

**SIMULATION DE FLUIDE BASÉE SUR LA MÉTHODE
HYDRODYNAMIQUE DES PARTICULES LISSÉES**

par

Sun-Lay Gagneux

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 24 octobre 2022

Le 24 octobre 2022

Le jury a accepté le mémoire de Sun-Lay Gagneux dans sa version finale

Membres du jury

Professeur Richard Egli
Directeur de recherche
Département informatique

Professeur associé Khalid Djado
Codirecteur de recherche
Département informatique

Professeur Guillaume Gilet
Membre interne
Département informatique

Professeur Manuel Lafond
Président-rapporteur
Département informatique

Sommaire

La simulation des fluides basée sur la résolution d'un modèle physique a suscité énormément d'engouement dans les deux dernières décennies pour ses applications dans l'industrie, notamment dans l'univers des films ou des jeux vidéos, mais aussi en ingénierie aérospatiale ou dans les sciences fondamentales. Celles-ci demandent davantage de réalisme et de précision dans les phénomènes simulés et, bien que de nos jours, certaines méthodes sacrifient de la précision pour gagner en stabilité et en vitesse de simulation, certains phénomènes ne peuvent pas être simulés si l'implémentation du modèle physique n'est pas suffisamment fidèle, c'est le cas par exemple de la lévitation d'une balle sous un souffleur d'air. À partir de la méthode Hydrodynamique des Particules Lissées Sans Divergence (HPLSD, *Divergence-Free Smoothed Particle Hydrodynamics*) basée sur une approche lagrangienne, ce mémoire a pour objectif de présenter une méthode respectant davantage le modèle physique afin de simuler les phénomènes aérodynamiques complexes tout en permettant de garder la précision et la vitesse de simulation des simulateurs modernes. Notre apport consiste à introduire une nouvelle force de pénalité basée sur l'effet de Coandă corrigeant les erreurs induites par les suppositions de HPLSD ne respectant pas la contrainte de divergence zéro appartenant à la forme incompressible des équations de Navier-Stokes.

Mots-clés: HPL ; Simulation de fluide ; Méthode lagrangienne ; Équation de Navier-Stokes ; HPLSD ; Effet de Coandă.

Remerciements

Je tiens à remercier mon directeur de recherche Richard Egli de m'avoir accompagné durant tout mon parcours de maîtrise à temps partiel, de m'avoir aidé et supporté dans la bonne humeur pendant ces cinq longues années.

Je remercie également mon co-directeur de recherche Khalid Djado pour son aide tout aussi précieuse, son accompagnement et sans qui je n'aurais jamais eu la chance d'entreprendre ces recherches.

Je remercie également le professeur Fabrice Colin qui a pris le temps de me répondre et de faire la démonstration mathématique afin de clarifier certaines de mes questions.

Puis je remercie Manuel Lafond, président du jury, et Guillaume Gilet, membre du jury, d'avoir accepté d'évaluer mes travaux.

Je remercie ensuite ma tante et ma grand-mère qui m'ont aidé financièrement tout au long de mes études.

Je remercie ma famille en France qui m'a encouragé et a mis à disposition plusieurs ordinateurs. Notamment mon frère, Olivier, pour son assistance dans la coordination à distance de mes simulations.

Enfin, je remercie toutes les personnes à mon travail qui se sont révélées patientes et m'ont permis de travailler à temps partiel pendant toute la durée de mes études.

Abréviations

HPL	Hydrodynamique des particules lissées (<i>Smoothed Particle Hydrodynamics</i>)
HPLSE	Hydrodynamique des particules lissées avec Équation d'État (<i>State Equation Smoothed Particle Hydrodynamics</i>)
HPLFC	Hydrodynamique des Particules Lissées Faiblement Compressible (<i>Weakly Compressible Smoothed Particle Hydrodynamics</i>)
HPLIPC	Hydrodynamique des Particules Lissées Incompressible Predictif-Correctif (<i>Predictive-Corrective Incompressible Smoothed Particle Hydrodynamics</i>)
HPLII	Hydrodynamique des Particules Lissées Implicitement Incompressible (<i>Implicit Incompressible Smoothed Particle Hydrodynamics</i>)
HPLSD	Hydrodynamique des Particules Lissées Sans Divergence (<i>Divergence-Free Smoothed Particle Hydrodynamics</i>)
CFL	Condition de Courant-Friedrichs-Lewy
EPP	Équation de Pression de Poisson (<i>Pressure Poisson Equation</i>)
IUDM	Instruction Unique, Données Multiples (<i>Single Instruction, Multiple Data (SIMD)</i>)
SLS	Schéma Semi-Lagrangian (<i>Semi-Lagrangian Scheme</i>)

Table des matières

Sommaire	ii
Remerciements	iii
Abréviations	iv
Table des matières	v
Liste des figures	viii
Liste des tableaux	x
Liste des algorithmes	xi
Introduction	1
1 Fondements pour la simulation des fluides	5
1.1 Notions mathématiques	5
1.1.1 Advection	6
1.2 Notions physiques	6
1.3 Différentes approches pour simuler des fluides	8
1.3.1 Approche eulérienne	9
1.3.2 Approche lagrangienne	10
1.3.3 Approches mixtes	12
1.3.4 Approches non basées sur la physique	15

TABLE DES MATIÈRES

2	L'approche HPL	16
2.1	Fondements	16
2.1.1	Méthode d'interpolation	16
2.1.2	Discretisation de l'opérateur gradient appliquée à l'interpolation HPL	20
2.2	Le concept de la simulation HPL	21
2.2.1	Initialisation	21
2.2.2	Recherche du voisinage	22
2.2.3	Mise à jour du champ scalaire de densité	24
2.2.4	Résolution de la forme incompressible des équations de Navier-Stokes	24
2.2.5	Pression	29
2.2.6	Méthode d'Euler semi-implicite	47
2.3	Conditions de stabilité	47
2.4	Gestion des frontières	49
2.4.1	Les méthodes pour gérer les frontières	49
3	Lévitiation d'objets dans un fluide	56
3.1	Lévitiation d'une balle dans un souffleur	56
3.2	Principe de Bernoulli	59
3.3	Effet Coandă	59
3.4	Effet d'une pression négative	60
3.5	Implémentation	63
3.5.1	Gestion des conditions aux frontières	63
3.5.2	Initialisation des quantités (scalaires et vectorielles) physiques des particules de fluide	66
3.5.3	Le moteur de simulation	66
3.6	Résultats	71
3.6.1	Simulation de validation	71
3.6.2	Balle en lévitation sous un souffleur en position droite	74
3.6.3	Balle en lévitation sous un souffleur en position incliné	77
3.7	Limitation	83

TABLE DES MATIÈRES

Conclusion	85
Bibliographie	87
A Rappels mathématiques	94
A.1 Gradient	94
A.2 Divergence	95
A.3 Laplacien	95
A.4 Différences finies	96
B Liens et détails des vidéos	97
B.1 SPlisHSPlasH	97
B.2 Lien vers notre application <i>Storm</i>	97
B.3 Vidéos	98
B.3.1 Bande-annonce	98
B.3.2 Résultats d’optimisation de la recherche de voisinage avec grille spatiale 3D	98
B.3.3 Simulation d’une scène en configuration de cassure de barrage	99
B.3.4 Test de flottaison	100
B.3.5 Vidéo de la réalité montrant la lévitation d’une balle de ping-pong sous un sèche-cheveux	101
B.3.6 Simulation de la lévitation d’une balle sous un souffleur positionné à la verticale	101
B.3.7 Simulation de la lévitation d’une balle sous un souffleur incliné à sept degrés sans effet Coandă	101
B.3.8 Simulation de la lévitation d’une balle sous un souffleur incliné à sept degrés avec effet Coandă	102

Liste des figures

1.1	Anatomie d'une cellule de coordonnée (i, j) dans la grille eulérienne.	9
2.1	Tracé graphique du noyau spline cubique et son gradient.	19
2.2	Découpage de l'espace en grille uniforme pour une recherche optimale du voisinage.	25
2.3	Tracé graphique du laplacien du noyau spline cubique.	26
2.4	Simulation avec et sans gravité d'une balle coulant dans l'eau.	28
2.5	Simulation de trois balles de densité différentes immergées dans de l'eau.	29
2.6	Simulation HPLSE avec différentes constantes de raideur.	32
2.7	Idée générale derrière la gestion des frontières échantillonnées en particule.	50
2.8	Schéma 2D d'un voisinage rempli autour d'une particule de fluide	51
2.9	Coupes 2D montrant la contribution au voisinage autour de la particule rouge au contact des particules frontières vertes.	52
2.10	Artefact entraîné par la condition de non-collage aux frontières dans le cas d'une balle immergée.	55
3.1	Correction partielle de notre implémentation du résolveur de l'EPP prenant l'invariance de densité en source d'entrée.	58
3.2	Artefact d'agrégation non physique des particules en raison d'une pression négative	61
3.3	Surface de la courbe de la forme symétrique de l'approximation du gradient appliquée à la force de pression en fonction des valeurs de densité des particules i et j	63

LISTE DES FIGURES

3.4	Échantillonnage des corps solides.	64
3.5	Trous formés dans le maillage lors de l'échantillonnage en disques de Poisson.	65
3.6	Remplissage du domaine en fonction de diverses valeurs de coefficient c_V initialisant la masse d'une particule.	67
3.7	Simulation en cassure de barrage avec plusieurs solides.	72
3.8	Rotation du solide jaune sous l'effet d'une masse d'eau tombante. . .	73
3.9	Balle de ping-pong en lévitation sous un souffleur (sèche-cheveux) en position droite.	75
3.10	Simulation d'une balle sous un souffleur placé à la verticale.	76
3.11	Balle de ping-pong en lévitation sous un souffleur en position incliné de sept degrés dans la réalité.	78
3.12	Simulation d'une balle en lévitation sous un souffleur incliné à sept degrés.	79
3.13	Schéma de la zone d'effet cylindrique du souffleur.	81
3.13	Schéma de la zone d'effet cylindrique du souffleur.	82
3.14	Coupe de fluide autour d'un solide montrant le vide créé par notre implémentation de l'effet Coandă.	84

Liste des tableaux

3.1 Performances entre <i>Storm</i> et <i>SPlisHSPlasH</i>	68
B.1 Gain de performance obtenue avec la recherche de voisinage utilisant une grille spatiale 3D.	99

Liste des algorithmes

1	Génération des particules de fluide dans le domaine	22
2	Initialisation du simulateur	23
3	Algorithme HPLFC	36
4	Algorithme HPLIPC	39
5	Algorithme HPLII	42
6	Résolveur EPP d'invariance de densité de HPLSD	44
7	Résolveur EPP en divergence de vitesse de HPLSD	45
8	Algorithme HPLSD	46

Introduction

Un fluide est un milieu matériel continuellement déformable sous l'effet d'une contrainte de cisaillement. Il apparaît donc que les liquides comme l'eau et les gaz comme l'air sont regroupés sous cette appellation [2, pp.11-12].

Depuis plusieurs décennies, la simulation des fluides s'est davantage concentrée à résoudre un modèle physique basé principalement sur les équations de Navier-Stokes. Les enjeux techniques d'aujourd'hui demandent néanmoins de faire un compromis entre la vitesse de simulation et la précision des résultats; le choix étant déterminé par les objectifs finaux pour lesquels la simulation est produite. L'industrie cinématographique et du multimédia, de plus en plus exigeante dans la création des films d'animation, d'effets spéciaux et des jeux vidéos de haute qualité tout en conservant le même budget, repousse continuellement les limites des méthodes utilisées. De nos jours, confier la tâche à un artiste d'éplucher manuellement la simulation image par image pour ensuite cacher les artefacts visuels d'un mauvais comportement du fluide est un travail de plus en plus difficile, fastidieux et coûteux en temps, donc en argent. C'est la raison pour laquelle une multitude de travaux ont été réalisés depuis les six dernières décennies afin de produire rapidement un rendu de fluide au comportement presque parfait se basant sur la réalité. Par exemple, dans *Alita : L'Ange conquérant*¹, lorsque le personnage principal émerge de l'eau, capturer les effets à petite échelle de l'eau qui ruisselle et dépose une légère pellicule mouillée sur le corps du personnage a été un défi technique d'une importance majeure pour produire un film photoréaliste [43]. Néanmoins, plus la méthode et les modèles mathématiques et physiques sont complexes et exhaustifs (e.g. utilisation de la méthode des éléments finis [3, pp.11-14] pour résoudre la formulation générale des équations de Navier-Stokes), plus le temps

1. *Alita : L'Ange conquérant*, film sorti en 2019.

INTRODUCTION

de simulation et la complexité de l'implémentation augmentent. Cela est acceptable pour des applications en ingénierie comme la conception navale ou aérospatiale où une imprécision peut amener à de lourdes conséquences, mais cela l'est moins dans les films aux ressources et budget limités et ne nécessitant pas une telle précision, ou dans les jeux vidéos nécessitant des simulations en temps réel à trente images par seconde ; quitte à faire des approximations pour sauver du temps, des coûts de production et de la performance, ceci au détriment de la précision des résultats physiques obtenus et en réduisant les cas de figure qu'il est possible de simuler. Par exemple, la forme incompressible des équations de Navier-Stokes, étant un modèle physique simplifié, comparé à sa formulation générale, peut être utilisée. Cependant, cette formulation est, comme son nom l'indique, utilisable que dans le cas où le fluide est considéré comme incompressible ; ainsi, seul un fluide se déplaçant suffisamment lentement en tout point le constituant pourra être simulé suivant ce modèle [2, p.64].

La méthode de simulation peut aussi être simplifiée. F.H. Harlow et J.E. Welch [24] proposent une approche eulérienne utilisant la méthode des différences finies pour résoudre la forme incompressible des équations de Navier-Stokes (voir la [sous-section 1.3.1](#) et la [section A.4](#)), leurs travaux sont ensuite repris par N. Foster et D. Metaxas [20] pour réaliser le premier simulateur de fluide basé sur une approche eulérienne. En parallèle, L.B. Lucy [29], R.A. Gingold et J.J. Monaghan [22] proposent la méthode hydrodynamique des particules lissées (méthode HPL, *Smoothed Particle Hydrodynamics*) basée sur une approche lagrangienne pour résoudre des problèmes en astrophysique. M. Müller et al. [30] introduisent ensuite HPL en informatique graphique pour simuler des fluides en 3D.

La méthode HPL a connu un tel engouement durant ces deux dernières décennies qu'une multitude de variantes ont été créées dans l'objectif d'améliorer la stabilité et la vitesse de simulation de la méthode originale, notamment l'une des dernières méthodes : HPL Sans Divergence (HPLSD, *Divergence-Free Smoothed Particle Hydrodynamics*) inventée par J. Bender et D. Koschier [4]. Cependant, la plupart des variantes créées dans la dernière décennie, y compris HPLSD, se sont davantage concentrées sur la modélisation des liquides et ont négligé celle des gaz.

L'objectif de nos travaux est donc de réaliser une simulation aérodynamique en interaction avec des objets solides à paroi fine (*thin shell*) avec la méthode HPLSD, cor-

INTRODUCTION

rigéant au passage les erreurs introduites dans plusieurs méthodes HPL, dont HPLSD, rendant ainsi ces méthodes inadaptées pour la simulation de certains effets aérodynamiques comme il est expliqué au [chapitre 3](#), notamment en raison d'un jeu de constante à paramétrer amenant plus de mille milliards de possibilités, ou des erreurs induites dans le modèle physique utilisé dans ces méthodes empêchant la correction de la force de pression dans des situations de dilatation, comme explicitées aux sections [3.1](#) et [2.2.5.1](#). Pour valider nos résultats, nous tenterons de faire léviter une balle sous un souffleur placé en position droite, puis incliné. Toutefois, la solution devra demeurer valide pour toutes les scènes pouvant être simulées avec la méthode HPLSD originale, telles que la cassure de barrage (*dam break*)² ou la chute ou la flottaison d'objets immergés et présentant des densités différentes.

Ce mémoire comporte trois chapitres et deux annexes. Les diverses notions mathématiques et physiques essentielles utilisées en simulation de fluide sont abordées au [chapitre 1](#), une attention particulière est portée sur la forme incompressible des équations de Navier-Stokes, donnée dans la [section 1.2](#). Puis un état de l'art en matière de simulation de fluide, basée ou non sur un modèle physique, est présenté dans le reste du [chapitre 1](#). Puisque nos recherches se concentrent davantage sur HPL, le [chapitre 2](#) explicite dans un premier temps, les fondements de la méthode ainsi que les nombreuses variantes étudiées et implémentées dans le cadre de nos travaux, c'est-à-dire [HPL Faiblement Compressible \(HPLFC\)](#), [HPL Incompressible Predictif-Correctif \(HPLIPC\)](#), [HPL Implicitement Incompressible \(HPLII\)](#) et enfin, [HPL Sans Divergence \(HPLSD\)](#); puis dans un deuxième temps, les différentes méthodes pour gérer les interactions fluides↔solides sont énoncées dans la [section 2.4](#). Il est à noter que toutes ces méthodes sont implémentées dans un simulateur de fluide dont le code source est ouvert, nommé *SPlisHSPlasH* (le lien est fourni dans la [section B.1](#)) et que nous utilisons pour valider chacun de nos résultats de simulation lorsque c'est possible. Nos résultats et la méthodologie mise en place sont décrits dans le [chapitre 3](#) avant de poursuivre dans la [conclusion](#) par un récapitulatif du problème empêchant HPLSD de faire léviter une balle sous un souffleur d'air, de la solution introduite dans nos travaux pour y parvenir, les artefacts entraînés par notre apport et enfin, les recherches futures qu'il est possible de mener en continuation de nos travaux. Pour

2. <https://www.youtube.com/watch?v=GSVKn9cx9Ao>

INTRODUCTION

finir, l'[annexe A](#) présente les bases mathématiques générales nécessaires pour la compréhension du mémoire et l'[annexe B](#) fournit tous nos résultats et les liens vers les vidéos accompagnées d'une description succincte pour chacune d'entre elles.

Chapitre 1

Fondements pour la simulation des fluides

Afin d'introduire les concepts utilisés derrière la simulation de fluide, les principes mathématiques et physiques suivis d'un état de l'art général en matière de simulation de fluide sont énoncés ici.

1.1 Notions mathématiques

Un fluide est un milieu matériel continuellement déformable sous l'effet d'une contrainte de cisaillement, la notion de champ scalaire et vectoriel est donc utilisée pour modéliser celui-ci [2, pp.11-13]. Un champ est une notion mathématique spécifiant une quantité (scalaire ou vectorielle) définie par une fonction de l'espace $f(x, y, z)$, et parfois du temps $f(x, y, z, t)$ [23, pp.268-269]. Des opérateurs mathématiques, tels que le gradient ($\nabla\phi$ avec ϕ une quantité d'un champ scalaire caractérisant le fluide), la divergence ($\nabla\vec{u}$ avec \vec{u} une quantité vectorielle d'un champ vectoriel) et enfin, le laplacien ($\nabla^2\phi$ ou $\nabla^2\vec{u}$), sont utilisés pour manipuler ces champs. Davantage de détails sur ces opérateurs sont rappelés dans l'[annexe A](#).

1.2. NOTIONS PHYSIQUES

1.1.1 Advection

Soit \vec{v} le champ de vitesse, l'opérateur d'advection $\vec{v} \cdot \nabla$ est donné par

$$\vec{v} \cdot \nabla = \begin{bmatrix} v_x \frac{\partial}{\partial x}, \\ v_y \frac{\partial}{\partial y}, \\ v_z \frac{\partial}{\partial z}, \end{bmatrix}$$

avec $\vec{v} = (v_x, v_y, v_z)$. Soit A un champ scalaire ou vectoriel, alors l'opérateur advection appliqué sur A est donné par

$$(\vec{v} \cdot \nabla)A. \tag{1.1}$$

Dans le cadre d'une simulation de fluide, l'advection représente comment les quantités de A sont transportées avec le mouvement induit par le champ de vitesse [9, p.9].

1.2 Notions physiques

En mécanique classique du point, le mouvement d'un point soumis à des forces extérieures \vec{F}_{ext} est décrit par la seconde loi de Newton tel que

$$\sum \vec{F}_{ext} = m\vec{a} \tag{1.2}$$

avec m la masse du système et \vec{a} son accélération.

Ensuite, la troisième loi de Newton déclarant que toute force, appliquée d'un système A sur un système B , produit une force de même amplitude, mais de sens contraire appliquée de B sur A , soit

$$\vec{F}_{A \rightarrow B} = -\vec{F}_{B \rightarrow A}. \tag{1.3}$$

Puisque $\vec{a} = \frac{d\vec{v}}{dt}$, l'équation (1.2) peut se réécrire en termes de changement de quantité de mouvement tel que

$$\sum \vec{F}_{ext} = \frac{d(m\vec{v})}{dt}. \tag{1.4}$$

1.2. NOTIONS PHYSIQUES

La forme incompressible des équations de Navier-Stokes a la même forme que la seconde loi de Newton (équations (1.2) et (1.4)). Selon J. Bender et D. Koschier [4], sa formulation est

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\vec{u} + \vec{g} + \vec{a}_{ext}, \quad (1.5)$$

$$\nabla \cdot \vec{u} = 0, \quad (1.6)$$

avec \vec{u} le champ de vitesse du fluide dépendant de l'espace x, y, z et du temps t , ρ la valeur du champ de densité, p la valeur du champ de pression, \vec{g} l'accélération de pesanteur, ν la viscosité cinématique du fluide (donnée par la relation avec la viscosité dynamique η telle que $\nu = \frac{\eta}{\rho}$) et \vec{a}_{ext} toutes les accélérations induites par les forces extérieures appliquées sur le fluide.

À noter que la notation $\frac{D\vec{u}}{Dt}$ est celle de la dérivée particulaire, telle que

$$\frac{D\vec{u}}{Dt} = \frac{\partial\vec{u}}{\partial t} + \frac{\partial\vec{u}}{\partial x}\frac{dx}{dt} + \frac{\partial\vec{u}}{\partial y}\frac{dy}{dt} + \frac{\partial\vec{u}}{\partial z}\frac{dz}{dt}. \quad (1.7)$$

En observant chaque terme des équations de Navier-Stokes (1.5), nous déduisons :

- $-\frac{1}{\rho}\nabla p$ est l'accélération engendrée par le gradient du champ scalaire de pression.
- $\nu\nabla^2\vec{u}$ est le terme d'accélération entraîné par la viscosité. Celui-ci tend à diffuser le champ de vitesse pour le rendre uniforme et s'opposer à la déformation du fluide comme le souligne l'opérateur laplacien (section A.3) [23, pp.272-273].
- La constante \vec{g} correspond à l'accélération de pesanteur s'appliquant en tout point du fluide et dont la valeur admise est $9,80665 m.s^{-2}$ sur Terre selon le bureau international des poids et mesures [39, p.48]. Dans la littérature portant sur la forme incompressible des équations de Navier-Stokes, ce terme représente souvent la somme de toutes les accélérations, y compris la gravité, qui agissent sur le fluide [30]. Cependant, par souci de lecture, nous choisissons de souligner ce terme en le séparant des autres accélérations extérieures regroupées sous la dernière composante \vec{a}_{ext} .

Enfin, l'équation (1.6) est appelée la contrainte de divergence zéro qui, comme son

1.3. DIFFÉRENTES APPROCHES POUR SIMULER DES FLUIDES

nom l'indique, pose la contrainte que l'équation (1.5) puisse être appliquée sous cette forme seulement si le fluide est incompressible, c'est-à-dire que, si l'on considère un volume de fluide à un point de l'espace, toute quantité de fluide entrant dans ce volume doit faire ressortir la même quantité de fluide. Cette supposition est généralement admise pour des fluides dont le nombre de Mach M_a reste inférieur à 0,3 [2, p.64]. Celui-ci est donné d'après J.D. Anderson [2, p.39] comme étant

$$M_a = \frac{\|\vec{u}_{max}\|}{c_s}, \quad (1.8)$$

où $\|\vec{u}_{max}\|$ est la norme de la vitesse maximale atteinte en tout point du fluide et c_s est la célérité du son dans le fluide considéré. En admettant c_s comme étant approximativement 340 m.s^{-1} dans l'air [2, p.571] et 1500 m.s^{-1} dans l'eau [16], et selon l'équation du nombre de Mach (1.8), la contrainte de divergence zéro (équation (1.6)) est valide si la vitesse maximale du champ de vitesse appartenant au fluide ne dépasse pas 102 m.s^{-1} dans l'air et 450 m.s^{-1} dans l'eau. Sachant que la vitesse maximale du souffle d'un sèche-cheveux dans l'air, que nous assimilons à un souffleur, est d'environ 41 m.s^{-1} , ce qui est largement inférieur à 102 m.s^{-1} , le fluide sera considéré comme incompressible dans le reste de nos travaux.

Pour finir, l'équation de continuité caractérise le transport d'une quantité scalaire ou vectorielle quelconque à travers un flux [2, pp.127-132], aussi appelée loi de conservation de la masse, et se présentant ainsi sous sa forme différentielle :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0, \quad (1.9)$$

avec ρ , la densité du fluide et \vec{u} , la vitesse du fluide.

1.3 Différentes approches pour simuler des fluides

En informatique graphique, la simulation des fluides basée sur la physique est principalement divisée en deux sous-familles d'approches : l'approche eulérienne (sous-section 1.3.1) et l'approche lagrangienne (sous-section 1.3.2).

1.3. DIFFÉRENTES APPROCHES POUR SIMULER DES FLUIDES

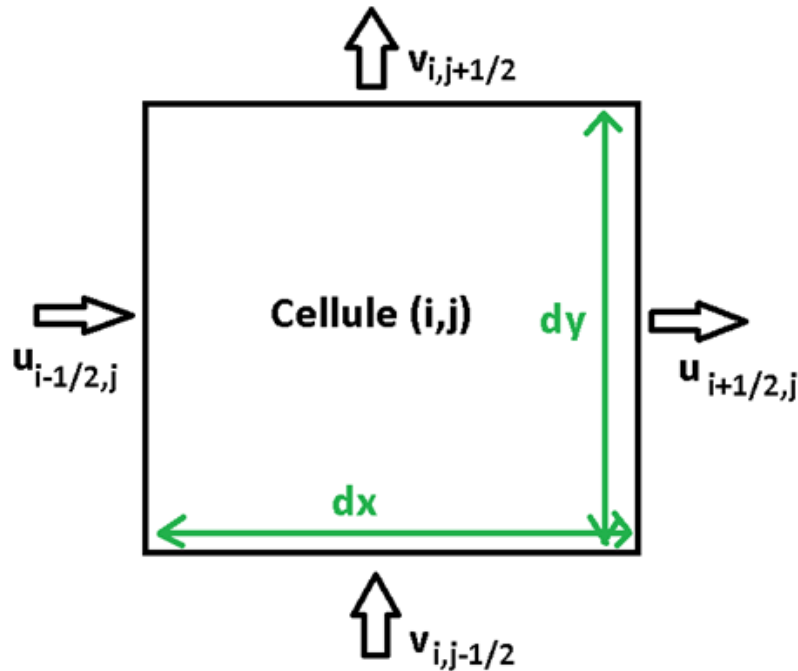


Figure 1.1 – Anatomie d’une cellule de coordonnée (i, j) dans la grille eulérienne. dx et dy sont les longueurs des côtés de la cellule dans les coordonnées x et y respectivement. Les valeurs de la vitesse du fluide \vec{u} et \vec{v} sont obtenues sur les côtés de la cellule, d’où leurs coordonnées décalées de $1/2$. Les flèches à côté des vitesses indiquent le sens du flux de fluide dans les cellules. Si le flux va dans le sens inverse de la flèche, la vitesse associée sera négative.

1.3.1 Approche eulérienne

Le premier simulateur de fluide par approche eulérienne en informatique graphique utilisant la méthode des différences finies (voir [section A.4](#)) fut implémenté par N. Foster et D. Metaxas [20] en s’inspirant des travaux de F.H. Harlow et J.E. Welch [24]. Nous référons ces articles si le lecteur souhaite approfondir ses connaissances puisque les méthodes dérivant de l’approche eulérienne sortent du cadre de nos travaux. Cette méthode consiste à échantillonner l’espace de simulation et non le fluide, puis à observer la quantité de fluide entrant et sortant de chacune des cellules de l’espace en utilisant la méthode des différences finies. L’anatomie d’une cellule en 2D est montrée dans la [figure 1.1](#), mais il est à noter que celle-ci peut facilement être extrapolée en 3D.

1.3. DIFFÉRENTES APPROCHES POUR SIMULER DES FLUIDES

Toutes les quantités des champs scalaires et vectoriels appartenant au fluide sont alors attribuées à chacune des cellules de la grille, tandis que le fluide est représenté par ses constantes comme sa densité de repos ρ_0 . Cela présente l'avantage de simuler le fluide comme étant explicitement incompressible par construction puisque la notion de champ de densité variant dans le temps est inexistante, car fixé à une constante [13, pp.6-7]. La contrainte de divergence zéro (équation (1.6)) est donc explicitement respectée. Cependant, puisque le repère de l'observateur reste fixe et ne suit pas le mouvement, et selon la règle de la dérivée en chaîne [9, p.8], la dérivée particulaire (équation (1.7)) peut être réécrite

$$\frac{d\vec{u}}{dt} = \frac{\partial\vec{u}}{\partial t} + \frac{\partial\vec{u}}{\partial x}u_x + \frac{\partial\vec{u}}{\partial y}u_y + \frac{\partial\vec{u}}{\partial z}u_z,$$

ce qui donne

$$\frac{d\vec{u}}{dt} = \frac{\partial\vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \cdot \vec{u}, \quad (1.10)$$

où $(\vec{u} \cdot \nabla) \cdot \vec{u}$ est le terme d'advection du fluide retrouvé dans la [sous-section 1.1.1](#).

Enfin, les équations (1.5) et (1.10) sont combinées pour arriver à la forme incompressible des équations de Navier-Stokes exprimées dans le repère eulérien telle que

$$\frac{\partial\vec{u}}{\partial t} = -(\vec{u} \cdot \nabla) \cdot \vec{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\vec{u} + \vec{g} + \vec{a}_{ext}. \quad (1.11)$$

Il est à noter ici que la composante de l'advection (équation (1.1)) n'est pas éliminée. Par conséquent, l'approche eulérienne tend à dissiper le mouvement du fluide en diffusant ses caractéristiques physiques, R. Bridson décrit ce phénomène comme étant la diffusion numérique [9, pp.37-39].

1.3.2 Approche lagrangienne

La méthode la plus célèbre de l'approche lagrangienne pure est hydrodynamique des particules lissées (HPL). Contrairement à l'approche eulérienne, cette approche consiste à échantillonner le fluide en particules et non l'espace de simulation.

Avant de continuer, il est important de souligner que le terme particules ne doit pas être confondu avec les molécules du fluide présentant un mouvement brownien, mais

1.3. DIFFÉRENTES APPROCHES POUR SIMULER DES FLUIDES

un échantillonnage spatial du fluide [26], c'est-à-dire chaque particule HPL possède un volume défini et porte les différentes quantités physiques, scalaires et vectorielles, caractéristiques du fluide à la position spatiale où celle-ci est positionnée. Puisqu'une particule est étudiée relativement à ses voisines, le repère de l'observateur se place et suit naturellement le mouvement de celle-ci. Par conséquent, le terme d'advection, donné dans la forme incompressible des équations de Navier-Stokes en approche eulérienne (1.11), disparaît, car la vitesse relative vis-à-vis d'une particule à elle-même est toujours nulle.

La dérivée particulaire (équation (1.7)) en repère lagrangien se réécrit donc

$$\frac{DA}{Dt} = \frac{dA}{dt} = \frac{\partial A}{\partial t}.$$

L'équation (1.5) exprimée en coordonnées lagrangiennes devient donc

$$\frac{\partial \vec{u}}{\partial t} = \frac{d\vec{u}}{dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{u} + \vec{g} + \vec{a}_{ext}. \quad (1.12)$$

Les premiers à proposer cette méthode furent L.B. Lucy [29], R.A. Gingold et J.J. Monaghan [22] afin de résoudre des problèmes d'astrophysique. Entre-temps, J.J. Monaghan [32] explicite les bases mathématiques et physiques, puis propose HPL pour simuler des mouvements aérodynamiques comme travaux futurs. La méthode fut ensuite utilisée pour faire des simulations de flamme explosive [47]. Néanmoins, il s'agit de l'implémentation faite par M. Müller et al. [30] qui a popularisé HPL en informatique graphique. Cette implémentation de base, dont les fondements sont explicités dans le [chapitre 2](#), repose sur la résolution d'une équation d'état afin de calculer le gradient du champ de pression retrouvé dans l'équation (1.12).

Enfin, puisque la densité selon HPL est une quantité scalaire qui fluctue contrairement aux méthodes issues de l'approche eulérienne, dont la densité est fixée à une constante par construction, le défi de ces dernières années fut de maintenir celle-ci approximativement égale à la densité cible du fluide ρ_0 moyennant une erreur de compressibilité acceptable comme explicité dans la [section 2.1](#). M. Becker et M. Teschner [11] proposent la méthode HPL Faiblement Compressible (abrégée HPLFC) en se basant sur la célérité du son dans le fluide considéré pour admettre cette erreur de

1.3. DIFFÉRENTES APPROCHES POUR SIMULER DES FLUIDES

compressibilité en dessous de 1% (voir la [sous-section 2.2.5.3](#)). Néanmoins, B. Solenthaler et R. Pajarola [44] décrivent que le fluide est rendu plus réactif à la moindre variation de densité, et oblige le pas de temps de la simulation Δt de diminuer pour respecter les conditions de stabilité de Courant-Friedrichs-Lewy (voir la [section 2.3](#)), ce qui augmente ainsi le temps global de simulation. Ils proposent la méthode HPL Incompressible Predictif-Correctif (abrégée HPLIPC) [44] en se basant sur les travaux de S.J. Cummins et M. Rudman [14] et S. Shao et al. [42] en réponse aux désavantages de HPLFC (voir la [sous-section 2.2.5.4](#)). Cette proposition a permis à HPL de devenir la méthode de choix pour simuler les effets d'éclaboussure réalistes [23, pp.291-292]. Par la suite, Ihmsen et al. [25] proposent HPL Implicitement Incompressible (abrégée HPLII) résolvant une Équation de Pression de Poisson (abrégée EPP) pour forcer l'invariance de densité. Cela permet ainsi d'augmenter davantage la stabilité de la simulation et la convergence du résolveur de pression, amenant une augmentation du pas de temps de la simulation [25] (voir la [sous-section 2.2.5.5](#)). Enfin, J. Bender et D. Koschier [4] proposent HPL Sans Divergence (abrégée HPLSD) résolvant deux EPPs avec des termes d'entrées différents pour réduire davantage l'erreur de divergence responsable de la compressibilité du fluide. Cette nouvelle méthode consomme moins de ressource mémoire et se présente plus stable, permettant ainsi un Δt plus important encore, vis-à-vis de HPLII (voir la [sous-section 2.2.5.7](#)) [4].

1.3.3 Approches mixtes

Dans l'approche eulérienne, un observateur regarde évoluer le fluide depuis un point fixe dans l'espace. Le fluide est alors incompressible par construction (voir la [sous-section 1.3.1](#)), mais a tendance à entraîner des diffusions numériques en raison de la composante d'advection présente dans la représentation eulérienne de la forme incompressible des équations de Navier-Stokes (équation (1.11)) [9, pp.37-39], mais aussi une réduction excessive du pas de temps de la simulation afin de respecter les conditions de stabilité de Courant-Friedrichs-Lewy (voir la [section 2.3](#)) directement dépendante de la longueur des côtés des cellules de la grille [40, 45].

Dans l'approche lagrangienne, l'observateur regarde évoluer le fluide autour de lui alors qu'il voyage avec la particule de fluide qu'il observe [40]. Cela a pour consé-

1.3. DIFFÉRENTES APPROCHES POUR SIMULER DES FLUIDES

quence d'éliminer la composante advection des équations de Navier-Stokes [26, pp.6-7], et donc des instabilités et artefacts amenés par les diffusions numériques, mais au détriment que certaines caractéristiques du fluide, comme la densité, peuvent varier et amener à des compressions et des dilatations locales du fluide.

Les approches mixtes ont pour objectif de combiner les deux approches tout en conservant les avantages sans les inconvénients. Par exemple, le schéma semi-lagrangien (abrégié SLS, *Semi-Lagrangian Scheme*) est parfois utilisé en prédiction météorologique [40]. SLS utilise une grille eulérienne de la même manière que l'approche eulérienne utilisant la méthode des différences finies (voir la [sous-section 1.3.1](#)), mais emploie une approche lagrangienne pour calculer l'advection [23, pp.301-319]. Plusieurs méthodes semi-lagrangiennes ont été proposées en informatique graphique et dont la différence repose surtout sur la manière dont l'advection est calculée, intégrée et reprojctée dans la grille eulérienne. La revue de ces méthodes sort du cadre de notre mémoire et nous n'en faisons qu'un résumé succinct sans entrer dans les détails, mais si le lecteur le souhaite, nous référons les articles de A. Staniforth et J. Côté [40] introduisant une revue des bases nécessaires pour comprendre comment l'advection semi-lagrangienne est effectuée, et l'article de J. Stam [45], introduisant la méthode nommée Fluides Stables (*Stable Fluids*), considérée comme une implémentation semi-lagrangienne extrêmement fructueuse employée en informatique graphique [19], et que nous pensons être l'une des méthodes de référence dans ce domaine.

Dans la méthode Fluides Stables [45], la forme incompressible des équations de Navier-Stokes (équations (1.11) et (1.6)) sont combinées en une seule équation telle que

$$\frac{\partial \vec{u}}{\partial t} = P(-(\vec{u} \cdot \nabla) \cdot \vec{u} + \nu \nabla^2 \vec{u} + \vec{a}_{ext}), \quad (1.13)$$

où P est un opérateur de projection projetant un champ de vitesse pour amener à respecter la contrainte de divergence zéro (équation (1.6)), et \vec{a}_{ext} est l'agrégation de toutes les accélérations extérieures s'appliquant sur le fluide.

Sous cette forme, le terme de la composante de pression présente dans la forme incompressible des équations de Navier-Stokes, $-\frac{1}{\rho} \nabla p$, disparaît car $P \nabla p = 0$ [45].

J. Stam [45] propose ainsi une méthode résolvant l'unique équation de Navier-Stokes (1.13) en composant des opérateurs prenant en entrée le champ de vitesse

1.3. DIFFÉRENTES APPROCHES POUR SIMULER DES FLUIDES

sortant des opérateurs précédents, telle que

$$\vec{w}_0(\vec{x}) \xrightarrow{\text{ajoute } \vec{a}_{ext}} \vec{w}_1(\vec{x}) \xrightarrow{\text{advection}} \vec{w}_2(\vec{x}) \xrightarrow{\text{applique la viscosité}} \vec{w}_3(\vec{x}) \xrightarrow{\text{applique P}} \vec{u}(\vec{x}), \quad (1.14)$$

avec $\vec{w}_i(\vec{x})$ le vecteur de vitesse à la position \vec{x} du champ de vitesse intermédiaire et $\vec{u}(\vec{x})$ le vecteur de vitesse finale respectant la contrainte de divergence zéro (équation (1.6)).

D’après l’équation (1.14), la première étape de simulation consiste donc à intégrer \vec{a}_{ext} au champ de vitesse dans la grille eulérienne, soit

$$\vec{w}_1(\vec{x}) = \vec{w}_0(\vec{x}) + \Delta t \vec{a}_{ext}. \quad (1.15)$$

Puis la seconde étape consiste à appliquer la composante d’advection. Pour ce faire, une particule sera créée à chaque position \vec{x} au centre de chaque cellule de la grille eulérienne. Puis J. Stam [45] utilise une méthode des caractéristiques pour retrouver la vitesse de la particule en examinant la vitesse qu’elle possédait au temps $t - \Delta t$ précédent et qui l’a amenée à la position \vec{x} courant. Pour cela, $\vec{x}(t - \Delta t) = \vec{x}(t) - \Delta t \vec{w}_1(\vec{x}(t))$. Une interpolation de la vitesse à la nouvelle position de cette particule, $\vec{x}(t - \Delta t)$, est réalisée en interpolant les vitesses aux points de la grille eulérienne, ou celle des autres particules. Le choix de la méthode d’interpolation a une importance sur la précision et l’efficacité de la méthode [40]. La méthode Fluides Stables utilise une interpolation linéaire [45] et obtient le champ de vitesse intermédiaire $\vec{w}_2(\vec{x}(t))$.

La troisième étape consiste à appliquer la composante de viscosité $\nu \nabla^2 \vec{w}_2(\vec{x}(t))$. La même méthode de discrétisation de l’opérateur laplacien, utilisée par N. Foster et D. Metaxas [20], pourra être utilisée pour des fluides peu visqueux [45].

La dernière étape, ayant pour objectif de rendre le système incompressible, consiste à résoudre une équation de Poisson telle que

$$\begin{aligned} \nabla^2 q &= \nabla \cdot \vec{w}_3(\vec{x}(t)), \\ \vec{u}(\vec{x}(t)) &= \vec{w}_3(\vec{x}(t)) - \nabla q, \end{aligned} \quad (1.16)$$

avec q un champ scalaire quelconque du fluide.

Nous référons le lecteur à l’article original de J. Stam [45] décrivant la méthode

1.3. DIFFÉRENTES APPROCHES POUR SIMULER DES FLUIDES

Fluides Stables pour la résolution de l'équation (1.16) ainsi que toutes démonstrations des concepts fournis précédemment.

1.3.4 Approches non basées sur la physique

Bien que cela sort du cadre de nos travaux, il est important de faire mention de l'existence grandissante en informatique graphique d'une autre catégorie de méthodes pour simuler les fluides. N'appartenant à aucune des deux approches basées sur la physique (voir les sous-section 1.3.1, sous-section 1.3.2 et sous-section 1.3.3); ces méthodes se basent sur le domaine de l'intelligence artificielle, telles que les réseaux de neurones ou les méthodes d'apprentissage. Par exemple, Y. Xie et al. [50] proposent de générer un champ de vecteurs physiques variant dans le temps en passant un champ de vitesse et un champ de vorticit e comme param etres d'entr ees d'un r eseau de neurones. P. Ma et al. [36] proposent de conserver les mod eles physiques classiques bas es sur les  equations de Navier-Stokes ( equations (1.5) et (1.6)), mais entra ınent un syst eme de r eseau de neurones pour g en erer les forces de contact aux fronti eres solides↔fluides et s'assurer que le fluide  epouse la forme du corps rigide  a son contact.

Outre le fait que ces m ethodes simulent la physique sans utiliser de mod ele complet, elles r eduisent de fa con non n egligeable le temps de simulation. Par exemple, S. Sato et al. [41] proposent de mod eliser des syst emes turbulents en combinant les donn ees d'une simulation pr ec edente  a basse r esolution et une image source id eale pour produire un mod ele de fluide en turbulence visuellement correct, sans payer le c ot  du calcul du mod ele physique sous-jacent. Ces m ethodes permettent aussi de produire des simulations de flux turbulent reproduisant la r ealit e avec une grande fid elit e selon J.N Kutz [27]. N eanmoins, celui-ci  enonce  egalement que ces m ethodes ont besoin d' etre entra ınees sur un jeu de donn ee de tr es haute qualit e et mod elisant une tr es grande vari et e de situations pour obtenir un tel niveau de r ealisme; et produire ces donn ees d'entra ınement se r ev ele  etre un processus extr emement c oteux en ressource de calcul et en temps.

Chapitre 2

L'approche HPL

2.1 Fondements

2.1.1 Méthode d'interpolation

HPL échantillonne spatialement le fluide en de multiples unités discrètes de volume appelées particules. Celles-ci sont porteuses des différentes quantités des champs scalaires et vectoriels caractérisant le fluide, comme la masse, la densité, la pression et la vitesse, qui sont calculées à n'importe quelle position du champ avec une formule d'interpolation définie au cœur de HPL [32], telle que

$$A_i = \int \frac{A_j}{\rho(\vec{x}_j)} W_{ij}(\vec{r}, h) dm_j \approx \sum_j A_j \frac{m_j}{\rho_j} W_{ij}(\vec{r}, h), \quad (2.1)$$

avec $W_{ij}(\vec{r}, h)$, une fonction d'interpolation dont la longueur de lissage est h . Par souci de lisibilité, la notation $W_{ij}(\vec{r}, h)$ est abrégée W_{ij} dans le reste de nos travaux. A étant la quantité interpolée du champ scalaire ou vectoriel, aux positions quelconques \vec{x}_i et \vec{x}_j des points considérés i et j , et $\vec{r} = \vec{x}_i - \vec{x}_j$. Enfin, m_j est la masse de la particule j donnée par $\rho(\vec{x}_j)dV_j = dm_j$ avec $\rho(\vec{x}_j)$ la densité à la position \vec{x}_j du champ scalaire de densité, abrégée ρ_j , et V_j le volume à la position \vec{x}_j .

La longueur de lissage h contrôle l'étendue du lissage de la fonction W_{ij} , c'est-

2.1. FONDEMENTS

à-dire le poids que A_j aura sur A_i lors de l'interpolation étant donnée sa position \vec{x}_j . Pour plus de détails, une démonstration amenant à la construction de l'équation (2.1) à partir de la distribution de Dirac est décrite par D. Koschier et al. [26]. Deux propriétés pour le noyau W_{ij} découlent directement de cette démonstration :

- La condition de normalisation assure que l'intégration du noyau sur tout le domaine soit égale à 1, donc que W_{ij} ne démultiplie pas l'échelle de la quantité A_i en \vec{x}_i , soit

$$\int_{\mathbb{R}} W_{ij} dx_j = 1. \quad (2.2)$$

Par conséquent, d'après D. Koschier et al. [26] :

$$\lim_{h \rightarrow 0} W_{ij} = \delta \left(\frac{\|\vec{r}^*\|}{h} \right).$$

- La condition de support compact force le noyau W_{ij} en \vec{r}^* à être borné entre $-h$ et h , soit

$$W_{ij}(\vec{r}^*, h) = 0 \quad \text{si } \|\vec{r}^*\| > h. \quad (2.3)$$

Enfin, deux autres conditions facultatives s'ajoutent pour éviter certains artefacts en simulation physique [26].

- La condition de positivité assurant que les champs physiques restent cohérents en ne changeant pas de signe, soit

$$W_{ij} \geq 0. \quad (2.4)$$

- La condition de symétrie spécifiant que l'apport de la quantité scalaire ou vectorielle en \vec{x}_j doit rester isotrope, c'est-à-dire que l'apport ne doit pas dépendre de la direction entre les particules, mais uniquement de la distance entre \vec{x}_j et \vec{x}_i , soit

$$W_{ij}(\vec{x}_i - \vec{x}_j, h) = W_{ij}(\vec{x}_j - \vec{x}_i, h).$$

Pour finir, une condition finale est introduite pour permettre l'utilisation de W_{ij} dans les équations de Navier-Stokes (1.5) et (1.6) : son noyau doit être continuellement dérivable au moins une fois sur l'ensemble du domaine $[-h, h]$.

2.1. FONDEMENTS

M.B. Liu et G.R. Liu [28] proposent plusieurs noyaux HPL viables respectant toutes les contraintes énumérées. Parmi ceux-ci, les noyaux spline cubiques produisent de bonnes estimations de la densité comme énoncé par D.J. Price [37, pp.761-762]. Nous avons donc choisi le noyau spline cubique utilisé par D. Koschier et al. [26], tel que :

Soit $q = \|\vec{r}\|/h$,

$$W(\vec{r}, h) = \sigma_d \begin{cases} 6(q^3 - q^2) + 1 & \text{si } 0 \leq q \leq \frac{1}{2}, \\ 2(1 - q)^3 & \text{si } \frac{1}{2} < q \leq 1, \\ 0 & \text{autrement,} \end{cases} \quad (2.5)$$

avec d égal au nombre de dimensions de la simulation et σ_d le coefficient de normalisation nécessaire pour respecter la condition de normalisation (équation (2.2)) telle que

$$\sigma_d = \begin{cases} \frac{40}{7\pi h^2} & \text{si } d = 2, \\ \frac{8}{\pi h^3} & \text{si } d = 3. \end{cases}$$

Après avoir dérivé une fois, le gradient du noyau spline cubique devient

$$\nabla W(\vec{r}, h) = \sigma_d^\nabla \begin{cases} q(3q - 2) & \text{si } 0 \leq q \leq \frac{1}{2}, \\ -(1 - q)^2 & \text{si } \frac{1}{2} < q \leq 1, \\ 0 & \text{autrement,} \end{cases} \quad (2.6)$$

avec σ_d^∇ jouant le même rôle que σ_d mais pour le noyau dérivé à l'ordre 1 tel que

$$\sigma_d^\nabla = \begin{cases} \frac{240}{7\pi h^2} & \text{si } d = 2, \\ \frac{48}{\pi h^4} & \text{si } d = 3. \end{cases}$$

La [figure 2.1](#) montre le tracé graphique de ce noyau et de son gradient.

À noter que la dérivée seconde utilisée pour le laplacien n'est pas nécessaire, car celle-ci peut être discrétisée en un gradient en appliquant l'équation (2.13) (voir la [sous-section 2.2.4.1](#)).

2.1. FONDEMENTS

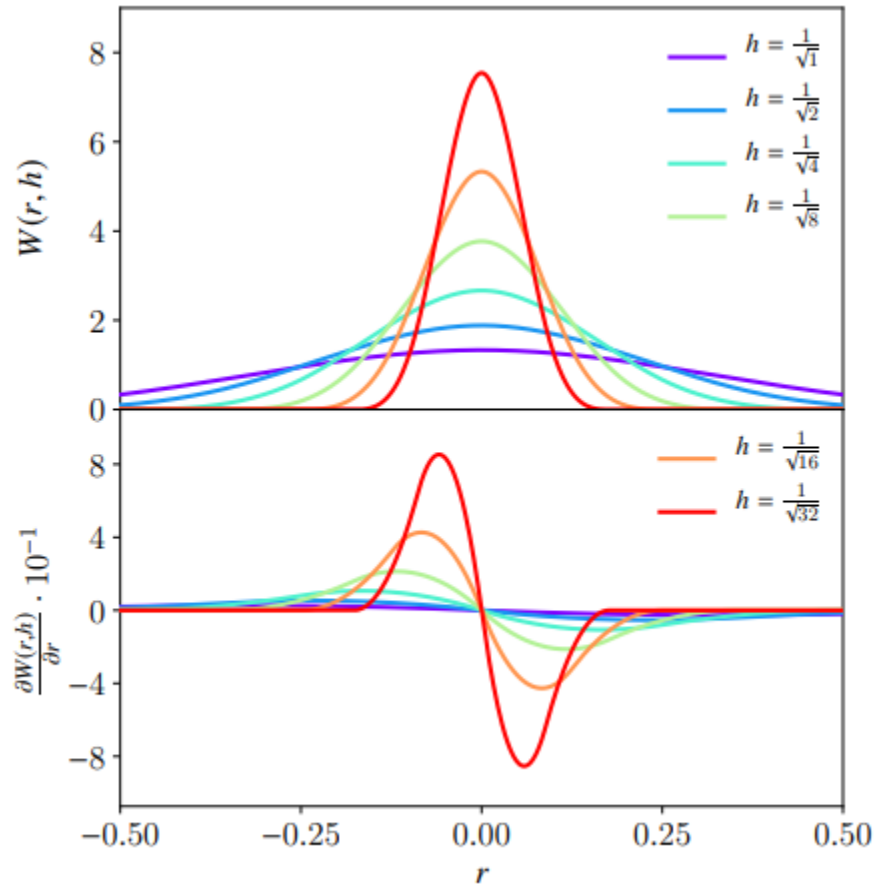


Figure 2.1 – Tracé graphique du noyau spline cubique et son gradient. La figure est extraite de l'article [26]. L'axe des abscisses représente $r = \|\vec{x}_i - \vec{x}_j\|$.

2.1. FONDEMENTS

2.1.2 Discrétisation de l'opérateur gradient appliquée à l'interpolation HPL

Pour finir avec les fondements constituant le concept de HPL, il est important de souligner une propriété de l'interpolation au cœur de HPL, valable seulement si les noyaux choisis respectent les conditions énumérées dans la [sous-section 2.1.1](#).

En appliquant l'opérateur gradient à la définition de l'interpolation en domaine discret (équation (2.1)) utilisée par HPL, alors

$$\nabla A_i \approx \sum_j A_j \frac{m_j}{\rho_j} \nabla W_{ij}. \quad (2.7)$$

En étudiant la série de Taylor, l'équation (2.7) doit être modifiée en sa forme différentielle pour corriger l'erreur induite par de mauvaises approximations et dont la démonstration peut se retrouver dans l'article [26], soit

$$\nabla A_i \approx \sum_j (A_j - A_i) \frac{m_j}{\rho_j} \nabla W_{ij}. \quad (2.8)$$

Bien que la forme différentielle (équation (2.8)) amène plus de précision, elle ne respecte pas la loi de la conservation du moment et peut amener à une simulation instable [37]. Une autre forme découlant de l'équation (2.7), nommée symétrique et dont la démonstration, donnée par M. Desbrun et M.-P. Gascuel [17, Appendix B - Pressure forces], est proposée telle que

$$\nabla A_i \approx \rho_i \sum_j m_j \left(\frac{A_j}{\rho_j^2} + \frac{A_i}{\rho_i^2} \right) \nabla W_{ij}. \quad (2.9)$$

Bien qu'étant moins précise que la forme différentielle (équation (2.8)), la forme symétrique (équation (2.9)) présente l'avantage de conserver le moment linéaire et angulaire des forces obtenues, ce qui est un critère essentiel à toutes simulations robustes [37, pp.773-776]. Selon l'article [26], il est donc conseillé d'utiliser la forme symétrique pour calculer les quantités scalaires ou vectorielles affectant directement le mouvement des particules comme les forces et les accélérations, tandis que la forme différentielle pourra être utilisée pour tout le reste.

2.2 Le concept de la simulation HPL

Dans cette partie, nous décrivons les cinq grandes étapes du déroulement en ordre chronologique de la simulation des fluides selon HPL, c'est-à-dire l'[Initialisation](#), la [Recherche du voisinage](#), la [Mise à jour du champ scalaire de densité](#), [Résolution de la forme incompressible des équations de Navier-Stokes](#) et enfin, la [Méthode d'Euler semi-implicite](#). Cependant, compte tenu des diverses variantes utilisées pour la résolution de la composante de pression retrouvée dans la forme incompressible des équations de Navier-Stokes, celle-ci est placée dans une sous-section séparée ([sous-section 2.2.5](#)) afin de faciliter la lecture.

2.2.1 Initialisation

Certaines initialisations doivent être réalisées avant de commencer la boucle principale de simulation.

Tout d'abord, soit d_p le diamètre d'une particule quelconque, il est recommandé d'initialiser la longueur de lissage du noyau h tel que $h = 2d_p$ [[11](#), [26](#), [32](#), [35](#)]. Pour notre part, nous introduisons une constante e_h , tel que

$$h = e_h d_p, \tag{2.10}$$

afin de tester la pertinence de cet estimé, et dont une valeur e_h de 2,5 déterminée empiriquement est préférable pour les gaz tandis que 2 est une bonne valeur pour les liquides. Toutefois, il est à noter que cette conclusion dépend des autres constantes caractérisant la simulation.

Une fois h initialisé, l'espace de simulation peut être partitionné avec une grille spatiale. Cette étape facultative, détaillée dans la [sous-section 2.2.2](#), permet d'optimiser tous les algorithmes de recherche de voisinage subséquents.

Pour finir, les particules du fluide et des solides sont générées dans le domaine (voir la [sous-section 3.5.1](#) pour la génération des particules des solides). Ensuite les particules de fluide en collision directe ou à l'intérieur des solides sont éliminées. Si cette étape est mal réalisée, une surévaluation de la densité provoquera un pic de pression amenant à une explosion ou une instabilité au début de la simulation. À

2.2. LE CONCEPT DE LA SIMULATION HPL

noter qu'il est possible d'optimiser cette étape en ne considérant uniquement que les particules du fluide à l'intérieur d'une boîte englobante autour du corps du solide. L'[algorithme 1](#) montre la génération des particules du fluide au début de la simulation dans un domaine considéré comme un parallélépipède rectangle.

Algorithme 1 Génération des particules de fluide dans le domaine

```
1: fonction GÉNÈREBLOCDEPARTICULES( $d_p$ , xMin, xMax, yMin, yMax, zMin,
   zMax, liste corps solides)
2:    $liste^{particules} \leftarrow [ ]$ 
3:    $rayon_d \leftarrow d_p/2, 0$ 
4:    $x \leftarrow xMin + rayon_d$ 
5:   pour  $x \leftarrow xMin + rayon_d; x \leq xMax - rayon_d; x \leftarrow x + d_p$  faire
6:     pour  $y \leftarrow yMin + rayon_d; y \leq yMax - rayon_d; y \leftarrow y + d_p$  faire
7:       pour  $z \leftarrow zMin + rayon_d; z \leq zMax - rayon_d; z \leftarrow z + d_p$  faire
8:         si Vecteur( $x, y, z$ ) pas dans corps solides alors ▷ Possible
           d'optimiser le calcul avec des boîtes englobantes.
9:          $liste^{particules}.ajoute(x, y, z)$ 
10:        fin si
11:      fin pour
12:    fin pour
13:  fin pour
14:  retourne  $liste^{particules}$ 
15: fin fonction
```

Chaque particule de fluide i générée possède toutes les quantités scalaires et vectorielles caractéristiques du fluide telles qu'une densité ρ_i , une masse m_i ou une vitesse \vec{v}_i , ... Plus de détails sur l'initialisation de ces différentes quantités sont fournis dans la [sous-section 3.5.2](#).

L'[algorithme 2](#) résume toutes les étapes décrites précédemment.

2.2.2 Recherche du voisinage

À chaque interpolation de l'une des quantités (scalaires et vectorielles) physiques de la particule, le voisinage de ladite particule doit être parcouru. L'implémentation naïve consiste à itérer sur chaque particule et voir si celle-ci se situe à une distance inférieure à la longueur de lissage h . Néanmoins, cet algorithme est en $\mathcal{O}(kn_p^2)$, avec

2.2. LE CONCEPT DE LA SIMULATION HPL

Algorithme 2 Initialisation du simulateur

```

1: fonction INITIALISATION MOTEUR
2:   Lit le fichier de configuration
3:   si  $e_h > 0$  alors           ▷ Si l'utilisateur a entré manuellement une valeur
4:      $h = e_h d_p$ 
5:   sinon
6:      $h = 2d_p$ 
7:   fin si
8:   Partitionne l'espace de simulation avec une grille spatiale (voir la sous-section 2.2.2)
9:   Échantillonne corps solides (voir la section 2.4)
10:  Construit les boîtes englobantes pour les corps solides
11:  GÉNÈREBLOCDEPARTICULES( $d_p, xMin, xMax, yMin, yMax, zMin, zMax$ ,
    corps solides)
12:  Effectue les initialisations propres aux différentes méthodes HPL (voir la sous-section 2.2.5)
13: fin fonction

```

k le nombre de fois où le voisinage est demandé par pas de temps de la simulation, et n_p le nombre de particules impliquées dans la simulation. Toutefois, puisque d'après la condition de support compact du noyau (équation (2.3)), la plupart des apports occasionnés disparaissent, il est possible de précalculer les particules voisines à une distance $\|\vec{r}\|$, telle que $\|\vec{r}\| \leq h$. La complexité de l'algorithme est alors réduite à une seule recherche par pas de temps ($k = 1$). Néanmoins, cela ne change rien à la complexité de la recherche initiale du voisinage qui reste en $\mathcal{O}(n_p^2)$. Pour optimiser celle-ci, une grille spatiale uniforme est utilisée, faisant le compromis entre la vitesse d'exécution de la simulation et la consommation mémoire. La grille spatiale enregistre la position et l'index de toutes les particules dans des cellules, dont la longueur de côté est égale à la longueur du noyau de lissage h . Sachant la position de la particule i aux coordonnées $\{x_i, y_i, z_i\}$, le triplet d'indice $\{a, b, c\}$ repérant la cellule contenant i est

$$\{a, b, c\} = \left\{ \left\lceil \frac{x_i}{h} \right\rceil, \left\lceil \frac{y_i}{h} \right\rceil, \left\lceil \frac{z_i}{h} \right\rceil \right\}.$$

Il est donc possible d'accélérer la recherche du voisinage de la particule i en ne considérant que les particules contenues dans la cellule $\{a, b, c\}$ et dans les vingt-six cellules

2.2. LE CONCEPT DE LA SIMULATION HPL

adjacentes pour une simulation en trois dimensions et huit cellules pour une simulation en deux dimensions, ce qui est moins coûteux que la recherche sur tout le domaine pour chaque particule. Peu importe où la particule considérée dans la [figure 2.2](#) soit placée dans sa cellule, le cercle de rayon h centrée sur celle-ci ne sortira jamais de l'espace délimité par les cellules marquées en vert ; h étant la longueur de lissage du noyau et choisi pour être la longueur des côtés des cellules. Il est donc inutile de considérer les particules contenues dans les cellules marquées d'une croix rouge.

Dans notre cas, le temps de simulation a été réduit de plus de 96% pour simuler une scène en cassure de barrage avec mille-sept-cent particules de fluide et dix-mille particules de solides, plus de détails sur les résultats obtenus sont fournis dans la [sous-section B.3.2](#).

2.2.3 Mise à jour du champ scalaire de densité

La seconde étape consiste à mettre à jour le champ scalaire de densité du fluide. En reprenant l'équation (2.7), mais en substituant A pour ρ , on obtient

$$\rho_i = \sum_j m_j W_{ij}. \quad (2.11)$$

2.2.4 Résolution de la forme incompressible des équations de Navier-Stokes

L'objectif est de résoudre la forme incompressible des équations de Navier-Stokes (1.12) et (1.6) pour obtenir le champ d'accélération \vec{a} de toutes les particules. Nous décrivons ici chaque terme ([Viscosité](#), [Gravité](#)) à l'exception de la composante de la pression, abordée séparément dans la [sous-section 2.2.5](#) afin de faciliter la lecture en raison des nombreuses variantes calculant celle-ci et implémentées dans le cadre de nos travaux.

2.2.4.1 Viscosité

La viscosité est responsable de générer des forces s'opposant à la déformation du fluide. Elle dissipe l'énergie du système en uniformisant le champ de vitesse comme le

2.2. LE CONCEPT DE LA SIMULATION HPL

montre l'opérateur laplacien [23, pp.272-273]. En d'autres termes, elle est responsable d'entraîner les particules voisines, dont la vitesse est plus faible, et de ralentir celles dont la vitesse est plus élevée que la particule considérée. Deux grandes approches pour calculer la viscosité existent [26] :

- L'approche implicite produit de meilleurs résultats pour des fluides hautement visqueux [26]. L'air et l'eau étant relativement peu visqueux, cette approche n'a

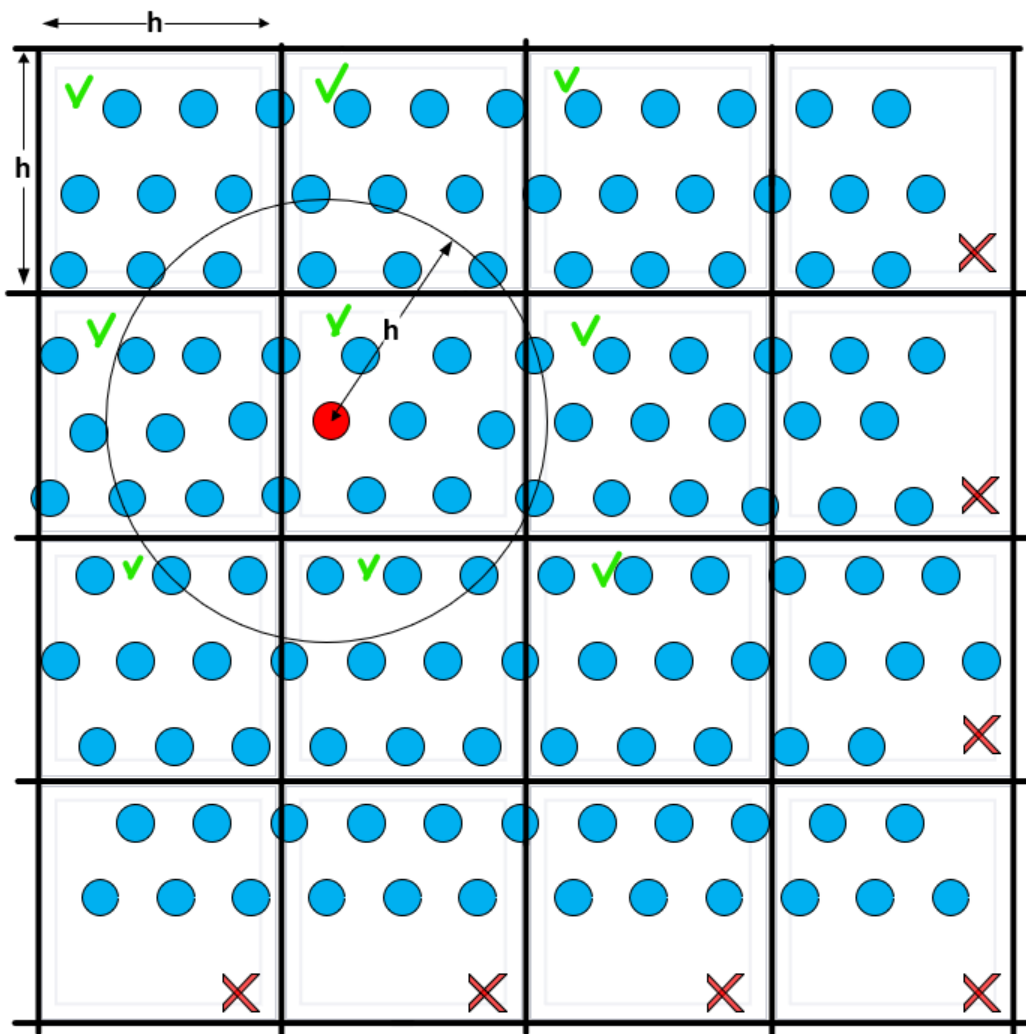


Figure 2.2 – Découpage de l'espace en grille uniforme pour une recherche optimale du voisinage. La particule considérée est montrée en rouge.

2.2. LE CONCEPT DE LA SIMULATION HPL

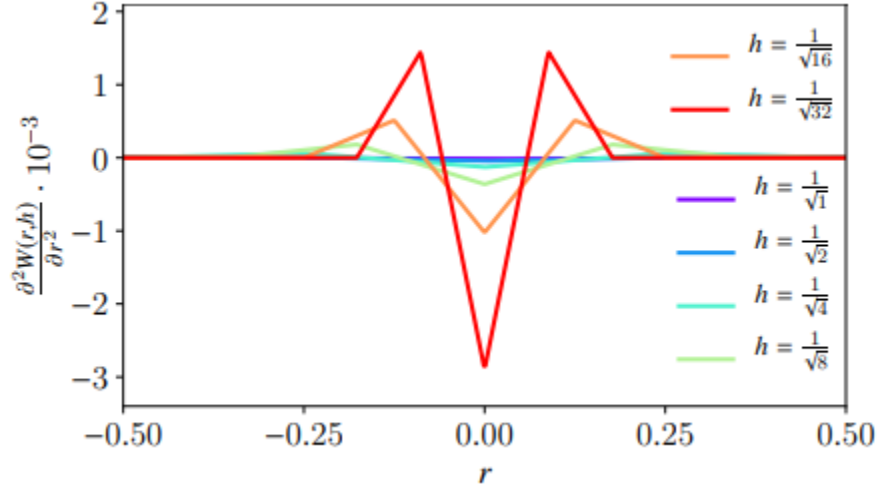


Figure 2.3 – Tracé graphique du laplacien du noyau spline cubique. La figure est extraite de l'article [26]. L'axe des abscisses représente $r = \|\vec{x}_i - \vec{x}_j\|$.

pas été abordée dans le cadre de nos travaux, mais le lecteur pourra se référer aux travaux de T. Takahashi et al. [46] pour davantage de détails.

- L'approche explicite résout directement le terme de viscosité $\nu \nabla^2 \vec{v}_i$ présent dans l'équation (1.12). En appliquant l'équation (2.7), cela donne :

$$\nabla^2 \vec{v}_i = \sum_j \frac{m_j}{\rho_j} \vec{v}_j \nabla^2 W_{ij}. \quad (2.12)$$

Néanmoins, J.J. Monaghan déclare que l'équation (2.12) est sensible au désordre des particules, et le laplacien d'un noyau de type gaussien, tel que le noyau spline cubique (équation (2.5)), s'annule et change de signe comme montré à la figure 2.3, ce qui ne respecte pas la condition de positivité (équation (2.4) à la sous-section 2.1.1). L'équation (2.12) peut donc amener à des incohérences dans le calcul des forces de viscosité [35].

À la place, L. Brookshaw [10] propose une approximation du laplacien par une combinaison d'un gradient et une méthode des différences finies, telle que

$$\nabla^2 \vec{v}_i \approx \sum_j \frac{m_j}{\rho_j} \vec{v}_j \nabla^2 W_{ij} \approx 2(d+2) \sum_j \frac{m_j}{\rho_j} \frac{\vec{v}_{ij} \cdot \vec{x}_{ij}}{\|\vec{x}_{ij}\|^2} \nabla W_{ij}, \quad (2.13)$$

2.2. LE CONCEPT DE LA SIMULATION HPL

avec d le nombre de dimensions de la simulation, $\vec{x}_{ij} = \vec{x}_i - \vec{x}_j$ et $\vec{v}_{ij} = \vec{v}_i - \vec{v}_j$.

J.J. Monaghan [35] introduit un terme $0,01h^2$ pour éviter des singularités dans la simulation. Si l'équation (2.13) est appliquée à l'équation (2.12), on obtient alors

$$\nabla^2 \vec{v}_i = 2(d+2) \sum_j \frac{m_j}{\rho_j} \frac{\vec{v}_j \cdot \vec{x}_{ij}}{\|\vec{x}_{ij}\|^2 + 0,01h^2} \nabla W_{ij}. \quad (2.14)$$

Dans un second temps, J.J. Monaghan [32] présente la méthode XHPL (X représentant le facteur inconnu) pour calculer la viscosité, telle que

$$\vec{\tilde{v}}_i = \vec{v}_i + \alpha \sum_j \frac{m_j}{\rho_j} (\vec{v}_j - \vec{v}_i) W_{ij}, \quad (2.15)$$

avec $\vec{\tilde{v}}_i$ la vitesse temporaire de la particule i après lissage par XHPL, et α un coefficient constant défini par l'utilisateur tel que $0 \leq \alpha < 1$. XHPL présente l'avantage de réduire le désordre des particules en réduisant directement la différence de vitesse entre deux particules, sans tenir compte de la trajectoire de celles-ci vis-à-vis de leur différence de position. Cependant, α est un coefficient non physique déterminé empiriquement par l'utilisateur [26, p.22].

2.2.4.2 Gravité

La composante de gravité dans la forme incompressible des équations de Navier-Stokes (1.12) est soulignée ici, car celle-ci joue le rôle spécial d'agrégateur des particules. En admettant que la gravité soit présente et dirigée vers le bas, la figure 2.4 montre que les particules de fluide se rassemblent vers le fond du domaine, ne laissant pas de vide en bas. Si la gravité est manquante, puisque selon la condition de support compact du noyau (équation (2.3)), deux particules situées à une distance supérieure à la longueur de lissage du noyau h n'appliquent aucune force de pression ni de viscosité entre elles, alors elles demeureront fixes ou conserveront un mouvement rectiligne uniforme selon la première loi de Newton. Du vide se formera donc autour de la balle pendant sa chute comme montré sur la figure 2.4.

2.2. LE CONCEPT DE LA SIMULATION HPL

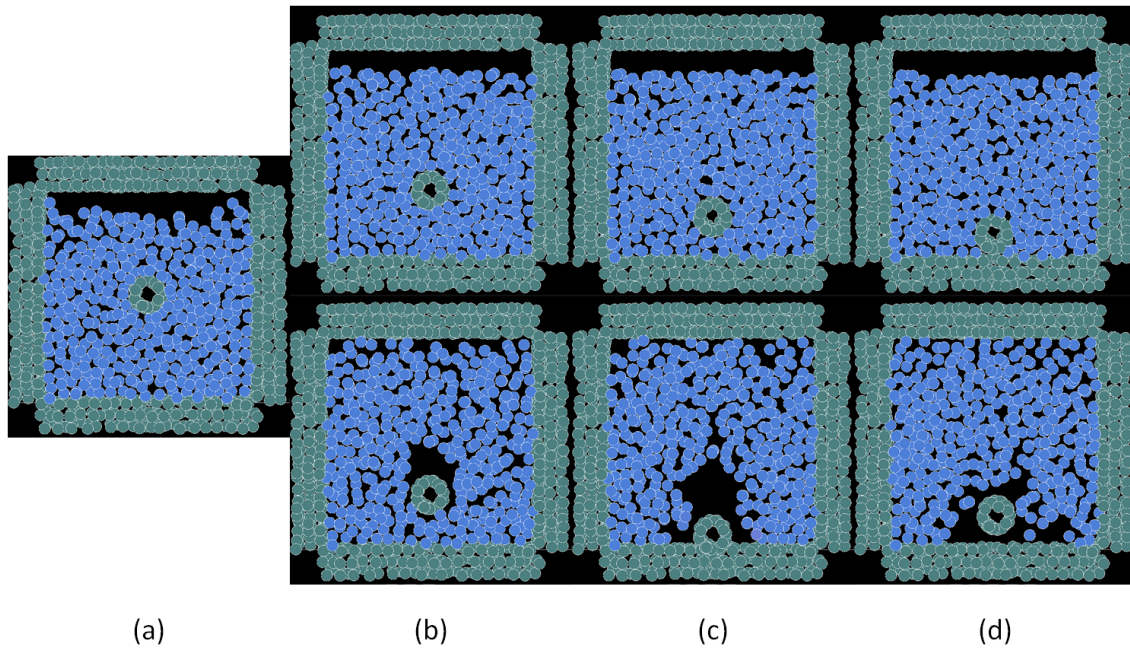


Figure 2.4 – Simulation avec et sans gravité d’une balle coulant dans l’eau. La méthode utilisée est HPLSD (voir la [sous-section 2.2.5.7](#)). Le plan proche (*near plane*) et le plan éloigné (*far plane*) ont été manipulés pour produire une coupe mince centrée sur la balle. La gravité est activée sur le fluide dans les images du haut, et enlevée dans les images du bas. (a) est une impression écran prise au point de départ de la simulation, soit $t = 0 s$. (b), (c) et (d) représentent l’état de la simulation à $t = 0,3 s$; $t = 0,4 s$ et $t = 0,6 s$ respectivement.

Enfin, la [figure 2.5](#) montre que la différence de poids induite par la gravité permet à la balle rouge, dont la densité ρ_r est plus élevée que celle du fluide ρ_0 , d’être davantage entraînée vers le fond. À l’inverse, la balle bleue avec ρ_b inférieure à ρ_0 est repoussée vers le sommet du domaine, tandis que la balle verte avec ρ_v égale à ρ_0 reste stable. Il apparaît sur les images du bas où la gravité est désactivée sur le fluide, que les trois balles coulent, peu importe leur densité. La gravité est donc nécessaire pour générer des forces de pression capables de s’opposer au poids de la balle et la faire remonter.

2.2. LE CONCEPT DE LA SIMULATION HPL

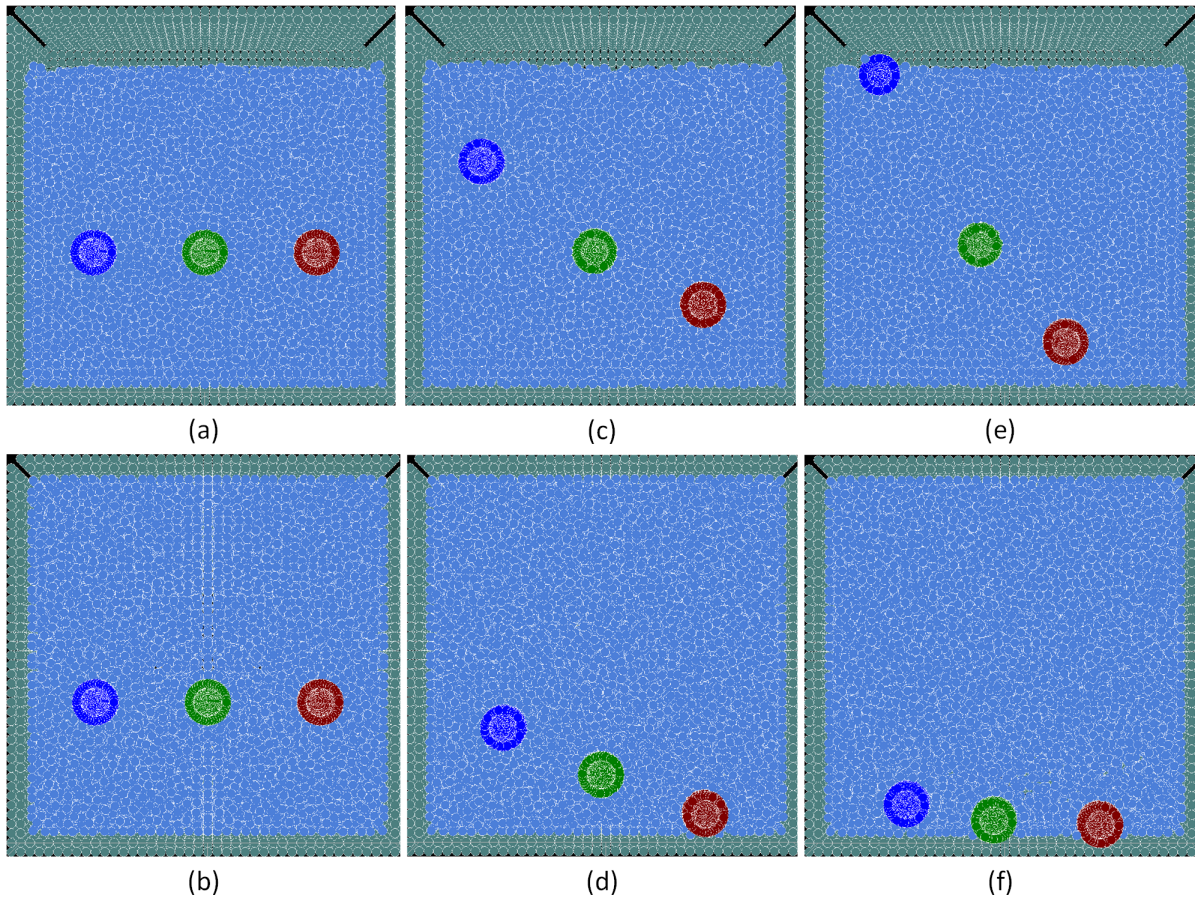


Figure 2.5 – Simulation de trois balles de densité différentes immergées dans de l’eau. La méthode utilisée est HPLSD. Le fluide possède une densité $\rho_0 = 1000 \text{ kg.m}^{-3}$. La balle bleue possède un $\rho_b = 400 \text{ kg.m}^{-3}$, la balle verte possède un $\rho_v = 1000 \text{ kg.m}^{-3}$ et la balle rouge possède un $\rho_r = 1400 \text{ kg.m}^{-3}$. La gravité activée en permanence sur la balle, est activée sur le fluide dans les images (a), (c) et (e), et désactivée uniquement sur le fluide dans (b), (d) et (f). (a) et (b) montre le moment de départ de la simulation, soit $t = 0 \text{ s}$; (c) et (d) à $t = 1,5 \text{ s}$; (e) et (f) à $t = 3 \text{ s}$.

2.2.5 Pression

La composante de pression dans la forme incompressible des équations de Navier-Stokes (1.12) maintient les particules à distance les unes des autres et empêche le fluide de se comprimer. Selon D. Koshier et al., la pression est proportionnelle à la déviation de volume. Le champ scalaire de pression peut être calculé localement, à

2.2. LE CONCEPT DE LA SIMULATION HPL

partir d'une déviation de densité ou à partir d'une divergence de vitesse, à l'aide d'une équation d'état ou d'une équation de Pression de Poisson (abrégée EPP) [26, p.11].

2.2.5.1 Équation d'état

Si le fluide est un gaz, la loi des gaz parfaits $p_i V_i = C$ peut être utilisée comme équation d'état [13, p.14], avec $C = nRT$ une constante dépendante de la quantité de matière n , de la constante universelle des gaz parfaits R et de la température du système T . Celle-ci peut être réécrite sous la forme $p_i = C \frac{\rho_i}{m_i}$. En admettant la masse d'une particule de fluide fixée à une constante m , alors

$$p_i = k \rho_i, \quad (2.16)$$

avec $k = \frac{C}{m}$ une autre constante appelée constante de raideur servant de gain pour moduler la force de pression dans le système. L'équation (2.16) est parfois retrouvée dans la littérature sous la forme

$$p_i = k(\rho_i - \rho_0), \quad (2.17)$$

avec ρ_0 la densité cible du fluide. L'équation (2.17) est souvent préférée à l'équation (2.16) car elle tend à ramener la densité autour de la densité cible.

Enfin, X. Chermain [13] conseille l'équation de Tait si le fluide est un liquide, soit

$$p_i = k_1 \left(\left(\frac{\rho_i}{\rho_0} \right)^{k_2} - 1 \right). \quad (2.18)$$

À noter que les équations (2.17) et (2.18) ne sont pas utilisées sous cette forme puisqu'une pression négative n'est pas tolérée dans la plupart des méthodes découlant de HPL, car celle-ci entraîne des instabilités numériques telles que les instabilités de traction (*tensile instabilities*) [4, 5, 25, 26, 28, 31, 34] [21, p.26] [32, p.558] [37, p.775]. Davantage de détails sont présentés dans la [section 3.4](#). Bien que diverses solutions pour éviter les instabilités de traction ont été formulées, telles que l'introduction d'une pression artificielle [34], nous avons utilisé la méthode commune en informatique graphique, présentée dans les méthodes [HPL Implicitement Incompressible \(HPLII\)](#)

2.2. LE CONCEPT DE LA SIMULATION HPL

et **HPL Sans Divergence (HPLSD)**, consistant à assigner zéro à la valeur de la pression négative afin de corriger les problèmes d'agrégations des particules à la surface des solides [4, 25]. En conséquence, la divergence de pression est forcée à 0 à l'intérieur des zones de pression négative, ce qui empêche la balle d'être aspirée dans le flux d'air généré par le souffleur, comme expliqué dans la [section 3.1](#).

L'équation de Tait (2.18) se réécrit donc

$$p_i = \max\left(k_1 \left(\left(\frac{\rho_i}{\rho_0} \right)^{k_2} - 1 \right), 0\right). \quad (2.19)$$

Enfin, les constantes de raideurs k_1 et k_2 contrôlent la déviation de la densité. Plus celles-ci augmentent, plus les forces de pressions engendrées augmentent et empêchent le fluide de se comprimer. Cependant, une valeur trop élevée provoquera des instabilités selon les conditions de Courant-Friedrichs-Lewy (abrégé CFL) décrite dans la [section 2.3](#). Le pas de temps Δt de la simulation devra alors être réduit, ce qui augmentera le temps global de la simulation. À l'opposé, une valeur trop faible permettra une plus grande déviation de la densité, amenant à des simulations non réalistes où la surface d'un liquide semble rebondir dans un cycle de compression et dilatation. Ce phénomène est mis en évidence dans la [figure 2.6](#) montrant une simulation utilisant la méthode HPL avec Équation d'État (HPLSE) [30] dans une scène en configuration de cassure de barrage. Il apparaît que le fluide se comprime et se dilate davantage lorsque les constantes de raideurs utilisées dans l'équation de Tait (2.19) sont réduites comme le témoigne la hauteur du fluide.

L'objectif des méthodes que nous avons implémentées est de calculer k_1 et k_2 pour rendre le fluide incompressible, tout en conservant Δt le plus élevé possible. Pour ce faire, ces méthodes vont, soit se baser sur une heuristique comme HPLFC, soit s'aider d'une EPP comme HPLSD.

2.2.5.2 Équation de Pression de Poisson (EPP)

L'idée générale derrière l'EPP est d'obtenir un champ d'accélération, associé au champ de pression, amenant un champ de vitesse qui, une fois advecté, permet d'obtenir un déplacement de telle sorte que celui-ci respecte la contrainte de divergence

2.2. LE CONCEPT DE LA SIMULATION HPL

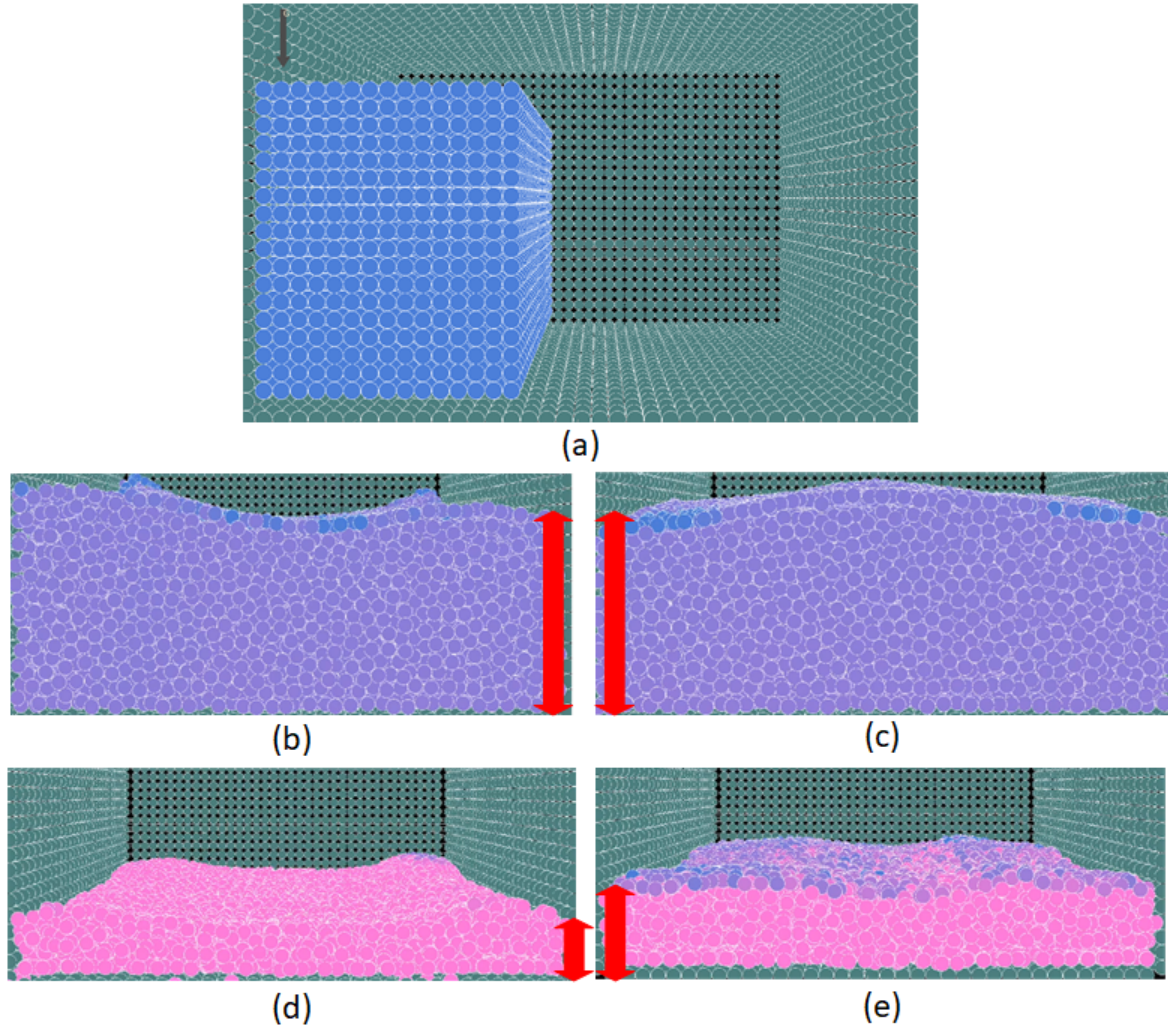


Figure 2.6 – Simulation HPLSE avec différentes constantes de raideur. (a) est obtenue au temps initial de simulation, soit $t = 0 s$. Les images (b) et (d) sont obtenues à $t = 0,5 s$ tandis que (c) et (e) sont obtenues à $t = 1,0 s$. La densité cible du fluide est $\rho_0 = 1000 kg.m^{-3}$ et la viscosité dynamique est $\eta = 0,001 Pa.s$. Les simulations des images (b) et (c) utilisent des constantes de raideur $k_1 = 50000$ et $k_2 = 7$ et les simulations en (d) et (e) avec $k_1 = 5000$ et $k_2 = 7$. Plus la couleur d'une particule s'approche du rose, plus sa densité est élevée vis-à-vis de ρ_0 . Plus sa couleur s'approche du bleue, plus sa densité est faible vis-à-vis de ρ_0 . Une couleur violette témoigne d'une densité égale à ρ_0 . Les flèches rouges indiquent la hauteur moyenne du fluide.

2.2. LE CONCEPT DE LA SIMULATION HPL

zéro (équation (1.6)) au temps futur de la simulation [26, p.11]. Voici un résumé de la démonstration retrouvée dans l'article [26, p.12] :

Après avoir calculé toutes les forces excepté la pression (voir sous-section 2.2.4.1 et sous-section 2.2.4.2), un champ d'accélération non associé à la pression \vec{a}^{np} est obtenu (np signifiant *non-pressure*), puis un champ de vitesse prédite non reliée à la pression \vec{v}^* en appliquant la première équation de la méthode d'Euler semi-implicite (équation (2.45)) sur \vec{a}^{np} (voir la sous-section 2.2.6). Enfin, en appliquant l'équation de continuité (1.9), une densité prédite ρ^* est obtenue telle que

$$\rho^* = \rho(t) - \Delta t \rho(t) \nabla \cdot \vec{v}^*. \quad (2.20)$$

À une particule i donnée, cela donne

$$\rho_i^* = \rho_i + \Delta t \sum_j m_j (\vec{v}_i^* - \vec{v}_j^*) \cdot \nabla W_{ij}. \quad (2.21)$$

Ensuite, l'accélération liée à la pression \vec{a}^p dans la forme incompressible des équations de Navier-Stokes (1.12) est isolée telle que

$$\vec{a}^p = -\frac{\nabla p(t)}{\rho(t)}. \quad (2.22)$$

D'après la première équation de la méthode d'Euler semi-implicite (voir la sous-section 2.2.6), la vitesse liée à la pression \vec{v}^p est donnée par

$$\vec{v}^p = \Delta t \vec{a}^p = -\Delta t \frac{\nabla p(t)}{\rho(t)}. \quad (2.23)$$

À noter que la vitesse au temps précédent a été enlevée, car on s'intéresse à la vitesse induite au temps de simulation courant t causée par \vec{a}^p .

En fusionnant les équations (2.20) et (2.23) et, considérant que, selon l'équation de continuité (1.9), la variation de densité doit tendre vers 0, donc que ρ^* doit tendre vers ρ_0 , alors il est possible d'écrire

$$\frac{\rho_0 - \rho^*}{\Delta t} - \rho(t) \nabla \cdot \left(\Delta t \frac{\nabla p(t)}{\rho(t)} \right) = 0, \quad (2.24)$$

2.2. LE CONCEPT DE LA SIMULATION HPL

se simplifiant en

$$\Delta t^2 \nabla^2 p(t) = \rho_0 - \rho^*. \quad (2.25)$$

Ici, la pression $p(t)$ du champ scalaire de pression advectée est inconnue. Il y a donc autant d'équations (2.25) qu'il y a d'inconnues, c'est-à-dire, de particules.

À noter que l'équation (2.25) annulant $\frac{\rho_0 - \rho^*}{\Delta t}$ prend l'invariance de densité comme source d'entrée. Il existe cependant une autre forme de l'EPP prenant la divergence de vitesse en source d'entrée. Contrairement à la première, celle-ci réécrit l'équation (2.24) comme étant

$$\nabla \vec{v}^* - \nabla \cdot \left(\Delta t \frac{\nabla p(t)}{\rho(t)} \right) = 0. \quad (2.26)$$

Une démonstration très similaire à celle de l'EPP prenant l'invariance de densité en source d'entrée peut être retrouvée dans le même article [26, p.12]. Ces deux formulations de la EPP ne sont pas équivalentes, car celle utilisant l'invariance de densité comme source d'entrée tend à faire osciller la densité autour de ρ_0 , mais celle utilisant la divergence de vitesse tend à faire dériver continuellement la valeur du volume du fluide [26, p.12].

2.2.5.3 HPL Faiblement Compressible (HPLFC)

HPLFC est introduit par l'article de M. Becker et M. Techner [11]. La méthode reprend les bases énoncées par M. Müller et al. [30], mais s'inspire des travaux de J.J. Monaghan [35] pour calculer automatiquement les constantes de raideurs impliquées dans l'équation d'état (2.19), afin de conserver une mesure de la compressibilité du fluide $\frac{\Delta \rho}{\rho_0}$ inférieure à 1%. Pour cela, k_2 est fixée à 7 et k_1 est donnée par

$$k_1 = \frac{\rho_0 c_s^2}{7}, \quad (2.27)$$

avec c_s la célérité du son dans le fluide. J.J. Monaghan [35] montre que la compressibilité du fluide varie selon c_s par la relation

$$\epsilon \approx \frac{|\rho - \rho_0|}{\rho_0} \approx \left(\frac{\|\vec{v}_f^{max}\|}{c_s} \right)^2, \quad (2.28)$$

2.2. LE CONCEPT DE LA SIMULATION HPL

avec \vec{v}_f^{max} , la vitesse maximale atteinte pour toutes les particules de fluide durant la simulation et ϵ , la fluctuation de densité relative du fluide donnant une mesure de sa compressibilité et fixée à 1% dans le cadre de HPLFC. Pour notre part, notre simulation se passe dans l'air où la constante $k_2 = 1$ pour respecter la loi des gaz parfaits (équations (2.16) ou (2.17)). De plus, un souffleur est ajouté à notre simulation et dont la vitesse de sortie d'air est de $\|\vec{v}_f^{max}\| = 38 \text{ m.s}^{-1}$, donnant une compressibilité de $\left(\frac{\|\vec{v}_f^{max}\|}{c_s}\right)^2 = \left(\frac{38}{340}\right)^2$ et produisant une fluctuation de densité $\epsilon \approx 1\%$.

En considérant les équations (2.7) et (2.9), la force de pression \vec{F}_i^p , appliquée sur la particule i , est donnée par

$$\vec{F}_i^p = -m_i \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij}. \quad (2.29)$$

Un des avantages d'utiliser une équation d'état est que la \vec{F}_i^p est obtenue indépendamment des autres forces. Cela permet d'éviter une nouvelle itération sur chacune des particules, donc une nouvelle recherche de voisinage.

Pour conclure, si l'utilisateur décide des constantes de raideurs k_1 et k_2 sans utiliser l'équation (2.27), alors l'implémentation retombera sur celle proposée initialement dans les travaux de M. Müller et al. [30], nommée HPL avec Équation d'État (HPLSE).

2.2.5.4 HPL Incompressible Predictif-Correctif (HPLIPC)

HPLIPC est introduite en informatique graphique par B. Solenthaler et R. Pajarola [44]. Selon l'article, l'heuristique de HPLFC produisait des constantes de raideur élevées dans l'équation d'état (2.18) et diminuait de fait Δt en concordance avec la condition du CFL (voir la section 2.3). L'objectif est donc de rendre le fluide incompressible tout en conservant un pas de temps Δt relativement élevé comparé à HPLFC. Une démonstration plus détaillée de la méthode et des équations utilisées est fournie dans l'article [26, pp.14-15].

L'idée générale est de prédire, puis corriger itérativement l'état de compression du système, au temps de simulation suivant, à l'aide de la force de pression, pour enfin minimiser la divergence de densité en dessous d'un seuil ϵ^{err} défini par l'utilisateur,

2.2. LE CONCEPT DE LA SIMULATION HPL

Algorithme 3 Algorithme HPLFC

1: **si** HPLFC **alors** ▷ Sinon HPLSE où l'utilisateur décide des constantes.
2: $k_2 \leftarrow 7$
3: $k_1 \leftarrow \frac{\rho_0 c_s^2}{7}$ (voir l'équation (2.27))
4: **fin si**
5: **boucle** ▷ Boucle de Simulation
6: **pour tout** particule i **faire**
7: Construit voisinage de i (sous-section 2.2.2)
8: **fin pour**
9: **pour tout** particule i **faire**
10: Calcule ρ_i à la position \vec{x}_i avec l'équation (2.11)
11: Calcule p_i avec l'équation (2.19)
12: **fin pour**
13: **pour tout** particule i **faire**
14: $\vec{F}_{f_i} \leftarrow \vec{F}_i^{viscosity} = \eta \nabla^2 \vec{v}_i$ en utilisant les équations (2.14) ou (2.15)
15: $\vec{F}_{f_i} \leftarrow \vec{F}_{f_i} + \vec{F}_i^p$ en utilisant l'équation (2.29)
16: $\vec{F}_{f_i} \leftarrow \vec{F}_{f_i} + \vec{F}_{ext}$ ▷ Toutes les forces additionnelles (exemple : souffleur)
17: $\vec{a}_i \leftarrow \vec{g}$ ▷ Gravité
18: $\vec{a}_i \leftarrow \vec{a}_i + \frac{\vec{F}_{f_i}}{m_i}$ (selon la seconde loi de Newton (équation (1.2)))
19: **fin pour**
20: **pour tout** particule i **faire**
21: met à jour \vec{v}_i puis \vec{x}_i avec méthode d'Euler semi-implicite (voir sous-section 2.2.6, équations (2.45), (2.46))
22: **fin pour**
23: **fin boucle**

2.2. LE CONCEPT DE LA SIMULATION HPL

mais sans résoudre directement une EPP (voir la [sous-section 2.2.5.2](#)) [44]. Tout d'abord, ρ^* , la densité prédite à l'itération courante après application de toutes les accélérations hormis la pression \vec{a}^{np} , est estimée telle que

$$\begin{aligned}\rho^* &= \sum_j m_j W_{ij} + \\ &\quad \Delta t \sum_j m_j (\vec{v}_i - \vec{v}_j) \cdot \nabla W_{ij} + \\ &\quad \Delta t^2 \sum_j m_j (\vec{a}_i^{np} - \vec{a}_j^{np}) \cdot \nabla W_{ij}.\end{aligned}$$

Soit la constante de raideur k^{PCI} telle que

$$k^{PCI} = \frac{\rho_0^2}{2m_i^2 \Delta t^2 \left(\sum_j \nabla W_{ij} \cdot \sum_j \nabla W_{ij} + \sum_j (\nabla W_{ij} \cdot \nabla W_{ij}) \right)}, \quad (2.30)$$

alors p_i , la valeur du champ scalaire de pression à la particule i , peut être déduite de l'équation d'état (2.17) et l'équation (2.30) telle que

$$p_i = k^{PCI} (\rho^* - \rho_0). \quad (2.31)$$

La variation de pression entraînée par la correction Δp^e à l'itération e est donnée par

$$\Delta p^e = k^{PCI} \left(\rho_0 - \rho^* - \Delta t^2 \sum_j m_j (\vec{a}_i^p - \vec{a}_j^p) \cdot \nabla W_{ij} \right),$$

avec \vec{a}_i^p l'accélération induite par la force de pression sur la particule i , obtenue en injectant la formulation symétrique (équation (2.9)) dans les équations de Navier-Stokes telle que

$$\vec{a}_i^p = - \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij}. \quad (2.32)$$

Cela donne p_i^{e+1} la pression à la particule i et à l'itération suivante $e + 1$ telle que

$$p_i^{e+1} = p_i^e + k^{PCI} \left(\rho_0 - \rho^* - \Delta t^2 \sum_j m_j (\vec{a}_i^p - \vec{a}_j^p) \cdot \nabla W_{ij} \right). \quad (2.33)$$

2.2. LE CONCEPT DE LA SIMULATION HPL

À partir de la première estimation $e = 0$ obtenue en utilisant l'équation (2.31), HPLIPC va itérer sur toutes les particules et estimer un champ de pression p^e en utilisant l'équation (2.33) jusqu'à ce que $\frac{(\rho_0 - \rho_i^* + \rho_i^{p^e})}{\rho_0}$ soit inférieure à ϵ^{err} .

HPLIPC évite la résolution du système d'équation linéaire d'une EPP (voir la [sous-section 2.2.5.2](#)) en appliquant l'équation d'état (2.31) à la place avec une constante de raideur k^{PCI} calculée automatiquement. Cependant, k^{PCI} est globale à toutes les particules du fluide et doit être initialisée en considérant une particule idéale dont le voisinage est parfaitement rempli [44], mais est aussi dépendante de Δt d'après l'équation (2.30). Elle devra donc être recalculée pour tout changement à Δt , par exemple, pour respecter la condition du CFL (voir la [section 2.3](#)). Pour éviter cela, nous introduisons k_{ti}^{PCI} indépendante de Δt telle que

$$k_{ti}^{PCI} = k^{PCI} \Delta t^2 = \frac{\rho_0^2}{2m_i^2 \left(\sum_j \nabla W_{ij} \cdot \sum_j \nabla W_{ij} + \sum_j (\nabla W_{ij} \cdot \nabla W_{ij}) \right)}. \quad (2.34)$$

2.2.5.5 HPL Implicitement Incompressible (HPLII)

HPLII fut introduite par M. Ihmsen et al. [25], la méthode résout l'équation de l'EPP (2.25) prenant l'invariance de densité comme source d'entrée ([sous-section 2.2.5.2](#)). Or, d'après l'équation (2.20), l'équation (2.25) devient

$$\rho_0 - \rho_i^* = \rho_0 - \rho_i + \Delta t \rho_i \nabla \cdot \vec{v}^*.$$

En utilisant la forme différentielle de la discrétisation de l'opérateur gradient dans HPL (équation (2.8)) tout en considérant $\rho_i \approx \rho_j$, alors

$$\rho_0 - \rho_i^* = \rho_0 - \rho_i - \Delta t \sum_j m_j (\vec{v}_i^* - \vec{v}_j^*) \cdot \nabla W_{ij}.$$

Après une démonstration pouvant être retrouvée dans l'article de D. Koschier et al. [26, pp.12-13], cela donne

$$\Delta t^2 \sum_j m_j (\vec{a}_i^p - \vec{a}_j^p) \cdot \nabla W_{ij} = \rho_0 - \rho^*. \quad (2.35)$$

2.2. LE CONCEPT DE LA SIMULATION HPL

Algorithme 4 Algorithme HPLIPC

```

1: Initialise  $k_{ti}^{PCI}$  avec l'équation (2.34)
2: Initialise  $k^{PCI} \leftarrow \frac{k_{ti}^{PCI}}{\Delta t^2}$ 
3: boucle ▷ Boucle de Simulation
4:   pour tout particule  $i$  faire
5:     Construit voisinage de  $i$  (sous-section 2.2.2)
6:   fin pour
7:   si  $\Delta t$  a changé alors
8:     Régénère  $k^{PCI} \leftarrow \frac{k_{ti}^{PCI}}{\Delta t^2}$ 
9:   fin si
10:  pour tout particule  $i$  faire
11:    Calcule  $\rho_i$  à la position  $\vec{x}_i$  avec l'équation (2.11)
12:  fin pour
13:  pour tout particule  $i$  faire
14:     $\vec{F}_{fi} \leftarrow \vec{F}_i^{viscosity} = \eta \nabla^2 \vec{v}_i$  en utilisant l'équation (2.14) ou (2.15)
15:     $\vec{F}_{fi} \leftarrow \vec{F}_{fi} + \vec{F}_{ext}$  ▷ Toutes les forces additionnelles (souffleur, ...)
16:     $\vec{a}_i^{np} \leftarrow \vec{g} + \frac{\vec{F}_{fi}}{m_i}$  (voir l'équation (1.2)) ▷  $\vec{a}_i^{np}$  est l'accélération sans la  

    composante de pression de l'équation (1.12)
17:     $\vec{v}_i^* \leftarrow \vec{v}_i + \Delta t \vec{a}_i^{np}$  (voir équation (2.45) dans sous-section 2.2.6)
18:    calcule  $d_{ii}$  avec l'équation (2.37)
19:  fin pour
20:   $iter \leftarrow 0$ 
21:  tant que  $\rho^{avg} \geq \epsilon^{err}$  et  $iter < maxIter$  faire
22:    pour tout particule  $i$  faire
23:      calcule  $\vec{a}_i^p \leftarrow -m_i \frac{2p_i}{\rho_0^2} \sum_j \nabla W_{ij}$  ▷ voir la démonstration dans l'article  

      [44] à partir de l'équation (2.32)
24:    fin pour
25:    pour tout particule  $i$  faire
26:      calcule  $\rho_i^{p^{e+1}} \leftarrow -\Delta t^2 \sum_j m_j (\vec{a}_i^p - \vec{a}_j^p) \cdot \nabla W_{ij}$ 
27:      calcule  $p_i^{e+1} \leftarrow p_i^e + k^{PCI} (\rho_0 - \rho^* + \rho_i^{p^e})$ 
28:    fin pour
29:     $iter \leftarrow iter + 1$ 
30:  fin tant que
31:  pour tout particule  $i$  faire
32:    met à jour  $\vec{v}_i$  puis  $\vec{x}_i$  avec méthode d'Euler semi-implicite (voir sous-  

    section 2.2.6, équations (2.45) et (2.46))
33:  fin pour
34: fin boucle

```

2.2. LE CONCEPT DE LA SIMULATION HPL

L'équation (2.35) peut être considérée comme un système linéaire à n particules équations de type $Ap = B$ avec $B = \rho_0 - \rho^*$. À noter que B est une matrice de même dimension que p puisque chaque particule possède une valeur ρ^* . Chaque équation possède une pression inconnue p_i à la particule i , mais aussi les valeurs des pressions des voisines p_j de i ainsi que les pressions des voisines de j , i compris. Donc A est une matrice carrée avec autant de colonnes/lignes que de particules, et dont les coefficients sont tous nuls, sauf pour les éléments de la diagonale α_{ii} et les éléments a_{ij} avec j une particule voisine de i , i étant la particule considérée, mais aussi l'index de la ligne de la matrice A . M. Ihmsen et al. [25] proposent d'utiliser une méthode Jacobienne pour résoudre $Ap = B$. Tout d'abord, toutes les pressions sont initialisées à zéro, puis une itération est réalisée en appliquant

$$p_i^{e+1} = \max\left((1 - \omega) p_i^e + \frac{\omega}{\alpha_{ii}} \left(\rho_0 - \rho_i^* - (Ap)_{i \neq j}^e \right), 0 \right), \quad (2.36)$$

avec ω , un coefficient de relaxation tel que $\omega = 0,5$. En effet, des valeurs plus petites augmentent le nombre d'itérations nécessaires pour converger, alors que des valeurs plus importantes rendent la simulation instable [25] [26, p.13]. L'opérateur max dans l'équation (2.36) empêche les instabilités de traction en interdisant les pressions négatives [25]. Toutefois, cela empêche la balle de léviter sous le souffleur, car il n'y a pas de correction dans le cas d'une dilatation du fluide, comme expliqué dans la [section 3.1](#). Enfin, l'équation (2.36) requiert de calculer α_{ii} et $(Ap)_{i \neq j}^e$, e étant l'itération de correction courante.

Concernant le calcul de $(Ap)_{i \neq j}^e$, d'après la méthode d'Euler semi-implicite (voir la [sous-section 2.2.6](#)) et selon l'équation (2.29), le déplacement causé par la force de pression \vec{F}_i^p sur la particule i est donné par

$$\Delta \vec{x}_i^p = \Delta t^2 \frac{\vec{F}_i^p}{m_i} = -\Delta t^2 \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \cdot \nabla W_{ij}.$$

2.2. LE CONCEPT DE LA SIMULATION HPL

Ce qui donne

$$\Delta t^2 \frac{\vec{F}_i^p}{m_i} = \underbrace{\left(-\Delta t^2 \sum_j \left(\frac{m_j}{\rho_i^2} \cdot \nabla W_{ij} \right) \right)}_{d_{ii}} p_i + \underbrace{\left(-\Delta t^2 \sum_j \left(\frac{m_j}{\rho_j^2} \cdot \nabla W_{ij} \right) \right)}_{d_{ij}} p_j, \quad (2.37)$$

où d_{ii} est le déplacement de la particule i causé par la pression p_i , tandis que d_{ij} est le déplacement de la particule i causé par la pression p_j .

D'autre part, M. Ihmsen et al. [25] formulent α_{ii} comme étant

$$\alpha_{ii} = \sum_j m_j (d_{ii} - d_{ji}) \nabla W_{ij}. \quad (2.38)$$

Après démonstration se retrouvant dans les articles [25] [26, pp.12-14], la pression p_i^{e+1} à l'itération de correction suivante est donnée par

$$p_i^{e+1} = \max\left(p_i^e + \frac{\omega}{\alpha_{ii}} \left(\rho_0 - \rho_i^* - \sum_j \left((d_{ij} - d_{jj}) p_j - \sum_{k \neq i} d_{jk} p_k \right) \cdot \nabla W_{ij} \right), 0 \right). \quad (2.39)$$

L'algorithme 5 explicite l'implémentation du résolveur de l'EPP utilisé par HPLII.

2.2.5.6 Relation entre HPLIPC et HPLII

D. Koschier et al. [26] démontrent que

$$k^{PCI} = \frac{\omega}{\alpha_{ii}}.$$

La différence de performance entre les deux méthodes est essentiellement parce que HPLII résout une EPP, donc un système d'équation linéaire en calculant α_{ii} pour chaque particule, alors que HPLIPC ne calcule qu'une constante de raideur globale appliquée à toutes les particules, ce qui peut se révéler suboptimal. En effet, si $k^{PCI} < \frac{\omega}{\alpha_{ii}}$, alors la convergence de HPLIPC est plus mauvaise que HPLII, entraînant plus d'itération pour produire la même correction. À l'inverse, HPLIPC se révèle instable si $k^{PCI} > \frac{\omega}{\alpha_{ii}}$ [26, pp.15-16].

2.2. LE CONCEPT DE LA SIMULATION HPL

Algorithme 5 Algorithme HPLII

Entrée: $maxIter \geq 1$ ▷ Pour éviter les boucles infinies

1: **boucle** ▷ Boucle de simulation

2: **pour tout** particule i **faire**

3: Construit voisinage de i (sous-section 2.2.2)

4: **fin pour**

5: **pour tout** particule i **faire**

6: Calcule ρ_i à la position \vec{x}_i avec l'équation (2.11)

7: **fin pour**

8: **pour tout** particule i **faire**

9: $\vec{F}_{f_i} \leftarrow \vec{F}_i^{viscosity} = \eta \nabla^2 \vec{v}_i$ en utilisant les équations (2.14) ou (2.15)

10: $\vec{F}_{f_i} \leftarrow \vec{F}_{f_i} + \vec{F}_{ext}$ ▷ Toutes les forces additionnelles (exemple : souffleur)

11: $\vec{a}_i^{np} \leftarrow \vec{g}$ ▷ \vec{a}_i^{np} est l'accélération sans la composante de pression de l'équation 1.12)

12: $\vec{a}_i^{np} \leftarrow \vec{a}_i^{np} + \frac{\vec{F}_{f_i}}{m_i}$ selon la seconde loi de Newton (équation (1.2))

13: $v_i^* \leftarrow v_i + \Delta t \vec{a}_i^{np}$ (voir équation (2.45) dans la sous-section 2.2.6)

14: calcule d_{ii} comme énoncé dans l'équation (2.37)

15: **fin pour**

16: **pour tout** particule i **faire**

17: calcule α_{ii} comme énoncé dans l'équation (2.38)

18: calcule ρ_i^* comme énoncé dans l'équation (2.21)

19: $p_i \leftarrow \omega p_i$ ▷ ω le coefficient de relaxation défini entre 0 et 1 [25]

20: **fin pour**

21: $iter \leftarrow 0$

22: **tant que** $\rho^{avg} \geq \epsilon^{err}$ **et** $iter < maxIter$ **faire**

23: **pour tout** particule i **faire**

24: calcule $d_{ij} p_j^e$ comme énoncé dans l'équation (2.37)

25: **fin pour**

26: **pour tout** particule i **faire**

27: calcule p_i^{e+1} comme énoncé dans l'équation (2.39)

28: $p_i \leftarrow p_i^{e+1}$

29: **fin pour**

30: $iter \leftarrow iter + 1$

31: **fin tant que**

32: **pour tout** particule i **faire**

33: calcule \vec{a}_i^p comme énoncé dans l'équation (2.32)

34: $v_i \leftarrow v_i^* + \Delta t \vec{a}_i^p$

35: $\vec{x}_i \leftarrow \vec{x}_i + \Delta t v_i$ ▷ méthode d'Euler semi-implicite (équation (2.46) à la sous-section 2.2.6)

36: **fin pour**

37: **fin boucle**

2.2. LE CONCEPT DE LA SIMULATION HPL

2.2.5.7 HPL Sans Divergence (HPLSD)

HPLSD, introduit par J. Bender et D. Koschier [4], résout les deux formulations de l'EPP, l'une avec l'invariance de densité comme source d'entrée (équation (2.25)) et l'autre avec la divergence de vitesse (équation (2.26)). La motivation étant que la première EPP résolvant l'invariance de densité calcule les positions des particules pour rendre le fluide incompressible, mais n'assure pas la contrainte de divergence zéro (équation (1.6)) [4] et la densité tendra à osciller autour de ρ_0 [26, p.12] (voir la sous-section 2.2.5.2). C'est la raison pour laquelle HPLSD utilise la seconde EPP pour forcer le champ de vitesse à respecter la contrainte de divergence zéro.

Tout d'abord, à l'instar de HPLII, HPLSD calcule une constante de raideur par particule k_i^{HPLSD} , mise à jour une seule fois par pas de temps de la simulation et utilisée pour les deux solveurs de l'EPP, telle que

$$k_i^{HPLSD} = \frac{\rho_i^2}{\|\sum_j m_j \nabla W_{ij}\|^2 + \sum_j \|m_j \nabla W_{ij}\|^2}. \quad (2.40)$$

Puis, le solveur pour l'invariance de densité a pour objectif de résoudre l'EPP en prenant l'invariance de densité comme source d'entrée (équation (2.25)). Pour ce faire, la densité globale ρ^{avg} calculée sur tout le domaine est amenée autour de la densité cible ρ_0 en admettant une certaine erreur $\epsilon^{densité}$ choisie par l'utilisateur. Cependant, plus cette erreur sera proche de zéro, plus la correction sera précise, mais plus le nombre d'itérations nécessaire à la convergence du solveur augmentera.

L'article [26, p.17] montre que

$$p_i^\rho = \frac{k_i^{HPLSD}}{\Delta t^2} (\rho_i^* - \rho_0). \quad (2.41)$$

Une fois que p_i^ρ et p_j^ρ calculées pour la particule i et ses voisines j (pour rappel, la densité prédite ρ^* est obtenue avec l'équation (2.21)), le calcul de la vitesse corrigée \vec{v}_i^* devient possible en utilisant la forme symétrique de la discrétisation de l'opérateur gradient (équation (2.9)), soit

$$\vec{v}_i^{*e+1} = \vec{v}_i^{*e} - \Delta t \sum_j m_j \left(\frac{p_i^\rho}{\rho_i^2} + \frac{p_j^\rho}{\rho_j^2} \right) \nabla W_{ij}. \quad (2.42)$$

2.2. LE CONCEPT DE LA SIMULATION HPL

L’[algorithme 6](#) présente une possible implémentation du résolveur de HPLSD avec l’invariance de densité comme source d’entrée.

Algorithme 6 Résolveur EPP d’invariance de densité de HPLSD

Entrée: $maxIter \geq 1$ ▷ Pour éviter les boucles infinies

Entrée: $\epsilon^{densité} > 0$

```

1: fonction RÉSOUTDENSITÉEPP
2:    $iter \leftarrow 0$ 
3:   tant que  $(\rho^{avg} - \rho_0) > \epsilon^{densité}$  et  $(iter < maxIter)$  faire
4:      $iter \leftarrow iter + 1$ 
5:     pour tout particule  $i$  faire
6:       Calcule  $\rho_i^*$  avec l’équation (2.21)
7:     fin pour
8:     pour tout particule  $i$  faire
9:       Calcule  $p_i^\rho$  avec l’équation (2.41)
10:      Calcule  $p_j^\rho$  avec l’équation (2.41)
11:      Calcule  $\vec{v}_i^{*e+1}$  avec l’équation (2.42)
12:    fin pour
13:  fin tant que
14: fin fonction

```

Enfin, le résolveur pour la divergence zéro a pour objectif de résoudre l’EPP avec le terme de la divergence du champ de vitesse comme entrée (équation (2.26)) afin de forcer explicitement la contrainte de divergence zéro (équation (1.6)) qui, combinée avec l’équation de continuité (1.9), donne

$$\frac{D\rho^{avg}}{Dt} = 0,$$

avec ρ^{avg} la moyenne de densité de toutes les particules du fluide. $\frac{D\rho^{avg}}{Dt}$ est une autre façon de mesurer la valeur de la divergence dans le domaine. Si celle-ci passe en dessous de l’erreur $\epsilon^{vitesse}$, alors le fluide sera considéré comme incompressible.

Selon l’article [26, p.17], la pression de correction de la vitesse p_i^v est donnée par

$$p_i^v = \frac{k_i^{HPLSD}}{\Delta t} \frac{D\rho_i}{Dt}, \quad (2.43)$$

permettant de calculer la vitesse corrigée v_i^* en utilisant la forme symétrique de la

2.2. LE CONCEPT DE LA SIMULATION HPL

discrétisation de l'opérateur gradient (équation (2.9)) telle que

$$v_i^{*e+1} = v_i^{*e} - \Delta t \sum_j m_j \left(\frac{p_i^v}{\rho_i^2} + \frac{p_j^v}{\rho_j^2} \right) \nabla W_{ij}. \quad (2.44)$$

Algorithme 7 Résolveur EPP en divergence de vitesse de HPLSD

Entrée: $maxIter \geq 1$ ▷ Pour éviter les boucles infinies

Entrée: $\epsilon^{vitesse} > 0$

```

1: fonction RÉSOUTVITESSEEPP
2:    $iter \leftarrow 0$ 
3:   tant que  $(\frac{D\rho^{avg}}{Dt}) \geq \epsilon^{vitesse}$  et  $(iter < maxIter)$  faire
4:      $iter \leftarrow iter + 1$ 
5:     pour tout particule  $i$  faire ▷ Équation de continuité pour calculer  $\frac{D\rho}{Dt}$ 
6:        $\frac{D\rho_i}{Dt} = -\rho_i \nabla \cdot v_i^* = \sum_j m_j (v_i^* - v_j^*) \cdot \nabla W_{ij}$ 
7:     fin pour
8:     pour tout particule  $i$  faire
9:       Calcule  $p_i^v$  avec l'équation (2.43)
10:      Calcule  $p_j^v$  avec l'équation (2.43)
11:      Calcule  $\bar{v}_i^{*e+1}$  avec l'équation (2.44)
12:     fin pour
13:   fin tant que
14: fin fonction

```

Pour conclure, la simulation HPLSD résout deux EPPs, ce qui est plus coûteux que d'en résoudre une seule. Néanmoins, le second solveur converge plus vite que s'il avait été utilisé seul, et la résolution des deux EPPs ensemble, amène une meilleure stabilité de la simulation, augmentant alors davantage Δt [4, 26].

L'algorithme complet est explicité dans l'algorithme 8. Néanmoins, il est à noter que nous n'avons pas suivi exactement l'algorithme proposé dans les articles [4, 26], mais nous nous sommes inspirés du simulateur de fluide complet et au code source ouvert, nommé *SPlisHSPlasH*, dont le lien est fourni dans la section B.1, implémenté par J. Bender et al. [26].

Enfin, le noyau d'interpolation simple W_{ij} et sa forme dérivée à l'ordre 1, utilisés dans les deux solveurs de la EPP, doivent provenir du même noyau pour que les étapes de prédiction et de correction restent compatibles [4].

2.2. LE CONCEPT DE LA SIMULATION HPL

Algorithme 8 Algorithme HPLSD

- 1: **boucle** ▷ Boucle de Simulation
 - 2: **pour tout** particule i **faire**
 - 3: Construit voisinage de i ▷ voir [sous-section 2.2.2](#)
 - 4: **fin pour**
 - 5: **pour tout** particule i **faire**
 - 6: Calcule ρ_i à la position \vec{x}_i avec l'équation (2.11)
 - 7: **fin pour**
 - 8: **pour tout** particule i **faire**
 - 9: Calcule coefficient de raideur k^{HPLSD} avec l'équation (2.40)
 - 10: **fin pour**
 - 11: RÉSOUTDENSITÉEPP ▷ [algorithme 6](#)
 - 12: **pour tout** particule i **faire**
 - 13: $\vec{F}_{f_i} \leftarrow \vec{F}_i^{viscosity} = \eta \nabla^2 \vec{v}_i$ avec les équations (2.14) ou (2.15)
 - 14: $\vec{F}_{f_i} \leftarrow \vec{F}_{f_i} + \vec{F}_{ext}$ ▷ Toutes les forces additionnelles comme celle du souffleur
 - 15: $\vec{a}_i^{np} \leftarrow \vec{g}$ ▷ Gravité (\vec{a}_i^{np} est l'accélération sans la composante de pression des équations de Navier-Stokes (1.12))
 - 16: $\vec{a}_i^{np} \leftarrow \vec{a}_i^{np} + \frac{\vec{F}_{f_i}}{m_i}$ ▷ voir équation (1.2)
 - 17: $\vec{v}_i^* \leftarrow \vec{v}_i + \Delta t \vec{a}_i^{np}$ ▷ voir équation (2.45) dans [sous-section 2.2.6](#)
 - 18: **fin pour**
 - 19: RÉSOUTVITESSEEPP ▷ [algorithme 7](#)
 - 20: **pour tout** particule i **faire**
 - 21: $\vec{v}_i \leftarrow \vec{v}_i^*$
 - 22: met à jour \vec{x}_i avec la méthode d'Euler semi-implicite (voir équation (2.46) dans [sous-section 2.2.6](#))
 - 23: **fin pour**
 - 24: **fin boucle**
-

2.3. CONDITIONS DE STABILITÉ

2.2.5.8 Bilan comparatif

Nous implémentons chacune des méthodes HPLFC, HPLIPC, HPLII et HPLSD telles qu’elles ont été présentées dans leurs articles originaux en nous inspirant de l’implémentation réalisée dans *SPLisHSPlasH* (le lien est fourni dans la [section B.1](#)).

Chacune de ces méthodes possède ses avantages et ses inconvénients. HPLFC est plus facile à implémenter et à paramétrer, mais le pas de temps Δt doit être réduit comparativement aux autres méthodes qui cherchent à calculer automatiquement une constante de raideur qui, excepté dans le cas de HPLIPC, est variable à chaque temps t de la simulation, pour permettre l’incompressibilité sous une certaine erreur. Un tableau comparatif des gains en performance entre ces diverses méthodes est présenté par J. Bender et al. [5] [26, pp.14-15].

2.2.6 Méthode d’Euler semi-implicite

À partir du champ d’accélération \vec{a} au temps de simulation suivante $t + \Delta t$, la méthode d’Euler semi-implicite met à jour la vitesse \vec{v} avec

$$\vec{v}_i(t + \Delta t) = \Delta t \vec{a}_i(t + \Delta t) + \vec{v}_i(t), \quad (2.45)$$

permettant ensuite la mise à jour de la position $\vec{x}_i(t + \Delta t)$ avec

$$\vec{x}_i(t + \Delta t) = \Delta t \vec{v}_i(t + \Delta t) + \vec{x}_i(t). \quad (2.46)$$

2.3 Conditions de stabilité

D’après l’équation (2.46), si $\|\vec{v}_i\|\Delta t > 2h$, avec h la longueur de lissage du noyau utilisé, alors la particule i passera au travers du voisinage d’une autre particule sans contribuer à l’interpolation de ses quantités scalaires et vectorielles caractéristiques, ni recevoir de contribution, provoquant ainsi des erreurs dans les calculs et des instabilités dans la simulation. La condition de Courant-Friedrichs-Lewy [12], comme présentée par D. Koschier et al. [26, p.9], pose alors une condition pour que la simu-

2.3. CONDITIONS DE STABILITÉ

lation converge vers un résultat stable, telle que

$$\Delta t \leq \lambda \frac{2h}{\|\vec{v}_{max}\|}, \quad (2.47)$$

avec λ une constante définie par l'utilisateur entre 0 et 1. En pratique, la valeur $\lambda = 0,4$ est choisie, car bien que cela ne garantisse pas la stabilité de la simulation, cette valeur représente un bon compromis dans la plupart des situations [26, p.9].

Enfin, nous introduisons une autre condition de stabilité nécessaire de respecter dans le cadre d'une simulation en domaine fermé et rempli, en plus du CFL :

Si le domaine est fermé, alors il existe un volume, $V_{domaine}$, défini comme le volume interne du solide constituant les frontières du domaine.

À noter que le volume d'un solide fermé et convexe peut être approximé en sommant les volumes des tétraèdres irréguliers, dont les bases sont les triangles composant la surface du maillage du solide, et dont le sommet, partagé par tous les tétraèdres, est un point quelconque fixe à l'intérieur du solide. Il est évident que pour le cas d'un parallélépipède rectangle, la formule théorique Longueur \times Largeur \times Profondeur se révèle plus simple à utiliser et plus précise. Cependant, nous pensons que cette manière d'approximer le volume est meilleure pour les géométries sphériques et cylindriques puisqu'une coupe de sphère ou de cylindre, lorsque modélisée par un maillage sur ordinateur, n'est pas parfaitement circulaire.

Soit V_{b_k} le volume du solide k différent de celui constituant les frontières du domaine, s'il est rempli, alors

$$\sum_i V_i = V_{domaine} - \sum_k V_{b_k},$$

avec V_i le volume de la particule i tel que $\rho_i = \frac{m_i}{V_i}$, cela donne

$$\sum_i \frac{m_i}{\rho_i} = V_{domaine} - \sum_k V_{b_k}. \quad (2.48)$$

Sachant que le fluide à l'équilibre tendra vers sa densité cible ρ_0 , et que sa masse est fixée à une masse idéale m , le nombre de particules n_{max} à générer dans le domaine

2.4. GESTION DES FRONTIÈRES

au départ de la simulation est donné à partir de l'équation (2.48) tel que

$$n_{max} = \frac{\rho_0(V_{domain} - \sum_k V_{b_k})}{m}. \quad (2.49)$$

Soit n_p , le nombre de particules dans le domaine. En supposant le domaine comme rigide, si $n_p < n_{max}$, alors des zones de dilatation du fluide se formeront dans le domaine, si $n_p > n_{max}$, alors il y a plus de matière dans le domaine qu'il ne peut réellement en contenir et la simulation n'atteindra jamais un état d'équilibre, car la densité du fluide ρ supérieure à la densité cible ρ_0 n'atteindra jamais celle-ci en tout point du domaine. Le modèle utilisé, soit la forme incompressible des équations de Navier-Stokes, n'est alors pas respecté par construction et les dépassements amèneront des instabilités numériques.

2.4 Gestion des frontières

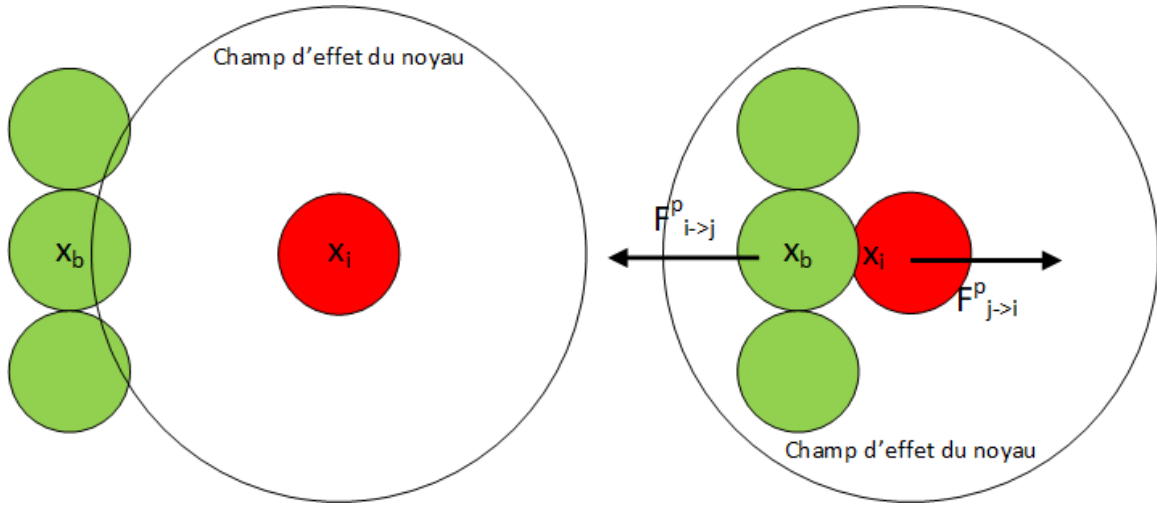
Tout corps solide constitue une frontière et induit une interaction fluide \leftrightarrow solide. L'interface fluide \leftrightarrow fluide n'est pas considérée puisque la simulation de plusieurs fluides ensemble n'a pas été abordée dans nos travaux.

2.4.1 Les méthodes pour gérer les frontières

La gestion des frontières pose des défis techniques majeurs en simulation des fluides. N. Akinci et al. [1] présentent une solution en échantillonnant la surface du solide avec des particules frontières. Ce modèle présente l'avantage de générer des forces du solide sur le fluide, tout en conservant une implémentation consistante avec le modèle particulaire préétabli. À l'opposé, d'autres travaux menés par J. Bender et al. [6] proposent de représenter la surface des solides avec des surfaces implicites.

Le modèle particulaire de N. Akinci et al. [1], utilisé dans le cadre de nos travaux, pose plusieurs problèmes. Tout d'abord, les particules frontières doivent empêcher les particules de fluide d'entrer naturellement dans le solide qu'elles composent. Cependant, l'implémentation d'un système de collision entre particules n'est pas triviale à réaliser et la force de pression est utilisée pour repousser le fluide tentant de pénétrer

2.4. GESTION DES FRONTIÈRES



(a) Interaction d'une particule de fluide arrivant sur la frontière d'un solide, mais hors de la portée du champ d'effet du noyau.

(b) Interaction d'une particule de fluide chevauchant la frontière d'un solide.

Figure 2.7 – Idée générale derrière la gestion des frontières échantillonnées en particule. Les particules frontières b sont en vert et la particule de fluide i est en rouge.

à l'intérieur du solide, comme expliqué sur la [figure 2.7](#). Si la particule i est éloignée de la frontière comme montré sur la [figure 2.7a](#), sa densité ρ est donc inférieure à sa densité cible ρ_0 . Donc $p_i = 0$ ce qui entraîne que \vec{F}_i^p la force de pression s'exerçant sur i est nulle. Lorsque la particule i entre au contact de la frontière (voir la [figure 2.7b](#)), sa densité devient plus élevée en la chevauchant, jusqu'au moment où $\rho_i > \rho_0$, donc $p_i > 0$ entraînant $\vec{F}_{j \rightarrow i}^p \neq \vec{0}$. Les deux particules de fluide et de frontières sont alors repoussées l'une de l'autre, car d'après la troisième loi de Newton (équation (1.3)), le corps solide b reçoit $\vec{F}_{i \rightarrow j}^p = -\vec{F}_{j \rightarrow i}^p$ provenant de la particule i appliquée en \vec{x}_b . Néanmoins, cette méthode douce n'assure pas explicitement la non-pénétration du fluide à l'intérieur du solide.

M. Müller et al. [30] proposent de vérifier si la particule du fluide quitte le domaine solide, et le cas échéant, ramène celle-ci à l'intérieur en modifiant manuellement sa position. Puisque ce déplacement ne suit aucun modèle physique, il est à observer que si une particule i se retrouve téléportée à côté d'une particule j , alors celle-ci entrera dans le voisinage de j , possiblement déjà rempli, provoquant alors un pic de pression au temps de simulation suivant et amenant à des instabilités numériques.

2.4. GESTION DES FRONTIÈRES

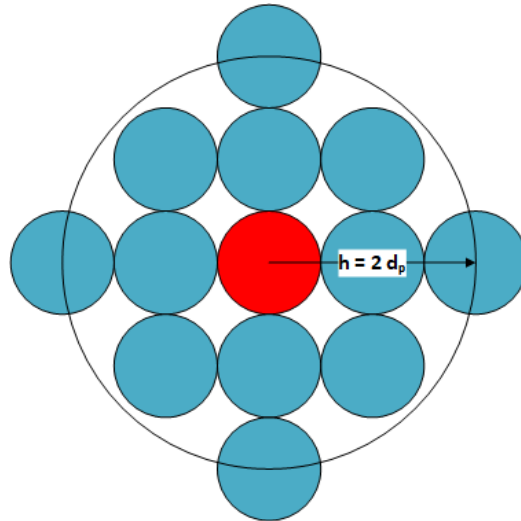


Figure 2.8 – Schéma 2D d'un voisinage rempli autour d'une particule de fluide en rouge. Le rayon d'effet du noyau est montré par le cercle noir de rayon $h = 2d_p$, h étant la longueur de lissage du noyau et d_p étant le diamètre d'une particule. Les particules n'ayant aucune contribution au voisinage ne sont pas montrées.

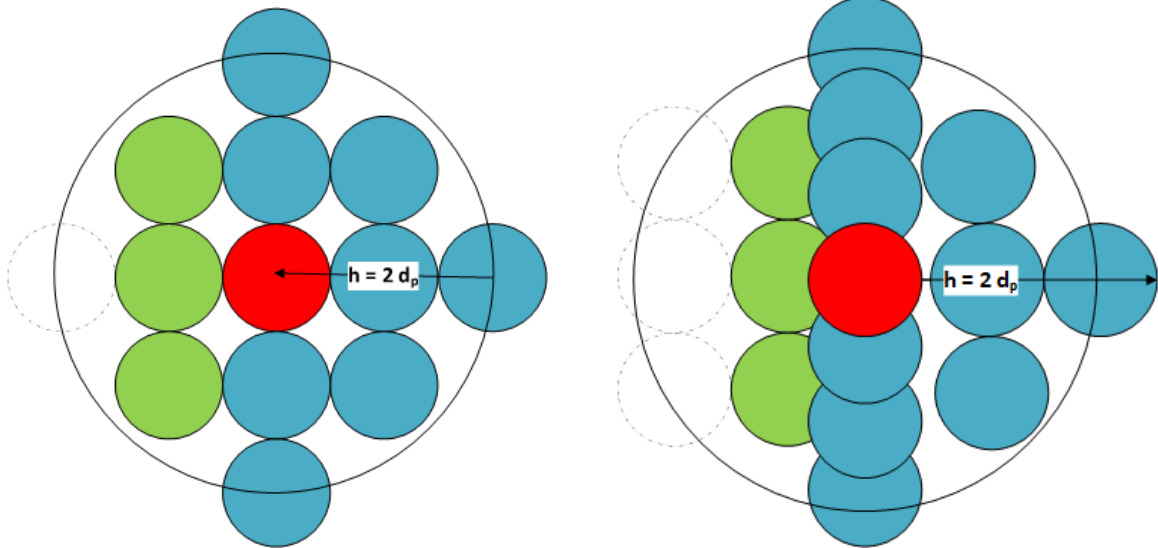
Le second problème provoqué par le modèle particulaire est que la génération des particules frontières doit être parfaite, au risque d'entraîner des artefacts notables. Si l'échantillonnage est trop clairsemé, les particules de fluide passeront au travers de la surface. Si l'échantillonnage est trop dense, une surestimation du voisinage provoquera un pic de pression et une surconsommation de la mémoire. Enfin, si l'échantillonnage est non uniforme, des biais de trajectoires se produiront lors du calcul des forces en raison d'une asymétrie des contributions [26, pp.18-22].

Pour finir, puisqu'il n'y a pas de particules de l'autre côté de la surface mince d'un solide, une sous-estimation de la densité se produit pour la particule de fluide au contact du solide en raison du manque de voisinage. Les particules de fluides ont alors tendance à se faire pousser contre la frontière (voir la [figure 2.9](#)) contrairement à une particule au milieu du fluide dont le voisinage est rempli (voir la [figure 2.8](#)).

D. Violeau et B.D. Rogers [49] énumèrent trois familles de solutions pour corriger ce dernier problème :

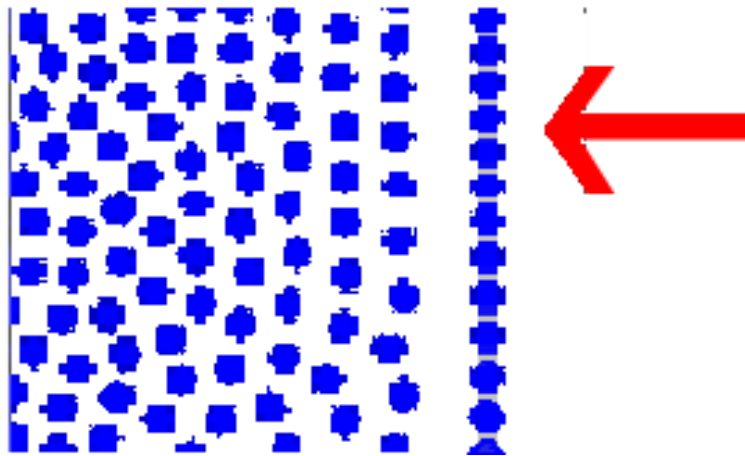
- La première consiste à générer des particules frontières fictives, soit avec plusieurs couches de particules frontières, soit avec des particules fantômes reflétant

2.4. GESTION DES FRONTIÈRES



(a) Figure du voisinage dont la contribution de la particule en pointillé absente derrière la frontière est manquante.

(b) Figure de l'effet du manque de voisinage amenant une sous-estimation de la densité et donc de la pression. Les particules manquantes pour obtenir un voisinage rempli sont marquées en pointillé.



(c) Coupe d'une simulation 2D montrant l'artefact d'agglutinement des particules de fluide (représentées par des cercles bleus) sur la frontière.

Figure 2.9 – Coupes 2D montrant la contribution au voisinage autour de la particule rouge au contact des particules frontières vertes.

2.4. GESTION DES FRONTIÈRES

l'arrangement des particules de fluide au contact de la surface [38]. Un résumé de la méthode HPL Fantômes (*Ghost Smoothed Particle Hydrodynamics*) est présenté avec nos résultats dans la [sous-section 2.4.1.1](#).

- La deuxième introduit des forces de pénalité, comme les forces de répulsion basées sur le potentiel intermoléculaire de Lennard-Jones, venant compléter la force de pression [33], ou la force de pression peut se refléter artificiellement sur la particule de fluide au contact d'une particule frontière [26, pp.18-22].
- La troisième compense le manque de voisinage en ajoutant un second terme de normalisation à l'équation (2.1). Considérant l'équation (2.11) pour calculer la densité d'une particule de fluide proche d'une particule frontière, N. Akinci et al. [1] proposent

$$\frac{\rho_i}{m_i} = \sum_{j_f} W_{ij_f} + \gamma_\rho \sum_{j_b} W_{ij_b}, \quad (2.50)$$

avec i la particule de fluide considérée, j_f une particule de fluide j dans le voisinage de i et j_b une particule de frontière j dans le voisinage de i . γ_ρ est un coefficient dépendant du noyau utilisé, de sa longueur de lissage et de la position des particules de fluide vis-à-vis de la particule du solide. Il est calculé en considérant le cas d'une particule de fluide au contact de la frontière, et dont le voisinage est parfaitement rempli [26, p.20]. Considérant le volume $V_i = \frac{m_i}{\rho_i}$ à la particule i , alors l'équation (2.50) se réécrit

$$\gamma_\rho = \frac{\frac{1}{V_i} - \sum_{j_f} W_{ij_f}}{\sum_{j_b} W_{ij_b}}. \quad (2.51)$$

Enfin, le volume d'une particule de frontière i_b est donné dans l'article [26, pp.18-22] comme étant

$$V_{i_b} = \frac{\gamma_\rho}{\sum_{j_b} W_{i_b j_b}}. \quad (2.52)$$

Sachant que $\rho = m/V$, la masse artificielle de la particule de frontière m_{i_b} est

$$m_{i_b} = \rho_0 V_{i_b} = \rho_0 \frac{\gamma_\rho}{\sum_{j_b} W_{i_b j_b}}. \quad (2.53)$$

À noter que nous pensons que cette approche présente aussi une solution élé-

2.4. GESTION DES FRONTIÈRES

gante au second problème d'échantillonnage imparfait.

2.4.1.1 HPL Fantômes

Nous présentons ici la méthode HPL Fantômes, telle qu'énoncée par H. Schechter et R. Bridson [38]. L'objectif est d'empêcher la non pénétration du fluide à l'intérieur des frontières du solide tout en évitant l'agglutination non naturelle des particules de fluides au contact de la surface (voir la [figure 2.9c](#)). Des particules fictives nommées particules fantômes sont générées derrière les frontières fluides↔solides et à l'interface liquide-air ; puis les particules fantômes se voient attribuer les valeurs de certaines quantités physiques, comme la masse et la densité, des particules de fluide les plus proches ; tandis que la vitesse est obtenue telle que

$$\vec{u}_g = \vec{u}_b \cdot \vec{n} + \vec{u}_i^T,$$

avec \vec{u}_b la vitesse de la particule de solide, \vec{u}_i^T la vitesse tangentielle de la particule de fluide i vis-à-vis de la surface du solide et \vec{n} la normale à la surface du solide à la position de la particule de solide à la frontière. Ceci permet d'assurer le respect de la condition de non-collage (*no-stick*) aux frontières mobiles, énoncée par R. Bridson [9] telle que

$$\vec{u}_i \cdot \vec{n} = \vec{u}_b \cdot \vec{n}, \tag{2.54}$$

avec \vec{u}_i la vitesse de la particule de fluide i , \vec{u}_b la vitesse de la particule de solide et \vec{n} la normale à la surface du solide à la position de la particule de solide à la frontière. Cette condition énonce explicitement que la composante normale de la vitesse de la particule de fluide ne doit pas s'éloigner ou pénétrer la surface du solide. H. Schechter et R. Bridson [38] mettent l'accent sur l'importance de respecter la condition de non-collage aux frontières (équation (2.54)), nous citons :

*«In typical scenarios, liquid can only separate from a solid if air can rush in to fill the gap : by default liquid shouldn't be able to leave, leading to the standard no-stick $\mathbf{v} \cdot \mathbf{n} = 0$ boundary condition.»*¹ ([38], p.2)

1. Typiquement, un liquide ne peut se séparer d'un solide que si l'air peut venir combler le vide engendré : un liquide ne devrait pas être capable de s'en aller par défaut, amenant à la condition de non-collage standard aux frontières $\vec{v} \cdot \vec{n} = 0$.

2.4. GESTION DES FRONTIÈRES

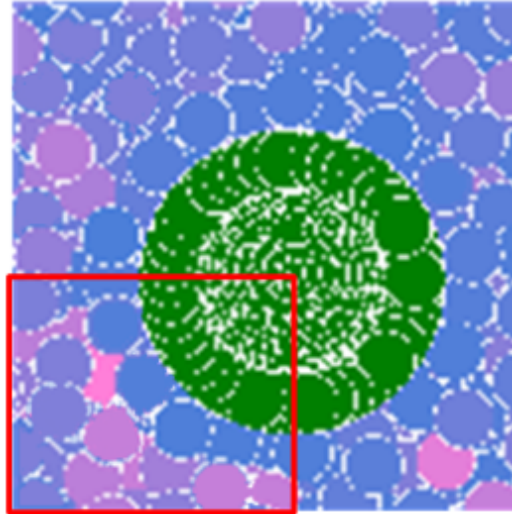


Figure 2.10 – Artefact entraîné par la condition de non-collage aux frontières dans le cas d’une balle immergée. Les particules de solide sont vertes. La couleur des particules de fluide varient entre le bleu (la norme du vecteur vitesse est faible) et le rouge (la norme du vecteur vitesse est élevée).

Cependant, nous pensons que bien que cela soit valide pour un liquide arrosant un solide immergé dans l’air, cela ne s’applique pas bien pour un solide en immersion totale dans un liquide ou un gaz passant à proximité de sa surface, car c’est justement contraire à l’effet de Coandă générant un effet de succion des particules environnantes (voir la [section 3.3](#)). Lorsque nous avons tenté d’implémenter la condition de non-collage aux frontières (équation (2.54)), il apparaît dans le rectangle rouge de la [figure 2.10](#) qu’une couronne non naturelle de particules de fluide immobiles est générée autour du solide, faisant alors rempart entre le solide et les particules du flux à vitesse élevée et passant à proximité. Cette séparation emmène une contribution plus faible des particules mobiles, empêchant donc d’attirer la balle dans les zones de dilatation engendrées par le flux d’air. Par conséquent, cette méthode n’est pas adaptée à la lévitation de la balle.

Chapitre 3

Lévitiation d'objets dans un fluide

L'objectif principal de nos travaux est de produire un simulateur de fluide réaliste capable de simuler des effets hydrodynamiques et aérodynamiques. Nous avons choisi l'expérience de la lévitation de la balle sous un souffleur d'air pour valider que les phénomènes modélisés soient cohérents avec la réalité. En effet, si le modèle choisi est incomplet ou que l'implémentation ne respecte pas suffisamment le modèle, la balle ne restera pas en lévitation dans le flux d'air. Nous avons implémenté les méthodes HPLFC, HPLIPC, HPLII et HPLSD dont les bases sont détaillées dans le [chapitre 2](#). Puis, nous nous sommes particulièrement concentrés sur la méthode HPLSD puisque d'après les comparaisons réalisées par J. Bender et D. Koschier [4], il s'agit de la méthode la plus stable avec un pas de temps Δt le plus grand parmi les méthodes implémentées dans nos travaux. Les liens URL des vidéos de la réalité et de nos résultats de simulation sont fournis en [annexe B](#).

3.1 Lévitiation d'une balle dans un souffleur

Le souffleur est modélisé par un champ de force dont la norme est déterminée empiriquement. Il s'applique à toutes les particules entrant dans sa zone d'effet. Néanmoins, il n'a pas été possible de maintenir la balle en lévitation en utilisant la méthode HPLSD originale décrite par J. Bender et al. [4]. Celle-ci était éjectée du

3.1. LÉVITATION D'UNE BALLE DANS UN SOUFFLEUR

flux et tombait sous le souffleur. Nous soupçonnons deux hypothèses expliquant cet échec :

- En raison du nombre de constantes à paramétrer par l'utilisateur, dépassant les 40 constantes dans notre implémentation, et dont certaines doivent être déterminées empiriquement, car n'ayant aucun lien direct avec des constantes physiques réelles. Il est très difficile de trouver un jeu de constantes amenant à une simulation réaliste. À titre d'exemple, 40 booléens (vrai/faux) à paramétrer amène plus de mille milliards de possibilités.
- Notre implémentation du solveur de l'EPP en invariance de densité utilisé dans HPLSD ne produit pas une correction satisfaisante lorsque $\rho_i < \rho_0$. Dans ce cas spécifique, la pression relative à cette particule est négative selon l'équation d'état (2.17), or selon HPLSD, la valeur est forcée à zéro et la différence de densité ($\rho^{avg} - \rho_0$) $> \epsilon^{densité}$, contrôlant l'arrêt de la correction du solveur de l'EPP en invariance de densité est ignorée (voir l'algorithme 6). Celui-ci quittera donc sa boucle itérative de prédiction-corrrection sans produire de force de pression corrective. La force de pression empêchant la compression du fluide, c'est-à-dire quand $\rho_i > \rho_0$, n'empêche pas la dilatation de celui-ci (quand $\rho_i < \rho_0$). Il apparaît alors que la contrainte de divergence zéro (équation (1.6)), traduisant l'incompressibilité du fluide, n'est qu'en partie respectée. Si la correction du solveur de l'EPP avait été parfaite, alors toutes les particules auraient été colorées de la même couleur sur la figure 3.1. Or, certaines particules bleues (densité $\rho_i < \rho_0$) sont observées dans le rectangle rouge au milieu des particules roses (densité $\rho_i = \rho_0$). En temps normal, cette dilatation locale reste invisible, car la méthode HPL est surtout utilisée pour simuler un liquide en immersion dans de l'air faussement simulé, et la gravité assure que le liquide se comprime naturellement vers le fond du domaine (voir la sous-section 2.2.4.2). Néanmoins, si la contrainte d'incompressibilité du fluide n'est pas respectée, notamment sur le passage du flux d'air, alors le modèle physique choisi, c'est-à-dire la forme incompressible des équations de Navier-Stokes, se révèle invalide puisque sa condition d'utilisation (contrainte de divergence zéro) n'est pas respectée. La conséquence est montrée à la sous-section B.3.7.

Pour corriger la dilatation du fluide, il est donc nécessaire de se pencher sur

3.1. LÉVITATION D'UNE BALLE DANS UN SOUFFLEUR

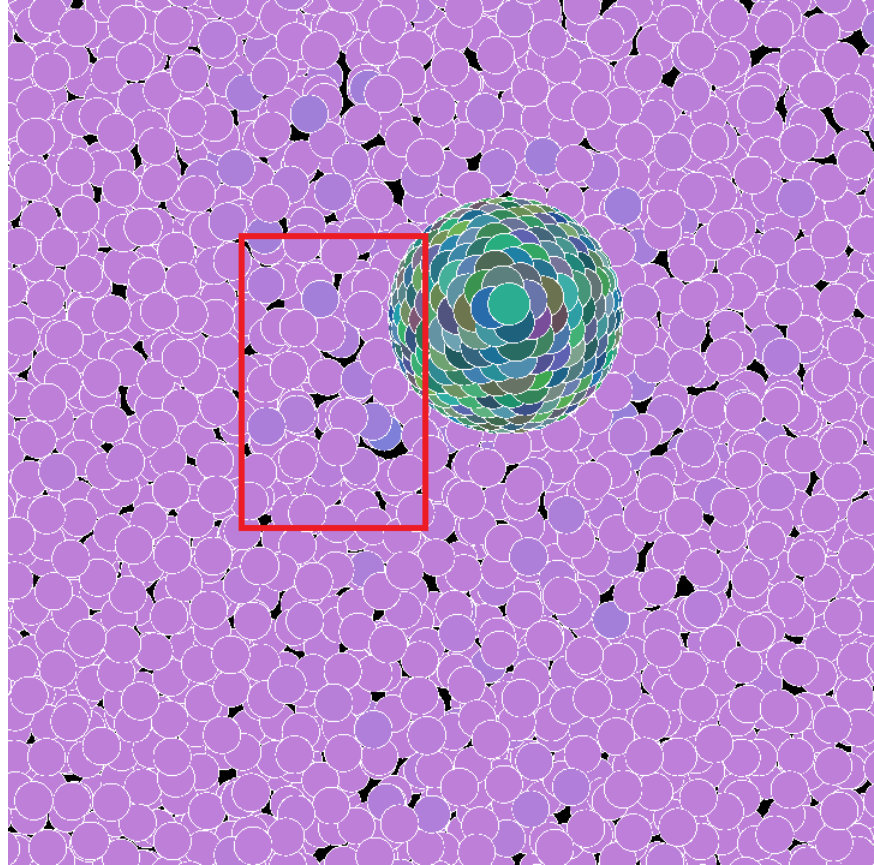


Figure 3.1 – Correction partielle de notre implémentation du résolveur de l'EPP prenant l'invariance de densité en source d'entrée. La méthode de simulation utilisée est HPLSD. Le plan proche et le plan éloigné ont été rapprochés pour produire une coupe mince du domaine de simulation. Une particule de fluide bleue possède une densité plus faible qu'une particule rose.

3.2. PRINCIPE DE BERNOULLI

les principes responsables de la lévitation de la balle, afin de compléter la partie manquante de la force de pression amenant une succion du gaz environnant. Deux explications furent énoncées :

- Le principe de Bernoulli apporte une relation mathématique utilisable pour expliquer le phénomène de lévitation d'une balle dans un flux d'air [7, p.350].
- Certaines ressources supportent néanmoins que le principe de Bernoulli a peu d'apports dans le phénomène et que l'effet de Coandă en soit le principal responsable [18].

La littérature n'est pas encore clairement unanime sur le sujet, mais nous n'avons pas trouvé de référence sérieuse pour appuyer ceci. Cependant, nous ne débattons pas de ce point dans nos travaux, car nous ne possédons pas les connaissances techniques pour l'affirmer ou le réfuter.

3.2 Principe de Bernoulli

Le principe de Bernoulli décrit une relation mathématique entre la pression et la vitesse du fluide sur une même ligne aérodynamique (*streamline*) pour un écoulement rotationnel ou à deux points quelconques pour un écoulement irrotationnel [2, pp.209-213], soit

$$p + \frac{1}{2}\rho||\vec{u}||^2 = Constante. \quad (3.1)$$

D'après l'équation de Bernoulli (3.1), il est apparent que lorsque la vitesse $||\vec{u}||$ du fluide augmente alors la pression p diminue, et inversement, la pression augmente si la vitesse diminue [2, pp.209-213].

Pour rappel, un écoulement est dit irrotationnel si $\nabla \times \vec{u} = \vec{0}$ en tout point de celui-ci. Sinon l'écoulement est dit rotationnel. Pour plus de détail, le lecteur est invité à regarder le livre de J.D. Anderson [2, pp.169-170].

3.3 Effet Coandă

Selon la définition, l'effet Coandă est la tendance d'un fluide de s'attacher et de suivre la surface d'un solide quelconque sur laquelle il s'écoule [48]. Il s'agit prin-

3.4. EFFET D'UNE PRESSION NÉGATIVE

cipalement d'une observation et nous n'avons pas trouvé de relation mathématique associée à cet effet. Cependant, D.J. Tritton [48] explique que l'effet Coandă est le résultat du déplacement des particules proches de la surface du solide entraînées par les particules dans le flux du fluide en raison de la viscosité. Ce déplacement crée du vide entre le flux et le solide, ce qui a pour effet d'aspirer du fluide depuis les zones environnantes par un effet de succion, y compris le fluide présent dans le flux, ce qui le dévie pour épouser la surface du solide. Comme explicité dans la [section 3.1](#), l'EPP en invariance de densité ne semble pas produire de correction si $\rho_i < \rho_0$, or l'effet Coandă s'explique parce qu'une force de pression résultant de la succion se produit pour ramener du fluide dans le vide qui se crée. Puisque cette observation est valable pour tous les fluides tandis que le principe de Bernoulli n'est valable que pour les fluides irrotationnels [2, pp.209-213], nous avons donc préféré utiliser cet effet que nous supposons être la pièce manquante de la force de pression pour corriger la dilatation du fluide proche du solide dans le flux.

Une force artificielle entraînant un effet de succion, $\vec{F}_{Coandă}$, a donc été créée de toute pièce en accord avec les explications du phénomène (effet de Coandă). La force de pression calculée par HPLSD a pour objectif de garder les particules à distance les unes des autres pour imposer la non-compression du fluide (c'est-à-dire empêcher $\rho_i > \rho_0$), tandis que notre force de succion assure la non-dilatation du fluide autour d'un solide (c'est-à-dire empêcher $\rho_i < \rho_0$ autour du solide). Celle-ci est donnée par

$$\vec{F}_{Coandă} = \frac{m_i}{\Delta t^2} \sum_j C_j \vec{x}_{ij}, \quad (3.2)$$

avec C_j une constante sans dimension attribuée uniquement aux particules de solide, définie par l'utilisateur, pour moduler l'amplitude de la force de pénalité. Celle-ci peut posséder une valeur différente entre les divers solides.

3.4 Effet d'une pression négative

Nous avons recours à une force additionnelle présentée dans la [section 3.3](#), car il est généralement admis dans la littérature [4, 5, 25, 26, 28, 31, 34], [21, p.26] [32, p.558] [37, p.775] qu'il est nécessaire de toujours conserver une pression positive pour

3.4. EFFET D'UNE PRESSION NÉGATIVE

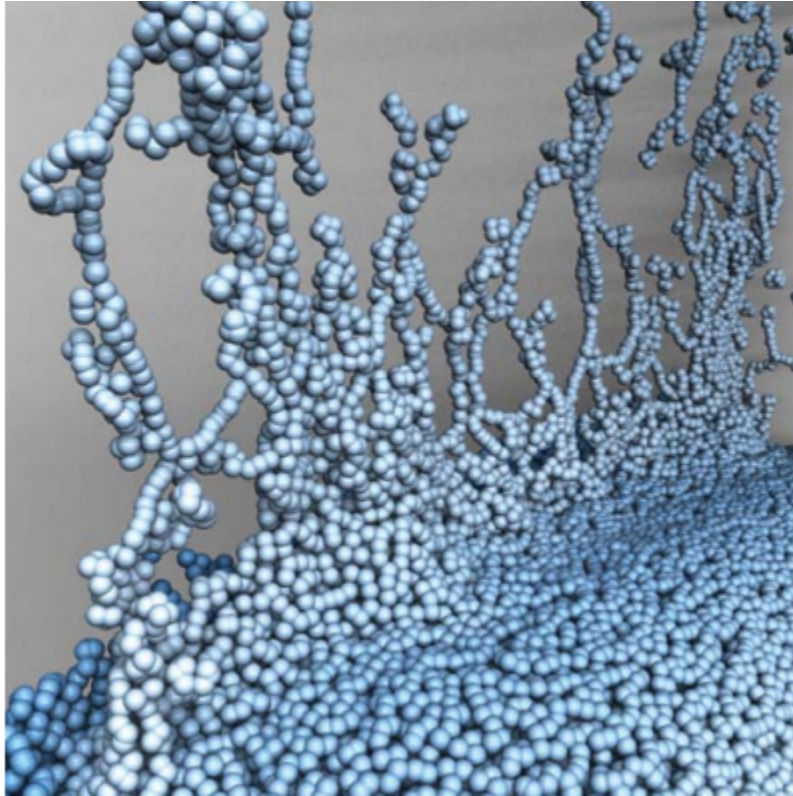


Figure 3.2 – Artefact d’agrégation non physique des particules en raison d’une pression négative montrant l’effet des instabilités de traction et de couplage. L’image est extraite de l’article de M. Ihmsen et al. [25].

éviter les instabilités de traction (*tensile instabilities*) et les instabilités de couplage (*pairing instabilities*) [37, p.775].

Nous référons à l’article de D.J. Price [37, pp.775-776] pour une définition suivie d’une analyse complète de ces deux types d’instabilités et dont voici un résumé :

- Les instabilités de traction sont données comme une agrégation anormale non physique entre deux particules voisines en raison que la force de pression, normalement utilisée comme force répulsive, change de signe et devient attractive [37, p.775] comme montré à la [figure 3.3](#).
- Les instabilités de couplage sont données comme une erreur survenant lorsqu’une paire de particules voisines se rapprochent suffisamment l’une de l’autre, la répulsion amenée par la force de pression entre la paire atteint un maximum

3.4. EFFET D'UNE PRESSION NÉGATIVE

quand $\frac{\|\vec{x}_{ij}\|}{h} \approx \frac{2}{3}$, puis diminue jusqu'à atteindre une force nulle lorsque les particules de la paire se chevauchent [37, pp.775-776].

Ces deux instabilités empêchent une paire de particules s'étant attirée en raison des instabilités de traction, de se séparer par la suite en raison des instabilités de couplage, amenant ainsi à des agrégations compactes non physiques des particules comme montré dans la [figure 3.2](#).

La raison pour laquelle il est alors nécessaire d'empêcher une pression négative pour éviter ces instabilités est présentée dans l'analyse de D.J. Price énonçant que l'approximation du gradient faite selon la forme symétrique (équation (2.9)), notamment son application pour générer la force de pression donnée par l'équation (2.29), est une mauvaise approximation. Toutefois, l'équation (2.9) permet la conservation de la quantité de mouvement et contient une procédure intrinsèque pour réordonner localement les particules, condition nécessaire pour obtenir une simulation stable [37, pp.773-776].

La [figure 3.3](#) montre la valeur calculée de la force de pression s'appliquant entre deux particules faisant varier leur densité autour de la densité de repos du fluide. La couleur orange témoigne d'une résultante positive de la force de pression amenant à une répulsion des deux particules. À l'inverse, la couleur bleue témoigne d'une résultante négative de la force de pression amenant à une attraction des deux particules. Il apparaît alors que lorsque $\rho_i > \rho_0$ et $\rho_j > \rho_0$, la forme symétrique produit toujours une force de pression positive. Quand $\rho_i < \rho_0$ et/ou $\rho_j < \rho_0$, la force de pression générée est négative sous une certaine valeur. Or une force de pression négative entraîne l'agrégation anormale des particules produisant des instabilités de traction et de couplage, d'où la nécessité de certaines méthodes HPL (HPLII ou HPLSD) d'empêcher explicitement la densité ρ_i de devenir inférieure à ρ_0 , en assignant manuellement zéro à la valeur de la pression. Toutefois, cette solution introduit des artefacts où la contrainte de divergence zéro (équation (1.6)) n'est plus respectée, et donc que la forme incompressible des équations de Navier-Stokes devient inapplicable comme expliqué dans la [section 3.1](#).

3.5. IMPLÉMENTATION

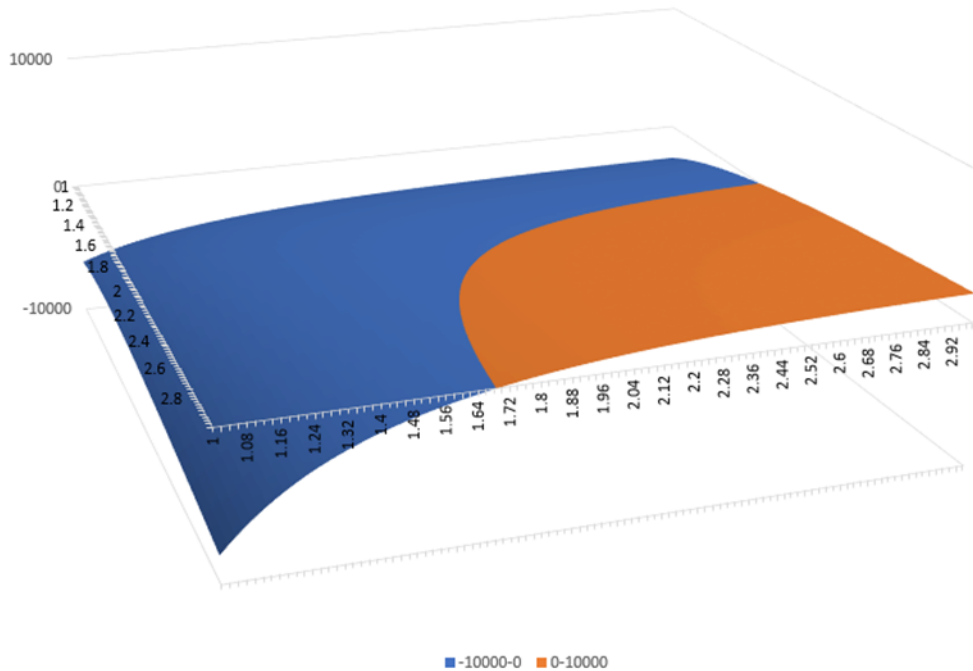


Figure 3.3 – Surface de la courbe de la forme symétrique de l’approximation du gradient appliquée à la force de pression en fonction des valeurs de densité des particules i et j . La couleur orange témoigne d’une force de pression positive tandis que la couleur bleue témoigne d’une force négative. La densité de repos ρ_0 est fixée à $2,0 \text{ kg.m}^{-3}$. ρ_i et ρ_j varient entre $1,0 \text{ kg.m}^{-3}$ et $3,0 \text{ kg.m}^{-3}$. La masse et la distance entre les particules i et j sont fixées à une constante quelconque. Les pressions p_i et p_j sont calculées à partir de ρ_i et ρ_j respectivement en utilisant l’équation de Tait (2.18).

3.5 Implémentation

Nous avons réalisé notre propre moteur de simulation, nommé *Storm*, et dont le lien *Github* est fourni dans la [section B.2](#).

3.5.1 Gestion des conditions aux frontières

La surface des corps solides est échantillonnée en particule frontière selon l’approche particulière de N. Akinci et al. [1] (voir la [section 2.4](#)), soit par un échantillonnage en disques de Poisson (*Poisson Disk Sampling*), et dont nous référons le lecteur

3.5. IMPLÉMENTATION

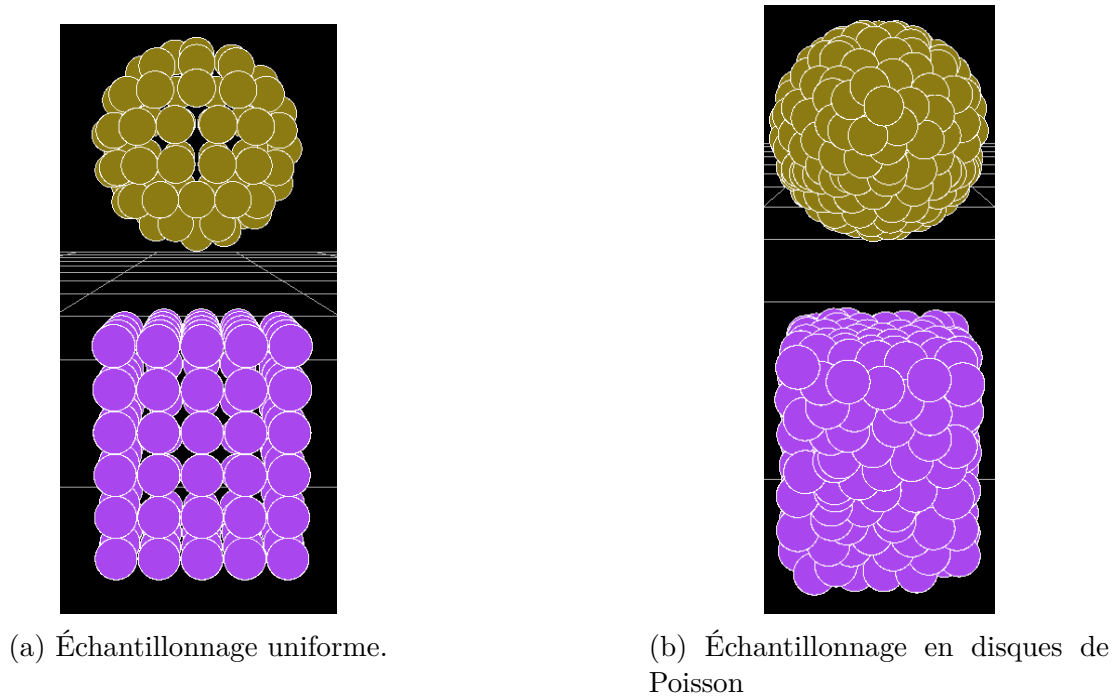


Figure 3.4 – Échantillonnage des corps solides.

aux travaux de R. Bridson [8] pour davantage de détails ; soit par un échantillonnage uniforme d’après les travaux de M. Deserno [15]. Les deux façons d’échantillonner montrées dans la [figure 3.4](#) ont été comparées ensemble et, bien que l’échantillonnage en disques de Poisson distribue les particules uniformément le long de la surface du solide, il apparaît que le maillage présente localement des trous, comme montré dans la [figure 3.5](#). Toutefois, l’échantillonnage en disques de Poisson permet la distribution uniforme sur une géométrie complexe, tandis que l’échantillonnage uniforme ne semble adapté que pour les géométries simples (parallélépipède rectangle, sphère). En effet, N. Akinici [1] spécifie que celui-ci semble produire un échantillonnage trop dense à la jointure des maillages où la courbure est importante. Il est donc nécessaire de normaliser les noyaux en appliquant le coefficient γ obtenu depuis l’équation (2.51) pour compenser les biais de trajectoires, conséquence de la non-uniformité locale du maillage (voir la [section 2.4](#)).

La solution consistant à échantillonner plusieurs couches sur la surface d’un solide a aussi été réalisée pour une particule de fluide au contact de la frontière afin de

3.5. IMPLÉMENTATION

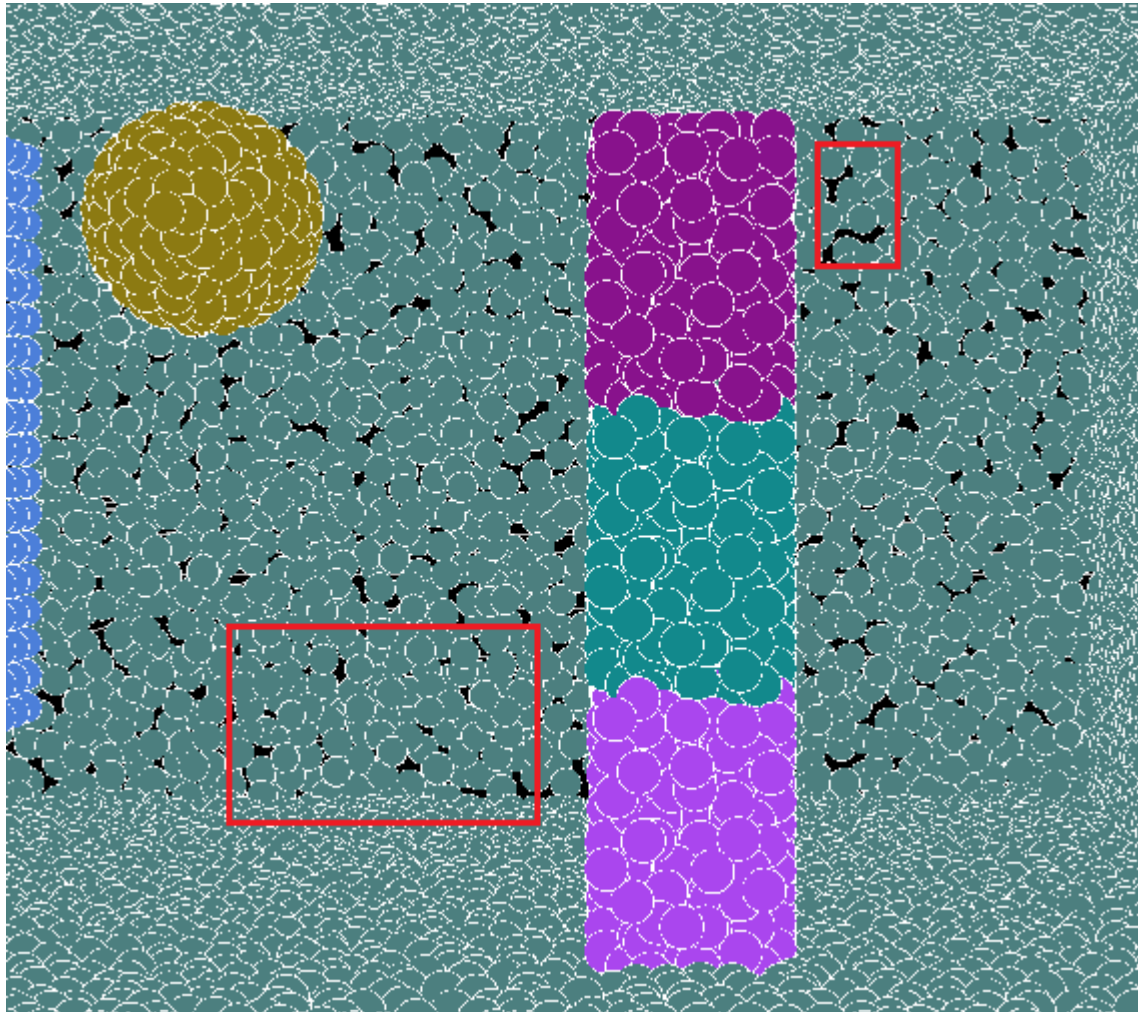


Figure 3.5 – Trous formés dans le maillage lors de l'échantillonnage en disques de Poisson. Les carrés rouges soulignent des zones où des trous peuvent être aperçus.

3.5. IMPLÉMENTATION

réduire la sous-estimation du voisinage à l'intérieur du solide (voir la [section 2.4](#)). Cependant, puisque le nombre de particules augmente, cela accroît la consommation mémoire et le temps de calcul par pas de temps de la simulation. La correction de l'interpolation à la frontière selon l'équation (2.53) a donc été préférée.

3.5.2 Initialisation des quantités (scalaires et vectorielles) physiques des particules de fluide

Chaque particule de fluide i générée possède une densité ρ_i initialisée à la densité de repos du fluide ρ_0 , une vitesse \vec{v}_i , une position \vec{x}_i , la résultante des forces \vec{F}_i s'appliquant sur celle-ci et une masse $m_i = \rho_0 V_p$ avec V_p , le volume d'une particule quelconque p , tel que $V_p = c_V d_p^3$; c_V étant un coefficient déterminé par l'utilisateur et d_p le diamètre d'une particule quelconque tel que décrit à l'équation (2.10).

À noter que D. Koschier et al. [26] proposent un simulateur de fluide dont le code source est ouvert, nommé *SPlisHSPlasH* (le lien est fourni dans la [section B.1](#)), qui implémente un grand nombre de méthodes HPL présentées dans la [sous-section 2.2.5](#) et dont nous nous sommes inspirés pour notre implémentation. Celui-ci fixe empiriquement $c_V = 0,8$ afin d'éviter les pics de pression au début de la simulation, néanmoins nous avons déterminé expérimentalement que cette valeur ne permet pas de faire léviter une balle dans un souffleur. La [figure 3.6](#) montre qu'il existe une corrélation directe entre c_V et le remplissage du domaine. En effet, si $c_V = 0,8$, alors le fluide occupera un volume de 80% de son volume initial et le domaine se videra donc à 20%. À l'opposé, une valeur c_V égale ou très proche de 1,0 amènera le fluide à occuper tout le volume du domaine, ce qui est plus adaptée pour la simulation d'un gaz selon nous, contrairement à un liquide dont la perte de volume est peu gênante pour l'utilisateur, puisque le liquide doit se tasser au fond du domaine par l'effet de la gravité.

3.5.3 Le moteur de simulation

La librairie DirectX est utilisée pour réaliser le moteur graphique, mais la visualisation du fluide reste minimaliste : toutes les particules sont représentées par des sphères puisque les méthodes de visualisation, n'ayant aucune incidence sur les phéno-

3.5. IMPLÉMENTATION

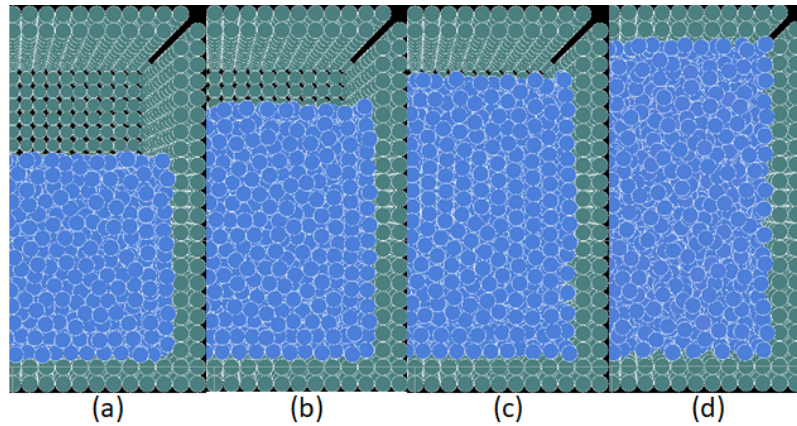


Figure 3.6 – Remplissage du domaine en fonction de diverses valeurs de coefficient c_V initialisant la masse d’une particule. Le fluide remplit à 60% du domaine quand $c_V = 0,6$ dans (a), 80% du domaine quand $c_V = 0,8$ dans (b), 90% du domaine quand $c_V = 0,9$ dans (c) et tout le volume quand $c_V = 1,0$ dans (d).

mènes physiques sous-jacents et que l’air est un fluide transparent dans des conditions naturelles terrestre, n’ont que peu d’intérêt dans nos travaux.

La librairie PhysX est utilisée comme moteur de simulation physique afin de simuler le mouvement de tous les solides (rotation, translation, collisions). Néanmoins aucune particule de fluide n’est connue de PhysX. Notre moteur calcule et envoie la résultante des forces s’appliquant sur les particules de solide dynamique en réaction avec toutes les particules de fluide de son voisinage, et s’occupe de toutes les interactions fluide \leftrightarrow fluide avant de demander à PhysX de recalculer la position de tous les solides pour le temps de simulation courant. À noter qu’afin d’optimiser le calcul, aucune résultante des forces n’est envoyée à PhysX pour les particules des solides statiques, mais ceux-ci appliquent néanmoins la force de réaction due à la troisième loi de Newton (équation (1.3)) sur chacune des particules de fluide ayant participé au calcul de celui-ci.

Pour finir, notre moteur utilise toutes les ressources de l’unité centrale de traitement (*Central Processing Unit (CPU)*) disponibles. La raison pour laquelle le processeur graphique (*Graphics Processing Unit (GPU)*) n’a pas été utilisé fut pour faciliter le développement et le débogage. Néanmoins, les performances sont restées satisfaisantes, car il a été possible de répartir l’ensemble des particules sur des boucles

3.5. IMPLÉMENTATION

Tableau 3.1 – Performances entre *Storm* et *SPlisHSPlasH*.

moteur	scene	nombre de particule	Δt [s]	temps de calcul pour 1s de simulation [s]
<i>Storm</i>	cassure de barrage	125k	0,001	84,3
	balle en lévitation	200k	0,0002	415,3
<i>SPlisHSPlasH</i>	cassure de barrage	125k	0,001	46,0
	balle en lévitation	200k	-	-

de traitement en parallèle, et de traiter certains calculs comme la recherche de voisinage en utilisant des instructions IUDM (Instruction Unique, Données Multiples ; soit *Single Instruction Multiple Data (SIMD)*). Cependant, il apparaît sur le [tableau 3.1](#) que l’implémentation de HPLSD faite par *SPlisHSPlasH* est 83% plus performante que notre implémentation avec une scène simulant de l’eau en configuration de cassure de barrage avec 125k particules. Cela s’explique car bien que *Storm* utilise les jeux d’instructions *SSE 4.2* et *AVX-512* pour la recherche de voisinage et la mise à jour de la position des particules, *Storm* n’emploie pas ces jeux d’instructions dans les solveurs de l’EPP employés par HPLSD contrairement à *SPlisHSPlasH*. En contrepartie, *SPlisHSPlasH* est instable pour simuler une balle immergée sous un souffleur d’air placé en position droite (voir la [sous-section 3.6.2](#)) contrairement à *Storm*.

Toutefois, malgré cette différence de performance, nous avons préféré continuer nos recherches sur *Storm* au lieu de *SPlisHSPlasH* pour deux raisons principales :

- Les configurations de scène dans *SPlisHSPlasH* sont en *JSON*. Toutefois, ce format n’accepte pas d’y insérer des commentaires. En raison du nombre important de paramètres associés à la simulation (voir la [section 3.1](#)), ne pas pouvoir commenter temporairement des valeurs pour en essayer d’autres se révèle être un obstacle majeur pour trouver le bon jeu de paramètres. Nous avons donc choisi d’utiliser le format *XML* qui, selon nous, est plus propice pour effectuer des tests efficaces.
- Il n’y avait plus besoin de synchroniser les modifications au code de *SPlisHSPlasH* avec nos changements.

Les algorithmes [9](#) et [10](#) reprennent l’initialisation et la boucle principale de la simulation respectivement.

3.5. IMPLÉMENTATION

Algorithme 9 Moteur Initialisation

- 1: **fonction** INITIALISATION
 - 2: Partitionne l'espace en grille pour optimiser la recherche de particule selon leur position ([sous-section 2.2.2](#))
 - 3: Génère les particules de fluides dans le domaine
 - 4: Échantillonne les solides en particule (échantillonnage uniforme, échantillonnage en disques de Poisson ou à partir d'un fichier)
 - 5: Retire les particules de fluide dans les solides
 - 6: **fin fonction**
-

3.5. IMPLÉMENTATION

Algorithme 10 Moteur Exécution

```
1: INITIALISATION => Algorithme 9
2: tant que  $t_{écoulé} < t_{fin\_de\_simulation}$  faire
3:   pour tout particule  $i$  faire
4:     Enregistre la position de la particule dans la partition de l'espace
5:   fin pour
6:   pour tout particule  $i$  faire
7:     Génère le voisinage de la particule en considérant la cellule de la partition
       contenant la particule et toutes les cellules voisines (sous-section 2.2.2)
8:   fin pour
9:   si HPLFC alors
10:    Applique HPLFC telle que formulée dans la sous-section 2.2.5.3
11:  sinon si HPLIPC alors
12:    Applique HPLIPC telle que formulée dans la sous-section 2.2.5.4
13:  sinon si HPLII alors
14:    Applique HPLII telle que formulée dans la sous-section 2.2.5.5
15:  sinon si HPLSD alors
16:    Applique HPLSD telle que formulée dans la sous-section 2.2.5.7
17:  fin si
18:  Mets à jour la position des solides avec PhysX
19:  pour tout particule  $i$  faire ▷ Euler semi-Implicite
20:    Calcule  $v_{i+} = \Delta t a_i$  (voir équation (2.45))
21:    Calcule  $x_{i+} = \Delta t v_i$  (voir équation (2.46))
22:  fin pour
23:   $t_{écoulé+} = \Delta t$ 
24:  Faire toutes les opérations annexes (enregistrement dans un fichier pour re-
       lecture en temps réel, copie position des particules vers le fil d'exécution (thread)
       graphique, ...)
25: fin tant que
```

3.6. RÉSULTATS

3.6 Résultats

Excepté les simulations de validations présentées dans la [sous-section 3.6.1](#) ayant servi pour tester les interactions fluides↔solides, tous nos résultats simulant une balle en lévitation sous un souffleur ([sous-section 3.6.2](#) et [sous-section 3.6.3](#)) ont été comparés avec des tests dans la réalité dont les liens vidéos sont fournis en [annexe B](#).

3.6.1 Simulation de validation

Une scène en configuration de cassure de barrage interagissant avec de multiples objets solides a été créée afin de valider l'implémentation correcte du couplage bidirectionnel fluide↔solide essentiel pour faire léviter une balle dans un flux d'air (voir [figure 3.7](#)).

Il apparaît d'abord que le fluide, apparenté à de l'eau, emporte les solides. Donc que des forces sont appliquées du fluide sur les solides. Le parallélépipède rectangle jaune n'est quasiment pas emporté en raison de sa masse/densité largement supérieure à l'eau, tandis que les autres solides sont aisément emportés. En retour, les solides dévient le fluide sur le passage de la vague et obligent les particules de fluide à les contourner pour se rendre derrière. Le fluide ne traverse donc pas les solides, car ceux-ci appliquent des forces qui repoussent le fluide et l'empêchent de pénétrer. Nos résultats, dont le lien vers la vidéo de la simulation est retrouvé dans la [sous-section B.3.3](#), modélisent donc le couplage bidirectionnel fluide↔solide.

À cela, nous validons aussi la rotation des solides avec une scène où un solide dont la translation est fixée à $\vec{0}$ en son centre, mais pouvant tourner librement autour de tout les axes de rotation, est maintenu sous une masse d'eau. Ce qu'il convient d'observer dans les résultats présentés par la [figure 3.8](#), est qu'après que la masse d'eau se soit déversée sur un côté du solide, celui-ci, entraîné par le mouvement du fluide, tourne autour de l'axe passant par son centre et sortant de l'image (perpendiculaire au sens du mouvement de la chute d'eau). Puis la rotation s'inverse quand la vague rebondit sur la paroi du domaine et repousse l'autre versant du solide en dessous.

3.6. RÉSULTATS

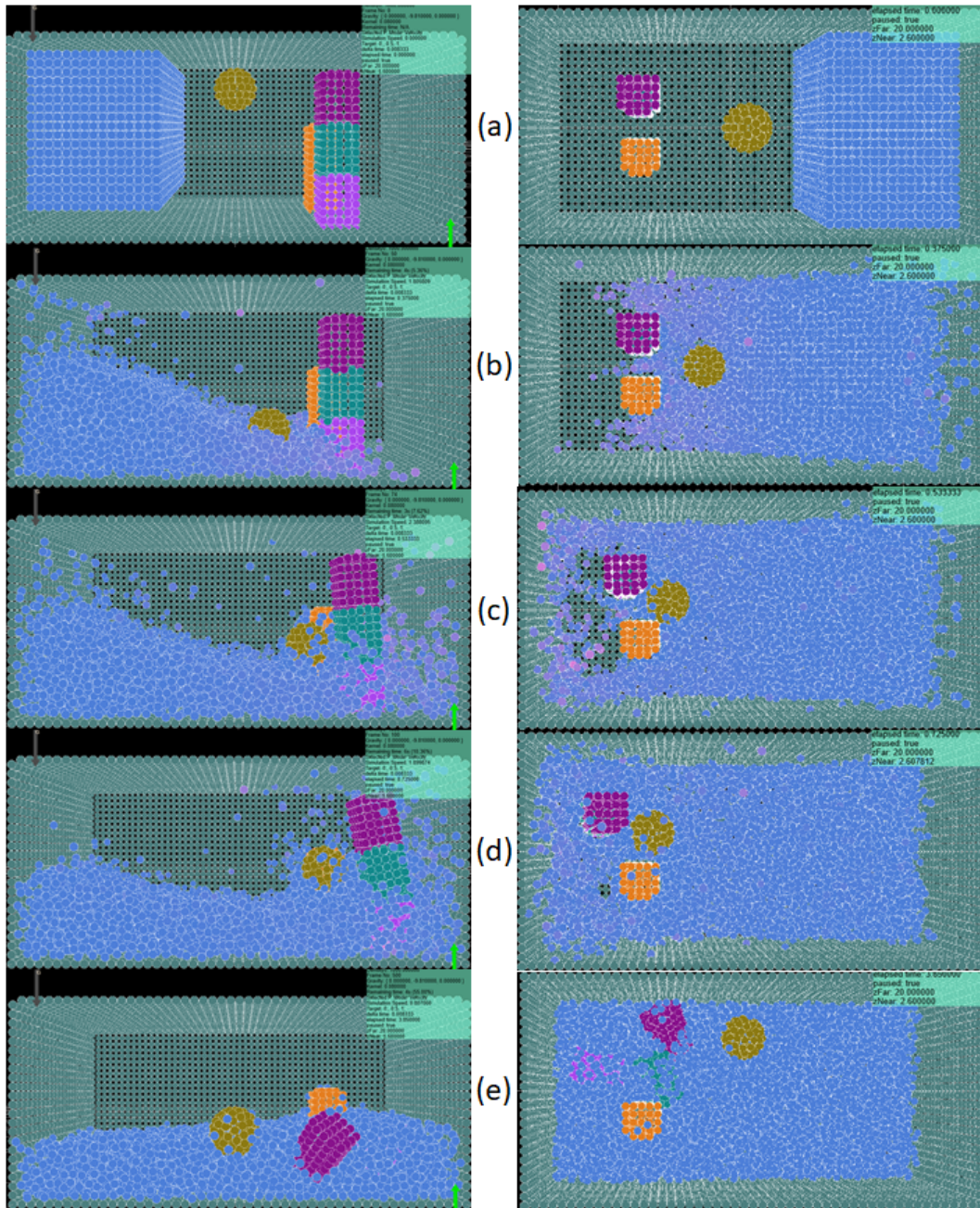


Figure 3.7 – Simulation en cassure de barrage avec plusieurs solides. Le fluide présente les constantes physiques de l'eau. La balle jaune pèse $m = 2,5 \text{ kg}$ pour un volume $V = 0,0042 \text{ m}^3$, le parallélépipède rectangle jaune pèse $m = 125 \text{ kg}$ pour un volume $V = 0,01125 \text{ m}^3$ tandis que les autres parallélépipèdes rectangles pèsent $m = 8 \text{ kg}$ pour un volume $V = 0,0045 \text{ m}^3$. (a) présente la simulation au temps $t = 0 \text{ s}$, (b) à $t = 0,375 \text{ s}$, (c) à $t = 0,533 \text{ s}$, (d) à $t = 0,725 \text{ s}$ et (e) à $t = 3,85 \text{ s}$. Les images de gauche montrent une vue de face et celles de droites ont une vue du dessus.

3.6. RÉSULTATS

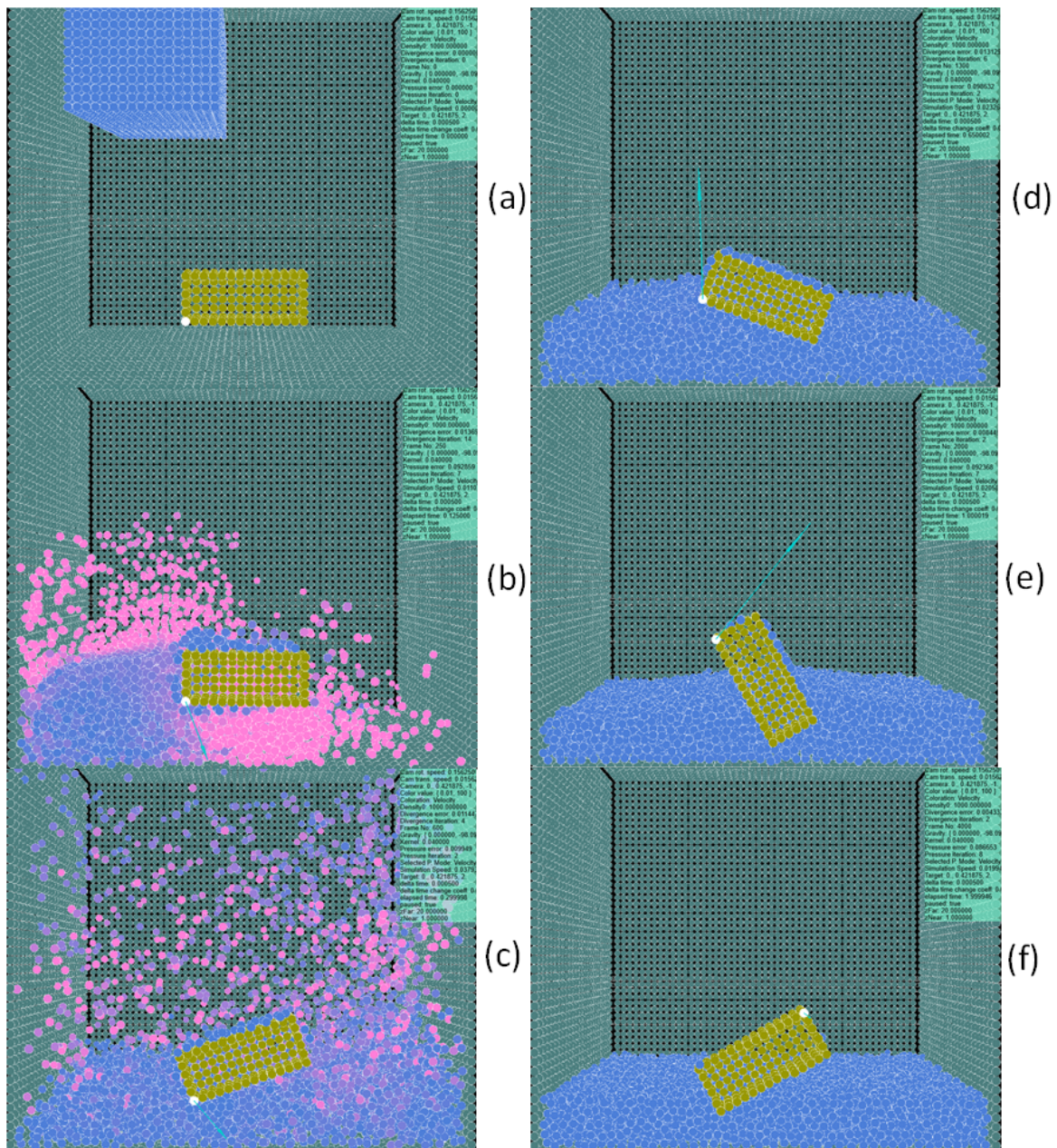


Figure 3.8 – Rotation du solide jaune sous l’effet d’une masse d’eau tombante. Une couleur de particule de fluide rouge traduit une norme de vitesse plus élevée. (a) présente la simulation au temps $t = 0 s$, (b) à $t = 0,125 s$, (c) à $t = 0,3 s$, (d) à $t = 0,65 s$, (e) à $t = 1 s$ et (f) à $t = 2 s$. La flèche bleue montre le vecteur vitesse à la particule blanche du solide.

3.6. RÉSULTATS

3.6.2 Balle en lévitation sous un souffleur en position droite

D'après les vidéos de la réalité montrées dans la [sous-section B.3.5](#), montrant un souffleur (sèche-cheveux) tenu à la verticale vis-à-vis du sol et dont la vitesse de sortie d'air est d'environ 40 m.s^{-1} , il apparaît dans la [figure 3.9](#) qu'une fois à l'équilibre, notre balle de ping-pong d'une masse $m = 3 \text{ g}$ et d'un diamètre de 40 mm (soit $0,02 \text{ m}$ de rayon), reste dans le flux d'air tout en oscillant perpendiculairement à l'axe du flux d'air (gauche-droite et avant-arrière) et colinéairement à l'axe du flux d'air (haut-bas). Sa rotation oscille, quant-à-elle, entre -20 et 20 degrés autour des axes perpendiculaires à la direction du flux de l'air. Nous pensons que sa rotation autour de l'axe colinéaire à la direction du flux d'air est due aux erreurs d'expérience introduites par nos doigts lors du lâché de la balle, car lorsque la balle est maintenue en lévitation suffisamment longtemps, celle-ci s'arrête naturellement de tourner autour de l'axe colinéaire à la direction du flux d'air. Toutefois, la balle continue de tourner autour des axes perpendiculaires à la direction du flux d'air, même après une dizaine de minutes d'attente.

3.6. RÉSULTATS



Figure 3.9 – Balle de ping-pong en lévitation sous un souffleur (sèche-cheveux) en position droite. Les flèches vertes montrent l’amplitude des oscillations de la balle et la flèche rouge donne l’axe du flux d’air. La gravité dirigée vers le bas est donnée par l’axe du poteau de la lampe en arrière-plan.

Dans un premier temps, un milieu composé d’eau a été simulé (densité de repos $\rho_0 = 1000 \text{ kg.m}^{-3}$ et viscosité dynamique $\eta = 0,001 \text{ Pa.s}$). Puisque la viscosité de l’eau est plus importante que celle de l’air, une meilleure stabilité de la simulation a pu être observée, car plus de viscosité entraîne plus de force engendrée tendant à uniformiser le champ de vitesse [23, pp.272-273] et donc de dissiper l’énergie du système. Cet accroissement de stabilité permet une augmentation du pas de temps Δt , réduisant ainsi le temps global de simulation. Puisque l’air a été substitué par de l’eau, les différentes constantes impliquées ont dû être démultipliées pour rester cohérentes, notre balle possède alors une masse de $m = 3 \text{ kg}$.

Puis, dans un deuxième temps, les quantités (scalaires et vectorielles) physiques du fluide ont été approchées des valeurs de celles admises pour l’air ($\rho_0 = 1,3 \text{ kg.m}^{-3}$ et $\eta = 0,000018 \text{ Pa.s}$) et dont les résultats sont montrés sur la [figure 3.10](#) et dans la

3.6. RÉSULTATS

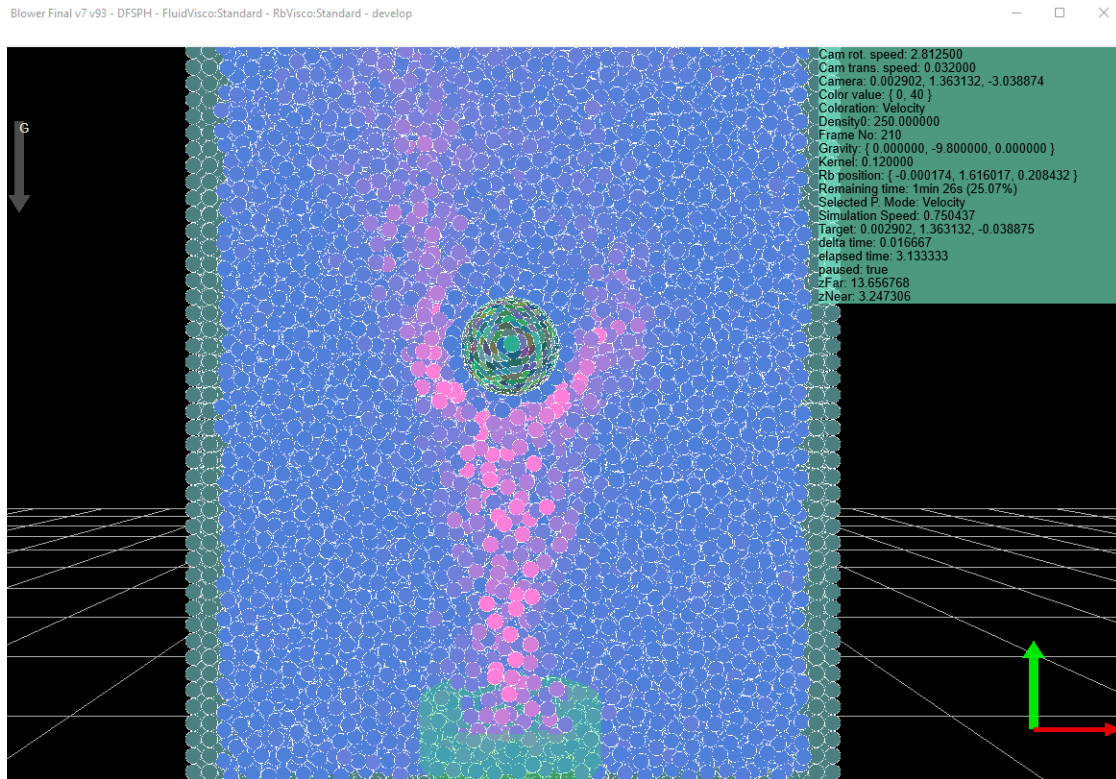


Figure 3.10 – Simulation d’une balle sous un souffleur placé à la verticale. La zone d’effet du souffleur est montrée par le cylindre vert. Plus la vitesse d’une particule de fluide est importante (jusqu’à 40 m.s^{-1}), plus sa couleur s’approchera du rouge. La flèche grise représente le vecteur de la gravité. La densité de repos du fluide est $\rho_0 = 250 \text{ kg.m}^{-3}$, sa viscosité dynamique est $\eta = 0,001 \text{ Pa.s}$ et le pas de temps de simulation est $\Delta t = 0,0002 \text{ s}$. Le coefficient de la force de Coandă est maintenu à $C_j = 0,00001$.

vidéo de la simulation dont le lien est fourni dans la [sous-section B.3.6](#).

Cependant, nous ne sommes pas parvenus à trouver un jeu de constantes valides pour simuler de l’air réel avec la méthode HPLSD et faire léviter une balle sous le souffleur, car la simulation devenait instable au fur et à mesure que ρ_0 s’approchait de zéro. À ce moment, il fallait, dans un premier temps, diminuer davantage le pas de temps Δt , augmentant ainsi le temps global de simulation. Dans un deuxième temps, augmenter la viscosité du fluide afin que les mouvements des particules générant des dilatations ramènent d’autres particules pour combler le vide. Cette dernière solu-

3.6. RÉSULTATS

tion n'est pas envisageable pour simuler un fluide peu visqueux comme l'air. Comme explicité dans la [section 3.1](#), nous soupçonnons que le résolveur de l'EPP prenant l'invariance de densité en source d'entrée ne corrige pas correctement la divergence de densité dans le cas d'une dilatation (soit le cas où $\rho_i < \rho_0$ selon l'équation (2.17)). Néanmoins, il apparaît que cette mauvaise correction est davantage mise en évidence avec un fluide peu visqueux puisque la balle lévite lorsque celui-ci est très visqueux, mais trouver un jeu de constante afin de compenser l'augmentation de viscosité se révèle être un travail fastidieux pour l'utilisateur et qui ne s'ajuste pas facilement à tout changement des conditions initiales. C'est la raison pour laquelle une force de pénalité basée sur l'effet de Coandă est ajoutée pour compenser le manque de correction du résolveur et maintenir la balle en lévitation tout en conservant une viscosité proche de celle de l'air (voir [section 3.3](#)).

3.6.3 Balle en lévitation sous un souffleur en position incliné

Somme toute, cette simulation est la même que celle en position droite (voir [sous-section 3.6.2](#)) excepté l'inclinaison à sept degrés du souffleur. La vidéo de la réalité est montrée dans la [figure 3.11](#). Il n'a pas été possible d'incliner davantage notre souffleur, car la balle sort et tombe du jet d'air; nous estimons empiriquement que, considérant le matériel à disposition, l'angle maximal se trouve entre dix et douze degrés. Néanmoins, sous un angle entre sept et dix degrés, la balle oscillait grandement et se maintenait très difficilement dans le flux d'air. N'arrivant jamais dans un état stationnaire, nous avons donc choisi de réduire cet angle à sept degrés.

3.6. RÉSULTATS

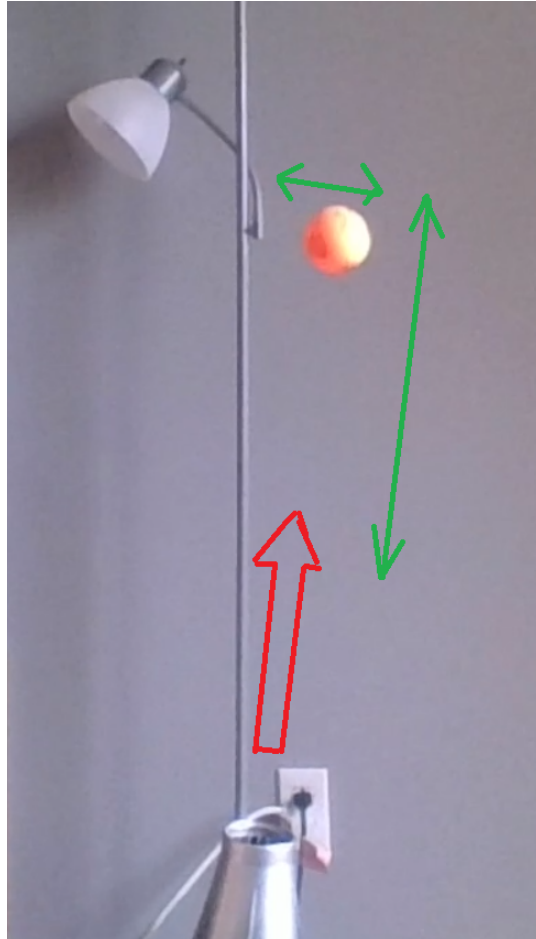
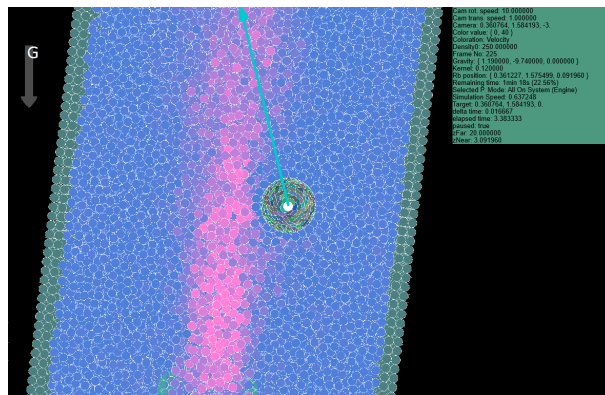


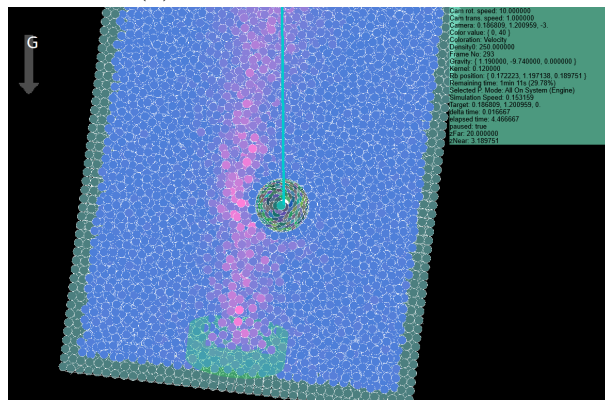
Figure 3.11 – Balle de ping-pong en lévitation sous un souffleur en position incliné de sept degrés dans la réalité. Les flèches vertes montrent l’amplitude des oscillations de la balle et la flèche rouge donne l’axe du flux d’air. La gravité dirigée vers le bas est colinéaire à l’axe du poteau de la lampe en arrière-plan.

Les résultats de la simulation utilisant notre force de pénalité associée à l’effet de Coandă (voir [section 3.3](#)) sont présentés sur la [figure 3.12](#) où il apparaît que la balle revient dans le flux de fluide, dont les quantités physiques s’approchent de celles de l’air. Le lien de la vidéo de cette simulation montrant la lévitation au complet est fourni dans la [sous-section B.3.8](#). À l’inverse, la balle ne parvient pas à se maintenir dans le flux sans notre force de pénalité associée à l’effet de Coandă comme montré dans la vidéo de la simulation dont le lien est fourni en [sous-section B.3.7](#).

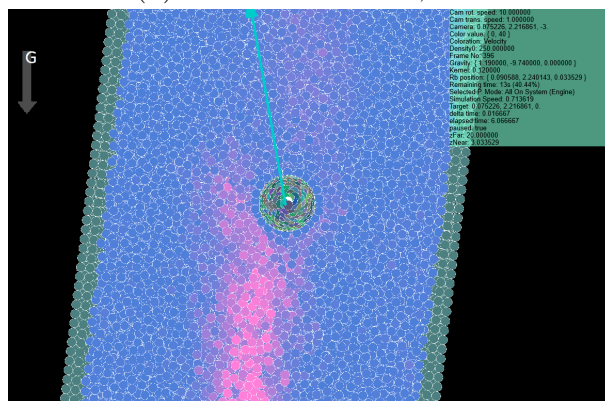
3.6. RÉSULTATS



(a) Simulation à $t = 3,38$ s.



(b) Simulation à $t = 4,47$ s.



(c) Simulation à $t = 6,07$ s.

Figure 3.12 – Simulation d’une balle en lévitation sous un souffleur incliné à sept degrés. La flèche bleue partant de la balle montre la somme des forces s’appliquant sur celle-ci au temps considéré. La flèche grise représente le vecteur de gravité. Plus la vitesse d’une particule de fluide est importante (jusqu’à $40 \text{ m}\cdot\text{s}^{-1}$), plus sa couleur s’approchera du rouge.

3.6. RÉSULTATS

À noter que pour des raisons d'optimisations, l'inclinaison est réalisée en inclinant la direction du vecteur de la force de gravité et non l'inclinaison du souffleur. La caméra d'observation a été tournée pour que le vecteur de la gravité soit dirigé visuellement vers le bas. En effet, supposant que le souffleur est un cylindre droit dont les bases sont parallèles au plan du sol, avec une position \vec{x}_c au centre du cylindre, une demi-hauteur h_s donnée par la plus courte distance entre la base et le centre du cylindre \vec{x}_c et enfin, un rayon r_c ; pour déterminer si une particule p positionnée en \vec{x} se trouve dans la zone d'effet du souffleur, soit $\vec{x}_{ic} = \vec{x} - \vec{x}_c$, alors ces deux égalités doivent être vérifiées :

$$|\vec{x}_{ic} \cdot \vec{b}| < h_s \quad (3.3)$$

et

$$(\vec{x}_{ic} \cdot \vec{a})^2 + (\vec{x}_{ic} \cdot \vec{c})^2 < r_c^2, \quad (3.4)$$

avec \vec{a} , \vec{b} et \vec{c} les vecteurs formant la base orthonormée du cylindre et dont \vec{b} est le vecteur unitaire colinéaire à l'orientation du souffleur. Ici, \vec{a} , \vec{b} et \vec{c} coïncident avec la base orthonormée du domaine de simulation si, et seulement si, le cylindre n'a subi aucune rotation. Dans ce cas spécifique, puisque $\vec{a} = \{1; 0; 0\}$, $\vec{b} = \{0; 1; 0\}$ et $\vec{c} = \{0; 0; 1\}$, il est possible d'optimiser la vérification en utilisant directement les coordonnées de \vec{x}_{ic} , sans passer par les trois produits scalaires appliqués à chaque particule. Ceci est démontré dans la [figure 3.13](#) résumant les différents vecteurs impliqués avec une utilisation des équations (3.3) et (3.4) pour déterminer si la particule p_i est à l'intérieur de la zone d'effet du souffleur cylindrique. La [figure 3.13a](#) montre le cas de p_i en dehors du cylindre, car bien que l'équation (3.4) soit respectée, l'équation (3.3) ne l'est pas. À l'inverse, p_i dans la [figure 3.13b](#) se situe en dehors du cylindre, car respecte l'équation (3.3), mais pas l'équation (3.4). Enfin, p_i dans la [figure 3.13c](#) est à l'intérieur du cylindre car elle respecte les deux équations (3.3) et (3.4).

3.6. RÉSULTATS

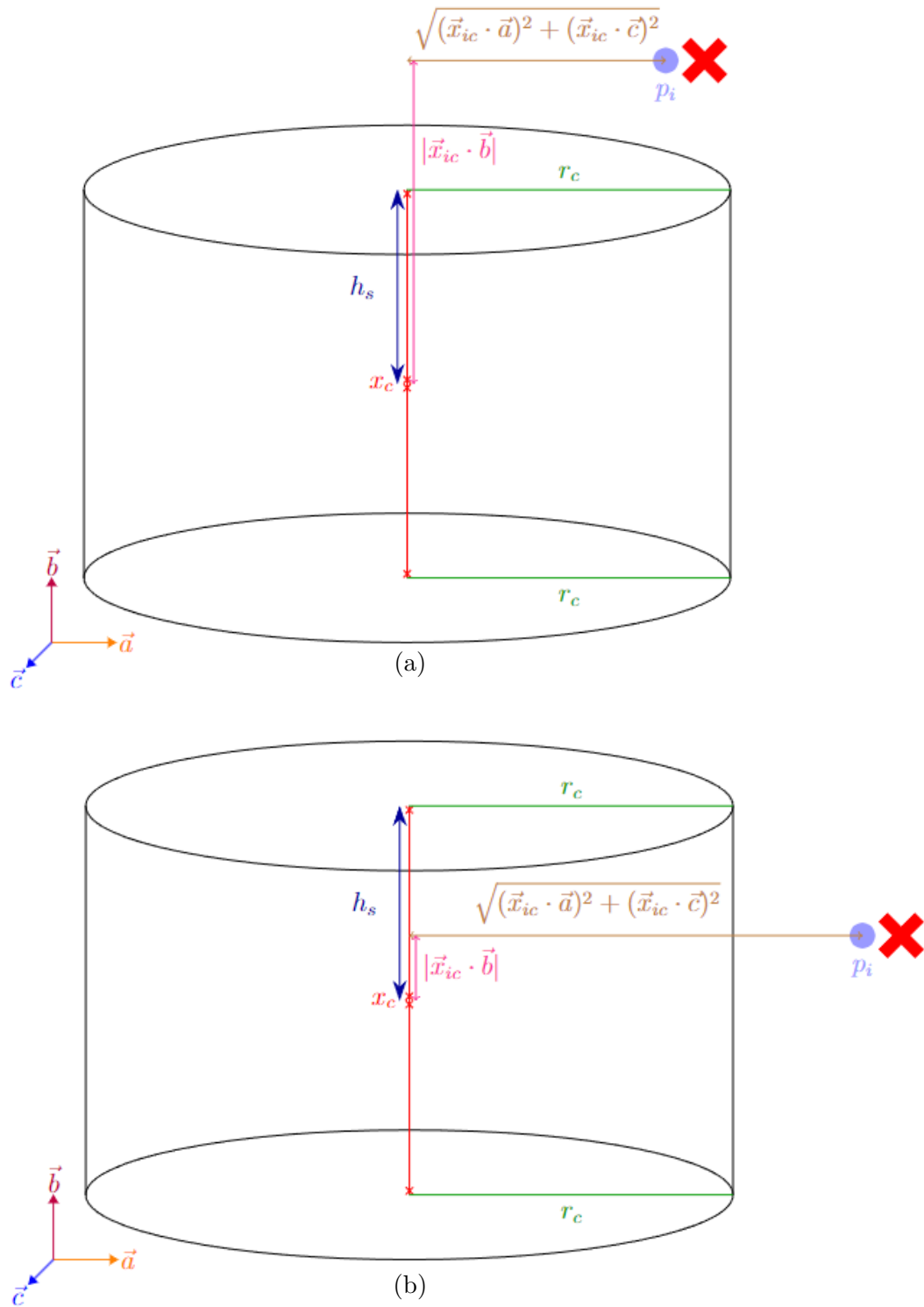


Figure 3.13 – Schéma de la zone d'effet cylindrique du souffleur.

3.6. RÉSULTATS

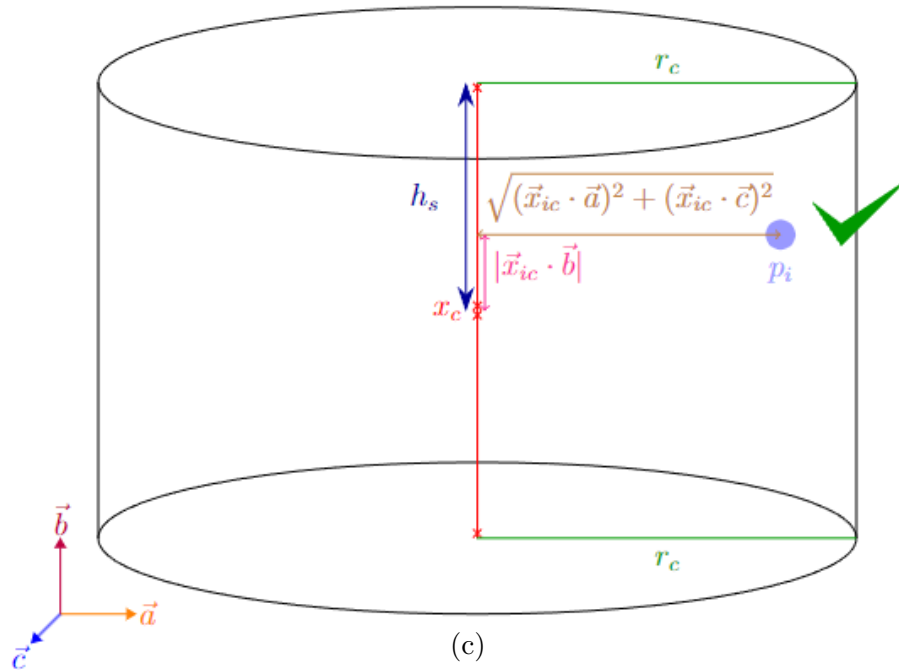


Figure 3.13 – Schéma de la zone d'effet cylindrique du souffleur.

Pour finir, le cylindre d'effet du souffleur n'est pas le seul objet nécessitant une transformation : puisque les tests dans la réalité se produisent avec une balle en lévitation dans une pièce dont il convient de supposer les murs suffisamment éloignés pour n'avoir aucun effet sur le flux d'air du souffleur. Or, puisque le fluide doit remplir l'ensemble du domaine, plus celui-ci est grand, plus le nombre de particules est important et en conséquence, les performances et la consommation mémoire s'en trouveront davantage impactées. Il convient donc de limiter la taille du domaine autour de la région d'intérêt pour diminuer le temps de simulation. Il est cependant nécessaire de s'assurer que les parois du domaine ne viennent pas faire obstacle ou dévier le sens du flux. Pour cela, les parois doivent être parallèles à la direction du flux du fluide (et suffisamment éloignées). La rotation du souffleur entraîne alors la rotation du domaine, ce qui impacte tous les algorithmes en lien avec l'espace du domaine tels que la grille spatiale (voir la [sous-section 2.2.2](#)) amenant à la recherche du voisinage.

3.7. LIMITATION

3.7 Limitation

La composante liée à l'effet Coandă ajoutée dans la forme incompressible des équations de Navier-Stokes est aussi soumise à la condition de support compacte du noyau (équation (2.3)), c'est-à-dire que toute particule positionnée à une distance $\|\vec{x}_{ij}\|$ supérieure à h , la longueur de lissage du noyau, n'aura aucune contribution. Cependant, selon l'équation (3.2), notre force de Coandă est directement proportionnelle à la distance entre la particule de solide i et la particule de fluide j dans le voisinage de i . Donc $\|\vec{x}_{ij}\|$ est maximale quand $\|\vec{x}_{ij}\| = h$, ce qui signifie que $\|\vec{F}_{Coanda}\|$ est maximale pour $\|\vec{x}_{ij}\| = h$ et est nulle pour $\|\vec{x}_{ij}\| > h$ d'après la condition de support compacte du noyau. La [figure 3.14](#) montre que cette cassure dans la continuité de la fonction calculant la force de pénalité basée sur l'effet de Coandă entraîne des artefacts où des zones de vide sont susceptibles d'apparaître autour du solide. Les particules légèrement à l'extérieur du cercle gris ne sont pas aspirées par la succion entraînée par la force de pénalité basée sur l'effet de Coandă, tandis que les particules à l'intérieur, mais juste avant la limite, sont fortement attirées en raison de leur éloignement du solide, ce qui accentue le vide montré en rouge.

La première solution à ce problème pouvant être imaginée serait d'utiliser un noyau de type gaussien (e.g. noyau spline cubique) afin de permettre à ce que la fonction de contribution de la force de pénalité basée sur l'effet de Coandă soit en tout point continue.

La deuxième solution serait de ne pas utiliser cette force de pénalité basée sur l'effet de Coandă qui, il convient de le souligner, n'est qu'un subterfuge pour remplacer la partie manquante de la composante de pression qui aurait dû corriger la dilatation du fluide. À la place, il serait davantage recommandé comme travaux futurs de corriger le solveur de l'EPP en invariance de densité utilisé par HPLSD afin de corriger correctement la densité des particules lorsque $\rho_i < \rho_0$ (voir la [section 3.1](#)).

3.7. LIMITATION

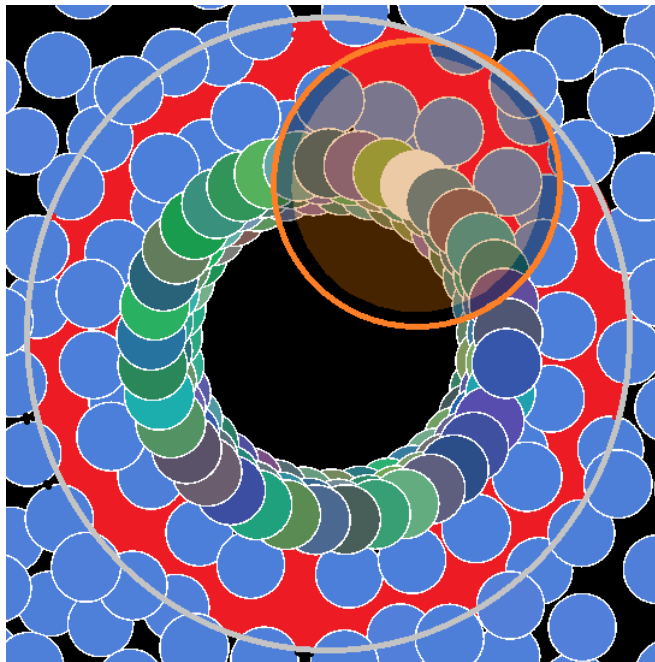


Figure 3.14 – Coupe de fluide autour d'un solide montrant le vide créé par notre implémentation de l'effet Coandă. Le cercle orange montre le champ d'action du noyau à la particule blanche. Le cercle gris montre le champ d'action total généré par toutes les particules de solide ensemble. La zone rouge montre le vide à l'intérieur de ce champ d'action.

Conclusion

Nous avons implémenté un moteur de fluide reposant sur une méthode purement lagrangienne, nommée hydrodynamique des particules lissées, afin de simuler l'interaction fluide↔solide dans un milieu immergé nécessaire à la lévitation de la balle dans un flux d'air en modifiant la méthode HPLSD [4]. Bien que ce test semble très spécifique, celui-ci a permis de mettre en évidence une erreur introduite dans plusieurs méthodes HPL ([HPL Implicitement Incompressible \(HPLII\)](#) ou [HPL Sans Divergence \(HPLSD\)](#)) qui interdisent les valeurs de pression négatives produites si la densité d'une particule de fluide ρ est inférieure à la densité de repos ρ_0 . Les instabilités de traction décrites par D.J. Price [37, p.757] sont alors empêchées, mais en contrepartie, les forces de pression en situation de dilatation n'entraînent aucune corrections. Sans ces corrections, il n'a pas été possible de maintenir la balle en lévitation sans augmenter la viscosité, ce qui amène à une simulation d'un fluide visqueux plus apparenté à du miel, et donc très différent de l'effet escompté. Notre simulateur est aussi capable de simuler d'autres situations usuelles (vague déferlante emportant divers solides dans son sillage, ou la simulation d'un objet immergé flottant ou coulant en fonction de sa densité). Cependant, puisque les solides déformables, tels que les cordes, n'ont pas été intégrés par faute de temps, certaines autres situations complexes, comme celui où deux balles attachées se rapprochent sous l'effet d'un souffleur, l'interaction d'un parachute qui doit rester ouvert et freiner la descente d'un objet attaché, ou bien le vol d'un cerf-volant en mouvement dans le vent, n'ont pas pu être testées. En contrepartie, si les instabilités de tractions sont corrigés tout en conservant les corrections induites par la force de pression, nous sommes confiant que cela permettrait à HPL de pouvoir modéliser correctement et avec plus de réalisme des objets en interaction avec des gaz ou aspirés par le fluide environnant.

CONCLUSION

Notre apport a donc principalement consisté à corriger ce problème. Nous avons introduit une nouvelle force de pénalité basée sur l'effet de Coandă pour compenser ce manque de correction des forces de pression quand $\rho < \rho_0$ autour du corps solide. Un nouveau paramètre non physique à déterminer empiriquement est introduit pour moduler cette force de pénalité basée sur l'effet de Coandă. Notre formulation de cette force entraîne néanmoins des artefacts où une couronne de vide est susceptible de se créer à la lisière de la portée de la fonction d'interpolation de HPL, puisque cette force n'utilise pas un noyau HPL respectant les conditions énumérées dans la [sous-section 2.1.1](#). Enfin, notre force s'applique uniquement autour du solide et n'entraîne donc pas de correction de la dilatation dans tout le domaine. Il serait alors intéressant comme travaux futurs de régler le problème à sa source en introduisant une nouvelle méthode générant une force de pression dans le cas de $\rho < \rho_0$.

Pour finir, notre moteur ne fait usage que de l'unité centrale de traitement et même s'il a été optimisé avec l'emploi d'une grille spatiale pour la recherche du voisinage et une architecture en multiple fils d'exécution (*multithread*) pour maximiser le parallélisme ; en raison du temps de simulation pouvant prendre plus d'une heure et demie pour obtenir quelques secondes de simulations, il convient qu'un travail d'optimisation est aussi envisageable dans le futur, notamment en employant le processeur graphique et tous les jeux d'instructions IUDM disponibles pour le calcul des boucles de corrections des solveurs de l'EPP employés dans la méthode HPLSD.

Bibliographie

- [1] N. Akinici, M. Ihmsen, G. Akinici, B. Solenthaler, et M. Teschner, « Versatile Rigid-Fluid Coupling for Incompressible SPH, » *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–8, juillet 2012. doi:[10.1145/2185520.2185558](https://doi.org/10.1145/2185520.2185558)
- [2] J. D. Anderson Jr., « Fundamentals of Aerodynamics, Sixth Edition. » Tata McGraw-Hill Education, 2017.
- [3] N. Ashgriz et J. Mostaghimi, « An Introduction to Computational Fluid Dynamics. » Fluid flow handbook, University of Toronto, ON, Canada : McGraw-Hill Professional, vol. 1, 2002.
- [4] J. Bender et D. Koschier, « Divergence-free Smoothed Particle Hydrodynamics, » dans *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, série SCA '15, pp. 147–155. New York, NY, USA : Association for Computing Machinery (ACM), août 2015. doi:[10.1145/2786784.2786796](https://doi.org/10.1145/2786784.2786796)
- [5] J. Bender et D. Koschier, « Divergence-Free SPH for Incompressible and Viscous Fluids, » *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 3, pp. 1193–1206, mars 2017. doi:[10.1109/TVCG.2016.2578335](https://doi.org/10.1109/TVCG.2016.2578335)
- [6] J. Bender, T. Kugelstadt, M. Weiler, et D. Koschier, « Volume Maps : An Implicit Boundary Representation for SPH, » dans *Motion, Interaction and Games*, série MIG '19. New York, NY, USA : Association for Computing Machinery (ACM), octobre 2019. doi:[10.1145/3359566.3360077](https://doi.org/10.1145/3359566.3360077)

BIBLIOGRAPHIE

- [7] E. H. Brandt, « Levitation in Physics, » *Science*, vol. 243, no. 4889, pp. 349–355, janvier 1989. doi:[10.1126/science.243.4889.349](https://doi.org/10.1126/science.243.4889.349)
- [8] R. Bridson, « Fast Poisson Disk Sampling in Arbitrary Dimensions, » dans *ACM SIGGRAPH 2007 Sketches*, série SIGGRAPH '07, vol. 10, no. 1, p. 22. New York, NY, USA : Association for Computing Machinery (ACM), août 2007. doi:[10.1145/1278780.1278807](https://doi.org/10.1145/1278780.1278807)
- [9] R. Bridson, « Fluid Simulation for Computer Graphics, Second Edition. » New York, NY, USA : AK Peters/CRC Press, septembre 2015. doi:[10.1201/9781315266008](https://doi.org/10.1201/9781315266008)
- [10] L. Brookshaw, « A Method of Calculating Radiative Heat Diffusion in Particle Simulations, » *Publications of the Astronomical Society of Australia*, vol. 6, no. 2, pp. 207–210, 1985. doi:[10.1017/S1323358000018117](https://doi.org/10.1017/S1323358000018117)
- [11] M. Becker et M. Teschner, « Weakly compressible SPH for free surface flows, » dans *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, série SCA '07, pp. 209–217. Goslar, DEU : Eurographics Association, août 2007.
- [12] R. Courant, K. Friedrichs, et H. Lewy, « On the Partial Difference Equations of Mathematical Physics, » *IBM Journal of Research and Development*, vol. 11, no. 2, pp. 215–234, mars 1967. doi:[10.1147/rd.112.0215](https://doi.org/10.1147/rd.112.0215)
- [13] X. Chermain, « Simulation d'effets aérodynamiques et hydrodynamiques basée sur une méthode lagrangienne, » Mémoire de maîtrise, Département d'informatique, Université de Sherbrooke, QC, Canada, 05 février 2016.
Récupéré de <http://hdl.handle.net/11143/8391>
- [14] S. J. Cummins et M. Rudman, « An SPH projection method, » *Journal of computational physics*, vol. 152, no. 2, pp. 584–607, juillet 1999. doi:[10.1006/jcph.1999.6246](https://doi.org/10.1006/jcph.1999.6246)
- [15] M. Deserno, « How to generate equidistributed points on the surface of a sphere, » *If Polymerforschung (Ed.)*, vol. 99, no. 2, pp. 1–es, septembre 2004.

BIBLIOGRAPHIE

- [16] V. A. Del Grosso, « Sound speed in pure water and sea water, » *The Journal of the Acoustical Society of America*, vol. 47, no. 3B, pp. 947–949, mars 1970. doi:[10.1121/1.1911987](https://doi.org/10.1121/1.1911987)
- [17] M. Desbrun et M.-P. Gascuel, « Smoothed Particles : A new paradigm for animating highly deformable bodies, » dans *Computer Animation and Simulation '96*, R. Boulic et G. Hégron, éditeurs, pp. 61–76. Vienna : Springer, 1996. doi:[10.1007/978-3-7091-7486-9_5](https://doi.org/10.1007/978-3-7091-7486-9_5)
- [18] P. Eastwell, « Bernoulli? Perhaps, but What About Viscosity?. » *Science Education Review*, vol. 6, no. 1, pp. 1–13, 2007.
- [19] S. Elcott, Y. Tong, E. Kanso, P. Schröder, et M. Desbrun, « Stable, Circulation-Preserving, Simplicial Fluids, » *ACM Transactions on Graphics (TOG)*, vol. 26, no. 1, pp. 4–es, jan 2007. doi:[10.1145/1189762.1189766](https://doi.org/10.1145/1189762.1189766)
- [20] N. Foster et D. Metaxas, « Realistic Animation of Liquids, » *Graphical models and image processing*, vol. 58, no. 5, pp. 471–483, septembre 1996. doi:[10.1006/gmip.1996.0039](https://doi.org/10.1006/gmip.1996.0039)
- [21] D. A. Fulk, « A numerical analysis of smoothed particle hydrodynamics, » Thèse de doctorat, Air Force Institute of Technology, Department of the Air Force, Air University, Wright-Patterson Air Force Base, Ohio, septembre 1994.
- [22] R. A. Gingold et J. J. Monaghan, « Smoothed particle hydrodynamics : theory and application to non-spherical stars, » *Monthly Notices of the Royal Astronomical Society*, vol. 181, no. 3, pp. 375–389, novembre 1977. doi:[10.1093/mnras/181.3.375](https://doi.org/10.1093/mnras/181.3.375)
- [23] D. House et J. C. Keyser, « Foundations of Physically Based Modeling and Animation. » New York, NY, USA : AK Peters/CRC Press, décembre 2016. doi:[10.1201/9781315373140](https://doi.org/10.1201/9781315373140)
- [24] F. H. Harlow et J. E. Welch, « Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, » *The physics of fluids*, vol. 8, no. 12, pp. 2182–2189, décembre 1965.

BIBLIOGRAPHIE

- [25] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, et M. Teschner, « Implicit incompressible SPH, » *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 426–435, mars 2014. doi:[10.1109/TVCG.2013.105](https://doi.org/10.1109/TVCG.2013.105)
- [26] D. Koschier, J. Bender, B. Solenthaler, et M. Teschner, « Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids, » dans *Eurographics 2019 - Tutorials*, W. Jakob et E. Puppo, éditeurs. Eurographics Association, 2019. doi:[10.2312/egt.20191035](https://doi.org/10.2312/egt.20191035)
- [27] J. N. Kutz, « Deep learning in fluid dynamics, » *Journal of Fluid Mechanics*, vol. 814, pp. 1–4, mars 2017. doi:[10.1017/jfm.2016.803](https://doi.org/10.1017/jfm.2016.803)
- [28] M. B. Liu et G. R. Liu, « Smoothed Particle Hydrodynamics (SPH) : an Overview and Recent Developments, » *Archives of Computational Methods in Engineering*, vol. 17, no. 1, pp. 25–76, février 2010. doi:[10.1007/s11831-010-9040-7](https://doi.org/10.1007/s11831-010-9040-7)
- [29] L. B. Lucy, « A numerical approach to the testing of the fission hypothesis, » *The astronomical journal*, vol. 82, pp. 1013–1024, décembre 1977. doi:[10.1086/112164](https://doi.org/10.1086/112164)
- [30] M. Müller, D. Charypar, et M. Gross, « Particle-Based Fluid Simulation for Interactive Applications, » dans *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, série SCA’03, pp. 154—159. Goslar, DEU : Eurographics Association, juillet 2003.
- [31] M. Macklin et M. Müller, « Position Based Fluids, » *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, juillet 2013. doi:[10.1145/2461912.2461984](https://doi.org/10.1145/2461912.2461984)
- [32] J. J. Monaghan, « SMOOTHED PARTICLE HYDRODYNAMICS, » *Annual review of astronomy and astrophysics*, vol. 30, no. 1, pp. 543–574, septembre 1992. doi:[10.1146/annurev.aa.30.090192.002551](https://doi.org/10.1146/annurev.aa.30.090192.002551)

BIBLIOGRAPHIE

- [33] J. J. Monaghan, « Simulating Free Surface Flows with SPH, » *Journal of computational physics*, vol. 110, no. 2, pp. 399–406, février 1994. doi:[10.1006/jcph.1994.1034](https://doi.org/10.1006/jcph.1994.1034)
- [34] J. Monaghan, « SPH without a Tensile Instability, » *Journal of computational physics*, vol. 159, no. 2, pp. 290–311, avril 2000. doi:[10.1006/jcph.2000.6439](https://doi.org/10.1006/jcph.2000.6439)
- [35] J. J. Monaghan, « Smoothed particle hydrodynamics, » *Reports on progress in physics*, vol. 68, no. 8, pp. 1703–1759, juillet 2005. doi:[10.1088/0034-4885/68/8/r01](https://doi.org/10.1088/0034-4885/68/8/r01)
- [36] P. Ma, Y. Tian, Z. Pan, B. Ren, et D. Manocha, « Fluid Directed Rigid Body Control using Deep Reinforcement Learning, » *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–11, juillet 2018. doi:[10.1145/3197517.3201334](https://doi.org/10.1145/3197517.3201334)
- [37] D. J. Price, « Smoothed particle hydrodynamics and magneto-hydrodynamics, » *Journal of Computational Physics, Special Issue : Computational Plasma Physics*, vol. 231, no. 3, pp. 759–794, février 2012. doi:[10.1016/j.jcp.2010.12.011](https://doi.org/10.1016/j.jcp.2010.12.011)
- [38] H. Schechter et R. Bridson, « Ghost SPH for Animating Water, » *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–8, juillet 2012. doi:[10.1145/2185520.2185557](https://doi.org/10.1145/2185520.2185557)
- [39] Bureau international des poids et mesures, « Le Système international d’unités (SI), 9e édition. » Pavillon de Breteuil, F-92312 Sèvres Cedex France : BIPM, 20 mai 2019.
- [40] A. Staniforth et J. Côté, « Semi-Lagrangian Integration Schemes for Atmospheric Models—A Review, » *Monthly Weather Review*, vol. 119, no. 9, pp. 2206–2223, septembre 1991. doi :[10.1175/1520-0493\(1991\)119<2206:SLISFA>2.0.CO;2](https://doi.org/10.1175/1520-0493(1991)119<2206:SLISFA>2.0.CO;2)
- [41] S. Sato, Y. Dobashi, T. Kim, et T. Nishita, « Example-Based Turbulence Style Transfer, » *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–9, juillet 2018. doi:[10.1145/3197517.3201398](https://doi.org/10.1145/3197517.3201398)

BIBLIOGRAPHIE

- [42] S. Shao et E. Y. Lo, « Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface, » *Advances in water resources*, vol. 26, no. 7, pp. 787–800, juillet 2003. doi:[10.1016/S0309-1708\(03\)00030-7](https://doi.org/10.1016/S0309-1708(03)00030-7)
- [43] A. Stomakhin, A. Moffat, et G. Boyle, « A Practical Guide to Thin Film and Drips Simulation, » dans *ACM SIGGRAPH 2019 Talks*, série SIGGRAPH '19. New York, NY, USA : Association for Computing Machinery (ACM), juillet 2019. doi:[10.1145/3306307.3328141](https://doi.org/10.1145/3306307.3328141)
- [44] B. Solenthaler et R. Pajarola, « Predictive-Corrective Incompressible SPH, » dans *ACM SIGGRAPH 2009 Papers*, série SIGGRAPH '09, vol. 28, no. 3, pp. 1–6. New York, NY, USA : Association for Computing Machinery (ACM), juillet 2009. doi:[10.1145/1576246.1531346](https://doi.org/10.1145/1576246.1531346)
- [45] J. Stam, « Stable Fluids, » dans *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, série SIGGRAPH '99, pp. 121–128, juillet 1999. doi:[10.1145/311535.311548](https://doi.org/10.1145/311535.311548)
- [46] T. Takahashi, Y. Dobashi, I. Fujishiro, T. Nishita, et M. C. Lin, « Implicit Formulation for SPH-based Viscous Fluids, » dans *Computer Graphics Forum*, vol. 34, no. 2, pp. 493–502. Wiley Online Library, juin 2015. doi:[10.1111/cgf.12578](https://doi.org/10.1111/cgf.12578)
- [47] D. Takeshita, S. Ota, M. Tamura, T. Fujimoto, K. Muraoka, et N. Chiba, « Particle-Based Visual Simulation of Explosive Flames, » dans *11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings. Canmore, AB, Canada*, pp. 482–486, octobre 2003. doi:[10.1109/PCCGA.2003.1238299](https://doi.org/10.1109/PCCGA.2003.1238299)
- [48] D. J. Tritton, « Physical Fluid Dynamics. » New York, NY, USA : Van Nostrand Reinhold Company, 1977, chapitre 22.7 Coanda effect, pp. 284–286. doi:[10.1007/978-94-009-9992-3](https://doi.org/10.1007/978-94-009-9992-3)
- [49] D. Violeau et B. D. Rogers, « Smoothed particle hydrodynamics (SPH) for free-surface flows : past, present and future, » *Journal of Hydraulic Research*, vol. 54, no. 1, pp. 1–26, janvier 2016. doi:[10.1080/00221686.2015.1119209](https://doi.org/10.1080/00221686.2015.1119209)

BIBLIOGRAPHIE

- [50] Y. Xie, E. Franz, M. Chu, et N. Thuerey, « TempoGAN : A Temporally Coherent, Volumetric GAN for Super-Resolution Fluid Flow, » *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, août 2018. doi:[10.1145/3197517.3201304](https://doi.org/10.1145/3197517.3201304)

Annexe A

Rappels mathématiques

Les bases mathématiques nécessaires à la compréhension du mémoire, notamment tous les opérateurs usuels utilisés tels que le gradient dans la [section A.1](#), la divergence dans la [section A.2](#) et le laplacien dans la [section A.3](#), sont rappelées dans cette annexe. Puis les formules utilisées dans la méthode des différences finies sont rappelées dans la [section A.4](#).

A.1 Gradient

Nous pouvons appliquer l'opérateur gradient pour obtenir un champ vectoriel $\nabla\phi(x)$ à partir d'un champ scalaire $\phi(x)$ [[23](#), pp.269-270]. Le symbole ∇ (prononcé « *nabla* ») est défini comme

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix}.$$

En considérant l'espace défini en 3D par les vecteurs \vec{x} , \vec{y} , \vec{z} . Un gradient opérant sur un champ scalaire $\phi(x, y, z)$ forme un champ vectoriel constitué des dérivées partielles, tel que

$$\nabla\phi = \begin{bmatrix} \frac{\partial\phi}{\partial x} \\ \frac{\partial\phi}{\partial y} \\ \frac{\partial\phi}{\partial z} \end{bmatrix}.$$

A.2. DIVERGENCE

Puisque $\frac{\partial \phi}{\partial x}$, $\frac{\partial \phi}{\partial y}$ et $\frac{\partial \phi}{\partial z}$ indiquent le changement spatial en un point local du champ scalaire, nous pouvons dire que le gradient détermine comment ce champ varie dans l'espace autour de ce point [23, pp.269-270].

A.2 Divergence

La divergence s'applique sur un champ vectoriel pour produire un champ scalaire [23, pp.270-271]. Considérant un vecteur $\vec{u} = [u_x, u_y, u_z]$, l'opérateur de divergence appliqué à \vec{u} est donné par le produit scalaire entre ∇ et \vec{u} tel que

$$\nabla \cdot \vec{u} = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right] \cdot \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}.$$

A.3 Laplacien

L'opérateur laplacien s'applique indifféremment sur un champ vectoriel ou scalaire et retourne un champ du même type [23, pp.272-273], soit

$$\nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

Appliqué à un champ scalaire $\phi(x, y, z, t)$, nous obtenons

$$\nabla^2 \phi = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}.$$

Appliqué à un champ vectoriel $\vec{u}(x, y, z, t)$, nous obtenons

$$\nabla^2 \vec{u} = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \\ \frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2} \\ \frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2} \end{bmatrix}.$$

Le laplacien est connu pour fournir un comportement de diffusion des termes sur lequel il s'applique [23, pp.272-273], c'est-à-dire que celui-ci opère essentiellement un

A.4. DIFFÉRENCES FINIES

lissage des valeurs vers une moyenne en considérant toutes les grandeurs voisines.

A.4 Différences finies

Soit un pas Δx , la définition mathématique de la dérivée en x appliquée à une fonction $f(x)$ quelconque est donnée par

$$\frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}. \quad (\text{A.1})$$

La méthode des différences finies est une méthode pour approximer les résultats des équations différentielles [23, pp.293-295] en réécrivant l'équation (A.1) de trois manières :

(a) **Opérateur de différence avant :**

$$\frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}.$$

(b) **Opérateur de différence arrière :**

$$\frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x) - f(x - \Delta x)}{\Delta x}.$$

(c) **Opérateur de différence centrée :**

$$\frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \frac{\Delta x}{2}) - f(x - \frac{\Delta x}{2})}{\Delta x}.$$

À noter qu'en simulation physique sur ordinateur, la fonction $f(x)$ ne peut pas être modélisée comme une fonction continue sur son ensemble de définition. Celle-ci sera échantillonnée, donc discrétisée, pour un ensemble de valeurs de x séparées par un pas Δx petit, mais pas nécessairement proche de zéro, dépendant de la balance choisie entre la précision souhaitée des résultats contre le temps de simulation et la consommation mémoire utilisée. Ceci aura alors pour conséquence que la méthode des différences finies fournira une approximation de la valeur de $\frac{df(x)}{dx}$ dont la précision dépendra de Δx .

Annexe B

Liens et détails des vidéos

Ici sont présentées les explications des vidéos présentant nos résultats, ainsi que les divers liens introduits dans notre mémoire.

B.1 SPLisHSPlasH

Nous nous sommes inspirés du moteur *SPLisHSPlasH* implémenté par J. Bender et al. en tant que référence pour l'implémentation de [HPL Faiblement Compressible \(HPLFC\)](#), [HPL Incompressible Predictif-Correctif \(HPLIPC\)](#), [HPL Implicitement Incompressible \(HPLII\)](#) et [HPL Sans Divergence \(HPLSD\)](#). Ce moteur fut surtout utilisé pour valider nos résultats dans des scènes plus standard comme celle de la cassure de barrage.

On peut retrouver le code sur [Github](#).

B.2 Lien vers notre application *Storm*

Storm est notre moteur de simulation écrit intégralement en C/C++. Pour nous aider, nous avons développé quelques outils annexes pour faciliter son utilisation dans le cadre de nos recherches. Notamment, nous avons intégré un interpréteur lua, ainsi qu'un outil séparé afin d'envoyer nos commandes lua pendant l'exécution pour contrôler le simulateur et aider à l'interprétation des résultats. Un visualiseur de

B.3. VIDÉOS

journaux d'évènements (*log*) a aussi été développé afin de localiser plus aisément les problèmes dans la simulation.

Le lien du moteur est sur *Github*¹.

B.3 Vidéos

B.3.1 Bande-annonce

La bande-annonce est une vidéo² montrant brièvement les moments clés de plusieurs de nos simulations.

B.3.2 Résultats d'optimisation de la recherche de voisinage avec grille spatiale 3D

Deux vidéos (avant³ et après⁴ optimisation) ont été réalisées pour illustrer nos résultats.

La chute d'un parallélépipède rectangle d'eau avec 1728 particules de fluide tombant dans un domaine de 9960 particules de solides a été simulée en utilisant la méthode [HPL Faiblement Compressible \(HPLFC\)](#) afin que le calcul d'un pas de temps de simulation soit toujours comparable entre les exécutions.

La simulation sans optimisation calcule le voisinage de chacune des particules de fluide au moment de calculer la densité, puis au moment de calculer les forces de viscosité et de pression, en itérant sur toutes les particules de fluide et de solide présentes dans le domaine pour déterminer si celle-ci est une particule voisine.

La simulation avec optimisation calcule le voisinage de chacune des particules de fluide en utilisant une grille spatiale 3D décrite dans la [sous-section 2.2.2](#). Le calcul est effectué une seule fois au début de chaque pas de temps et la liste des particules voisines est mise en cache pour être réutilisée à chaque recherche de voisinage exécutée au même temps de simulation.

1. <https://github.com/SunlayGGX/Storm>
2. <https://youtu.be/gQmAg7tY-ig>
3. https://youtu.be/vJfeoJ1_TrS
4. <https://youtu.be/m42ezr4IrYk>

B.3. VIDÉOS

Tableau B.1 – Gain de performance obtenue avec la recherche de voisinage utilisant une grille spatiale 3D.

	nombre de particule	Δt [s]	temps de calcul pour 1s de simulation [s]	ratio	réduction du temps de calcul
Avant	12k	0,001	27	100%	-
Après	12k	0,001	1	3,7%	96,3%

Selon les résultats présentés dans la [tableau B.1](#) simulant de l'eau avec la méthode HPLFC (voir la [sous-section 2.2.5.3](#)), il a fallu vingt-sept secondes de calcul pour simuler une seconde sans l'utilisation de la grille spatiale 3D, contre une seconde de calcul pour une seconde de simulation avec la grille spatiale 3D. Sachant que le pas de temps est fixé à $\Delta t = 0,001$ s entre les simulations, il apparaît une réduction du temps de calcul de 96% lorsque une grille spatiale 3D est utilisée.

B.3.3 Simulation d'une scène en configuration de cassure de barrage

La vidéo ⁵ montre la simulation d'une scène en configuration de cassure de barrage prise sous deux angles de vue différents (une vue de côté, puis une vue en plongée à la verticale depuis le dessus du domaine de 2 m de longueur, 1 m de hauteur et 1 m de profondeur).

Le fluide est apparenté à de l'eau (sa densité est $\rho = 1000 \text{ kg}^{-3}$ et sa viscosité dynamique est $\eta = 0,001 \text{ Pa.s}$). Le parallélépipède rectangle jaune pèse $m = 125 \text{ kg}$ pour un volume $V = 0,01125 \text{ m}^3$ (soit un $\rho = 11111 \text{ kg.m}^{-3}$) tandis que les autres parallélépipèdes rectangles pèsent chacun $m = 8 \text{ kg}$ pour $V = 0,0045 \text{ m}^3$ (soit $\rho = 1778 \text{ kg.m}^{-3}$). La balle jaune pèse $m = 2,5 \text{ kg}$ pour $V = 0,0042 \text{ m}^3$ (soit $\rho = 595 \text{ kg.m}^{-3}$).

Cette simulation illustre et valide l'implémentation du couplage bi-directionnel fluide↔solide.

5. <https://youtu.be/6529Ng45lfg>

B.3. VIDÉOS

B.3.4 Test de flottaison

L'effet de la gravité sur la simulation est démontrée dans la vidéo ⁶ de nos résultats montrant deux simulations de trois balles avec des densités différentes, immergées dans un fluide apparenté à de l'eau (sa densité est $\rho = 1000 \text{ kg}^{-3}$ et sa viscosité dynamique est $\eta = 0,001 \text{ Pa.s}$). La balle bleue a une densité $\rho = 500,0 \text{ kg.m}^{-3} < \rho_0$, la balle verte a une densité $\rho = 1000,0 \text{ kg.m}^{-3} = \rho_0$ et la balle rouge a une densité $\rho = 1500,0 \text{ kg.m}^{-3} > \rho_0$. L'effet de la gravité est enlevé sur le domaine de gauche tandis qu'il reste présent dans le domaine de droite. Celle-ci reste appliquée sur les solides, peu importe le domaine de droite ou de gauche. La coloration des particules du fluide est réalisée en fonction de la densité. Une coloration s'approchant vers le bleu/violet traduit une densité du fluide $\rho_i < \rho_0$ (une dilatation), à l'inverse, plus la couleur est rouge, plus $\rho_i > \rho_0$ (une compression). Une coloration rose signifie que $\rho_i = \rho_0$.

En théorie, nous nous attendons à ce que la balle bleue, dont la densité est plus faible que celle du fluide, flotte (remonte en haut du domaine), que la balle verte, dont la densité égale à celle du fluide, reste en suspension à la même hauteur, et enfin, que la balle rouge, dont la densité est supérieure que celle du fluide, coule (descend dans le fond du domaine).

Nous observons pendant la simulation :

- La gravité est responsable de comprimer le fluide vers le fond du domaine et empêche les dilatations locales. Sans la gravité (domaine de gauche), le domaine se remplit entièrement, mais des zones de coloration bleues/violettes apparaissent, traduisant une dilatation locale du fluide. Avec la gravité (domaine de droite), le fluide ne présente aucune dilatation, mais du vide se forme en haut du domaine.
- La gravité est nécessaire pour faire flotter les balles dans le domaine. Sans gravité, elles coulent toutes les trois qu'importe leur densité, car la pression faisant remonter les balles en les repoussant vers le haut du domaine est mal calculée.

6. <https://youtu.be/jIqduxLrLCU>

B.3. VIDÉOS

B.3.5 Vidéo de la réalité montrant la lévitation d'une balle de ping-pong sous un sèche-cheveux

Deux vidéos de la réalité, la première utilisant un sèche-cheveux positionné à la verticale⁷ et la seconde où le sèche-cheveux est incliné à sept degrés⁸, ont été réalisées pour valider que les simulations obtenues respectent le comportement attendu.

B.3.6 Simulation de la lévitation d'une balle sous un souffleur positionné à la verticale

La simulation⁹ utilise la méthode [HPL Sans Divergence \(HPLSD\)](#) ainsi que la force de pénalité basée sur l'effet de Coandă pour maintenir une balle en lévitation au-dessus d'un souffleur positionné à la verticale. Le fluide a une densité de repos $\rho_0 = 250 \text{ kg.m}^{-3}$, sa viscosité dynamique est $\eta = 0,001 \text{ Pa.s}$ et le coefficient de la force de pénalité basée sur l'effet de Coandă est fixé à $C_j = 0,00001$.

B.3.7 Simulation de la lévitation d'une balle sous un souffleur incliné à sept degrés sans effet Coandă

La simulation¹⁰ utilise la méthode [HPL Sans Divergence \(HPLSD\)](#) sans la force de pénalité basée sur l'effet de Coandă. Le fluide a une densité de repos $\rho_0 = 250 \text{ kg.m}^{-3}$ et sa viscosité dynamique $\eta = 0,001 \text{ Pa.s}$. La balle lâchée au-dessus d'un souffleur incliné avec un angle de sept degrés par rapport à la verticale ne parvient pas à se maintenir en lévitation comme expliqué dans la [sous-section 3.6.3](#).

7. <https://youtu.be/vKRyXPgQbKY>

8. <https://youtu.be/JYosyiaJQiw>

9. <https://youtu.be/uaPb8rvhJTU>

10. <https://youtu.be/HN13dJjXWwA>

B.3. VIDÉOS

B.3.8 Simulation de la lévitation d'une balle sous un souffleur incliné à sept degrés avec effet Coandă

La simulation ¹¹ utilise la méthode [HPL Sans Divergence \(HPLSD\)](#) avec la force de pénalité basée sur l'effet de Coandă. Le fluide a une densité de repos $\rho_0 = 250 \text{ kg.m}^{-3}$, sa viscosité dynamique est $\eta = 0,001 \text{ Pa.s}$ et le coefficient de la force de pénalité basée sur l'effet de Coandă est fixé à $C_j = 0,00001$. La balle lâchée au-dessus d'un souffleur incliné avec un angle de sept degrés par rapport à la verticale parvient à se maintenir en lévitation comme expliqué dans la [sous-section 3.6.3](#). Son comportement, tant au niveau des oscillations gauche-droite et avant-arrière que ses rotations autour des axes perpendiculaires aux flux d'air, est cohérent avec le comportement observé dans la réalité (voir la [sous-section B.3.5](#))

11. <https://youtu.be/u5fyrQVooVA>