stokes equations and are based on the Helmholtz-Hodge theorem. Here, we present its extension to dilatable fluid ($\nabla \cdot \rho\mathbf{u} = 0$). The idea is to split the momentum balance equation into two steps:

$$\frac{\rho^n}{\delta t}(\mathbf{u}^* - \mathbf{u}^n) + \nabla \cdot \bar{\bar{\Sigma}}^{n,*} = \mathbf{s}^n \tag{3.14}$$

$$\frac{\rho^n}{\delta t}\left(\mathbf{u}^{n+1} - \mathbf{u}^*\right) + \nabla \phi^{n+1} = \mathbf{0}_{\mathbb{R}^d} \tag{3.15}$$

with $\mathbf{u}^*$ a provisional variable called the predicted velocity, $\bar{\bar{\sigma}}^* = \mu^n\left(\nabla\mathbf{u}^* + (\nabla\mathbf{u}^*)^T\right)$ the predicted viscous stress tensor and $\phi^{n+1} = p^{n+1} - p^n$ the pressure corrector. Then, the fractional-step algorithm reads as follows:

1. *Prediction*: only time, inertia, viscous and source terms are considered, the pressure term is kept at the previous time step. An intermediate velocity, called predicted and denoted $\mathbf{u}^*$ is computed.

2. *Projection*: the pressure corrector field is computed using the predicted velocity and mass balance equation.

3. *Correction*: the velocity is computed using the pressure corrector gradient.

When taking the divergence of equation (3.15), the mass balance equation appears and can be substituted, heading to the projection (*cf.* equation (3.17)). Finally, the following three steps algorithm is obtained:

$$\frac{\rho^n}{\delta t}\mathbf{u}^* + \nabla \cdot \bar{\bar{\Sigma}}^{n,*} = \mathbf{s}^n + \frac{\rho^n}{\delta t}\mathbf{u}^n \tag{3.16}$$

$$\Delta\phi^{n+1} = \frac{1}{\delta t}\nabla \cdot \left(\rho^n \mathbf{u}^*\right) \tag{3.17}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\delta t}{\rho^n}\nabla\phi^{n+1} \quad \text{and} \quad p^{n+1} = p^n + \phi^{n+1} \tag{3.18}$$

Where equations (3.16), (3.17) and (3.18) respectively correspond to the prediction, projection and correction steps. Note that writing those equations implies that the fluid is dilatable and p^n is at least of class C^2 (*i.e.* p is of class C^2 in space).

3.1.3. Space discretization

3.1.3.1. Weak formulation

Let $\mathcal{V}(\Omega) \subseteq (\mathcal{L}^2(\Omega))^d$ and $\mathcal{W}(\Omega) \subseteq \mathcal{L}^2(\Omega)$ be two vector space of functions supposed sufficiently regular. In the fractional-step algorithm, the equations are solved sequentially (prediction, projection and correction). Thus, we obtain the three weak forms, one for each equation.
Prediction:
Find $\mathbf{u}^* \in \mathcal{V}(\Omega)$ such as, for all $\mathbf{v} \in \mathcal{V}(\Omega)$:

$$\int_\Omega \left(\frac{\rho^n}{\delta t}(\mathbf{u}^* - \mathbf{u}^n) + \nabla \cdot \bar{\bar{\Sigma}}^{n,*} \right) \circ \mathbf{v} = \int_\Omega \mathbf{s}^n \circ \mathbf{v} \tag{3.19}$$

Projection (for dilatable fluid):
Find $\phi^{n+1} \in \mathcal{W}(\Omega)$ such as, for all $q \in \mathcal{W}(\Omega)$:

$$\int_\Omega \Delta\phi^{n+1}\, q = \frac{1}{\delta t} \int_\Omega \nabla \cdot \left(\rho^n \mathbf{u}^* \right) q \tag{3.20}$$

Correction:
Find $\mathbf{u}^{n+1} \in \mathcal{V}(\Omega)$ such as, for all $\mathbf{v} \in \mathcal{V}(\Omega)$:

$$\frac{1}{\delta t} \int_\Omega \rho^n \mathbf{u}^{n+1} \circ \mathbf{v} = \frac{1}{\delta t} \int_\Omega \left(\rho^n \mathbf{u}^* - \nabla\phi^{n+1} \right) \circ \mathbf{v} \tag{3.21}$$

where $\circ$ denotes the Hadamard product. The next step is to use the integration by parts formula in the weak forms of the prediction and projection to highlight boundary terms.
Prediction:
Find $\mathbf{u}^* \in \mathcal{V}(\Omega)$ such as, for all $\mathbf{v} \in \mathcal{V}(\Omega)$:

$$\int_\Omega \frac{\rho^n}{\delta t}(\mathbf{u}^* - \mathbf{u}^n) \circ \mathbf{v} - \int_\Omega \mathrm{diag}(\bar{\bar{\Sigma}}^{n,*} \nabla \mathbf{v}^T) = \int_\Omega \mathbf{s}^n \circ \mathbf{v} - \int_{\partial\Omega} \bar{\bar{\Sigma}}^{n,*} \mathbf{n}_{\partial\Omega} \circ \mathbf{v} \tag{3.22}$$

Projection:
Find $\phi^{n+1} \in \mathcal{W}(\Omega)$ such as, for all $q \in \mathcal{W}(\Omega)$:

$$-\int_\Omega \nabla\phi^{n+1} \cdot \nabla q = -\frac{1}{\delta t} \int_\Omega \rho^n \mathbf{u}^* \cdot \nabla q - \int_{\partial\Omega} q \left(\nabla\phi^{n+1} - \frac{\rho^n \mathbf{u}^*}{\delta t} \right) \cdot \mathbf{n}_{\partial\Omega} \tag{3.23}$$

From here, solving the correction equation is straightforward. For the proof of existence and uniqueness in the case of the Navier-Stokes equations, the interested reader can refer to [GR86]

3.1.3.2. Finite Element Method

As already mentioned, the PDF method was initially developed in the finite-volume framework using a finite-difference MAC scheme. Here, we present a Galerkin (*i.e.* the

weak formulation test functions and the decompositions of variables use the same discrete basis) Finite Element Method (FEM) version that has been implemented in a preexisting CEA application. As the location of the vector and scalar unknowns is completely different from the MAC scheme, new properties appear for the PDF projection equation (*cf.* Remark 11). In this section, the used FEM formulation is briefly reminded. For general information about solving the Navier-Stokes equations with the FEM, one can refer to [GR86].

The computation domain Ω is divided in $N_E \in \mathbb{N}$ hexahedral elements, each denoted K_e with $e \in [\![1, N_E]\!]$ and Ω_e the portion of Ω associated to the element K_e (*cf.* Figure 2.1). Those elements are composed of nodes and the total number of nodes is denoted N_N. The mixed FEM is used, which means that the discrete unknowns of the problem are decomposed in two different FE basis. For the velocity, a $\mathbb{Q}_1$ basis (*i.e.* trilinear decomposition at nodes) is used while, for pressure, a $\mathbb{Q}_0$ basis (*i.e.* the discrete pressure field is constant by element) is used. This pair of elements is known to be unstable and can induce checkerboard pattern for the pressure. Indeed, the pressure is defined to within an element of the kernel of the gradient operator [GR86]. When the gradient operator is well defined, its kernel is only composed of constant functions but, in this $\mathbb{Q}_0$–$\mathbb{Q}_1$ discretization, it also contains some checkerboard-like functions. Nevertheless, this pair is kept for two main reasons: i) it counts very few degrees of freedom per element – 8 for velocity and 1 for pressure compared to 27 and 8 when using a $\mathbb{Q}_2$–$\mathbb{Q}_1$ pair which, for instance, is stable – so it is quite fast; ii) the pressure instabilities are soften when the diffusivity is high enough – *i.e.* in laminar cases or when the turbulent viscosity is high, which is a case of interest for safety passive system design. Moreover, let us notice that the general idea of the PDF method (*cf.* Section 1.4.2.6) does not depend strongly on the space discretization. If needed, it could be adapted to another pair of elements with little effort.

In addition, let us define:

- $n_e = \text{card}(J_e)$ the number of nodes belonging to the element K_e ($e \in [\![1, N_E]\!]$),
- ρ_e^n: the value of the density in element K_e (computed using the tabulated EOS of water),
- u_{ae}: the value of the velocity along the direction x_a approximated at the centroid of element K_e at previous time step,
- p_e^n: the value of the pressure in element K_e,
- $\phi_e^n = p_e^n - p_e^{n-1}$: the value of the pressure corrector in element K_e,
- $\mu_e^n = \frac{\nu_e^n}{\rho_e^n}$: the value of the dynamic viscosity in element K_e,
- φ_i: the basis function associated to the node $i \in [\![1, N_N]\!]$,
- λ^n: the coordinates of $\mathbf{u}^n$ in the $\mathbb{Q}_1$ basis such as $\mathbf{u}^n \approx \sum_{i=1}^{N_N} \lambda^n \varphi_i$ ($N_n d$ values),

- λ_e^n (resp. s_e^n): the coordinates of $\mathbf{u}^n$ (resp. $\mathbf{s}^n$) in the $\mathbb{Q}_1$ FE basis restricted to element K_e ($m_e = n_e\, d$ values).

where n in superscript denotes values at time step n. Using the variables FE basis decomposition and the weak formulation of the problem (presented in Section 3.1.3.1), the elemental matrices, for an element number e, are constructed as follows:
Mass matrix:

$$\mathbf{M}_e \rho_e^n = \left(M_{ij}^e\right) \rho_e^n \quad \text{with} \quad \forall (i,j) \in (J_e)^2, \quad M_{ij}^e = \int_{\Omega_e} \varphi_i \varphi_j \tag{3.24}$$

Gradient-divergence matrix:

$$\mathbf{B}_e = \left(B_{ai}^e\right) \quad \text{with} \quad \forall (a,i) \in [\![1,d]\!] \times J_e, \quad B_{ai}^e = \int_{\Omega_e} \partial_{x_a} \varphi_i \tag{3.25}$$

Advective matrix:

$$\mathbf{N}_e \rho_e^n = \left(N_{ij}^e\right) \rho_e^n \text{ with } \forall (i,j) \in (J_e)^2,\ N_{ij}^e = \sum_{a=1}^{d} \left(u_{ae} \int_{\Omega_e} (\partial_{x_a} \varphi_j) \varphi_i \right) \tag{3.26}$$

Diffusive matrix:

$$\mathbf{D}_e = \left(D_{ij}^e\right) \text{ with } \forall (i,j) \in (J_e)^2,\ D_{ij}^e = \sum_{a=1}^{d} \left(\int_{\Omega_e} (\partial_{x_a} \varphi_j)(\partial_{x_a} \varphi_i) \right) \tag{3.27}$$

Here we consider the lumping of the mass matrix, leading to a diagonal matrix that can be more-easily inverted. It means that the mass matrix can be rewritten as:

$$\forall\, i \in (J_e)^2,\ M_{ii}^e \approx \int_{\Omega_e} \varphi_i. \tag{3.28}$$

Obviously, this involves a loss of information. However, in case of steady-state computations, the dynamic behavior is not of interest, neither the form of the mass matrix. And, in case of transient computations, R. L. T. BEVAN *et al.* have shown that the impact of the lumping process on the results is moderate [BN12].

***Remark* 3** *The idea to lump the implicit part of the contribution of the forcing term to the mass matrix is motivated by the presence of the penalty parameter η in the PDF forcing terms. Indeed, the factor η^{-1} can badly degrade the mass matrix conditioning as $\eta \ll 1$. Thus, lumping the contribution of the forcing term in conjunction with the use of a diagonal preconditioner could greatly enhance the robustness of the method without degrading the results too much (cf. Chapter 5).*

By taking the weak forms of equations (3.14) and (3.15) and integrating by parts the diffusive and pressure terms, we obtain the following time and space discrete elemental resolution algorithm for the projection scheme, written for an element K_e:

$$\left(\frac{1}{\delta t}\mathbf{M}_e\rho_e^n + \mathbf{D}_e\mu_e^n + \mathbf{N}_e\rho_e^n\right)\lambda_e^* = \frac{1}{\delta t}\mathbf{M}_e\rho_e^n\lambda_e^n + \mathbf{B}_e p_e^n + \mathbf{M}_e s_e^n \tag{3.29}$$

$$\mathbf{B}_e^T\mathbf{M}_e^{-1}\mathbf{B}_e\phi_e^{n+1} = -\frac{1}{\delta t}\mathbf{B}_e^T\rho_e^n\lambda_e^* \tag{3.30}$$

$$\lambda_e^{n+1} = \lambda_e^* + \delta t\mathbf{M}_e^{-1}\frac{1}{\rho_e^n}\mathbf{B}_e\phi_e^{n+1} \tag{3.31}$$

Then, each elemental system is assembled in a typical FE fashion to obtain the global system representative of the whole domain.

***Remark* 4** *M_e is invertible and that computing its inverse is trivial as it is a diagonal matrix due to the lumping technique.*

***Remark* 5** *The matrix related to the Laplace operator, in the projection equation (3.30), is not directly constructed from the weak form. Indeed, the discrete divergence matrix $\mathbf{B}_e^T$ is applied at a discrete level to the correction equation (3.31), considering the fact that the discrete mass equation stands as:*

$$\mathbf{B}_e^T\rho_e^n\lambda_e^{n+1} = 0 \tag{3.32}$$

***Remark* 6** *Also, a Balancing Tensor Diffusivity (BTD) correction is sometimes used. It consists in adding extra diffusion in the streamline direction, with a diffusivity depending on a given parameter and the time step, in order to stabilize the convection.*

3.2. Immersed Boundary

3.2.1. Penalized Direct Forcing

To preserve an implicit forcing term, M. Belliard *et al.* developed a new technique, called Penalized Direct Forcing (PDF) [BF10], inspired by the Direct Forcing methods and penalization techniques [Ang99; ABF99; Sar09; Sar+08]. The PDF method has been applied to the incompressible Navier-Stokes equations in the framework of the finite volume method using a finite difference scheme [IBF14]. In this case, dividing the Navier-Stokes equation terms by ρ, the forcing term is written as follows:

$$\mathbf{f}^{n+1} = \frac{\chi}{\eta\delta t}\left(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^{n+1}\right) \tag{3.33}$$

with $\eta \in \mathbb{R}^{+*}$, such as $\eta \ll 1$. This formulation provides several advantages:

- The forcing term is implicit due to the penalization.
- The forcing term has a suitable mathematical formulation that allows an easy adaptation to the projection of fractional-step schemes due to a formulation similar to a discrete time derivative (*cf.* Section 3.2.2).

In the context of the compressible Navier-Stokes equations considered in this paper, the forcing term has to be modified in order to preserve its good mathematical properties. It is defined as follows:

$$\mathbf{f}^{n+1} := \frac{\chi}{\eta \delta t} \rho^n \left(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^{n+1} \right) \tag{3.34}$$

Remark 7 *The PDF forcing term of equation 3.34 can be viewed as a penalty term with an "effective penalty parameter" η', defined such as $\eta' = \frac{\eta \delta t}{\rho^n}$, depending on both space and time.*

3.2.2. Adaptation to the projection scheme

When using a projection scheme together with IBM, one have shown that the forcing term should be split between the prediction and correction equations in order to preserve the accuracy of the method [IK07; Dom08; GH10]. Inspired by the work of M. Belliard and C. Fournier [BF10], the following splitting is defined ($\mathbf{f}_P^{n+1} + \mathbf{f}_C^{n+1} = \mathbf{f}^{n+1}$):

$$\mathbf{f}_P^{n+1} := \frac{\chi}{\eta \delta t} \rho^n \left(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^* \right) \tag{3.35}$$

$$\mathbf{f}_C^{n+1} := \frac{\chi}{\eta \delta t} \rho^n \left(\mathbf{u}^* - \mathbf{u}^{n+1} \right) \tag{3.36}$$

where $\mathbf{f}_P^{n+1}$ and $\mathbf{f}_C^{n+1}$ are the parts of the forcing term respectively added during the prediction and correction steps.

Those two forcing terms (*cf.* equations (3.35) and (3.36)) are respectively added in equations (3.14) and (3.15). Then, the fractional-step scheme becomes:

$$\frac{\rho^n}{\delta t} (\mathbf{u}^* - \mathbf{u}^n) + \nabla \cdot \bar{\bar{\Sigma}}^{n,*} = \mathbf{s}^n + \frac{\chi}{\eta \delta t} \rho^n \left(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^* \right) \tag{3.37}$$

$$\frac{\rho^n}{\delta t} \left(\mathbf{u}^{n+1} - \mathbf{u}^* \right) + \nabla \phi^{n+1} = \frac{\chi}{\eta \delta t} \rho^n \left(\mathbf{u}^* - \mathbf{u}^{n+1} \right) \tag{3.38}$$

and, by gathering terms together in the correction equation (3.38), we obtain:

$$\frac{\rho^n}{\delta t} \mathbf{u}^{n+1} + \frac{\eta}{\eta + \chi} \nabla \phi^{n+1} = \frac{\rho^n}{\delta t} \mathbf{u}^* \tag{3.39}$$

However, the characteristic function χ is discontinuous among the immersed boundary Γ. Therefore, the divergence cannot be directly applied to (3.39) to get the projection equation. There are two ways of dealing with this issue:

- To consider distributional derivatives and equalities almost everywhere, *i.e.* some kind of weak formulation (*cf.* Section 3.2.3 and Appendix A).

- To apply the divergence at a space-discrete level (what is done in Section 3.2.4).

***Remark* 8** *Beyond the preservation of the accuracy of the fractional-step method, equation (3.39) provides a way to isolate $\nabla\phi^{n+1}$ on one side of the immersed obstacle. It is similar to the standard Neumann condition $\nabla(p^{n+1}-p^n)\cdot\mathbf{n}=0$ on the obstacles, provided that the initial condition verifies it.*

3.2.3. Adaptation of the weak formulation

When we add the forcing terms mentioned in Section 3.2.2, in their respective equations, we obtain the following weak form of the fractional-step algorithm:
Prediction:
Find $\mathbf{u}^*\in\mathcal{V}(\Omega)$ such as, for all $\mathbf{v}\in\mathcal{V}(\Omega)$:

$$\int_\Omega\left(\frac{\rho^n}{\delta t}(\mathbf{u}^*-\mathbf{u}^n)+\nabla\cdot\bar{\bar{\Sigma}}^{n,*}\right)\circ\mathbf{v}=\int_\Omega\left(\mathbf{s}^n+\mathbf{f}_P^{n+1}\right)\circ\mathbf{v} \tag{3.40}$$

Projection (for dilatable fluid):
Knowing that $\eta\left(\eta+\chi\right)^{-1}\nabla\phi^{n+1}$ is weakly differentiable on Ω (sum, division and product of weakly differentiable functions, *cf.* Appendix A for more details), find $\phi^{n+1}\in\mathcal{W}(\Omega)$ such as, for all $q\in\mathcal{W}(\Omega)$:

$$\int_\Omega\left(\nabla\cdot\left(\frac{\eta}{\eta+\chi}\nabla\phi^{n+1}\right)\right)q=\frac{1}{\delta t}\int_\Omega\nabla\cdot\left(\rho^n\mathbf{u}^*\right)q \tag{3.41}$$

Correction:
Find $\mathbf{u}^{n+1}\in\mathcal{V}(\Omega)$ such as, for all $\mathbf{v}\in\mathcal{V}(\Omega)$:

$$\frac{1}{\delta t}\int_\Omega\rho^n\mathbf{u}^{n+1}\circ\mathbf{v}=\frac{1}{\delta t}\int_\Omega\left(\rho^n\mathbf{u}^*-\frac{\eta}{\eta+\chi}\delta t\nabla\phi^{n+1}\right)\circ\mathbf{v} \tag{3.42}$$

Then, the integration by part is used to deal with the discontinuous characteristic function.
Prediction:
Find $\mathbf{u}^*\in\mathcal{V}(\Omega)$ such as, for all $\mathbf{v}\in\mathcal{V}(\Omega)$:

$$\int_\Omega\frac{\rho^n}{\delta t}(\mathbf{u}^*-\mathbf{u}^n)\circ\mathbf{v}-\int_\Omega\mathrm{diag}(\bar{\bar{\Sigma}}^{n,*}\nabla\mathbf{v}^T)=\int_\Omega\left(\mathbf{s}^n+\mathbf{f}_P^{n+1}\right)\circ\mathbf{v}-\int_{\partial\Omega}\bar{\bar{\Sigma}}^{n,*}\mathbf{n}_{\partial\Omega}\circ\mathbf{v} \tag{3.43}$$

Projection:
Find $\phi^{n+1}\in\mathcal{W}(\Omega)$ such as, for all $q\in\mathcal{W}(\Omega)$:

$$-\int_\Omega\frac{\eta}{\eta+\chi}\nabla\phi^{n+1}\cdot\nabla q=-\frac{1}{\delta t}\int_\Omega\rho^n\mathbf{u}^*\cdot\nabla q-\int_{\partial\Omega}\left(\frac{\eta}{\eta+\chi}\nabla\phi^{n+1}-\frac{\rho^n\mathbf{u}^*}{\delta t}\right)q\cdot\mathbf{n}_{\partial\Omega} \tag{3.44}$$

Once again, the correction is straightforward to solve.

3.2.4. Adaptation to the FEM

Concerning the $\mathbb{Q}_1$-$\mathbb{Q}_0$ formulation of the PDF equations (3.37) to (3.39), the first issue to address is the discretization of function χ. This topic is tackled in Section 2.2 but we remind the definition the definition of the discrete characteristic function:

$$\forall e \in [\![1, N_E]\!], \begin{cases} \chi_e = 1 & \text{if } \exists i \in [\![1, N_S]\!] \ / \ S_e^i \neq \emptyset \\ \chi_e = 0 & \text{else} \end{cases} \tag{3.45}$$

and we also introduce a notation to enhance the readability of the document:

$$\forall e \in [\![1, N_E]\!], \ \xi_e = 1 + \frac{\chi_e}{\eta} \tag{3.46}$$

***Remark* 9** *χ_e can be seen as a $\mathbb{Q}_0$ decomposition of the characteristic function χ.*

Then, the discrete prediction (*cf.* equation (3.29)) can be rewritten:

$$\begin{aligned} \left(\frac{1}{\delta t}\mathbf{M}_e \rho_e^n \xi_e + \mathbf{D}_e \mu_e^n + \mathbf{N}_e \rho_e^n\right) \lambda_e^* = & \frac{1}{\delta t}\mathbf{M}_e \rho_e^n \lambda_e^n + \frac{1}{\delta t}\mathbf{M}_e \rho_e^n (\xi_e - 1)\lambda_\Gamma^{n+1} \\ & + \mathbf{B}_e p_e^n + \mathbf{M}_e s_e^n \end{aligned} \tag{3.47}$$

where $\lambda_\Gamma^{n+1} \in \mathbb{R}^{m_e}$ are the components of the discrete imposed velocity on Γ in the $\mathbb{Q}_1$ FE basis (*i.e.* at nodes) with $n_e \in \mathbb{N}$ the number of nodes belonging to the element K_e. At this stage, those components are considered as known values coming directly from the IB condition – what we call a direct assignment.

***Remark* 10** *In the prediction equation, the PDF forcing is split into an implicit contribution (an addition to the mass matrix via the coefficient ξ_e) and a source term.*

The discrete projection equation is obtained the same way. From equation (3.39), we get:

$$\frac{1}{\delta t}\mathbf{M}_e \rho_e^n \left(\lambda_e^{n+1} - \lambda_e^*\right) - \frac{1}{\xi_e}\mathbf{B}_e \phi_e^{n+1} = \mathbf{0}_{\mathbb{R}^{m_e}} \tag{3.48}$$

and, proceeding with this equation as indicated in Remark 5 (left multiplying by $B_e^T M_e^{-1}$ and considering the discrete mass balance), the following projection equation, only depending on the pressure corrector, can be obtained:

$$\mathbf{B}_e^T \mathbf{M}_e^{-1} \frac{1}{\xi_e}\mathbf{B}_e \phi_e^{n+1} = -\frac{1}{\delta t}\mathbf{B}_e^T \rho_e^n \lambda_e^* \tag{3.49}$$

Finally, once the discrete correction equation is obtained from equation (3.48), the full algorithm including PDF terms is summarized as follows:

$$\left(\frac{1}{\delta t}\widehat{\mathbf{M}}_e\rho_e^n + \mathbf{D}_e\mu_e^n + \mathbf{N}_e\rho_e^n\right)\lambda_e^* = \frac{1}{\delta t}\widehat{\mathbf{M}}_e\rho_e^n\lambda_e^n + \mathbf{B}_e p_e^n + \mathbf{M}_e s_e^n + \frac{1}{\delta t}\mathbf{M}_e\rho_e^n(\xi_e - 1)\left(\lambda_\Gamma^{n+1} - \lambda_e^n\right) \tag{3.50}$$

$$\mathbf{B}_e^T\widehat{\mathbf{M}}_e^{-1}\mathbf{B}_e\phi_e^{n+1} = -\frac{1}{\delta t}\mathbf{B}_e^T\rho_e^n\lambda_e^* \tag{3.51}$$

$$\lambda_e^{n+1} = \lambda_e^* + \delta t\widehat{\mathbf{M}}_e^{-1}\frac{1}{\rho_e^n}\mathbf{B}_e\phi_e^{n+1} \tag{3.52}$$

where $\widehat{\mathbf{M}}_e = \mathbf{M}_e\xi_e$ denotes a modified mass matrix. Using this later one in the time terms, the PDF fractional-step algorithm is very similar to the standard algorithm (equations (3.29) to (3.31)) with the extra source term $\frac{1}{\delta t}\mathbf{M}_e\rho_e^n(\xi_e - 1)(\lambda_\Gamma^{n+1} - \lambda_e^n)$ that vanishes when $\chi_e = 0$.

***Remark* 11** *In this FEM scheme, as all the components of the pressure-correction gradient term $\widehat{\mathbf{M}}_e^{-1}\mathbf{B}_e\phi_e^{n+1}$ are located at the same element nodes, the isolation mentioned in Remark 8 occurs for all the space directions contrary to the previous finite difference scheme (isolation in the face-normal direction only) [IBF14].*

***Remark* 12** *In a way, the proposed PDF method, when the forcing term is split, can be considered as hybrid (with respect to the categories presented in Chapter 1) because it involves both an addition of new terms (prediction equation) and a modification of operators (kind of Laplace operator in the projection equation).*

4. Interpolation of the imposed velocity

Summary

4.1. Preamble

In this short chapter we discuss theoretical aspects related to the proposed interpolation strategies. The idea is to define methods to compute $\mathbf{u}_\Gamma$:

1. $\mathbf{u}_\Gamma$ is directly equal to the velocity of the immersed boundary (*i.e.* Dirichlet BC), what we call the "Direct Assignment",

2. $\mathbf{u}_\Gamma$ is linearly interpolated within the direction normal to the immersed boundary, what we call the "Directional Interpolation" (*cf.* Section 4.2),

3. $\mathbf{u}_\Gamma$ is linearly interpolated using an approximate normal derivative computed using neighbors values, what we call the "Multi-directional Interpolation" or "Mean Gradient Interpolation" (*cf.* Section 4.3),

4. $\mathbf{u}_\Gamma$ is linearly interpolated whether using the "Directional Interpolation" or the "Multi-directional Interpolation" depending on the geometry, what we call the "Hybrid Strategy" (*cf.* Section 4.4).

As mention in Chapter 1, in theory, we expect the direct assignment to be first order accurate in space whereas the interpolation strategies should allow to reach order 2 , at least for the quantities decomposed in a $\mathbb{Q}_1$ basis (velocity and pressure gradient) in steady cases. Indeed, the $\mathbb{Q}_1$ Finite Element is second order accurate and the linear interpolation also gives a second order approximation (compared to the first order "staircase" approximation) of the immersed boundary. Moreover, we have experimental evidence showing that the splitting of the PDF source term, coupled

linear interpolation of the imposed velocity, allows to reach order 2 in practice with a Finite Difference scheme [IBF14].

We can say that this assertion is approximatively verified in practice in the 2D tilted Poiseuille case (*cf.* Section 5.4.1) with order values superior to 1.8 in $\mathscr{L}^2$ norm and superior to 1.7 in $\mathscr{L}^\infty$ norm (depending on the considered quantity and tilt angle) for all interpolation techniques (except for the multi-directional in $\mathscr{L}^\infty$with a titl angles of 11° and 30°: the value is close to 1). Considering the Taylor-Couette and circular cylinder cases (*cf.* Sections 5.4.2 and 5.4.3), for all interpolation techniques, the assertion is verified for the velocity in $\mathscr{L}^2$, with order values superior to 1.7, but not in $\mathscr{L}^\infty$ norm, with values of 1.5 or less.

Nota: From now on, the time index exponent notation will be omitted in this section for purpose of readability, keeping in mind that the imposed velocity is interpolated at each time step.

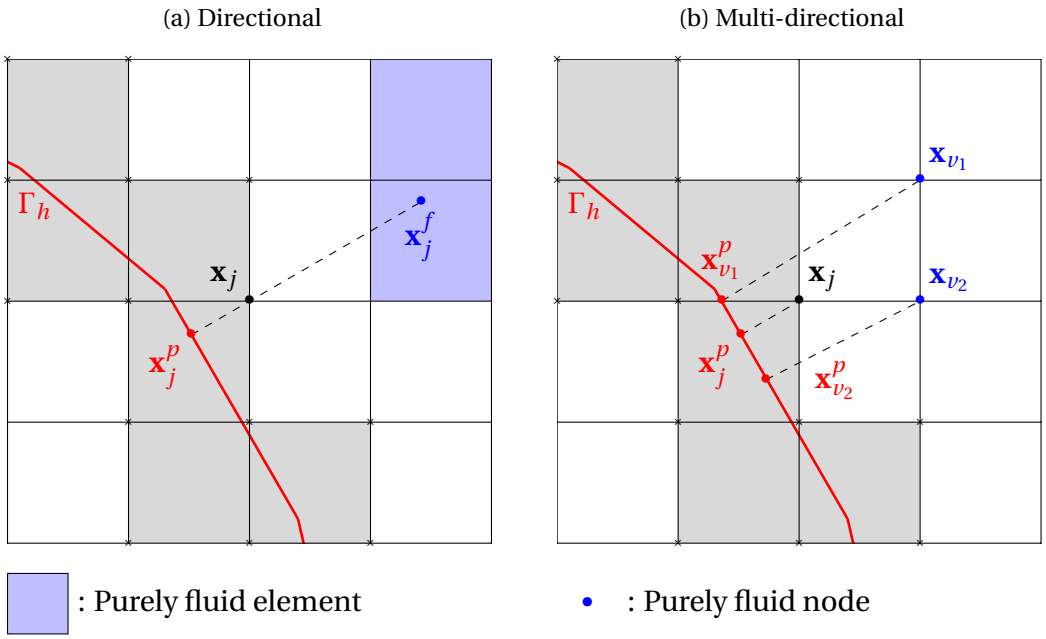


Figure 4.1. Schematic drawings of the different interpolations techniques.

4.2. Directional interpolation

Given the coordinates $\mathbf{x}_j^p$ of the approximate projection of node j, we can reconstruct an outward normal vector:

$$\forall j \in J,\ \mathbf{n}_j = \frac{\mathbf{x}_j - \mathbf{x}_j^p}{|\mathbf{x}_j - \mathbf{x}_j^p|} \tag{4.1}$$

and then we find a point $\mathbf{x}_j^f$ in the prolongation of $\mathbf{n}_j$ (*cf.* Figure 4.1a) such as:

$$\forall j \in J,\ \mathbf{x}_j^f = \mathbf{x}_j + d_j^f\, c\, \mathbf{n}_j \tag{4.2}$$

with the distance d_j^f, the maximum distance between a node j and its neighbors, defined as follows:

$$\forall j \in J,\ d_j^f = \max_{e \in E_j^0} \left(\max_{i \in J_e} \left| \mathbf{x}_j - \mathbf{x}_i \right| \right) \tag{4.3}$$

and $c \in]1, +\infty]$ a coefficient (*cf.* Figure 2.4 for the definition of E_j^0). The velocity at $\mathbf{x}_j^f$ is computed using the FE basis functions:

$$\forall j \in J,\ \mathbf{u}(\mathbf{x} = \mathbf{x}_j^f) \approx \sum_{k=1}^{N_N} \lambda_k \varphi_k \left(\mathbf{x} = \mathbf{x}_j^f \right) \tag{4.4}$$

with $\lambda_k \in \mathbb{R}^d$ the component k of the decomposition of the velocity in the FE basis and φ_k the $\mathbb{Q}_1$ basis function associated to the node k. However, as the support of φ_k is compact and equal to ${}_k\Omega$, *i.e.* $\varphi_k \in \mathscr{D}({}_k\Omega)$, with:

$$\forall k \in [\![1, N_N]\!],\ {}_k\Omega = \bigcup_{e \in E_k^0} \Omega_e \tag{4.5}$$

only the nodes belonging to the element K_f containing the point $\mathbf{x}_j^f$ imply a non-zero term in the sum. Then equation (4.4) becomes:

$$\forall j \in J, \mathbf{u}(\mathbf{x} = \mathbf{x}_j^f \approx \mathbf{u}_j^f = \sum_{k \in J_f} \lambda_k \varphi_k \left(\mathbf{x} = \mathbf{x}_j^f \right) \tag{4.6}$$

Note that K_f has to be "purely fluid", which means that all nodes of J_f must belong only to elements not crossed by the boundary or, said otherwise:

$$\forall f \in [\![1, N_E]\!],\ \left(K_f \text{ is "purely fluid"} \right) \Longleftrightarrow \left(\forall k \in J_f,\ \forall e \in E_k^0,\ \chi_e = 0 \right) \tag{4.7}$$

in order for $\mathbf{u}_j^f$ to be clearly defined. Indeed, if it is not the case, we obtain a circular definition for $\mathbf{u}_j$. To prevent this from happening in practice, we take $c = 1.1$ at first. Then, if the element K_f is not "purely fluid", we retry with $c = 2.1$.

Finally, if we denote $\mathbf{u}_j^p = \mathbf{u}(\mathbf{x} = \mathbf{x}_j^p)$ (given by the IB condition), we can interpolate the velocity at node j as follows:

$$\forall j \in J,\ \mathbf{u}_\Gamma(\mathbf{x} = \mathbf{x}_j) := \mathbf{u}(\mathbf{x} = \mathbf{x}_j) \approx \mathbf{u}_j = \mathbf{u}_j^p + \frac{\mathbf{u}_j^f - \mathbf{u}_j^p}{|\mathbf{x}_j^f - \mathbf{x}_j^p|} |\mathbf{x}_j - \mathbf{x}_j^p| \tag{4.8}$$

4.3. Multi-directional interpolation

The idea of this approach (inspired by [IBF14] and roughly schematized in Figure 4.1b) can be summarize as follows:

1. For each purely fluid node surrounding the node $j \in J$, we compute an approximate derivative, in the direction normal to Γ_h, of the fluid velocity.
2. We compute the arithmetic mean of the obtained approximate derivatives.
3. We use this mean normal derivative to compute $\mathbf{u}_j$.

However, to be able to compute approximate normal derivative of the velocity at the nodes surrounding j, we need the projections of those nodes on Γ_h. Those purely fluid neighbors are denoted as follows:

$$\forall j \in [\![1, N_N]\!], \quad V_j = \left\{ v \in [\![1, N_N]\!] \,\middle|\, \exists e \in E_j^0 / v \in J_e \text{ and } \forall f \in E_v^0,\ \chi_f = 0 \right\} \tag{4.9}$$

Regarding geometrical data reconstruction, the method to achieve the projection slightly differs between the weighting and optimization approaches:

In the weighting approach: First, new elemental projections are computed:

$$\forall e \in [\![1, N_E]\!] / \chi_e = 1, \forall j \in J_e, \forall v \in V_j, \quad \mathbf{p}_e^v = \mathbf{x}_v + l_e^v \mathbf{n}_e \tag{4.10}$$

with the definition of l_e^v being identical to the one presented in Section 2.3.1. Then they are assembled in the following way:

$$\forall j \in [\![1, N_N]\!] / \alpha_j \neq 0, \forall v \in V_j, \quad \mathbf{x}_v^p = \frac{1}{\alpha_v} \sum_{e \in E_j^\chi} \alpha_e^v \mathbf{p}_e^v \tag{4.11}$$

with:

$$\forall j \in [\![1, N_N]\!] / \alpha_j \neq 0, \forall v \in V_j, \quad \alpha_v = \sum_{e \in E_j^\chi} \alpha_e^v \tag{4.12}$$

In the optimization approach: For nodes of V_j we must collect facets which intersect neighbors or neighbors elements, which means:

$$\forall v \in \bigcup_{j \in J} V_j, \quad I_v = \left\{ k \in [\![1, N_S]\!] \,\middle|\, \exists e \in E_v^1 / S_e^k \neq \emptyset \right\} \tag{4.13}$$

with the definition of E_v^0 given in Figure 2.4. Then the approximated minimization problem (2.14) is also solved for nodes v where $I_v \neq \emptyset$ using the methodology presented in Section 2.3.2.

Provided the projections of neighbors, the velocity imposed at node j is computed as follows:

$$\forall\, j \in J,\ \mathbf{u}_\Gamma(\mathbf{x}=\mathbf{x}_j) \approx \mathbf{u}_j = \mathbf{u}_j^p + \gamma_j \sum_{\nu \in V_j} \frac{\mathbf{u}_\nu - \mathbf{u}_\nu^p}{|\mathbf{x}_\nu - \mathbf{x}_\nu^p|} |\mathbf{x}_j - \mathbf{x}_j^p| \tag{4.14}$$

with:

$$\forall\, j \in J,\quad \begin{cases} \gamma_j = \dfrac{1}{\mathrm{card}(V_j)} & \text{if } \mathrm{card}(V_j) > 0 \\ \gamma_j = 0 & \text{else} \end{cases} \tag{4.15}$$

4.4. Hybrid strategy

In future works, we will investigate turbulent immersed wall laws. As wall laws are usually directional, we propose a hybrid strategy consisting in applying the directional approach where possible (*i.e.* nodes j for which the fluid point $\mathbf{x}_j^f$ is located in a purely fluid element with $c = 1.1$) and multi-directional approach elsewhere.

4.5. Towards the Neumann BC

In this chapter, the different ways that we developed in order to impose a Dirichlet immersed BC for the velocity were covered. However, one of the main advantages of the above described interpolation techniques resides in the fact that they can also be used to impose Neumann boundary conditions (even if no test case presented in Chapter 5 uses this type of BC, we detail the methodology here, as a perspective).

Indeed, if we denote $\partial\mathbf{u} : \Gamma \to \mathbb{R}^d$ the Neumann immersed BC in each direction:

- **In the directional approach:** we can consider that $\partial\mathbf{u}$ represents the value of the linear interpolation slope in each direction or, said otherwise:

$$\forall\, j \in J,\ \frac{\mathbf{u}_j^f - \mathbf{u}_j^p}{|\mathbf{x}_j^f - \mathbf{x}_j^p|} \approx \partial\mathbf{u}(\mathbf{x}=\mathbf{x}_j^p) \tag{4.16}$$

  provided that we have a satisfactory way to evaluate $\partial\mathbf{u}$ at each $\mathbf{x}_j^p$ (as $\mathbf{x}_j^p$ may not be located exactly on Γ). With that, we can compute an approximate value for each $\mathbf{u}_j^p$:

$$\forall\, j \in J,\ \mathbf{u}_j^p \approx \mathbf{u}_j^f - |\mathbf{x}_j^f - \mathbf{x}_j^p|\, \partial\mathbf{u}(\mathbf{x}=\mathbf{x}_j^p) \tag{4.17}$$

  and then use equation (4.8) to compute $\mathbf{u}_\Gamma$ which means:

$$\forall\, j \in J,\ \mathbf{u}_\Gamma(\mathbf{x}=\mathbf{x}_j) \approx \mathbf{u}_j^f + \left(|\mathbf{x}_j - \mathbf{x}_j^p| - |\mathbf{x}_j^f - \mathbf{x}_j^p|\right)\, \partial\mathbf{u}(\mathbf{x}=\mathbf{x}_j^p) \tag{4.18}$$

or, as $\mathbf{x}_j^p$, $\mathbf{x}_j$ and $\mathbf{x}_j^f$ are aligned:

$$\forall\, j \in J,\ \mathbf{u}_\Gamma(\mathbf{x}=\mathbf{x}_j) \approx \mathbf{u}_j^f - |\mathbf{x}_j^f - \mathbf{x}_j|\, \partial\mathbf{u}(\mathbf{x}=\mathbf{x}_j^p) \tag{4.19}$$

or, as

- **In the multi-directional approach:** the idea is quite the same except that, instead of only using the value of the Neumann BC as the slope of the linear interpolation between $\mathbf{x}_j^p$ and $\mathbf{x}_j^f$, we use it for each purely fluid neighbor of $\mathbf{x}_j$:

$$\forall\, j \in J,\ \forall\, v \in V_j\ \frac{\mathbf{u}_v - \mathbf{u}_v^p}{|\mathbf{x}_v - \mathbf{x}_v^p|} \approx \partial\mathbf{u}(\mathbf{x}=\mathbf{x}_v^p) \tag{4.20}$$

  provided that, once again, we have a satisfactory way to evaluate $\partial\mathbf{u}$ at each $\mathbf{x}_v^p$. Then, similar the directional approach, we are able to compute each $\mathbf{u}_v^p$ and provide a value to $\mathbf{u}_\Gamma$ using equation (4.14).

Hence, we are able to impose the two usual wall BC on the immersed boundary:

- Slip: Neumann BC in the direction normal to the immersed boundary, Dirichlet BC in the tangential direction(s),
- No slip: Dirichlet BC in all directions.

As other interesting perspectives, that were not tackled during this PhD, we can mention:

- The investigation of ways to impose pressure BC via linear interpolation,
- Using turbulent wall laws instead of linear functions to compute $\mathbf{u}_\Gamma$ (a new PhD project focusing on this topic was launched in October 2020).

5. Numerical results and discussions

Summary

5.1. Preamble

5.1.1. Definition of indicators and approximations

We detail the definitions of discrete $\mathscr{L}^2$ and $\mathscr{L}^\infty$ norms used in this thesis. For a scalar quantity a, we have:

$$|a|_2 \approx |a|_2^h = \left(\frac{1}{n} \sum_{i=1}^{n} a_i^2 \right)^{\frac{1}{2}} \tag{5.1}$$

$$|a|_\infty \approx |a|_\infty^h = \max_{i \in [\![1,n]\!]} |a_i| \tag{5.2}$$

where a_i is the value of the quantity a computed at node or element i depending on the discretization ($\mathbb{Q}_0$ or $\mathbb{Q}_1$) and n the number of nodes or elements in the fluid domain ($j \in [\![1, N_N]\!] / \mathbf{x}_j \in \Omega_f$ or $e \in [\![1, N_E]\!] / \Omega_e \cap \Omega_f \neq \emptyset$). With those definitions, we are able to construct the relative norm of the error related to the quantity a, provided its analytical solution s:

$$|a|_2^r = \frac{|a-s|_2^h}{|s|_2^h} \tag{5.3}$$

$$|a|_\infty^r = \frac{|a-s|_\infty^h}{|s|_\infty^h} \tag{5.4}$$

We also detail our approximation of the aerodynamic force induced by the flow around an obstacle. At a fully continuous level, the definition of the aerodynamic force is the following:

$$\mathbf{t}_\Gamma := \int_\Gamma (\bar{\bar{\sigma}} - \bar{\bar{I}} p) \cdot \mathbf{n}_\Gamma \tag{5.5}$$

and thank to the divergence theorem we have:

$$\mathbf{t}_\Gamma = \int_{\Omega_s} \nabla \cdot (\bar{\bar{\sigma}} - \bar{\bar{I}} p) \cdot \mathbf{n}_\Gamma \tag{5.6}$$

If we denote:

$$\Omega_s^h = \bigcup_{e \in \mathscr{C}} \Omega_e \quad \text{with} \quad \mathscr{C} = \{ e \in [\![1, N_E]\!] / \Omega_e \cap \Omega_s \neq \emptyset \} \tag{5.7}$$

we can approximate the aerodynamic force at a time-discrete level:

$$\mathbf{t}_\Gamma = \int_{\Omega_s} \nabla \cdot (\bar{\bar{\sigma}} - \bar{\bar{I}} p^{n+1}) \approx \int_{\Omega_s^h} \nabla \cdot (\bar{\bar{\sigma}}^{n+1} - \bar{\bar{I}} p^{n+1}) = \mathbf{t}_\Gamma^h \tag{5.8}$$

Then, if we consider the semi-discrete weak formulation of the complete momentum balance equation on Ω_s^h, we have (without considering the gravity term):

$$\int_{\Omega_s^h} \left[\rho^n \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\delta t} + \nabla \cdot (\rho^n \mathbf{u}^n \otimes \mathbf{u}^{n+1}) \right] + \int_{\Omega_s^h} \nabla \cdot (\bar{\bar{\sigma}}^{n+1} - \bar{\bar{I}} p^{n+1}) = \int_{\Omega_s^h} \mathbf{f}^{n+1} \tag{5.9}$$

where $\mathbf{f}^{n+1}$ is the PDF term. Therefore, the approximate value of the aerodynamic force can be computed as follows:

$$\mathbf{t}_\Gamma^h = \int_{\Omega_s^h} \mathbf{f}^{n+1} - \int_{\Omega_s^h} \left[\rho^n \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\delta t} + \nabla \cdot (\rho^n \mathbf{u}^n \otimes \mathbf{u}^{n+1}) \right] \tag{5.10}$$

As a first approximation, the integral of the total derivative is assumed negligible – *i.e.* steady state, uniformity and symmetry assumptions – which can lead to incorrect estimations (*cf.* Section 5.5). Thus we have:

$$\mathbf{t}_\Gamma^h \approx \int_{\Omega_s^h} \mathbf{f}^{n+1} = \int_{\Omega} \mathbf{f}^{n+1} \tag{5.11}$$

as the forcing term is zero outside Ω_s^h. Finally, the aerodynamic coefficients (drag and lift) are computed as follows:

$$C_d = \frac{2\mathbf{t}_\Gamma^h \cdot e_x}{\rho {U_\infty}^2 d} \quad \text{and} \quad C_l = \frac{2\mathbf{t}_\Gamma^h \cdot e_y}{\rho {U_\infty}^2 d} \tag{5.12}$$

with U_∞ a reference-velocity magnitude (magnitude of the inlet velocity for both cylinder and NACA airfoil cases), d a reference length (diameter for cylinder and chord length for a NACA airfoil) and assuming that x (resp. y) is the streamwise (resp. spanwise) flow direction. In the case of the unsteady flow around a circular cylinder, we also consider the Strouhal number, defined as follows:

$$St = \frac{2r}{U_\infty} f \tag{5.13}$$

with r the radius of the cylinder and f the vortex detachment frequency.

5.1.2. Development history

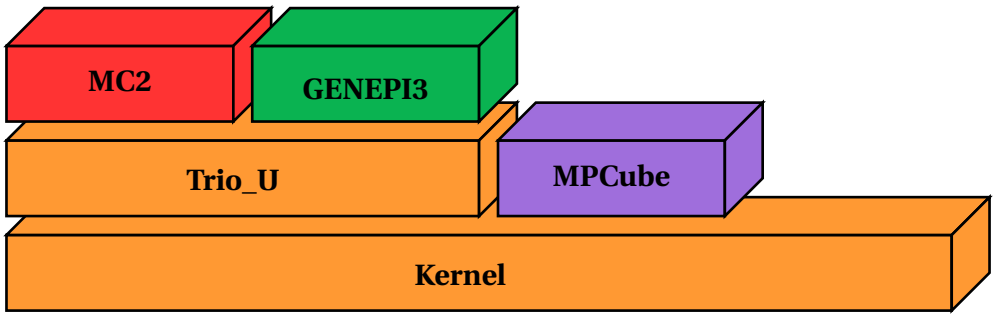


Figure 5.1. Trio_U and Kernel before the creation of TRUST (source [CEAa]).

Before 1994, there was no common kernel between the different CFD codes developed by the CEA. There were only disseminated applications, such as GENEPI (FE code written in FORTRAN dedicated to Steam Generator numerical modeling) or FLICA (FE thermal hydraulics code used to study the flow within the core). After 1994, in an effort to pool the different developments, Trio_U, and its Kernel (both written in C++), were created (*cf.* Figure 5.1). With them was born GENEPI3, the version of GENEPI based on Trio_U (*i.e.* partially written in C++). Other CEA codes were modified to take advantage of the common developments (notably parallelization) and validation material incorporated in Trio_U and its Kernel, for instance FLICA and MC2 (used to simulate the flow through the reactor core but at a different scale compared to FLICA). The codes based on Trio_U or its kernel are called BALTIK, an acronym for **B**uild an **A**pplication **L**inked to **TrI**o_U **K**ernel.

In 2015, Trio_U was divided in two parts: TRUST and TrioCFD (cf Figure 5.2).

- TRUST (**TR**io_**U** **S**oftware for **T**hermohydraulics) basically contains all the kernel as well as code coupling, verification and validation tools. Its strength resides in its extensive validation basis.
- TrioCFD is a BALTIK based on TRUST. It contains all turbulence models as well as front tracking methods and other closure laws or models.

In the same motion, TRUST became open source (but not the BALTIK as they include experimental data or models, which remain intellectual property of the CEA).

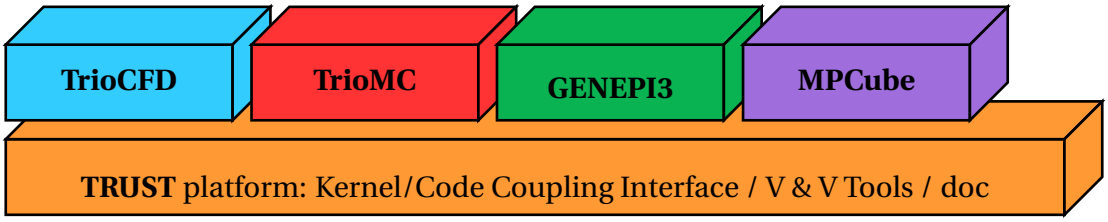


Figure 5.2. TRUST platform and the CEA codes based on it (source [CEAa]).

At the beginning of this PhD project, the Penalized Direct Forcing method and the interpolation strategies were developed as a part of GENEPI3 (*cf.* Figure 5.2). This choice was mainly motivated by two reasons:

1. As briefly introduced in Section 1.5, some physical models of interest for the aimed application are included in the code,
   - HEM,
   - weakly compressibility,
   - tabulated EOS of water,
   - *etc.*
2. The code contains a feature (called opaque obstacle) that can be easily converted in the PDF forcing terms and, therefore, facilitate the implementation.

However, later on, we decided to transfer our developments from GENEPI3 to the TRUST platform so all the BALTIK could use the order 2 PDF method. Hence, during this PhD, results have been obtained using three different software environments:

1. the version of GENEPI3 based on the version 1.7.4 of the TRUST platform at this stage, only the direct assignment and directional interpolation were implemented),
2. the version 1.8.2 of the TRUST platform alone at this stage, multi-directional and hybrid interpolations were added),
3. the version of GENEPI3 based on the version 1.8.2 of the TRUST platform. Indeed, as the TRUST platform is updated more frequently than GENEPI3, there were a shift between the up-to-date version of TRUST and the version used by GENEPI. We had to adapt GENEPI3 to the last version of TRUST.

For more details about GENEPI3, the interested reader can refer to [Gra+90; GOch] and, for more details about TRUST, to [CEAb; CEAa].

5.2. List of cases

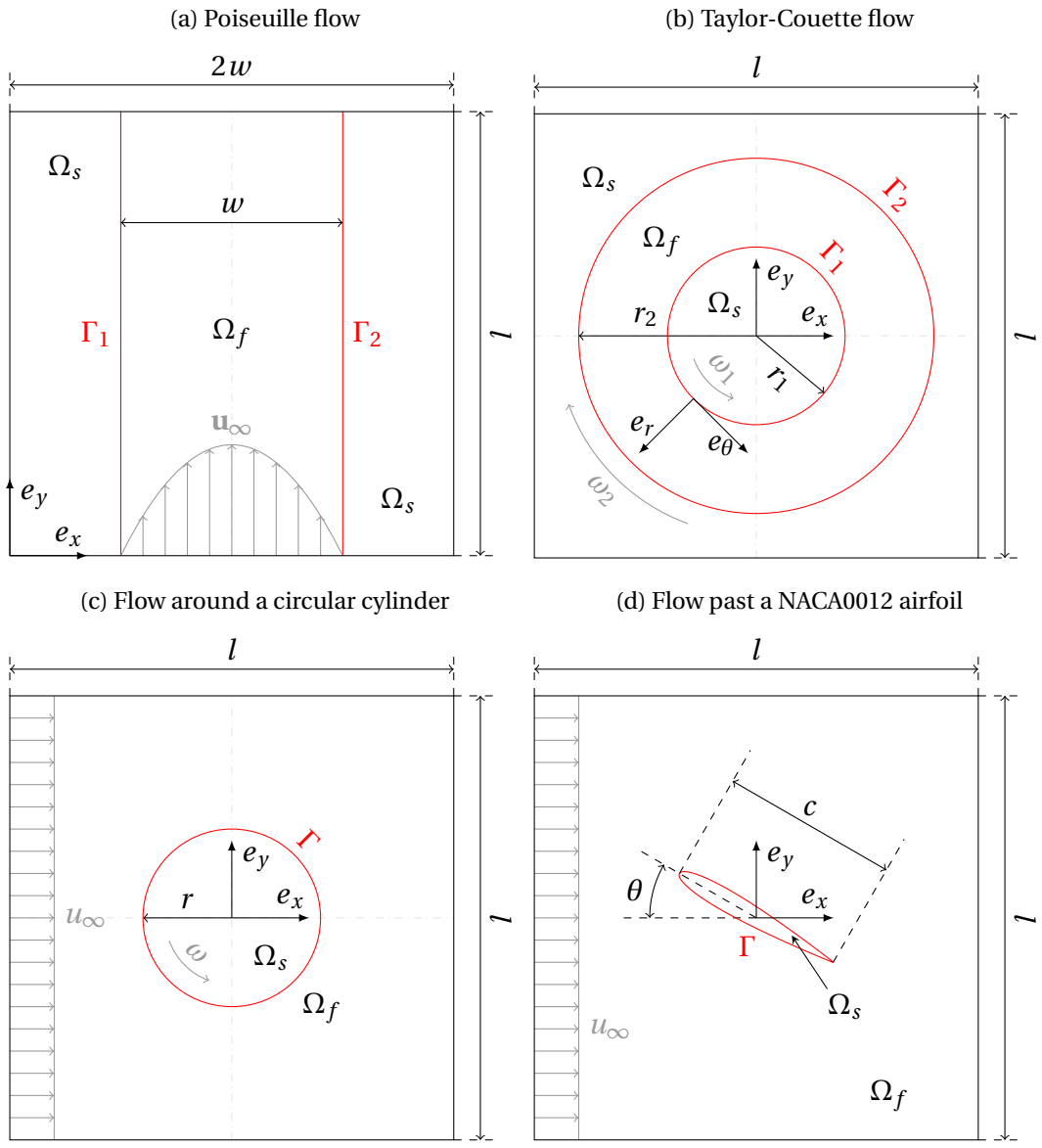


Figure 5.3. Schematic representation of the computational domain for the different laminar validation cases.

In practice, the IB are considered "Infinitely thin" in all the test cases listed in this section, meaning that the governing equations are solved in the non-physical domain too – we keep the notation of Ω_s in the academic test cases in reference to the physical problems. This property is granted by construction as the proposed method treats both sides of an IB in the exact same way.

5.2.1. Aligned laminar 2D Poiseuille flow

Poiseuille flow refers to a viscous fluid flow between two parallel plane plates, what we call channel, or within a cylindrical tube. In dimension two, which is our case of interest for verification purposes, both cases are equivalent. Figure 5.3a, paired with Table 5.1, gives our test case configuration where:

- Ω_f represents the physical domain whereas Ω_s represents the non-physical one ($\Omega = \Omega_f \cup \Omega_s$ is the computational or fictitious domain).
- Γ_1 and Γ_2 are the two immersed boundaries at which a no-slip condition is considered.
- $\mathbf{u}_\infty$ represents the inlet velocity field (given by the analytical solutions of the Poiseuille flow problem detailed hereafter) and $U_\infty = \max(|\mathbf{u}_\infty|)$.

In this case, we consider a constant viscosity (so that the Reynolds number does not depend on space) and its value is computed as follows:

$$\mu = \frac{\rho U_\infty w}{Re} \tag{5.14}$$

or said otherwise: $\mu = \nu\,\rho$. If we do the numerical application using the value in Table 5.1, we obtain:

$$\mu = 1014\ \text{kg.m}^{-1}.\text{s}^{-1} \tag{5.15}$$

Nota: GENEPI3 was used to carry out this test case (*cf.* Section 5.1.2) so, in practice, the density of the fluid is taken equal to the one of water at 20°C and 1 bar (or 10^5 Pa) to match the tabulated EOS of water.

5.2.1.1. Analytical solution

The distance between the plates is assumed widely smaller than the dimensions of the plates (*i.e.* $w \ll l$). Therefore, with x and y the coordinates along e_x and e_y respectively, we can use the lubrication theory to obtain the following analytical solutions:

$$\forall (x,y) \in [0.5w, 1.5w] \times \mathbb{R}, \quad \mathbf{u} \cdot e_x = 0 \tag{5.16}$$

$$\mathbf{u} \cdot e_y = \frac{4U_\infty (x - 0.5w)}{w}\left(1 - \frac{x - 0.5w}{w}\right) \tag{5.17}$$

$$\nabla p \cdot e_x = 0 \tag{5.18}$$

$$\nabla p \cdot e_y = -\frac{8\rho U_\infty \nu}{w^2} \tag{5.19}$$

Table 5.1. Geometrical and physical parameters used for the 2D Poiseuille flow (channel aligned with the mesh).

w (m)	l (m)	U_∞ (m.s^{-1})	ρ (kg.m^{-3})	ν (m^2.s^{-1})	$Re = U_\infty w \nu^{-1}$
1	4	1	1014	1	1

5.2.2. Tilted laminar 2D Poiseuille flow

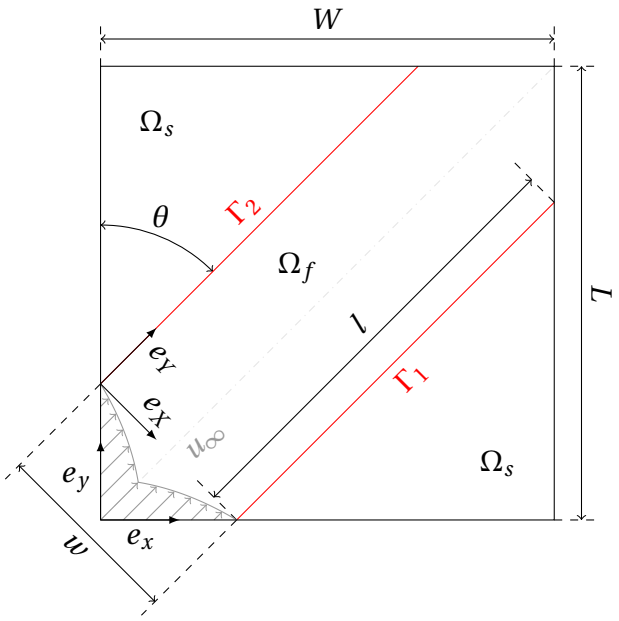


Figure 5.4. Schematic representation of the computational domain for the tilted 2D Poiseuille flow.

This case is almost the same as the previous one, only the orientation of the channel in the fictitious domain changes. Indeed, in the previous test case (*cf.* Section 5.2.1) the immersed boundaries are aligned with the mesh axis (here e_y) but, in general, it will not be the case for industrial applications – it is the interest of the method. Hence, a case in which the immersed boundaries are not coincident with the mesh is needed to test the robustness of the method.

The same notations are used to describe the case via Figure 5.4 and Table 5.2, except that:

- (e_X, e_Y) is the rotated frame with $e_X = \cos(\theta)e_x - \sin(\theta)e_y$ and $e_Y = \sin(\theta)e_x + \cos(\theta)e_y$
- W and L are respectively the actual width and length of the domain with $W = w\cos(\theta) + l\sin(\theta)$ and $L = w\sin(\theta) + l\cos(\theta)$

Table 5.2. Geometrical and physical parameters used for the 2D Poiseuille flow (channel tilted).

w(m)	l(m)	θ(°)	U_∞(m.s^{-1})	ρ(kg.m^{-3})	ν(m^2.s^{-1})	$Re = U_\infty w \nu^{-1}$
1	4	11	1	1014	1	1
1	4	30	1	1014	1	1
1	4	45	1	1014	1	1

The viscosity is also considered constant and computed using equation (5.14). The numerical application using the values of Table 5.2 gives:

$$\mu = 1014 \text{ kg.m}^{-1}.\text{s}^{-1} \tag{5.20}$$

Nota: GENEPI3 was used to carry out this test case (*cf.* Section 5.1.2) so, in practice, the density of the fluid is taken equal to the one of water at 20°C and 1 bar (or 10^5 Pa) to match the tabulated EOS of water.

5.2.2.1. Analytical solution

The analytical solutions are computed in the same way as in Section 5.2.1.1, except that they are expressed in (e_X, e_Y):

$$\forall (X,Y) \in [0,w] \times \mathbb{R}, \quad \mathbf{u} \cdot e_X = 0 \tag{5.21}$$

$$\mathbf{u} \cdot e_Y = \frac{4U_\infty X}{w}\left(1 - \frac{X}{w}\right) \tag{5.22}$$

$$\nabla p \cdot e_X = 0 \tag{5.23}$$

$$\nabla p \cdot e_Y = -\frac{8\rho U_\infty \nu}{w^2} \tag{5.24}$$

5.2.3. Laminar Taylor-Couette flow

Taylor-Couette flow refers to the flow between two infinitely long concentric circular cylinders which are rotating at different angular velocities. Figure 5.3b, paired with Table 5.3, gives the test case configuration where:

- Ω_f represents the physical domain whereas Ω_s represents the non-physical one ($\Omega = \Omega_f \cup \Omega_s$ is the computational or fictitious domain).
- Γ_1 and Γ_2 are the immersed boundaries corresponding respectively to the inner and outer cylinders. They correspond to a no-slip condition, which means a Dirichlet BC with the Dirichlet fluid velocities given by the cylinders velocities.

Table 5.3. Geometrical and physical parameters used for the laminar Taylor-Couette flow.

r_1(m)	r_2(m)	ω_1(rad.s^{-1})	ω_2(rad.s^{-1})	ν(m^2.s^{-1})	$Re = \lvert\omega\rvert r_1^2 \nu^{-1}$
5×10^{-1}	1	1	-1	2.5×10^{-1}	1

- r_1 and r_2 are respectively the radii of the inner and outer cylinders.
- $l = 2r_2 + \frac{1}{4}$ is the side of the square domain.
- ω_1 and ω_2 are respectively the inner and outer cylinders angular velocities.
- (e_r, e_θ) is the polar frame.

Once again, the viscosity is considered constant and computed using the definition and value of the Reynolds number. The numerical application using the values of Table 5.3, with $\rho = 1$ kg.m^{-3}, gives:

$$\mu = 2.5 \times 10^{-1} \text{ kg.m}^{-1}.\text{s}^{-1} \tag{5.25}$$

The stability of this flow in ensured by a condition on the Taylor number, denoted *Ta*:

$$Ta := \frac{{\omega_1}^2 r_m (r_2 - r_1)^3}{\nu^2} < Ta_c \quad \text{with} \quad r_m = \frac{r_1 + r_2}{2} \tag{5.26}$$

where $Ta_c \approx 1.712$ is the critical Taylor number computed by linear analysis [GHP01]. The numerical application gives, in our case, $Ta = 1$, so the criterion is respected.

5.2.3.1. Analytical solution

Using lubrication theory once again, the flow reaches a steady state, in which the fluid velocity is purely azimuthal:

$$\forall (r,\theta) \in [r_1, r_2] \times [0, 2\pi[, \quad \mathbf{u} \cdot e_\theta = \omega_1 \frac{a - b^2}{1 - b^2} r + {\omega_1 r_1}^2 \frac{1 - a}{1 - b^2} \frac{1}{r} \tag{5.27}$$

with:

$$a = \frac{\omega_2}{\omega_1} \quad \text{and} \quad b = \frac{r_1}{r_2} \tag{5.28}$$

5.2.4. Laminar flow around a circular cylinder

The flow around a circular cylinder is a widely studied problem in the field of fluid dynamics. Figure 5.3c, paired with Table 5.4, gives the test case configuration where:

Table 5.4. Geometrical and physical parameters used for the laminar flow around a cylinder.

r (m)	U_∞ (m.s^{-1})	ω (rad.s^{-1})	μ (kg.m^{-1}.s^{-1})	$Re = 2\rho U_\infty r \mu^{-1}$
5×10^{-1}	1	0	5×10^{-2}	2×10^{1}
5×10^{-1}	1	2	5×10^{-2}	2×10^{1}
5×10^{-1}	1	0	10^{-2}	10^{2}
5×10^{-1}	1	2	10^{-2}	10^{2}

- Ω_f represents the physical domain whereas Ω_s represents the non-physical one ($\Omega = \Omega_f \cup \Omega_s$ is the computational or fictitious domain).
- Γ is the immersed boundary corresponding to the surface of the cylinder.
- r is the cylinder radius.
- $l = 120r$ is the side of the square domain. It is rather large to avoid boundary effects.
- $\mathbf{u}_\infty$ represents the uniform inlet velocity field.
- ω is the cylinder angular velocity.

Once again, the viscosity is considered constant and computed using the definition and value of the Reynolds number. The values are gathered in Table 5.4.

There is no analytical solution, but many experiments and simulations give macroscopic indicators, such as the drag and lift coefficients, as comparison elements in the static case (*i.e.* $\omega = 0$ rad.s^{-1}) and the rotating case (*i.e.* $\omega \neq 0$ rad.s^{-1}), even if this last case has not been extensively studied experimentally.

5.2.5. Laminar flow past a NACA0012 airfoil

The flow past a NACA airfoil is a typical issue in aerodynamics. Usually, due to the high velocity of flying objects such as planes, it involves turbulence modeling. However, some laminar cases have been studied and simulated by R.C. SWANSON *et al.* [SL16]. This provides elements of comparison and test configurations, detailed in Figure 5.3d and table 5.5 where:

- Ω_f represents the physical domain whereas Ω_s represents the non-physical one ($\Omega = \Omega_f \cup \Omega_s$ is the computational or fictitious domain).
- Γ is the immersed boundary corresponding to the surface of the NACA airfoil.
- θ is the angle of attack.

Table 5.5. Geometrical and physical parameters used for the laminar flow past a NACA0012 airfoil.

θ (°)	c (m)	U_∞ (m.s^{-1})	ν (m^2.s^{-1})	$Re = U_\infty c \nu^{-1}$
0	1	5×10^{-1}	1×10^{-4}	5×10^3
1	1	5×10^{-1}	1×10^{-4}	5×10^3

- c is the chord of the NACA airfoil.
- $l = 10c$ is the side of the square domain.
- $\mathbf{u}_\infty$ represents the uniform inlet velocity field.

All those configurations are supposed to reach a steady state, as stated in [SL16]. Once again, the viscosity is considered constant and its value is determined via the definition of the Reynolds number which gives:

$$\mu = 0.1014 \text{ kg.m}^{-1}.\text{s}^{-1} \tag{5.29}$$

Nota: GENEPI3 was used to carry out this test case (*cf.* Section 5.1.2) so, in practice, the density of the fluid is taken equal to the one of water at 20°C and 1 bar (or 10^5 Pa) to match the tabulated EOS of water.

5.2.6. Industrial case involving the flow limiter

This case is representative of the flow into a hydraulic diode like the one presented in Figure 1.7. As we can see in Figures 5.5a, we have a volume mesh – which represents the fluid located in a planar model of the downcomer of a PWR including the inlet and outlet vessel – and a surface mesh – which comes from a CAD software and represents the shape of the flow limiter. The combination of the two, gives a volume mesh (*cf.* Figure 5.5b with embedded data about the immersed boundary (characteristic function, normal vector, *etc.*).

The case in itself corresponds to a LB-LOCA induced by a break on the cold inlet vessel (*cf.* Section 1.2). Thus, as the primary circuit is pressurized, the flow comes from the bottom of the core, with a flowrate of 5.2 kg.s^{-1} (this estimation is coming from system scale computations), to the break (*i.e.* the inlet vessel). The outflow pressure is fixed at an *ad hoc* value of 50 bar. In this case, we consider $\mu = \mu_f + \mu_T$ (*cf.* Section 3.1.1.3) where μ_f is given by the tabulated EOS of water at 50 bar (the internal energy balance is not solved so the enthalpy value is constant and equal to 1.1×10^6 J.kg^{-1}) and μ_T is given by the Schlichting's scalar turbulence model with $a_S = 0.047$ and the characteristic length of turbulence $L_T = 0.3$ m. For a more detailed description of the case, the interested reader can refer to [Bel18].

(a) Separated meshes

(b) Conjoint meshes

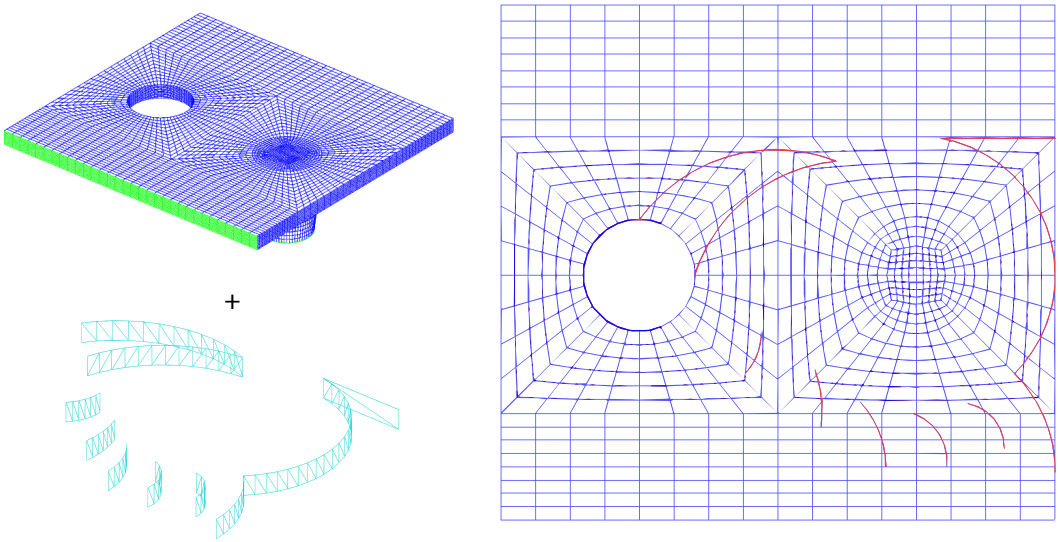


Figure 5.5. Mesh configuration of the case involving a device representative of the flow limiter.

5.3. Penalty parameter convergence study

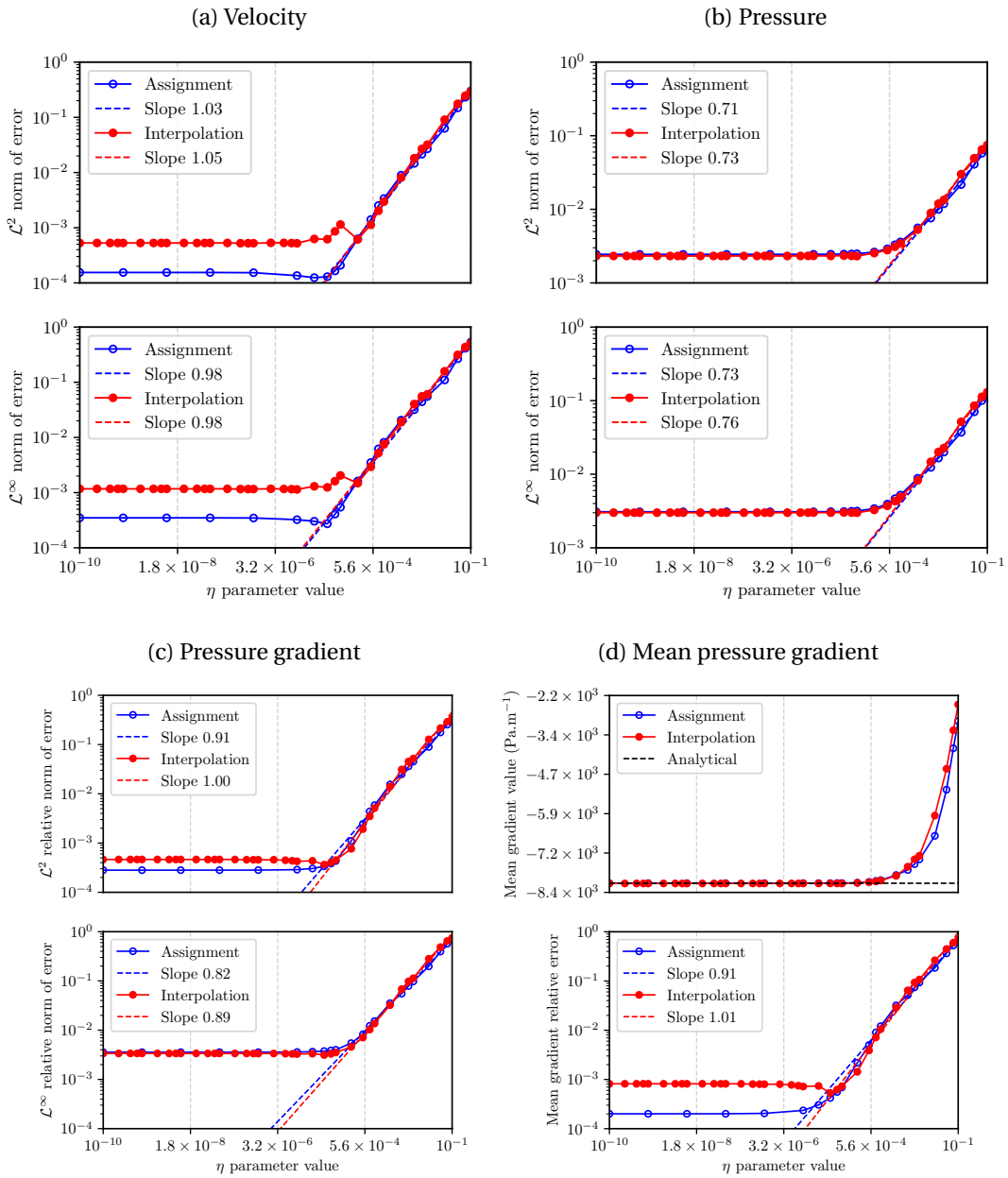


Figure 5.6. Evolution of relative norms ($\mathscr{L}^2$ and $\mathscr{L}^\infty$) of the error related to various variables with respect to the value of the penalty parameter η (direct assignment and directional interpolation).

The behavior of the PDF method with respect to the penalty parameter η have been studied in the case of a 2D Poiseuille flow. The channel is aligned with the grid and

the immersed boundaries are conforming to element interfaces in order to minimize the spatial error and focus on the penalty error. Only directional interpolation is considered here (*cf.* Section 4.2) , without loss of generality for the other interpolation techniques. Indeed, as only plane IB are considered and the space step is sufficiently low, there is no apparent reason to lack an interpolation point located in a purely fluid element for any of the nodes in the vicinity of the obstacles (*i.e.* hybrid interpolation provides the same results as the directional approach). Moreover, each node in the vicinity of the IB has only one purely fluid neighbor located in the direction normal the IB plane (*i.e.* directional and multi-directional approach are equivalent).

Figure 5.6a shows the $\mathscr{L}^2$ and $\mathscr{L}^\infty$ relative norms of the error of the component of the velocity oriented along the channel axis with respect to the penalty parameter. The blue curve marked with empty circles corresponds to results obtained with the variant of the method in which the value of the imposed velocity is directly assigned to the immersed Dirichlet boundary condition value. The red curve marked with disks correspond to the results obtained with the directional velocity interpolation technique described in Section 4.2. In both cases, the numerical order of convergence is close to 1, which is consistent with the conclusions of P. ANGOT *et al.* in [Ang99; ABF99].

However, it can be noted that the converged value of the relative error, in both norms, is higher when using the interpolation technique. This could be explained by the fact that the interpolation process adds a spatial error which is absent when using the direct assignment – in that case, the exact solution value is directly assigned to the imposed velocity because the immersed boundaries are conforming. The evolution of the pressure gradient with respect to the penalty parameter η (*cf.* Figure 5.6c) seems similar except that there is no noteworthy difference between the interpolation and assignment techniques. As pressure gradient is not interpolated, this tends to confirm the assumption about an added space error due to the interpolation process. Concerning pressure, in Figure 5.6b, results are similar for both techniques and an order of convergence of about 3/4 is found – a value which is, once again, consistent with the conclusions of P. ANGOT. Furthermore, it can be noted that the value of η for which the penalty error becomes lower than the other errors (mainly space error) is rather high (about 10^{-5}), which means that we don't need to degrade the conditioning of the computation matrices too much. Finally, another indicator has been computed: the mean pressure gradient between two sections of the channel. It is defined as follows:

$$\frac{\Delta p}{\Delta y} := \frac{p_2 - p_1}{y_2 - y_1} \tag{5.30}$$

Where:

- y_1 and y_2 are the coordinates along y axis of two sections of the channel S_1 and S_2 ($y_1 < y_2$).
- p_1 and p_2 are the arithmetic means of the computed pressure field in elements

crossed by the section S_1 and S_2.

It can be noted that this indicator also converges to the pressure gradient value with an order of about 1 – *cf.* the evolution of the error (computed between the indicator and the analytical value of the pressure gradient along the channel) plotted on Figure 5.6d.

***Remark* 13** *All the norms have been computed on a part of the fluid domain far from boundary to avoid boundary effects (i.e. for* $0.5 < x < 1.5$ *and* $1 < y < 3$*).*

5.4. Mesh convergence studies

5.4.1. 2D Poiseuille flow

The 2D Poiseuille flow with a tilted channel was used to carry out a mesh convergence study. For the three considered tilt angles (11, 30 and 45°), the space convergence of the velocity components, in both Cartesian (*cf.* Figures 5.8a, 5.8b, 5.11a, 5.11b, 5.13a and 5.13b) and channel (*cf.* Figures 5.9a, 5.12a and 5.14a) frames, is numerically observed using both $\mathscr{L}^2$ and $\mathscr{L}^\infty$ norms. Moreover, the relative norm of error is systematically lower when using the velocity interpolation, no matter the technique used, whereas the spatial order of convergence is systematically higher. In $\mathscr{L}^2$ norm, an order of almost 2 is systematically reached for the velocity components when using linear interpolation, which is consistent with the theory. In $\mathscr{L}^\infty$ norm, the convergence order is similar except for the Cartesian components when using multi-directional interpolation with a tilt angle of 11 or 30°. In these cases, the convergence order is about 1, same value as the direction assignment technique. This may be explained by the fact that the directional approach interpolates purely along e_X axis whereas the multi-directional approach interpolates using values at different coordinates in the (e_X, e_Y) frame. The 45° case is peculiar because the immersed boundaries correspond to elements' diagonals. Therefore, the forced nodes which are shared by two elements with $\chi_e = 1$ (*cf.* $\mathbf{x}_2$ in Figure 5.10) are directly located on the immersed boundary (*i.e.* no need of interpolation). The forced nodes belonging only to one element with $\chi_e = 1$ (*cf.* $\mathbf{x}_1$) have three fluid nodes but with a regular pattern ($\mathbf{x}_{\nu_2}$ which in the prolongation of $\mathbf{x}_2$ along e_X axis and $\mathbf{x}_{\nu_1}$ and $\mathbf{x}_{\nu_3}$ are symmetrical with respect to $\mathbf{x}_{\nu_2} - \mathbf{x}_2$).

Concerning pressure, due to the appearance of checkerboard patterns induced by the instability of the pair $\mathbb{Q}_1$–$\mathbb{Q}_0$ (*cf.* Figure 5.15), conclusions about convergence order are not sure. Nevertheless, as the pressure is still a linear combination of a unique function and an element of the kernel of the gradient operator [GR86], the pressure gradient remains mainly unaffected. Indeed, it has almost the same behavior as the velocity components in both Cartesian (*cf.* Figures 5.8c, 5.8d, 5.11c, 5.11d, 5.13c and 5.13d) and channel (*cf.* Figures 5.9b, 5.12b and 5.14b) frames, provided that we consider its value far enough from the immersed boundary to avoid the effect of the non-physical pressure values computed in the elements crossed by the immersed

obstacle because of the factor ξ_e in the projection equation (3.51). Excluding the nodes which are close to the immersed boundary to compute the pressure gradient error values could explain why the computed order is higher than 2. Overall, we can say that the velocity interpolation technique reduces the error, enhances mesh convergence and improve the agreement with the analytical solutions even if, in some cases, the multi-directional approach can lead to higher local errors.

***Remark* 14** *Hybrid and directional interpolation provide exactly the same results because only plane obstacles are considered, as explained in Section 5.3.*

***Remark* 15** *The error values obtained with the directional and hybrid interpolation are the same because a fluid point is found for all nodes without having to increase c (cf. Section 4.2).*

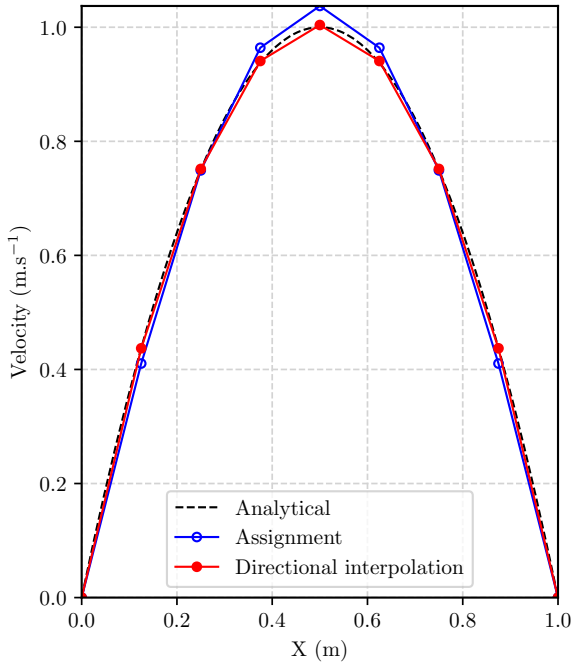


Figure 5.7. Velocity profile computed for the tilted Poiseuille flow ($\theta = 45°$, $Y = 2$ m) using both direct assignment and directional linear interpolation.

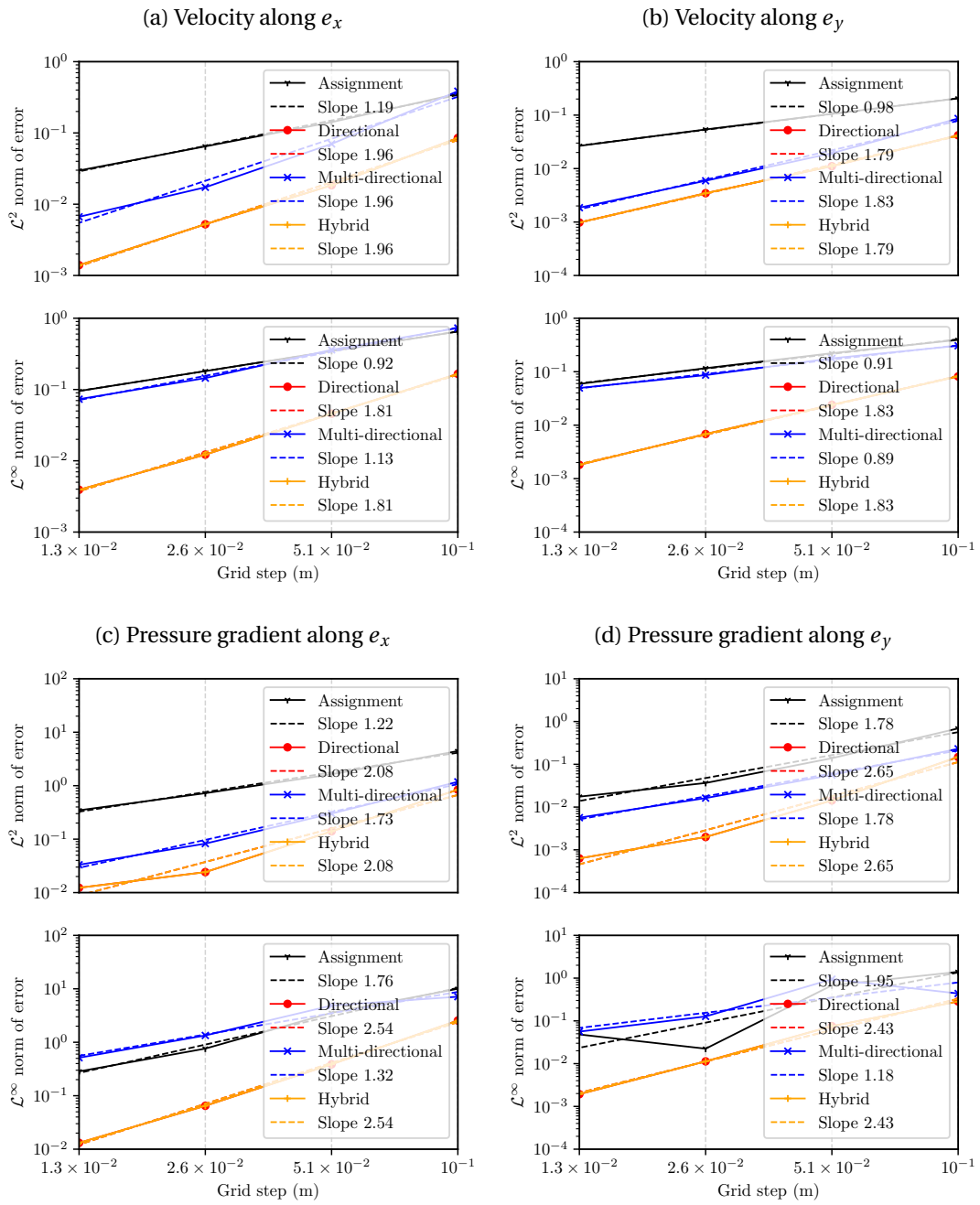


Figure 5.8. Poiseuille flow: evolution of relative norms ($\mathscr{L}^2$ and $\mathscr{L}^\infty$) of the error related to the component of the velocity and pressure gradient with respect to the value of the grid step. Those results are obtained with tilt angle of 11°.

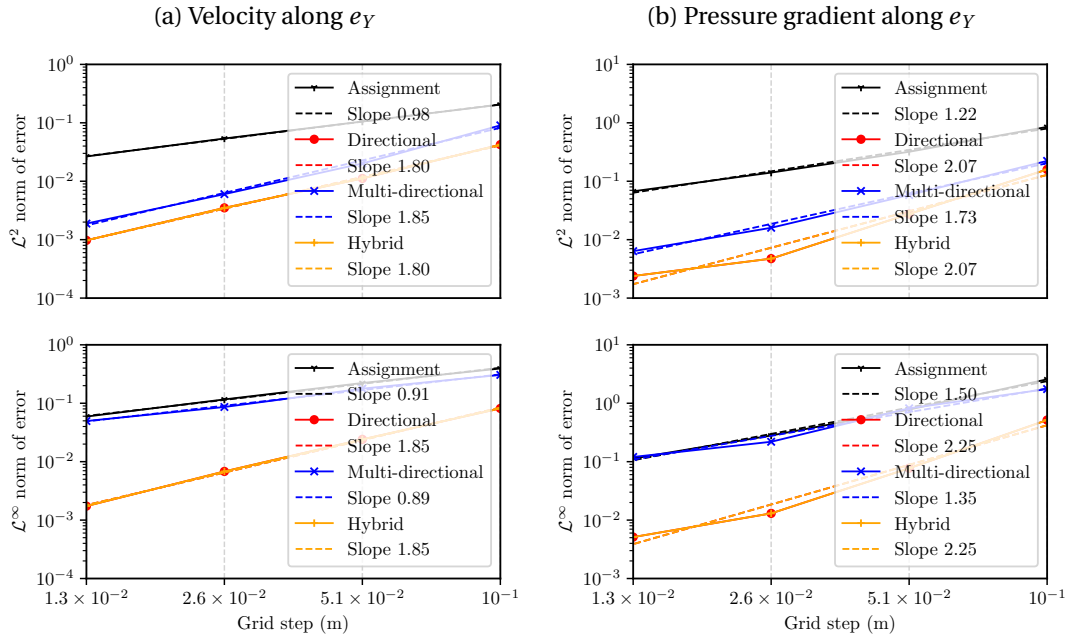


Figure 5.9. Poiseuille flow: evolution of relative norms ($\mathscr{L}^2$ and $\mathscr{L}^\infty$) of the error related to the component of the velocity along e_Y and the pressure gradient along e_Y with respect to the value of the grid step. Those results are obtained with tilt angle of 11°.

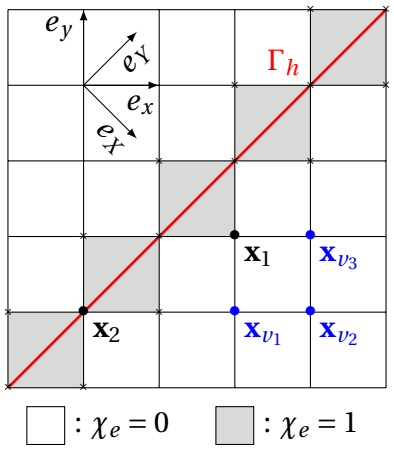


Figure 5.10. Poiseuille flow: schematic representation of the computational grid with a 45° angle.

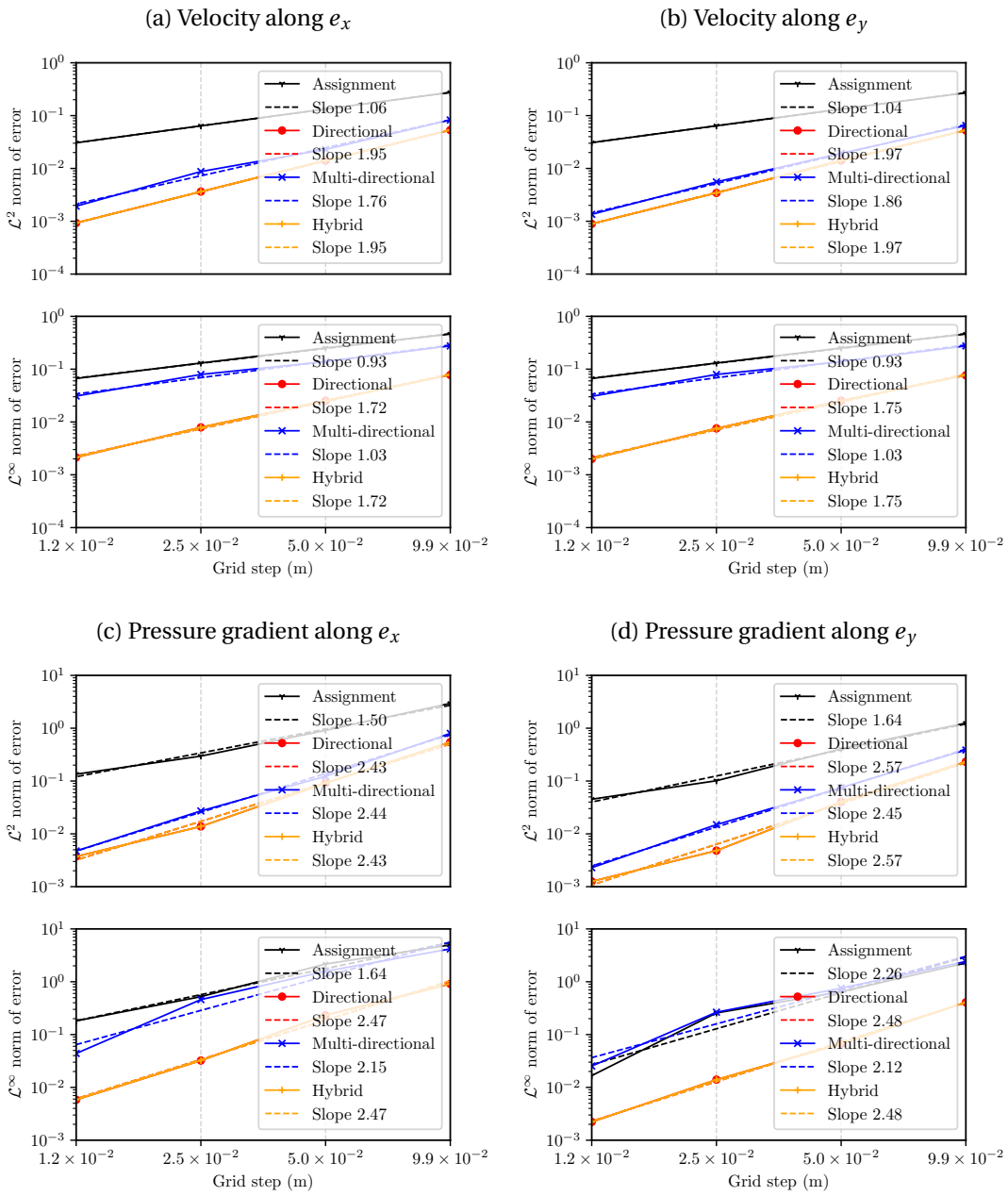


Figure 5.11. Poiseuille flow: evolution of relative norms ($\mathscr{L}^2$ and $\mathscr{L}^\infty$) of the error related to the component of the velocity and pressure gradient with respect to the value of the grid step. Those results are obtained with tilt angle of 30°.

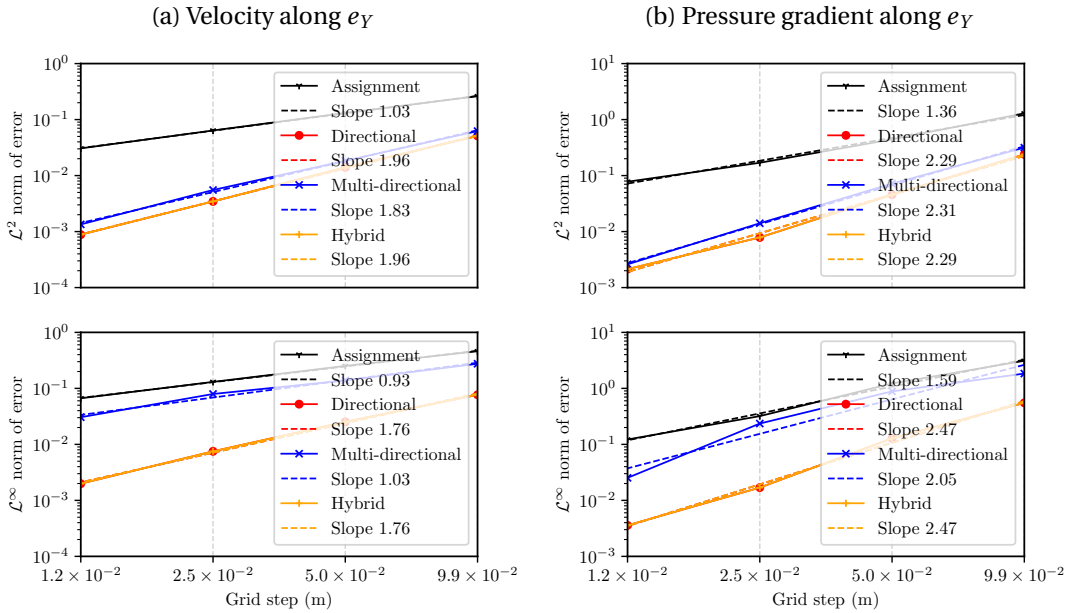


Figure 5.12. Poiseuille flow: evolution of relative norms ($\mathscr{L}^2$ and $\mathscr{L}^\infty$) of the error related to the component of the velocity along e_Y and the pressure gradient along e_Y with respect to the value of the grid step. Those results are obtained with tilt angle of 30°.

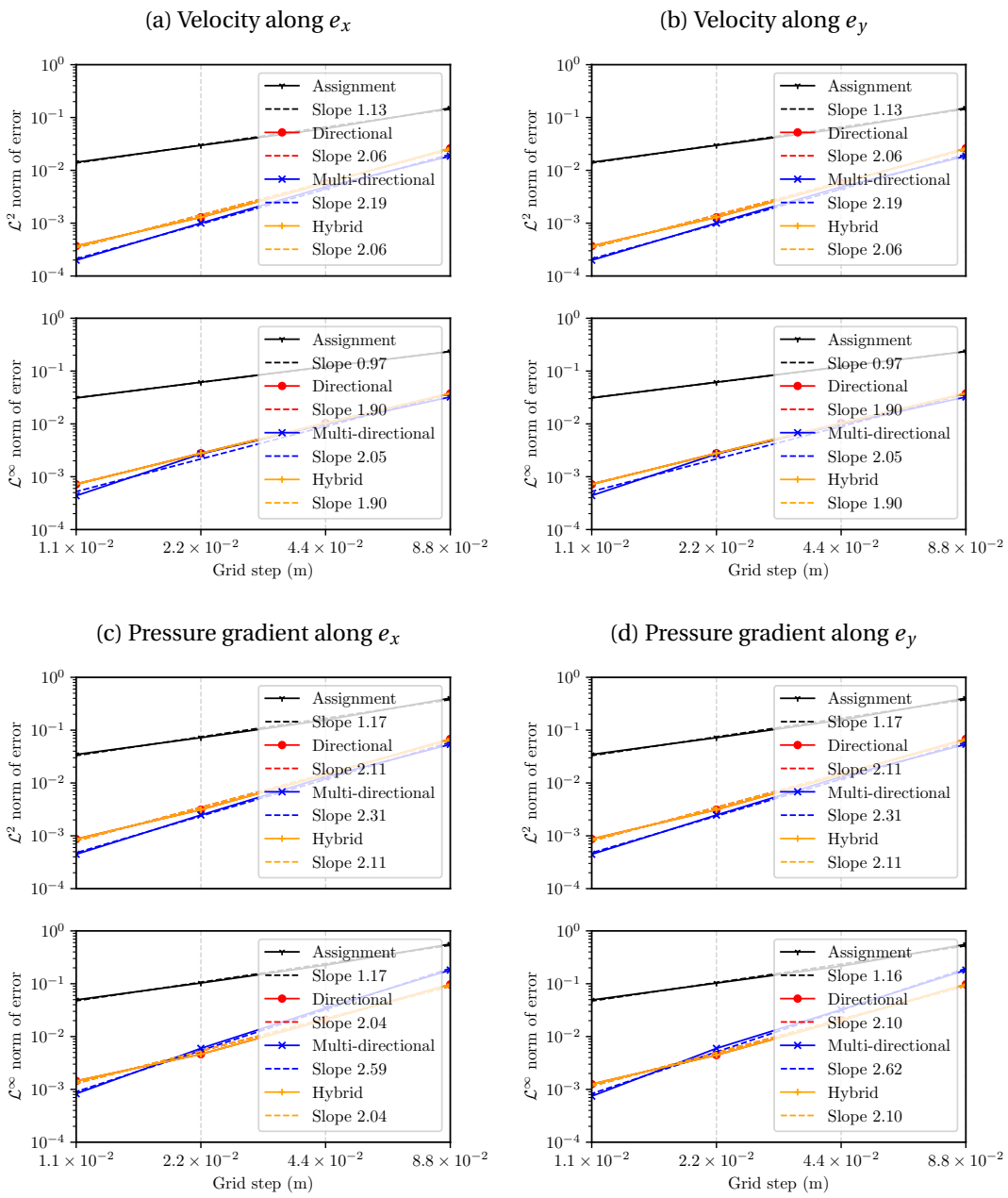


Figure 5.13. Poiseuille flow: evolution of relative norms ($\mathscr{L}^2$ and $\mathscr{L}^\infty$) of the error related to the component of the velocity along e_x and e_y with respect to the value of the grid step. Those results are obtained with tilt angle of 45°.

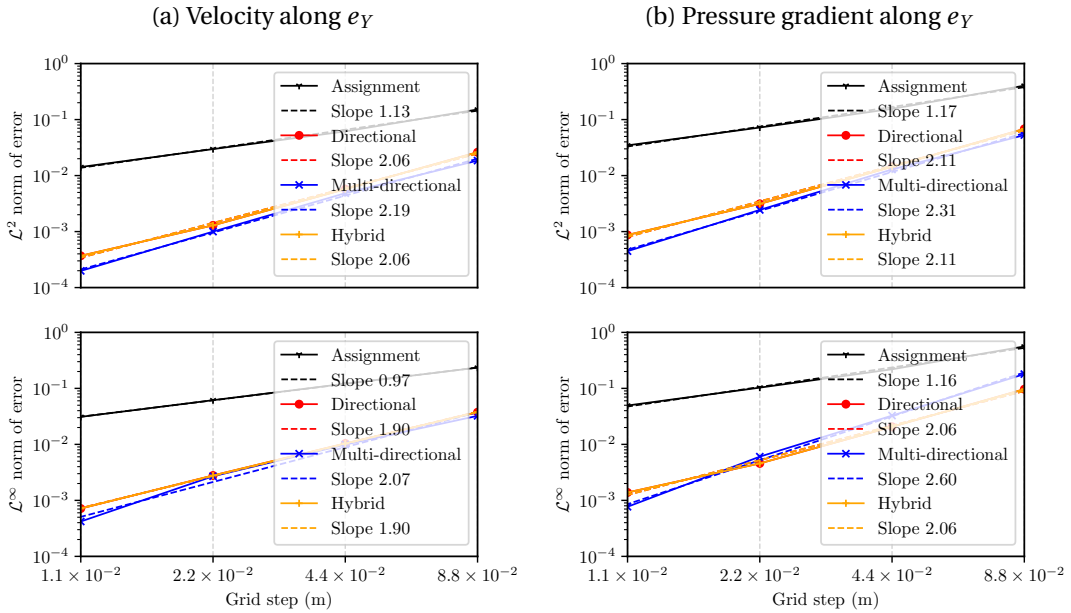


Figure 5.14. Poiseuille flow: evolution of relative norms ($\mathscr{L}^2$ and $\mathscr{L}^\infty$) of the error related to the component of the velocity along e_Y and the pressure gradient along e_Y with respect to the value of the grid step. Those results are obtained with tilt angle of 45°.

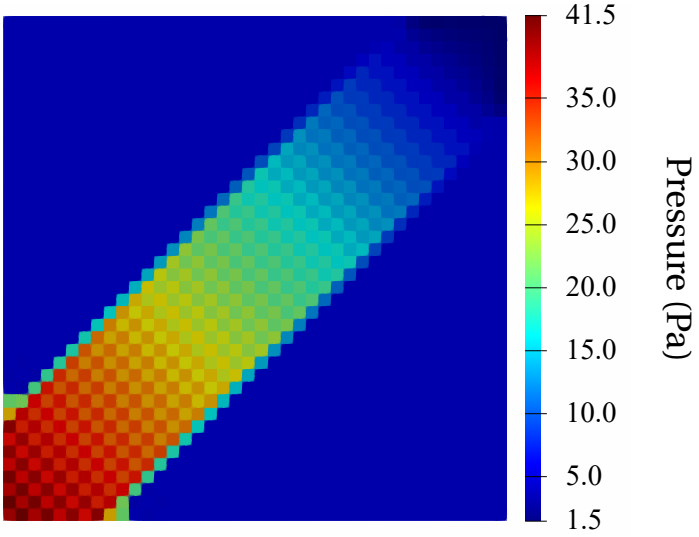


Figure 5.15. Example of checkerboard pressure patterns that can appear for the Poiseuille flow with a 45° tilt angle.

5.4.2. Taylor-Couette flow

The laminar Taylor-Couette case, introduced in Section 5.2.3, was also used to carry out a mesh convergence study. Figure 5.16a shows that, overall, the directional interpolation qualitatively enhances the agreement with the analytical solution. In addition, the spatial convergence is numerically assessed in Figure 5.16b. The order of convergence, when using interpolation, is not as close to 2 as the one computed for the tilted Poiseuille flow (between 1.7 and 1.8 in $\mathscr{L}^2$ norm and about 1.5 in $\mathscr{L}^\infty$ norm); this could be explained by the geometrical complexity of the immersed boundary (*i.e.* even if smooth, a circle is more complex to model than a plane). However, the different interpolation techniques still greatly enhance the results (a factor 10^{-1} applied on the $\mathscr{L}^2$ norm). It can be noted that the multi-directional interpolation seems more sensitive to the mesh refinement as it can produce higher local errors when the space resolution is low. Once again, the hybrid and directional interpolation provide the same results. This could be explained by the fact that, similarly to plane geometries, regular geometries does not produce (or, at most, only a few) nodes for which no interpolation point is found.

(a) Azimuthal velocity profile

(b) Azimuthal velocity convergence

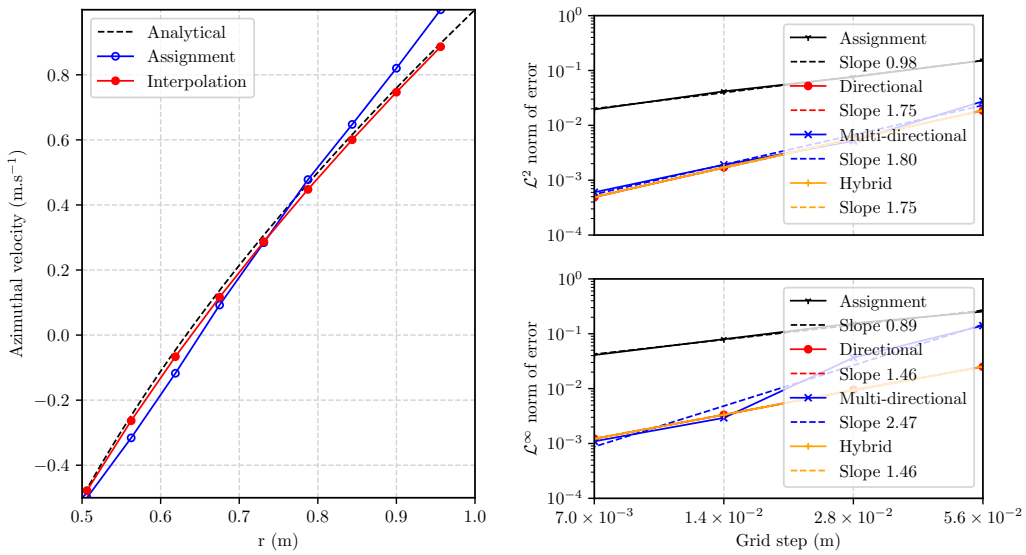


Figure 5.16. Taylor-Couette flow: evolution of azimuthal velocity along the radius (directional interpolation only) and evolution of the relative $\mathscr{L}^2$ and $\mathscr{L}^\infty$ norms of error of the azimuthal velocity with respect to the grid step.

5.4.3. Steady flow around a circular cylinder

Then, another mesh convergence study was carried out for the steady laminar flow around a cylinder at $Re = 20$, introduced in Section 5.2.4, in both static and rotating configurations. As there is no analytical solution for the Navier-Stokes equations in this case, the simulation results obtained using the finest grid are considered as a reference. Moreover, as we are only concerned with convergence aspects in this study, we can consider a small domain ($10d \times 10d$) without loss of generality. When looking at the mesh convergence curves, in $\mathscr{L}^2$ and $\mathscr{L}^\infty$ norm, for both static (*cf.* Figure 5.17) and rotating (*cf.* Figure 5.18) cylinders, the conclusion is the same as the one formulated for the Poiseuille and Taylor-Couette flows: the linear interpolation of the velocity in the vicinity of the obstacles reduces the error between the computed and reference solutions while increasing the rate of convergence. However, the order of convergence is slightly lower than two in $\mathscr{L}^2$ norm when using the linear interpolation (in [1.7, 1.9]) and slightly larger than 1 when using only direct assignment (1.3); this could be linked to the lack of analytical solution. This difference in convergence order is also noticed in the $\mathscr{L}^\infty$ norm: in [1.3, 1.5] when using the linear interpolation and in [0.7, 0.9] when using the direct assignment of the velocity. Furthermore, we can say that the multi-directional approach seems to provide slightly superior convergence rate but, overall, results are similar.

5.4.4. Unsteady flow around a circular cylinder

To study convergence at $Re = 100$ (unsteady flow regime), as we use an adaptative time step, which depends on the grid step (*i.e.* the solutions of unsteady cases desynchronize when the mesh changes), only the mean value of the drag coefficient is considered. The results obtained for the different interpolation techniques without using the BTD scheme are summarized in Figure 5.19. The difference in convergence order between the different linear interpolation techniques and the direct assignment is clearly noticeable in both rotating and static configurations – the values for the direct assignment are, like in the steady cases, overestimated: about 1.2 instead of 1 in theory – even if the values are systematically inferior to 1.8, except for the multi-directional interpolation in the rotating case, which reach a computed convergence order of 2.1). Overall, the multi-directional interpolation seems to provide the best space convergence behavior for the cylinder case in unsteady regime.

***Remark* 16** *In all four configurations, the directional and hybrid approach provide the same results, which tends to confirm the explanation presented in Section 5.4.2.*

***Remark* 17** *In fact, as expected (cf. Section 4.1), the convergence order values are not only affected by the method chosen to compute the imposed velocity in the vicinity of the immersed boundary, but also by the FE discretization scheme itself. For instance, if we change the BTD method, used to numerically stabilize the convection term, for the upwind scheme (which is more diffusive), the space convergence order values can*

be relatively different as displayed in Figure 5.20. Indeed, in the static configuration, the computed space order is not higher than 1.2, even when using linear interpolation. The linear interpolation techniques enhance the space convergence only in the rotating case, nonetheless the values of the order [1.6,1.9] are still inferior to the one obtained with a less diffusive convection scheme [1.7,2.1] (cf. Figure 5.19b).

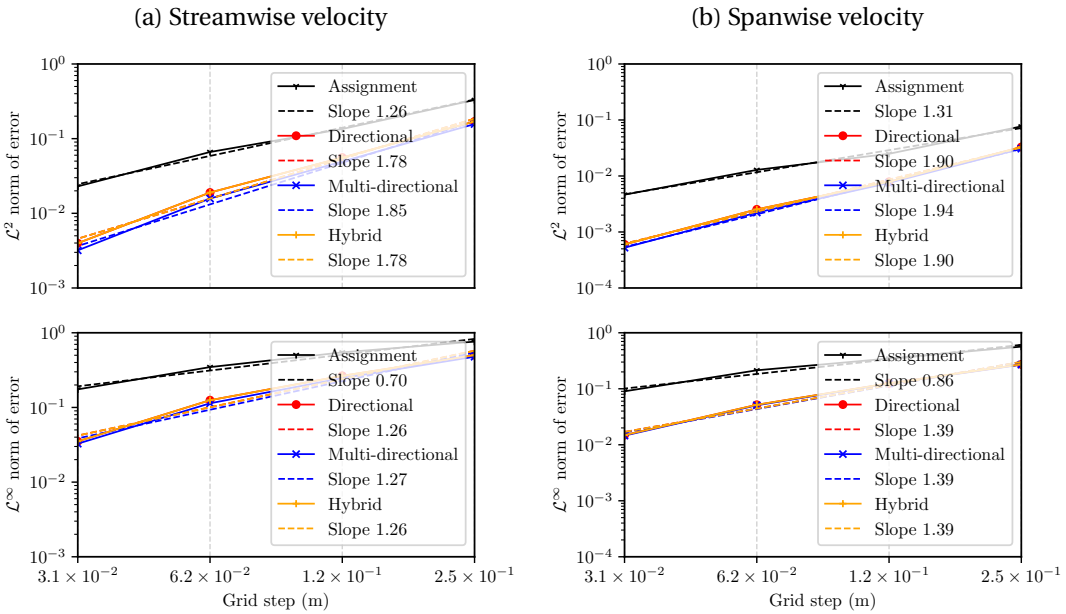


Figure 5.17. Mesh convergence ($\mathscr{L}^2$ and $\mathscr{L}^\infty$) of the velocity for the steady flow around a static circular cylinder ($Re = 20$).

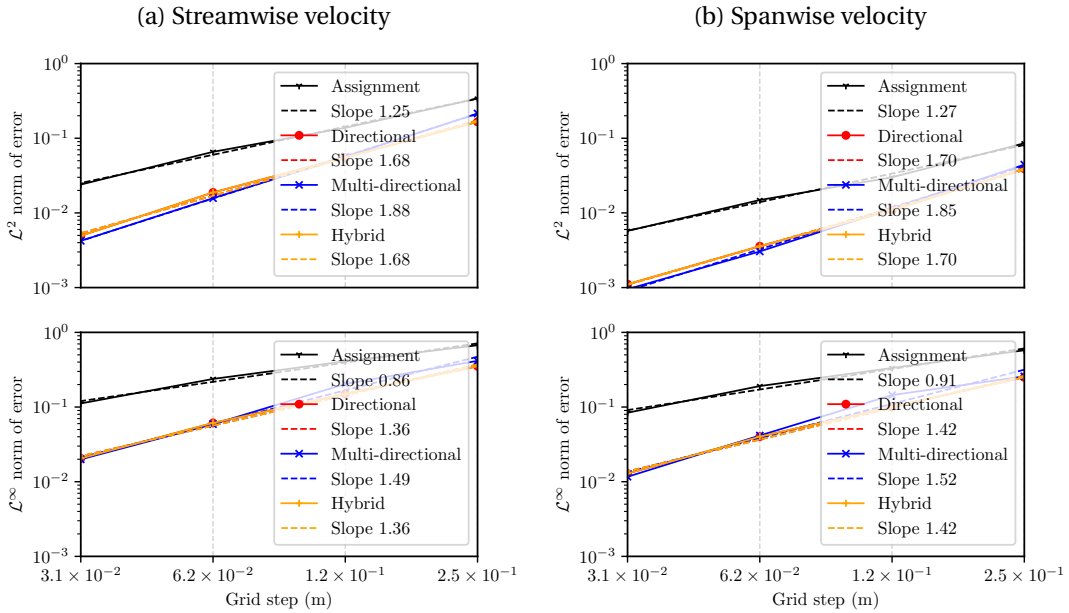


Figure 5.18. Mesh convergence ($\mathscr{L}^2$ and $\mathscr{L}^\infty$) of the velocity for the steady flow around a rotating circular cylinder ($Re = 20$).

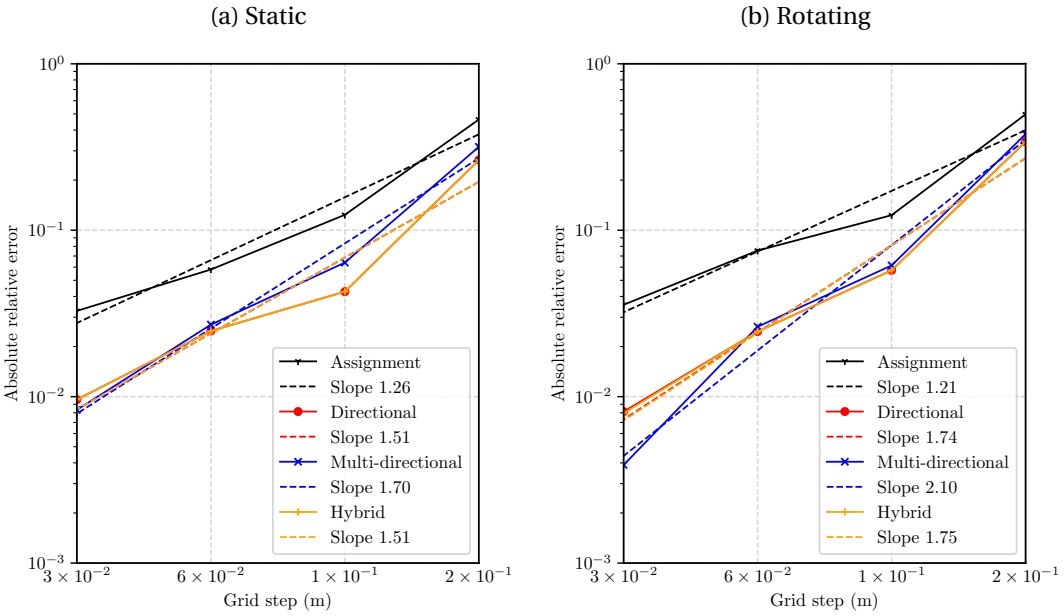


Figure 5.19. Mesh convergence of the mean value of the drag coefficient for the unsteady flow around a circular cylinder ($Re = 100$) using the BTD scheme for the convection term.

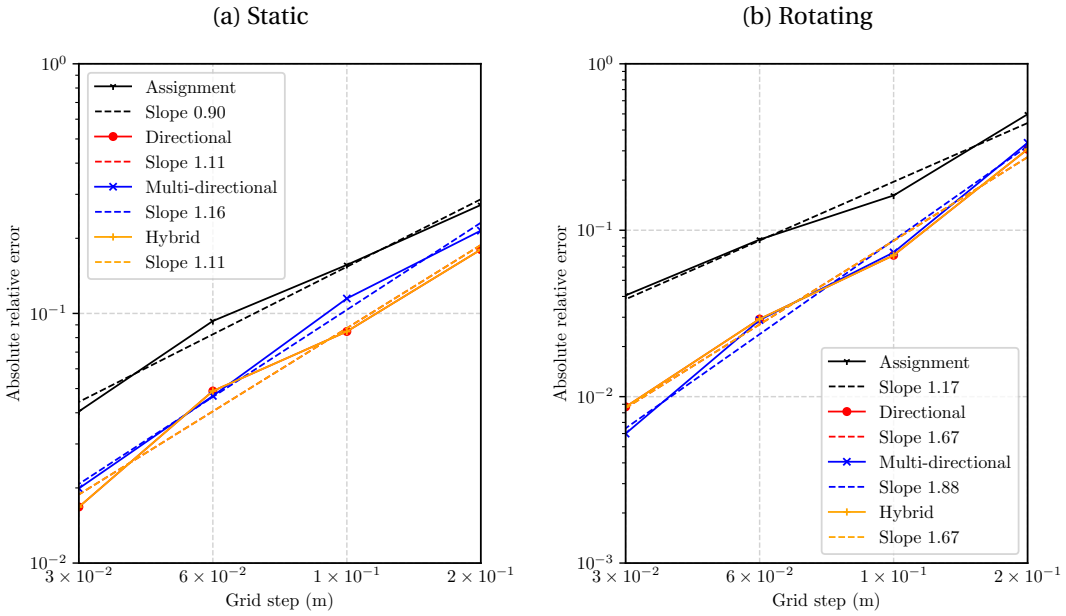


Figure 5.20. Mesh convergence of the mean value of the drag coefficient for the unsteady flow around a circular cylinder ($Re = 100$) without using a highly diffusive upwind scheme for the convection term.

5.5. Global quantities studies

After penalty parameter and grid step convergence, we studied the values of several macroscopic quantities (mainly aerodynamic coefficients and dimensionless numbers) over different test cases.

5.5.1. Flow around a circular cylinder

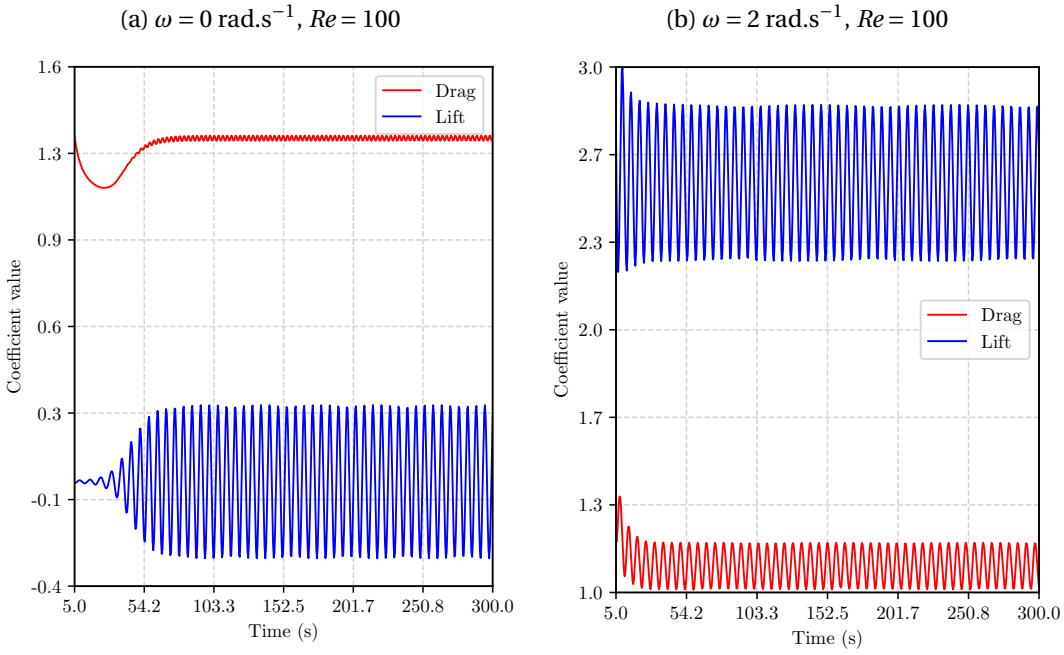


Figure 5.21. Time evolution of the drag and lift coefficients in both static and rotating configurations at $Re = 100$ (directional interpolation).

First, we considered the flow around a cylinder, on a much larger domain ($120r \times 120r$) to minimize potential boundary effects. Appendix B shows streamlines, vorticity and pressure contours for $Re \in \{20, 100\}$ and for both static and rotating configurations. Globally, the shapes are in good agreement with the literature, even if the Balancing Tensor Diffusivity (BTD) scheme – used to deal with the advective term – induces an extra numerical diffusion [Gre+84]. Global quantities such as drag and lift coefficients (respectively C_d and C_l with, in unsteady cases, their mean values denoted with a bar and their fluctuations denoted with an apostrophe), the angle between the aerodynamic force and the horizontal axis (α) and the Strouhal number (St) have been computed and compiled in Tables 5.6, 5.7, 5.8, 5.9 and 5.10. The time evolution of the aerodynamic coefficients is shown in Figure 5.21, highlighting their periodic oscillations in unsteady regimes. Concerning the numerical stabilization of the convective term, some results are obtained with the BTD scheme and some other with the upwind scheme (*cf.* Remark 6). As a whole, we can see that the obtained results are in very good agreement with the reference values, in both steady (*i.e.* $Re = 20$) and (*i.e.* $Re = 100$) unsteady cases. Moreover, linear interpolation techniques seems to provide values that are closer to the reference ones – regarding hybrid interpolation, Remark 16 still applies.

Table 5.6. Aerodynamic coefficients computed for the steady laminar flow around a static circular cylinder ($Re = 20$) using direct assignment (label "A") and linear interpolation techniques ("B": directional, "C": multi-directional, "D" : hybrid).

		C_d				L_w			
		A	B	C	D	A	B	C	D
	BTD	2.154	2.075	–	–	1.166	0.901	–	–
	Upwind	2.129	2.083	2.083	2.083	1.045	0.915	0.917	0.915
References	[IBF14]	2.059	–	2.054	–	0.925	–	0.900	–
	[Ye+99]	2.03				0.92			
	[Cho+07]	2.02				0.90			
	[TC07]	2.06				0.94			
	[LF05]	2.06				0.93			
	[For80]	2.00				0.91			
	[Tri59]	2.09				–			

Table 5.7. Aerodynamic coefficients computed for the steady laminar flow around a rotating circular cylinder ($Re = 20$) using direct assignment (label "A") and linear interpolation techniques ("B": directional, "C": multi-directional, "D" : hybrid).

				References				
		BTD	Upwind	[IBF14]	[Chu06]	[SBD02]	[IT90]	[BDY89]
C_d	A	1.984	2.041	1.8608	1.888	1.85	1.925	2.000
	B	1.913	2.006	–				
	C	–	2.006	1.8679				
	D	–	2.006	–				
C_l	A	3.032	2.602	2.9419	2.629	2.75	2.617	2.740
	B	2.868	2.507	–				
	C	–	2.507	2.7745				
	D	–	2.507	–				
$\alpha(°)$	A	57.11	51.87	57.68	54.31	56	53.66	53.87
	B	56.02	51.34	–				
	C	–	51.34	56.05				
	D	–	51.34	–				

Table 5.8. Mean value and fluctuations of the drag coefficient computed for the unsteady laminar flow around a static circular cylinder ($Re = 100$) using direct assignment (label "A") and linear interpolation techniques ("B": directional, "C": multi-directional, "D" : hybrid).

		$\overline{C_d}$				$C_d{}'$			
		A	B	C	D	A	B	C	D
Present work		1.363	1.325	1.321	1.325	0.012	0.011	0.011	0.011
References	[IBF14]	1.347				0.009			
	[SBD02]	1.337				0.009			
	[LF05]	1.340				0.009			
	[Cho+07]	1.340				0.011			
	[CLS10]	1.350				0.012			
	[CB10]	1.317				0.009			
	[JMW12]	1.376				0.010			

Table 5.9. Lift coefficient fluctuations and Strouhal number computed for the unsteady laminar flow around a static circular cylinder ($Re = 100$) using direct assignment (label "A") and linear interpolation techniques ("B": directional, "C": multi-directional, "D" : hybrid).

		$C_l{}'$				St			
		A	B	C	D	A	B	C	D
Present work		0.343	0.295	0.294	0.295	0.161	0.165	0.165	0.165
References	[IBF14]	0.326				0.165			
	[SBD02]	0.326				0.165			
	[LF05]	0.333				0.166			
	[Cho+07]	0.315				0.164			
	[CLS10]	0.303				0.167			
	[CB10]	0.349				0.170			
	[JMW12]	0.339				0.170			
	[Nor03]	0.227				0.164			

Table 5.10. Aerodynamic coefficients related to the unsteady laminar flow around a rotating circular cylinder (Re = 100) using direct assignment (label "A") and linear interpolation techniques ("B": directional, "C": multi-directional, "D" : hybrid).

	Present work		References			
			[IBF14]	[SBD02]	[Chu06]	[KCL99]
$\overline{C_d}$	A	1.138	1.12	1.1080	1.1890	1.0979
	B	1.102				
	C	1.098				
	D	1.102				
$C_d{}'$	A	0.105	0.11	0.0986	0.1195	0.0988
	B	0.091				
	C	0.090				
	D	0.091				
$\overline{C_l}$	A	2.641	2.51	2.5040	2.4050	2.4833
	B	2.562				
	C	2.560				
	D	2.562				
$C_l{}'$	A	0.369	0.37	0.3616	0.4427	0.3603
	B	0.301				
	C	0.297				
	D	0.301				
St	A	0.163	0.165	0.1658	0.1732	0.1650
	B	0.166				
	C	0.166				
	D	0.166				

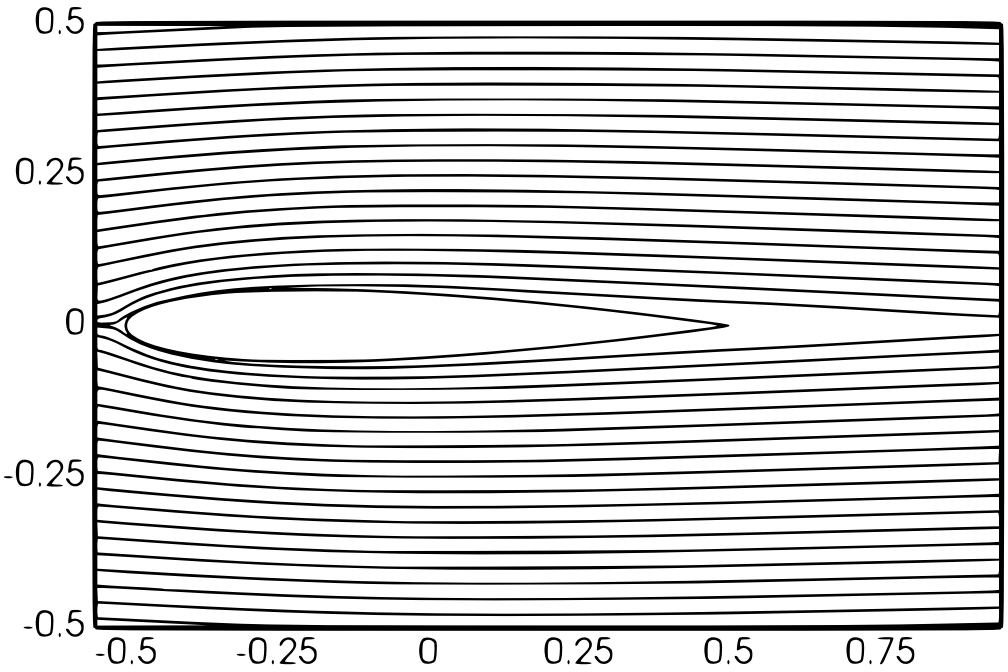
5.5.2. Flow past a NACA0012 airfoil

Second, we successfully ran simulations for flow past a NACA0012 airfoil in all the test configurations detailed in table 5.5 of Section 5.2.5. Figures 5.22 and 5.23 give a representation of the streamlines, velocity magnitude and pressure computed with an angle of attack of 0°. Shapes and recirculation length are very similar to the one presented in [SL16] (about 0.17 m in both cases). Concerning aerodynamic coefficients, we gathered the steady regime results obtained for angles of attack of 0 and 1° in table 5.11. Taking into account that an angle difference of 1° cannot be exactly reproduced using our immersed boundary geometrical model, we conclude that the aerodynamic coefficients are in a good agreement. However, for cases involving an angle of attack superior to 1°, our method was not able to recover steady state solutions so only the mean values of the aerodynamic coefficients are shown in Table 5.11. This can be induced by several phenomena such as:

- Residuals being too large because of high convergence criteria (as suggested by R.C. SWANSON),
- Boundary effects (the size of the used domain is smaller than the one used for the circular cylinder cases),
- Grid being too coarse and not adapted to the problem (*i.e.* staircase description of the obstacle which could decrease the transition between steady / unsteady regimes in terms of Reynolds number),
- The use of super-convergence algorithm in the work of R.C. SWANSON.

As shown in Table 5.11, the use of upwind scheme (which is more diffusive than the BTD scheme) for the stabilization of the convection term leads to results that are still in good agreement with the reference results. Once again, the different interpolation techniques provide very similar behaviors so it is not clear if the hybrid interpolation provides an advantage or not, even if the geometry present an irregularity (trailing edge).

(a) Streamlines



(b) Velocity magnitude

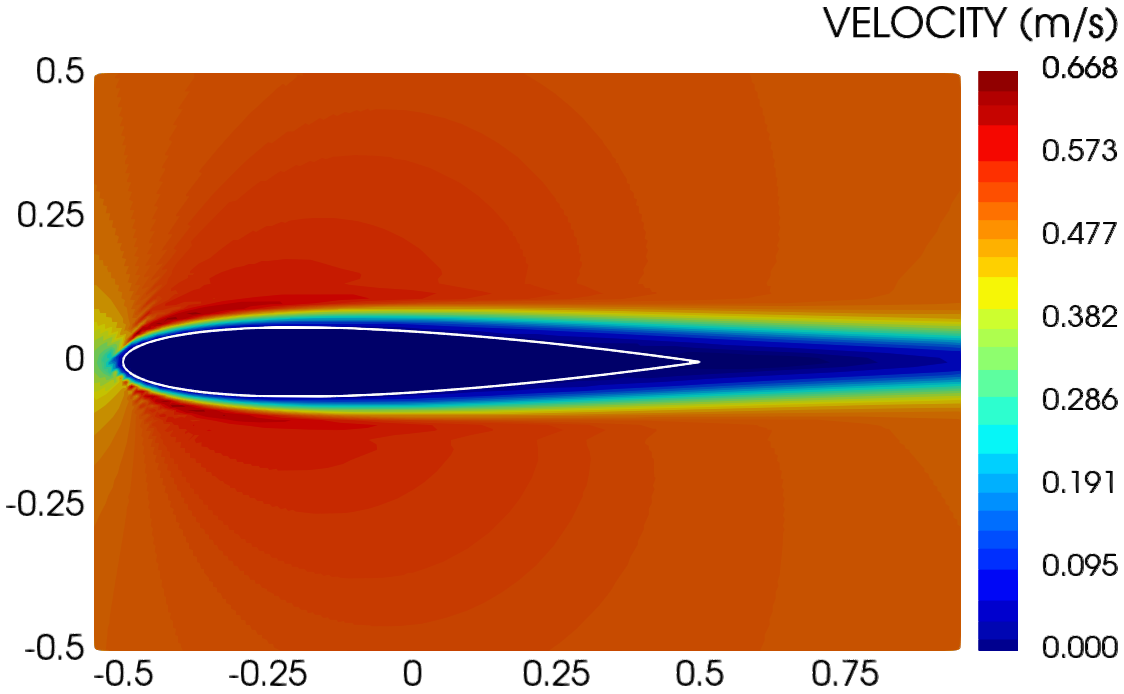


Figure 5.22. Computed velocity magnitude map and streamlines in the case of the flow past a NACA0012 airfoil with an angle of attack of 0°.

Table 5.11. Aerodynamic coefficients related to the laminar flow past a NACA airfoil using direct assignment (label "A") and linear interpolation techniques ("B": directional, "C": multi-directional, "D" : hybrid)

		Upwind	BTD		Reference [SL16]	
	θ	0	0	1	0	1
$\overline{C_d}$	A	0.0590	–	–	0.0555	0.0559
	B	0.0596	0.0563	0.0571		
	C	0.0596	–	–		
	D	0.0596	–	–		
$\overline{C_l}$	A	0.0022	–	–	0.0000	0.0184
	B	0.0021	0.0094	0.0092		
	C	0.0021	–	–		
	D	0.0021	–	–		

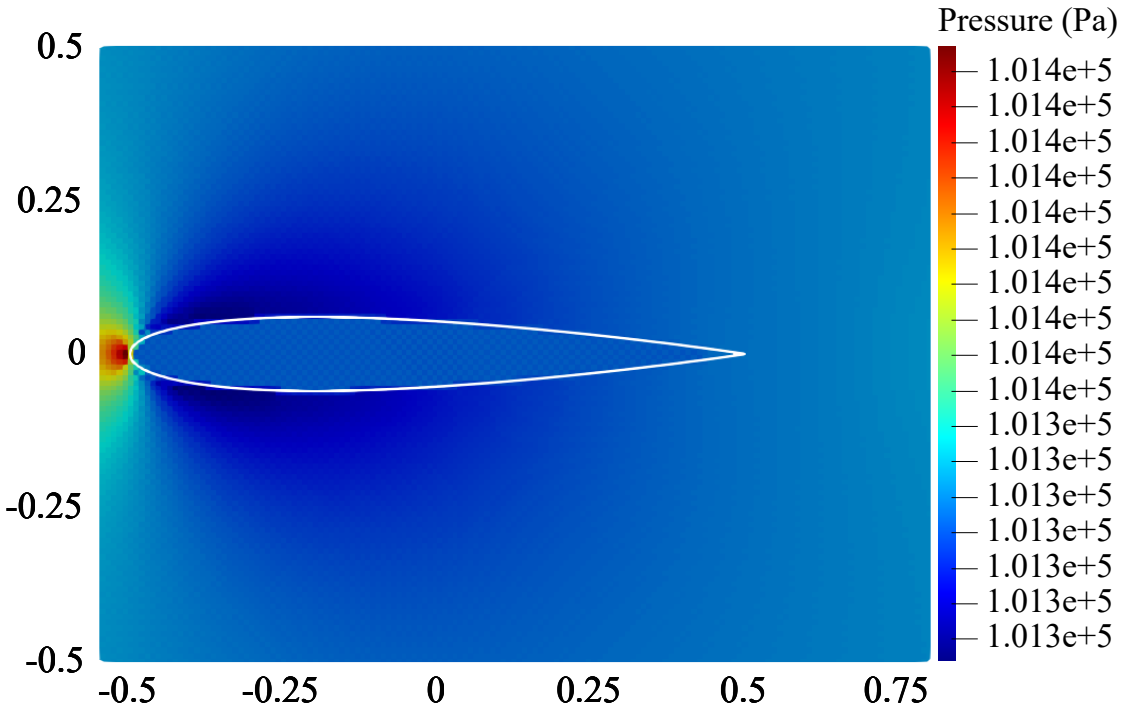


Figure 5.23. Pressure map of the flow past a NACA0012 airfoil with directional interpolation.

5.6. An industrial case

Finally, we successfully ran simulations of the 3D flow through immersed boundaries representative of the flow limiter (device presented in Section 1.2 and case described in Section 5.2.6) in 3D which demonstrates the generality and robustness of our method. As shown in Figures 5.24 and 5.25b, the expected behavior (*i.e.* the creation of a vortex to dissipate kinetic energy via turbulence) is respected. However, for this relatively coarse discretization, we can note the appearance of a checkerboard pressure pattern which is linked to the instability of the pair $\mathbb{Q}_1/\mathbb{Q}_0$ (*cf.* Section 3.1.3.2). Fortunately, the effect of this phenomenon on the velocity is moderate because only the pressure gradient is considered in the prediction equation. In addition, the gap between the pressure extrema of this chessboard pattern tends to decrease when the space step decreases. Still, in order to soften this phenomenon even more, we could change the mesh configuration. Indeed, as can be seen on Figure 5.5b in Section 5.2.6, the maximum density of cells is located in the cold leg (or vessel inlet) and the density of cells in the vicinity of the fins is rather low. This might not be very appropriate and the idea of increasing the density of the cells at the vicinity of the immersed boundaries while reducing it in the cold leg (in order to preserve the same number of cells and so a similar computation time) could be investigated.

We also used the results of those simulations to carry out a mesh convergence study, with the solution computed using finest grid considered as reference. We focused on the evolution of the head loss coefficient K over the different grids. Taking the normal mixture mass flux exited via the cold leg $\rho_{bc}\mathbf{u}_{bc}\cdot\mathbf{n}_{bc}$ as a reference, we define K as follows:

$$K = 2\frac{\langle p_i\rangle - \langle p_o\rangle}{\langle \rho_{bc}\mathbf{u}_{bc}\cdot\mathbf{n}_{bc}\rangle \langle \mathbf{u}_{bc}\cdot\mathbf{n}_{bc}\rangle}$$

where the symbol $\langle . \rangle$ denotes an area average and $\langle p_i\rangle$ (resp. $\langle p_o\rangle$) the area averaged inlet (resp. outlet) pressure. Furthermore, as the grids are not uniform (*cf.* Figure 5.5), we define an average space step $h = N^{-\frac{1}{3}}$ with N the number of cells of the considered grid. The results, obtained are gathered in Table 5.12. The values are consistent with the preliminary studies [Bel18]. Moreover, those results show that the directional interpolation tends to enhance the spatial convergence (*cf.* Figure 5.25a).

Moreover, on one hand, we can see that the values obtained with the directional interpolation are closer to the one obtained on the finest grid – this is encouraging for future turbulent wall law interpolation. On the other hand, the results obtained using the hybrid interpolation, even if they are not exactly the same as the ones obtained with the directional interpolation in the case of this complex 3D geometry, do not show any significant improvement either. This could be explained by the similarity between the directional and multi-directional approaches (they both use linear interpolation in the end) as well as the rather limited number of nodes for which no purely fluid node is found: the value of $\mathbf{u}_\Gamma$ slightly differs for only a few nodes, leading to a limited effect in the computation of global quantities such as the head loss coefficient.

Table 5.12. Values of the head loss coefficient K, related to the flow limiter case, computed with the PDF method ("A": direct assignment, "B": directional interpolation, "C": multi-directional, "D": hybrid).

		K			
N	h (m)	A	B	C	D
6,080	5.48×10^{-2}	28.3	17.7	43.5	17.7
48,640	2.74×10^{-2}	13.8	14.6	15.8	14.4
164,160	1.83×10^{-2}	6.6	5.78	6.76	5.75
794,880	1.08×10^{-2}	5.7	5.45	5.65	5.48
3,594,240	6.53×10^{-3}	5.5	5.4	–	–

What is also interesting to note is the threshold effect between 48,640 and 164,160 elements, clearly visible in Figure 5.25a. Indeed, with less than 164,160 elements, the space resolution is not sufficient to model the channels between the fins properly. This means that, as there are too few elements in those areas, the method is unable to find purely fluid nodes (resp. points located in a purely fluid elements) to achieve the multi-directional (resp. directional) interpolation. Hence, the direct assignment is used and, as the imposed velocity is null, the channels look obstructed with a grid of less than 164,160 elements (*cf.* Figure 5.26). These obstructed channels lead to an overestimate of the pressure loss coefficient.

In addition, thanks to C. BOURCIER, some body-fitted simulations (with about 240,000 elements) have been carried out (*cf.* Figure 5.27b). The value of K obtained with the FE element discretization (resp. Volume Finite Element discretization) is 5.40 (resp. 4.61). Those values are very close to the one obtained with the PDF method. However, the body-fitted meshes are way more difficult to construct (especially if a boundary layer of a given thickness is needed) and possess highly distorted elements (*cf.* elements located at the extremities of the fins in Figure 5.27a). This highlights the interest of our approach, especially for topological shape optimization.

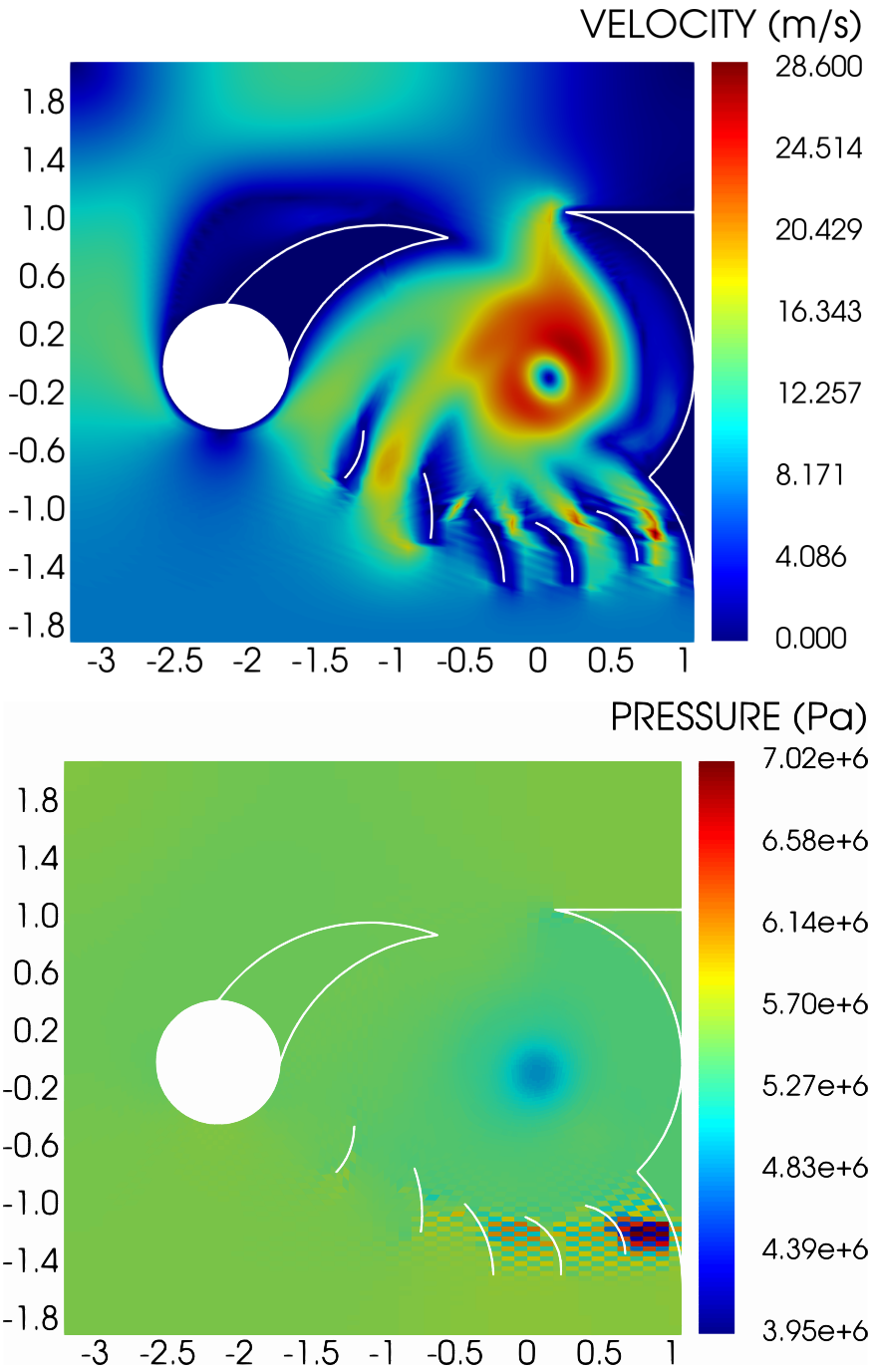


Figure 5.24. Pressure and velocity maps computed in the case involving a flow limiter with a grid composed of 164160 elements.

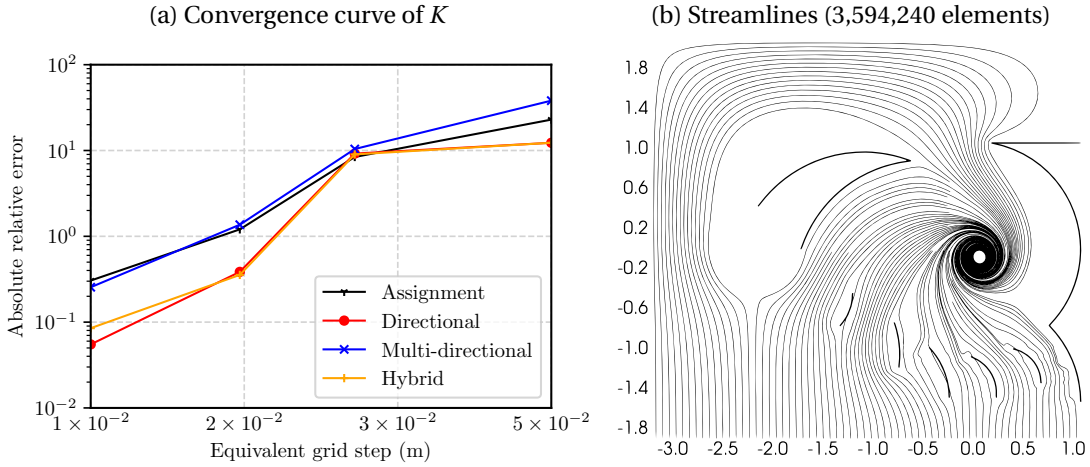


Figure 5.25. Qualitative and convergence results related to the flow limiter case.

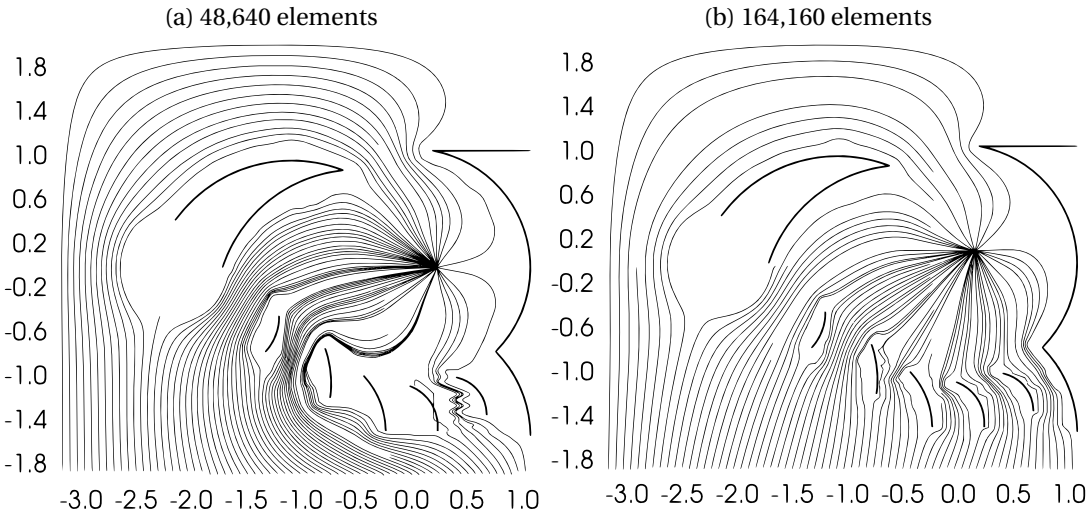


Figure 5.26. Streamlines computed with two different grids. Some channels between the fins are obstructed on the coarser grid (left) whereas it is not the case on the finer one (right).

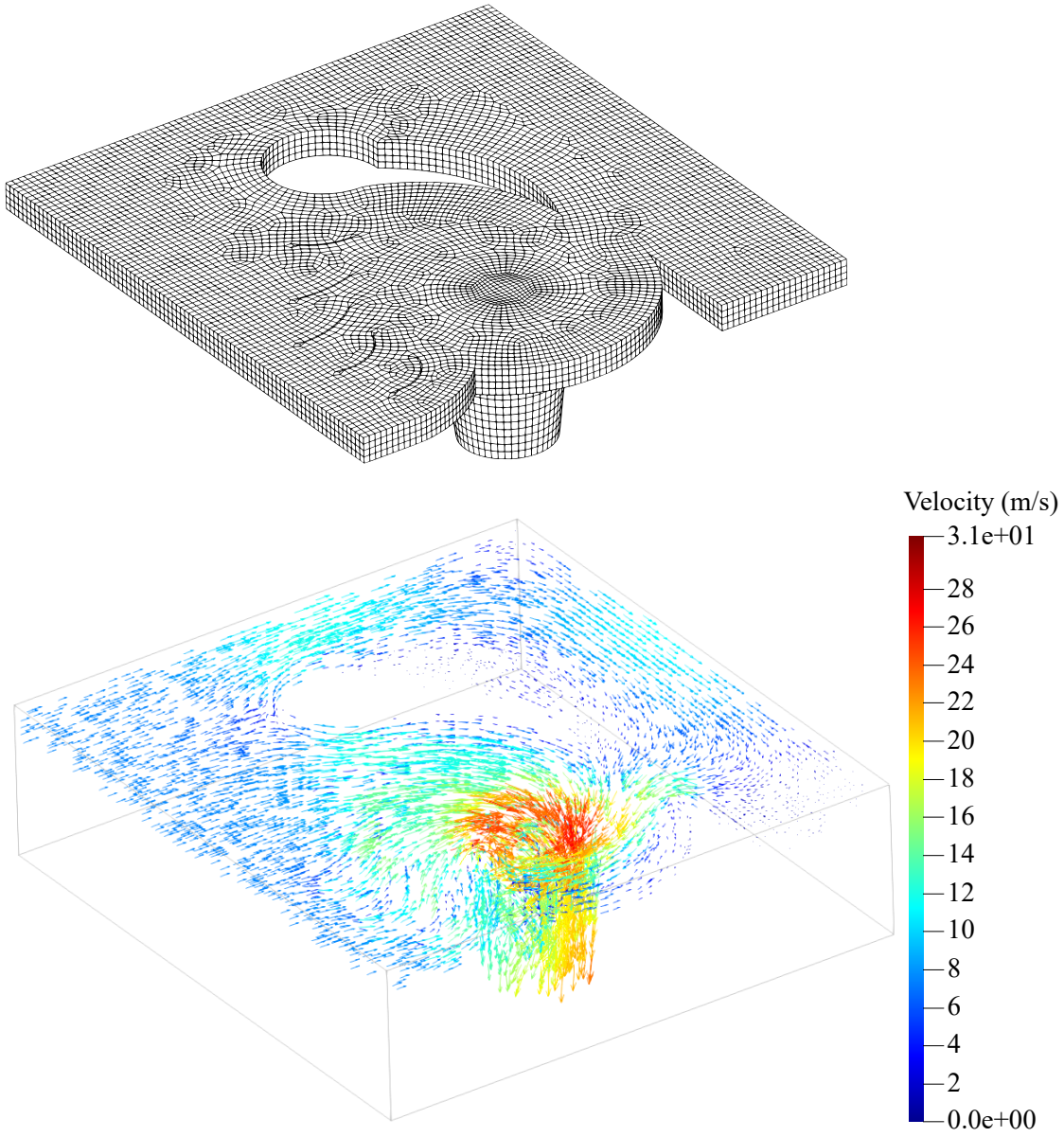


Figure 5.27. Body-fitted hexahedral mesh of the flow limiter and velocity field computer with GENEPI3.

Conclusion

In the introduction of this document (*cf.* Chapter 1), the context of my PhD project was introduced. The CEA is interested in modeling innovative nuclear passive safety systems (for second and third generation of nuclear reactors) involving thin complex geometries. New simulation needs arose and so the goal of my PhD thesis was to develop a tool capable of modeling such devices. As we chose a fictitious domain approach to ensure low computation times, generalities and theoretical aspects of this type of methods were discussed with a focus on IBM.

The second chapter deals with the issue of input geometrical data. Indeed, as the geometry is not directly contained in the computation grid, methods are needed to compute geometrical data which are necessary to run IBM simulation. Moreover, additional data (such normal vector, normal projection, *etc.*) are needed to linearly interpolate field variables in the vicitinity of the immersed obstacles in order to reach order 2 in space. Two different approaches, each including several variants, are considered: weighting and optimization. Their algorithms are detailed and results, as well as performances, are compared over academic (cylinder) and quasi-industrial (NACA0012 airfoil) test cases. It turns out that the error values obtained are similar but the weighting approach offers way lower computation times.

On one hand, Chapter 3 introduces the problem of interest and its governing equations (dilatable Navier-Stokes). Numerical methods are also tackled: first time discretization (fractional step algoritm or projection scheme) and then space discretization (FEM). Finally, the theoretical aspects of the Penalized Direct Forcing, an IBM which inherits from penalty and Direct Forcing methods, are introduced. This method was initially developed and tested together with a space-time discretization based on a finite difference scheme. Here, we adapt the method to a Galerkin finite element discretization with a lumping of the mass matrix. On the other hand, the Chapter 4 focuses on reaching second order by linear interpolation. Two different techniques are presented: one involving interpolation in the direction normal to the immersed obstacle and the other involving multi-directional interpolation. A hybrid strategy is proposed too.

At last, Chapter 5 compile all simulations and discussions. A list of validation test cases is presented, as well as numerical results. The steady laminar Poiseuille and Taylor-Couette flows, for which analytical solutions are known, allowed us to carry out numerical convergence studies, with respect to the penalty parameter and the grid step. It is worth noting that all the different linear interpolation techniques reduce the difference between the computation results and the analytical solution while increasing the spatial order of convergence (reaching two for the velocity in $\mathscr{L}^2$

norm). Another validation case presented in this document is the laminar flow around a circular cylinder. It is declined in four configurations: steady regime with static cylinder, steady regime with rotating cylinder, unsteady regime with static cylinder and unsteady regime with rotating cylinder. Even if no analytical solution is available for the flow around a cylinder, a mesh convergence study is presented (the numerical results computed with the finest grid are considered as a reference) and shows, once again, that the, linear interpolation techniques allow to reach order 2 in space. The results obtained with our approach is also compared to experimental data and other simulations via quantities such as aerodynamic coefficients and the Strouhal number on the circular cylinder test case and a laminar flow past a NACA0012 airfoil test case. Globally, the values obtained are in good agreement with the literature which provides valuable validation data. An industrial study, involving a steady turbulent flow past a complex geometry (representative of a flow limiter) is also presented. Overall, the results are in good agreement with the previous preliminary studies [Bel18]. Moreover, simulations involving finer grids have been carried out, showing that the present method is more robust than the one used in [Bel18]. Finally, those industrial test cases tends to show that the directional interpolation approach provided the faster rate of convergence and lowest errors.

In the near future, two developments are considered. The first one consists in interpolating the normal component of the pressure gradient in the vicinity of the immersed boundaries. Indeed, the velocity is interpolated using the presented methodology but not the pressure gradient (nor pressure corrector). Yet, the pressure gradient appears in the Navier-Stokes prediction equation. An idea to deal with this issue could be to interpolate the normal pressure gradient or corrector (as we consider $\nabla p^{n+1} \cdot \mathbf{n} = 0$ as immersed boundary condition for the pressure) using the same methodology as the one used of the velocity. The second one involves turbulence modeling: the idea is to extend the interpolation process to turbulent wall laws (*i.e.* power wall law [WJS18]). Thus, we will be able to carry out turbulent simulations of the flow limiter involving well-established turbulent models as the wall laws, RANS and LES models.

The modeling of other passive safety systems, such as the advanced accumulator [Shi11], is also considered to demonstrate the interest of using our method to carry out design and shape optimization studies.

Bibliography

[Ang99] P. Angot. “Analysis of Singular Perturbations on the Brinkman Problem for Fictitious Domain of Viscous Flows”. In: *Mathematical Methods in the Applied Sciences* 22 (1999), pp. 1395–1412 (cit. on pp. 37, 38, 66, 91).

[Ang10] P. Angot. “A fictitious domain model for the Stokes/Brinkman problem with jump embedded boundary conditions”. In: *C.R. Acad. Sci. Paris, Ser. I* 348 (2010), pp. 697–702 (cit. on p. 34).

[ABF99] P. Angot, C.H. Bruneau, and P. Fabrie. “A penalization method to take into account obstacles in incompressible viscous flows”. In: *Numerische Mathematik* 81 (1999), pp. 497–520 (cit. on pp. 4, 6, 37, 38, 66, 91).

[AL17] P. Angot and Z. Li. “An augmented IIM & Preconditioning technique for jump embedded boundary conditions”. In: *International journal of numerical analysis and modeling* 14 (2017), pp. 712–729 (cit. on p. 34).

[BDY89] H.M. Badr, S.C.R. Dennis, and P.J.S. Young. “Steady and unsteady flow past a rotating circular cylinder at low Reynolds numbers”. In: *Computers & Fluids* 17.4 (1989), pp. 579–609. ISSN: 0045-7930 (cit. on p. 107).

[BMZ16] A. Baeza, P. Mulet, and D. Zorío. “High Order Boundary Extrapolation Technique for Finite Difference Methods on Complex Domains with Cartesian Meshes”. In: *Journal of Scientific Computing* 66 (2016), pp. 761–791 (cit. on pp. 33, 36).

[Bel18] M. Belliard. “Numerical Modeling of an in-vessel flow limiter using an Immersed Boundary Approach”. In: *Nuclear Engineering and Design* 330 (2018), pp. 437–449 (cit. on pp. 4, 6, 27, 28, 58, 88, 113, 119).

[BF10] M. Belliard and C. Fournier. “Penalized Direct Forcing and projection schemes for Navier-Stokes”. In: *C.R. Acad. Sci. Paris, Ser. I* 348 (2010), pp. 1133–1136 (cit. on pp. 6, 40, 66, 67).

[BI15] M. Belliard and C. Introïni. “Trio_u thermal-hydraulic simulations of flows induced by a stirrer and bubbling in a molten glass bath”. In: *Joint EUROMECH/ERCOFTAC Colloquium 549, Immersed Boundary Methods: Current Status and Future Research Directions*. Leiden, The Netherlands, June 17-19, 2015 (cit. on p. 30).

[BGV09] T. Belytschko, R. Gracie, and G. Ventura. "A review of extended/generalized finite element methods for material modelings". In: *Modelling and Simulation in Materials Science and Engineering* 17 (2009), p. 043001 (cit. on p. 36).

[BN12] R. L. T. Bevan and P. Nithiarasu. "Accelerating incompressible flow calculations using a quasi-implicit scheme: local and dual time stepping approaches". In: *Computational Mechanics* 50 (2012), pp. 687–693 (cit. on p. 65).

[BG20] A. Bharadwaj S and S. Ghosh. "Data reconstruction at surface in immersed-boundary methods". In: *Computers and Fluids* 196 (2020), p. 104236 (cit. on p. 33).

[BBS20] G. Billo, M. Belliard, and P. Sagaut. "A Finite Element Penalized Direct Forcing Immersed Boundary Method for infinitely thin obstacles in a dilatable flow". Submitted to *Computers & Mathematics with Applications* (under review). 2020 (cit. on pp. 4, 6).

[BCM01] D. L. Brown, R. Cortez, and M. L. Minion. "Accurate Projection Methods for the Incompressible Navier-Stokes Equations". In: *Journal of Computational Physics* 168 (2001), pp. 464–499 (cit. on pp. 39, 62).

[BHL20] E. Burman, P. Hansbo, and M.G. Larson. "A cut finite element method for a model of pressure in fractured media". In: *Numerische Mathematik* 146.4 (2020), pp. 783–818 (cit. on p. 36).

[CV01] J.P. Caltagirone and S. Vincent. "Sur une méthode de pénalisation tensorielle pour la résolution des équations de Navier–Stokes". In: *Comptes Rendus de l'Académie des Sciences - Series IIB - Mechanics* 329.8 (2001). In French, pp. 607–613. ISSN: 1620-7742 (cit. on p. 37).

[CEAa] CEA, ed. *TRUST Generic Guide V1.8.0*. URL: https://sourceforge.net/projects/trust-platform/files/TRUST_Generic_Guide.pdf/download (visited on 03/18/2020) (cit. on pp. 80, 81).

[CEAb] CEA, ed. *TRUST platform SourceForge mainlink*. URL: https://sourceforge.net/projects/trust-platform/ (visited on 03/18/2020) (cit. on p. 81).

[CB10] Y. Cheny and O. Botella. "The LS-STAG method: A new immersed boundary / level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties". In: *Journal of Computational Physics* 229.4 (2010), pp. 1043–1076. ISSN: 0021-9991 (cit. on p. 108).

[CAT20] C. Chi, A. Abdelsamie, and D. Thévenin. "A directional ghost-cell immersed boundary method for incompressible flows". In: *Journal of Computational Physics* 404 (2020), pp. 593–623 (cit. on pp. 33, 36).

[CLS10] P.H. Chiu, R.K. Lin, and Tony W.H. Sheu. "A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier-Stokes equations in time-varying complex geometries". In: *Journal of Computational Physics* 229.12 (2010), pp. 4476–4500. ISSN: 0021-9991 (cit. on p. 108).

[Cho+07] J.I. Choi, R.C. Oberoi, Jack R. Edwards, et al. "An immersed boundary method for complex incompressible flows". In: *Journal of Computational Physics* 224.2 (2007), pp. 757–784. ISSN: 0021-9991 (cit. on pp. 107, 108).

[Chu+08] I.C. Chu, C.H. Song, B.H. Cho, et al. "Development of passive flow controlling safety injection tank for APR1400". In: *Nuclear Engineering and Design* 238.1 (2008), pp. 200–206. ISSN: 0029-5493 (cit. on p. 28).

[Chu06] M.H. Chung. "Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape". In: *Computers & Fluids* 35.6 (2006), pp. 607–623. ISSN: 0045-7930 (cit. on pp. 107, 109).

[CW12] M. Cisternino and L. Weynans. "A parallel second order Cartesian method for elliptic interface problems". In: *Communications in Computational Physics* 12 (2012), pp. 1562–1587 (cit. on p. 34).

[Cle00] S. Clerc. "Numerical Simulation of the Homogeneous Equilibrium Model for Two-phase Flows." In: *Journal of Computational Physics* 161 (2000), pp. 354–375 (cit. on p. 58).

[DDR15] M. Dauge, A. Düster, and E. Rank. "Theoretical and Numerical Investigation of the Finite Cell Method". In: *Journal of Scientific Computing* 65 (2015), pp. 1039–1064 (cit. on p. 35).

[Dev+17] D. Devendran, D.T. Graves, H. Johansen, et al. "A fourth-order cartesian grid embedded boundary method for poisson's equation". In: *Communications in Applied Mathematics and Computational Science* 12 (2017), pp. 51–79 (cit. on p. 35).

[DS21] P. Di Stolfo and A. Schröder. "Reliable Residual-Based Error Estimation for the Finite Cell Method". In: *Journal of Scientific Computing* 87.12 (2021) (cit. on p. 35).

[Dom08] F. Domenichini. "On the consistency of the direct forcing method in the fractional step solution of the Navier-Stokes equations". In: *Journal of Computational Physics* 227 (2008), pp. 6372–6384 (cit. on pp. 41, 67).

[Düs+08] A. Düster, J. Parvizian, Z. Yang, et al. "The finite cell method for three-dimensional problems of solid mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 197.45-48 (2008), pp. 3768–3782 (cit. on p. 35).

[Fad+00] E.A. Fadlun, R. Verzicco, P. Orlandi, et al. "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations". In: *Journal of Computational Physics* 161 (2000), pp. 35–60 (cit. on pp. 33, 36).

[Fed02] R.P. Fedkiw. "Coupling an Eularian fluid calculation to a Lagrangian solid calculation with the ghost fluid method". In: *Journal of Computational Physics* 175.1 (2002), pp. 200–224 (cit. on p. 36).

[For80] B. Fornberg. "A numerical study of steady viscous flow past a circular cylinder". In: *Journal of Fluid Mechanics* 98.4 (1980), pp. 819–855 (cit. on p. 107).

[GG95] V. Girault and R. Glowinski. "Error Analysis of a Fictitious Domain Method applied to a Dirichlet Problem". In: *Japan J. Indust. Appl. Math.* 12 (1995), pp. 487–514 (cit. on p. 37).

[GR86] V. Girault and P.A. Raviart. *Finite Element Methods for Navier-Stokes Equations*. Springer-Verlag, 1986 (cit. on pp. 63, 64, 92).

[Glo+99] R. Glowinski, T.-W. Pan, T.I. Hesla, et al. "A distributed Lagrange multiplier/fictitious domain method for particulate flows". In: *International Journal of Multiphase Flow* 25 (1999), pp. 755–794 (cit. on pp. 30, 37).

[GPP94] R. Glowinski, T.-W. Pan, and J. Periaux. "A fictitious domain method for Dirichlet problem and applications". In: *Computer Methods in Applied Mechanics and Engineering* 111 (1994), pp. 283–303 (cit. on pp. 36, 37).

[Gra+90] M. Grandotto, J.L. Cheissoux, J.P. Gaillard, et al. "A 3D finite element code for solving two-phase flow analysis in PWR steam generators". In: *European Two-Phase Flow Group Meeting*. Joint Research Center Ispra, Varese, Italy, May 1990 (cit. on p. 81).

[GOch] M. Grandotto and P. Obry. "Calculs des écoulements diphasiques dans les échangeurs par une méthode aux éléments finis". In: *Revue Européenne des Eléments Finis* 5.1 (1996 [in French]), pp. 53–74 (cit. on pp. 58, 59, 81).

[Gre+84] P. M. Gresho, S. T. Chan, R. L. Lee, et al. "A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations. Part 1: Theory". In: *International Journal for Numerical Methods in Fluids* 4.6 (1984), pp. 557–598 (cit. on p. 106).

[GH10] R.D. Guy and D.A. Hertenstgine. "On the accuracy of direct forcing immersed boundary methods with projection methods". In: *Journal of Computational Physics* 229 (2010), pp. 2479–2496 (cit. on pp. 41, 67).

[GHP01] E. Guyon, J.P. Hulin, and L. Petit. *Physical Hydrodynamics*. Oxford University Press, 2001 (cit. on p. 86).

[HH02] A. Hansbo and P. Hansbo. "An unfitted finite element method, based on Nitsche's method, for elliptic interface problems". In: *Computer methods in applied mechanics and engineering* 191.47-48 (2002), pp. 5537–5552 (cit. on p. 35).

[HLZ14] P. Hansbo, M.G. Larson, and S. Zahedi. "A cut finite element method for a Stokes interface problem". In: *Applied Numerical Mathematics* 85 (2014), pp. 90–114. DOI: 10.1016/j.apnum.2014.06.009 (cit. on pp. 35, 36).

[IV03] G. Iaccarino and R. Verzicco. "Immersed boundary technique for turbulent flow simulations". In: *Applied Mechanics Reviews* 56.3 (May 2003), pp. 331–347 (cit. on pp. 33, 42).

[IK07] T. Ikeno and T. Kajishima. "Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations". In: *Journal of Computational Physics* 226 (2007), pp. 1485–1508 (cit. on pp. 33, 41, 67).

[IT90] D.B. Ingham and T. Tang. "A numerical investigation into the steady flow past a rotating circular cylinder at low and intermediate Reynolds numbers". In: *Journal of Computational Physics* 87.1 (1990), pp. 91–107. ISSN: 0021-9991 (cit. on p. 107).

[IBF14] C. Introïni, M. Belliard, and C. Fournier. "A second order penalized direct forcing for hybrid Cartesian/immersed boundary flow simulations". In: *Computers & Fluids* 90 (Feb. 2014), pp. 21–41 (cit. on pp. 6, 33, 40, 41, 45, 47, 66, 70, 72, 74, 107–109, 134).

[Ism04] M. Ismail. "The Fat Boundary Method for the numerical resolution of elliptic problems. Application to 3D fluid flows". PhD thesis. Univeristé Pierre et Marie Curie - Paris VI, France, 2004 (cit. on p. 37).

[JMW12] C. Ji, A. Munjiza, and J.J.R. Williams. "A novel iterative direct-forcing immersed boundary method and its finite volume applications". In: *Journal of Computational Physics* 231.4 (2012), pp. 1797–1821. ISSN: 0021-9991 (cit. on p. 108).

[JC98] H. Johansen and P. Colella. "A Cartesian Grid Embedded Boundary Method for Poisson's Equation on Irregular Domains". In: *Journal of Computational Physics* 147 (1998), pp. 60–85 (cit. on p. 35).

[KCL99] S. Kang, H. Choi, and S. Lee. "Laminar flow past a rotating circular cylinder". In: *Physics of Fluids* 11.11 (1999), pp. 3312–3321 (cit. on p. 109).

[KPS17] D.F. Khabarova, A.V. Podzerko, and E.K. Spiridonov. "Experimental Investigation of Fluidic Diodes". In: *Procedia Engineering* 206 (2017). International Conference on Industrial Engineering, ICIE 2017, pp. 93–98. ISSN: 1877-7058 (cit. on p. 28).

[Kha+00] K. Khadra, P. Angot, S. Parneix, et al. “Fictitious Domain Approach for Numerical Modelling of Navier-Stokes Equations”. In: *International Journal for Numerical Methods in Fluids* 34.8 (2000), pp. 651–684 (cit. on p. 31).

[LL03] L. Lee and R.J. LeVeque. “An immersed interface method for incompressible Navier-Stokes equations”. In: *Journal on Scientific Computing* 25 (2003), pp. 832–856 (cit. on p. 34).

[LL94] R.J. LeVeque and Z. Li. “The immersed interface method for elliptic equations with discontinuous coefficients and singular sources”. In: *J. Numeri. Anal.* 31 (1994), pp. 1019–1044 (cit. on p. 34).

[Li98] Z. Li. “The immersed interface method using a finite element formulation”. In: *Applied Numerical Mathematics* 27 (1998), pp. 253–267 (cit. on p. 34).

[Li03] Z. Li. “An overview of the Immersed Interface Method and its applications”. In: *TAIWANESE JOURNAL OF MATHEMATICS* 7 (2003), pp. 1–49 (cit. on pp. 34, 37).

[LF05] M.N. Linnick and H.F. Fasel. “A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains”. In: *Journal of Computational Physics* 204.1 (2005), pp. 157–192. ISSN: 0021-9991 (cit. on pp. 107, 108).

[MWD20] L. Manueco, P.E. Weiss, and S. Deck. “On the estimation of unsteady aerodynamic forces and wal spectral content with immersed boundary conditions”. In: *Computers and Fluids* 201 (2020), p. 104471 (cit. on p. 32).

[Mar82] G. I. Marchuk. *Methods of Numerical Mathematics*. second edition. Springer-Verlag New York, 1982. ISBN: 978-1-4613-8152-5 (cit. on pp. 4, 6).

[Mau01] B. Maury. “A fat boundary method for the Poisson problem in a domain with holes”. In: *Journal of Scientific Computing* 16 (2001), pp. 319–339 (cit. on p. 37).

[Moh97] J. Mohd-Yusof. “Combined Immersed-Boundary/B-spline methods for simulations of flow in complex geometries”. In: *Annual Research Briefs* (1997), pp. 317–327 (cit. on pp. 4, 39).

[Nit71] J. Nitsche. “Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind”. In: *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*. Vol. 36. 1. Springer. 1971, pp. 9–15 (cit. on p. 35).

[Nor03] C. Norberg. “Fluctuating lift on a circular cylinder: review and new measurements”. In: *Journal of Fluids and Structures* 17.1 (2003), pp. 57–96. ISSN: 0889-9746 (cit. on p. 108).

[PDR07] A. Parvizian, A. Düster, and E. Rank. “Finite cell method: h- and p-extension for embedded domain problems in solid mechanics”. In: *Computational Mechanics* 41.1 (2007), pp. 121–133 (cit. on p. 35).

[Pes72] C.S. Peskin. "Flow Patterns around heart vavles: A digital computer method for solving the equations of motion". PhD thesis. Albert Einstein College of Medecine, 1972 (cit. on pp. 32, 34, 38).

[Pes02] C.S. Peskin. "The immersed boundary method". In: *Acta Numerica* 11 (2002), pp. 479–517 (cit. on pp. 6, 38).

[Ram06] I. Ramière. "Méthodes de domaine fictif pour des problèmes elliptiques avec conditions aux limites générales en vue de la simulation numérique d'écoulements diphasiques". (In French). PhD thesis. Université de Provence - Aix-Marseille I, 2006 (cit. on pp. 31, 32).

[RAB07a] I. Ramière, P. Angot, and M. Belliard. "A fictitious domain approach with spread interface for elliptic problems with general boundary conditions". In: *Computer methods in applied mechanics and engineering* 196 (2007), pp. 766–781 (cit. on pp. 37, 38).

[RAB07b] I. Ramière, Ph. Angot, and M. Belliard. "A general fictitious domain method with immersed jumps and non-conforming structured mesh". In: *Journal of Computational Physics* 225.2 (2007), pp. 1347–1387 (cit. on p. 34).

[Ria+18] H. Riahi, M. Meldi, J. Favier, et al. "A pressure-corrected Immersed Boundary Method for the numerical simulation of compressible flows". In: *Preprint submitted to Journal of Computational Physics* (2018), pp. 361–383 (cit. on p. 35).

[Rod00] F. Roddier. *Distributions et transformation de Fourier : á l'usage des physiciens et des ingénieur*. (In French). McGraw-Hill, 2000, pp. 56–61 (cit. on p. 132).

[RPB99] A.M. Roma, C.S. Peskin, and M.J. Berger. "An Adaptative Version of the Immersed Boundary Method". In: *Journal of Computational Physics* 153 (2 1999), pp. 509–534 (cit. on p. 32).

[Sar09] A. Sarthou. "Methodes de domaines fictifs d'ordres élevés pour les équations elliptiques et de Navier-Stokes. Application au couplage fluide-structure". (In French). PhD thesis. Université de Bordeaux I, 2009 (cit. on pp. 37, 66).

[Sar+08] A. Sarthou, S. Vincent, P. Angot, et al. "The Sub Mesh Penalty Method". In: *Finite Volumes for Complex Applications V*. 2008, p. 633 (cit. on pp. 37, 66).

[Sau63] V.K. Saul'ev. "On the solution of some boundary value problems on high performance computers by fictitious domain method". In: *Siberian Math. Journal* 4.4 (1963). (in Russian), pp. 912–925 (cit. on pp. 30, 31).

[Sch68] H. Schlichting. *Boundary Layer Theory*. French. New York, USA: Mac Graw Hill, 1968, p. 35 (cit. on pp. 59, 60).

[Sch+06] P. Schwartz, M. Barad, P. Collela, et al. “A Cartesian Grid Embedded Boundary Method for the heat equation and Poisson’s equation in three dimensions”. In: *Journal of Computational Physics* 211 (2006), pp. 531–550 (cit. on p. 35).

[Shi+18] S. Shin, J.H. Jeong, D.K. Lim, et al. “Design of SFR fluidic diode axial port using topology optimization”. In: *Nuclear Engineering and Design* 338 (2018), pp. 63–73. ISSN: 0029-5493 (cit. on p. 28).

[Shi11] T. Shiraishi. “Design of the advanced accumulator for the pressurized water reactor”. In: *Nuclear Engineering and Design* 241.9 (2011), pp. 3910–3924. ISSN: 0029-5493 (cit. on pp. 4, 6, 27, 29, 119).

[SLK20] S. Sticko, G. Ludvigsson, and G. Kreiss. “High-order cut finite elements for the elastic wave equation”. In: *Advances in Computational Mathematics* 46.3 (2020) (cit. on p. 36).

[SBD02] D. Stojković, M. Breuer, and F. Durst. “Effect of high rotation rates on the laminar flow around a circular cylinder”. In: *Physics of Fluids* 14.9 (2002), pp. 3160–3178 (cit. on pp. 107–109).

[SB17] E. Stratta and M. Belliard. “Thermal-hydraulic study of passive safety systems based on the hydraulic diode principle for the management of large-break loss of coolant accidents”. In: *17th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-17)*. Qujiang Int. Conference Center, Xi’an, China, September 3-8, 2017 (cit. on pp. 27, 28).

[SL16] R.C. Swanson and S. Langer. “Steady-state laminar flow solutions for NACA 0012 airfoil”. In: *Computers and Fluids* 126 (2016), pp. 102–128 (cit. on pp. 87, 88, 110, 112).

[TC07] K. Taira and T. Colonius. “The immersed boundary method: A projection approach”. In: *Journal of Computational Physics* 225.2 (2007), pp. 2118–2137. ISSN: 0021-9991 (cit. on p. 107).

[Tri59] D. J. Tritton. “Experiments on the flow past a circular cylinder at low Reynolds numbers”. In: *Journal of Fluid Mechanics* 6.4 (1959), pp. 547–567 (cit. on p. 107).

[TF03] Y.H. Tseng and J.H. Ferziger. “A ghost-cell immersed boundary method for flow in complex geometry”. In: *Journal of Computational Physics* 192 (2003), pp. 593–623 (cit. on pp. 33, 36).

[VSD21] F. Varren, M. Silberstein, and S. Dutreuil. *Modéliser et simuler: Epistémologies et pratiques de la modélisation et de la simulation*. Ed. by Éditions Matériologiques. Second Edition. 2021. ISBN: 978-2-37361-270-7 (cit. on p. 26).

[Vin+07] S. Vincent, J.P. Caltagirone, S. Delage-Santacreu, et al. “Implicit penalty methods on Eulerian grids for the simulation of incompressible multiphase flows”. In: *La Houille Blanche* 2 (June 2007), pp. 53–59 (cit. on p. 37).

[Vin+11] S. Vincent, A. Sarthou, J.P. Caltagirone, et al. “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”. In: *Journal of Computational Physics* 230.4 (2011), pp. 956–983. ISSN: 0021-9991 (cit. on p. 30).

[Wag+01] G.J. Wagner, N. Moës, W.K. Liu, et al. “The extended finite element method for rigid particles in Stokes flow”. In: *Int. journ. for numerical methods in engineering* 51.3 (2001), pp. 293–313 (cit. on p. 33).

[Wey19] L. Weynans. *Convergence of a cartesian method for elliptic problems with immersed interfaces*. Tech. rep. 8872. INRIA, 2019 (cit. on p. 34).

[WJS18] S. Wilhelm, J. Jacob, and P. Sagaut. “An explicit power-law-based wall model for lattice Boltzmann method-Reynolds-averaged numerical simulations of the flow around airfoils”. In: *Physics of Fluids* 30 (2018), p. 065111 (cit. on pp. 5, 6, 119).

[YAA09] A. Yazid, N. Abdelkader, and H. Abdelmadjid. “A state-of-the-art review of the X-FEM for computational fracture mechanics”. In: *Applied Mathematical Modelling* 33 (2009), pp. 4269–4282 (cit. on pp. 33, 36).

[Ye+99] T. Ye, R. Mittal, H.S. Udaykumar, et al. “An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries”. In: *Journal of Computational Physics* 156.2 (1999), pp. 209–240. ISSN: 0021-9991 (cit. on p. 107).

[ZW06] Y.C. Zhou and G.W. Wei. “On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method”. In: *Journal of Computational Physics* 219.1 (2006), pp. 228–246. ISSN: 0021-9991 (cit. on p. 34).

Appendices

Summary

A. Weak formulation and distributions

A.1. Preamble

In this appendix, we try to compare the projection equation obtained in the classical two subproblems formulation (in which jump terms appears) and in our approach (formulated in the framework of distributions). Indeed, we wonder to what extent the source term added by the PDF in the projection equation can be seen as a model of the jump terms.

A.2. Classical subproblems formulation

Instead of considering a unique problem on the fictitious domain Ω, let us write two subproblems:

$$\begin{cases} \partial_t(\rho_f \mathbf{u}_f) + \nabla\cdot(\rho_f \mathbf{u}_f \otimes \mathbf{u}_f - \bar{\bar{\sigma}}_f + \bar{\bar{I}} p_f) = \mathbf{s} & \text{on } \Omega_f \\ \nabla\cdot(\rho_f \mathbf{u}_f) = 0 & \text{on } \Omega_f \\ +\text{BC} & \text{on } \partial\Omega_f \\ +\text{IC} & \text{on } \Omega_f \end{cases} \tag{A.1}$$

$$\begin{cases} \partial_t(\rho_s \mathbf{u}_s) + \nabla\cdot(\rho_s \mathbf{u}_s \otimes \mathbf{u}_s - \bar{\bar{\sigma}}_s + \bar{\bar{I}} p_s) = \mathbf{s} & \text{on } \Omega_s \\ \nabla\cdot(\rho_s \mathbf{u}_s) = 0 & \text{on } \Omega_s \\ +\text{BC} & \text{on } \partial\Omega_s \\ +\text{IC} & \text{on } \Omega_s \end{cases} \tag{A.2}$$

For each subproblem, we use the same time scheme as the one presented in Section 3.1.2.2 and come up with the weak formulation of the two projection equations, as shown in Section 3.1.3.1:
Find $\left(\phi_f^{n+1}, \phi_s^{n+1}\right) \in \mathcal{W}(\Omega_f) \times \mathcal{W}(\Omega_s)$ *such as, for all* $\left(q_f, q_s\right) \in \mathcal{W}(\Omega_f) \times \mathcal{W}(\Omega_s)$:

$$-\int_{\Omega_f} \nabla\phi_f^{n+1}\cdot\nabla q_f = -\frac{1}{\delta t}\int_{\Omega_f} \rho_f^n \mathbf{u}_f^*\cdot\nabla q_f - \int_{\partial\Omega_f} q_f\left(\nabla\phi_f^{n+1} - \rho_f^n \mathbf{u}_f^*\right)\cdot\mathbf{n}_{\partial\Omega_f} \tag{A.3}$$

$$-\int_{\Omega_s} \nabla\phi_s^{n+1}\cdot\nabla q_s = -\frac{1}{\delta t}\int_{\Omega_s} \rho_s^n \mathbf{u}_s^*\cdot\nabla q_s - \int_{\partial\Omega_s} q_s\left(\nabla\phi_s^{n+1} - \rho_s^n \mathbf{u}_s^*\right)\cdot\mathbf{n}_{\partial\Omega_s} \tag{A.4}$$

Yet, knowing that $\Gamma = \partial\Omega_f \cap \partial\Omega_s$, we can separate the boundary integrals between $\partial\Omega_f\backslash\Gamma$, $\partial\Omega_s\backslash\Gamma$ and Γ:
Find $\left(\phi_f^{n+1}, \phi_s^{n+1}\right) \in \mathcal{W}(\Omega_f) \times \mathcal{W}(\Omega_s)$ *such as, for all* $\left(q_f, q_s\right) \in \mathcal{W}(\Omega_f) \times \mathcal{W}(\Omega_s)$:

$$-\int_{\Omega_f}\nabla\phi_f^{n+1}\cdot\nabla q_f=-\frac{1}{\delta t}\int_{\Omega_f}\rho_f^n\mathbf{u}_f^*\cdot\nabla q_f-\int_{\partial\Omega_f\backslash\Gamma}q_f\left(\nabla\phi_f^{n+1}-\rho_f^n\mathbf{u}_f^*\right)\cdot\mathbf{n}_{\partial\Omega_f\backslash\Gamma}\qquad(A.5)$$
$$-\int_{\Gamma}q_f\left(\nabla\phi_f^{n+1}-\rho_f^n\mathbf{u}_f^*\right)\cdot\mathbf{n}_{\Gamma}$$
$$-\int_{\Omega_s}\nabla\phi_s^{n+1}\cdot\nabla q_s=-\frac{1}{\delta t}\int_{\Omega_s}\rho_s^n\mathbf{u}_s^*\cdot\nabla q_s-\int_{\partial\Omega_s\backslash\Gamma}q_s\left(\nabla\phi_s^{n+1}-\rho_s^n\mathbf{u}_s^*\right)\cdot\mathbf{n}_{\partial\Omega_s\backslash\Gamma}\qquad(A.6)$$
$$+\int_{\Gamma}q_s\left(\nabla\phi_s^{n+1}-\rho_s^n\mathbf{u}_s^*\right)\cdot\mathbf{n}_{\Gamma}$$

Then, if we try writing the union problem on a fictitious domain $\Omega=\Omega_f\cup\Omega_s$ with $\partial\Omega=(\partial\Omega_f\cup\partial\Omega_s)\backslash\Gamma$ by introducing $\mathbf{u}^*$, ρ^n and ϕ^{n+1} such as:

$$\phi^{n+1}\big|_{\Omega_f}=\phi_f^{n+1},\ \mathbf{u}^*\big|_{\Omega_f}=\mathbf{u}_f^*,\ \rho^n\big|_{\Omega_f}=\rho_f^n\qquad(A.7)$$
$$\phi^{n+1}\big|_{\Omega_s}=\phi_s^{n+1},\ \mathbf{u}^*\big|_{\Omega_s}=\mathbf{u}_s^*,\ \rho^n\big|_{\Omega_s}=\rho_s^n\qquad(A.8)$$
$$\phi^{n+1}\big|_{\partial\Omega_f\backslash\Gamma}=\phi_f^{n+1}\Big|_{\partial\Omega_f\backslash\Gamma},\ \mathbf{u}^*\big|_{\partial\Omega_f\backslash\Gamma}=\mathbf{u}_f^*\Big|_{\partial\Omega_f\backslash\Gamma},\ \rho^n\big|_{\partial\Omega_f\backslash\Gamma}=\rho_f^n\Big|_{\partial\Omega_f\backslash\Gamma}\qquad(A.9)$$
$$\phi^{n+1}\big|_{\partial\Omega_s\backslash\Gamma}=\phi_s^{n+1}\big|_{\partial\Omega_s\backslash\Gamma},\ \mathbf{u}^*\big|_{\partial\Omega_s\backslash\Gamma}=\mathbf{u}_s^*\big|_{\partial\Omega_s\backslash\Gamma},\ \rho^n\big|_{\partial\Omega_s\backslash\Gamma}=\rho_s^n\big|_{\partial\Omega_s\backslash\Gamma}\qquad(A.10)$$

and the vector space $\mathscr{W}(\Omega)$ such as:

$$\forall\, q\in\mathscr{W}(\Omega),\ q\big|_{\Omega_f}\in\mathscr{W}(\Omega_f)\text{ and }q\big|_{\Omega_s}\in\mathscr{W}(\Omega_s)\qquad(A.11)$$

we obtain the following equivalent weak formulation:
Find $\phi^{n+1}\in\mathscr{W}(\Omega)$ *such as, for all* $q\in\mathscr{W}(\Omega)$:

$$-\int_{\Omega}\nabla\phi^{n+1}\cdot\nabla q=-\frac{1}{\delta t}\int_{\Omega}\rho^n\mathbf{u}^*\cdot\nabla q-\int_{\partial\Omega\backslash\Gamma}q\left(\nabla\phi^{n+1}-\rho^n\mathbf{u}^*\right)\cdot\mathbf{n}_{\partial\Omega\backslash\Gamma}\qquad(A.12)$$
$$-\int_{\Gamma}q\left(\nabla\phi_f^{n+1}-\rho_f^n\mathbf{u}_f^*\right)\cdot\mathbf{n}_{\Gamma}+\int_{\Gamma}q\left(\nabla\phi_s^{n+1}-\rho_s^n\mathbf{u}_s^*\right)\cdot\mathbf{n}_{\Gamma}$$

A.3. Distribution framework

Let us introduce some notations. $\mathscr{D}(\Omega)$ is the set of test functions (or smooth compactly supported functions) on Ω and $\mathscr{L}^1_{\text{loc}}(\Omega)$ is the set of locally integrable functions on Ω with $\Omega\subseteq\mathbb{R}^d$. We denote the distribution T_g associated to a locally integrable function g:

$$g\in\mathscr{L}^1_{\text{loc}}(\Omega),\ T_g:\mathscr{D}(\Omega)\longrightarrow\mathbb{R}\,;\ v\mapsto\int_{\Omega}g\ v\qquad(A.13)$$

Obviously, T_g is in the dual space of $\mathscr{D}(\Omega)$, denoted $(\mathscr{D}(\Omega))'$. We also denote:

$$g \in \mathscr{L}^1_{\text{loc}}(\Omega),\ T_g \in (\mathscr{D}(\Omega))',\ \forall v \in \mathscr{D}(\Omega),\ \langle g, v\rangle := T_g(v) = \int_\Omega g\ v \tag{A.14}$$

The derivative of a distribution, coming from a formal integration by parts, is defined as follow:

$$\forall i \in [\![1,d]\!],\ \forall v \in \mathscr{D}(\Omega),\ \langle \partial_{x_i} g, v\rangle := -\langle g, \partial_{x_i} v\rangle \tag{A.15}$$

If we refer to the book by F. RODDIER [Rod00], the spatial derivative of a distribution associated to the function $g : \mathbb{R}^d \longrightarrow \mathbb{R}$ defined and derivable on $\mathbb{R}^d \setminus \Gamma$ and presenting a discontinuity on Γ – where Γ is a closed surface delimiting a volume $\Omega_s \subset \mathbb{R}^d$ such as $\Omega_s \cap \Gamma = \emptyset$ – is given by :

$$\forall v \in \mathscr{D}(\mathbb{R}^d), \langle \partial_{x_i} g, v\rangle = \langle \{\partial_{x_i} g\}, v\rangle - \langle \zeta_i \cos(\theta_i)\delta_\Gamma, v\rangle \tag{A.16}$$

where $\langle \{\partial_{x_i} g\}, v\rangle$ is the distribution associated to the regular part of the partial derivative of g along the direction x_i ($i \in [\![1,d]\!]$), δ_Γ is a superficial Dirac distribution associated to Γ, θ_i the angle between the direction x_i and the normal vector of Γ and , ζ_i is the jump of the function g across the surface Γ along direction x_i which is defined as follows:

$$\forall \mathbf{x} \in \Gamma, \zeta_i = g(\mathbf{x}_i^+) - g(\mathbf{x}_i^-) \tag{A.17}$$

with:

- $g(\mathbf{x}_i^+)$ the right limit – oriented along direction x_i – of function g at point of coordinates $\mathbf{x}$,
- $g(\mathbf{x}_i^-)$ the left limit of function g at point of coordinates $\mathbf{x}$.

Now, let us consider the function $h : \Omega \longrightarrow \mathbb{R}$ such as:

$$\forall \mathbf{x} \in \Omega,\ h(\mathbf{x}) = \frac{\eta}{\eta + \chi(\mathbf{x})} \tag{A.18}$$

or, said otherwise, as $\eta \ll 1$:

$$\forall \mathbf{x} \in \Omega,\ h(\mathbf{x}) \approx \begin{cases} \eta, & \text{if } \mathbf{x} \in \Gamma \\ 1, & \text{else} \end{cases} \tag{A.19}$$

This function h, can be seen as the sum of two function h_1 and h_2 such as:

$$\forall \mathbf{x} \in \Omega,\ h_1(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \Omega_s \\ \eta, & \text{else} \end{cases} \quad \text{and } h_2(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \in \Omega_s \cup \Gamma \\ 1-\eta, & \text{else} \end{cases} \tag{A.20}$$

Then, if we consider the pressure corrector gradient (for purpose of clarity, time index notations are given up) in the weak form of the projection equation (3.41) in the framework of distributions we get:

$$\forall i \in [\![1,d]\!],\ \forall v \in \mathscr{D}(\Omega),\ \left\langle h\partial_{x_i}\phi, v\right\rangle = \left\langle h_1\partial_{x_i}\phi, v\right\rangle + \left\langle h_2\partial_{x_i}\phi, v\right\rangle \tag{A.21}$$

if we assume that, for all $i \in [\![1,d]\!]$, $\partial_{x_i}\phi \in \mathscr{L}^1_{\text{loc}}(\Omega)$ at least. Now, if we assume that $\partial_{x_i}\phi$ is differentiable in Ω, we can use equation (A.16):

$$\begin{aligned}\forall i \in [\![1,d]\!],\ \forall v \in \mathscr{D}(\Omega),\ \left\langle \partial_{x_i}\left(h\partial_{x_i}\phi\right), v\right\rangle &= \left\langle \partial_{x_i}\left(h_1\partial_{x_i}\phi\right), v\right\rangle + \left\langle \partial_{x_i}\left(h_2\partial_{x_i}\phi\right), v\right\rangle \\ &= \left\langle \left\{\partial_{x_i}\left(\partial_{x_i}\left(h_1+h_2\right)\phi\right)\right\}, v\right\rangle \\ &+ \left\langle \zeta_i^{h_1}\cos(\theta_i)\delta_\Gamma, v\right\rangle + \left\langle \zeta_i^{h_2}\cos(\theta_i)\delta_\Gamma, v\right\rangle \\ &= \left\langle \left\{\partial^2_{x_i}\phi\right\}, v\right\rangle + \left\langle \left(\zeta_i^{h_1}+\zeta_i^{h_2}\right)\cos(\theta_i)\delta_\Gamma, v\right\rangle\end{aligned} \tag{A.22}$$

with:

$$\begin{aligned}\forall \mathbf{x} \in \Gamma,\ \zeta_i^{h_1} &= h_1(\mathbf{x}_i^+)\partial_{x_i}\phi(\mathbf{x}_i^+) - h_1(\mathbf{x}_i^-)\partial_{x_i}\phi(\mathbf{x}_i^-) \\ \zeta_i^{h_2} &= h_2(\mathbf{x}_i^+)\partial_{x_i}\phi(\mathbf{x}_i^+) - h_2(\mathbf{x}_i^-)\partial_{x_i}\phi(\mathbf{x}_i^-)\end{aligned} \tag{A.23}$$

Then, if we assume continuity of $\nabla\phi$ through Γ, the jump terms reduce to the jumps of h_1 and h_2 functions, which means :

$$\begin{aligned}&\forall i \in [\![1,d]\!],\ \forall \mathbf{x} \in \Gamma,\ \partial_{x_i}\phi(\mathbf{x}_i^+) = \partial_{x_i}\phi(\mathbf{x}_i^-) \\ \Rightarrow &\forall i \in [\![1,d]\!],\ \forall \mathbf{x} \in \Gamma,\ \zeta_i^{h_1} = h_1(\mathbf{x}_i^+) - h_1(\mathbf{x}_i^-) = -\left(h_2(\mathbf{x}_i^+) - h_2(\mathbf{x}_i^-)\right) = -\zeta_i^{h_2} \\ \Rightarrow &\forall i \in [\![1,d]\!],\ \forall \mathbf{x} \in \Gamma,\ \zeta_i^{h_1} + \zeta_i^{h_2} = 0 \\ \Rightarrow &\forall i \in [\![1,d]\!],\ \forall v \in \mathscr{D}(\Omega),\ \left\langle \partial_{x_i}\left(h\partial_{x_i}\phi\right), v\right\rangle = \left\langle \left\{\partial^2_{x_i}\phi\right\}, v\right\rangle\end{aligned} \tag{A.24}$$

Hence, if we consider the BC on Γ:

$$\begin{aligned}\mathbf{u}_s\cdot\mathbf{n}_\Gamma|_\Gamma &= \mathbf{u}_f\cdot\mathbf{n}_\Gamma\big|_\Gamma = 0 \\ \nabla\phi_s\cdot\mathbf{n}_\Gamma\big|_\Gamma &= \nabla\phi_f\cdot\mathbf{n}_\Gamma\big|_\Gamma = 0\end{aligned} \tag{A.25}$$

the integral over Γ vanishes in equation (A.12) and we obtain equivalent formulations with the two approaches (subproblems and distributions).

B. Topology of the flow around a circular cylinder

In this appendix, we saw some qualitative results obtained using GENEPI3 with directional interpolation. Those streamlines (*cf.* Figure B.1), vorticity (*cf.* Figure B.2) and pressure (*cf.* Figure B.3) contours show a good agreement with the ones presented in the litterature, especially in [IBF14].

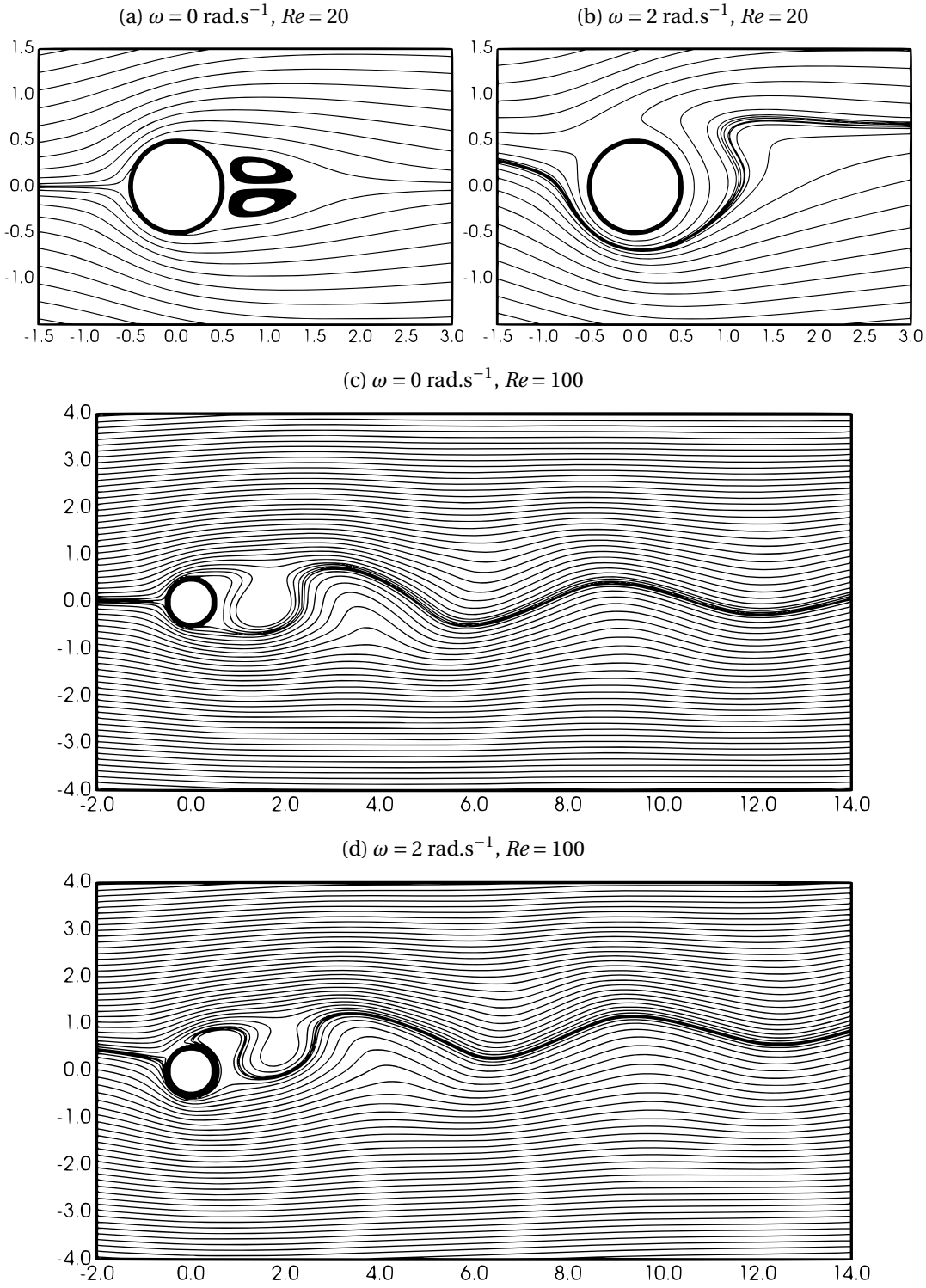


Figure B.1. Streamlines computed for various configurations of the flow around a circular cylinder.

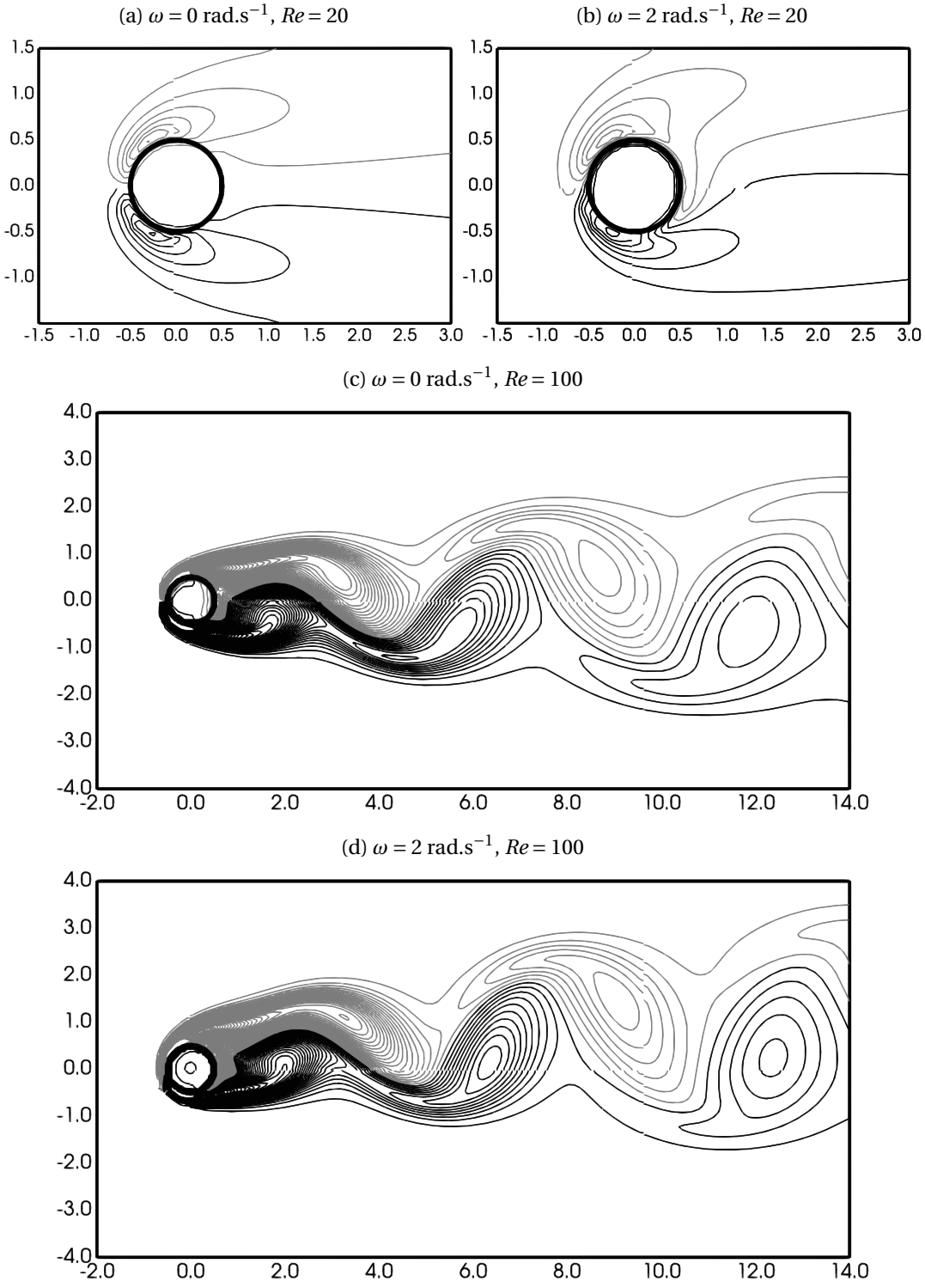
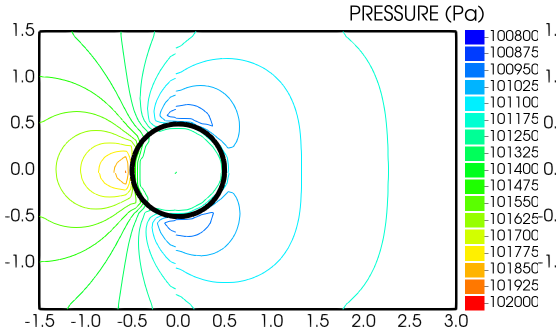
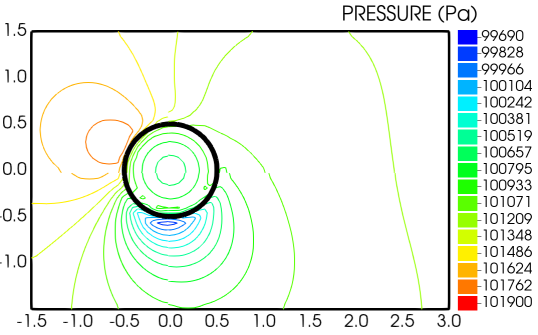


Figure B.2. Vorticity contours for various configurations of the flow around a circular cylinder.

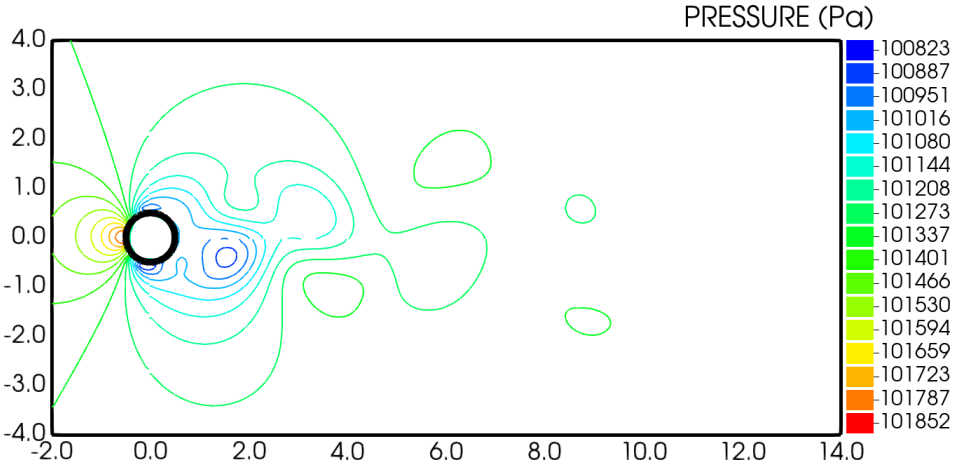
(a) $\omega = 0$ rad.s^{-1}, $Re = 20$



(b) $\omega = 2$ rad.s^{-1}, $Re = 20$



(c) $\omega = 0$ rad.s^{-1}, $Re = 100$



(d) $\omega = 2$ rad.s^{-1}, $Re = 100$

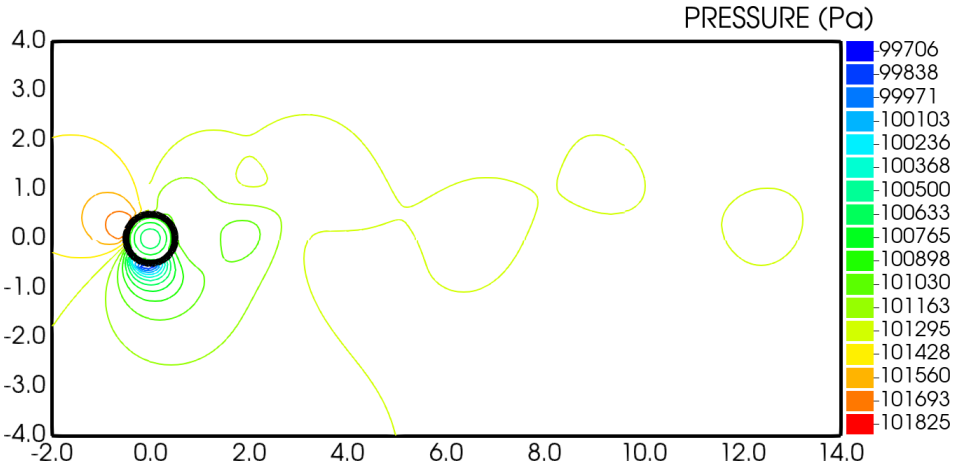


Figure B.3. Pressure contours for various configurations of the flow around a circular cylinder.