UNIVERSITÉ PARIS-EST

École Doctorale SIE

Thèse

Présentée pour l'obtention du grade de DOCTEUR
DE L'UNIVERSITE PARIS-EST
par

Agnès Leroy

Un nouveau modèle SPH incompressible : vers l'application à des cas industriels

---

Spécialité : Mécanique des Fluides

Soutenue le 17 Novembre 2014 devant un jury composé de :

Rapporteur	**Prof. David Le Touzé**	(Ecole Centrale de Nantes)
Rapporteur	**Prof. Thomas Rung**	(Technical University of Hamburg-Harburg)
Examinateur	**Prof. Mickhaël Balabane**	(Université Paris XIII)
Examinateur	**Dr. Jean-Frédéric Gerbeau**	(INRIA)
Examinateur	**Dr. Benedict Rogers**	(The University of Manchester)
Directeur de thèse	**Dr. Damien Violeau**	(EDF R&D & Université Paris-Est)
Invité	**M. Martin Ferrand**	(EDF R&D)







Thèse effectuée au sein du **Laboratoire d'Hydraulique Saint-Venant**
de l'Université Paris-Est,

6, quai Watier
BP 49
78401 Chatou cedex
France

Financements : ANR (bourse CIFRE # 2011-0264) et EDF R&D

Résumé

Cette thèse a pour objet le développement d'un modèle numérique de simulation des fluides fondé sur la méthode *Smoothed Particle Hydrodynamics* (SPH). SPH est une méthode de simulation numérique sans maillage présentant un certain nombre d'avantages par rapport aux méthodes Eulériennes. Elle permet notamment de modéliser des écoulements à surface libre ou interfaces fortement déformées. Ce travail s'adresse principalement à quatre problématiques liées aux fondements de la méthode SPH : l'imposition des conditions aux limites, la prédiction précise des champs de pression, l'implémentation d'un modèle thermique et la réduction des temps de calcul. L'objectif est de modéliser des écoulements industriels complexes par la méthode SPH, en complément de ce qui peut se faire avec des méthodes à maillage. Typiquement, les problèmes visés sont des écoulements 3-D à surface libre ou confinés, pouvant interagir avec des structures mobiles et/ou transporter des scalaires, notamment des scalaires actifs (*e.g.* température). Dans ce but, on propose ici un modèle SPH incompressible (ISPH) fondé sur une représentation semi-analytique des conditions aux limites. La technique des conditions aux limites semi-analytiques permet d'imposer des conditions sur la pression de manière précise et physique, contrairement à ce qui se fait avec des conditions aux limites classiques en SPH. Un modèle $k - \epsilon$ a été incorporé à ce nouveau modèle ISPH, à partir des travaux de Ferrand *et al.* (2013). Un modèle de flottabilité a également été ajouté, reposant sur l'approximation de Boussinesq. Les interactions entre flottabilité et turbulence sont prises en compte. Enfin, on établit une formulation pour les frontières ouvertes dans le nouveau modèle.
La validation en 2-D a été réalisée sur un ensemble de cas-tests permettant d'estimer les capacités de prédiction du nouveau modèle en ce qui concerne les écoulements isothermes et non-isothermes, laminaires ou turbulents. Des cas confinés sont présentés, ainsi que des écoulements à surface libre (l'un d'eux incluant un corps solide mobile dans l'écoulement). La formulation pour les frontières ouvertes a été testée sur un canal de Poiseuille plan laminaire et sur deux cas de propagation d'une onde solitaire. Des comparaisons sont présentées avec des méthodes à maillage, ainsi qu'avec un modèle SPH quasi-incompressible (WCSPH) avec le même type de conditions aux limites. Les résultats montrent que le modèle permet de représenter des écoulements dans des domaines à géométrie complexe, tout en améliorant la prédiction des champs de pression par rapport à la méthode WCSPH. L'extension du modèle en trois dimensions a été réalisée dans un code massivement parallèle fonctionnant sur carte graphique (GPU). Deux cas de validation en 3-D sont proposés, ainsi que des résultats sur un cas simple d'application en 3-D.

Mots-clé : écoulements incompressibles, simulation numérique, SPH, turbulence, conditions aux limites, température, flottabilité.

A New Incompressible SPH Model: Towards Industrial Applications

Abstract

In this work a numerical model for fluid flow simulation was developed, based on the Smoothed Particle Hydrodynamics (SPH) method. SPH is a meshless Lagrangian Computational Fluid Dynamics (CFD) method that offers some advantages compared to mesh-based Eulerian methods. In particular, it is able to model flows presenting highly distorted free-surfaces or interfaces. This work tackles four issues concerning the SPH method: the imposition of boundary conditions, the accuracy of the pressure prediction, the modelling of buoyancy effects and the reduction of computational time. The aim is to model complex industrial flows with the SPH method, as a complement of what can be done with mesh-based methods. Typically, the targetted problems are 3-D free-surface or confined flows that may interact with moving solids and/or transport scalars, in particular active scalars (*e.g.* temperature). To achieve this goal, a new incompressible SPH (ISPH) model is proposed, based on semi-analytical boundary conditions. This technique for the representation of boundary conditions in SPH makes it possible to accurately prescribe consistent pressure boundary conditions, contrary to what is done with classical boundary conditions in SPH. A $k-\epsilon$ turbulence closure is included in the new ISPH model. A buoyancy model was also added, based on the Boussinesq approximation. The interactions between buoyancy and turbulence are modelled. Finally, a formulation for open boundary conditions is proposed in this framework.
The 2-D validation was performed on a set of test-cases that made it possible to assess the prediction capabilities of the new model regarding isothermal and non-isothermal flows, in laminar or turbulent regime. Confined cases are presented, as well as free-surface flows (one of them including a moving rigid body in the flow). The open boundary formulation was tested on a laminar plane Poiseuille flow and on two cases of propagation of a solitary wave. Comparisons with mesh-based methods are provided with, as well as comparisons with a weakly-compressible SPH (WCSPH) model using the same kind of boundary conditions. The results show that the model is able to represent flows in complex boundary geometries, while improving the pressure prediction compared to the WCSPH method. The extension of the model to 3-D was done in a massively parallel code running on a Graphic Processing Unit (GPU). Two validation cases in 3-D are presented, as well as preliminary results on a simple 3-D application case.

Keywords: incompressible flows, numerical simulation, SPH, turbulence, boundary conditions, temperature, buoyancy.

Remerciements

Je tiens à remercier en premier lieu mon directeur de thèse, Damien Violeau, pour l'encadrement qu'il m'a apporté tout au long de ces trois ans. Il me semble important de souligner, non seulement la structuration très claire qu'il a donnée au projet, mais aussi l'accompagnement scientifique rigoureux qu'il m'a apporté au quotidien. Mes remerciements s'adressent également à Martin Ferrand, pour tout le temps qu'il m'a consacré, que ce soit pour répondre à mes questions, me donner des pistes de travail ou relire ma production scientifique. Je tiens aussi à remercier Arno Mayrhofer qui m'a entre autres aidée à progresser très rapidement au début de la thèse, et Christophe Kassiotis, en particulier pour avoir fait tourner les simulations OpenFOAM et pour m'avoir épaulée quand j'ai commencé à travailler sur Sphynx, le code GPU. Merci aussi à François-Xavier Morel pour sa contribution sur Sphynx et toutes nos discussions. Votre aide à tous m'a vraiment été précieuse au cours de ces trois années et c'est toujours un plaisir de travailler avec vous. Merci à Thomas Rung, David Le Touzé et Ben Rogers pour leurs remarques et corrections concernant le mémoire, et à Sébastien Boyaval pour sa relecture attentive du Chapitre 1. Mes remerciements s'adressent également à Mickhaël Balabane pour avoir accepté de présider mon jury de thèse et à Jean-Frédéric Gerbeau pour avoir accepté de faire partie du jury. Merci aussi à Alexandre Ern et Frédéric Dias qui avaient initialement accepté de faire partie du jury, avant que la date de soutenance ne soit modifiée. J'adresse aussi ces remerciements à EDF R&D et à l'ANRT qui ont financé cette thèse dans le cadre de la convention CIFRE #2011-0264, et au Laboratoire d'Hydraulique Saint-Venant au sein duquel j'ai effectué cette thèse.

Bien d'autres personnes ont apporté leur contribution à ce projet: merci notamment à Antoine Joly, Dominique Laurence, Alexis Hérault, Sofiane Benamadouche. Merci également à Sara Pavan pour tout le soutien qu'elle m'a apporté au long de ces trois années où nous avons partagé un bureau et pour les séances d'escalade à Fontainebleau (avec Jacques d'ailleurs, merci pour le choix de la corde...). De manière plus générale merci aux membres du Laboratoire d'Hydraulique Saint-Venant et du LNHE, en partiulier des groupes P74 et P73.

Enfin, j'adresse ces remerciements à ma famille, en particulier mes parents, mon frère et mes deux soeurs, ainsi qu'à mes amis du rdvbc et de prépa pour le soutien qu'ils m'ont apporté lorsque j'en ai eu besoin et pour tous les bons moments passés ensemble. J'ai renoncé à écrire tous vos noms ici, j'espère que vous ne m'en tiendrez pas rigueur, mais je citerai au moins celui de Julie Chrétien pour le live tweet fantastique qu'elle a fait pendant la soutenance : merci.

Contents

Introduction **1**

1 Governing equations and modelling choices **5**

1.1 Navier–Stokes equations for incompressible flows 6

1.1.1 Formulation 6

1.1.2 Projection methods 9

1.2 Turbulence modelling 15

1.2.1 Reynolds-Averaged Navier–Stokes models 16

1.2.2 Large Eddy Simulation 22

1.3 Buoyancy modelling 23

1.3.1 Diffusion equation on the temperature 23

1.3.2 Buoyancy effects in the momentum equation 26

1.4 System of equations to be solved and associated set of boundary conditions 27

2 Smoothed Particle Hydrodynamics: Literature review **31**

2.1 Introduction to SPH 32

2.2 Classical SPH interpolation and differential operators 34

2.2.1 SPH interpolation 34

2.2.2 First order differential operators in SPH 39

2.2.3 Second order differential operator in SPH 44

2.3 Modelling incompressible flows with SPH 46

2.3.1 Classical weakly-compressible approach 46

2.3.2 Truly incompressible SPH 55

2.4 Wall boundary conditions in SPH 63

2.4.1 Classical treatment of the wall boundary conditions 63

2.4.2 Unified semi-analytical wall boundary conditions 67
2.5 Imposition of free-surface boundaries in SPH 73
2.5.1 Pressure condition 73
2.5.2 Conditions on the velocity, the temperature, k and ϵ 74
2.6 Imposition of open boundaries in SPH 75
2.7 Turbulence modelling and treatment of the viscous term in SPH 77
2.7.1 Turbulence modelling: state-of-the-art in SPH 77
2.7.2 SPH k and ϵ equations 78
2.7.3 Treatment of the viscous term 78
2.8 Buoyancy modelling in SPH 79
2.9 Reduction of the computational times through parallel programming 79

3 A new incompressible SPH model 83
3.1 Preliminary considerations 84
3.2 Space-time discretisation of the governing equations 84
3.2.1 Time discretisation 85
3.2.2 Space discretisation 87
3.3 Wall boundary conditions 91
3.3.1 Wall boundary conditions on the velocity 92
3.3.2 Wall boundary conditions on k and ϵ 94
3.3.3 Wall boundary conditions on the temperature 96
3.3.4 Wall boundary conditions on the pressure 98
3.4 Free-surface conditions 99
3.5 Open boundaries 100
3.5.1 Particles creation/destruction 101
3.5.2 Imposition of the inflow/outflow boundary conditions 103
3.5.3 Inflow boundaries 104
3.5.4 Outflow boundaries 105
3.6 Solving the pressure Poisson equation 106
3.7 Numerical stability 108
3.8 Computation of the kernel renormalisation factor and its gradient 110
3.8.1 Computation of the kernel renormalisation factor and its gradient in 2-D . 110

3.8.2 Computation of the kernel renormalisation factor and its gradient in 3-D . 113
3.9 Parallelisation in a GPU framework 113

4 Validation on 2-D cases 115
4.1 Nomenclature 116
4.2 Validation on isothermal 2-D cases 117
4.2.1 Laminar flows 117
4.2.2 Turbulent flows 134
4.3 Validation on non-isothermal 2-D cases 142
4.3.1 Laminar flows 142
4.3.2 Turbulent flows 149

5 3-D cases: validation and preliminary application results 153
5.1 Laminar flow in a circular pipe with inflow/outflow boundaries 154
5.2 Dam-break over an obstacle 155
5.3 Connected pipes case 158

Conclusions 163

Appendix A Analytical computation of $\nabla\gamma_a$ in 3-D 167

Bibliography 185

List of Figures

Introduction 1

1 Governing equations and modelling choices 5

2 SPH: Literature review 31

2.1 SPH modelling of the fusion of two spiral galaxies presenting central black holes [89]. 32

2.2 Snapshot of an SPH simulation of a 2D shock at very high-speed between an aluminium circular projectile and a thin aluminium plate [17]. 33

2.3 Complete modelling of a 2-jets horizontal Pelton turbine (colours represent the velocity) [84]. 33

2.4 Plot of the non-normalised 5th order Wendland, 5th order B-spline and truncated Gaussian kernels (left) and their first derivatives (right) in 2-D. 38

2.5 An example of highly anisotropic and inhomogeneous particles arrangement on the Taylor-Green vortices case with a classical Chorin projection method adapted to SPH [76]. 57

2.6 Sketch of the main three techniques for wall boundary modelling in SPH. (a): Boundary forces; (b) Dummy particles; (c) Mirror particles. 65

2.7 Sketch of the kernel truncation. γ_a is the integral under w on the blue area. 66

2.8 Sketch of the continuous medium discretisation with the USAW boundary conditions technique. 68

2.9 values of the SPH divergence of the position (x, z) at all the particles of a square tank discretised with 40×40 particles, with $D_a^{\gamma,-}$ and $D_a^{\gamma,-,1}$. (x, z). Comparison with the analytical value $Div(x, z) = 2$. 73

2.10 Sketch of the buffer layer method for the imposition of open boundaries. 75

2.11 Sketch of the process of particles creation [60]. 76

2.12 3-D modelling of the Goulours dam spillway (EDF, south of France) with the SPARTACUS-3D code. About 1.1 million particles and 7.7×10^5 boundary elements were used [94]. 80

2.13 3-D modelling of a breaking wave with the GPUSPH code. Representation of the free-surface shape and of the three-dimensional large-scale coherent structures under a plunging wave. About 4.5 million particles were used [31]. 81

2.14 Examples of 3-D cases simulated with the Dual SPHysics code: modelling of a wave impacting a breakwater [3] (left) and an oil rig [26] (right). The latter simulation involved about one billion particles. 82

3 A new incompressible SPH model **83**

3.1 Sketch of the position of the fictitious point a' used for the imposition of Neumann wall boundary conditions on $\boldsymbol{u}$ and ϵ. 93

3.2 Open boundary technique based on the USAW boundary conditions: sketch of the notations for the computation of the fraction of segment area $\beta_{a,v}$ (represented in red) attributed to a vertex at the outlet [60]. (a) 2-D case; (b) 3-D case. The coefficient associated to the vertex v_0 is proportional to the red area so that the largest amount of mass goes to the closest vertex particle. 102

3.3 Notations for the calculation of $\tilde{\gamma}_{as}$. 112

4 Validation **115**

4.1 Lid-driven cavity case for $Re = 1000$: qualitative comparison of the results. . . . 118

4.2 Lid-driven cavity for $Re = 100$. Dimensionless velocity profiles. 119

4.3 Lid-driven cavity for $Re = 400$. Dimensionless velocity and pressure profiles. . . 120

4.4 Lid-driven cavity for $Re = 1000$. Dimensionless velocity profiles. 121

4.5 Lid-driven cavity for $Re = 1000$. Convergence study. 121

4.6 Lid-driven cavity for $Re = 1000$. Dimensionless pressure profiles. 122

4.7 Lid-driven cavity for $Re = 1000$. Dimensionless velocity and pressure profiles with 500×500 particles. 123

4.8 Infinite array of cylinders in a channel: sketch of the geometry with $l^+ = 6$ [30, 78]. 123

4.9 Infinite array of cylinders in a channel: dimensionless drag force as a function of the inter-cylinder distance. 124

4.10 Infinite array of cylinders in a channel: velocity profiles for the case $l^+ = 6$. . . . 126

4.11 Infinite array of cylinders in a channel ($l^+ = 6$). Evolution of the drag coefficient and of the relative error as functions of the discretisation. 126

4.12 Laminar plane Poiseuille channel flow with inflow/outflow conditions. Dimensionless velocity profile. 127

4.13 Dam-break over a wedge. Qualitative representation of the free-surface shapes and pressure fields. 128

4.14 Dam-break over a wedge. Time evolution of the pressure force applied on the left-side of the wedge. 129

4.15 Water wheel test-case: scheme of the geometry. 130

4.16 Water wheel test-case. Qualitative representation of the free-surface shapes and velocity fields. 131

4.17 Water wheel test-case. Evolution of the smoothed pressure force magnitude applied on the bucket P. 132

4.18 Dam-break with an outflow condition: velocity field shape and comparison with the free-surface shape of a non-cut dam-break (black dots). 132

4.19 Sketch of the geometries of the two cases of solitary wave propagation. 133

4.20 Propagation of a solitary wave on a flat bottom with $\nu = 10^{-2}m^2s^{-1}$: the colours correspond to the velocity magnitude obtained with ISPH-USAW and the black lines to the analytical solution of equation (4.13). This simulation was run with 63206 particles ($\delta r = 0.01m$). 134

4.21 Propagation of a solitary wave on a flat bottom with $\nu = 10^{-6}m^2s^{-1}$: the colours correspond to the velocity magnitude obtained with ISPH-USAW and the black lines to the analytical solution of equation (4.13). This simulation was run with 63206 particles ($\delta r = 0.01m$). 135

4.22 Propagation of a solitary wave on a slope: pressure field obtained with ISPH-USAW with 30315 particles ($\delta r = 0.01m$) and $\nu = 10^{-2}m^2s^{-1}$. 135

4.23 Propagation of a solitary wave on a slope: pressure field obtained with ISPH-USAW with 30315 particles ($\delta r = 0.01m$) and $\nu = 10^{-6}m^2s^{-1}$. 137

4.24 Turbulent Poiseuille channel flow at $Re_* = 640$. Dimensionless velocity profiles. . 138

4.25 Turbulent Poiseuille channel flow at $Re_* = 640$. Profiles of dimensionless turbulent kinetic energy and dissipation rate. 138

4.26 Fish-pass after $20s$. Qualitative representation of the results. 140

4.27 Fish-pass after $20s$. Mean velocity profiles at three different locations. 140

4.28 Fish-pass after $20s$. Pressure profiles at three different locations. 141

4.29 Fish-pass after $20s$. Turbulent kinetic energy profiles at three different locations. . 141

4.30 Fish-pass after $20s$. Dissipation rate profiles at three different locations. 141

4.31 Fish-pass after $20s$. Velocity and pressure profiles on a wall. 142

4.32 Laminar plane Poiseuille flow: sketch of the two configurations. 143

4.33 Laminar plane Poiseuille flow: profiles of dimensionless temperature (left) and dynamic pressure (right) on the vertical section of the channel obtained with ISPH-USAW, compared to the theoretical solutions of Table 4.2. 144

4.34 Differentially heated square cavity at $Ra = 10^5$. Shape of the temperature (right) and velocity (left) fields obtained with ISPH-USAW (top) and FV (bottom) after convergence. 145

4.35 Differentially heated square cavity at $Ra = 10^5$. Profiles of velocity (left) and temperature (right) in $x^+ = 1/2$ and $z^+ = 1/2$ obtained with ISPH-USAW and FV after convergence. The horizontal profiles are also compared to the ones obtained by Wan *et al.* [154] with the discrete singular convolution method. 145

4.36 Differentially heated square cavity. Evolution of the Nusselt number along the cold wall of the cavity at $Ra = 10^3,\ 10^4$ and 10^5. Comparison of the results obtained with ISPH-USAW, FV and discrete singular convolution [154] after convergence. . 146

4.37 Differentially heated lid-driven cavity. Shape of the temperature (right) and velocity (left) fields obtained with ISPH-USAW (top) and FV (bottom) after convergence. 147

4.38 Differentially heated lid-driven cavity. Profiles of velocity in $x^+ = 1/2$ and $z^+ = 1/2$ obtained with ISPH-USAW and FV after convergence. 147

4.39 Differentially heated lid-driven cavity. Profiles of temperature in $z^+ = 1/2$ (left) and $x^+ = 1/2$ (right) obtained with ISPH-USAW and FV after convergence. . . . 148

4.40 Differentially heated lid-driven cavity. Evolution of the Nusselt number along the lid (dash lines) and the lower wall (solid lines). Comparison of the results obtained with ISPH-USAW and FV after convergence. 148

4.41 Lock-exchange: shape of the temperature field obtained with ISPH-USAW at $t^+ = 10$. 148

4.42 Lock-exchange: temperature contours at $t^+ = 5, 10, 15, 20$. Top: ISPH-USAW. Bottom: Härtel *et al.* [44]. 149

4.43 Turbulent plane Poiseuille flow. Profiles of velocity (top) and temperature (bottom) after convergence. Comparison of ISPH-USAW and FV with DNS results provided by Kasagi & Iida [58]. 150

4.44 Turbulent plane Poiseuille flow. Profiles of turbulent kinetic energy (top) and dissipation rate (bottom) after convergence. Comparison of ISPH-USAW and FV with DNS results provided by Kasagi & Iida [58]. 151

4.45 Differentially heated rectangular cavity. Sketch of the case (left) and shape of the temperature field after convergence with ISPH-USAW (middle) and FV (right). . . 151

4.46 Differentially heated rectangular cavity. Profiles of temperature after convergence along $x^+ = 0.125$ (left) and $z^+ = 1$ (right). Comparison of ISPH-USAW and FV with DNS results provided by Trias *et al.* [142]. 152

5 3-D cases: validation and preliminary application results 153

5.1 Laminar flow in a 3-D circular pipe with inflow/outflow conditions. Shape of the velocity profile obtained with ISPH-USAW using $\delta r^+ = 0.3$ at $t = 35s$. 155

5.2 Laminar flow in a 3-D circular pipe with inflow/outflow conditions. Dimensionless velocity profile. 156

5.3 Schematic 3-D dam-break over an obstacle. Sketch of the geometry [55]. 156

5.4 Schematic 3-D dam-break over an obstacle. Case dimensions and location of the pressure and water height probes [55]. 157

5.5 Schematic 3-D dam-break over an obstacle. Shape of the free-surface and velocity field obtained with ISPH-USAW with $\delta r = 0.02m$ at several dimensionless times. 158

5.6 Schematic 3-D dam-break over an obstacle. Time-evolution of the water depth at probes number 2 and 4. 159

5.7 Schematic 3-D dam-break over an obstacle. Time-evolution of the pressure at probes number 2 and 5. 159

5.8 Laminar flow in two connected pipes at different temperatures. Sketch of the geometry. 160

5.9 Laminar flow in two connected pipes at different temperatures. Shape of the temperature field at several times. 161

A Analytical computation of $\boldsymbol{\nabla}\gamma_a$ in 3-D with the Wendland kernel 167

A.1 3-D sketch for the computation of the gradient of γ_a in 3-D. $\boldsymbol{\nabla}\gamma_a$ is computed as a sum of integrals over the segments s. 168

A.2 3-D and 2-D notations for the computation of the gradient of γ_a in 3-D. ϕ is equal to zero everywhere outside the orange sphere in 3-D. $\boldsymbol{\nabla}\gamma_{as}$ is computed as an integral over the red line but the contribution of the dashed red lines is equal to zero. 169

A.3 Definition of the 3-D, 2-D and 1-D coordinates for the computation of the gradient of γ_a in 3-D. On the left the triangle is completely contained in the sphere of interaction of particle a, but not on the right where the clipped quantities differ from the non-clipped ones. In this sketch l_{ev_0} is higher than 0 whereas l_{ev_2} is lower than zero. 170

List of Tables

1 Governing equations and modelling choices **5**

1.1 Values of the $k-\epsilon$ model constants [69] 19

1.2 Summary of the boundary conditions imposed on the fields at the solid walls $\partial\Omega_w$, the free-surface $\partial\Omega_\eta$, the inflow $\partial\Omega_i$ and outflow $\partial\Omega_o$ boundaries. 29

2 SPH: Literature review **31**

3 A new incompressible SPH model **83**

3.1 Non-preconditioned Bi-CGSTAB algorithm for the resolution of $\boldsymbol{Ax}=\boldsymbol{B}$ [153]. . 108

3.2 Summary of the various cases for the calculation of $\tilde{\gamma}_{as}$ in 2-D. 113

4 Validation **115**

4.1 CPU times obtained on several test-cases. 136

4.2 Laminar plane Poiseuille flow: theoretical solutions for T^+ and p^{*^+} 143

5 3-D cases: validation and preliminary application results **153**

A Analytical computation of $\boldsymbol{\nabla}\gamma_a$ in 3-D with the Wendland kernel 167

Nomenclature

Remarks

A denotes an arbitrary field, $A_a = A(\boldsymbol{r}_a)$, $A_s = A(\boldsymbol{r}_s)$ and $A_{ab} = A_a - A_b$.

Abbreviations

CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
DNS	Direct Numerical Simulation
FV	Finite Volumes
GPU	Graphics Processing Unit
ISPH	Incompressible SPH
LES	Large Eddy Simulation
RANS	Reynolds-Averaged Navier–Stokes
RSTM	Reynolds-Stress Transport Models
SPH	Smoothed Particle Hydrodynamics
USAW	Unified semi-analytical wall boundary conditions
VoF	Volume of Fluid
WCSPH	Weakly Compressible SPH

Greek symbols

β	Coefficient of thermal expansion	(K^{-1})
ξ	Constant of the Tait state equation	
δ	Dirac distribution	(m^{-d})

δr	Initial interparticle distance	(m)
δr_{as}	See equation (3.36)	(m)
ΔT	Maximum temperature difference of the flow	(K)
δt	Time-step	(s)
ϵ	Dissipation rate of the turbulent kinetic energy	$(m^2\ s^{-3})$
γ_a	Kernel renormalisation factor	
κ	von Kármán constant	
λ	First Lamé coefficient equal to $\zeta - \frac{2}{3}\mu$	$(kg\ m^{-1}\ s^{-1})$
λ_T	Thermal conductivity	$(kg\ m\ s^{-3}\ K^{-1})$
μ	Dynamic molecular viscosity	$(kg\ m^{-1}\ s^{-1})$
μ_E	Effective dynamic viscosity $(\mu + \mu_T)$	$(kg\ m^{-1}\ s^{-1})$
μ_ϵ	$\mu + \frac{\mu_T}{\sigma_\epsilon}$	$(kg\ m^{-1}\ s^{-1})$
μ_k	$\mu + \frac{\mu_T}{\sigma_k}$	$(kg\ m^{-1}\ s^{-1})$
μ_T	Dynamic eddy viscosity	$(kg\ m^{-1}\ s^{-1})$
ν	Kinematic molecular viscosity	$(m^2\ s^{-1})$
ν_E	Effective kinematic viscosity $(\nu + \nu_T)$	$(m^2\ s^{-1})$
ν_T	Kinematic eddy viscosity	$(m^2\ s^{-1})$
Ω	Computational domain	
$\partial\Omega$	Boundary of the computational domain	
$\partial\Omega_\eta$	Free-surface boundary of the computational domain	
$\partial\Omega_i$	Inflow boundary of the computational domain	
$\partial\Omega_{i/o}$	Inflow/outflow boundary of the computational domain	
$\partial\Omega_o$	Outflow boundary of the computational domain	
$\partial\Omega_w$	Solid wall boundary of the computational domain	
ρ	Density	$(kg\ m^3)$
ρ_0	Reference density in WCSPH	$(kg\ m^3)$

$\boldsymbol{\sigma}$ Cauchy stress tensor ..... $(kg\ m^{-1}\ s^{-2})$

σ_ϵ Constant of the $k-\epsilon$ model

σ_k Constant of the $k-\epsilon$ model

$\boldsymbol{\tau}$ Shear-stress vector ..... $(kg\ m^{-1}\ s^{-2})$

ζ Bulk viscosity ..... $(kg\ m^{-1}\ s^{-1})$

Roman symbols

c_0 Numerical speed of sound ..... $(m\ s^{-1})$

C_{ϵ_1} Constant of the $k-\epsilon$ model

C_{ϵ_2} Constant of the $k-\epsilon$ model

$C_{\epsilon_2,Y}$ C_{ϵ_2} constant of the $k-\epsilon$ model after the Yap correction

$C_{\epsilon 3}$ Constant of the $k-\epsilon$ model

C_μ Constant of the $k-\epsilon$ model

d Geometrical dimension of the computational domain

D_a Simplest classical SPH divergence operator (see equation (2.40)) ..... (m^{-1})

D_a^- Symmetric classical SPH divergence operator (see equation (2.41)) ..... (m^{-1})

$D_a^{\gamma,-}$ Symmetric USAW SPH divergence operator (see equation (2.146)) ..... (m^{-1})

D_a^+ Antisymmetric classical SPH divergence operator (see equation (2.42)) ..... (m^{-1})

$\mathcal{F}$ Set of free particles

$\mathbb{G}$ Buoyancy term in the $k-\epsilon$ equations ..... $(m^2 s^{-3})$

$\boldsymbol{g}$ Gravity field ..... $(m\ s^{-2})$

$\boldsymbol{G}_a$ Simplest classical SPH gradient operator (see equation (2.33)) ..... (m^{-1})

$\boldsymbol{G}_a^-$ Symmetric classical SPH gradient operator (see equation (2.36)) ..... (m^{-1})

$\boldsymbol{G}_a^{\gamma,-}$ Symmetric USAW SPH gradient operator (see equation (2.144)) ..... (m^{-1})

$\boldsymbol{G}_a^+$ Antisymmetric classical SPH gradient operator (see equation (2.39)) ..... (m^{-1})

$\boldsymbol{G}_a^{\gamma,+}$ Antisymmetric USAW SPH gradient operator (see equation (2.145)) ..... (m^{-1})

Gr Grashoff number

H	Reference free-surface elevation	(m)
h	Smoothing length	(m)
K	Thermal diffusivity	$(m^2\ s^{-1})$
k	Turbulent kinetic energy	$(m^2\ s^{-2})$
K_E	Effective thermal diffusivity $(K + K_T)$	$(m^2\ s^{-1})$
K_T	Turbulent thermal diffusivity	$(m^2\ s^{-1})$
L	Characteristic length scale	(m)
L_a^γ	Ferrand SPH Laplacian operator (see equation (2.153))	(m^{-2})
L_a	Morris SPH Laplacian operator (see equation (2.58))	(m^{-2})
L_t	Size of the large eddies $\sim \frac{k^{3/2}}{\epsilon}$	(m)
m_b	Mass of particle b	(kg)
$\boldsymbol{n}$	Unit normal vector to the boundary	
$\boldsymbol{n}_s$	Inward unit normal vector to the boundary at segment s	
Nu	Nusselt number	
$\mathcal{P}$	Set of fluid particles	
p	Pressure	(Pa)
$\mathbb{P}$	Production of turbulent kinetic energy	$(m^2 s^{-3})$
Pr	Prandlt number	
Pr_T	Turbulent Prandlt number	
p^*	Dynamic pressure $(p^* = p + \rho g z)$	(Pa)
$\tilde{p}$	Modified pressure defined as $\tilde{p} = p + \frac{2}{3}\rho k$	(Pa)
q_{max}	Dimensionless kernel support size	
$\boldsymbol{R}$	Reynolds stress tensor	$(m^2 s^{-2})$
$\boldsymbol{r}$	Position	(m)
Ra	Rayleigh number	
Re	Reynolds number	

$\mathcal{S}$	Set of boundary elements (segments)	
s	Scalar strain-rate ($s = \sqrt{2\boldsymbol{s} : \boldsymbol{s}}$)	(s^{-1})
$\boldsymbol{s}$	Strain-rate tensor	(s^{-1})
S	Mean scalar strain rate ($S = \sqrt{2\boldsymbol{S} : \boldsymbol{S}}$)	(s^{-1})
$\boldsymbol{S}$	Mean strain rate tensor	(s^{-1})
T	Temperature	(K)
t	Time	(s)
T_0	Mean temperature of the flow	(K)
T_c	Lowest temperature of the flow	(K)
T_h	Highest temperature of the flow	(K)
$\boldsymbol{u}$	Eulerian velocity	$(m\ s^{-1})$
U	Characteristic velocity scale	$(m\ s^{-1})$
u_k	Friction velocity based on the turbulent kinetic energy	$(m\ s^{-1})$
u_*	Friction velocity based on the log-law	$(m\ s^{-1})$
$\mathcal{V}$	Set of vertex particles	
$\boldsymbol{v}$	Lagrangian velocity	$(m\ s^{-1})$
V_b	Volume of particle b	(m^d)
v_{max}	Maximum velocity of the flow	$(m\ s^{-1})$
$\tilde{\boldsymbol{v}}^{n+1}$	Estimation of the velocity obtained through the projection methods first step	$(m\ s^{-1})$
w	Kernel function	(m^{-d})
y^+	Dimensionless distance to a wall in turbulent regime ($y^+ = \frac{yu_*}{\nu}$)	

Other Symbols

$\boldsymbol{\nabla}\gamma_a$	Gradient of the renormalisation factor γ_a	(m^{-1})
$\boldsymbol{\nabla}\gamma_{as}$	Contribution of segment s to the gradient of γ_a ($\boldsymbol{\nabla}\gamma_a = \sum_{s \in S} \boldsymbol{\nabla}\gamma_{as}$)	(m^{-1})
$\boldsymbol{\nabla} w_{ab}$	Kernel gradient ($= \boldsymbol{\nabla}_{\boldsymbol{r}_a} w(\boldsymbol{r}_{ab})$)	(m^{-d+1})

Introduction

Il existe de nombreuses possibilités en ce qui concerne la construction de modèles numériques en mécanique des fluides. On peut cependant les classer dans deux catégories : les méthodes eulériennes et les méthodes lagrangiennes. Le choix d'un point de vue eulérien a donné lieu à une première classe de méthodes où les grandeurs physiques sont calculées dans un référentiel immobile de l'espace. Cependant, il existe des situations pour lesquelles cette discrétisation n'est pas la plus adaptée, notamment si l'écoulement subit de très fortes distorsions. Afin de modéliser de tels écoulements, les méthodes de type lagrangien peuvent s'avérer efficaces. Dans ce cas, le calcul des grandeurs physiques est fait dans des référentiels mobiles liés à des "points de matière", se déplaçant avec eux au cours du temps. Ce travail porte sur des développements effectués dans un logiciel de simulation par la méthode Smoothed Particle Hydrodynamics (SPH), qui est la méthode lagrangienne la plus connue. Cette dernière, bien que présentant divers avantages par rapport à des méthodes classiques de modélisation eulériennes, n'a pas encore atteint la même maturité. L'objectif de cette thèse est de présenter des contributions à un modèle de simulation par la méthode SPH en vue de l'application à des cas industriels. Le Chapitre 1 introduit les choix de modélisation, les équations à résoudre et leur jeu de conditions aux limites (dans un formalisme continu pour les fluides). Un état de l'art de la méthode SPH pour la résolution des équations de Navier–Stokes est présenté au Chapitre 2. Le Chapitre 3 concerne la description du modèle proposé. La validation du modèle en 2-D et en 3-D fait l'objet des Chapitres 4 et 5. Au Chapitre 5 des résultats préliminaires sur un cas d'application sont également présentés.

There are many possibilities regarding the construction of numerical models in computational fluid dynamics (CFD). They may belong to two different types of approaches, namely the Eulerian and Lagrangian approaches. The choice of an Eulerian point of view gave rise to models in which the physical quantities are calculated in motionless frames of reference[1]. Space is then discretised through a mesh and the physical quantities are estimated at the cells over time, based on the fluxes values through the faces. The Finite Elements (FE), Finite Volume (FV), Finite Differences methods, among others, belong to that class of numerical methods. Their mathematical foundations are well understood, with demonstrated convergence properties and the possibility to estimate error propagation in the models, and they are the most widely used methods in CFD. On the other hand, in Lagrangian methods the physical quantities are calculated at moving points of space commonly called particles, that are associated to "small amounts of matter" they carry. The space discretisation is then composed of these moving particles. The idea is to solve systems of discrete equations in which the interactions between particles depend on their mutual distances and on the physical quantities they carry (velocity, density, pressure, etc.). The particles motion is determined by the discretised equations of motion, which corresponds to the convection of the physical quantities and to the distribution of interpolation points. Note that the treatment of convection in Lagrangian methods is thus straightforward, whereas it is a very complex problem in the case of Eulerian methods. The Finite Point, Diffuse Element, Free Mesh, Dissipative Particle Dynamics, Moving Particle Semi-implicit methods belong to the class of Lagrangian numerical methods, as well as the Smoothed Particle Hydrodynamics (SPH) method which is the most known and used of them[2].

While Eulerian methods are very well suited to the study of confined flows and free-surface flows with low rates of distorsion, their application to highly distorted free-surface flows, or to flows around complex moving objects, is more problematic. The Lagrangian framework seems more adapted to such flows, meaning that if Lagrangian methods were as accurate as Eulerian methods, with the possibility to estimate error propagation and nice convergence properties, it would seem more advantageous to use Lagrangian instead of Eulerian methods on such flows. However this is not the case, so that developments for this kind of simulations are being done in Eulerian methods, with the Volume of Fluid (VoF) or level-set methods for instance, and in hybrid methods called Arbitrary Lagrangian-Eulerian (ALE) mesh-based methods. On the other hand, Lagrangian approaches are getting increasing importance, in particular the SPH method.

To this day, the latter has been used in the industry to simulate complex free-surface flows, most of time involving moving objects, that Eulerian methods can hardly handle [84]. Though, this was quite punctual since the method suffers from a number of issues that prevent its wider use. On the one hand, there remains many unanswered questions regarding the convergence properties of the method, its numerical stability, etc., which makes the method less reliable than FE or FV. On the other hand, modelling incompressible flows with SPH has classically been done through weakly

[1] Note that the mesh may move at an imposed velocity in the case of Arbitrary Lagrangian-Eulerian (ALE) approaches.

[2] Some mesh-free methods strictly speaking have a mesh, and there exists considerable work on hybrid methods, *e.g.* the Particle Finite Elements Methods.

compressible SPH (WCSPH) models, as is thoroughly described in [103]. The pressure is then calculated through an equation of state, which causes the pressure prediction to be noisy and, in many cases, inaccurate. This is very problematic when it comes to estimating the forces applied on solids in the flow. Thus, incompressible SPH (ISPH) models have been developed to improve the accuracy of the pressure prediction, the latter then being computed through the resolution of a Poisson equation. Another issue is the representation of boundaries in SPH, which requires special treatments in order to prevent the particles from crossing walls or to impose inflow/outflow conditions (some SPH models also require a free-surface treatment, but not all of them). The boundary conditions associated with engineering turbulence modelling approaches are quite increasing the demands for an accurate boundary condition management which also offers the flexibility for all kinds of Dirichlet, Neumann and Robin conditions. The modelling of turbulence remains problematic in SPH, as well as other crucial phenomena in fluid mechanics such as active scalars effects, sediment transport, air entrainment, etc., which could turn SPH into a very advantageous modelling tool. Although an important amount of work among the SPH community has already been provided to address these issues, much work remains to be done. Note that the high computational times of the method used to be a problem too, but it has recently been tackled through the use of Graphics Cards Units (GPU) [45], which are powerful massively parallel processors originally aimed at video game rendering.

The aim of this thesis is to build an SPH model for industrial applications, able to:

- represent 3-D free-surface or confined flows in laminar or turbulent regime;
- accurately predict the pressure forces applied on rigid bodies in the flow;
- represent active scalars effects (*e.g.* temperature) and their interaction with turbulence;
- include the presence of open boundaries.

Note that in this thesis the framework of temperature was chosen for the sake of simplicity in the notations[3], but the model applies to other active scalars. An SPH model was thus built, based on the most recent breakthroughs in SPH regarding the quality of pressure predictions, the boundary conditions treatment, inflow/outflow boundaries, turbulence and buoyancy modelling. The main achievements of this work are:

- the development of an ISPH model consistent with the unified semi-analytical boundary conditions technique;
- the improvement of an existing SPH $k - \epsilon$ turbulence closure and its introduction into the ISPH model;
- the introduction of a buoyancy model based on the Boussinesq approximation into ISPH that accounts for buoyancy/turbulence interactions;

[3]The vocabulary differs much between temperature effects and scalar concentration effects, although the mathematics behind it is similar.

- the development of an open boundary formulation for ISPH with the unified semi-analytical boundary conditions;
- the extension of the model to 3-D and its implementation in a GPU framework.

At first, the model was implemented in a 2-D sequential SPH code called SPARTACUS, but later on it was implemented in a 2-D/3-D massively parallel SPH code called Sphynx, able to run on one GPU card. This was mandatory to keep the computational times compatible with industrial requirements. The GPU code was derived from the open-source code called GPUSPH, and was written in the Cuda programming language.

The structure of this thesis is articulated in five Chapters:

- In Chapter 1, the Navier–Stokes equations for incompressible flows are introduced, along with the set of boundary conditions considered in this work. Then, a quick review of the projection methods for their resolution is given, followed by a review of the techniques for turbulence and buoyancy treatments in the framework of continuous fluids. The outcome of this Chapter is the complete system of equations to be solved in a continuous framework and the associated set of boundary conditions.
- Chapter 2 is a literature review concerning the SPH method, that focuses on its application to the set of equations and boundary conditions outlined in Chapter 1. The existing possibilities regarding the space-time discretisation of the equations are described. Then, the most recent techniques for wall, free-surface and open boundary treatments are described. This is followed by a review of the existing turbulence and buoyancy SPH models, and finally by a review of what has been achieved in SPH regarding the reduction of computational times through parallel programming (including GPU).
- The description of the present SPH model is the subject of Chapter 3. First, the space-time discretisation of the system of equations outlined in Chapter 1 is described. Turbulence and buoyancy treatments are included in this description, their interactions being taken into account. Then, the treatments for wall boundary conditions, free-surface boundary conditions and inlet/outlet boundary conditions are explained. After that, the resolution of the pressure Poisson equation is dealt with. Finally, a technique for the analytical computation of boundary integrals, necessary with the boundary conditions used herein, is described.
- Chapter 4 deals with the validation of the model on 2-D cases. First, 2-D isothermal laminar and turbulent flows are considered. Various free-surface and confined flows are presented, some of them involving open-boundaries, one of them involving a moving rigid body driving the flow. Then, 2-D non-isothermal laminar and turbulent flows are considered, all of them confined.
- In Chapter 5, the validation on 3-D flows and preliminary results on an industrial application are presented.

Chapter 1

Governing equations and modelling choices

L'objectif de ce Chapitre est double : d'une part, poser les bases du modèle développé au cours de cette thèse, qui sera présenté au Chapitre 3. D'autre part, introduire les notations utilisées dans la suite du document. Dans un premier temps, les équations de Navier–Stokes sont présentées, ainsi que le jeu de conditions aux limites utilisé dans ce travail. Un grand nombre de méthodes ont été développées pour la résolution des équations de Navier–Stokes incompressibles. Une synthèse rapide des diverses méthodes de projection est faite. Ensuite, la modélisation de la turbulence est abordée, principalement concentrée sur les modèles fondés sur la moyenne de Reynolds. Enfin, on aborde la modélisation de l'influence de scalaires actifs sur les écoulements et leurs interactions avec la turbulence. Ces thèmes peuvent être étudiés bien plus en profondeur, dans les ouvrages [77], [147], [114] et [120] par exemple. On a pris soin de bien indiquer quelles hypothèses et approximations ont été faites lors des choix de modélisation. Des pistes d'améliorations du modèle présenté au Chapitre 3 apparaissent à ce niveau. Ce Chapitre aboutit au système d'équations que l'on souhaite résoudre ainsi qu'aux conditions aux limites envisagées.

This Chapter aims at introducing the notations used in this work, as well as defining the scope of the model that was developed, and that will be presented in Chapter 3. It does not claim to be an exhaustive review of the existing literature about the Navier–Stokes equations, turbulence and buoyancy modelling. Such reviews can be found in the literature, *e.g.* in [77], [147], [114] or [120]. Care was given to laying the stress on the approximations that were made in the modelling choices. Some insights are given about possible improvements of the model presented in Chapter 3.

1.1 Navier–Stokes equations for incompressible flows

1.1.1 Formulation

The Navier–Stokes equations for incompressible flows consist of two equations: the continuity and the momentum equations. We first consider a possibly compressible flow in a domain Ω of dimension d (in practice 2 or 3). The compressible continuity equation represents the mass conservation in a continuous medium and reads:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{1.1}$$

where ρ is the density, $\boldsymbol{v}$ the velocity, and t is the time (the continuous divergence operator is denoted by $\nabla\cdot$). The Navier–Stokes momentum equation is obtained from the Cauchy momentum equation for a continuous medium where a behaviour law is introduced to model the stress tensor. We recall the Cauchy equation:

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \boldsymbol{\nabla})\boldsymbol{v} = \frac{1}{\rho}\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \boldsymbol{g} \tag{1.2}$$

In this equation, $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\boldsymbol{g}$ is the acceleration due to gravity (the continuous gradient operator is denoted by $\boldsymbol{\nabla}$). Note that equation (1.2) was written in an Eulerian framework, where the position field $\boldsymbol{r}$ is given by the initial condition $\boldsymbol{r}(t_0) = \boldsymbol{r}_0$ with t_0 the initial time.

On the other hand, the behaviour law used for a Newtonian fluid reads [1]:

$$\begin{aligned} \boldsymbol{\sigma} &= -p\boldsymbol{I}_d + \boldsymbol{\tau} \\ \boldsymbol{\tau} &= \lambda \, \nabla \cdot \boldsymbol{v} \, \boldsymbol{I}_d + 2\mu \boldsymbol{s} \end{aligned} \tag{1.3}$$

where $\boldsymbol{I}_d$ is the identity tensor in dimension d, $\boldsymbol{\tau}$ is the shear-stress tensor, $\boldsymbol{s}$ is the strain-rate tensor, p is the pressure, μ is the dynamic molecular viscosity and λ is equal to $\zeta - \frac{2}{3}\mu$ where ζ is

[1] In this work, non-Newtonian fluids are not considered since at this stage the scope of applications of the developed model concerns water or air flows. Though, it is possible to introduce non-Newtonian models in SPH as was done in [48] for example.

the bulk viscosity. The tensor $\boldsymbol{s}$ is defined as:

$$\boldsymbol{s} = \frac{1}{2}\left[\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^T\right] \tag{1.4}$$

where ^{T} denotes the transpose of a vector or tensor. The behaviour law (1.3) was obtained by introducing a model for viscosity in fluids based on an analogy with a model for particle friction.

It is possible to write system (1.2) in a Lagrangian form by changing the definition of the position field $\boldsymbol{r}$, and define it as the position field at time t considering the initial condition $\boldsymbol{r}_0$. Then, the time derivative of the position is the velocity field. With this change of variable in (1.2) and the behaviour law (1.3), the Navier–Stokes equations in their Lagrangian form for a compressible flow are obtained:

$$\begin{cases} \dfrac{d\rho}{dt} = -\rho \nabla \cdot \boldsymbol{v} \\ \dfrac{d\boldsymbol{v}}{dt} = -\dfrac{1}{\rho}\boldsymbol{\nabla} p + \dfrac{1}{\rho}\boldsymbol{\nabla}\cdot\left(\mu\left[\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^T\right]\right) + \dfrac{1}{\rho}\boldsymbol{\nabla}(\lambda \nabla\cdot\boldsymbol{v}) + \boldsymbol{g} \\ \dfrac{d\boldsymbol{r}}{dt} = \boldsymbol{v} \end{cases} \tag{1.5}$$

The third line of this system is an equation of advection which accounts for the fact that in the Lagrangian framework the equations are solved at moving points of space. This system is not closed and it is necessary to introduce an equation of state linking the density to the pressure in order to numerically solve it. For fluids like water, this equation is that introduced by Tait [135]:

$$p = \frac{\rho_0 c_0^2}{\xi}\left(\left(\frac{\rho}{\rho_0}\right)^{\xi} - 1\right) \tag{1.6}$$

where ξ is a constant, usually taken equal to 7 for water, ρ_0 is a reference density ($1000 kg.m^{-3}$ for water) and c_0 is a speed of sound. In SPH, a weakly-compressible form of system (1.5) has classically been solved (see Chapter 2, Section 2.3.1). In this case, only slight variations of the density are possible (typically a Mach number lower than 0.1), which is achieved by setting a large enough value for c_0.

In this work, an SPH model is built for the resolution of the incompressible Navier–Stokes equations (see Chapter 3), *i.e.* with $\nabla \cdot \boldsymbol{v} = 0$. Besides we only consider homogeneous flows so that $\rho =$ constant. Since $\nabla \cdot \boldsymbol{v} = 0$, the terms involving λ in (1.3) vanish. The Navier–Stokes equations in their Lagrangian form for an incompressible flow then read:

$$\begin{cases} \nabla \cdot \boldsymbol{v} = 0 \\ \dfrac{d\boldsymbol{v}}{dt} = -\dfrac{1}{\rho}\boldsymbol{\nabla} p + \dfrac{1}{\rho}\boldsymbol{\nabla}\cdot\left(\mu\left[\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^T\right]\right) + \boldsymbol{g} \\ \dfrac{d\boldsymbol{r}}{dt} = \boldsymbol{v} \end{cases} \tag{1.7}$$

Note that if the viscosity is constant in space, for an incompressible flow the term involving $(\boldsymbol{\nabla}\boldsymbol{v})^T$ vanishes while the other viscous term reads $\nu\boldsymbol{\nabla}^2\boldsymbol{v}$ where ν is the kinematic molecular viscosity ($\nu = \frac{\mu}{\rho}$). Here we chose to consider possible variations of the viscosity due to our use of an eddy viscosity model to represent turbulence effects (see Section 1.2). The viscosity can also vary due to temperature variations or in case of multifluid flows, though this was not considered in this work.

The closure of system (1.7) involves the imposition of initial conditions on the position and the velocity, and of boundary conditions on the velocity[2]. The pressure is the Lagrangian multiplier that stems from the minimisation of the momentum equation under the constraint $\nabla \cdot \boldsymbol{v} = 0$ [9]. Methods to compute it are described in section 1.1.2. While the initial conditions are usually quite easily set, the boundary conditions can prove problematic to prescribe in numerical models (in particular in SPH as we will see in Chapters 2 and 3). The boundary condition may set the value of the field itself, in which case it is called a Dirichlet condition. It may also set components of the field gradient and is then called a Neumann condition. Robin conditions correspond to a combination of the two previous types. Let $\partial\Omega$ be the boundary of the computational domain Ω, decomposed into $\partial\Omega_w$ the solid walls, $\partial\Omega_\eta$ the free-surface, $\partial\Omega_i$ the inflow boundaries and $\partial\Omega_o$ the outflow boundaries. The boundary condition imposed on the velocity at solid walls, considering a viscous fluid, is a Dirichlet condition:

$$\boldsymbol{v}|_{\partial\Omega_w} = \boldsymbol{v}_w \tag{1.8}$$

This condition is called a no-slip condition, $\boldsymbol{v}_w$ being the velocity of the wall. On the other hand, considering that the vertical coordinate z of the free-surface is known (z pointing upwards), the boundary condition at the free-surface reads:

$$\begin{cases} \left(\dfrac{\partial z}{\partial t} + \boldsymbol{v}\cdot\boldsymbol{n}\right)_{\partial\Omega_\eta} = 0 \\ \boldsymbol{\tau}\cdot\boldsymbol{n}|_{\partial\Omega_\eta} = 0 \end{cases} \tag{1.9}$$

where $\boldsymbol{n}$ is the unit outward normal to the boundary. The first line is the kinematic condition while the second line is the dynamic condition at the free-surface that corresponds to the continuity of stresses across an interface. Finally, the boundary conditions imposed at open boundaries depend on the type of boundary considered, whether an inlet or an outlet. Usually, at an inlet a Dirichlet condition is imposed:

$$\boldsymbol{v}|_{\partial\Omega_i} = \boldsymbol{v}_i \tag{1.10}$$

whereas at an outlet a homogeneous Neumann condition is imposed:

$$\left.\frac{\partial\boldsymbol{v}}{\partial\boldsymbol{n}}\right|_{\partial\Omega_o} = 0 \tag{1.11}$$

[2]Note that in fact, with these initial and boundary conditions the existence and uniqueness of a global solution to the 3-D Navier–Stokes equations (with any source term and on any time interval) was not proved, although it was proved on various particular cases [77].

1.1.2 Projection methods

In this section a review of the projection methods for the resolution of (1.7) is given, beginning with general considerations regarding this kind of methods. A very quick review is then given of the numerous variants of the projection method, based on the 2006 paper by Guermond *et al.* [40]. In this section the viscosity is considered as constant for the sake of simplicity in the notations.

1.1.2.1 General considerations on the projection methods

While the numerical resolution of (1.5) (using (1.6)) can be done through classical fractional step methods, the resolution of (1.7) is made complex by the pressure-velocity coupling in the equations, that prevents the use of such schemes. Indeed, as said earlier the pressure is the Lagrangian constraint that enforces the divergence-free constraint [9]. In 1968, Chorin [15] and Temam [138] introduced a projection-method for the approximate resolution of (1.7), which made it possible to solve a sequence of decoupled equations on the velocity and on the pressure at each time-step. Such an algorithm is very interesting in terms of computational cost. Its theory is based on the Helmholtz-Hodge decomposition, which states that any vector $\boldsymbol{v}$ can be decomposed into the sum of a curl-free vector and a divergence-free vector. Indeed, let us consider two Euclidean vectorial spaces E and F:

$$\begin{aligned} E &= \left\{\boldsymbol{v} \in \mathcal{C}^1(\Omega, \mathbb{R}^3),\ \boldsymbol{v}\cdot\boldsymbol{n}|_{\partial\Omega_w} = 0\right\} \\ F &= \left\{p \in \mathcal{C}^1(\Omega, \mathbb{R}),\ p|_{\partial\Omega_\eta} = 0\right\} \end{aligned} \tag{1.12}$$

with the following scalar products on E and F:

$$\begin{aligned} \langle \boldsymbol{u}, \boldsymbol{v}\rangle &= \int_\Omega \boldsymbol{u}\cdot\boldsymbol{v} d\Omega \\ (p, q) &= \int_\Omega pq d\Omega \end{aligned} \tag{1.13}$$

Considering that $\int_\Omega \nabla\cdot(p\boldsymbol{v})d\Omega = (p, \nabla\cdot\boldsymbol{v}) + \langle\boldsymbol{\nabla} p, \boldsymbol{v}\rangle$, the following relation is found:

$$(p, \nabla\cdot\boldsymbol{v}) + \langle\boldsymbol{\nabla} p, \boldsymbol{v}\rangle = \oint_{\partial\Omega} p\boldsymbol{v}\cdot\boldsymbol{n} d\Gamma = 0 \tag{1.14}$$

which shows that the gradient and divergence operators are skew-adjoint in these spaces. An important consequence is that the kernel of $(\nabla\cdot)$, denoted by K, is orthogonal to the image of $(-\boldsymbol{\nabla})$. Thus, the following property holds:

$$\forall \boldsymbol{v} \in E, \exists!\, (\tilde{\boldsymbol{v}}, p) \in K \times F,\ \boldsymbol{v} = \tilde{\boldsymbol{v}} + \boldsymbol{\nabla} p \tag{1.15}$$

Applying the divergence operator to this equation gives:

$$\nabla \cdot \boldsymbol{v} = \nabla^2 p \tag{1.16}$$

where ∇^2 is the Laplacian operator. This then yields:

$$\tilde{\boldsymbol{v}} = \boldsymbol{v} - \boldsymbol{\nabla} \left[\left(\nabla^2 \right)^{-1} \nabla \cdot \boldsymbol{v} \right] \tag{1.17}$$

This shows that the projection operator defined by:

$$\boldsymbol{P} = \left(\boldsymbol{I}_d - \boldsymbol{\nabla} \left[\left(\nabla^2 \right)^{-1} \nabla \cdot \right] \right) \tag{1.18}$$

projects any vector of E onto the space of divergence-free vector fields, provided that the divergence and gradient operators are skew-adjoint.

The idea of projection methods is to split the resolution of the momentum equation into two sub-steps. In the first one, an estimation of the velocity is computed, which does not satisfy the incompressibility constraint. Then, the projection operator $\boldsymbol{P}$ is used to project this velocity field onto the space of divergence-free vector fields.

1.1.2.2 Pressure-correction schemes

In all the variants of pressure-correction schemes, the estimated velocity is computed based on the viscous and external forces. Some variants also take the pressure gradient of the former time-step into account at this stage. In the second sub-step, the estimated velocity is corrected through its projection onto the vectorial space K of divergence-free vectors. In what follows the main two kinds of pressure-correction schemes are briefly presented.

***i*) Non-incremental pressure-correction scheme**:
This scheme is the original one proposed by Chorin and Temam in 1968 [15, 139]. In the first sub-step, the velocity estimation is based on the viscous and external forces only:

$$\frac{\tilde{\boldsymbol{v}}^{n+1} - \boldsymbol{v}^n}{\delta t} = \nu \boldsymbol{\nabla}^2 \tilde{\boldsymbol{v}}^{n+1} + \boldsymbol{g} \tag{1.19}$$

$\tilde{\boldsymbol{v}}^{n+1}$ is the estimated velocity field, δt is the time-step size and the superscripts n correspond to the time iteration number. The pressure gradient then intervenes in the second sub-step:

$$\frac{\boldsymbol{v}^{n+1} - \tilde{\boldsymbol{v}}^{n+1}}{\delta t} = -\frac{1}{\rho} \boldsymbol{\nabla} p^{n+1} \tag{1.20}$$

which corresponds to the projection of $\tilde{\boldsymbol{v}}^{n+1}$ onto the divergence-free vectorial space. Indeed, the

pressure p^{n+1} involved in (1.20) was previously computed through a pressure Poisson equation:

$$\nabla^2 p^{n+1} = \frac{\rho}{\delta t} \nabla \cdot \tilde{\boldsymbol{v}}^{n+1} \tag{1.21}$$

which corresponds to the enforcement of the incompressibility constraint $\nabla \cdot \boldsymbol{v}^{n+1} = 0$ on (1.20). In this scheme the wall boundary conditions applied to the velocity field read:

$$\begin{cases} \tilde{\boldsymbol{v}}^{n+1}|_{\partial\Omega_w} = 0 \\ \boldsymbol{v}^{n+1} \cdot \boldsymbol{n}|_{\partial\Omega_w} = 0 \end{cases} \tag{1.22}$$

Note that the condition $\tilde{\boldsymbol{v}}^{n+1} = 0$ is necessary because the viscous term is treated implicitly. On the other hand, pressure boundary conditions are now necessary for the resolution of (1.21), which involves second derivatives of the pressure. The pressure wall boundary condition is obtained by projecting equation (1.20) onto the normal to the wall, which yields:

$$\boldsymbol{\nabla} p^{n+1} \cdot \boldsymbol{n}|_{\partial\Omega_w} = -\frac{\rho}{\delta t} \left(\boldsymbol{v}^{n+1} - \tilde{\boldsymbol{v}}^{n+1} \right) \cdot \boldsymbol{n}|_{\partial\Omega_w} = 0 \tag{1.23}$$

due to the conditions (1.22). This homogeneous Neumann condition is artificial and was shown to induce a numerical boundary layer which deteriorates the scheme convergence [119]. Note that the correct pressure wall boundary condition is obtained by projecting the momentum equation (2nd line of (1.7)) onto the normal to the wall, which yields:

$$\frac{\partial}{\partial \boldsymbol{n}} \left(\frac{v^2}{2} + \frac{p^*}{\rho} \right) \Bigg|_{\partial\Omega_w} = \left(\nu \boldsymbol{\nabla}^2 \boldsymbol{v} \right) \cdot \boldsymbol{n}|_{\partial\Omega_w} \tag{1.24}$$

where the dynamic pressure p^* was defined as:

$$p^* = p + \rho g z \tag{1.25}$$

Recall that z is the vertical coordinate, oriented upwards. Treating the viscous term explicitly, in the Lagrangian framework there is no need to impose boundary conditions on $\tilde{\boldsymbol{v}}^{n+1}$ and the wall boundary condition on $\boldsymbol{v}^{n+1}$ is imposed through (1.8). Equation (1.19) is then replaced by:

$$\frac{\tilde{\boldsymbol{v}}^{n+1} - \boldsymbol{v}^n}{\delta t} = \nu \boldsymbol{\nabla}^2 \boldsymbol{v}^n + \boldsymbol{g} \tag{1.26}$$

The pressure wall boundary condition, obtained by projecting (1.20) onto the normal to the wall, is now non-homogeneous since the velocity boundary condition has changed, and reads:

$$\frac{\partial p^{n+1}}{\partial \boldsymbol{n}} \Bigg|_{\partial\Omega_w} = \frac{\rho}{\delta t} \tilde{\boldsymbol{v}}^{n+1} \cdot \boldsymbol{n}|_{\partial\Omega_w} = \left(\rho \boldsymbol{g} + \mu \boldsymbol{\nabla}^2 \boldsymbol{v}^n \right) \cdot \boldsymbol{n}|_{\partial\Omega_w} \tag{1.27}$$

which is in agreement with (1.24). This boundary condition was shown to yield more accurate results than a homogeneous Neumann condition [40].

At the free-surface and open boundaries the pressure boundary conditions are imposed through:

$$\begin{cases} p|_{\partial\Omega_\eta} = p_{atm} \\ \left.\dfrac{\partial p}{\partial \boldsymbol{n}}\right|_{\partial\Omega_i} = 0 \\ p|_{\partial\Omega_o} = p_o \end{cases} \tag{1.28}$$

where p_{atm} is the atmospheric pressure, considered as constant in this work so that it is taken equal to zero. p_o is the pressure at the outflow boundary, which can be imposed as constant for example. Note that in order to model an outlet through which the fluid is free to flow, it is recommended to use a radiative condition such as the one proposed by Orlanski [111]:

$$\left(\frac{\partial p}{\partial t} + C\frac{\partial p}{\partial \boldsymbol{n}}\right)_{\partial\Omega_o} = 0 \tag{1.29}$$

with C a celerity usually taken as $\sqrt{gH}$, H being the elevation of the free-surface above the bed at the outlet.

***ii*) Incremental pressure-correction schemes**:

It is also possible to explicitly include the pressure gradient in the first sub-step of the algorithm in order to increase its accuracy. This corresponds to a second kind of pressure-correction schemes. The main two of them are the standard and the rotational incremental pressure-correction schemes. The standard one is built as:

$$\begin{aligned} &\frac{1}{2\delta t}\left(3\tilde{\boldsymbol{v}}^{n+1} - 4\boldsymbol{v}^n + \boldsymbol{v}^{n-1}\right) - \nu\boldsymbol{\nabla}^2\tilde{\boldsymbol{v}}^{n+1} = \boldsymbol{g} - \frac{1}{\rho}\boldsymbol{\nabla}p^n \\ &\begin{cases} \dfrac{3}{2\delta t}\left(\boldsymbol{v}^{n+1} - \tilde{\boldsymbol{v}}^{n+1}\right) + \dfrac{1}{\rho}\left(\boldsymbol{\nabla}p^{n+1} - \boldsymbol{\nabla}p^n\right) = 0 \\ \nabla\cdot\boldsymbol{v}^{n+1} = 0 \end{cases} \end{aligned} \tag{1.30}$$

with the velocity wall boundary conditions given by (1.22). Once again the pressure wall boundary condition is a homogeneous Neumann, which induces the same problems as in the previous scheme with an implicit viscous term.

The rotational incremental pressure-correction scheme is built as:

$$\begin{aligned} &\frac{1}{2\delta t}\left(3\tilde{\boldsymbol{v}}^{n+1} - 4\boldsymbol{v}^n + \boldsymbol{v}^{n-1}\right) - \nu\boldsymbol{\nabla}^2\tilde{\boldsymbol{v}}^{n+1} = \boldsymbol{g} - \frac{1}{\rho}\boldsymbol{\nabla}p^n \\ &\begin{cases} \dfrac{3}{2\delta t}\left(\boldsymbol{v}^{n+1} - \tilde{\boldsymbol{v}}^{n+1}\right) + \dfrac{1}{\rho}\boldsymbol{\nabla}\phi^{n+1} = 0 \\ \nabla\cdot\boldsymbol{v}^{n+1} = 0 \end{cases} \end{aligned} \tag{1.31}$$

where ϕ^{n+1} is a modified pressure defined as:

$$\phi^{n+1} = p^{n+1} - p^n + \mu \nabla \cdot \tilde{\boldsymbol{v}}^{n+1} \tag{1.32}$$

The pressure Poisson equation to be solved then reads:

$$\nabla^2 \phi^{n+1} = \frac{3\rho}{2\delta t} \nabla \cdot \tilde{\boldsymbol{v}}^{n+1} \tag{1.33}$$

The wall boundary conditions are still given by (1.22), but this time this yields the non-homogeneous wall boundary condition on the pressure given by (1.27).

In fact, it is possible to define a general framework to describe any pressure correction scheme, which reads:

$$\begin{cases} \dfrac{1}{\delta t}\left(\beta_q \boldsymbol{v}^{n+1} - \displaystyle\sum_{j=0}^{q-1} \beta_j \boldsymbol{v}^{n-j}\right) - \nu \boldsymbol{\nabla}^2 \tilde{\boldsymbol{v}}^{n+1} = \boldsymbol{g} - \dfrac{1}{\rho}\boldsymbol{\nabla} p^{\star,n+1} \\ \begin{cases} \dfrac{\beta_q}{\delta t}\left(\boldsymbol{v}^{n+1} - \tilde{\boldsymbol{v}}^{n+1}\right) + \dfrac{1}{\rho}\boldsymbol{\nabla}\phi^{n+1} = 0 \\ \nabla \cdot \boldsymbol{v}^{n+1} = 0 \end{cases} \\ \phi^{n+1} = p^{n+1} - p^{\star,n+1} + \chi \mu \nabla \cdot \tilde{\boldsymbol{v}}^{n+1} \end{cases} \tag{1.34}$$

In the prediction step, a qth order backward difference formula (see [11]) is used to approximate the time-derivative of the velocity, assuming the latter is continuous:

$$\frac{d\boldsymbol{v}^{n+1}}{dt} \doteq \frac{1}{\delta t}\left(\beta_q \boldsymbol{v}^{n+1} - \sum_{j=0}^{q-1} \beta_j \boldsymbol{v}^{n-j}\right) \tag{1.35}$$

where the β_q and β_j are the formula coefficients. Besides, a rth order extrapolation of the pressure, $p^{\star,n+1}$, is used in the prediction step:

$$p^{\star,n+1} = \sum_{j=0}^{r-1} \gamma_j p^{n-j} \tag{1.36}$$

On the other hand, χ is either equal to 0 or 1, depending on the kind of scheme considered: $\chi = 0$ yields standard schemes while $\chi = 1$ yields rotational schemes. The case $(q = 1, r = 0, \chi = 0)$ corresponds to the Chorin projection scheme. The case $(q = 2, r = 1, \chi = 0)$ corresponds to the standard incremental scheme. The case $(q = 2, r = 1, \chi = 1)$ corresponds to the rotational incremental scheme. It was shown that when choosing $r = q - 1$, the consistency error is of order q on the velocity in H^1 norm, while it is of order $r = q - 1$ on the pressure in L^2 norm. When choosing $r = q$, the consistency error is of order q on the velocity and the pressure (in H^1 and L^2 norm respectively). However, it was observed that with a homogeneous Neumann wall boundary condition on the pressure (1.23), the schemes do not reach these orders of convergence. They only

reach them with the non-homogeneous Neumann condition given by (1.27) (these convergence studies were done in a Finite Elements formalism, see [40]).

1.1.2.3 Velocity-correction schemes

This kind of schemes is the counterpart of the pressure-correction schemes in that the role of the pressure and viscous terms are inverted: in the first sub-step, the pressure gradient is involved while the viscous term is either ignored or treated explicitly. As in the previous case this kind of schemes declines into non-incremental and incremental schemes. Here only the non-incremental scheme is shown, which is built as:

$$\begin{cases} \dfrac{\boldsymbol{v}^{n+1}-\tilde{\boldsymbol{v}}^{n}}{\delta t} = \boldsymbol{g} - \dfrac{1}{\rho}\boldsymbol{\nabla} p^{n+1} \\ \nabla \cdot \boldsymbol{v}^{n+1} = 0 \end{cases} \qquad \frac{\tilde{\boldsymbol{v}}^{n+1}-\boldsymbol{v}^{n+1}}{\delta t} = \nu \boldsymbol{\nabla}^2 \tilde{\boldsymbol{v}}^{n+1} \tag{1.37}$$

with the same boundary conditions as before (1.22). Besides, as with pressure-correction schemes the pressure at time $n+1$ is computed through a Poisson equation (1.21). The same kind of incremental schemes as with the pressure-correction technique can be built, in a standard or rotational form. As before, only the rotational scheme leads to a consistent Neumann wall boundary condition on the pressure. As for the pressure-correction schemes, it is possible to increase the accuracy of the scheme by using a higher order backward difference formula to approximate the time-derivative of the velocity in the prediction step, together with a higher order extrapolation of the pressure. As before, the expected orders of convergence are only achieved with the non-homogeneous Neumann boundary condition given by (1.27). A more complete review of the velocity-correction schemes can be found in [40].

1.1.2.4 Consistent splitting schemes

Another kind of method, called splitting schemes, consists in computing the velocity in a first sub-step, treating the pressure explicitly, and then solve a weak form of the pressure Poisson equation. An example of such scheme reads:

$$\begin{cases} \dfrac{\boldsymbol{v}^{n+1}-\tilde{\boldsymbol{v}}^{n}}{\delta t} = \boldsymbol{g} - \dfrac{1}{\rho}\boldsymbol{\nabla} p^{n} - \nu \boldsymbol{\nabla}^2 \tilde{\boldsymbol{v}}^{n+1} \\ \left(\boldsymbol{\nabla} p^{n+1}, \boldsymbol{\nabla} q\right) = \left(\boldsymbol{g} + \nu \boldsymbol{\nabla}^2 \tilde{\boldsymbol{v}}^{n+1}, \boldsymbol{\nabla} q\right), \forall q \in H^1(\Omega) \end{cases} \tag{1.38}$$

Once again it is possible to use a qth order backward difference formula to approximate the time derivative of the velocity (1.35) in the first line of (1.38), together with a rth order extrapolation of the pressure instead of p^n (1.36). However, the scheme (1.38) leads to the homogeneous Neumann

condition on the pressure, which is artificial. Another possibility is to apply a similar technique as for the construction of a rotational pressure or velocity correction scheme. This yields consistent-splitting schemes as the one below:

$$\begin{cases} \dfrac{\boldsymbol{v}^{n+1}-\tilde{\boldsymbol{v}}^{n}}{\delta t} = \boldsymbol{g} - \dfrac{1}{\rho}\boldsymbol{\nabla} p^{n} - \nu \boldsymbol{\nabla}^{2}\tilde{\boldsymbol{v}}^{n+1} \\ \left(\boldsymbol{\nabla}\psi^{n+1}, \boldsymbol{\nabla} q\right) = \left(\boldsymbol{g} + \nu\boldsymbol{\nabla}^{2}\tilde{\boldsymbol{v}}^{n+1} - \dfrac{1}{\rho}\boldsymbol{\nabla} p^{n}, \boldsymbol{\nabla} q\right), \forall q \in H^{1}(\Omega) \\ p^{n+1} = \psi^{n+1} + p^{n} - \nu \nabla \cdot \boldsymbol{v}^{n+1} \end{cases} \tag{1.39}$$

With this kind of scheme the non-homogeneous Neumann condition on the pressure is recovered. Once again it is possible to increase the accuracy of the scheme through a backward difference formula (1.35) and a different pressure extrapolation (1.36). The framework of these splitting approaches is interesting since it makes it possible to show the existence and unicity of a solution to the pressure Poisson equation regardless of the domain topology, which is not so easy with the pressure-correction and velocity-correction schemes presented above. It has been widely used in Finite Elements.

1.2 Turbulence modelling

It is a well-known feature of the Navier–Stokes equations that they present a chaotic behaviour for sufficiently high values of the Reynolds number, $Re = \frac{UL}{\nu}$ (L being a characteristic length scale and U a characteristic velocity scale of the flow). This corresponds to the existence of turbulent flows, which are geometrically complex, rapidly varying over time and very sensitive to initial conditions. It is possible to numerically model these flows without introducing any model for turbulence: such an approach is called Direct Numerical Simulation (DNS). However, since turbulent structures reach very small scales compared to the main flow structures, this requires very fine 3-D space discretisations. Besides, it also requires very small values of the time-step size, which leads to tremendously high computational times. DNS is thus used to numerically study turbulence and to obtain results on reference cases. Though, it is not suitable to industrial applications due to its computational cost.

Other methods were thus developed in order to model the chaotic behaviour of turbulent flows without having to resolve all the flow scales. The first models introduced for turbulence rely on the Reynolds-average formalism. Later on, Large Eddy Simulation (LES) models were developed, which represent the turbulent eddies down to a certain scale and use a sub-grid model to represent the effects of the smaller eddies.

1.2.1 Reynolds-Averaged Navier–Stokes models

This kind of approach was the one chosen in this work for turbulence modelling, thus its description is quite detailed in order to introduce our notations for the following Chapters and also in order to give a clear idea about the assumptions and approximations of the $k-\epsilon$ model.

While a turbulent field is highly variable in space and time, it presents a smooth and less variable mean. This observation led to the idea of applying a statistical mean operator to the equations. For N occurrences of a flow the Reynolds average applied to a field A is defined as:

$$\overline{A} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} A^i \tag{1.40}$$

This newly defined field $\overline{A}$ is called the mean field and is a flow feature: it is not sensitive to small perturbations of the initial conditions. Considering an instantaneous field A (dropping the superscript i which referred to the instance number), it is then written as:

$$A = \overline{A} + A' \tag{1.41}$$

where A' is a fluctuating field which is different between two instances of the flow. The mean field, contrarily to the instantaneous one, is smooth and reproducible. It may be constant over time or invariant along a direction, while the instantaneous field is always time-variable and three-dimensional. Thus, an approach for modelling turbulent flows is to try to simulate the mean fields. Applying the Reynolds average operator to the Navier–Stokes equations (1.7) yields the Reynolds-Averaged Navier–Stokes (RANS) equations for incompressible flows:

$$\begin{cases} \nabla \cdot \overline{\boldsymbol{v}} = 0 \\ \dfrac{d\overline{\boldsymbol{v}}}{dt} = -\dfrac{1}{\rho}\boldsymbol{\nabla}\overline{p} + \dfrac{1}{\rho}\boldsymbol{\nabla}\cdot\left(\mu\left[\boldsymbol{\nabla}\overline{\boldsymbol{v}} + (\boldsymbol{\nabla}\overline{\boldsymbol{v}})^T\right]\right) + \boldsymbol{g} - \boldsymbol{\nabla}\cdot\boldsymbol{R} \\ \dfrac{d\overline{\boldsymbol{r}}}{dt} = \overline{\boldsymbol{v}} \end{cases} \tag{1.42}$$

The application of the Reynolds average operator to the non-linear convection term in the momentum equation (2nd term in the left-hand side of (1.2)) led to an additional stress tensor $\boldsymbol{R}$ called the Reynolds stress tensor and defined as:

$$\boldsymbol{R} = \overline{\boldsymbol{v}' \otimes \boldsymbol{v}'} \tag{1.43}$$

A transport equation on the components of the Reynolds stress tensor (called Reynolds stresses) can be obtained by subtracting (1.42) to (1.7), tensorially multiplying the result by $\boldsymbol{v}'$ and then applying the Reynolds average operator. Though, this does not close the problem since this transport equation involves new unknown terms, in particular third order moments of $\boldsymbol{v}'$. It is thus necessary to find a heuristic closure law in order to solve the system.

A first-order closure of the system consists in writing a closure law linking the second order moments of the fluctuating velocity to its first order moments, without solving the transport equation on $\boldsymbol{R}$. In other words, it relies on the construction of a behaviour law that expresses $\boldsymbol{R}$ as a function of $\overline{\boldsymbol{v}}$. Such a behaviour law was proposed by Boussinesq through a model similar to the Stokes model (1.3) for the Cauchy stress, which aims at representing the diffusion and dissipation effects of the turbulent eddies through an eddy viscosity, as well as the additional "pressure" they induce in the flow. The Boussinesq model reads:

$$\boldsymbol{R} = \frac{2}{3}k\boldsymbol{I}_d - 2\nu_T\boldsymbol{S} \tag{1.44}$$

where $\boldsymbol{S}$ is the mean strain rate tensor:

$$\boldsymbol{S} \doteqdot \overline{\boldsymbol{s}} = \frac{1}{2}\left(\boldsymbol{\nabla}\overline{\boldsymbol{v}} + (\boldsymbol{\nabla}\overline{\boldsymbol{v}})^T\right) \tag{1.45}$$

k is the kinetic energy of the fluctuating velocity field per unit mass (called turbulent kinetic energy):

$$k = \frac{1}{2}tr\boldsymbol{R} = \frac{1}{2}\overline{|\boldsymbol{v}'|^2} \tag{1.46}$$

and ν_T is the eddy viscosity. With this model the RANS equations are written as:

$$\begin{cases} \nabla \cdot \overline{\boldsymbol{v}} = 0 \\ \dfrac{d\overline{\boldsymbol{v}}}{dt} = -\dfrac{1}{\rho}\boldsymbol{\nabla}\tilde{p} + \dfrac{1}{\rho}\boldsymbol{\nabla}\cdot[2\mu_E\boldsymbol{S}] + \boldsymbol{g} \\ \dfrac{d\overline{\boldsymbol{r}}}{dt} = \overline{\boldsymbol{v}} \end{cases} \tag{1.47}$$

where $\tilde{p} = \overline{p} + \dfrac{2}{3}\rho k$ and $\mu_E = \mu + \mu_T$ is an effective viscosity, with $\mu_T = \rho\nu_T$. The remaining task is then to build a model for the turbulent kinetic energy and eddy viscosity computations. This is done starting from the transport equation on k, which is obtained by taking the trace of the transport equation on the Reynolds stresses, and reads:

$$\frac{dk}{dt} = \mathbb{P} + \nabla \cdot \boldsymbol{Q}^k - \epsilon \tag{1.48}$$

where $\mathbb{P}$ is a production term whose definition is $\mathbb{P} = -\boldsymbol{R}\!:\!\boldsymbol{S}$ [3]. By using the Boussinesq model (1.44), it can be written as[4]:

$$\mathbb{P} = \nu_T S^2 \tag{1.49}$$

with S the scalar mean rate-of-strain defined as:

$$S = \sqrt{2\boldsymbol{S} : \boldsymbol{S}} \tag{1.50}$$

[3] We recall that $\boldsymbol{A}\!:\!\boldsymbol{B} = tr(\boldsymbol{A}\boldsymbol{B}^T) = A_{ij}B_{ij}$ with the Einstein notation.

[4] Strictly speaking, the term $-\frac{2}{3}k\nabla\cdot\boldsymbol{v}$ should be taken into account for flows that are not truly incompressible.

On the other hand, ϵ corresponds to a dissipation of turbulent kinetic energy (transformed into thermal energy) due to the viscosity, defined as:

$$\epsilon = \nu \sum_i \sum_j \overline{\left(\frac{\partial v_i'}{\partial x_j}\right)^2} \tag{1.51}$$

where $\frac{\partial v_i'}{\partial x_j}$ denotes the derivative of the ith component of $\boldsymbol{v}'$ with respect to the jth coordinate. Although it is possible to write an exact transport equation on ϵ, the latter includes complex terms that cannot be explicitly calculated. This is why, in the $k-\epsilon$ model ϵ is computed through a simplified equation that reproduces the k equation (see equation (1.54) below).

In (1.48), $\boldsymbol{Q}^k$ is the flux of k, which represents a transport of kinetic and potential energy by the eddies and the molecular viscosity. The flux of k can be modelled through a diffusion term:

$$\nabla \cdot \boldsymbol{Q}^k = \frac{1}{\rho} \nabla \cdot (\mu_k \boldsymbol{\nabla} k) \tag{1.52}$$

where μ_k is defined as $\mu_k = \mu + \dfrac{\mu_T}{\sigma_k}$, σ_k being a model constant.

Then, a variety of models is available to compute the eddy viscosity, from the simplest and coarsest (zero equation mixing length model) to more complex ones like the $k-\epsilon$ model. The latter was chosen in this work for its simplicity and wide use in the industry. It is a two-equation model, which means k and ϵ are computed through transport equations. The Kolmogorov dimensional analysis [63] leads to a definition of the eddy viscosity as a function of k and ϵ, which corresponds to the fact that the large turbulent eddies are the ones that most interact with the mean flow. ν_T is thus written as proportional to the length scale of the large eddies, $L_t \sim \frac{k^{3/2}}{\epsilon}$, which yields:

$$\nu_T = C_\mu \frac{k^2}{\epsilon} \tag{1.53}$$

C_μ is the Prandlt-Kolmogorov constant which value was determined through experiments. The transport equation on k is given by (1.48), while the dissipation ϵ is computed through a similar equation:

$$\frac{d\epsilon}{dt} = \frac{\epsilon}{k}\left(C_{\epsilon_1}\mathbb{P} - C_{\epsilon_2}\epsilon\right) + \frac{1}{\rho}\nabla \cdot (\mu_\epsilon \boldsymbol{\nabla}\epsilon) \tag{1.54}$$

where μ_ϵ is defined as: $\mu_\epsilon = \mu + \frac{\mu_T}{\sigma_\epsilon}$, σ_ϵ being a constant. C_{ϵ_1} and C_{ϵ_2} are also constants of the model. All the model constants are given in Table 1.1. Equation (1.54) has no theoretical background but relies on empirical considerations. The term $\frac{\epsilon}{k}$ ensures the equation homogeneity and the source terms are supposed proportional to the ones in the k equation. It is the frequency of the large eddies.

Note that the $k-\epsilon$ model is not accurate concerning non-inertial and streamline curvature effects, as well as severe deviation from local equilibrium. Besides, it was shown that computing the production term through (1.49) leads to over-estimations of k and thus of ν_T. In order to avoid this,

Table 1.1: Values of the $k-\epsilon$ model constants [69]

κ	C_μ	C_{ϵ_1}	C_{ϵ_2}	σ_k	σ_ϵ	Pr_T
0.41	0.09	1.44	1.92	1.0	1.3	0.85

Guimet & Laurence [41] proposed to restrict the production term to a linear behaviour for high values of the rate-of-strain, obtained from the equilibrium between $\mathbb{P}$ and ϵ for fully developed homogeneous turbulence. This yields a linear-quadratic model for the production[5]:

$$\mathbb{P} = \min\left(\sqrt{C_\mu}kS, \nu_T S^2\right) \tag{1.55}$$

Another issue is that the size of the large eddies, given by $L_t \sim \frac{k^{3/2}}{\epsilon}$, may be predicted arbitrarily large, which is not physical since L_t should be bounded at least by L, the characteristic size of the flow. To remedy this issue, Yap [156] proposed a modification of the C_{ϵ_2} coefficient in order to increase the dissipation of turbulent kinetic energy:

$$C_{\epsilon_2,Y} = \max\left(C_{\epsilon_2} - \max\left[0, 0.83\left(\frac{L_t}{L}-1\right)\left(\frac{L_t}{L}\right)^2\right], 0\right) \tag{1.56}$$

The Boussinesq model used to close the equations (1.44) establishes a linear relation between $\boldsymbol{R}$ and $\boldsymbol{S}$. It is also possible to use a non-linear model (see *e.g.* [113]). This still corresponds to a first-order closure. On the other hand, a second-order closure of the system consists in writing a closure law linking the third order moments (and other unknown terms in the governing equation for $\boldsymbol{R}$) of the fluctuating velocity to its second order moments and solving the transport equation on the second moments (*i.e.* the Reynolds stresses). Such models are called Reynolds Stress Transport Models (RSTM) [68].

For a weakly-compressible flow the density fluctuations are restricted such that ρ and $\overline{\rho}$ are assumed to be equal and applying the Reynolds average operator to the continuity equation of (1.5) gives:

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\overline{\boldsymbol{v}} \tag{1.57}$$

This is the only difference compared to the truly incompressible model. Indeed, in what was written above a possibly varying density field was considered. Note, however, that the $k-\epsilon$ model takes a more complicated shape for highly compressible flows [99].

The presence of walls in turbulent flows makes the latter anisotropic and increases the production of turbulence due to shearing effects. Modelling near-wall turbulence is crucial in order to correctly reproduce the flows, since the no-slip condition leads to large values of the velocity gradient at

[5] This essentially recovers the SST modification [96], although it does not include a low-Reynolds treatment.

the walls, which generates turbulence. Let us denote by y^+ the dimensionless distance to a wall, defined as:

$$y^+ = \frac{yu_*}{\nu} \tag{1.58}$$

with y the wall normal coordinate and u_* a friction velocity:

$$u_*^2 = \nu \left. \frac{dv_\tau}{dy} \right|_{y=0} \tag{1.59}$$

where v_τ is the wall tangential velocity component. The observation of the turbulent flow between two horizontal parallel plane walls (this configuration is called the plane Poiseuille channel) led to a sub-division of the near-wall region into three areas [152]:

- the viscous sub-layer: $0 < y^+ < 8$
- the buffer layer: $8 < y^+ < 30$
- the inertial sub-layer: $30 < y^+ < 0.2e^+$

where e^+ is the dimensionless half-height of the channel, defined by $e^+ = \frac{eu_*}{\nu}$ with e the half-height. The turbulence is negligible in the viscous sub-layer while the viscous effects are small in the inertial sub-layer. In the latter, the velocity profile distribution along the normal to the wall follows a logarithmic law, so that this zone is also called the logarithmic zone.

Directly simulating near-wall turbulence requires very fine meshes and modified turbulence models (low-Reynolds-number turbulence models). Then, the computational points closest to the wall must be located in the viscous sub-layer. This is computationally expensive, especially for flows with high-Reynolds numbers. This led to the development of wall functions, based on semi-empirical formulae, which are used to reproduce near-wall effects with coarser discretisations. This corresponds to high-Reynolds-number turbulence models and requires the computational points closest to the wall to be located in the inertial layer.

In Eulerian models, this can be done by designing the mesh so that the first calculation point is in the logarithmic zone. Another possibility is to solve the discretised equations on a 'classical' mesh, where the nodes located on the wall are treated in the same way as if they were shifted in the normal direction so as to be in the logarithmic zone, as for instance in [66]. This makes it possible to resolve the region with more than one or two points. This may also be done in a Lagrangian framework. The velocity field is then set at the wall in order to have a value for the wall shear-stress that makes it possible to reproduce a turbulent plane Poiseuille flow case. In a Lagrangian framework, this can be done by defining an Eulerian mean velocity, whose tangential component takes non-zero values at the walls. This velocity field only serves to compute the rate-of-strain tensor and the viscous forces. At the wall, the shear-stress vector is thus set through:

$$\boldsymbol{\tau} = \mu_E \frac{\partial \overline{\boldsymbol{u}}}{\partial \boldsymbol{n}} = -\rho u_*^2 \frac{\overline{\boldsymbol{u}}}{|\overline{\boldsymbol{u}}|} \tag{1.60}$$

where $\overline{\boldsymbol{u}}$ is the Eulerian mean velocity and u_* is assumed to satisfy the logarithmic law (considering a smooth velocity profile):

$$\frac{v_\tau}{u_*} = \frac{1}{\kappa}\ln\left(\frac{y u_*}{\nu}\right) + 5.2 \tag{1.61}$$

where κ is the von Kármán constant (see Table 1.1). Recall that y is the distance to the wall. u_* may then be computed through an iterative process. The following wall functions can be deduced from the equilibrium $\mathbb{P} = \epsilon$ in the logarithmic zone (recall this holds for fully developed turbulence):

$$\begin{cases} k|_{\partial\Omega_w} = \left.\dfrac{u_*^2}{\sqrt{C_\mu}}\right|_{\partial\Omega_w} \\ \epsilon|_{\partial\Omega_w} = \left.\dfrac{u_*^3}{\kappa y}\right|_{\partial\Omega_w} \end{cases} \tag{1.62}$$

However, it is more recommended to impose Neumann boundary conditions on k and ϵ instead of these Dirichlet conditions, in order to avoid coupling the boundary conditions of the momentum equation and the k and ϵ equations. Indeed, imposing the Dirichlet boundary conditions (1.62) makes the boundary values of k and ϵ depend solely on u_*, thus on $\overline{\boldsymbol{u}}$. The normal derivatives of k and ϵ can be derived from (1.62):

$$\begin{cases} \boldsymbol{\nabla} k \cdot \boldsymbol{n}|_{\partial\Omega_w} = 0 \\ \boldsymbol{\nabla} \epsilon \cdot \boldsymbol{n}|_{\partial\Omega_w} = \left.-\dfrac{u_*^3}{\kappa y^2}\right|_{\partial\Omega_w} \end{cases} \tag{1.63}$$

There exists many variants of the wall functions, see for example [14, 28, 66, 70]. More details about the wall boundary conditions in turbulence models can also be found in [13].

The inflow boundary conditions on k and ϵ read:

$$\begin{cases} k|_{\partial\Omega_i} = k_i \\ \epsilon|_{\partial\Omega_i} = \epsilon_i \end{cases} \tag{1.64}$$

where k_i and ϵ_i are imposed values of these fields, which may be set as:

$$\begin{cases} k_i = \dfrac{3}{2}\left(\overline{u}I\right)^2 \\ \epsilon_i = C_\mu \dfrac{k^{3/2}}{l_0} \end{cases} \tag{1.65}$$

with l_0 the mixing length, which takes similar values as L_t, the size of the large eddies, and is set by the user[6], and I the turbulence intensity, which can be obtained from experiments and is usually taken equal to $0.16Re^{-1/8}$ for duct flows.

[6]For instance in a Smoothed Particle Hydrodynamics simulation it may be set as the kernel support size (see section 2.2.1).

On the other hand, the outflow boundary conditions on k and ϵ read:

$$\begin{cases} \boldsymbol{\nabla} k \cdot \boldsymbol{n}|_{\partial\Omega_o} = 0 \\ \boldsymbol{\nabla} \epsilon \cdot \boldsymbol{n}|_{\partial\Omega_o} = 0 \end{cases} \tag{1.66}$$

Finally, the free-surface boundary conditions to be applied to k is a homogeneous Neumann (no flux of energy in the absence of wind):

$$\boldsymbol{\nabla} k \cdot \boldsymbol{n}|_{\partial\Omega_\eta} = 0 \tag{1.67}$$

while some authors recommend to link the value of ϵ with that of k at the free-surface through [107]:

$$\epsilon|_{\partial\Omega_\eta} = \left. \frac{k^{3/2}}{\alpha_\epsilon H} \right|_{\partial\Omega_\eta} \tag{1.68}$$

with $\alpha_\epsilon = 0.18$ a constant and H the water depth. Although this condition is not applicable in case of flows presenting complex free-surface shapes like a breaking wave, so that in this work the free-surface condition imposed on ϵ can be assumed to be a homogeneous Neumann condition, as a first approximation. Note that the imposition of free-surface boundary conditions on ϵ in general is still an open question.

1.2.2 Large Eddy Simulation

The development of LES in SPH was not the topic of the present work, though it is mentioned here due to its significant importance in turbulence modelling. Indeed, in many industrial and environmental cases it is necessary to obtain the fluctuating fields, which cannot be achieved with RANS models, sophisticated as they may be. As mentioned above, the LES technique represents the turbulent eddies down to a certain scale and uses a subgrid-scale model to represent the effects of the smaller eddies. To do so, considering an incompressible flow the velocity field is filtered according to:

$$\tilde{\boldsymbol{v}} = \int_\Omega \boldsymbol{v}(\boldsymbol{r}', t) G_\Delta(\boldsymbol{r}, \boldsymbol{r}') d\boldsymbol{r}' \tag{1.69}$$

where $\tilde{\boldsymbol{v}}$ is the filtered velocity field[7] and G_Δ is a filter function, chosen so that it behaves like a low-pass filter, allowing to keep only the largest turbulent structures. The filter function may take the shape of a Gaussian or of a rectangular function, its characteristic size Δ corresponding to the size of the smallest modelled structures. Δ is of the order of magnitude of the spatial resolution. The filter function satisfies a normalisation condition:

$$\forall \boldsymbol{r}, \int_\Omega G_\Delta(\boldsymbol{r}, \boldsymbol{r}') d\boldsymbol{r}' = 1 \tag{1.70}$$

[7] Here $\tilde{\boldsymbol{v}}$ should not be confused with the estimated velocity in the projection methods of section 1.1.2.

The characteristic time of the smallest simulated structures being higher than in a DNS, the time-step size can be increased compared to the latter, which also reduces computational times. On the other hand, LES still requires 3-D simulations in order to be consistent.

Although the filtering operator does not present the same properties as the Reynolds-average operator, it is possible to show that the equation of motion of the filtered fields can be written as:

$$\frac{d\tilde{\boldsymbol{v}}}{dt} = -\frac{1}{\rho}\boldsymbol{\nabla}\tilde{p} + \nu\nabla^2\tilde{\boldsymbol{v}} - \boldsymbol{\nabla}\cdot\boldsymbol{\tau}_R + \boldsymbol{g} \tag{1.71}$$

where $\tilde{p}$ is the filtered pressure field and $\boldsymbol{\tau}_R$ is a tensor representing the impact of the subgrid scale structures on the filtered field. A model for this tensor is required in order to close the equations, which is usually an eddy viscosity model. Indeed, taking Δ in the range of medium-sized eddies the turbulent structures can be considered as isotropic and in quasi-equilibrium. Thus, a similar model than the Boussinesq one (1.44) can be used with a subgrid viscosity computed according to a mixing length model. Several models are then available to find the length to be used for the subgrid viscosity, like the Smagorinsky model [131], or more complex ones [114]. Note that near-wall turbulence also needs to be modelled in LES, which is done in a similar way as in the RANS models on the filtered velocity field.

1.3 Buoyancy modelling

Many industrial and environmental flows involve fluids which density varies as a function of the temperature or of a scalar concentration like salinity. In many of these flows the Mach number is low (< 0.3) so that they are weakly-compressible (the variations of density due to velocity variations can be neglected as a first approximation). Such flows are subject to buoyancy effects due to gravity, which may generate density currents and stratifications. Besides, in most cases they are turbulent. It is then important for numerical models to represent the buoyancy effects in combination with turbulence effects. There are important differences in terms of vocabulary between flows where the active scalar is the temperature and where it is a scalar concentration. In order to avoid introducing too many notations, the framework of non-isothermal flows was chosen in this work, although the model also applies to other active scalars. In this section the diffusion equation on the temperature is derived from the energy equation on the enthalpy and the boundary conditions necessary for the closure of the system are described. Then, the effects of buoyancy on the equation of motion and on the k and ϵ equations are described.

1.3.1 Diffusion equation on the temperature

Let h be the enthalpy defined in thermodynamics through $h = e + \frac{p}{\rho}$, e being the internal energy per unit mass. It is possible to show [151] through energy balances that h satisfies the following

equation:

$$\rho\frac{dh}{dt} = -\nabla\cdot\boldsymbol{q} + \frac{dp}{dt} + \boldsymbol{\tau}:\boldsymbol{s} \tag{1.72}$$

where $\boldsymbol{q}$ is the heat flux vector, given by the Fourier law of heat conduction:

$$\boldsymbol{q} = -\lambda_T\boldsymbol{\nabla}T \tag{1.73}$$

T being the temperature and λ_T the thermal conductivity. On the other hand, $\boldsymbol{\tau}$ was defined in (1.3) and $\boldsymbol{s}$ in (1.4), and $\boldsymbol{\tau}:\boldsymbol{s}$ represents the dissipation of mechanical energy. In thermodynamics the variation of h is expressed as a function of the variations of temperature and of pressure through:

$$dh = C_p dT + \frac{1}{\rho}(1-\beta T)dp \tag{1.74}$$

where β is the thermal expansion coefficient defined by:

$$\beta = -\frac{1}{\rho}\left.\frac{\partial\rho}{\partial T}\right|_p \tag{1.75}$$

Substituting (1.74) into (1.72) then gives:

$$\rho C_p\frac{dT}{dt} = \nabla\cdot(\lambda_T\boldsymbol{\nabla}T) + \beta T\frac{dp}{dt} + \boldsymbol{\tau}:\boldsymbol{s} \tag{1.76}$$

For low-velocity flows, the terms $\beta T\frac{dp}{dt}$ and $\boldsymbol{\tau}:\boldsymbol{s}$ can be neglected before the others so that this equation becomes:

$$\rho C_p\frac{dT}{dt} = \nabla\cdot(\lambda_T\boldsymbol{\nabla}T) \tag{1.77}$$

If λ_T is constant, which is generally valid when $\frac{\delta\rho}{\rho} << 1$, this equation is written as:

$$\frac{dT}{dt} = K\nabla^2 T \tag{1.78}$$

with $K = \frac{\lambda_T}{\rho C_p}$ the thermal diffusivity. This equation of diffusion on the temperature must be solved additionally to the Navier–Stokes equations. Note that $\frac{dT}{dt}$ implicitly includes the fluid velocity since $\frac{dT}{dt} = \frac{\partial T}{\partial t} + \boldsymbol{v}\cdot\boldsymbol{\nabla}T$, so that equation (1.78) is coupled to the momentum equation.

When a RANS approach is used, for an incompressible flow the Reynolds-averaging of equation (1.77) yields:

$$\frac{d\overline{T}}{dt} = \nabla\cdot(K\boldsymbol{\nabla}\overline{T}) - \nabla\cdot(\overline{\boldsymbol{v}'T'}) \tag{1.79}$$

where $\overline{\boldsymbol{v}'T'}$ is the turbulent heat flux. The same kind of model as for the Reynolds stresses (as well as fluxes of k and ϵ) can be used, namely a turbulent thermal diffusivity model, defining:

$$\overline{\boldsymbol{v}'T'} = -K_T\boldsymbol{\nabla}\overline{T} \tag{1.80}$$

thus assuming that the turbulent heat flux is aligned with the mean temperature gradient. K_T is the turbulent thermal diffusivity. Then, the diffusion equation on the temperature reads:

$$\frac{d\overline{T}}{dt} = \nabla \cdot (K_E \boldsymbol{\nabla} \overline{T}) \tag{1.81}$$

where an effective thermal diffusivity $K_E = K + K_T$ was defined. K_T is usually taken as proportional to the eddy viscosity, the ratio of the two being (by definition) the turbulent Prandlt number:

$$Pr_T = \frac{\nu_T}{K_T} \tag{1.82}$$

Although the latter is not constant in a flow, neither universal, it is often taken as constant in CFD codes for the sake of simplicity. The value used for Pr_T in this work is given in Table 1.1.

At solid walls the boundary conditions applied to the temperature in laminar mode can be of Neumann type (*e.g.* adiabatic wall, imposed heat flux) or Dirichlet type (*e.g.* isothermal wall). With a $k - \epsilon$ model it is necessary to impose a wall function on the temperature since the temperature gradients close to the walls are large in turbulent mode, which generates turbulence as in the case of the velocity gradients. Considering a 1-D fully developed flow field and thermal field in a channel, the integration of the temperature equation along the normal to the wall, from the wall to the centre of the channel reads:

$$-Q_w = K_E \frac{d\overline{T}}{dy} \tag{1.83}$$

where y is the normal distance to the wall and Q_w the heat flux applied at the wall. Integrating once more yields:

$$\int_{T_w}^{\overline{T}} d\overline{T} = -Q_w \int_0^y \frac{dy}{K_E} \tag{1.84}$$

where T_w is the wall temperature. Defining the dimensionless variable:

$$T^+ = \frac{(T_w - \overline{T})u_*}{Q_w} \tag{1.85}$$

equation (1.84) can be written as:

$$T^+ = \int_0^{y^+} \frac{\nu dy^+}{K_E} \tag{1.86}$$

where y^+ is defined by (1.58). The integration of this equation can be done assuming a decomposition of the near-wall region into a laminar layer where T^+ varies linearly with y^+ and a turbulent layer where it follows a logarithmic law, as in [13]. It is also possible to use a three-layers model

(see [29]) through:

$$\begin{cases} T^+ = Pr\, y^+ & \text{if } y^+ < y_1^+ \\ T^+ = a_2 - \dfrac{Pr_T}{2a_1 \left(y^+\right)^2} & \text{if } y_1^+ \leq y^+ < y_2^+ \\ T^+ = \dfrac{Pr_T}{\kappa} \ln y^+ + a_3 & \text{if } y^+ > y_2^+ \end{cases} \tag{1.87}$$

where the following constants were defined:

$$\begin{cases} y_1^+ = \left(\dfrac{a_4}{Pr}\right)^{1/3} \\ y_2^+ = \sqrt{\dfrac{a_4 \kappa}{Pr_T}} \\ a_1 = \dfrac{Pr_T}{a_4} \\ a_2 = 15 Pr^{2/3} \\ a_3 = 15 Pr^{2/3} - \dfrac{Pr_T}{2\kappa} \left(1 + \ln \dfrac{a_4 \kappa}{Pr_T}\right) \\ a_4 = 1000 \end{cases} \tag{1.88}$$

Recall that κ is defined in the Table 1.1. Finally, $Pr = \frac{\nu}{K}$ is the molecular Prandlt number.

At the free-surface a homogeneous Neumann condition is imposed (no heat-flux). On the other hand, at inflow boundaries a Dirichlet condition is set on the temperature, whereas at outflow boundaries a homogeneous Neumann condition is prescribed (like for k and ϵ).

1.3.2 Buoyancy effects in the momentum equation

The density variations in buoyant flows mainly affect the flow dynamics through the gravity term. In a numerical model one possibility is to let the density vary according to equation (1.75). Then the expression of the momentum equation is not modified but care must be taken when solving the Navier–Stokes equations that the density is a varying quantity. With such a model a weakly-compressible formalism must be adopted since the continuity equation in the one of (1.5). The equation of state is then modified since the pressure depends on the temperature, besides the density.

An alternative approach is to apply the so-called Boussinesq approximation for flows where $\frac{\delta\rho}{\rho} << 1$[8], which enables the treatment of buoyancy affecting the fluid motion by means of the gravity term only. Then, the fluid density is considered as constant. This framework was the one chosen

[8]The upper limit for the Boussinesq approximation validity is considered in [151] to be $\frac{\delta\rho}{\rho} < 0.1$.

in this work for the sake of simplicity. The Navier–Stokes equations then read:

$$\begin{cases} \nabla \cdot \overline{\boldsymbol{v}} = 0 \\ \dfrac{d\overline{\boldsymbol{v}}}{dt} = -\dfrac{1}{\rho}\boldsymbol{\nabla} p + \dfrac{1}{\rho}\boldsymbol{\nabla} \cdot \left(\mu_E \left[\boldsymbol{\nabla}\overline{\boldsymbol{u}} + (\boldsymbol{\nabla}\overline{\boldsymbol{u}})^T\right]\right) + \left[1 - \beta(\overline{T} - T_0)\right]\boldsymbol{g} \\ \dfrac{d\overline{\boldsymbol{r}}}{dt} = \overline{\boldsymbol{v}} \end{cases} \tag{1.89}$$

where T_0 is the mean temperature of the flow. Recall $\boldsymbol{u}$ is now used instead of $\boldsymbol{v}$ in the viscous force, as said in section 1.2.1 when presenting the turbulent wall boundary conditions.

In the RANS formalism, when deriving the Reynolds stress equation a new term $\mathbb{G} = -\beta \boldsymbol{g} \cdot \overline{\boldsymbol{v}'T'}$ appears, which thus also appears in the k equation. This term is modelled through equation (1.80) which yields the following modified equations on k and ϵ:

$$\begin{cases} \dfrac{dk}{dt} = \mathbb{P} + \mathbb{G} - \epsilon + \dfrac{1}{\rho}\nabla \cdot (\mu_k \boldsymbol{\nabla} k) \\ \dfrac{d\epsilon}{dt} = \dfrac{\epsilon}{k}\left(C_{\epsilon_1}\mathbb{P} + C_{\epsilon_3}\mathbb{G} - C_{\epsilon_2}\epsilon\right) + \dfrac{1}{\rho}\nabla \cdot (\mu_\epsilon \boldsymbol{\nabla}\epsilon) \end{cases} \tag{1.90}$$

where $\mathbb{G}$ is now defined through:

$$\mathbb{G} = \beta K_T \boldsymbol{\nabla}\overline{T} \cdot \boldsymbol{g} \tag{1.91}$$

In the equation on ϵ, the constant C_{ϵ_3} was introduced in order to represent the fact that stable stratifications weaken turbulence. It is thus taken as equal to one if $\mathbb{G}$ is negative, and zero otherwise.

Note that the other constants of the $k - \epsilon$ model and the wall functions are considered as unaffected by the temperature variations, which is questionable, even in the frame of the Boussinesq approximation.

1.4 System of equations to be solved and associated set of boundary conditions

In the subsequent Chapters the overbar referring to the mean fields in turbulent mode is dropped for the sake of simplicity. Nevertheless, the reader should bear in mind that all the resolved equations refer to the mean fields when the $k - \epsilon$ turbulence closure is used. The system of equations to be

solved reads:

$$\begin{cases} \nabla \cdot \boldsymbol{v} = 0 \\ \dfrac{d\boldsymbol{v}}{dt} = -\dfrac{1}{\rho}\boldsymbol{\nabla}\tilde{p} + \dfrac{1}{\rho}\boldsymbol{\nabla}\cdot\left(\mu_E\left[\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^T\right]\right) + \left[1-\beta(T-T_0)\right]\boldsymbol{g} \\ \dfrac{d\boldsymbol{r}}{dt} = \boldsymbol{v} \\ \dfrac{dk}{dt} = \mathbb{P} + \mathbb{G} - \epsilon + \dfrac{1}{\rho}\nabla\cdot(\mu_k\boldsymbol{\nabla}k) \\ \dfrac{d\epsilon}{dt} = \dfrac{\epsilon}{k}\left(C_{\epsilon_1}\mathbb{P} + C_{\epsilon_3}\mathbb{G} - C_{\epsilon_{2,Y}}\epsilon\right) + \dfrac{1}{\rho}\nabla\cdot(\mu_\epsilon\boldsymbol{\nabla}\epsilon) \\ \dfrac{dT}{dt} = K_E\nabla^2 T \end{cases} \tag{1.92}$$

Recall that $\boldsymbol{v}$ is the Lagrangian velocity used to move the particles while $\boldsymbol{u}$ is an Eulerian velocity used to better represent the near-wall turbulence. The $k-\epsilon$ model constants are given in Table 1.1 (p.19) and the following variables were defined:

$$\begin{cases} \tilde{p} = p + \dfrac{2}{3}\rho k \\ \mu_E = \mu + \mu_T, \;\; \mu_T = \rho C_\mu \dfrac{k^2}{\epsilon}, \;\; \mu_k = \mu + \dfrac{\mu_T}{\sigma_k}, \;\; \mu_\epsilon = \mu + \dfrac{\mu_T}{\sigma_\epsilon} \\ \mathbb{P} = \min\left(\sqrt{C_\mu}kS, \nu_T S^2\right), \;\; S = \sqrt{2\boldsymbol{S}:\boldsymbol{S}}, \;\; \boldsymbol{S} = \frac{1}{2}\left(\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^T\right) \\ K_E = K + K_T, \;\; K_T = \dfrac{\mu_T}{\rho Pr_T}, \;\; \mathbb{G} = \beta K_T \boldsymbol{\nabla} T\cdot\boldsymbol{g} \end{cases} \tag{1.93}$$

Besides, the Yap correction is applied:

$$C_{\epsilon_{2,Y}} = C_{\epsilon_2} - \max\left[0, 0.83\left(\frac{L_t}{L}-1\right)\left(\frac{L_t}{L}\right)^2\right] \tag{1.94}$$

with $L_t \doteqdot \dfrac{k^{3/2}}{\epsilon}$ and L the characteristic length of the flow. The set of boundary conditions associated to these equations is summarised in Table 1.2. Recall that $p^* = p + \rho g z$, y is the normal distance to a wall and u_* is a friction velocity computed through equation (1.61).

Table 1.2: Summary of the boundary conditions imposed on the fields at the solid walls $\partial\Omega_w$, the free-surface $\partial\Omega_\eta$, the inflow $\partial\Omega_i$ and outflow $\partial\Omega_o$ boundaries.

Field \ Location	Walls $\partial\Omega_w$	Free-surface $\partial\Omega_\eta$	Inlet $\partial\Omega_i$	Outlet $\partial\Omega_o$
$\boldsymbol{v}$	$\boldsymbol{v}\vert_{\partial\Omega_w} = \boldsymbol{v}_w$	$\left(\frac{\partial z}{\partial t} + \boldsymbol{v}\cdot\boldsymbol{n}\right)_{\partial\Omega_\eta} = 0$ $\boldsymbol{\tau}\cdot\boldsymbol{n}\vert_{\partial\Omega_\eta} = 0$	$\boldsymbol{v}\vert_{\partial\Omega_i} = \boldsymbol{v}_i$	$\left.\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{n}}\right\vert_{\partial\Omega_o} = 0$
p	$\frac{\partial}{\partial \boldsymbol{n}}\left(\frac{v^2}{2}+\frac{p^*}{\rho}\right)_{\partial\Omega_w} = \left(\nu\boldsymbol{\nabla}^2\boldsymbol{v}\right)_{\partial\Omega_w}\cdot\boldsymbol{n}$	$p\vert_{\partial\Omega_\eta} = 0$	$\left.\frac{\partial p}{\partial \boldsymbol{n}}\right\vert_{\partial\Omega_i} = 0$	$p\vert_{\partial\Omega_o} = p_o$
k	$\boldsymbol{\nabla}k\cdot\boldsymbol{n}\vert_{\partial\Omega_w} = 0$	$\left.\frac{\partial k}{\partial \boldsymbol{n}}\right\vert_{\partial\Omega_\eta} = 0$	$k\vert_{\partial\Omega_i} = \left.\frac{3}{2}\left(\overline{u}I\right)^2\right\vert_{\partial\Omega_i}$	$\left.\frac{\partial k}{\partial \boldsymbol{n}}\right\vert_{\partial\Omega_o} = 0$
ϵ	$\boldsymbol{\nabla}\epsilon\cdot\boldsymbol{n}\vert_{\partial\Omega_w} = \left.-\frac{u_*^3}{\kappa y^2}\right\vert_{\partial\Omega_w}$	$\left.\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right\vert_{\partial\Omega_\eta} = 0$	$\epsilon\vert_{\partial\Omega_i} = \left.C_\mu\frac{k^{3/2}}{l_0}\right\vert_{\partial\Omega_i}$	$\left.\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right\vert_{\partial\Omega_o} = 0$
T	$T\vert_{\partial\Omega_w}$ or $\left.\frac{\partial T}{\partial \boldsymbol{n}}\right\vert_{\partial\Omega_w}$ imposed (T_w imposed through (1.87) in turbulent regime)	$\left.\frac{\partial T}{\partial \boldsymbol{n}}\right\vert_{\partial\Omega_\eta} = 0$	$T\vert_{\partial\Omega_i} = T_i$	$\left.\frac{\partial T}{\partial \boldsymbol{n}}\right\vert_{\partial\Omega_o} = 0$

Chapter 2

Smoothed Particle Hydrodynamics: Literature review

Ce Chapitre a pour objet une présentation de l'état de l'art concernant la méthode Smoothed Particle Hydrodynamics (SPH), en particulier dans son application au système d'équations auquel le Chapitre 1 a abouti, avec les conditions aux limites envisagées. Dans un premier temps, les possibilités en ce qui concerne la discrétisation spatiale des équations sont détaillées. Ensuite, leur discrétisation temporelle est abordée, avec la possibilité d'avoir des schémas quasi-incompressibles ou incompressibles. Les techniques existantes pour la représentation des conditions aux limites sont alors passées en revue, en ce qui concerne les parois solides, les surfaces libres et les frontières ouvertes. Les modèles de turbulence et de flottabilité ayant été développés pour SPH sont ensuite présentés, avant de finir par une synthèse de ce qui a été fait en termes de parallélisme massif pour les algorithmes SPH depuis les années 2000. L'objectif de ce Chapitre est de mettre en relief les techniques existantes ayant été utilisées dans la construction du modèle développé dans ce travail, qui sera présenté au Chapitre suivant.

As mentioned in the Introduction, the Smoothed Particle Hydrodynamics (SPH) method is a Lagrangian method for fluid flow simulation. In SPH the continuous medium is discretised into a set of particles, which are interpolation points to which physical quantities are associated (velocity, density, pressure, etc.). These variables fulfil a set of discrete differential equations, which are solved using a time discretisation and defining space-discretised differential operators. We will see in section 2.2 that in SPH the particles interactions depend on their mutual distances and on the physical quantities they carry. The computed velocity of the particles is then used to move them, which corresponds to a new distribution of the interpolation points and to the convection of the physical quantities. In this Chapter an overview of SPH is provided, mainly focused on its application to the resolution of the Navier–Stokes equations. It is mainly based on the 2005 paper by Monaghan [103] and on the book *Fluid Mechanics and the SPH Method* [147].

2.1 Introduction to SPH

The SPH method was created in the late 70's in the field of Astrophysics by Lucy [80] and by Monaghan and Gingold [102]. Their aim was to model non-axisymmetric problems in unbounded media that may undergo very large stretching. Compared to classical Eulerian methods, in this field SPH presents several advantages such as the possibility to model highly distorted media and to avoid building a mesh for the entire computational domain, much of which being often empty and devoid of fluid. Moreover, the framework of SPH, which relies on particles interactions, is well adapted to including complex physics quite easily. SPH made it possible to model violent phenomena in which matter is highly distorted, possibly non-axisymmetric and involving non-linear interactions between particles. Phenomena such as galaxies colliding, star formations, supernova explosions, etc. were modelled with SPH. For example, Figure 2.1 shows pictures of an SPH simulation of merging galaxies [89]. In this field, ongoing research relative to SPH aims at modelling phenomena such as planet formations, solar system formations from dust and gas clouds, or the electromagnetic interactions between colliding celestial bodies.

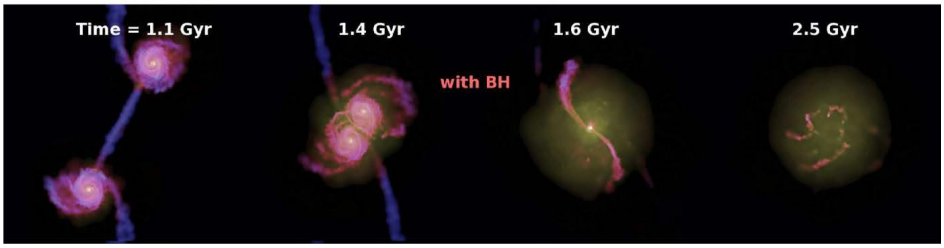


Figure 2.1: SPH modelling of the fusion of two spiral galaxies presenting central black holes [89].

Besides astrophysics, the method was applied to solid mechanics to model shocks and fracture [57]. Such phenomena are key-issues in solid mechanics, that are met with in several industrial sectors.

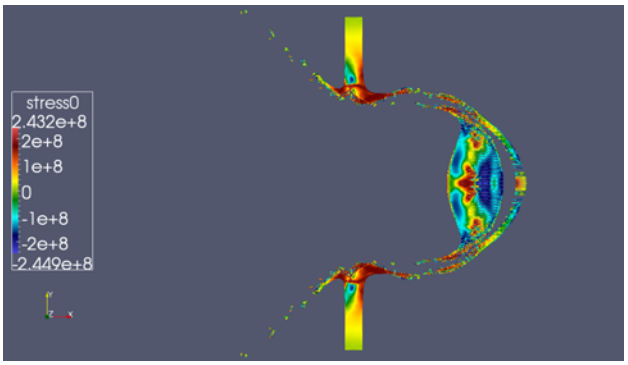


Figure 2.2: Snapshot of an SPH simulation of a 2D shock at very high-speed between an aluminium circular projectile and a thin aluminium plate [17].

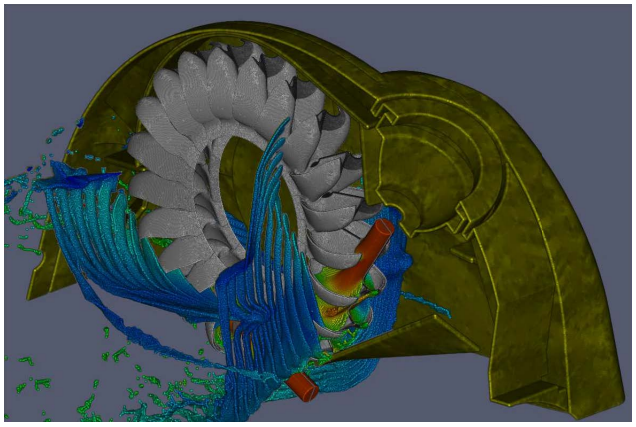


Figure 2.3: Complete modelling of a 2-jets horizontal Pelton turbine (colours represent the velocity) [84].

They involve very high distorsion of the continuous medium, which makes their modelling through mesh-based methods problematic. Figure 2.2 shows an example of shock modelling with SPH [17].

In the field of fluid mechanics, SPH is also a promising method. To this day, it has been used in the industry to simulate complex free-surface flows, most of time involving moving objects, where Eulerian methods can struggle or perform poorly. For example, Figure 2.3 shows a snapshot of a 3-D simulation of the flow around a Pelton turbine with SPH [84]. In fluid dynamics, SPH was mostly applied to the resolution of the Navier–Stokes equations, although some authors applied it to the resolution of the shallow-water equations (see *e.g.* [144, 158]). In the following sections, a literature review of SPH is provided regarding its application to the resolution of the Navier–Stokes equations.

2.2 Classical SPH interpolation and differential operators

2.2.1 SPH interpolation

2.2.1.1 Construction of the SPH interpolation

The SPH interpolation corresponds to an estimation of the density from an arbitrary distribution of point mass particles, where a local sampling of the mass distribution is done (in a sphere centred around the sampling point) and where the density estimate is smoothed. Using a local sampling instead of a global one (*e.g.* a fixed mesh as in the Marker-In-Cell [43] or Particle-In-Cell methods [50]) results in higher accuracy, while the smoothing reduces the noise in the density estimation. This technique makes it possible to estimate the value of a field at any point of space, based on the value of this field at neighbouring points, with a certain error (expressing the latter proved complicated in general configurations, see section 2.2.1.3).

The SPH interpolation is thus built in two steps: a continuous one (the smoothing) and a discrete one (the sampling). Let us first consider the smoothing step. An exact estimation of an arbitrary function at a point can be obtained through the convolution between this function and the delta Dirac distribution. Let A be an arbitrary scalar field defined on the domain Ω. The value of A at position $\boldsymbol{r}$ is given by:

$$A(\boldsymbol{r}) = \int_{\Omega} A(\boldsymbol{r}')\delta(\boldsymbol{r} - \boldsymbol{r}')d\boldsymbol{r}' \tag{2.1}$$

The delta Dirac distribution is defined as:

$$\delta : \begin{cases} \mathcal{D} \to \mathbb{R} \\ \phi \to < \delta, \phi > = \displaystyle\int_{-\infty}^{+\infty} \delta(x)\phi(x)dx = \phi(0) \end{cases} \tag{2.2}$$

and $\delta(\boldsymbol{r}) = \delta(x)\delta(y)\delta(z)$ where $\boldsymbol{r} = x\boldsymbol{e}_x + y\boldsymbol{e}_y + z\boldsymbol{e}_z$ and $\boldsymbol{e}_x$, $\boldsymbol{e}_y$, $\boldsymbol{e}_z$ are unit vectors in the x, y, z directions respectively. More information about the theory of distributions can be found in [128]. The delta Dirac distribution gives an exact estimate of a field at position $\boldsymbol{r}$, but it is not possible to define it numerically. It is thus necessary to represent it through a function called herein a kernel and denoted by $w(\boldsymbol{r} - \boldsymbol{r}')$. The latter has similar properties as the delta Dirac distribution but is defined on a non-null interval of space. As a consequence, the value of A at $\boldsymbol{r}$ is estimated through a continuous interpolation that involves its values at surrounding points. This reads:

$$[A]_c(\boldsymbol{r}) = \int_{\Omega_r} A(\boldsymbol{r}')w(\boldsymbol{r} - \boldsymbol{r}')d\boldsymbol{r}' \tag{2.3}$$

The smoothing is achieved through the choice of a kernel function that decreases with the distance to the interpolation point (usually a bell-shaped function). This choice will be discussed in section 2.2.1.2. Before that, let us go on with the construction of the SPH interpolation, for which in a

second step the continuous medium is discretised into particles. The latter are macroscopic points of matter to which physical quantities are associated like a pressure, a temperature, a velocity, etc. They also serve as moving interpolation points and follow the fluid trajectories. The continuous interpolation (2.3) is then approximated by a discrete sum over the particles b surrounding the interpolation point placed at $\boldsymbol{r}_a$, a being a particle. The resulting discrete interpolation reads:

$$[A]_d(\boldsymbol{r}_a) = \sum_{b \in \mathcal{P}} V_b A_b w_{ab} \tag{2.4}$$

where $\mathcal{P}$ is the set of all fluid particles, the subscripts a,b represent the particles, $A_b = A(\boldsymbol{r}_b)$ and V_b is the volume of particle b. $w_{ab} = w(\boldsymbol{r}_a - \boldsymbol{r}_b)$ was also defined. The volume of a particle is defined as $V_b = \frac{m_b}{\rho_b}$. The value of the particle mass m_b is based on the initial volume V_b^0 and the reference density ρ_0, and is computed as $m_b = V_b^0 \rho_0$. In some works, the particles masses are considered as time-variable, but in the present work they are considered as constant (except at inlet/outlet particles in our model for in/outflow boundaries, as will be explained in the section 3.5). The initial volume in dimension d is calculated through $V_b^0 = \delta r^d$ where δr is the interparticle distance, taking care that at the initial time the particles are placed on a Cartesian grid. Note that with this definition of the particle volume and mass, a partition of unity is achieved at the beginning of the simulation, but it is not conserved during the simulation where the fraction of total fluid volume carried by a particle b is not equal to $\frac{m_b}{\rho_b}$.

2.2.1.2 Definition of the kernel function

The kernel function is most of time radial (*i.e.* $w(\boldsymbol{r} - \boldsymbol{r}') = w(|\boldsymbol{r} - \boldsymbol{r}'|)$) and its value decreases as the distance between $\boldsymbol{r}$ and $\boldsymbol{r}'$ gets bigger, so that the interpolation is smooth. It can be defined on a compact or an infinite support, denoted by Ω_r for a kernel centred on $\boldsymbol{r}$. When the support is compact its size is usually parametrised by the so-called smoothing length h. In practice infinite supports are not used since it would mean that each particle interacts with all the particles of the domain, which is computationally too expensive. Thus, in this work only kernels with compact supports are considered. The kernel function must be sufficiently smooth (at least $\mathcal{C}^1$), so as to be able to compute interpolations of the fields derivatives, as we will see in Sections 2.2.2 and 2.2.3. Besides, it must tend to the Dirac distribution (2.2) (in the sense of distributions) when its support size tends to zero:

$$w(\boldsymbol{r} - \boldsymbol{r}') \xrightarrow{h \to 0} \delta(\boldsymbol{r} - \boldsymbol{r}') \tag{2.5}$$

Regarding the accuracy of the continuous SPH interpolation (2.3), a second order Taylor expansion of $A(\boldsymbol{r}')$ around $\boldsymbol{r}$ yields:

$$A(\boldsymbol{r}') = A(\boldsymbol{r}) - \frac{\partial A}{\partial \boldsymbol{r}} \cdot (\boldsymbol{r} - \boldsymbol{r}') + \mathcal{O}(|\boldsymbol{r} - \boldsymbol{r}'|^2) \tag{2.6}$$

Substituting this expression in (2.3) gives:

$$[A]_c(\boldsymbol{r}) = A(\boldsymbol{r},t)\int_{\Omega_r} w_h(\boldsymbol{r}-\boldsymbol{r}')d\boldsymbol{r}' - \frac{\partial A}{\partial \boldsymbol{r}}\cdot\int_{\Omega_r} w_h(\boldsymbol{r}-\boldsymbol{r}')(\boldsymbol{r}-\boldsymbol{r}')d\boldsymbol{r}' + \mathcal{O}(|\boldsymbol{r}-\boldsymbol{r}'|^2) \quad (2.7)$$

Thus it appears that to obtain a first order consistent continuous SPH interpolation the two following conditions must be satisfied:

$$\int_{\Omega_r} w(\boldsymbol{r}-\boldsymbol{r}')d\boldsymbol{r}' = 1 \quad (2.8)$$

$$\int_{\Omega_r} w(\boldsymbol{r}-\boldsymbol{r}')(\boldsymbol{r}-\boldsymbol{r}')d\boldsymbol{r}' = \boldsymbol{0} \quad (2.9)$$

The kernel functions are then built so that they satisfy these conditions. Equation (2.8) is a normalising condition easily obtained for unbounded flows through a normalising coefficient. For condition (2.9) to be satisfied the kernel function must be even and Ω_r must be central-symmetrically invariant, which is true for radial functions. In the case of a bounded domain, the latter condition and (2.8) are not respected in the vicinity of the boundary, due to the kernel support truncation. This observation led to the development of techniques for boundary conditions based on a wall normalising correction of (2.3), which makes property (2.8) valid even close to the boundary (see Section 2.4.2).

Coming back to the building of a kernel, the most intuitive choice is a Gaussian function, since it satisfies (2.5). Though, it is also possible to build piecewise compactly-supported polynomials having the required properties. The kernel function used in this work is the 5th order Wendland kernel, a polynomial kernel defined as:

$$\begin{cases} w(|\boldsymbol{r}-\boldsymbol{r}'|) = \frac{\alpha_{W,d}}{h^d} f_W(q) \\ q \doteqdot \dfrac{|\boldsymbol{r}-\boldsymbol{r}'|}{h} \end{cases} \quad (2.10)$$

where $\alpha_{W,d}$ is a normalising constant, which depends on the problem dimension d. The function f_W is defined as:

$$f_W(q) = \begin{cases} \left(1-\dfrac{q}{2}\right)^4 (1+2q) & 0 \le q \le q_{max} \\ 0 & q_{max} < q \end{cases} \quad (2.11)$$

where $q_{max} = 2$ is the dimensionless size of the kernel support. Its first derivative reads:

$$f'_W(q) = \begin{cases} -5q\left(1-\dfrac{q}{2}\right)^3 & 0 \le q \le q_{max} \\ 0 & q_{max} < q \end{cases} \quad (2.12)$$

The normalising constants in 2-D and 3-D are:

$$\alpha_{W,2} = \frac{7}{4\pi}\ ,\ \alpha_{W,3} = \frac{21}{16\pi} \quad (2.13)$$

Other kernels that are are widely used in the SPH literature are the B-splines. In particular the $5th$ order B-spline is defined as:

$$w(|\boldsymbol{r}-\boldsymbol{r}'|)=\frac{\alpha_{5,d}}{h^d}f_5(q) \tag{2.14}$$

with:

$$f_5(q)=\begin{cases}(3-q)^5-6\,(2-q)^5+15\,(1-q)^5 & 0\leq q\leq 1\\ (3-q)^5-6\,(2-q)^5 & 1\leq q\leq 2\\ (3-q)^5 & 2\leq q\leq 3\\ 0 & if\ \ 3<q\end{cases} \tag{2.15}$$

and the normalising constant in 2-D and 3-D reads:

$$\alpha_{5,2}=\frac{7}{478\pi} \tag{2.16}$$

$$\alpha_{5,3}=\frac{1}{120\pi} \tag{2.17}$$

The derivative of $f_5(q)$ then reads:

$$f_5'(q)=-5\begin{cases}(3-q)^4-6\,(2-q)^4+15\,(1-q)^4 & 0\leq q\leq 1\\ (3-q)^4-6\,(2-q)^4 & 1\leq q\leq 2\\ (3-q)^4 & 2\leq q\leq 3\\ 0 & if\ \ 3<q\end{cases} \tag{2.18}$$

Another kernel quite often used in the literature is the truncated Gaussian, defined as:

$$w(|\boldsymbol{r}-\boldsymbol{r}'|)=\frac{\alpha_{G,d,h_c}}{h^d}\left(e^{-q^2}-e^{-\left(\frac{h_c}{h}\right)^2}\right) \tag{2.19}$$

where most of time $h_c=3h$ and the normalising constant in 2-D and 3-D reads:

$$\alpha_{G,2,3h}=\frac{1}{\pi\,(1-10e^{-9})} \tag{2.20}$$

$$\alpha_{G,3,3h}=\frac{1}{\pi\,(\sqrt{\pi}-36e^{-9})} \tag{2.21}$$

Figure 2.4 shows plots of these kernels and their first derivatives.

The ratio $\frac{h}{\delta r}$ appears as important since it is linked to the number of neighbours taken into account in the interpolation when the particles are arranged in a homogeneous and isotropic configuration (recall that δr is the initial interparticular distance). In this work this ratio is fixed during a simulation, although it is possible to allow it to vary.

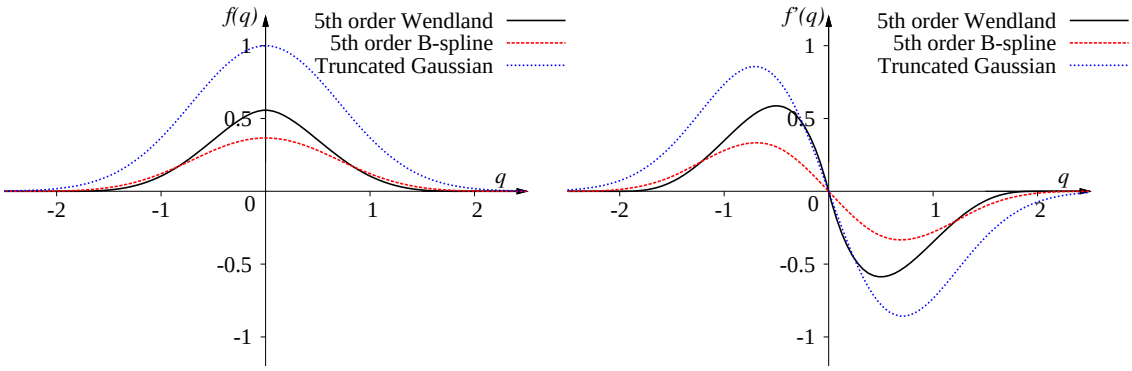


Figure 2.4: Plot of the non-normalised 5th order Wendland, 5th order B-spline and truncated Gaussian kernels (left) and their first derivatives (right) in 2-D.

2.2.1.3 Accuracy of the SPH interpolation

The error made through the SPH interpolation of a field A can be written as the sum of a continuous error E_c (the integration error) and a discrete error E_d, with:

$$E_c = [A]_c(\boldsymbol{r}_a) - \int_\Omega A(\boldsymbol{r}')\delta(\boldsymbol{r}_a - \boldsymbol{r}')d\boldsymbol{r}' \tag{2.22}$$

$$E_d = [A]_d(\boldsymbol{r}_a) - [A]_c(\boldsymbol{r}_a) \tag{2.23}$$

where $[A]_c(\boldsymbol{r}_a)$ is the continuous interpolation of A at $\boldsymbol{r}_a$, defined through (2.3) and $[A]_d(\boldsymbol{r}_a)$ is the discrete interpolation of A at $\boldsymbol{r}_a$, defined through (2.4). We saw that provided the kernel function fulfils properties (2.8) and (2.9), the integration error E_c is order h^2. More precisely, following [147], the integration error is expressed as:

$$E_c = \frac{C_{w,2}}{2d}\nabla^2 A(\boldsymbol{r}_a)h^2 + \mathcal{O}(h^4) \tag{2.24}$$

where $C_{w,2}$ is a factor that depends on the kernel function:

$$C_{w,2} = \alpha_{w,d} S_d \int_0^{q_{max}} f(q) q^{n+1} dq \tag{2.25}$$

$\alpha_{w,d}$ is the kernel normalising constant, $f(q)$ the non-normalised kernel function (see section 2.2.1.2) and S_d is the area of a d-sphere with unitary radius:

$$S_d = \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})} \tag{2.26}$$

Γ being the Gamma Euler function (see [2] for its definition).

However, estimating the discretisation error is very complex since the particles move and reach disordered configurations. For random arrangements of particles it is possible to estimate the mean error since the discrete interpolation operator is then similar to a Monte Carlo type statistical evaluation [114]. Though, the particles are not placed randomly: the disorder is lower and determined by the discrete equations of motion. It is interesting to consider a simple case where the particles are placed on a Cartesian grid on an unbounded domain. It is then possible to evaluate the discretisation error through [147]:

$$E_d = d\, A(\boldsymbol{r}_a)\hat{w}\left(K^+_{\delta r}\right) - \frac{1}{2}\nabla^2 A(\boldsymbol{r}_a)h^2\hat{w}''\left(K^+_{\delta r}\right) + \mathcal{O}(h^3) \tag{2.27}$$

where d is the geometrical dimension of the domain, $\hat{w}$ is the Fourier transform of w and $K^+_{\delta r} = \frac{2h\pi}{\delta r}$ is a dimensionless wave number.

Thus, we see that (under the crude assumption of a Cartesian particle distribution) the total interpolation error varies in h^2 but reaches a lower bound $\left(d\, A(\boldsymbol{r}_a)\hat{w}\left(K^+_{\delta r}\right)\right)$ for small enough values of h (since $\frac{h}{\delta r}$ is considered as constant in this work). It is also important to note that the Fourier transform of the kernel function plays a crucial role in the error being made. Note that when the kernel has a decreasing Fourier transform (which is the case for the Gaussian kernel but not for the B-splines and the 5th order Wendland kernel), the error decreases when increasing $\frac{h}{\delta r}$. On the other hand, at a fixed ratio $\frac{h}{\delta r}$, decreasing h (and thus refining the discretisation) reduces the error until the lower bound of the later is reached. As a consequence, the convergence of the SPH interpolation can only be achieved when h tends to 0 while $\frac{h}{\delta r}$ tends towards infinity.

A similar consistency analysis was done with an arbitrary distribution of points in 1-D by Quinlan *et al.* [117], and later on in 3-D by Amicarelli *et al.* [5] (although both suppose that the particles volumes form a partition of unity, which is not the case with the definition $V_b = \frac{m_b}{\rho_b}$). It comes out that the accuracy of the SPH interpolation also depends on the particles distribution: the error is lowest with a homogeneous particle distribution (*e.g.* Cartesian), provided the conditions (2.8) and (2.9) are fulfilled.

2.2.2 First order differential operators in SPH

In order to solve the system (1.92) it is necessary to build discrete differential operators of first and second order. In this section the construction of SPH gradient and divergence operators is detailed. It is based on the SPH interpolation defined in the previous Section, considering unbounded domains[1].

[1] Actually they are used with classical treatments of the boundary conditions in SPH, as we will see in Section 2.4.

2.2.2.1 SPH gradient

According to (2.3), the continuous interpolation of the gradient of an arbitrary $\mathcal{C}^1$ field A at position $\boldsymbol{r}$ reads:

$$[\boldsymbol{\nabla} A]_c(\boldsymbol{r}) = \int_{\Omega\cap\Omega_r} \frac{\partial A(\boldsymbol{r}')}{\partial \boldsymbol{r}'} w(\boldsymbol{r}-\boldsymbol{r}')d\boldsymbol{r}' \tag{2.28}$$

An integration by part of the right-hand side yields:

$$[\boldsymbol{\nabla} A]_c(\boldsymbol{r}) = -\int_{\partial\Omega\cap\Omega_r} A(\boldsymbol{r}')w(\boldsymbol{r}-\boldsymbol{r}')\boldsymbol{n}(\boldsymbol{r}')d\Gamma' - \int_{\Omega\cap\Omega_r} A(\boldsymbol{r}')\frac{\partial w(\boldsymbol{r}-\boldsymbol{r}')}{\partial \boldsymbol{r}'}d\boldsymbol{r}' \tag{2.29}$$

where $\partial\Omega$ is the domain boundary, $\boldsymbol{n}(\boldsymbol{r}')$ is the inward[2] normal to the boundary at $\boldsymbol{r}'$ and $d\Gamma'$ is a surface element of $\partial\Omega\cap\Omega_r$. For an unbounded domain the first integral cancels out[3]. Then, equation (2.29) becomes:

$$[\boldsymbol{\nabla} A]_c(\boldsymbol{r}) = -\int_{\Omega\cap\Omega_r} A(\boldsymbol{r}')\frac{\partial w(\boldsymbol{r}-\boldsymbol{r}')}{\partial \boldsymbol{r}'}d\boldsymbol{r}' \tag{2.30}$$

Since the kernel is radial it is symmetric, its gradient is antisymmetric:

$$\frac{\partial w_h(\boldsymbol{r}-\boldsymbol{r}')}{\partial \boldsymbol{r}'} = -\frac{\partial w(\boldsymbol{r}-\boldsymbol{r}')}{\partial \boldsymbol{r}} \doteqdot -\boldsymbol{\nabla} w(\boldsymbol{r}-\boldsymbol{r}') \tag{2.31}$$

so that finally:

$$[\boldsymbol{\nabla} A]_c(\boldsymbol{r}) = \int_{\Omega\cap\Omega_r} A(\boldsymbol{r}')\boldsymbol{\nabla} w(\boldsymbol{r}-\boldsymbol{r}')d\boldsymbol{r}' \tag{2.32}$$

The discrete interpolation corresponding to this classical continuous interpolation of the SPH gradient reads:

$$\boldsymbol{G}_a\{A_b\} = \sum_{b\in\mathcal{P}} V_b A_b \boldsymbol{\nabla} w_{ab} \tag{2.33}$$

where $\mathcal{P}$ is the set of all fluid particles and:

$$\boldsymbol{\nabla} w_{ab} \doteqdot \frac{dw(r_{ab})}{d\boldsymbol{r}_a} = -\boldsymbol{\nabla} w_{ba} \tag{2.34}$$

with $r_{ab} = |\boldsymbol{r}_a - \boldsymbol{r}_b|$. It is thus possible to compute an approximate value of the gradient of a field from the field values at surrounding particles and the kernel gradient value, which is known. However, we see that with (2.33) the SPH gradient of a constant is not equal to zero. Thus, other expressions for the SPH gradient have been looked for. For example it can be defined by applying

[2] Note that in this Chapter and the subsequent ones, we use the convention of inward normal vector $\boldsymbol{n}$, contrary to Chapter 1.

[3] With classical treatments of the boundary conditions in SPH, the boundary term is considered as equal to zero (see Section 2.4).

the same procedure as shown above to the right-hand side of the following equality:

$$\boldsymbol{\nabla} A = \frac{1}{\rho} \left[\boldsymbol{\nabla}(\rho A) - A \boldsymbol{\nabla} \rho \right] \tag{2.35}$$

Then another expression for the SPH gradient is obtained:

$$\boldsymbol{G}_a^-\{A_b\} = -\frac{1}{\rho_a} \sum_{b \in \mathcal{P}} m_b A_{ab} \boldsymbol{\nabla} w_{ab} \tag{2.36}$$

where $A_{ab} = A_a - A_b$. It can also be applied to a vector field, which reads:

$$\boldsymbol{G}_a^-\{\boldsymbol{A}_b\} = -\frac{1}{\rho_a} \sum_{b \in \mathcal{P}} m_b \boldsymbol{A}_{ab} \otimes \boldsymbol{\nabla} w_{ab} \tag{2.37}$$

These expressions are symmetric (see (2.34)) and thus equal to zero for a constant field (zero-order consistency). In an SPH model it is used to compute velocity gradients for example. It ensures that a constant velocity field does not lead to artificial turbulent kinetic energy production (see equation (1.55)).

Since the gradient operator is also involved in the pressure forces in the Navier–Stokes equations, it is important to have an expression for the SPH gradient which conserves linear momentum. The gradient as defined through (2.36) does not fulfil this requirement since it is symmetric, while the action-reaction principle requires an antisymmetric gradient. To build such an SPH operator it is possible to start from the right-hand side of the following equality:

$$\boldsymbol{\nabla} A = \rho \boldsymbol{\nabla} \left(\frac{A}{\rho} \right) + \frac{A}{\rho} \boldsymbol{\nabla} \rho \tag{2.38}$$

This leads to the following expression for the SPH gradient:

$$\boldsymbol{G}_a^+\{A_b\} = \rho_a \sum_{b \in \mathcal{P}} m_b \left(\frac{A_a}{\rho_a^2} + \frac{A_b}{\rho_b^2} \right) \boldsymbol{\nabla} w_{ab} \tag{2.39}$$

This time the operator fulfils the action-reaction principle (in the absence of walls) when applied to the pressure.

2.2.2.2 SPH divergence

The same reasoning can be applied to obtain an expression for the SPH divergence of a field. Expressions similar to (2.33), (2.36) and (2.39) are then obtained. The SPH divergence derived from the continuous interpolation of $\nabla \cdot \boldsymbol{A}$ reads:

$$D_a\{\boldsymbol{A}_b\} = \sum_{b \in \mathcal{P}} V_b \boldsymbol{A}_b \cdot \boldsymbol{\nabla} w_{ab} \tag{2.40}$$

On the other hand, the symmetric form of the SPH divergence reads:

$$D_a^-\{\boldsymbol{A}_b\} = -\frac{1}{\rho_a}\sum_{b\in\mathcal{P}} m_b \boldsymbol{A}_{ab}\cdot\boldsymbol{\nabla} w_{ab} \tag{2.41}$$

whereas the antisymmetric expression reads:

$$D_a^+\{\boldsymbol{A}_b\} = \rho_a \sum_{b\in\mathcal{P}} V_b \left(\frac{\boldsymbol{A}_a}{\rho_a^2} + \frac{\boldsymbol{A}_b}{\rho_b^2}\right)\cdot\nabla w_{ab} \tag{2.42}$$

The latter was not used in this work, except for the construction of a Laplacian operator (see section 2.2.3). It has been used in few works only for the velocity divergence computation (see *e.g.* [129]).

2.2.2.3 Properties of the first order differential operators in SPH

It is interesting to note that the operators $\boldsymbol{G}_a^+$ and D_a^- are skew-adjoint, like the continuous gradient and divergence operators under relevant boundary conditions (see section 1.1.2.1 of Chapter 1). To prove that, let us define the following inner products on the vectorial spaces of $\{A_b\}$ and $\{\boldsymbol{A}_b\}$ (they are of finite dimension, proportional to the number of particles):

$$\begin{aligned}
\langle\{\boldsymbol{A}_a\},\{\boldsymbol{B}_a\}\rangle &= \sum_{a\in\mathcal{P}} V_a \boldsymbol{A}_a\cdot\boldsymbol{B}_a \approx \int_\Omega \boldsymbol{A}(\boldsymbol{r})\cdot\boldsymbol{B}(\boldsymbol{r})d\Omega \\
(\{A_a\},\{B_a\}) &= \sum_{a\in\mathcal{P}} V_a A_a B_a \approx \int_\Omega A(\boldsymbol{r})B(\boldsymbol{r})d\Omega
\end{aligned} \tag{2.43}$$

(compare to (1.13)). Then the following relation is found [147]:

$$\begin{aligned}
\langle \boldsymbol{G}_a^+\{A_b\},\{\boldsymbol{B}_a\}\rangle &= \sum_{a,b\in\mathcal{P}} m_b m_a \left(\frac{A_a}{\rho_a^2}+\frac{A_b}{\rho_b^2}\right)\boldsymbol{B}_a\cdot\boldsymbol{\nabla} w_{ab} \\
&= -\sum_{a,b\in\mathcal{P}} m_b m_a \left(\frac{A_a}{\rho_a^2}+\frac{A_b}{\rho_b^2}\right)\boldsymbol{B}_b\cdot\boldsymbol{\nabla} w_{ab} \\
&= \frac{1}{2}\sum_{a,b\in\mathcal{P}} m_b m_a \left(\frac{A_a}{\rho_a^2}+\frac{A_b}{\rho_b^2}\right)\boldsymbol{B}_{ab}\cdot\boldsymbol{\nabla} w_{ab}
\end{aligned} \tag{2.44}$$

(the second line holding due to the antisymmetry of the kernel gradient and the last line being an average of the first two ones). Concerning the inner product with the SPH divergence, the same

reasoning yields :

$$\begin{aligned}(\{A_a\}, D_a^-\{\boldsymbol{B}_b\}) &= -\sum_{a,b\in\mathcal{P}} m_a m_b \frac{A_a}{\rho_a^2}\boldsymbol{B}_{ab}\cdot\boldsymbol{\nabla}w_{ab} \\ &= -\frac{1}{2}\sum_{a,b\in\mathcal{P}} m_a m_b \left(\frac{A_a}{\rho_a^2}+\frac{A_b}{\rho_b^2}\right)\boldsymbol{B}_{ab}\cdot\boldsymbol{\nabla}w_{ab}\end{aligned} \tag{2.45}$$

Thus, the following relation holds:

$$\left(\{A_a\}, D_a^-\{\boldsymbol{B}_b\}\right) = -\langle \boldsymbol{G}_a^+\{A_b\}, \{\boldsymbol{B}_a\}\rangle \tag{2.46}$$

which shows that the $\boldsymbol{G}_a^+$ and D_a^- operators are skew-adjoint, as said earlier. This property will prove important when making a choice for the SPH operators. Indeed, in WCSPH this property yields energy conservation [115] while in ISPH it is useful for the accuracy of the projection method (see sections 2.3.1.1 and 2.3.2). Note that the $\boldsymbol{G}_a^-$ and D_a^+ operators are also skew-adjoint, which can be proved in the same way.

An issue with the classical SPH divergence and gradient operators is that they lack accuracy. The antisymmetric forms of these operators (D_a^+ and $\boldsymbol{G}_a^+$) are not even zero-order consistent. The symmetric forms (D_a^- and $\boldsymbol{G}_a^-$) are zero-order consistent (as pointed out for $\boldsymbol{G}_a^-$ in section 2.2.2.1) and can be made 1st order consistent through a renormalisation technique [20, 109, 145]. The idea is to impose the gradient of the position to be equal to the identity tensor through a renormalisation matrix:

$$-\sum_{b\in\mathcal{P}} V_b \boldsymbol{r}_{ab}\otimes(\boldsymbol{M}_a\boldsymbol{\nabla}w_{ab}) = \boldsymbol{I} \tag{2.47}$$

where $\boldsymbol{M}_a$ is a renormalisation matrix defined through:

$$\boldsymbol{M}_a = \left[-\left(\sum_{b\in\mathcal{P}} V_b \boldsymbol{r}_{ab}\otimes\boldsymbol{\nabla}w_{ab}\right)^T\right]^{-1} \tag{2.48}$$

The modified first-order consistent gradient operator thus reads:

$$\boldsymbol{G}_a^{-,1}\{A_b\} = -\sum_{b\in\mathcal{P}} V_b A_{ab}\boldsymbol{M}_a\boldsymbol{\nabla}w_{ab} \tag{2.49}$$

The same applies to the symmetric SPH divergence operator:

$$D_a^{-,1}\{\boldsymbol{A}_b\} = -\frac{1}{\rho_a}\sum_{b\in\mathcal{P}} m_b \boldsymbol{A}_{ab}\cdot\boldsymbol{M}_a\nabla w_{ab} \tag{2.50}$$

2.2.3 Second order differential operator in SPH

2.2.3.1 Construction of a Laplacian operator in SPH

An SPH form of the Laplacian operator is needed to solve system (1.92), in order to compute the viscous term, the temperature, k and ϵ diffusion terms and to write the pressure Poisson equation in case of an incompressible scheme. The first possibility is to proceed as in Section 2.2.2 and to write:

$$L_a\{A_b\} = \sum_{b\in\mathcal{P}} A_b \nabla^2 w_{ab} \tag{2.51}$$

where L_a is the discrete SPH Laplacian. Though, this expression depends on the second derivative of the kernel function, which makes it very sensitive to particle disorder [103]. Besides, the second derivative of the kernel may change signs and this expression is not antisymmetric, which is not representative of forces interactions (although it is possible to make it antisymmetric as we did for $\boldsymbol{G}_a^+$). A better formulation consists in writing the Laplacian as the SPH divergence of an SPH gradient, as proposed by Cummins & Rudmann [19]. This is much more interesting since it makes the Laplacian operator consistent with the divergence and gradient operators, as it is in a continuous framework. Using such an operator in a projection method thus makes the projection exact provided the gradient and divergence operators are skew-adjoint (see sections 1.1.2 and 2.2.2.3). Let us consider the general case of the interpolation of $\nabla\cdot(B\boldsymbol{\nabla}A)$, where B is a diffusion coefficient for the $\mathcal{C}^2$ field A. It is then possible to define the discrete SPH Laplacian as:

$$L_a\{B_b, A_b\} = D_a\{B_b\boldsymbol{G}_b\{A_c\}\} = -\sum_{b\in\mathcal{P}} V_b B_b \left(\sum_{c\in\mathcal{P}} V_c A_c \boldsymbol{\nabla} w_{bc}\right)\cdot\boldsymbol{\nabla} w_{ab} \tag{2.52}$$

Recall that D_a and $\boldsymbol{G}_a$ were defined in equations (2.40) and (2.33). This expression involves a two-fold summation over the neighbours, which is computationally very expensive. Besides, Cummins & Rudmann [19] showed that using this form of the SPH Laplacian is not satisfactory (see section 2.3.2.1).

A third way to write the discrete SPH Laplacian operator was proposed by Morris *et al.* [104]. Once again the Laplacian is written as the divergence of a gradient, but this time while the divergence is taken in an SPH form, the gradient is expressed through a finite difference approximation. Let us consider a general case where a varying diffusion coefficient B is applied. First, the Laplacian is made symmetric through the equality:

$$\nabla\cdot(B\boldsymbol{\nabla}A) = B\boldsymbol{\nabla}A\cdot\boldsymbol{\nabla}1 + \nabla\cdot(B\boldsymbol{\nabla}A) \tag{2.53}$$

which is the first step to build an antisymmetric divergence operator. The continuous SPH interpo-

lation of the right-hand side then reads, after integration by parts:

$$\begin{aligned}\left[\nabla\cdot(B\boldsymbol{\nabla}A)\right]_c(\boldsymbol{r}_a) = &\int_{\Omega\cap\Omega_a}\left[B(\boldsymbol{r}_a)\boldsymbol{\nabla}A(\boldsymbol{r}_a)+B(\boldsymbol{r}')\boldsymbol{\nabla}A(\boldsymbol{r}')\right]\cdot\boldsymbol{\nabla}w(\boldsymbol{r}_a-\boldsymbol{r}')d\boldsymbol{r}'\\ &-\int_{\partial\Omega\cap\Omega_a}\left[B(\boldsymbol{r}_a)\boldsymbol{\nabla}A(\boldsymbol{r}_a)+B(\boldsymbol{r}')\boldsymbol{\nabla}A(\boldsymbol{r}')\right]\cdot\boldsymbol{n}(\boldsymbol{r}')w(\boldsymbol{r}_a-\boldsymbol{r}')d\Gamma'\end{aligned} \tag{2.54}$$

Note that the normal to the wall is oriented inwards (hence the minus in front of the boundary integral). The boundary term vanishes in the absence of walls or with classical SPH boundary conditions (see section 2.4.1). A finite difference approximation is then used to estimate the gradient of the field in the volumic integral:

$$B(\boldsymbol{r}_a)\boldsymbol{\nabla}A(\boldsymbol{r}_a)\cdot(\boldsymbol{r}_a-\boldsymbol{r}')\simeq\overline{B}(\boldsymbol{r}_a,\boldsymbol{r}')(A(\boldsymbol{r}_a)-A(\boldsymbol{r}'))\simeq-B(\boldsymbol{r}')\boldsymbol{\nabla}A(\boldsymbol{r}')\cdot(\boldsymbol{r}'-\boldsymbol{r}_a) \tag{2.55}$$

where $\overline{B}(\boldsymbol{r}_a,\boldsymbol{r}')$ is a mean of the diffusion coefficient between $\boldsymbol{r}_a$ and $\boldsymbol{r}'$. It may be chosen as an arithmetic mean: $\overline{B}(\boldsymbol{r}_a,\boldsymbol{r}')=\frac{B(\boldsymbol{r}_a)+B(\boldsymbol{r}')}{2}$ or a harmonic mean: $\overline{B}(\boldsymbol{r}_a,\boldsymbol{r}')=\frac{B(\boldsymbol{r}_a)B(\boldsymbol{r}')}{B(\boldsymbol{r}_a)+B(\boldsymbol{r}')}$. In this work an arithmetic mean was used. The continuous interpolation of the Laplacian then reads:

$$\left[\nabla\cdot(B\boldsymbol{\nabla}A)\right]_c(\boldsymbol{r}_a)=2\int_{\Omega\cap\Omega_a}\overline{B}(\boldsymbol{r}_a,\boldsymbol{r}')\frac{A(\boldsymbol{r}_a)-A(\boldsymbol{r}')}{(\boldsymbol{r}_a-\boldsymbol{r}')^2}(\boldsymbol{r}_a-\boldsymbol{r}')\cdot\boldsymbol{\nabla}w(\boldsymbol{r}_a-\boldsymbol{r}')d\boldsymbol{r}' \tag{2.56}$$

After approximating the volumic integral by the summation over the fluid particles, the following SPH Laplacian operator is obtained, which will be referred to as the Morris Laplacian in what follows:

$$L_a\{B_b,A_b\}=2\sum_{b\in\mathcal{P}}V_b\overline{B}_{ab}\frac{A_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab} \tag{2.57}$$

Recall that $A_{ab}\doteqdot A_a-A_b$. In the case where $B=1$ everywhere, this SPH Laplacian may be simplified into:

$$L_a\{A_b\}=2\sum_{b\in\mathcal{P}}V_b\frac{A_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab} \tag{2.58}$$

Note also that the Morris Laplacian can be applied to vectors, which reads:

$$\boldsymbol{L}_a\{B_b,\boldsymbol{A}_b\}=2\sum_{b\in\mathcal{P}}V_b\overline{B}_{ab}\frac{\boldsymbol{A}_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab} \tag{2.59}$$

Other SPH Laplacian operators exist, like the one proposed by Monaghan & Gingold [101], but the one used in this work is the Morris Laplacian (2.57) since it was shown in [10] that it provides better results when applied to the viscous forces on a Poiseuille channel flow and on a lid-driven cavity case for a range of Reynolds numbers. However, the Morris Laplacian operator was shown in [16] to be inconsistent close to the free-surface, which suggests that another Laplacian operator like the one proposed by Monaghan & Gingold may be more suited to free-surface flows (although this was not tested in the present thesis).

2.2.3.2 Accuracy of the SPH Laplacian operator

The Morris Laplacian operator used herein is only zero-order consistent, and making it first order or second order consistent proved quite complex. Schwaiger [127] proposed a method to build second-order consistent Laplacian operator, but this included an approximation. Later on, Fatehi & Manzari [32] gave an exact formulation of this technique, which requires the calculation of a fourth-order tensor. The Laplacian operator is then given by:

$$L_a\{A_b\} = \widehat{\boldsymbol{M}}_a : \left[2 \sum_{b \in \mathcal{P}} V_b \boldsymbol{r}_{ab} \otimes \nabla w_{ab} \left(\frac{A_{ab}}{r_{ab}^2} - \frac{\boldsymbol{r}_{ab}}{r_{ab}^2} \cdot \boldsymbol{G}_b^{-,1}\{A_c\} \right) \right] \tag{2.60}$$

where $\boldsymbol{G}_b^{-,1}$ is defined through (2.49) and the fourth order symmetric tensor $\widehat{\boldsymbol{M}}_a$ is given by:

$$\widehat{\boldsymbol{M}}_a : \left[\begin{array}{l} \displaystyle\sum_{b \in \mathcal{P}} V_b \frac{\boldsymbol{r}_{ab}}{r_{ab}^2} \otimes \boldsymbol{r}_{ab} \otimes \boldsymbol{r}_{ab} \otimes \boldsymbol{\nabla} w_{ab} \\ \displaystyle + \left(\sum_{b \in \mathcal{P}} \frac{V_b}{r_{ab}^2} \boldsymbol{r}_{ab} \otimes \boldsymbol{r}_{ab} \otimes \boldsymbol{\nabla} w_{ab} \right) \cdot \boldsymbol{M}_a \cdot \left(\sum_{b \in \mathcal{P}} V_b \boldsymbol{r}_{ab} \otimes \boldsymbol{r}_{ab} \otimes \boldsymbol{\nabla} w_{ab} \right) \end{array} \right] = -\boldsymbol{I} \tag{2.61}$$

with $\boldsymbol{M}_a$ given by equation (2.48). The calculation of $\widehat{\boldsymbol{M}}_a$ comes to solving a system of four equations with four unknowns for each particle in 2-D[4], and is thus computationally expensive.

2.3 Modelling incompressible flows with SPH

2.3.1 Classical weakly-compressible approach

2.3.1.1 Formulation

Modelling incompressible flows with SPH has classically been done through weakly compressible SPH (WCSPH) models, as is thoroughly described in [103]. In WCSPH the Navier–Stokes equations are solved in their weakly-compressible Lagrangian form (1.5) using the equation of state (1.6) to close the system. The value of the numerical speed of sound c_0 in (1.6) is set so that the density variations are kept in the interval $\pm 1\%$. For confined flows c_0 is usually taken as $c_0 = 10 U_{max}$, where U_{max} is the maximum velocity of the flow. For free-surface flows, c_0 is taken as $c_0 = 10 \max(\sqrt{gH}, U_{max})$, where H is a reference free-surface elevation.

It is interesting to derive the space-discretised inviscid WCSPH equations from the Lagrangian of a

[4] Indeed, $\boldsymbol{I}$ is the identity matrix so $\widehat{\boldsymbol{M}}_a$ is a matrix too and the quantity in between brackets is a fourth order tensor.

non-dissipative discrete system of particles [12, 115, 147]. The discrete Lagrangian L is defined as:

$$L = T - V \tag{2.62}$$

where T is the kinetic energy:

$$T = \sum_{b \in \mathcal{P}} \frac{1}{2} m_b |\boldsymbol{v}_b|^2 \tag{2.63}$$

and V is the total potential energy:

$$V = \sum_{b \in \mathcal{P}} m_b e_{int,b}(\rho_b, s_b) - \sum_{b \in \mathcal{P}} m_b \boldsymbol{g} \cdot \boldsymbol{r}_b \tag{2.64}$$

where $e_{int,b}(\rho_b, s_b)$ is the internal energy per unit mass of particle b, which depends on its density ρ_b and on its entropy s_b, the latter being a constant for a non-dissipative isothermal system. The discrete (Euler-Lagrange) equation of motion for each particle a is then given by:

$$\forall a \in \mathcal{P},\ \frac{d}{dt}\left(\frac{\partial L}{\partial \boldsymbol{v}_a}\right) = \frac{\partial L}{\partial \boldsymbol{r}_a} \tag{2.65}$$

Differentiating (2.63) and (2.64) gives the partial derivatives of the Lagrangian. Its partial derivative with respect to the velocity is equal to the linear momentum: $\frac{\partial L}{\partial \boldsymbol{v}_a} = m_a \boldsymbol{v}_a$, while its partial derivative with respect to the position can be written as:

$$\frac{\partial L}{\partial \boldsymbol{r}_a} = -\sum_{b \in \mathcal{P}} m_b \frac{p_b}{\rho_b^2} \frac{\partial \rho_b}{\partial \boldsymbol{r}_a} + m_a \boldsymbol{g} \tag{2.66}$$

since the following relation holds:

$$\frac{\partial e_{int,b}}{\partial \boldsymbol{r}_a} = \frac{\partial e_{int,b}}{\partial \rho_b} \frac{\partial \rho_b}{\partial \boldsymbol{r}_a} = \frac{p_b}{\rho_b^2} \frac{\partial \rho_b}{\partial \boldsymbol{r}_a} \tag{2.67}$$

It is then necessary to estimate the quantity $\dfrac{\partial \rho_b}{\partial \boldsymbol{r}_a}$, which can be done starting from the discrete SPH interpolation (2.4) applied to the density:

$$\rho_b = \sum_{c \in \mathcal{P}} m_c w_{bc} \tag{2.68}$$

Differentiating this expression with respect to $\boldsymbol{r}_a$ and considering a constant smoothing length for the kernel gives:

$$\frac{\partial \rho_b}{\partial \boldsymbol{r}_a} = \sum_{c \in \mathcal{P}} m_c (\delta_{ba} - \delta_{ca}) \frac{\partial w_{bc}}{\partial \boldsymbol{r}_a} \tag{2.69}$$

where δ_{ba} is the Kronecker symbol equal to one when $b = a$ and zero otherwise. Writing equation (2.65) with these relations yields the following discrete equations of motion for all particles

a:

$$\begin{aligned}\frac{d\boldsymbol{v}_a}{dt} &= -\sum_{b\in\mathcal{P}} m_b \left(\frac{p_a}{\rho_a^2}+\frac{p_b}{\rho_b^2}\right)\boldsymbol{\nabla} w_{ab} + \boldsymbol{g} \\ &= -\frac{1}{\rho_a}\boldsymbol{G}_a^+\{p_b\} + \boldsymbol{g}\end{aligned} \tag{2.70}$$

where $\boldsymbol{G}_a^+$ is defined through (2.39). This is a discretised form of the inviscid momentum equation in (1.5) (the viscous forces will be dealt with in section 2.7.3). It can be shown (see [12]) that relation (2.70) is also obtained when discretising the continuity equation (first line of (1.5)) through:

$$\frac{d\rho_a}{dt} = -\rho_a D_a^-\{\boldsymbol{v}_b\} \tag{2.71}$$

which is a time-derivative of (2.68):

$$\frac{d\rho_a}{dt} = \sum_{b\in\mathcal{P}} m_b \frac{dw_{ab}}{dt} = \sum_{b\in\mathcal{P}} m_b \boldsymbol{v}_{ab}\cdot\boldsymbol{\nabla} w_{ab} = -\rho_a D_a^-\{\boldsymbol{v}_b\} \tag{2.72}$$

The internal force applied on a by the rest of the fluid is obtained from (2.66) and reads:

$$\boldsymbol{F}_a^{int} = \sum_{b\in\mathcal{P}} m_b \frac{p_b}{\rho_b^2}\frac{\partial\rho_b}{\partial\boldsymbol{r}_a} \tag{2.73}$$

The total virtual work of internal forces then reads:

$$\begin{aligned}\sum_{a\in\mathcal{P}} \boldsymbol{F}_a^{int}\cdot d\boldsymbol{r}_a &= \sum_{b\in\mathcal{P}} \frac{m_b p_b}{\rho_b^2}\sum_{a\in\mathcal{P}}\frac{\partial\rho_b}{\partial\boldsymbol{r}_a}\cdot d\boldsymbol{r}_a = \sum_{b\in\mathcal{P}}\frac{m_b p_b}{\rho_b^2} d\rho_b \\ &= -\sum_{b\in\mathcal{P}} V_b p_b D_b^-\{d\boldsymbol{r}_c\} \\ &= \sum_{a\in\mathcal{P}} V_a \boldsymbol{G}_a^+\{p_b\}\cdot d\boldsymbol{r}_a\end{aligned} \tag{2.74}$$

Relation (2.71) was used to obtain the second line, while the last line stems for the fact that $\boldsymbol{G}_a^+$ and D_a^- are skew-adjoint (see equation (2.46)). Since (2.74) holds for arbitrary $\{d\boldsymbol{r}_a\}$, the resulting discrete momentum equation is thus (2.70) as before.

As a conclusion, the discrete operators $\boldsymbol{G}_a^+$ and D_a^- are variationally consistent. As a consequence, they ensure energy conservation of an isolated non-dissipative system. This was shown in [92, 147] through the following reasoning: the energy of the system E for such a system is defined as:

$$E = T + V \tag{2.75}$$

so that:

$$E = \sum_{a\in\mathcal{P}} \frac{1}{2} m_a |\boldsymbol{v}_a|^2 - \sum_{a\in\mathcal{P}} m_a \boldsymbol{g}\cdot\boldsymbol{r}_a + \sum_{a\in\mathcal{P}} m_a e_{int,a}(\rho_a) \tag{2.76}$$

The time-derivative of the first two terms $(T + E_{gravity})$ then reads:

$$\begin{aligned} \frac{d(T + E_{gravity})}{dt} &= \sum_{a\in\mathcal{P}} m_a \boldsymbol{v}_a \cdot \left(\frac{d\boldsymbol{v}_a}{dt} + \boldsymbol{g} \right) \\ &= -\sum_{a\in\mathcal{P}} V_a \boldsymbol{v}_a \cdot \boldsymbol{G}_a^+ \{p_b\} \end{aligned} \tag{2.77}$$

the second line corresponding to the definition of the discrete momentum equation (2.70). On the other hand, the time-derivative of the internal energy E_{int} reads:

$$\begin{aligned} \frac{dE_{int}}{dt} &= \sum_{a\in\mathcal{P}} m_a \left(\frac{\partial e_{int}}{\partial \rho} \right)_a \frac{d\rho_a}{dt} \\ &= \sum_{a\in\mathcal{P}} \frac{m_a p_a}{\rho_a^2} \frac{d\rho_a}{dt} \\ &= -\sum_{a\in\mathcal{P}} V_a p_a D_a^- \{\boldsymbol{v}_b\} \\ &= \sum_{a\in\mathcal{P}} V_a \boldsymbol{v}_a \cdot \boldsymbol{G}_a^+ \{p_b\} \end{aligned} \tag{2.78}$$

where the second line is obtained from equation (2.67), the third line from the discrete continuity equation (2.71), and the fourth line from the skew-adjointness of the $\boldsymbol{G}_a^+$ and D_a^- operators. This shows that the time-derivative of the total energy (*i.e.* (2.77) + (2.78)) is equal to zero, so energy is conserved in the absence of viscous forces.

On the other hand, the conservation of angular momentum is ensured for an isolated system as long as the internal forces between particles are oriented along $\boldsymbol{r}_{ab}$, which is true here since $\boldsymbol{\nabla} w_{ab}$ is aligned with $\boldsymbol{r}_{ab}$ [147]. Besides, the linear momentum is conserved when using the antisymmetric operator $\boldsymbol{G}_a^+$, since in that case the action-reaction principle is fulfilled.

However, until now the time has been considered as continuous, while its discretisation may have effects on the conservation properties. In what follows time-stepping schemes are presented that ensure conservation of the required quantities.

2.3.1.2 Time-discretisation

Many methods can be used for the time discretisation of (1.5), but not all of them provide adequate conservation and stability properties. It is possible to show that a time-scheme that derives from a Lagrangian conserves total momentum. Let us consider the following first-order approximation of the time derivative of the position:

$$\frac{\boldsymbol{r}_a^n - \boldsymbol{r}_a^{n-1}}{\delta t} = \boldsymbol{v}_a^n \tag{2.79}$$

At time n, the action associated to the discrete Lagrangian[5] defined in the previous section (equations (2.62)-(2.64)) reads:

$$S\{\boldsymbol{r}_b^n\} = \sum_{b,n} L_b\left(\boldsymbol{r}_b^n, \boldsymbol{r}_b^{n+1}\right)\delta t \tag{2.80}$$

with:

$$\forall(b,n)\ \ L_b\left(\boldsymbol{r}_b^n, \boldsymbol{r}_b^{n+1}\right) = \frac{1}{2\delta t^2} m_b \left|\boldsymbol{r}_b^n - \boldsymbol{r}_b^{n-1}\right|^2 - V_b\{\boldsymbol{r}_c^n\} \tag{2.81}$$

where the particle potential energy $V_b\{\boldsymbol{r}_c^n\}$ is a function of the discrete set of particles positions since the internal energy depends on the density:

$$V_b\{\boldsymbol{r}_c^n\} = m_b\, e_{int,b}(\rho_b = f\{\boldsymbol{r}_c^n\}) - m_b \boldsymbol{g}\cdot\boldsymbol{r}_b^n \tag{2.82}$$

The variation of S due to any infinitesimal variation $\delta\boldsymbol{r}_b^n$ should be equal to zero in a non-dissipative system, which yields the relation[6]:

$$\forall(b,m),\ \ \frac{\partial L_b}{\partial \boldsymbol{x}_1}\left(\boldsymbol{r}_b^n, \boldsymbol{r}_b^{n-1}\right) + \frac{\partial L_b}{\partial \boldsymbol{x}_2}\left(\boldsymbol{r}_b^{n+1}, \boldsymbol{r}_b^n\right) = 0 \tag{2.83}$$

the notations $\dfrac{\partial}{\partial \boldsymbol{x}_1}$ and $\dfrac{\partial}{\partial \boldsymbol{x}_2}$ referring to the partial derivatives of a function with respect to its first and second variable respectively. Given the definition (2.81) of L_b, this yields the following relation:

$$\forall(a,n),\ \ m_a\left(\boldsymbol{v}_a^{n+1} - \boldsymbol{v}_a^n\right) = \boldsymbol{F}_a^n \delta t \tag{2.84}$$

where $\boldsymbol{F}_a^n$ is the total force applied on particle a at time n. From (2.79), (2.84) and (2.71) a time-scheme is obtained that allows to conserve total momentum and which reads:

$$\begin{cases} \dfrac{\boldsymbol{v}_a^{n+1} - \boldsymbol{v}_a^n}{\delta t} = \boldsymbol{F}_a^n \\ \dfrac{\boldsymbol{r}_a^{n+1} - \boldsymbol{r}_a^n}{\delta t} = \boldsymbol{v}_a^{n+1} \\ \dfrac{\rho_a^{n+1} - \rho_a^n}{\delta t} = -\rho_a^n D_a^-\{\boldsymbol{v}_b^{n+1}\} \end{cases} \tag{2.85}$$

where an implicit form of the continuity equation was chosen. The time-scheme used in the present WCSPH simulations (Chapter 4) is this first order sequential scheme. The fact that it derives from an action principle gives to this scheme the same properties as the Hamilton equations and is thus called symplectic [87]. It was chosen due to its conservation properties and its simplicity.

Note that starting from an explicit first-order approximation of the time derivative of the position (replacing $\boldsymbol{v}^n$ by $\boldsymbol{v}^{n-1}$ in (2.79)) yields a fully implicit time-scheme. The fully explicit time-scheme (used for instance in [21]) where all the variables at time $n+1$ are computed from the values at time n does not derive from a Lagrangian and thus does not ensure total energy conservation (even

[5] The action S is equal to the integration in time of the Lagrangian.
[6] This relation constitutes the discrete Lagrange equations [116].

with the choice of differential operators mentioned in the previous section) [147].

Using a higher order approximation of the time derivative of the position yields symplectic higher order time-schemes. The leap-frog time-scheme is often used in the SPH literature [103]: it is a second order symplectic time-scheme and is built as:

$$\begin{cases} \dfrac{\boldsymbol{v}_a^{n+1/2} - \boldsymbol{v}_a^n}{\delta t/2} = \boldsymbol{F}_a^n \\ \dfrac{\boldsymbol{r}_a^{n+1} - \boldsymbol{r}_a^n}{\delta t} = \boldsymbol{v}_a^{n+1/2} \\ \rho_a^{n+1} = \displaystyle\sum_{b\in\mathcal{P}} m_b w_{ab}^{n+1} \\ \dfrac{\boldsymbol{v}_a^{n+1} - \boldsymbol{v}_a^{n+1/2}}{\delta t/2} = \boldsymbol{F}_a^{n+1} \end{cases} \tag{2.86}$$

where the density interpolation (2.68) has been used in place of (2.71). Using such a scheme may improve the WCSPH results compared to the results presented in Chapter 4, although comparisons between a leap-frog and ISPH are not provided in this work. Note that there exists other time-stepping schemes that do not derive from a variational principle. In particular, non-symplectic Runge-Kutta schemes of 3rd or 4th order are often used in the literature (see *e.g.* [108]).

2.3.1.3 Numerical stability

Restrictions on the time-step size must be enforced in order to ensure numerical stability. Due to the complexity of theoretical stability analysis in SPH, empirical conditions on the time-step size are usually applied, inspired from the mesh-based methods [104]. The first one is the Courant-Friedrichs-Levy (CFL) condition, which ensures that the time-step remains lower than the maximal convection time on the smoothing length h during the simulation. Moreover, a condition relative to the viscous forces must be enforced. The time-step is then set through the following relation:

$$\delta t = \min\left(C_{CFL}\frac{h}{c_0}, C_{visq}\frac{h^2}{\nu}\right) \tag{2.87}$$

Recall that c_0 is the numerical speed of sound. The coefficients $C_{CFL} = 0.4$ and $C_{visq} = 0.125$ were determined based on numerical studies. The time-step size was set through equation (2.131) in this work, though it is interesting to bear in mind that a theoretical stability analysis of the WCSPH equations can be performed in arbitrary space dimension for unbounded flows [149]. The Neumann approach can be applied to the SPH equations considering the continuous SPH interpolants for the analysis, which yields a condition on the time-step size. This will be quickly exposed in what follows, more details being available in [149]. Let us consider the first order symplectic scheme (2.85) presented in the previous section. We consider a reference state where $\boldsymbol{v} = \boldsymbol{cst}$ and $\rho = \rho_0$, then search for small arbitrary perturbations $\delta\boldsymbol{r}$, $\delta\rho$ and $\delta\boldsymbol{v}$ to the fields. By

linearising (2.85) it is possible to show that the latter obeys the following set of equations:

$$\begin{aligned}
\frac{\delta \boldsymbol{v}_a^{n+1} - \delta \boldsymbol{v}_a^n}{\delta t} &= -\frac{c_0^2}{\rho_0} \int_\Omega \left[\delta \rho_a^n + \delta \rho(\boldsymbol{r}')^n\right] \boldsymbol{\nabla} w(\boldsymbol{r}_a - \boldsymbol{r}') d\boldsymbol{r}' \\
&\quad + 2\nu \int_\Omega \left[\delta \boldsymbol{v}_a^n + \delta \boldsymbol{v}(\boldsymbol{r}')^n\right] \frac{\boldsymbol{r}_a - \boldsymbol{r}'}{|\boldsymbol{r}_a - \boldsymbol{r}'|} \cdot \boldsymbol{\nabla} w(\boldsymbol{r}_a - \boldsymbol{r}') d\boldsymbol{r}' \\
\frac{\delta \boldsymbol{r}_a^{n+1} - \delta \boldsymbol{r}_a^n}{\delta t} &= \delta \boldsymbol{v}_a^{n+1} \\
\frac{\delta \rho_a^{n+1} - \delta \rho_a^n}{\delta t} &= \rho_0 \int_\Omega \left[\delta \boldsymbol{v}_a^{n+1} + \delta \boldsymbol{v}(\boldsymbol{r}')^{n+1}\right] \boldsymbol{\nabla} w(\boldsymbol{r}_a - \boldsymbol{r}') d\boldsymbol{r}'
\end{aligned} \tag{2.88}$$

(see the Appendix A in [149] for more details about the calculations). Let us consider the following space-periodic fluctuations:

$$\begin{aligned}
\delta \boldsymbol{v}_a^n &= c_0 \boldsymbol{V}(t^n) \exp\left(-i \boldsymbol{K} \cdot \boldsymbol{r}_a\right) \\
\delta \boldsymbol{r}_a^n &= \sigma \boldsymbol{R}(t^n) \exp\left(-i \boldsymbol{K} \cdot \boldsymbol{r}_a\right) \\
\delta \rho_a^n &= \rho_0 R(t^n) \exp\left(-i \boldsymbol{K} \cdot \boldsymbol{r}_a\right)
\end{aligned} \tag{2.89}$$

with $\boldsymbol{K}$ a wavevector and σ the kernel standard deviation, defined by:

$$\sigma^2 = \frac{1}{d} \int_\Omega \tilde{r}^2 w(\tilde{r}) d\tilde{\boldsymbol{r}} \tag{2.90}$$

Substituting the fluctuations in (2.88) with (2.89) and transforming the kernel gradient integrals into Fourier transforms of the kernel yields, after simplifications:

$$\begin{cases}
\dfrac{\boldsymbol{V}(t^{n+1}) - \boldsymbol{V}(t^n)}{\delta t} = \dfrac{i c_0}{\sigma} \widehat{w}(K^*) R(t^n) \boldsymbol{K}^* - \dfrac{\nu}{\sigma^2} F_2(K^*) \boldsymbol{V}(t^n) \\
\dfrac{\boldsymbol{R}(t^{n+1}) - \boldsymbol{R}(t^n)}{\delta t} = \dfrac{c_0}{\sigma} \boldsymbol{V}(t^{n+1}) \\
\dfrac{R(t^{n+1}) - \rho_a(t^n)}{\delta t} = \dfrac{i c_0}{\sigma} \widehat{w}(K^*) \boldsymbol{K}^* \cdot \boldsymbol{U}(t^{n+1})
\end{cases} \tag{2.91}$$

where $\widehat{w}$ is the Fourier transform of the kernel function, $\boldsymbol{K}^* \doteqdot \sigma \boldsymbol{K}$ is the dimensionless wave vector and $K^* = |\boldsymbol{K}^*|$ is the dimensionless wave number. F_2 is a function defined by:

$$F_2(K^*) \doteqdot 2\sigma^2 \int_\Omega [\exp(-i \boldsymbol{K} \cdot \tilde{\boldsymbol{r}} - 1] \frac{\tilde{\boldsymbol{r}}}{\tilde{r}} \cdot \boldsymbol{\nabla} w(\tilde{r}) d\tilde{\boldsymbol{r}} \tag{2.92}$$

Now, one may search for a wave-like solution where:

$$\begin{aligned}
\boldsymbol{V}(t) &= \boldsymbol{V}_0 \exp(i\omega t) \\
\boldsymbol{R}(t) &= \boldsymbol{R}_0 \exp(i\omega t) \\
R(t) &= R_0 \exp(i\omega t)
\end{aligned} \tag{2.93}$$

with ω a complex angular frequency. (2.91) then reads:

$$\begin{cases} \dfrac{\chi-1}{\delta t}\boldsymbol{V}_0 = \dfrac{ic_0}{\sigma}\widehat{w}(K^*)R_0\boldsymbol{K}^* - \dfrac{\nu}{\sigma^2}F_2(K^*)\boldsymbol{V}_0 \\ \dfrac{\chi-1}{\delta t}\boldsymbol{R}_0 = \chi\dfrac{c_0}{\sigma}\boldsymbol{V}_0 \\ \dfrac{\chi-1}{\delta t}R_0 = \chi\dfrac{ic_0}{\sigma}\widehat{w}(K^*)\boldsymbol{K}^*\cdot\boldsymbol{U}_0 \end{cases} \tag{2.94}$$

where $\chi \doteq \exp(i\omega\delta t)$ is the wave amplification factor: the numerical wave is multiplied by the complex number χ at each iteration, so that a stability condition is $|\chi| \leq 1$. From (2.94) the following equation is found:

$$[\chi - 1 + C_\nu F_2(K^*)](\chi-1){K^*}^2\boldsymbol{V}_0 = -\chi C^2 F_1(K^*)(\boldsymbol{K}^*\otimes\boldsymbol{K}^*)\boldsymbol{V}_0 \tag{2.95}$$

with $F_1(K^*) \doteq [K^*\widehat{w}(K^*)]^2$ and:

$$\begin{aligned} C &\doteq \frac{c_0\delta t}{\sigma} \\ C_\nu &\doteq \frac{\nu\delta t}{\sigma^2} = \frac{C}{Re_0} \end{aligned} \tag{2.96}$$

Re_0 being a numerical Reynolds number defined through $Re_0 \doteq \dfrac{c_0\sigma}{\nu}$. The eigenvalues of $\boldsymbol{K}^*\otimes\boldsymbol{K}^*$ are 0 and ${K^*}^2$. The eigenvalue ${K^*}^2$ yields the relation:

$$\chi^2 - \left[2 - C^2F_1(K^*) - C_\nu F_2(K^*)\right]\chi + 1 - C_\nu F_2(K^*) = 0 \tag{2.97}$$

The roots χ of this second order polynomial should have a modulus lower than one, which yields the condition:

$$C \leq \sqrt{2\min_{K^*}\frac{2 - C_\nu F_2(K^*)}{F_1(K^*)}} \tag{2.98}$$

The functions F_1 and F_2 only depend on the kernel function and can be analytically calculated, so that this condition defines a stability domain for C and C_ν.

On the other hand, the zero eigenvalue of $\boldsymbol{K}^*\otimes\boldsymbol{K}^*$ yields a condition that is always verified provided (2.98) is satisfied. Plotting the stability domain in the (C, C_ν)-plane for various kernels (among the classical SPH kernels: Gaussian, B-splines, Wendland kernels) showed that the stability domain is almost independent of the kernel choice for a given space discretisation σ. It was also shown in [149] that computing the density through an SPH interpolation (2.68) or from an SPH divergence of the velocity (2.71) does not affect the stability domain, neither the use of different SPH divergence and gradient operators in the SPH equations. Though, it was shown that the use of a Morris Laplacian (2.57) yields a larger stability domain than the one proposed by Monaghan & Gingold [101] (see end of section 2.2.3), especially for small values of Re_0.

Changing the time integration scheme changes the condition (2.98) and it was shown in [149] that

this has a great influence on the stability domain. Using a fully explicit scheme as in [21] yields a much reduced stability domain compared to the first and second order symplectic schemes (2.85) and (2.86), especially at high Reynolds numbers: the stability domain was shown to tend to zero when increasing the Reynolds number. This is an additional reason why the use of a fully explicit scheme is not recommended, besides the fact that is is not symplectic. On the other hand, numerical experiments seem to show that it is not possible to perform simulations at arbitrarily large Reynolds numbers with the first order symplectic scheme used in this work (2.85). The only scheme, among those tested in [149], which made it possible to perform stable simulations with zero viscosity, is the second order symplectic leap-frog scheme (2.86).

The formula (2.98) was proved to work perfectly for unbounded flows [149]. However, it does not hold anymore in the presence of walls or of a free-surface, neither in case of unsteady flows. This is why the time-step criterion given by equation (2.131) is usually used for SPH simulations, and was used in this work. Nevertheless, the abovementioned analysis helps understanding how the numerical stability acts in SPH.

2.3.1.4 Main drawbacks of WCSPH

The classical WCSPH method yields noisy pressure fields, due to the fact that the pressure is a function of a high power of the density through the equation of state. Small errors on the density then yield very large errors on the pressure. This issue led to the development of the incompressible schemes, which will be described in the next section. Though, many works have aimed at smoothing the pressure field in WCSPH, starting from the introduction of an artificial viscosity [102]. Most techniques for pressure smoothing now apply a diffusion term in the density computation, which can take different forms. Ferrari *et al.* [36] introduced such a diffusion term based on the theory of Riemann solvers introduced by Vila for SPH [145]. They used an approximate Riemann solver to obtain the following continuity equation:

$$\frac{d\rho_a}{dt} = \sum_{b\in\mathcal{P}} m_b \left(\boldsymbol{v}_{ab} + d_{ab} \frac{\boldsymbol{r}_{ab}}{r_{ab}} \frac{\rho_{ab}}{\rho_a} \right) \cdot \boldsymbol{\nabla} w_{ab} \tag{2.99}$$

with $d_{ab} = \max\left(d_a, d_b\right)$ where:

$$d_a = c_0 \sqrt{\left(\frac{\rho_a}{\rho_0}\right)^{\xi-1}} \tag{2.100}$$

The first term in (2.99) is the traditional continuity equation in SPH (2.71), while the second term corresponds to a density diffusion, which will be referred to as the Ferrari density diffusion in what follows. In [95], a very similar formulation was used based on the idea that the numerical fluctuations tend to turn a laminar flow into a “turbulent” flow. Thus, assuming the numerical noise is isotropic it is possible to recover a laminar flow by modelling the numerical noise based on

the concept of eddy diffusivity. This led to the following continuity equation:

$$\frac{d\rho_a}{dt} = \sum_{b\in\mathcal{P}} m_b \left(\boldsymbol{v}_{ab} + (K_a + K_b)\frac{\boldsymbol{r}_{ab}}{r_{ab}}\frac{\rho_{ab}}{\rho_a} \right) \cdot \boldsymbol{\nabla} w_{ab} \tag{2.101}$$

where K is a diffusivity computed through a mixing length model with $L_m \sim \frac{L}{10}$ and $Ma = 0.1$:

$$K = \frac{L}{\delta r}\frac{c_0 \delta r}{10^3} \tag{2.102}$$

This formulation was shown to dissipate less energy than the one proposed by Ferrari *et al.*. Many other formulations were introduced in order to smooth the pressure field in WCSPH, an example is the δ-SPH model [85]. A review of the three most used density corrections was provided by Antuono *et al.* [7]. Note that the Ferrari density correction was shown in [6] to be inconsistent at the free-surface. The diffusion technique used in the simulations presented in Chapter 4 is the one proposed by Ferrari *et al.* (2.99), but in the second term under the sum ρ_{ab} is replaced by $\rho_{ab} - \frac{\rho_0 g}{c_0^2}(z_b - z_a)$. This was shown to significantly improve the results on free-surface flows in [92].

Another issue with the WCSPH schemes on confined flows is that the use of a background pressure is necessary in order to ensure the stability of the simulations. A common way of imposing a background pressure is to modify the equation of state (1.6) so as to have:

$$p = \frac{\rho_0 c_0^2}{\xi}\left(\left(\frac{\rho}{\rho_0} \right)^{\xi} - 0.5 \right) \tag{2.103}$$

In this way the background pressure is equal to $0.5\frac{\rho_0 c_0^2}{\xi}$. Without the use of a background pressure, voids tend to form in the flow, making the simulations unstable. It was observed that the choice of its value influences the results in a significant way: larger values of the background pressure tend to reduce the accuracy in the fields prediction. There is thus a balance to find so that the simulation is stable and the results as accurate as possible. Besides, the background pressure was shown to influence the stability domain found in the previous section [149]: larger values of the background pressure tend to reduce the stability domain.

2.3.2 Truly incompressible SPH

While the previous section focused on the resolution of the weakly-compressible Navier–Stokes equations (1.5), in this section SPH models for solving the incompressible Navier–Stokes equations (1.7) through projection methods are considered. This kind of SPH model is called incompressible SPH (ISPH). Several projection methods were proposed in SPH, all of them being pressure-correction schemes. In SPH the incompressibility can be seen as the nullity of the velocity divergence or as the constancy of the density field when estimated through the SPH interpolation.

This led to two methods corresponding to a Chorin and Temam projection scheme but with different pressure Poisson equations, depending on whether the incompressibility constraint is imposed on the velocity or on the interpolated density. A third method was introduced afterwards that combines these two schemes. On the other hand, a rotational pressure-correction scheme was also proposed for SPH. These methods are detailed in what follows.

2.3.2.1 The Chorin projection method in SPH

Classical Chorin projection method in SPH:

The first adaptation of a projection scheme to SPH was proposed by Cummins & Rudman in 1999 [19]. It corresponds to the Chorin and Temam algorithm presented in section 1.1.2.2, with an explicit viscous term in the first substep. Later on, this algorithm was improved by Lee *et al.* [72] and comparisons were done with WCSPH methods, showing that ISPH provides better pressure fields than WCSPH. Based on the continuous formulation of the projection method proposed by Chorin and Temam (equations (1.19) to (1.21)), a laminar SPH projection method can be written as:

$$\begin{cases} \dfrac{\tilde{\boldsymbol{v}}_a^{n+1} - \boldsymbol{v}_a^n}{\delta t} = \boldsymbol{L}_a\{\nu, \boldsymbol{v}_b^n\} + \boldsymbol{g} \\ L_a\{p_b^{n+1}\} = \dfrac{\rho}{\delta t} D_a^-\{\tilde{\boldsymbol{v}}_b^{n+1}\} \\ \dfrac{\boldsymbol{v}_a^{n+1} - \tilde{\boldsymbol{v}}_a^{n+1}}{\delta t} = -\dfrac{1}{\rho}\boldsymbol{G}_a^+\{p_b^{n+1}\} \end{cases} \tag{2.104}$$

Recall that the discrete operators are defined by (2.39), (2.41), (2.58) and (2.59). The particle subscripts were dropped for the density since in this scheme it is considered as invariant. The position of the particles at the next time-step is then calculated through a second order time marching scheme:

$$\boldsymbol{r}_a^{n+1} = \boldsymbol{r}_a^n + \delta t \left(\frac{\boldsymbol{v}_a^{n+1} + \boldsymbol{v}_a^n}{2} \right) \tag{2.105}$$

The choice of skew-adjoint gradient and divergence operators in (2.104) is important (see section 2.2.2.3), which is why the $\boldsymbol{G}_a^+$ and D_a^- operators are used. This was also the case in the ISPH model tested by Lee *et al.* [72]. The choice of the SPH Laplacian operator is also crucial. Cummins & Rudman tested an exact operator ($L_a = D_a\{\boldsymbol{G}_b\}$) implying a double summation over the neighbours (equation (2.52)). This led to spurious checker-board effects due to the collocation of the pressure and velocity computations and could hamper the linear solver convergence (see section 2.3.2.3). The same problem is encountered with mesh-based collocated methods [123, 126]. Thus, an approximate projection is performed through the use of the Morris Laplacian (2.58).

It was shown that this method provides accurate and smooth pressure fields, but it presents instabilities since it leads to highly anisotropic and inhomogeneous particles arrangements which finally causes blowing-up. An example of this phenomenon is provided Figure 2.5 on the Taylor-Green

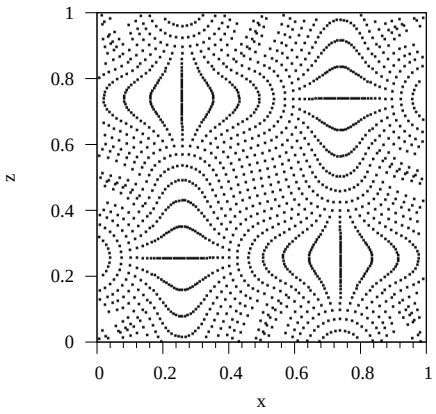


Figure 2.5: An example of highly anisotropic and inhomogeneous particles arrangement on the Taylor-Green vortices case with a classical Chorin projection method adapted to SPH [76].

vortices case [76]. This kind of instability can be avoided through the use of the constant density variant of the ISPH method [129]. However, stabilising methods for this classical Chorin scheme were also introduced. Xu *et al.* [155] proposed a particle shift based on a Fick law of diffusion so as to enforce a homogeneous arrangement of particles in the medium. This method was later improved by Lind *et al.* [76] who also proposed a treatment for free-surface flows. The particles positions are slightly shifted of $\delta \boldsymbol{r}_a$ at each time-step with:

$$\delta \boldsymbol{r}_a = -C_{shift} h^2 \boldsymbol{\nabla} C_a \tag{2.106}$$

where $\boldsymbol{\nabla} C_a$ is a concentration gradient computed as:

$$\boldsymbol{\nabla} C_a \approx \boldsymbol{G}_a\{1\} = \sum_{b \in \mathcal{P}} V_b \boldsymbol{\nabla} w_{ab} \tag{2.107}$$

In [76] the coefficient C_{shift} was taken equal to 0.5 with a 5th order B-spline kernel function. Lind *et al.* observed that with the kernel they used (a 5th order B-spline), defining the concentration gradient through (2.107) was not sufficient to avoid particle clustering. This is why they modified the concentration gradient based on the work by Monaghan [100] that aimed at reducing the so-called tensile instability in SPH. Their concentration gradient is then computed through:

$$\boldsymbol{\nabla} C_a = \sum_{b \in \mathcal{P}} V_b \left(1 + f_{ab}\right) \boldsymbol{\nabla} w_{ab} \tag{2.108}$$

where:

$$f_{ab} = R \left(\frac{w_{ab}}{w(\delta r)} \right)^n \tag{2.109}$$

It was shown in [25] that the 5th order Wendland kernel avoids particle clustering so that this

modification is not necessary with that kernel. On the other hand, close to the free-surface applying the particle shift would lead to a movement of the particles towards the empty space across it. Lind *et al.* then proposed to compute the diffusion through:

$$\delta \boldsymbol{r} = -C_{shift} h^2 \left(\frac{\partial C}{\partial s} \boldsymbol{s} + \alpha \left(\frac{\partial C}{\partial n} - \beta \right) \boldsymbol{n} \right) \tag{2.110}$$

instead of (2.106), where $\boldsymbol{s}$ and $\boldsymbol{n}$ are local tangent and normal and unit vectors to the free-surface. β is a reference value for the concentration gradient at the free-surface and $\alpha \in [0, 1]$ controls the diffusion amplitude along the normal to the free-surface. In their work it was set to 0 so as to have no normal component of the position shift close to the free-surface. Note that the concentration gradient (2.107) can be used to approximate $\boldsymbol{n} = \frac{\boldsymbol{\nabla} C}{\|\nabla C\|}$, so that $\frac{\partial C}{\partial s} = 0$. In this case the free-surface adaptation of the shift proposed by Lind *et al.* is thus equivalent to switching off the shift close to the free-surface. Once the particles have been moved by $\delta \boldsymbol{r}$, the corresponding convection term must be added to the velocity as well as all relevant quantities like the temperature, k and ϵ. For example, for the velocity this reads:

$$\boldsymbol{v}^{n+1} \leftarrow \boldsymbol{v}^{n+1} + \boldsymbol{\nabla} \boldsymbol{v}^{n+1} \cdot \delta \boldsymbol{r} + \mathcal{O}(\delta \boldsymbol{r}^2) \tag{2.111}$$

for each a, where $\boldsymbol{\nabla} \boldsymbol{v}^{n+1}$ is computed through a symmetric SPH gradient (2.37).

Finally, it should be noted that for all the ISPH methods based on this algorithm [19, 72, 76, 155] the pressure wall boundary condition was a homogeneous Neumann. Since these methods were based on ghost or dummy particles technique, this condition was not imposed exactly, but through a mirroring of the pressure field (see section 2.4). As explained in section 1.1.2, this kind of boundary conditions leads to a numerical boundary layer that prevents the scheme from reaching the expected accuracy.

Constant-density variant of the Chorin projection scheme:

This method was proposed by Shao & Lo in 2003 for SPH [129] and consists in computing the density field through an SPH interpolation and ensuring that it remains constant. To our knowledge, this algorithm was first proposed for MPS (Moving Particle Semi-Implicit method), which is very similar to SPH (see *e.g.* Koshizura *et al.* [64], Souto-Iglesias *et al.* [132]). The estimated velocity is computed as:

$$\frac{\tilde{\boldsymbol{v}}_a^{n+1} - \boldsymbol{v}_a^n}{\delta t} = \boldsymbol{L}_a\{\nu, \boldsymbol{v}_b^n\} + \boldsymbol{g} \tag{2.112}$$

This predicted velocity field is then used to update the particles positions:

$$\tilde{\boldsymbol{r}}_a^{n+1} = \boldsymbol{r}_a^n + \delta t \tilde{\boldsymbol{v}}_a^{n+1} \tag{2.113}$$

An intermediate density field $\tilde{\rho}_a^{n+1}$ is computed through the classical SPH interpolation of the

density:

$$\tilde{\rho}_a^{n+1} = \sum_{b \in \mathcal{P}} m_b w_{ab} \tag{2.114}$$

Supposing that the estimated density satisfies the continuity equation, we have:

$$\frac{\tilde{\rho_a}^{n+1} - \rho_a^n}{\delta t} = -\rho_a^n D_a^+ \{\tilde{\boldsymbol{v}}_b^{n+1}\} \tag{2.115}$$

The density-invariance gives $\rho_a^n = \rho$ where ρ is the reference density, and substituting (2.115) in (1.21), a new Poisson equation is obtained:

$$L_a \left\{ \frac{1}{\tilde{\rho}_b^{n+1}}, p_b^{n+1} \right\} = \frac{\rho - \tilde{\rho}_a^{n+1}}{\rho \delta t^2} \tag{2.116}$$

Once again a homogeneous Neumann condition is imposed on the pressure at the solid walls. The velocity field is then corrected through the second part of the momentum equation:

$$\frac{\boldsymbol{v}_a^{n+1} - \tilde{\boldsymbol{v}}_a^{n+1}}{\delta t} = -\frac{1}{\tilde{\rho}_a^{n+1}} \boldsymbol{G}_a^+ \{p_b^{n+1}\} \tag{2.117}$$

Finally the new position of the particles is computed through the same second order time marching scheme as previously:

$$\boldsymbol{r}_a^{n+1} = \boldsymbol{r}_a^n + \delta t \left(\frac{\boldsymbol{v}_a^{n+1} + \boldsymbol{v}_a^n}{2} \right) \tag{2.118}$$

Note that in the algorithm proposed by Shao & Lo [129] the 1st order differential operators were both antisymmetric and thus not skew-adjoint (D_a^+ and $\boldsymbol{G}_a^+$). Moreover, these operators are not zero order consistent. It was shown that this method does not present the instability problem of the divergence-free projection method, but it provides noisier pressure fields, which makes it less attractive compared to WCSPH methods [155].

Divergence-free and constant density variant of the Chorin projection scheme:

A third Chorin-type projection method for SPH was proposed bu Hu & Adams [52], which consists of a combination of the previous two methods. The principle is to split the time-step in two and to solve a Poisson equation for each half, thus imposing both a divergence-free velocity field and a constant density. Though, it does not prove necessary to solve the two Poisson equations at each time-step. To determine if such a process is necessary, at each time-step an estimation of the density is computed through an SPH interpolation for each particle. If the relative difference between this computed density and the reference density exceeds a user-defined criterion (for example 1%), the two Poisson equations are solved.

The first step is to compute a predicted velocity for the first half of the time-step and to update the

positions:

$$\begin{aligned}\tilde{\boldsymbol{v}}_a^{n+\frac{1}{2}} &= \boldsymbol{v}_a^n + \frac{\delta t}{2}\left(\boldsymbol{L}_a\{\nu, \boldsymbol{v}_b^n\} + \boldsymbol{g}\right)\\ \tilde{\boldsymbol{r}}_a^{n+1} &= \boldsymbol{r}_a^n + \tilde{\boldsymbol{v}}_a^{n+\frac{1}{2}}\delta t\end{aligned} \tag{2.119}$$

Then, the intermediate density is computed through:

$$\tilde{\rho}_a^{n+1} = \sum_{b\in\mathcal{P}} m_b w_{ab} \tag{2.120}$$

and the density-invariant Poisson equation is solved:

$$L_a\left\{\frac{1}{\tilde{\rho}_b^{n+1}}, p_b^{n+1}\right\} = 2\,\frac{\rho - \tilde{\rho}_a^{n+1}}{\rho\delta t^2} \tag{2.121}$$

After that, the velocity is corrected through:

$$\boldsymbol{v}_a^{n+1/2} = \tilde{\boldsymbol{v}}_a^{n+1/2} - \frac{\delta t}{2\rho}\boldsymbol{G}_a^+\{p_b^{n+1/2}\} \tag{2.122}$$

and the position is updated:

$$\boldsymbol{r}_a^{n+1} = \tilde{\boldsymbol{r}}_a^{n+1} + \boldsymbol{v}_a^{n+\frac{1}{2}}\delta t \tag{2.123}$$

This ends the first half of the time-step, followed by a divergence-free projection algorithm for the second half with first a velocity prediction:

$$\tilde{\boldsymbol{v}}_a^{n+1} = \tilde{\boldsymbol{v}}_a^{n+\frac{1}{2}} + \frac{\delta t}{2}\left(\boldsymbol{L}_a\{\nu, \boldsymbol{v}_b^n\} + \boldsymbol{g}\right) \tag{2.124}$$

Then the second pressure Poisson equation is solved:

$$L_a\{p_b^{n+1}\} = \frac{2\rho}{\delta t}D_a^-\{\tilde{\boldsymbol{v}}_b^{n+1}\} \tag{2.125}$$

and the velocity is corrected:

$$\boldsymbol{v}_a^{n+1} = \tilde{\boldsymbol{v}}_a^{n+1} - \frac{\delta t}{2\rho}\boldsymbol{G}_a^+\{p_b^{n+1}\} \tag{2.126}$$

In case the relative difference between the interpolated density (2.120) and the reference density ρ is low enough at the beginning of the time-step, only the divergence-free projection method is applied. It was shown that this method provides smooth and accurate pressure fields and is stable, but the computational cost is relatively high due to the resolution of two pressure Poisson equations. This method allows to reorder the particles positions in a more consistent way than with the particle shift proposed in [76, 155]. As before, a homogeneous Neumann condition is applied on the pressure at solid walls.

2.3.2.2 Rotational pressure-correction method in SPH

In 2012, Hosseini *et al.* [51] proposed a rotational pressure-correction scheme in SPH. This allowed to impose a non-homogeneous Neumann condition on the pressure at solid walls (see section 1.1.2.2). Their SPH algorithm reads:

$$\begin{cases} \dfrac{1}{2\delta t}\left(3\tilde{\boldsymbol{v}}_a^{n+1} - 4\boldsymbol{v}_a^n + \boldsymbol{v}_a^{n-1}\right) - \boldsymbol{L}_a\{\nu, \tilde{\boldsymbol{v}}_b^{n+1}\} = \boldsymbol{g} - \dfrac{1}{\rho}\boldsymbol{G}_a^{-,1}\{p_b^n\} \\ L_a\{\phi_b^{n+1}\} = \dfrac{3\rho}{2\delta t} D_a^{-,1}\{\tilde{\boldsymbol{v}}_b^{n+1}\} \\ \dfrac{1}{2\delta t}\left(3\boldsymbol{v}_a^{n+1} - 3\tilde{\boldsymbol{v}}_a^{n+1}\right) + \dfrac{1}{\rho}\boldsymbol{G}_a^{-,1}\{\phi_b^{n+1}\} = 0 \end{cases} \tag{2.127}$$

where ϕ_a^{n+1} is defined as:

$$\phi_a^{n+1} = p_a^{n+1} - p_a^n + D_a^{-,1}\{\mu, \tilde{\boldsymbol{v}}_b^{n+1}\} \tag{2.128}$$

Note that it is necessary to have an implicit viscous term in the first substep in order to obtain the consistent pressure wall boundary condition. In the validation cases presented in [51], the viscous term was explicit in the first substep which destroys the advantage of working with this scheme. The authors chose to use renormalised symmetric gradient and divergence operators (2.49), (2.50), which are not skew-adjoint but provide better accuracy. The Laplacian operator they used was a Morris Laplacian and they tried to make it first order consistent by applying the renormalisation matrix of the 1st order differential operators, although it seems better to use the 1st order Laplacian operator proposed by Fatehi & Manzari [32] (see section 2.2.3.2). The stabilisation method based on a particle shift [155] was used.

The main advantages of this methods compared to the ones presented in section 2.3.2.1 are the increased accuracy of the scheme (see section 1.1.2.2), and the fact that it yields a non-homogeneous Neumann condition on the pressure at the walls, which was shown to be more consistent than a homogeneous Neumann condition (see section 1.1.2.2, that was also observed by Hosseini *et al.*). This scheme is slightly more complex than the divergence-free Chorin-type scheme presented in section 2.3.2.1, which yields additional computational time. However, Hosseini *et al.* mentioned the rotational scheme is more robust than a Chorin-type one so they could use a time-step twice as large on a validation case of a flow around a square cylinder.

2.3.2.3 Resolution of the pressure Poisson equation

The pressure Poisson equation (*e.g.* the second line of (2.104) or (2.116)) corresponds to a linear system that can be written as:

$$\boldsymbol{A}\boldsymbol{P} = \boldsymbol{B} \tag{2.129}$$

with $\boldsymbol{A}$ the SPH Laplacian matrix, $\boldsymbol{P}$ the vector of unknowns (the discrete set of particles pressures) and $\boldsymbol{B}$ the right-hand side (divergence of the predicted velocity or $\frac{\rho-\tilde{\rho}_a^{n+1}}{\rho\delta t^2}$). Denoting the matrix indexes as a or b and considering the Morris Laplacian operator (2.58), the entries of matrix $\boldsymbol{A}$ read:

$$\begin{cases} A_{aa} = 2\sum_{b\in\mathcal{P}} V_b \frac{\boldsymbol{r}_{ab}}{r_{ab}^2}\cdot\boldsymbol{\nabla} w_{ab} \\ A_{ab} = -2V_b \frac{\boldsymbol{r}_{ab}}{r_{ab}^2}\cdot\boldsymbol{\nabla} w_{ab} \end{cases} \tag{2.130}$$

Thus, in case each particle has the same volume (which holds with classical boundary conditions, see section 2.4.1), the Laplacian matrix is symmetric. On the other hand, as we will see this is not the case with the algorithm proposed in this work that relies on another type of boundary conditions (see section 3.6). Many methods can be used to solve this system, the most used in the SPH literature being the Bi-CGSTAB [72] and GMRES methods.

2.3.2.4 Numerical stability

Restrictions on the time-step size are necessary in order to ensure the numerical stability of the ISPH schemes, in the same way as for WCSPH. A theoretical analysis of the numerical stability of ISPH has not been provided yet. Though, it would be possible to perform a Neumann stability analysis of the ISPH schemes, similarly to what was presented in section 2.3.1.3 regarding the WCSPH schemes. Anyway, since such analysis does not consider the presence of walls or of a free-surface, empirical conditions would have to be used. Such empirical conditions have been used in all the ISPH models in the literature. The condition relative to viscous forces is unchanged compared to WCSPH, but the CFL condition is modified and the maximum velocity of the flow v_{max} replaces the numerical speed of sound c_0 [19]. The consequence is that the time-step size is larger with ISPH than with WCSPH, since the maximum velocity of the flow is usually ten times smaller than the numerical speed of sound. Thus, the time-step size is determined by the relation:

$$\delta t = \min\left(C_{CFL}\frac{h}{v_{max}}, C_{visq}\frac{h^2}{\nu}\right) \tag{2.131}$$

The value of $C_{visq} = 0.125$ is the same as for WCSPH schemes, but several values of the CFL number C_{CFL} are found in the literature for ISPH. Indeed, while Cummins & Rudman introduced a CFL number equal to 0.25 [19], Lee *et al.* used the value 0.4 [72] (as in WCSPH) while Shao & Lo used 0.1 [129] and Hu & Adams used 0.25 [52]. It was shown on several test-cases in 2-D that the computational times are usually smaller with ISPH than with WCSPH [72, 155]. This shows that the use of larger time steps more than compensates the additional computational effort required by the resolution of the pressure Poisson equation. Note that Hosseini *et al.* give hints that their rotational scheme is more stable than the classical SPH Chorin-type scheme, but they do not provide values for the CFL number [51].

2.4 Wall boundary conditions in SPH

2.4.1 Classical treatment of the wall boundary conditions

2.4.1.1 Particle-based approaches

As mentioned in Section 2.2.1 the presence of domain boundaries makes the SPH interpolation inconsistent in their vicinity, where the kernel properties (2.8) and (2.9) are not satisfied. Besides, close to the boundaries, the surface integral in (2.29) does not cancel so that the formulations (2.39), (2.41) and (2.58) for the SPH differential operators are inaccurate.

A classical way of imposing wall boundary conditions in SPH is to leave these issues unaddressed, but to discretise the boundary through particles, and then impose repulsive forces between boundary particles and what is called herein free particles (particles moving according to the SPH equations). For example in Monaghan's method [103], the repulsive force employed derives from the Lennard-Jones potential. An illustration of this method is provided on Figure 2.6 (a). This method is easy to implement even for complex geometries and is computationally cheap, but leads to spurious behaviours of the particles, as pointed out by Ferrand *et al.* [35] for example. Indeed, none of the consistency issues are addressed and though the impermeability of the walls is ensured, the SPH equations are inaccurately solved close to the boundaries. One effect is that the fluid does not remain still near the walls in a hydrostatic case. Besides, this method makes it difficult - if not impossible - to accurately prescribe Neumann wall boundary conditions, which is a serious issue for a numerical model. In particular, when dealing with ISPH it is necessary to impose a Neumann condition on the pressure, which is probably not possible with this simple technique. Note that an improvement of this technique was proposed by Rogers & Dalrymple [121] with a more physical representation of the particles interactions.

Another very classical technique for modelling boundaries in SPH is the so-called ghost (or dummy) particles technique [118], which exists under many forms. It has been widely used in WCSPH [54] and ISPH formalisms [72, 129]. The idea is to model solid boundaries through particles, and place two or more layers of ghost particles beyond the boundary, so as to fill the void in the particle kernel (see Figure 2.6 (b)). Thus the inconsistency in the SPH interpolation close to the walls is removed (though one must be careful: with only two or even three layers of ghost particles the kernel support may still be truncated, depending on the kind of kernel and on the ratio $\frac{h}{\delta r}$ chosen). The second step then consists in assigning appropriate values to the ghost particles' fields, which is generally challenging. One possibility is to set the ghost particles' velocity to that of the wall (in this way it is easy to represent moving walls). An important feature of these ghost particles is that their density is non-zero (*e.g.* equal to the reference density) so that the density interpolation through the classical discrete SPH interpolation is more consistent close to the boundary. When solving the weakly-compressible form of the Navier–Stokes equations, this generates a repulsive force oriented

from the dummy particles to approaching particles. On the contrary, this force is attractive for a particle moving away from the wall. The closer the particle is to the wall, the higher the value of the force. This reasoning does not hold when dealing with an incompressible SPH model since the density is not computed anymore, but set as constant. As mentioned is section 2.3.2, the usual (but erroneous) wall boundary condition applied on the pressure (or on a modified pressure in the case of the rotational scheme, see section 2.3.2.2) is a homogeneous Neumann condition:

$$\left.\frac{\partial p}{\partial \boldsymbol{n}}\right|_{\partial\Omega_s} = 0 \tag{2.132}$$

One possibility is to consider the wall particles as unknowns in the pressure Poisson equation and to set the dummy particles pressure equal to that of the wall particles, as for example in [129]. Thus, in ISPH the Neumann condition on the pressure is only approximately imposed, the accuracy depending on the space discretisation. This inaccuracy also concerns the imposition of wall functions on the velocity and on the turbulent quantities in turbulence models [148].
Note that the dummy particles placement is very important, in particular when dealing with complex geometries. For example Takeda *et al.* [136] or Yildiz *et al.* [157], among others, proposed methods for the dummy particles placement in case of complex geometries, in particular involving curved walls. Though, the extension of such techniques to 3-D can prove quite complex. Besides, the way the fields are extrapolated from the free particles to the ghost particles may depend on the shape of the wall, as in [51]. Note also that the ghost particles technique, as well as the mirror particles technique which will be presented below, require additional memory space compared to the repulsive forces method, especially in 3-D.

The last main classical technique for boundary modelling in SPH is the mirror particles technique [75], also illustrated on Figure 2.6. This time the wall is not discretised into particles but for each fluid particle a 'mirror' particle is placed across the boundary. When imposing a homogeneous Neumann boundary condition, the values of the field are mirrored across the boundary. With this technique the kernel of the particles close to the walls is not truncated anymore. Though, as in the ghost particles case, the imposition of boundary conditions is inaccurate and imposing non-homogeneous Neumann boundary conditions is problematic. Besides, imposing a Dirichlet condition is not straightforward since the boundary is not discretised into particles. For example, in [134] a Dirichlet wall boundary condition on the temperature is imposed by considering a linear evolution of the temperature between the ghost and free particles (see section 2.8).

Most available ISPH models in the literature are based on ghost particles [51, 72, 76, 129, 155] or mirror particles [110]. Then, the imposition of the homogeneous Neumann wall boundary condition on the pressure (or on the modified pressure in the case of the rotational scheme of section 2.3.2.2) is done by manipulating the relevant entries in the linear system so that the value of the pressure is mirrored across the solid boundary. This is not an exact prescription of Neumann pressure wall boundary condition, and is a serious issue since the proper imposition of pressure boundary condition is crucial when solving the pressure Poisson equation.

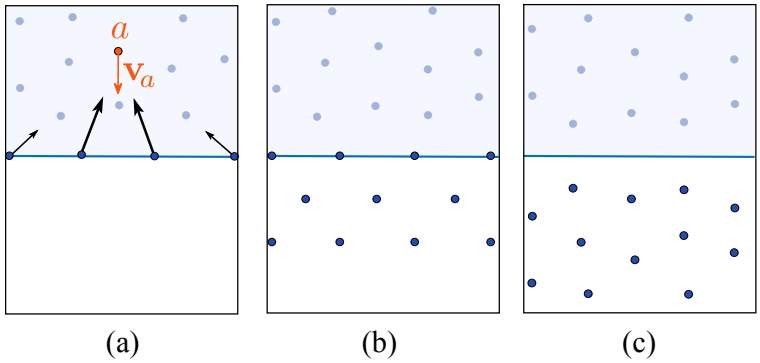


Figure 2.6: Sketch of the main three techniques for wall boundary modelling in SPH. (a): Boundary forces; (b) Dummy particles; (c) Mirror particles.

More generally, the use of approaches that try to manipulate the discretisation (*i.e.* add points inside the wall or carry particle-owned stencils that enrich the discretisation close to the domain boundaries) is questionable since wall-particle properties are irrelevant for wall gradients. This motivated a change in the boundary management, with boundary integral approaches as described in Sections 2.4.1.2 and 2.4.2.

2.4.1.2 Boundary integral approaches

Other methods to model solid boundaries were proposed, that rely on the use of a wall renormalisation factor in the SPH interpolation to restore the normalising property (2.8) in the vicinity of the walls. The continuous SPH interpolation (2.3) is then modified and reads:

$$[A]_c^\gamma(\boldsymbol{r}_a) = \frac{1}{\gamma_a}\int_{\Omega_a} A(\boldsymbol{r}')w(\boldsymbol{r}_a - \boldsymbol{r}')d\boldsymbol{r}' \tag{2.133}$$

where γ_a is a renormalisation factor defined by:

$$\gamma_a = \int_{\Omega\cap\Omega_a} w(\boldsymbol{r}_a - \boldsymbol{r}')d\boldsymbol{r}' \tag{2.134}$$

γ_a is thus equal to 1 far from the boundary and lower than one when the kernel support is truncated. This is illustrated in Figure 2.7. Then the discrete interpolation of a field reads:

$$[A]_d^\gamma(\boldsymbol{r}_a) = \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b A_b w_{ab} \tag{2.135}$$

in place of (2.4). This newly defined SPH interpolation leads to the application of a natural boundary force in the Navier–Stokes equations. Indeed, Kulasegaram *et al.* [65] found that deriving the SPH operators from a variational principle (as showed in section 2.3.1) led to the following modified

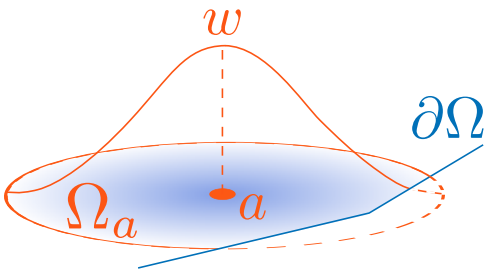


Figure 2.7: Sketch of the kernel truncation. γ_a is the integral under w on the blue area.

definitions of the SPH gradient and divergence operators:

$$\begin{aligned} \boldsymbol{G}_a^K\{A_b\} &= \rho_a \sum_{b\in\mathcal{P}} m_b \left(\frac{A_a}{\gamma_a\rho_a^2} + \frac{A_b}{\gamma_b\rho_b^2}\right)\boldsymbol{\nabla} w_{ab} - \frac{A_a}{\gamma_a}\boldsymbol{\nabla}\gamma_a \\ D_a^K\{\boldsymbol{A}_b\} &= -\frac{1}{\gamma_a\rho_a}\sum_{b\in\mathcal{P}} m_b \boldsymbol{A}_{ab}\cdot\boldsymbol{\nabla} w_{ab} + \frac{1}{\gamma_a}\boldsymbol{A}_a\cdot\boldsymbol{\nabla}\gamma_a \end{aligned} \tag{2.136}$$

where $\nabla\gamma_a$ is the gradient of γ_a. One should compare these formulae to (2.36) and (2.42). The total linear momentum is conserved due to the use of an action principle to define these operators. As a matter of fact, it is easy to check that $\boldsymbol{G}_a^K$ and D_a^K are skew-adjoint (see section 2.2.2.3). However, one may object that in the presence of walls the operators should not be skew-adjoint, since the sum of the terms in (2.46) should give the total pressure work on the boundary (see (1.14)). Besides, in these works the second order operator (Laplacian) was left unchanged. In [35], Ferrand *et al.* proposed a different formulation of the differential operators that addressed these issues. In particular, the Laplacian operator is modified according to the SPH interpolation defined through (2.133). In this framework, the imposition of boundary conditions can be done in a natural way through the boundary term of the new Laplacian operator. This was applied in [35] to the $k-\epsilon$ turbulence model where Neumann boundary conditions could be prescribed exactly on k and ϵ for the first time in SPH, the condition on ϵ being non-homogeneous. With this method the estimation of the fields is very accurate, even close to the walls. From now on these boundary conditions will be referred to as unified semi-analytical wall (USAW) boundary conditions. Their description is the object of the next section.

Before that, let us just evoke here the fact that the computation of γ_a and its gradient appears as an issue in these recent methods. Kulasegaram *et al.* [65] and De Leffe *et al.* [22] proposed approximate methods to calculate the renormalisation factor, but it appeared that an analytical computation of this factor is necessary in order to ensure the walls impermeability with complex geometries. Feldman and Bonet [33] proposed an analytical method for simple wall shapes with applications in 2-D. Later on Ferrand *et al.* [35] proposed a method to compute the gradient of γ_a

analytically in 2-D and then compute γ_a through a dynamic governing equation:

$$\frac{d\gamma_a}{dt} = \boldsymbol{\nabla}\gamma_a \cdot (\boldsymbol{v}_a - \boldsymbol{v}^{wall}) \tag{2.137}$$

where $\boldsymbol{v}^{wall}$ is the wall velocity. This made it possible to model walls with complex shapes in 2-D with sufficiently accurate values for γ_a. These two methods for an analytical computation of γ_a or its gradient are based on the application of the Gauss theorem so that the volume integral (2.134) is reduced to a surface integral, easier to express. Thus, with these methods the domain boundary must be represented as a surface. An extension of the method proposed by Ferrand *et al.* to 3-D was proposed by Mayrhofer *et al.* [92] but is computationally expensive.

In the present work (section 3.8.1), a method is proposed to compute γ_a analytically in 2-D with the possibility to represent arbitrarily complex shapes. This was inspired by the work of Feldman and Bonet [33] and of Ferrand *et al.* [35]. More recently, Violeau *et al.* [150] proposed another method for the analytical computation of γ_a and its gradient in 3-D based on the same technique.

2.4.2 Unified semi-analytical wall boundary conditions

Although the boundary conditions technique detailed below is not yet very much used in SPH, we include it in this literature review because the present work will take advantage of using it. In what follows, these boundary conditions are referred to as the USAW boundary conditions (for Unified Semi-Analytical Wall boundary conditions). Note that the SPH equations remain unchanged with this technique, but the SPH differential operators are modified, which is explained below.

2.4.2.1 Modified SPH interpolation

According to (2.133), the continuous SPH interpolation of the gradient of a field now reads:

$$[\boldsymbol{\nabla}A]_c^{\gamma}(\boldsymbol{r}_a) = \frac{1}{\gamma_a}\int_{\Omega\cap\Omega_a} \boldsymbol{\nabla}A(\boldsymbol{r}_a)w(\boldsymbol{r}_a - \boldsymbol{r}')d\boldsymbol{r}' \tag{2.138}$$

An integration by parts of this integral gives (considering that the normal $\boldsymbol{n}$ is oriented inwards):

$$\begin{aligned}[\boldsymbol{\nabla}A]_c^{\gamma}(\boldsymbol{r}_a) \quad &= \frac{1}{\gamma_a}\int_{\Omega\cap\Omega_a} A(\boldsymbol{r}_a)\boldsymbol{\nabla}w(\boldsymbol{r}_a - \boldsymbol{r}')d\boldsymbol{r}' \\ &-\frac{1}{\gamma_a}\int_{\partial\Omega\cap\Omega_a} A(\boldsymbol{r}_a)w(\boldsymbol{r}_a - \boldsymbol{r}')\boldsymbol{n}(\boldsymbol{r}')d\Gamma'\end{aligned} \tag{2.139}$$

With the USAW method the boundary term is discretised so that the boundary must be represented as a surface. It appeared easier from a computational point of view to split the boundary into a set of boundary elements called herein segments $s \in \mathcal{S}$. These boundary elements are indeed segments in 2-D (see Figure 2.8), and triangles or quadrangles in 3-D. They are not fluid particles and no

mass is attributed to them. This wall discretisation proved practical for the computation of the boundary integrals and allows the representation of complex geometries. Nevertheless, it appeared important to also place fluid particles at the boundary, which proved to increase the accuracy of the modified SPH operators close to the boundary. These boundary fluid particles are called herein vertex particles $v \in \mathcal{V}$. They are located at the extremities of the segments s so that the imposition of Dirichlet and Neumann boundary conditions is not collocated (the Neumann conditions being imposed through the segments). The vertex particles make it possible to efficiently impose Dirichlet boundary conditions. The discretisation of (2.139) then gives the following discrete SPH gradient in place of (2.33):

$$\boldsymbol{G}_a^{\gamma}\{A_b\} = \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b A_a \boldsymbol{\nabla} w_{ab} - \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}} A_s \boldsymbol{\nabla}\gamma_{as} \tag{2.140}$$

where $\boldsymbol{\nabla}\gamma_{as}$ was defined as:

$$\boldsymbol{\nabla}\gamma_{as} = \int_{\partial\Omega_s\cap\Omega_a} w(\boldsymbol{r}_a - \boldsymbol{r}')\boldsymbol{n}(\boldsymbol{r}')d\Gamma' \tag{2.141}$$

with $\partial\Omega_s$ the portion of boundary spanned by the segment s. Note that:

$$\boldsymbol{\nabla}\gamma_a = \sum_{s\in\mathcal{S}} \boldsymbol{\nabla}\gamma_{as} = \int_{\partial\Omega\cap\Omega_a} w(\boldsymbol{r}_a - \boldsymbol{r}')\boldsymbol{n}(\boldsymbol{r}')d\Gamma' \tag{2.142}$$

The set $\mathcal{P}$ in (2.140) represents the set of all fluid particles: free particles $f \in \mathcal{F}$ that move according

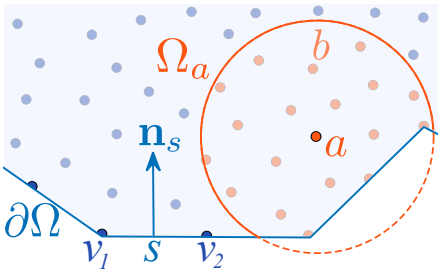


Figure 2.8: Sketch of the continuous medium discretisation with the USAW boundary conditions technique.

to the SPH equations and vertex particles $v \in \mathcal{V}$ that are located at the solid boundary (*i.e.* $\mathcal{P} = \mathcal{F} \cup \mathcal{V}$). Note that in order to have a partition of unity at the initial time, the mass of the vertex particles is set lower than that of the free particles. They are thus truncated particles. Besides, they are Eulerian particles since they do not move according to the SPH equations. Figure 2.8 shows a sketch of the different entities (free particles, vertex particles and wall segments) used in the medium discretisation with the USAW technique. Note that in equation (2.140) an approximation was made when discretising the volumic term (sum over $\mathcal{P}$), as in a classical SPH operator. Though, the discrete boundary term corresponds to a nearly exact discretisation of the continuous one, provided $\boldsymbol{\nabla}\gamma_{as}$ is computed analytically: the only approximations are that the field is considered

as constant along a segment and that the segments constitute a good geometrical description of the wall shape.

As mentioned in the previous section, the quantities $\boldsymbol{\nabla}\gamma_{as}$ are computed through an analytical formula proposed by Ferrand *et al.*, whereas γ_a is computed through a governing equation (2.137). The latter leads to an additional restriction on the time-step size in order to ensure numerical stability. Ferrand *et al.* proposed the following condition:

$$\delta t < C_\gamma \frac{1}{\max\limits_{a\in\mathcal{P}}\{|\boldsymbol{\nabla}\gamma_{as}\cdot\boldsymbol{v}_{as}|\}} \tag{2.143}$$

with $\boldsymbol{v}_{as} = \boldsymbol{v}_a - \boldsymbol{v}_s$ and $C_\gamma = 0.004$ a constant they set through numerical experiments.

2.4.2.2 First order operators with the USAW boundary conditions

In the same way as for the classical SPH operators, it is possible to define symmetric and antisymmetric versions of the SPH wall-renormalised differential operators. This was proposed by De Leffe *et al.* [22] and then by Ferrand *et al.* [35] with a more accurate formulation. The latter then propose the following form for the symmetric renormalised SPH gradient:

$$\boldsymbol{G}_a^{\gamma,-}\{A_b\} = -\frac{1}{\gamma_a\rho_a}\sum_{b\in\mathcal{P}} m_b A_{ab}\boldsymbol{\nabla}w_{ab} + \frac{1}{\gamma_a\rho_a}\sum_{s\in\mathcal{S}}\rho_s A_{as}\boldsymbol{\nabla}\gamma_{as} \tag{2.144}$$

in place of (2.36), while an antisymmetric renormalised SPH gradient is given by:

$$\boldsymbol{G}_a^{\gamma,+}\{A_b\} = \frac{\rho_a}{\gamma_a}\sum_{b\in\mathcal{P}} m_b\left(\frac{A_a}{\rho_a^2}+\frac{A_b}{\rho_b^2}\right)\boldsymbol{\nabla}w_{ab} - \frac{\rho_a}{\gamma_a}\sum_{s\in\mathcal{S}}\rho_s\left(\frac{A_a}{\rho_a^2}+\frac{A_s}{\rho_s^2}\right)\boldsymbol{\nabla}\gamma_{as} \tag{2.145}$$

in place of (2.39). As for classical operators, the antisymmetric gradient is used to compute the pressure gradient while the symmetric operator is used to compute velocity gradients for example.

Besides, a wall-renormalised symmetric SPH divergence operator is given by:

$$D_a^{\gamma,-}\{\boldsymbol{A}_b\} = -\frac{1}{\gamma_a\rho_a}\sum_{b\in\mathcal{P}} m_b\boldsymbol{A}_{ab}\cdot\boldsymbol{\nabla}w_{ab} + \frac{1}{\gamma_a\rho_a}\sum_{s\in\mathcal{S}}\rho_s\boldsymbol{A}_{as}\cdot\boldsymbol{\nabla}\gamma_{as} \tag{2.146}$$

The antisymmetric form of the divergence operator was not used in this work. One may compare (2.145) and (2.146) to the operators (2.136) proposed by Kulasegaram *et al.*.

The operators $\boldsymbol{G}_a^{\gamma,+}$ and $D_a^{\gamma,-}$ are not skew-adjoint anymore, contrary to the ones proposed by Kulasegaram (2.136). Indeed, Mayrhofer *et al.* [95] investigated the properties of the operators (2.145) and (2.146) and showed that they are skew-adjoint in their continuous forms, but they are only approximately skew-adjoint in their discrete form.

2.4.2.3 Second order operator with the USAW boundary conditions

A similar Laplacian operator as the one proposed by Morris *et al.* [104] can be defined in this framework, which was done by Ferrand *et al.* [35]. Let us consider the general case where a non-constant diffusion coefficient B is involved. Recall that the Laplacian is first made symmetric (see section 2.2.3):

$$\nabla \cdot (B\boldsymbol{\nabla} A) = B\boldsymbol{\nabla} A \cdot \boldsymbol{\nabla} 1 + \nabla \cdot (B\boldsymbol{\nabla} A) \tag{2.147}$$

which is the first step to build an antisymmetric divergence operator like (2.42). The continuous SPH interpolation of the right-hand side then reads after integration by parts:

$$\begin{aligned}\left[\nabla \cdot (B\boldsymbol{\nabla} A)\right]_c(\boldsymbol{r}_a) = &\frac{1}{\gamma_a}\int_{\Omega\cap\Omega_a}\left[B(\boldsymbol{r}_a)\boldsymbol{\nabla} A(\boldsymbol{r}_a) + B(\boldsymbol{r}')\boldsymbol{\nabla} A(\boldsymbol{r}')\right]\cdot\boldsymbol{\nabla} w(\boldsymbol{r}_a-\boldsymbol{r}')d\boldsymbol{r}' \\ &-\frac{1}{\gamma_a}\int_{\partial\Omega\cap\Omega_a}\left[B(\boldsymbol{r}_a)\boldsymbol{\nabla} A(\boldsymbol{r}_a) + B(\boldsymbol{r}')\boldsymbol{\nabla} A(\boldsymbol{r}')\right]\cdot\boldsymbol{n}(\boldsymbol{r}')w(\boldsymbol{r}_a-\boldsymbol{r}')d\Gamma'\end{aligned} \tag{2.148}$$

Compared to (2.54), we now added the normalising factor γ_a. The boundary term does not cancel anymore, and the finite difference approximation (2.55) is used to estimate the gradient of the field in the volumic integral, so that the continuous interpolation of the Laplacian reads:

$$\begin{aligned}\left[\nabla \cdot (B\boldsymbol{\nabla} A)\right]_c(\boldsymbol{r}_a) = &\frac{2}{\gamma_a}\int_{\Omega\cap\Omega_a}\overline{B}(\boldsymbol{r}_a,\boldsymbol{r}')\frac{A(\boldsymbol{r}_a)-A(\boldsymbol{r}')}{(\boldsymbol{r}_a-\boldsymbol{r}')^2}(\boldsymbol{r}_a-\boldsymbol{r}')\cdot\boldsymbol{\nabla} w(\boldsymbol{r}_a-\boldsymbol{r}')d\boldsymbol{r}' \\ &-\frac{1}{\gamma_a}\int_{\partial\Omega\cap\Omega_a}\left[B(\boldsymbol{r}_a)\nabla A(\boldsymbol{r}_a) + B(\boldsymbol{r}')\nabla A(\boldsymbol{r}')\right]\cdot\boldsymbol{n}(\boldsymbol{r}')w(\boldsymbol{r}_a-\boldsymbol{r}')d\Gamma'\end{aligned} \tag{2.149}$$

Recall that $\overline{B}(\boldsymbol{r}_a,\boldsymbol{r}')$ is a mean of the diffusion coefficient B between positions $\boldsymbol{r}_a$ and $\boldsymbol{r}'$, and that in the present work the arithmetic mean was used. Then the boundary integral is discretised into its sum over all the segments and the volumic integral is approximated by the summation over the fluid particles:

$$\begin{aligned}L_a^{\gamma}\{B_b,A_b\} &= \frac{2}{\gamma_a}\sum_{b\in\mathcal{P}}V_b\overline{B}_{ab}\frac{A_a-A_b}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\nabla w_{ab} \\ &-\tfrac{1}{\gamma_a}\sum_{s\in\mathcal{S}}\int_s\left[B_a\boldsymbol{\nabla} A_a + B(\boldsymbol{r}')\boldsymbol{\nabla} A(\boldsymbol{r}')\right]\cdot\boldsymbol{n}(\boldsymbol{r}')w(\boldsymbol{r}_a-\boldsymbol{r}')d\Gamma'\end{aligned} \tag{2.150}$$

Supposing that the gradient of the field is constant over each segment and using the definition of $\boldsymbol{\nabla}\gamma_{as}$ and replacing $\overline{B}_{ab}$ by $\frac{B_a+B_b}{2}$ finally gives:

$$L_a^{\gamma}\{B_b,A_b\} = \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}}m_b\frac{B_a+B_b}{\rho_b}\frac{A_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla} w_{ab} - \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}(B_a\boldsymbol{\nabla} A_a + B_s\boldsymbol{\nabla} A_s)\cdot\boldsymbol{\nabla}\gamma_{as} \tag{2.151}$$

This Laplacian operator can also be applied to a vector, which reads:

$$\boldsymbol{L}_a^\gamma\{B_b,\boldsymbol{A}_b\}=\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}}m_b\frac{B_a+B_b}{\rho_b}\frac{\boldsymbol{A}_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab}-\frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}(B_a\boldsymbol{\nabla}\boldsymbol{A}_a+B_s\boldsymbol{\nabla}\boldsymbol{A}_s)\cdot\boldsymbol{\nabla}\gamma_{as} \quad (2.152)$$

to be compared to (2.59). In case there is no diffusion coefficient, the Laplacian operator is written as:

$$L_a^\gamma\{A_b\}=\frac{2}{\gamma_a}\sum_{b\in\mathcal{P}}\frac{m_b}{\rho_b}\frac{A_a-A_b}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab}-\frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}(\boldsymbol{\nabla}A_a+\boldsymbol{\nabla}A_s)\cdot\boldsymbol{\nabla}\gamma_{as} \quad (2.153)$$

The boundary term of the Laplacian operator will appear frequently in what follows. It is defined as:

$$L_a^{bound}\{B_b,A_b\}=-\frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}(B_a\boldsymbol{\nabla}A_a+B_s\boldsymbol{\nabla}A_s)\cdot\boldsymbol{\nabla}\gamma_{as} \quad (2.154)$$

The same notations as for the Laplacian operator are used in case there is no diffusion coefficient ($L_a^{bound}\{A_b\}$) and for the Laplacian of a vector ($\boldsymbol{L}_a^{bound}\{B_b,\boldsymbol{A}_b\}$). It was chosen not to approximate the gradient of the field in these boundary terms since at the wall its value is known when imposing a Neumann boundary condition. Note that $(B\boldsymbol{\nabla}A)$ is the diffusive flux of A. It is important to note that with this definition, L_a^γ now depends not only on the discrete set $\{A_b\}$, but also on $B_a\boldsymbol{\nabla}A_a$ and the set $\{B_s\boldsymbol{\nabla}A_s\}$. Thus, in that case the Neumann condition is directly applied in L_a^{bound}. In section 3.3, a detailed explanation of how the wall boundary conditions are imposed with the USAW technique is given. Though it is interesting to note that with this definition of the Laplacian operator, it is possible to accurately impose arbitrary Neumann boundary conditions on the fields. When dealing with an ISPH model this is particularly interesting since, as we saw, it is important to impose a non-homogeneous Neumann boundary condition on the pressure in order to obtain a consistent projection method. In 2012, Macià *et al.* [81] applied the USAW boundary conditions to ISPH, but they focused on the prescription of Dirichlet boundary conditions on the pressure field, which is not appropriate in dynamic cases. Moreover, they did not present any applications of their ISPH model to 2-D or 3-D.

On the other hand, the possibility to accurately impose non-homogeneous Neumann conditions is crucial in turbulence models where wall functions must be imposed on the turbulent quantities and on the velocity, which corresponds to non-homogeneous Neumann boundary conditions through the diffusion terms (see section 1.2.1 of Chapter 1). It is also crucial when modelling the temperature diffusion, in which case non-zero heat-fluxes may be imposed through the walls. Besides a temperature wall function may be used in turbulent mode (see section 3.3.3). All these improvements will be proposed in Chapter 3.

2.4.2.4 First-order consistent USAW operators

The modified differential operators presented above are subject to the same accuracy issues as the classical operators (see section 2.2.2). To solve this problem, it is possible to employ the same renormalisation technique as the one presented in section 2.2.2.3. Though, the renormalising matrix is changed due to the boundary terms in the differential operators [90]. Thus, we define a $d \times d$ matrix, $\boldsymbol{M}_a^\gamma$ for each particle, so as to have:

$$-\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b \boldsymbol{r}_{ab} \otimes (\boldsymbol{M}_a^\gamma \nabla w_{ab}) + \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}} \boldsymbol{r}_{as} \otimes (\boldsymbol{M}_a^\gamma \nabla \gamma_{as}) = \boldsymbol{I} \tag{2.155}$$

thus:

$$\left(-\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b \boldsymbol{r}_{ab} \otimes (\nabla w_{ab})^T + \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}} \boldsymbol{r}_{as} \otimes (\nabla \gamma_{as})^T\right) (\boldsymbol{M}_a^\gamma)^T = \boldsymbol{I} \tag{2.156}$$

or:

$$\boldsymbol{M}_a^\gamma = \left[\left(-\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b \boldsymbol{r}_{ab} \otimes \nabla w_{ab} + \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}} \boldsymbol{r}_{as} \otimes \nabla \gamma_{as}\right)^T\right]^{-1} \tag{2.157}$$

Finally, the first order consistent gradient of A is obtained through:

$$\boldsymbol{G}_a^{\gamma,-,1}\{A_b\} = -\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b A_{ab} \boldsymbol{M}_a^\gamma \nabla w_{ab} + \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}} A_{as} \boldsymbol{M}_a^\gamma \nabla \gamma_{as} \tag{2.158}$$

The same can be done concerning the divergence operator, which yields:

$$D_a^{\gamma,-,1}\{\boldsymbol{A}_b\} = -\frac{1}{\gamma_a \rho_a}\sum_{b\in\mathcal{P}} m_b \boldsymbol{A}_{ab} \cdot \boldsymbol{M}_a^\gamma \nabla w_{ab} + \frac{1}{\gamma_a \rho_a}\sum_{s\in\mathcal{S}} \rho_s \boldsymbol{A}_{as} \cdot \boldsymbol{M}_a^\gamma \nabla \gamma_{as} \tag{2.159}$$

One may compare the above two formulae with (2.49) and (2.50). It is possible to check that the renormalised operators are indeed first-order consistent. For example Figure 2.9 shows the values of the SPH divergence of the position at all the particles of a square tank discretised with 40×40 particles, with $D_a^{\gamma,-}$ and $D_a^{\gamma,-,1}$. Similar results are obtained regarding the gradient operator.

On the other hand, the Laplacian operator given by equation (2.151) is not 1st order consistent. This can prove problematic, in particular in the case of free-surface flows where the pressure has a linear component. The works by Schwaiger *et al.* [127] and by Fatehi *et al.* [32] that aimed at building a 2nd order consistent Laplacian operator was done in the framework of classical boundary conditions (see section 2.2.3.2). In order to use such formulations in the framework of the USAW boundary conditions, it is necessary to adapt them so as to take the boundary term into account. Such a work is still to be conducted and looks challenging.

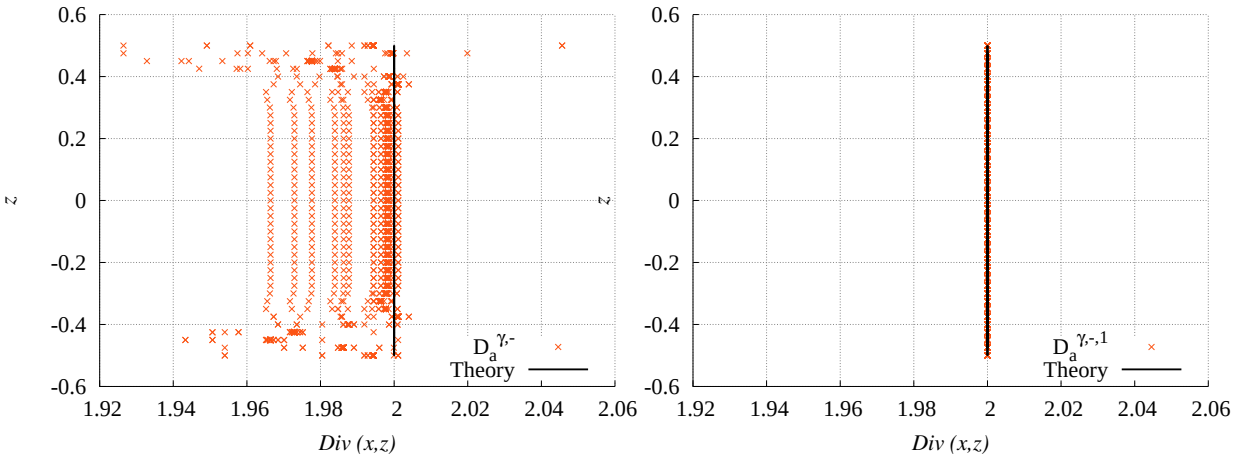


Figure 2.9: values of the SPH divergence of the position (x, z) at all the particles of a square tank discretised with 40×40 particles, with $D_a^{\gamma,-}$ and $D_a^{\gamma,-,1}$. (x, z). Comparison with the analytical value $Div(x, z) = 2$.

2.5 Imposition of free-surface boundaries in SPH

2.5.1 Pressure condition

In WCSPH, the density tends to zero when approaching the free-surface due to the absence of neighbours in the kernel support, which, due to the equation of state, makes the pressure tend to zero. Thus, the Dirichlet condition on p at the free-surface (see Table 1.2) is naturally fulfilled. This is not the case in ISPH models where it is thus necessary to impose the Dirichlet condition at the free-surface. Due to the kernel truncation close to the free-surface, and since the latter is not meshed so that the USAW technique does not restore the kernel normalisation condition, the only condition that can be imposed is a zero pressure at the free-surface. This thus excludes the idea of working on the dynamic pressure in an ISPH scheme for free-surface flows, where a non-zero Dirichlet should be imposed (at least with the present state-of-the-art). This also implies that it is necessary to detect the particles that belong to the free-surface in order to impose the Dirichlet condition. Lee *et al.* [72] proposed an algorithm based on the value of the position divergence computed through:

$$D_a^-\{\boldsymbol{r}_b\} = -\frac{1}{\rho_a}\sum_{b\in\mathcal{P}} m_b \boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla} w_{ab} \tag{2.160}$$

In dimension $d = 2$, a particle is considered as belonging to the free-surface when $D_a^-\{\boldsymbol{r}_b\} \leq 1.5$ and when $D_a^-\{\boldsymbol{r}_b\} \leq 2.4$ in dimension $d = 3$ [71, 72]. Other free-surface detection techniques exist, like the one proposed by Marrone *et al.* [86], but only the one presented above was tested in this work.

The imposition of a zero pressure at the free-surface in ISPH is done by manipulating the system entries. Let $\mathcal{E}$ be the set of free-surface particles and e a particle belonging to $\mathcal{E}$. Using the notations of section 2.3.2.3, the system to be solved for a set of n particles reads:

$$\begin{pmatrix} A_{aa} & \cdots & A_{ae} & \cdots & A_{an} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{ea} & \cdots & A_{ee} & \cdots & A_{en} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{na} & \cdots & A_{ne} & \cdots & A_{nn} \end{pmatrix} \begin{pmatrix} p_a \\ \vdots \\ p_e \\ \vdots \\ p_n \end{pmatrix} = \begin{pmatrix} B_a \\ \vdots \\ B_e \\ \vdots \\ B_n \end{pmatrix} \tag{2.161}$$

The component p_e of the unknown vector of pressures is known through the Dirichlet condition $p_e = 0$. Thus it is not necessary to do the product between the line of index e and the vector $\boldsymbol{P}$ and one can suppress the matrix line that corresponds to e, which yields a rectangular matrix. On the other hand, since the value p_e is known the product of the column of index e and $\boldsymbol{P}$ may be passed to the right-hand side of the equation, thus yielding a square matrix again. These manipulations are done with all the particles $e \in \mathcal{E}$. In the end the system to be solved does not involve the free-surface particles anymore and they appear in the right-hand side:

$$\begin{pmatrix} A_{aa} & \cdots & A_{an} \\ \vdots & \ddots & \vdots \\ A_{na} & \cdots & A_{nn} \end{pmatrix} \begin{pmatrix} p_a \\ \vdots \\ p_n \end{pmatrix} = \begin{pmatrix} B_a \\ \vdots \\ B_n \end{pmatrix} - \sum_{e \in \mathcal{E}} p_e \begin{pmatrix} A_{ae} \\ \vdots \\ A_{ne} \end{pmatrix} \tag{2.162}$$

Since a zero pressure is imposed at the free-surface the last term in this equation vanishes. Though, bear in mind that the interactions of the free-surface particles with the remaining particles is represented since they are involved in the pressure gradient computation.

2.5.2 Conditions on the velocity, the temperature, k and ϵ

According to Table 1.2, the free-surface boundary conditions imposed on $\boldsymbol{v}$ read:

$$\begin{cases} \left(\dfrac{\partial z}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{n} \right)_{\partial \Omega_\eta} = 0 \\ \boldsymbol{\tau} \cdot \boldsymbol{n}|_{\partial \Omega_\eta} = 0 \end{cases} \tag{2.163}$$

While the first condition is automatically verified in a Lagrangian framework, the second one (a homogeneous Neumann on $\mu \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{n}}$) is ensured by the absence of neighbours across the free-surface. Indeed, considering the framework of the USAW boundary conditions, at a solid wall imposing a homogeneous Neumann is done by cancelling the boundary term in the Laplacian operator. A similar process thus naturally happens close to a free-surface and the derivatives of the fields along the normal to the free-surface tend to zero when approaching it. Thus, nothing has to be done

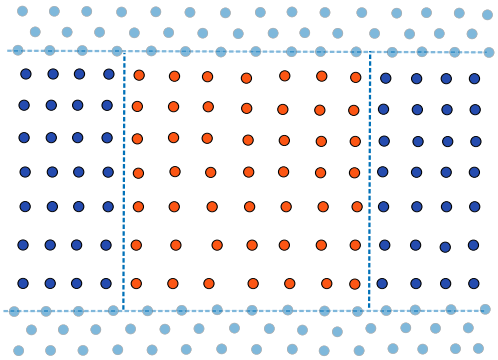


Figure 2.10: Sketch of the buffer layer method for the imposition of open boundaries, with the dummy particles technique for the walls representation. The buffer layers appear in dark blue.

regarding the imposition of a homogeneous Neumann at the free-surface. This also holds for the k, ϵ and T fields.

2.6 Imposition of open boundaries in SPH

The imposition of open boundaries in SPH has classically been done through the buffer layer method [51, 67, 143]. Layers of particles are placed beyond the boundary so as to fill the kernel supports in its vicinity. Usually four layers are used. The values of the fields in the buffer zone are imposed as the ones at the boundary. At an inflow boundary, a particle that enters the domain is changed into a free particle and its physical quantities are not imposed anymore. On the other hand, at an outflow boundary, a particle that leaves the domain is changed into a buffer particle and its physical quantities are prescribed. A sketch of this technique is provided in Figure 2.10 with the dummy particles technique for wall treatment. This way of handling ingoing and outgoing particles by sudden changes of their nature was shown to generate shocks. With WCSPH models, using Riemann solvers (based on [39]) partially solves this issue [82]. To our knowledge, the only ISPH model where inlet/outlet conditions were introduced is the one proposed by Hosseini *et al.* [51] based on a rotational projection scheme. They used the buffer layer technique, imposing a homogeneous Neumann condition on the pressure at the inlet and a Dirichlet condition at the outlet. This was done in an approximate way, by setting the pressure in the buffer zone through an SPH interpolation over the surrounding free particles.

However, the boundary layers technique is problematic for complex inlets where the flow may not be parallel to the boundary normal. In particular, it does not make it possible to generate waves at an inlet. Besides, Kassiotis *et al.* [59] encountered issues when trying to use it for coupling with a 1-D Finite Differences model. To remedy these problems, the unified semi-analytical technique for imposing wall boundary conditions was extended to the imposition of open boundaries by Kassio-

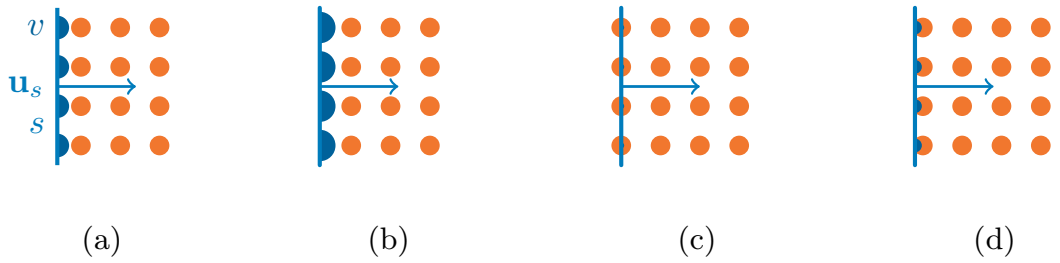


Figure 2.11: Sketch of the process of particles creation with vertices v and segments s at an inflow boundary [60]: a) the vertex masses grow due to the ingoing flux; b) their mass has reached the maximum threshold; c) new free particles are released and the vertex masses become negative; d) the vertex masses start growing again.

tis *et al.* [60]. This was done with a WCSPH model. The idea is to discretise the boundary through vertex particles and segments (see Figure 2.8), and then let the masses of the vertex particles belonging to open boundaries evolve over time as a function of the desired ingoing/outgoing mass flux through the open boundary segments. The vertex particles are then used to create/delete fluid particles, which is done by setting a minimum and a maximum value for their mass, proportional to the mass of the free particles. At an inlet, the mass flux is positive and the mass of a vertex increases until it reaches the higher threshold. Then, a fluid particle is created at that location while its mass is decreased by a reference mass. An illustration of this process is provided in Figure 2.11. At an outlet, the mass flux is negative and when a fluid particle crosses a segment it is deleted and its mass is distributed onto the vertices directly linked to the segment, a weight being associated to each of these vertices. Care was taken when defining the weights that the largest amount of mass is attributed to the closet vertex particle, which minimises mass displacement. This technique ensures that the total mass variation is a continuous function of time, so that the particle creation/destruction does not introduce any perturbation on the density or on the momentum. Besides, it keeps a correct particle distribution near the open boundaries. A more detailed description of this algorithm will be given in section 3.5.1 since it was used for the representation of open boundaries in this work and extended to ISPH.

In WCSPH, without additional treatment the particle creation/destruction induces variations of the density field with a constant velocity imposed at the inlet, which is incorrect. To avoid this, it is necessary to introduce additional terms in the SPH continuity equation (2.68). Their description is given in [34]. Note that in [60], the Riemann invariants of Euler equations were not used to prescribe the fields at the open boundaries, which was done afterwards by Ferrand *et al.* [34].

2.7 Turbulence modelling and treatment of the viscous term in SPH

2.7.1 Turbulence modelling: state-of-the-art in SPH

Two main issues have to be tackled regarding turbulence modelling with SPH. First, 3-D computations of turbulent flows through DNS requires massively parallel codes. The impressive development of GPU cards over the last few years and the adaptation of SPH codes to their architecture made massive parallelism financially accessible, thus solving this issue. A quasi DNS simulation on a minimal channel at $Re_* = 210$ was performed with WCSPH and the USAW boundary conditions in [91, 93] (though not in a GPU framework). The second issue concerns the imposition of wall functions in turbulence models (recall they are required for RANS[7] and LES models, see section 1.2), which is inaccurate when using classical SPH boundary conditions. As said earlier, some of the classical boundary models in SPH do not even allow for the imposition of complex Neumann boundary conditions, as it is required for $\boldsymbol{v}$ and ϵ (see section 2.4). In spite of this, RANS and LES turbulence models began to be developed in SPH with classical boundary conditions. In [148], a mixing length model, a $k - L_m$ model and a $k - \epsilon$ model were introduced in WCSPH with dummy particles. These models were applied to the simulation of plunging breaking solitary wave in [54]. Moreover, a $k - \epsilon$ model was introduced with the ghost particle technique in [23]. On the other hand, a 2-D LES model was introduced in WCSPH by Lo & Shao [79], which was also applied by Dalrymple & Rogers [20][8]. A 3-D LES model with dummy particles was also introduced in WCSPH and tested on the modelling of non-linear water waves [56].

Recently, the development of the USAW boundary conditions led to improvements of these turbulence models: in [35] a $k-\epsilon$ model in WCSPH was proposed with much improved results compared to [148]. On the other hand, a 3-D LES model based on the USAW boundary conditions was proposed in [92]. In this work a LES simulation of a 3-D turbulent channel flow was performed, but the results showed a clear deviation from the DNS results on that case. They identified inaccuracies in the prediction of pressure within the eddies, leading to a wrong isotropy redistribution through the pressure-strain correlations. Besides, they showed that a much finer discretisation than that they used was required to improve the results. This is problematic for industrial applications in terms of computational times.

Modelling turbulence through LES models is thus still problematic in SPH. On the other hand, RANS models in SPH now provide a quality of results quite close to mesh-based methods [35]. The most advanced RANS model that was adapted to SPH is the $k-\epsilon$ model. To our knowledge, RANS RSTM models (see section 1.2.1) were never introduced in an SPH framework. Thus, in this work a $k - \epsilon$ RANS turbulence closure was chosen, which presents limits (see section 1.2) but is widely used in the industry due to its simplicity and fairly good quality of results on most problems.

[7] Note that this holds for high-Reynolds and low-Reynolds models.
[8] Although the relevance of 2-D LES is quite doubtful since turbulence is a 3-D phenomenon.

2.7.2 SPH k and ϵ equations

With classical SPH boundary conditions, the space discretisation of the k and ϵ equations, (1.48) and (1.54) reads:

$$\begin{cases} \dfrac{dk_a}{dt} = \mathbb{P}_a - \epsilon_a + \dfrac{1}{\rho_a} L_a\{\mu_{k,b}, k_b\} \\ \dfrac{d\epsilon_a}{dt} = \dfrac{\epsilon_a}{k_a}\left(C_{\epsilon_1}\mathbb{P}_a - C_{\epsilon_2}\epsilon_a\right) + \dfrac{1}{\rho_a} L_a\{\mu_{\epsilon,b}, \epsilon_b\} \end{cases} \tag{2.164}$$

with the SPH Laplacian operator given by (2.58). The production term $\mathbb{P}_a$ may be computed through:

$$\mathbb{P}_a = \nu_{T,a} S_a^2 \tag{2.165}$$

with the scalar mean rate of strain defined as $S_a = \sqrt{2\boldsymbol{S}_a : \boldsymbol{S}_a}$ where $\boldsymbol{S}_a$ is computed through:

$$\boldsymbol{S}_a = \frac{1}{2}\left[\boldsymbol{G}_a^-\{\boldsymbol{u}_b\} + \boldsymbol{G}_a^-\{\boldsymbol{u}_b\}^T\right] \tag{2.166}$$

where the symmetric form of the SPH gradient operator $\boldsymbol{G}_a^-$ (2.144) may be used for the rate of strain computation for more accuracy.

2.7.3 Treatment of the viscous term

The Morris Laplacian operator (2.58) was recommended in [10] to compute the viscous term, which comes to writing:

$$\begin{aligned} \boldsymbol{\nabla}\cdot\boldsymbol{\tau}_a &= \boldsymbol{\nabla}\cdot\left[\mu_a\boldsymbol{\nabla}\boldsymbol{u}_a + \mu_a(\boldsymbol{\nabla}\boldsymbol{u}_a)^T\right] + \boldsymbol{\nabla}(\lambda_a\nabla\cdot\boldsymbol{u}_a) \\ &\approx \boldsymbol{\nabla}\cdot(\mu_a\boldsymbol{\nabla}\boldsymbol{u}_a) \\ &\approx \boldsymbol{L}_a(\{\mu_b\},\{\boldsymbol{u}_b\}) = 2\sum_{b\in\mathcal{P}} V_b\bar{\mu}_{ab}\frac{\boldsymbol{u}_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab} \end{aligned} \tag{2.167}$$

The second line was obtained by suppressing the term $\boldsymbol{\nabla}(\lambda_a\nabla\cdot\boldsymbol{u}_a)$, which is actually equal to zero for an incompressible flow and usually neglected in WCSPH. Besides, the term $(\mu_a\boldsymbol{\nabla}\boldsymbol{u}_a)^T$ was also suppressed, which is correct as long as the viscosity is constant. Though, the viscosity varies in case a RANS or LES turbulence closure is used (since an eddy viscosity is then added, see section 1.2.1). It also varies in multiphase flows or in case it is considered a function of the temperature. In these cases the transpose velocity gradient term $(\mu_a\boldsymbol{\nabla}\boldsymbol{u}_a)^T$ should be represented. However, that was not the case in the $k-\epsilon$ model proposed by Ferrand *et al.* [35] and in the LES model proposed by Mayrhofer [91]. A formulation of the viscous term that includes the transpose velocity gradient was proposed in [146] but it depends on the problem dimension and requires testing in 3-D.

In equation (2.167), the same notations as in section 1.2.1 were used: the viscous term involves an Eulerian velocity $\boldsymbol{u}$, which is non-zero at the walls in the RANS or LES formalism in order to better represent near-wall effects (see section 1.2.1). This technique was used in the $k-\epsilon$ model proposed

by Ferrand *et al.* [35] and in the LES model proposed by Mayrhofer [91].

Note that the viscous term can be treated implicitly or explicitly, although few works in SPH treat it implicitly (which requires a matrix inversion, see *e.g.* [141]).

2.8 Buoyancy modelling in SPH

To our knowledge, only two authors contributed to buoyancy modelling with SPH so far: Szewc *et al.* [134] and Ghasemi *et al.* [37]. Szewc *et al.* proposed two buoyancy models in a WCSPH framework, one based on the Boussinesq approximation (see section 1.3) and one with a variable density field. The boundary conditions were represented through mirror particles. At isothermal walls, the wall temperature was imposed through the prescription of the ghost particles' temperature according to $T_{a'} = 2T_w - T_a$. where a is the free particle, a' its mirror particle and T_w is the imposed temperature at the wall. This allowed them to prescribe the Dirichlet boundary condition, though not exactly. At adiabatic walls they prescribed $T_{a'} = T_a$, a classical way to impose a homogeneous Neumann. The results they obtained with the Boussinesq approximation and with the variable density model were very close to each other and well validated against mesh-based methods. They presented 2-D quantitative validation on the Rayleigh-Taylor instability and on a horizontally differentially heated square cavity.

On the other hand, Ghasemi *et al.* [37] proposed a buoyancy model in an ISPH framework, based on the Boussinesq approximation. They modelled the boundaries through ghost particles, imposing a homogeneous Neumann boundary condition on the pressure and either a Dirichlet or a homogeneous Neumann condition on the temperature. They presented qualitative 2-D validation on two lock-exchange cases.

In the above two works, no case presenting a non-zero imposed heat flux through a wall was presented. They are based on classical SPH boundary conditions, which makes the wall boundary conditions inaccurate. Besides, these works do not include any turbulence model.

2.9 Reduction of the computational times through parallel programming

The high number of neighbours for each particle (around 30 in 2-D, 250 in 3-D) makes the SPH method computationally expensive (much more than mesh-based methods). The number of particles required in 3-D simulations is usually too large to be handled by a single processor. This is a serious obstacle to the extension of the method to an industrial scale, but also to its development since even relatively small validation cases in 3-D may take days or even weeks on sequential SPH codes. Massive parallelism is thus a key-issue in SPH and closely linked to the growth of the method. As

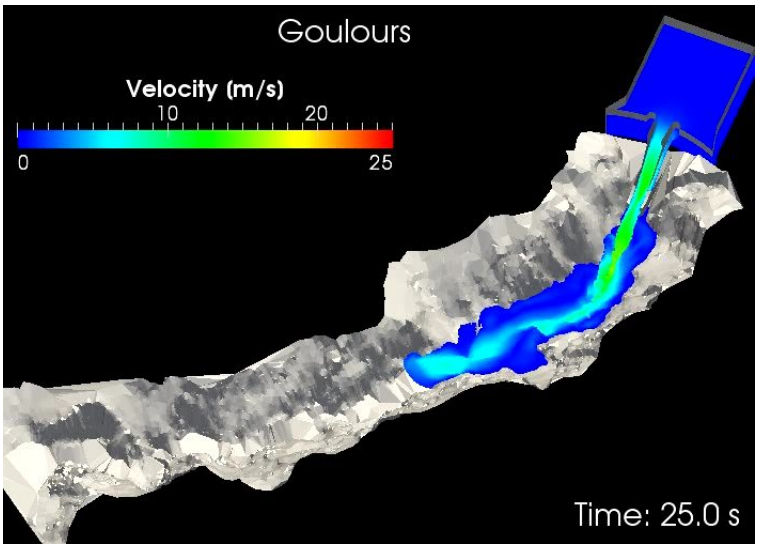


Figure 2.12: 3-D modelling of the Goulours dam spillway (EDF, south of France) with the SPARTACUS-3D code. About 1.1 million particles and 7.7×10^5 boundary elements were used [94].

in mesh-based methods, SPH was first made parallel on Central Processing Units (CPU) clusters. Such SPH codes began to be developed and tested in the 2000's. In astrophysics, the code GADGET, later improved into GADGET-2, was developed [133]. In hydrodynamics, Marongiu [83], then Issa *et al.* [54] and Moulinec *et al.* [53, 105, 106] developed a parallel SPH code called SPARTACUS-3D and presented applications to various cases like a flow across periodic hills. Later on, that code was run on an IBM BlueGene/L for the modelling of the water collapse in waterworks in 3-D [73, 74]. In particular, a simulation of the Goulours dam spillway was performed. Figure 2.12 is an illustration of this case, where the USAW boundary conditions were used. Other parallel SPH codes running on CPUs were developed around the world, like SPH-flow [88], the open-source code SPHysics [122] or a code developed in Trento [36]. Though, BlueGene-type machines are extremely expensive and very few research entities have access to such super-computers (access to large enough CPU clusters is also difficult). Until 2008, the extension of the SPH method to an industrial scale was thus still hindered by the difficulty to realise simulations involving high numbers of particles.

A technological breakthrough was achieved in the 2010's with the very fast development of SPH codes running on Graphical Processing Units (GPUs, also called graphic cards herein). The latter were at first dedicated to video game computing and quickly evolved into powerful parallel computing devices. Nowadays, GPU cards are also aimed at scientific computations, not only video games. Their computational power is such that they are able to match huge CPU clusters in terms of performance, turning a laptop into a very powerful computer. On the other hand, compared to CPU clusters, GPU cards are extremely cheap. As pointed out by Hérault *et al.* [45], the use of GPUs instead of CPU clusters for SPH is well suited since GPUs perform best on computationally intensive problems (instead of data intensive). In SPH, the operations-to-data ratio is quite high in comparison to mesh-based methods, which makes GPU computing more advantageous with SPH

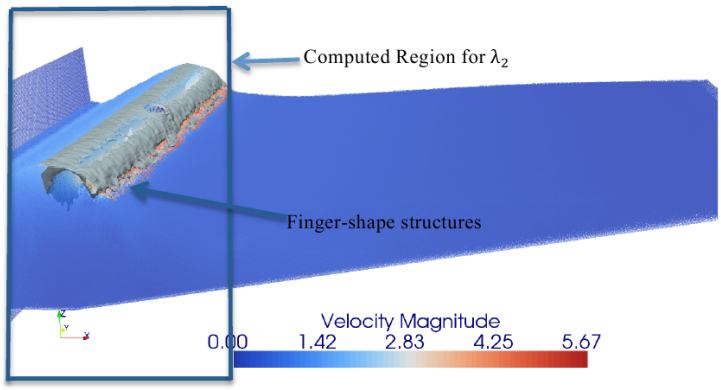


Figure 2.13: 3-D modelling of a breaking wave with the GPUSPH code. Representation of the free-surface shape and of the three-dimensional large-scale coherent structures under a plunging wave. About 4.5 million particles were used [31].

than with the latter. The first adaptation of SPH to GPU (Amada *et al.* in 2003 [4]) used it only for the computation of the forces, leaving the neighbours search computation to the CPU. In 2005, Kolb & Cuntz [62] used the GPU to compute the whole SPH algorithm, followed by Harada *et al.* in 2007 [42]. In these works, the OpenGL language (dedicated to graphics rendering) was used to program on the GPU, which was the only possibility at that time, but required a quite deep knowledge of computer graphics. To use OpenGL for scientific computing, conversions between mathematical operations and equivalent graphical rendering operations must be done. In 2007, the GPU provider Nvidia® introduced a new programming language: CUDA (standing for Compute Unified Device Architecture), solely dedicated to GPU programming, either for computer graphics or scientific calculations. The CUDA language is based on C++ with extensions to handle the specific needs of the GPU and its interfacing with the CPU host. From 2008, Hérault *et al.* developed an SPH code called GPUSPH, based on the CUDA language and running on one Nvidia graphic card [45, 47]. This was closely followed by the development of the DualSPHysics code (Crespo *et al.* in 2011 [18]). DualSPHysics is now able to run on multiple CPUs or on multiple GPUs [27], while GPUSPH can be run on multiple GPUs [124]. Figure 2.13 shows an example of simulation performed with the GPUSPH code while Figure 2.14 shows two examples of simulations performed with the DualSPHysics code.

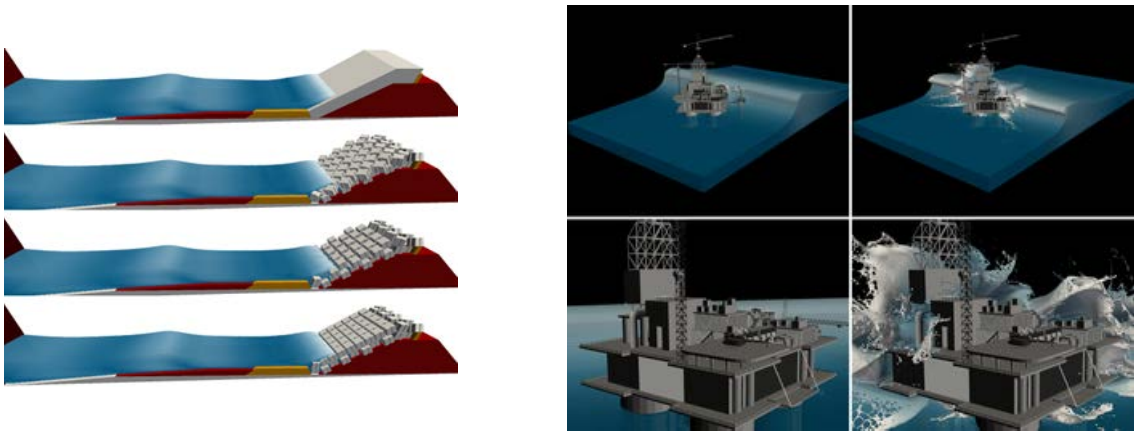


Figure 2.14: Examples of 3-D cases simulated with the Dual SPHysics code: modelling of a wave impacting a breakwater [3] (left) and an oil rig [26] (right). The latter simulation involved about one billion particles.

Chapter 3

A new incompressible SPH model

Dans ce Chapitre, le modèle développé dans ce travail est présenté, qui s'applique à la résolution du système d'équations auquel le Chapitre1 a abouti avec les conditions aux limites envisagées. Cette résolution repose sur la méthode numérique SPH décrite au Chapitre 2. L'objectif est de produire un modèle capable de prédire les champs de pression avec précision, qui puisse modéliser la turbulence et les effets de scalaires actifs. La possibilité d'imposer des conditions de frontière ouvertes ainsi que le parallélisme massif du code sont apparus comme primordiaux pour l'application à des écoulements industriels. En vue de cela, un modèle SPH incompressible a été développé, fondé sur la technique des conditions aux limites semi-analytiques pour la représentation des frontières du domaine de calcul. Un modèle de turbulence $k - \epsilon$ y a été incorporé, ainsi qu'un modèle pour les scalaires actifs reposant sur l'approximation de Boussinesq. On propose une technique pour la représentation des frontières ouvertes et le modèle est implémenté dans un code massivement parallèle fonctionnant sur carte graphique. Dans ce Chapitre, la discrétisation temporelle des équations est d'abord décrite, suivie de leur discrétisation spatiale. Ensuite, le traitement des conditions aux limites en paroi, en surface libre et aux frontières ouvertes est détaillé. La résolution de l'équation de Poisson sur la pression fait l'objet de la section suivante, et l'on voit comment les conditions aux limites sont prises en compte dans le système linéaire. Enfin, une section est consacrée au calcul des intégrales de bord apparaissant dans la technique des conditions aux limites semi-analytiques, avant de finir avec une courte section donnant quelques informations sur le code massivement parallèle.

3.1 Preliminary considerations

Several key-issues were identified regarding the SPH method, which still hinder its application to industrial cases. Here, we address several of them, namely:

- the lack of accuracy in the pressure prediction;
- the treatment of the wall and inflow/outflow boundary conditions;
- the modelling of turbulence;
- the modelling of buoyancy and its interactions with turbulence.

The proposed solutions are the following:

- using an incompressible SPH model;
- adapting it to the unified semi-analytical wall boundary conditions;
- developing inflow/outflow boundary conditions in this framework;
- improving the existing SPH $k-\epsilon$ model for turbulence and include it in the new ISPH model;
- including a buoyancy model in this framework, taking the interactions with turbulence into account.

Besides, the application on industrial cases requires the implementation of the developments in a massively parallel code. Here the GPU framework was chosen.

In section 3.2, the space-time discretisation of the equations (1.92) is described. We include laminar and turbulent (Reynolds-averaged) flows in the same framework, our purpose being to unify all wall boundary treatment from [35], including the Poisson equation, the heat equation and the $k-\epsilon$ model. In sections 3.3, 3.4 and 3.5 the imposition of boundary conditions at the solid walls, the free-surface and the inlet/outlet boundaries are described. In section 3.6 the resolution of the pressure Poisson equation is dealt with. We will also explain how the method proposed by Bonet and Feldman for the analytical computation of the wall renormalisation factor can be applied to our description of the solid boundaries in 2-D, in order to reduce computational time. The technique for the computation of the wall renormalisation factor in 3-D is also explained. Finally, section 3.9 quickly describes the main features of the parallel GPU code.

3.2 Space-time discretisation of the governing equations

Recall that the system of equations to be solved is the one given by (1.92), with the set of boundary conditions given in Table 1.2. In section (3.2.1) the time-discretisation of the equations is described,

followed by the space discretisation in section (3.2.2).

3.2.1 Time discretisation

In this section, we drop the particle subscripts, since we focus on time discretisation. In case the $k-\epsilon$ model is used, k and ϵ are calculated at the beginning of each time-step in order to compute the eddy viscosity. This is done through a semi-implicit time-scheme, which is the same as in [35] except for the buoyancy term $\mathbb{G}$:

$$\begin{cases} \dfrac{k^{n+1}-k^n}{\delta t} = \mathbb{P}^n + \mathbb{G}^n - \epsilon^n \dfrac{k^{n+1}}{k^n} + \dfrac{1}{\rho}\nabla\cdot(\mu_k^n \boldsymbol{\nabla} k^n) \\ \dfrac{\epsilon^{n+1}-\epsilon^n}{\delta t} = \dfrac{\epsilon^n}{k^n}\left(C_{\epsilon_1}\mathbb{P}^n + C_{\epsilon_3}\mathbb{G}^n - C^n_{\epsilon_2,Y}\epsilon^{n+1}\right) + \dfrac{1}{\rho}\nabla\cdot(\mu_\epsilon^n \boldsymbol{\nabla}\epsilon^n) \end{cases} \tag{3.1}$$

The definitions of all the variables are given in equation (1.93) and the $k-\epsilon$ model constants are given in Table 1.1. The negative dissipation terms are treated implicitly in order to avoid negative values of k and ϵ.

The production term $\mathbb{P}^n$ involves the velocity field at time n:

$$\mathbb{P}^n = \min\left(\sqrt{C_\mu}k^n S^n, \nu_T^n {S^n}^2\right) \tag{3.2}$$

The buoyancy production/destruction term $\mathbb{G}^n$ involves the temperature fields at time n and is made semi-implicit in case it is negative. This avoids negative values of k and ϵ:

$$\mathbb{G}^n = \begin{cases} \dfrac{\beta C_\mu}{Pr_T}\dfrac{k^n k^{n+1}}{\epsilon^n}\boldsymbol{\nabla} T^n\cdot\boldsymbol{g} & \text{if } \boldsymbol{\nabla} T^n\cdot\boldsymbol{g}\le 0 \\ \dfrac{\beta C_\mu}{Pr_T}\dfrac{(k^n)^2}{\epsilon^n}\boldsymbol{\nabla} T^n\cdot\boldsymbol{g} & \text{otherwise} \end{cases} \tag{3.3}$$

The Yap correction is applied:

$$C^n_{\epsilon_2,Y} = C_{\epsilon_2} - \max\left[0, 0.83\left(\frac{L_t^n}{L}-1\right)\left(\frac{L_t^n}{L}\right)^2\right] \tag{3.4}$$

with $L_t^n \doteqdot \dfrac{{k^n}^{3/2}}{\epsilon^n}$ and L the characteristic length of the flow.

Note that the velocity is usually initialised at zero and the initial values of k and ϵ, denoted by k_0 and ϵ_0, are usually set according to:

$$\begin{cases} k_0 = (0.002U)^2 \\ \epsilon_0 = 0.16\dfrac{\sqrt{k_0^3}}{L_m} \end{cases} \tag{3.5}$$

with U the characteristic velocity of the flow and a mixing length $L_m = \max(2\delta r, 10^{-5}m)$ [1]. The eddy viscosity is then given by:

$$\nu_T^{n+1} = C_\mu \frac{k^{n+1^2}}{\epsilon^{n+1}} \tag{3.6}$$

After the computation of ν_T^{n+1}, the time discretisation of (1.92) follows a Chorin predictor-corrector scheme with a pressure Poisson equation (see section 1.1.2). The time discretisation of the momentum equation with the incompressibility condition thus reads:

$$\begin{cases} \dfrac{\tilde{\boldsymbol{v}}^{n+1/2} - \boldsymbol{v}^n}{\delta t} = \dfrac{1}{\rho}\boldsymbol{\nabla}(\mu_E^{n+1}\boldsymbol{\nabla}\boldsymbol{u}^n) - [\beta(T^n - T_0) - 1]\,\boldsymbol{g} \\ \nabla^2\tilde{p}^{n+1} = \dfrac{\rho}{\delta t}\nabla\cdot\tilde{\boldsymbol{v}}^{n+1/2} \\ \dfrac{\boldsymbol{v}^{n+1/2} - \tilde{\boldsymbol{v}}^{n+1/2}}{\delta t} = -\dfrac{1}{\rho}\boldsymbol{\nabla}\tilde{p}^{n+1} \end{cases} \tag{3.7}$$

with $\tilde{p}^{n+1} = p^{n+1} + \frac{2}{3}\rho k^{n+1}$. Recall that the viscous force is based on an Eulerian velocity field $\boldsymbol{u}$ rather than the Lagrangian velocity $\boldsymbol{v}$, as explained in the section 1.2.1 of Chapter 1. Note that in case of confined flows, solving the system on the dynamic pressure proved to yield more accurate results than solving on the total pressure as was shown in system (3.7). Thus, in cases of confined flows $\tilde{p}$ is replaced by $p^* = \tilde{p} + \rho g z$ in the second and third lines of (3.7) and the term $[\beta(T - T_0) - 1]\,\boldsymbol{g}$ is replaced by $\beta(T - T_0)\boldsymbol{g}$ in the first line of (3.7). It was not possible to do so for free-surface flows because this leads to the imposition of a non-zero Dirichlet condition on the pressure, that is not correctly imposed with our treatment of the free-surface (see sections 2.5.1 and 3.4).

In the third line of (3.7), $\boldsymbol{v}^{n+1/2}$ is a velocity field that serves to move the particles on a first half time-step. Indeed, the position update is done through a second order time-scheme so that it is split into two parts:

$$\begin{cases} \dfrac{\boldsymbol{r}^{n+1/2} - \boldsymbol{r}^n}{\delta t/2} = \boldsymbol{v}^{n+1/2} \\ \dfrac{\boldsymbol{r}^{n+1} - \boldsymbol{r}^*}{\delta t/2} = \boldsymbol{v}^{n+1} \end{cases} \tag{3.8}$$

In between, a stabilising procedure consisting in a particle shift is applied, which is why a modified position $\boldsymbol{r}^*$ appears in the second line of (3.8) instead of $\boldsymbol{r}^{n+1/2}$. On the other hand, the heat equation is solved at the position $\boldsymbol{r}^{n+1/2}$:

$$\frac{T^{n+1} - T^n}{\delta t} = \nabla\cdot(K_E^{n+1}\boldsymbol{\nabla}T^n) \tag{3.9}$$

The modified position $\boldsymbol{r}^*$ is computed through:

$$\boldsymbol{r}^* = \boldsymbol{r}^{n+1/2} + \delta\boldsymbol{r} \tag{3.10}$$

[1] Note though that in some cases, like the turbulent plane Poiseuille flow, the initial values of k, ϵ and $\boldsymbol{v}$ are chosen differently so as to be closer to the expected steady-state solution (see Chapter 4).

with:

$$\delta \boldsymbol{r} = -C_{shift} h^2 \boldsymbol{\nabla} C \tag{3.11}$$

in the same way as Lind *et al.* in their ISPH model [76] (see section 2.3.2.1). C is a particle concentration, h is the smoothing length and C_{shift} is a coefficient set to 0.7 in this work for the Wendland kernel (based on numerical experiments). The convection term corresponding to the particles displacement of $\delta \boldsymbol{r}$ must be added to the other fields[2]. For example, on the velocity and the temperature, this reads:

$$\begin{cases} \boldsymbol{v}^{n+1} = \boldsymbol{v}^{n+1/2} + \boldsymbol{\nabla} \boldsymbol{v}^{n+1/2} \cdot \delta \boldsymbol{r} \\ T^{n+1} \leftarrow T^{n+1} + \boldsymbol{\nabla} T^{n+1} \cdot \delta \boldsymbol{r} \end{cases} \tag{3.12}$$

This correction is also applied to k and ϵ. To summarise, the position, temperature, k and ϵ updates are performed through:

$$\begin{cases} \dfrac{\boldsymbol{r}^{n+1/2} - \boldsymbol{r}^n}{\delta t/2} = \boldsymbol{v}^{n+1/2} \\ \dfrac{T^{n+1} - T^n}{\delta t} = \nabla \cdot (K_E^{n+1} \boldsymbol{\nabla} T^n) \\ \boldsymbol{r}^* = \boldsymbol{r}^{n+1/2} + \delta \boldsymbol{r} \ \text{ with } \ \delta \boldsymbol{r} = -C_{shift} h^2 \boldsymbol{\nabla} C \\ (T,k,\epsilon)^{n+1} \leftarrow (T,k,\epsilon)^{n+1} + \boldsymbol{\nabla} (T,k,\epsilon)^{n+1} \cdot \delta \boldsymbol{r} \\ \boldsymbol{v}^{n+1} = \boldsymbol{v}^{n+1/2} + \boldsymbol{\nabla} \boldsymbol{v}^{n+1/2} \cdot \delta \boldsymbol{r} \\ \dfrac{\boldsymbol{r}^{n+1} - \boldsymbol{r}^*}{\delta t/2} = \boldsymbol{v}^{n+1} \end{cases} \tag{3.13}$$

Our time-marching scheme is thus made of (3.1), (3.7) and (3.13). A possibility, instead of applying this particle shift (which does not respect the incompressibility constraint) would be to correct the particles positions by imposing the SPH interpolation of the density to be constant, as in [52] (see section 2.3.2.1). Though, this was not tested in this work and would require some care regarding the imposition of the boundary conditions at each sub-step.

3.2.2 Space discretisation

3.2.2.1 k and ϵ equations

A space discretisation of the k and ϵ equations (3.1) in the framework of the USAW boundary conditions was proposed by Ferrand *et al.* [35] (see section 2.7). Here the same approach is followed,

[2] Although it is not necessary for the pressure because the latter is not involved in any step between the particles displacement and the pressure Poisson equation, and then p^n only serves as an initial bet.

with the additional buoyancy term. Thus, (3.1) is space-discretised by:

$$\begin{cases} \dfrac{k_a^{n+1}-k_a^n}{\delta t} = \mathbb{P}_a^n + \mathbb{G}_a^n - \epsilon_a^n \dfrac{k_a^{n+1}}{k_a^n} + \dfrac{1}{\rho} L_a^{\gamma}\{\mu_{k,b}^n, k_b^n\} \\ \dfrac{\epsilon_a^{n+1}-\epsilon_a^n}{\delta t} = \dfrac{\epsilon_a^n}{k_a^n}\left(C_{\epsilon_1}\mathbb{P}_a^n + C_{\epsilon_3}\mathbb{G}_a^n - C_{\epsilon_2,Y,a}^n \epsilon_a^{n+1}\right) + \dfrac{1}{\rho} L_a^{\gamma}\{\mu_{\epsilon,b}^n, \epsilon_b^n\} \end{cases} \tag{3.14}$$

Since the density is constant in the model the particle subscript was dropped for ρ. The Ferrand Laplacian (2.153) is used for the computation of the diffusion terms on k and ϵ, which reads (dropping the time superscripts for the sake of simplicity):

$$\begin{aligned} L_a^{\gamma}\{\mu_{k,b}, k_b\} &= \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b\left(\mu_{k,a}+\mu_{k,b}\right)\frac{k_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab} - \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}\left(\mu_{k,a}\boldsymbol{\nabla}k_a + \mu_{k,s}\boldsymbol{\nabla}k_s\right)\cdot\boldsymbol{\nabla}\gamma_{as} \\ L_a^{\gamma}\{\mu_{\epsilon,a}, \epsilon_b\} &= \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b\left(\mu_{\epsilon,a}+\mu_{\epsilon,b}\right)\frac{\epsilon_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab} - \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}\left(\mu_{\epsilon,a}\boldsymbol{\nabla}\epsilon_a + \mu_{\epsilon,s}\boldsymbol{\nabla}\epsilon_s\right)\cdot\boldsymbol{\nabla}\gamma_{as} \end{aligned} \tag{3.15}$$

The terms $(\mu_{k,a}\boldsymbol{\nabla}k_a + \mu_{k,s}\boldsymbol{\nabla}k_s)$ and $(\mu_{\epsilon,a}\boldsymbol{\nabla}\epsilon_a + \mu_{\epsilon,s}\boldsymbol{\nabla}\epsilon_s)$ in the summations involving the segments will be used to impose Neumann boundary conditions on k and ϵ. This will be described in the sections 3.3, 3.4 and 3.5. Note that these boundary terms will then be denoted by $L_a^{bound}\{\mu_{k,b}, k_b\}$ and $L_a^{bound}\{\mu_{\epsilon,a}, \epsilon_b\}$[3], respectively.

The production term $\mathbb{P}_a^n$ is computed through:

$$\mathbb{P}_a^n = \min\left(\sqrt{C_\mu}k_a^n S_a^n, \nu_{T,a}^n {S_a^n}^2\right) \tag{3.16}$$

in agreement with (3.2), with the scalar mean rate of strain defined as $S_a^n = \sqrt{2\boldsymbol{S}_a^n : \boldsymbol{S}_a^n}$ where $\boldsymbol{S}_a^n$ is computed through:

$$\boldsymbol{S}_a^n = \frac{1}{2}\left[\boldsymbol{G}_a^{\gamma,-}\{\boldsymbol{u}_b^n\} + \boldsymbol{G}_a^{\gamma,-}\{\boldsymbol{u}_b^n\}^T\right] \tag{3.17}$$

the symmetric form of the SPH gradient operator $\boldsymbol{G}_a^{\gamma,-}$ (2.144) is used since it provides better accuracy than the antisymmetric one.

The buoyancy production/destruction term $\mathbb{G}_a^n$ is computed through:

$$\mathbb{G}_a^n = \begin{cases} \dfrac{\beta C_\mu}{Pr_{T,a}}\dfrac{k_a^n k_a^{n+1}}{\epsilon_a^n}\boldsymbol{G}_a^{\gamma,-}\{T_b^n\}\cdot\boldsymbol{g} & \text{if } \boldsymbol{G}_a^{\gamma,-}\{T_b^n\}\cdot\boldsymbol{g} \le 0 \\ \dfrac{\beta C_\mu}{Pr_{T,a}}\dfrac{(k_a^n)^2}{\epsilon_a^n}\boldsymbol{G}_a^{\gamma,-}\{T_b^n\}\cdot\boldsymbol{g} & \text{otherwise} \end{cases} \tag{3.18}$$

in agreement with (3.3). Once again, the symmetric form of the SPH gradient operator $\boldsymbol{G}_a^{\gamma,-}$ (2.144) is used for the sake of accuracy.

[3] the definition of L_a^{bound} is that of equation (2.154).

3.2.2.2 Momentum equation

The space discretisation of the momentum equation with the incompressibility constraint (3.7) is done through:

$$\begin{cases} \dfrac{\tilde{\boldsymbol{v}}_a^{n+1/2} - \boldsymbol{v}_a^n}{\delta t} = \dfrac{1}{\rho}\boldsymbol{L}_a^{\gamma}\{\mu_{E,b}^{n+1}, \boldsymbol{u}_b^n\} - [\beta(T_a^n - T_0) - 1]\,\boldsymbol{g} \\ L_a^{\gamma}\{\tilde{p}_b^{n+1}\} = \dfrac{\rho}{\delta t} D_a^{\gamma,-}\{\tilde{\boldsymbol{v}}_b^{n+1/2}\} \\ \dfrac{\boldsymbol{v}_a^{n+1/2} - \tilde{\boldsymbol{v}}_a^{n+1/2}}{\delta t} = -\dfrac{1}{\rho}\boldsymbol{G}_a^{\gamma,+}\{\tilde{p}_b^{n+1}\} \end{cases} \tag{3.19}$$

with the predicted velocity divergence computed through:

$$D_a^{\gamma,-}\{\tilde{\boldsymbol{v}}_b\} = -\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b \boldsymbol{v}_{ab}\cdot\boldsymbol{\nabla} w_{ab} + \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}} \boldsymbol{v}_{as}\cdot\boldsymbol{\nabla}\gamma_{as} \tag{3.20}$$

and the pressure gradient through:

$$\boldsymbol{G}_a^{\gamma,+}\{\tilde{p}_b\} = \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b\,(p_a + p_b)\,\boldsymbol{\nabla} w_{ab} - \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}} (p_a + p_s)\,\boldsymbol{\nabla}\gamma_{as} \tag{3.21}$$

In comparison with (2.145) and (2.146), the fluid density has been removed from the sums since it is now a constant. As discussed in section 1.1.2.1, it is important to employ skew-adjoint divergence and gradient operators in the projection method. This is why the $D_a^{\gamma,-}$ operator is used for the velocity divergence computation while the $\boldsymbol{G}_a^{\gamma,+}$ operator is used for the pressure gradient computation. However, these two operators are not exactly skew-adjoint since the integration of their boundary terms does not yield the right flux, so that the projection method is not exact. This issue seems avoidable through the construction of SPH divergence and gradient operators with the required skew-adjointness property, like the ones proposed by Kulasegaram *et al.* (2.136), although this was not tested in this work.

The Laplacian operator used for the viscous term discretisation is the Ferrand Laplacian applied to a vector (2.151), which reads:

$$\boldsymbol{L}_a^{\gamma}\{\mu_{E,b}, \boldsymbol{u}_b\} = \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b(\mu_{E,a} + \mu_{E,b})\frac{\boldsymbol{u}_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla} w_{ab} - \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}(\mu_{E,a}\boldsymbol{\nabla}\boldsymbol{u}_a + \mu_{E,s}\boldsymbol{\nabla}\boldsymbol{u}_s)\cdot\boldsymbol{\nabla}\gamma_{as} \tag{3.22}$$

On the other hand, the Laplacian operator used in the pressure Poisson equation is the Ferrand Laplacian (2.153), *i.e.* the Morris Laplacian adapted to the USAW boundary conditions. The pressure Laplacian then reads:

$$(\nabla^2\tilde{p})_a \approx L_a^{\gamma}\{\tilde{p}_b\} = \frac{2}{\gamma_a}\sum_{b\in\mathcal{P}} V_b\frac{\tilde{p}_{ab}}{r_{ab}^2}\mathbf{r}_{ab}\cdot\boldsymbol{\nabla} w_{ab} - \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}(\boldsymbol{\nabla}\tilde{p}_a + \boldsymbol{\nabla}\tilde{p}_s)\cdot\boldsymbol{\nabla}\gamma_{as} \tag{3.23}$$

One reason why this Laplacian operator is used, and not a compatible operator as (2.52), is the high computational cost of the latter, which involves a two-fold summation over the neighbours. This issue could be partially overcome by computing the Laplacian in two steps: first compute an SPH gradient of the pressure and store it, then compute its SPH divergence. Though, the use of a compatible Laplacian operator in the projection method was shown to lead to checker-board effects due to the collocation of the pressure and velocity computations, as already said in section 2.3.2.1. This is a second reason why the SPH projection method proposed in this work is not exact and until now no solution was proposed to this problem, even in mesh-based methods (to our knowledge).

The pressure Laplacian (3.23) is only zero-order consistent, which is an issue with regards to free-surface flows where the pressure has a linear component. Since the construction of a second-order consistent Laplacian operator is complex in the framework of the USAW boundary conditions, as said in section 2.4.2.4, it was chosen to keep a zero-order consistent Laplacian operator but to apply a hydrostatic correction in order to obtain better accuracy on free-surface flows. The idea is to solve the pressure Poisson equation on the dynamic pressure, which is an equivalent problem in a continuous framework. Thus, the pressure Poisson equation reads:

$$\frac{2}{\gamma_a}\sum_{b\in\mathcal{P}} V_b \frac{\tilde{p}_{ab}+\rho g z_{ab}}{r_{ab}^2}\mathbf{r}_{ab}\cdot\boldsymbol{\nabla} w_{ab} - \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}(\boldsymbol{\nabla}\tilde{p}_a + \boldsymbol{\nabla}\tilde{p}_s - 2\rho\boldsymbol{g})\cdot\boldsymbol{\nabla}\gamma_{as} = \frac{\rho}{\delta t} D_a^{\gamma,-}\{\tilde{\boldsymbol{v}}_b^{n+1/2}\} \tag{3.24}$$

This equation is a linear system with unknowns $\{\tilde{p}_b^{n+1}\}$. It is solved through an iterative linear solver, as explained in the section 3.6.

3.2.2.3 Particles displacement and temperature update

A Ferrand Laplacian is used for the temperature diffusion (second line of system (3.27)), which yields:

$$L_a^{\gamma}\{K_{E,b},T_b\} = \frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b\left(K_{E,a}+K_{E,b}\right)\frac{T_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla} w_{ab} - \frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}\left(K_{E,a}\boldsymbol{\nabla}T_a + K_{E,s}\boldsymbol{\nabla}T_s\right)\cdot\boldsymbol{\nabla}\gamma_{as} \tag{3.25}$$

On the other hand, the position shift $\delta\boldsymbol{r}_a$ (used in the fourth line of system (3.27)) is computed through the simplest USAW-SPH gradient (2.140):

$$\begin{aligned}\delta\boldsymbol{r}_a = -C_{shift}h^2(\boldsymbol{\nabla}C)_a &\approx -C_{\text{shift}}h^2\boldsymbol{G}_a^{\gamma}\{1\}\\ &= -C_{\text{shift}}h^2\left(\frac{1}{\gamma_a}\sum_{b\in\mathcal{P}} V_b\boldsymbol{\nabla}w_{ab} - \frac{1}{\gamma_a}\sum_{s\in S}\boldsymbol{\nabla}\gamma_{as}\right)\end{aligned} \tag{3.26}$$

In this expression, the boundary term running over the segments s prevents the particles from leaving the domain when the shifting is applied near a wall. C_{shift} is a diffusion coefficient which value was calibrated from various test cases (in particular the Taylor-Green vortices case [76, 155],

and a schematic 2-D dam-break case described in section 4.2.1.4) and taken equal to 0.7 for the Wendland kernel.

In their work, Lind *et al.* [76] observed it was necessary to introduce an additional term in the concentration gradient of equation (3.26) in order to avoid particle clumping (see equations (2.108) and (2.109)). This was not the case in the present work due to the fact that we use a Wendland kernel, which is known to avoid particle clumping due to the positiveness of its Fourier transform [25] (Lind *et al.* had used a quintic spline kernel, which does not satisfy this property).

As stated in Chapter 2, applying the particle shift close to the free-surface would lead to an unphysical behaviour of the particles due to the kernel truncation, which is not accounted for near the free-surface, even with our boundary conditions. To avoid this issue, the shift is not applied to the particles which distance to the free-surface is lower than $hq_{max}/2$ (q_{max} is the dimensionless kernel support size, see (2.10)). This criterion was established by numerical experiments on the dam-break over a wedge case (section 4.2.1.4). It was expressed as a function of hq_{max} in order to have the same number of particle layers not shifted near the free surface, regardless of the kernel and smoothing length choices. When defining the position shift through (3.26) and using $\boldsymbol{G}_a^{\gamma}\{1\}$ to determine the normal to the free-surface, this method is equivalent to the one proposed by Lind *et al.* [76] that consists in applying a modified particle shift near the free-surface (see equation (2.110)).

In the end, the particles displacement and update of their temperature is done through:

$$\begin{cases} \boldsymbol{r}_a^{n+1/2} = \boldsymbol{r}^n + \dfrac{\delta t}{2}\boldsymbol{v}_a^{n+1/2} \\ T_a^{n+1} = T_a^n + \delta t L_a^{\gamma}\{K_{E,b}^{n+1}, T_b^n\} \\ \boldsymbol{r}_a^* = \boldsymbol{r}_a^{n+1/2} + \delta \boldsymbol{r}_a \ \text{ with } \ \delta \boldsymbol{r}_a = -C_{\text{shift}} h^2 \boldsymbol{G}_a^{\gamma}\{1\} \\ (T_a, k_a, \epsilon_a)^{n+1} \leftarrow (T_a, k_a, \epsilon_a)^{n+1} + \boldsymbol{G}_a^{\gamma,-}\{(T_b, k_b, \epsilon_b)^{n+1}\} \cdot \delta \boldsymbol{r}_a \\ \boldsymbol{v}_a^{n+1} = \boldsymbol{v}_a^{n+1/2} + \boldsymbol{G}_a^{\gamma,-}\{\boldsymbol{v}_b^{n+1/2}\} \cdot \delta \boldsymbol{r}_a \\ \boldsymbol{r}_a^{n+1} = \boldsymbol{r}_a^* + \dfrac{\delta t}{2}\boldsymbol{v}_a^{n+1} \end{cases} \tag{3.27}$$

in agreement with (3.13). The fields corrections by the convection term associated to the position shift (4th to 7th lines of system (3.27)) are performed with a symmetric gradient operator $\boldsymbol{G}_a^{\gamma,-}$ for accuracy reasons.

3.3 Wall boundary conditions

The four following subsections summarise our wall boundary conditions on the velocity, p, T, k and ϵ. Our technique is based on an analogy with Finite Volume and was validated in this work regarding the pressure, the temperature and the $k-\epsilon$ turbulent model. Note that the Neumann wall boundary conditions are applied through the surface term of the Laplacian operator given

by (2.154), like in mesh-based methods, whereas the Dirichlet boundary conditions are imposed at the vertex particles which are involved in the summations over $b \in \mathcal{P}$ in the Laplacian, gradient and divergence operators. We recall here the expression of the discrete Laplacian boundary term, which will be much used in the following sections:

$$L_a^{bound}\{B_b, A_b\} = -\frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}(B_a\boldsymbol{\nabla}A_a + B_s\boldsymbol{\nabla}A_s)\cdot\boldsymbol{\nabla}\gamma_{as} \tag{3.28}$$

Except for the pressure, which case is treated in Section 3.3.4, the prescription of wall boundary conditions is done by imposing both the flux and the value of the field at the wall. Therefore, the compatibility of the fields values and fluxes at the wall must be ensured. This technique allows the Dirichlet and Neumann conditions to be imposed at different locations, which proved important in Finite Volumes.

3.3.1 Wall boundary conditions on the velocity

In the present section, as well as in sections 3.3.2 and 3.3.3, the particles a belong to $\mathcal{F}$ (free particles), whereas the particles b belong to $\mathcal{P} = \mathcal{F}\cup\mathcal{V}$ (free and vertex particles). A Dirichlet boundary condition is imposed on the Lagrangian velocity:

$$\boldsymbol{v}_v = \boldsymbol{v}_v^{wall},\ \boldsymbol{v}_s = \boldsymbol{v}_s^{wall} \tag{3.29}$$

the Lagrangian velocity of the walls $\boldsymbol{v}^{wall}$ being an input data. The model thus includes the treatment of forced wall movement through the velocities of the vertex and segments. The Lagrangian velocity at the segments is then defined by:

$$\boldsymbol{v}_s = \frac{1}{N_s}\sum_{V_s}\boldsymbol{v}_v \tag{3.30}$$

where V_s is the set of vertices linked to s and N_s its size (in 2D, $N_s = 2$). Note that since the Lagrangian velocity is not involved in any diffusion term, it is not necessary to impose a compatible Neumann condition on this field. On the other hand, a non-homogeneous Neumann condition is applied on $\boldsymbol{u}$ in the same way as in [35]. This is done through the boundary term of the velocity Laplacian in (3.22), which is written as:

$$\boldsymbol{L}_a^{bound}\{\mu_{E,b}, \boldsymbol{u}_b\} = -\frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}\left[\mu_{E,a}\left(\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}\right)_a + \mu_{E,s}\left(\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}\right)_s\right]|\boldsymbol{\nabla}\gamma_{as}| \tag{3.31}$$

Here we used the fact that $\boldsymbol{\nabla}\gamma_{as}$ is oriented along $\boldsymbol{n}_s$ by definition (see (2.141)) to write that $(\boldsymbol{\nabla}\boldsymbol{u})_s\cdot\boldsymbol{\nabla}\gamma_{as} = \left(\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}\right)_s\cdot\boldsymbol{\nabla}\gamma_{as}$. Defining a' a fictitious point placed at $\frac{\boldsymbol{r}_a+\boldsymbol{r}_s}{2}$ (see Figure 3.1) it is

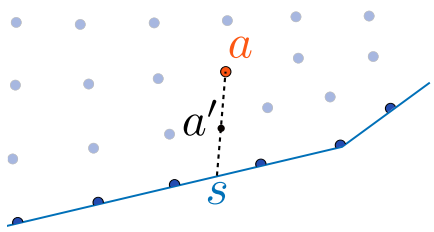


Figure 3.1: Sketch of the position of the fictitious point a' used for the imposition of Neumann wall boundary conditions on $\boldsymbol{u}$ and ϵ.

assumed that:

$$\left[\mu_{E,a}\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)_a + \mu_{E,s}\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)_s\right] \approx 2\mu_{E,a'}\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)_{a'} \tag{3.32}$$

The boundary term of the velocity Laplacian then reads:

$$\boldsymbol{L}_a^{bound}\{\mu_{E,b}, \boldsymbol{u}_b\} = -\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}}\mu_{E,a'}\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)_{a'}|\boldsymbol{\nabla}\gamma_{as}| \tag{3.33}$$

In the laminar case, the velocity distribution near the wall is almost linear, thus:

$$\mu\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)_{a'} \approx \mu\frac{\boldsymbol{v}_{as}\cdot\boldsymbol{t}_{as}}{\delta r_{as}}\boldsymbol{t}_{as} \tag{3.34}$$

where:

$$\boldsymbol{t}_{as} = \frac{\boldsymbol{v}_{as} - (\boldsymbol{v}_{as}\cdot\boldsymbol{n}_s)\,\boldsymbol{n}_s}{|\boldsymbol{v}_{as} - (\boldsymbol{v}_{as}\cdot\boldsymbol{n}_s)\,\boldsymbol{n}_s|} \tag{3.35}$$

and:

$$\delta r_{as} = \max(\boldsymbol{r}_{as}\cdot\boldsymbol{n}_s, \delta r) \tag{3.36}$$

with δr the initial interparticular space.

In the turbulent case, a two layers model is used for the velocity near the wall, according to (1.60) and (1.61). The Neumann condition then reads:

$$\mu_{E,a'}\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)_{a'}\cdot\boldsymbol{n}_s = u_{*,a'}^2\boldsymbol{t}_{as} \tag{3.37}$$

where $u_{*,a'}$ is the friction velocity (see section 1.2.1) at the wall seen by particle a, which is a solution of:

$$\begin{cases} \dfrac{\boldsymbol{v}_{as}\cdot\boldsymbol{t}_{as}}{u_{*,a'}} = y_{a'}^+ & \text{if } y_{a'}^+ \le y_{lim}^+ \\ \dfrac{\boldsymbol{v}_{as}\cdot\boldsymbol{t}_{as}}{u_{*,a'}} = \dfrac{1}{\kappa}\ln\left(\dfrac{\delta r_{as}u_{*,a'}}{\nu}\right) + 5.2 & \text{if } y_{a'}^+ > y_{lim}^+ \end{cases} \tag{3.38}$$

where $y^+_{lim} = \frac{1}{\kappa}$ with κ the Von Kármán constant (see Table 1.1, p.19) and:

$$y^+_{a'} = \frac{\delta r_{as} u_{k,a'}}{\nu} \tag{3.39}$$

In the inertial sub-layer (second line of (3.38)), $u_{*,a'}$ is computed through an iterative process (involving 9 iterations). In (3.39), $u_{k,a'}$ is a friction velocity based on the turbulent kinetic energy which is defined as:

$$u_{k,a'} = C_\mu^{\frac{1}{4}} k_a^{\frac{1}{2}} \tag{3.40}$$

Note that k_a is used in the above equation instead of $k_{a'}$ since a homogeneous Neumann condition is applied on k at the wall so it is assumed that $k_{a'} \approx k_a$. The second line of (3.38) is solved through an iterative process. On the other hand, a compatible Dirichlet condition is applied at the vertex particles (and segments) by letting their tangential velocity evolve according to the viscous term:

$$\boldsymbol{u}_v^{n+1} = \boldsymbol{u}_v^n + \delta t \, \frac{1}{\rho} \boldsymbol{L}_v^\gamma \{\mu_{E,b}, \boldsymbol{u}_b\} \tag{3.41}$$

The normal component of $\boldsymbol{u}_v^{n+1}$ is imposed to be equal to zero by projecting $\boldsymbol{u}_v^{n+1}$ (and $\boldsymbol{u}_s^{n+1}$) along the tangent to the wall:

$$\boldsymbol{u}_v^{n+1} \leftarrow \boldsymbol{u}_v^{n+1} - (\boldsymbol{u}_v^{n+1} \cdot \boldsymbol{n}_v) \, \boldsymbol{n}_v \tag{3.42}$$

This technique for imposing a kind of 'slip' velocity at the wall for high Reynolds number simulations was used in [35] for SPH and in other CFD codes [49].

3.3.2 Wall boundary conditions on k and ϵ

A homogeneous Neumann condition is applied on k as in [35], assuming that:

$$\mu_{k,a} \left(\frac{\partial k}{\partial \boldsymbol{n}} \right)_a \approx \mu_{k,s} \left(\frac{\partial k}{\partial \boldsymbol{n}} \right)_s = 0 \tag{3.43}$$

which gives:

$$L_a^{bound} \{\mu_{k,b}, k_b\} = 0 \tag{3.44}$$

A compatible Dirichlet boundary condition on k is imposed at all vertex particles v through:

$$k_v = \frac{1}{N_v} \sum_{s \in \mathcal{S}_v} k_s \ , \ k_s = \frac{1}{\alpha_s} \sum_{b \in \mathcal{F}} V_b k_b w_{sb} \tag{3.45}$$

where $\mathcal{S}_v$ is the set of segments linked to v, N_v is its size and α_s is the Shepard filter [130]:

$$\alpha_s = \sum_{b \in \mathcal{P}} V_b w_{sb} \tag{3.46}$$

As for the dissipation ϵ, it was necessary to improve the treatment of the diffusion boundary term in (3.15) compared to what was proposed in [35], in order to obtain better results close to the walls. A non-homogeneous Neumann condition is applied on ϵ in (3.15) by imposing the terms $\mu_{\epsilon,s}\left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_s$ and $\mu_{\epsilon,a}\left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_a$ in:

$$L_a^{bound}\left\{\mu_{\epsilon,b},\epsilon_b\right\} = -\frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}\left[\mu_{\epsilon,s}\left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_s + \mu_{\epsilon,a}\left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_a\right]|\boldsymbol{\nabla}\gamma_{as}| \tag{3.47}$$

Since ϵ quickly varies close to the wall a similar treatment as for the velocity field is applied:

$$\left[\mu_{\epsilon,s}\left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_s + \mu_{\epsilon,a}\left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_a\right] \approx 2\frac{\mu_{T,a'}}{\sigma_\epsilon}\left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_{a'} \tag{3.48}$$

where $\boldsymbol{r}_{a'} = \frac{1}{2}(\boldsymbol{r}_a + \boldsymbol{r}_s)$ as in section 3.3.1. Here it was considered that since $\mu << \mu_T$, $\mu_\epsilon \approx \frac{\mu_T}{\sigma_\epsilon}$. We assume that the theory of zero pressure-gradient turbulent boundary layer on a plane (local turbulent equilibrium) is valid and use the theoretical relations $\epsilon = \dfrac{u_k^3}{\kappa y}$ and $\mu_T = \kappa y u_k$, where y is a small distance to the wall, and thus obtain:

$$\frac{\mu_{T,a'}}{\sigma_\epsilon}\left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_{a'} = -\frac{2u_{k,a'}^4}{\sigma_\epsilon \delta r_{as}} \tag{3.49}$$

where $u_{k,a'}$ is defined through equation (3.40), so that the boundary term of the Laplacian applied to ϵ can be written as:

$$L_a^{bound}\left\{\mu_{\epsilon,b},\epsilon_b\right\} = \frac{4C_\mu}{\gamma_a\sigma_\epsilon}\sum_{s\in\mathcal{S}}\frac{k_a^2}{\delta r_{as}}|\boldsymbol{\nabla}\gamma_{as}| \tag{3.50}$$

On the other hand, the Dirichlet boundary condition is imposed at the vertex particles based on a FV-like formulation where the Dirichlet boundary condition on ϵ is 2nd order accurate in space on an orthogonal mesh. For this purpose, let us first consider a 1D situation with the same notations as before. We use the following Taylor series expansions:

$$\begin{cases} \epsilon_{a'} = \epsilon_a - \dfrac{\delta r_{as}}{2}\left(\dfrac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_a + \dfrac{\delta r_{as}^2}{8}\left(\dfrac{\partial^2 \epsilon}{\partial \boldsymbol{n}^2}\right)_a + \mathcal{O}(\delta r_{as}^3) \\ \epsilon_{a'} = \epsilon_s + \dfrac{\delta r_{as}}{2}\left(\dfrac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_s + \dfrac{\delta r_{as}^2}{8}\left(\dfrac{\partial^2 \epsilon}{\partial \boldsymbol{n}^2}\right)_s + \mathcal{O}(\delta r_{as}^3) \end{cases} \tag{3.51}$$

Subtracting these two equations yields:

$$\epsilon_s = \epsilon_a - \frac{\delta r_{as}}{2}\left[\left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_a + \left(\frac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_s\right] \tag{3.52}$$

In order to impose a Dirichlet boundary condition compatible with the Neumann condition imposed above, we use equations (3.48) and (3.49), which yields:

$$\epsilon_s = \epsilon_a - \delta r_{as} \left(\frac{\partial \epsilon}{\partial \boldsymbol{n}} \right)_{a'} = \epsilon_a + \frac{4 C_\mu^{3/4} k_a^{3/2}}{\kappa \delta r_{as}} \tag{3.53}$$

The extension to 2D is done by interpolating ϵ_s based on the value of the surrounding ϵ_a through:

$$\epsilon_s = \frac{1}{\alpha_s} \sum_{b \in \mathcal{F}} V_b \left(\epsilon_b + \frac{4 C_\mu^{3/4} k_b^{3/2}}{\kappa \delta r_{bs}} \right) w_{bs} \tag{3.54}$$

Finally, the Dirichlet boundary condition is imposed through the vertex particles by writing:

$$\epsilon_v = \frac{\epsilon_{s_1} + \epsilon_{s_2}}{2} \tag{3.55}$$

Note that the wall boundary conditions imposed on ϵ have a great impact on the flow representation. Starting from this second order formulation of the Dirichlet condition on ϵ gave better results on a turbulent Poiseuille flow case than using a first order formulation and working at point a instead of a'.

3.3.3 Wall boundary conditions on the temperature

We will now extend the ideas of sections 3.3.1 and 3.3.2 to the temperature. Again, what follows was inspired by the FV technique. Wall boundary conditions on the temperature may be of Neumann or Dirichlet type depending on the wall considered (adiabatic, isothermal, etc). In either case, their prescription is done by imposing both a heat flux through the segments and values of the temperature at the vertex particles. Thus, the compatibility of the temperature values and of the heat flux at the wall must be ensured. The surface part of the diffusion term in the temperature equation (3.25) is written as:

$$L_a^{bound}\{K_{E,b}, T_b\} = -\frac{2}{\gamma_a} \sum_{s \in \mathcal{S}} Q_{a'}^T |\boldsymbol{\nabla} \gamma_{as}| \tag{3.56}$$

where $Q_{a'}^T = K_{E,a'} \left(\frac{\partial T}{\partial \boldsymbol{n}} \right)_{a'}$ is the wall heat flux. As for the velocity and ϵ, it was considered that:

$$K_{E,s} \left(\frac{\partial T}{\partial \boldsymbol{n}} \right)_s + K_{E,a} \left(\frac{\partial T}{\partial \boldsymbol{n}} \right)_a \approx 2 K_{E,a'} \left(\frac{\partial T}{\partial \boldsymbol{n}} \right)_{a'} \tag{3.57}$$

The values of the heat flux and of the temperature at the wall depend on the type of wall boundary conditions.

3.3.3.1 Neumann wall boundary condition

In this case the temperature is considered as slowly varying close to the walls so that:

$$K_{E,a'}\left(\frac{\partial T}{\partial \boldsymbol{n}}\right)_{a'} \approx K_{E,s}\left(\frac{\partial T}{\partial \boldsymbol{n}}\right)_{s} \tag{3.58}$$

Thus the values of Q_s^T are directly imposed in the boundary term of the Laplacian according to the Neumann condition:

$$L_a^{bound}\{K_{E,b}, T_b\} = -\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}} Q_s^T |\boldsymbol{\nabla}\gamma_{as}| \tag{3.59}$$

A compatible Dirichlet condition is prescribed at the vertex particles, depending on the imposed flux:

$$T_v = \frac{1}{N_v}\sum_{s\in\mathcal{S}_v} T_s \,, T_s = \frac{1}{\alpha_s}\sum_{b\in\mathcal{F}} V_b\left(T_b - \frac{Q_s^T}{K}\delta r_{sb}\right) w_{sb} \tag{3.60}$$

by analogy with (3.54).

This formulation could be improved by considering rapidly varying temperature fields close to the wall. Then, the Neumann condition could be written in a similar way as for ϵ and the Dirichlet condition would depend on the temperature wall function. Since such improvements were not implemented, $K_{T,s}$ was used instead of $K_{T,a'}$ to build the compatible Dirichlet condition. This is why in the denominator of the flux term in (3.54) K appears instead of $K_{T,s}$, since ν_T is imposed as zero at the wall.

3.3.3.2 Dirichlet wall boundary condition

In this case, the value of the temperature is imposed at the vertex particles. The Q_s^T are then imposed in a consistent way with the Dirichlet condition, so that equation (3.56) reads:

$$L_a^{bound}\{K_{E,b}, T_b\} = -\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}} T_{*,a'} u_{k,a'} |\boldsymbol{\nabla}\gamma_{as}| \tag{3.61}$$

where $u_{k,a'}$ is a friction velocity and we defined $T_{*,a'} = \frac{Q_s^T}{u_{k,a'}}$ in analogy with (3.33) and (3.34). In laminar mode, a linear temperature distribution is applied:

$$u_{k,a'} T_{*,a'} = Q_s^T = K\frac{T_a - T_s}{\delta r_{as}} \tag{3.62}$$

whereas in turbulent mode, $u_{k,a'}$ is defined through (3.40). On the other hand, $T_{*,a'}$ is defined as:

$$T_{*,a'} = \frac{T_a - T_s}{T_{a'}^+} \tag{3.63}$$

where $T_{a'}^+$ (see section 1.3.1) is computed through a three-layer model according to (1.87):

$$\begin{cases} T_{a'}^+ = Pr\, y_{a'}^+ & \text{if } y_{a'}^+ < y_1^+ \\ T_{a'}^+ = a_2 - \dfrac{Pr_T}{2a_1 \left(y_{a'}^+\right)^2} & \text{if } y_1^+ \leq y_{a'}^+ < y_2^+ \\ T_{a'}^+ = \dfrac{Pr_T}{\kappa} \ln\left(y_{a'}^+\right) + a_3 & \text{if } y_{a'}^+ > y_2^+ \end{cases} \tag{3.64}$$

where $y_{a'}^+$ is defined as in (3.38) and $y_1^+, y_2^+, a_1, a_2, a_3$ are constants defined through (1.88). Recall that in this work the turbulent Prandlt number Pr_T is considered as constant and equal to 0.85.

3.3.4 Wall boundary conditions on the pressure

Recall that the (dynamic) pressure wall boundary condition reads (see section 1.1):

$$\frac{\partial}{\partial \boldsymbol{n}} \left(\frac{v^2}{2} + \frac{p^*}{\rho} \right) \Bigg|_{\partial\Omega_w} = \left(\nu \boldsymbol{\nabla}^2 \boldsymbol{u} \right) \cdot \boldsymbol{n} \big|_{\partial\Omega_w} \tag{3.65}$$

It was shown that the best pressure wall boundary condition in projection schemes (in terms of consistency) is a non-homogeneous Neumann condition that reads (see section 1.1.2):

$$\frac{\partial p^{n+1}}{\partial \boldsymbol{n}} \Bigg|_{\partial\Omega_w} = \frac{\rho}{\delta t} \tilde{\boldsymbol{v}}^{n+1} \cdot \boldsymbol{n} |_{\partial\Omega_w} = \left(\rho \boldsymbol{g} + \mu \boldsymbol{\nabla}^2 \boldsymbol{u}^n \right) \cdot \boldsymbol{n} |_{\partial\Omega_w} \tag{3.66}$$

which comes to neglecting the term in v^2 before the others in equation (3.65).

In the ISPH scheme proposed here the pressure boundary condition is imposed through the boundary term of the Laplacian operator involved in the pressure Poisson equation (3.23), similarly to what was presented in sections 3.3.1, 3.3.2 and 3.3.3.

Here we assume that $\left(\boldsymbol{\nabla} \tilde{p}\right)_a \cdot \boldsymbol{n}_s \approx \left(\boldsymbol{\nabla} \tilde{p}\right)_s \cdot \boldsymbol{n}_s$, which is justified by the fact that the pressure normal gradient does not vary much near the walls. The boundary term of the pressure Laplacian thus reads:

$$L_a^{bound}\{\tilde{p}_b\} = -\frac{2}{\gamma_a} \sum_{s \in \mathcal{S}} \boldsymbol{\nabla} \tilde{p}_s \cdot \boldsymbol{\nabla} \gamma_{as} \tag{3.67}$$

Projecting the second part of the momentum equation onto the normal $\boldsymbol{n}_v$ to the wall in v and substituting $\boldsymbol{v}_v^{n+1}$ by its imposed value yields:

$$\boldsymbol{\nabla} \tilde{p}_v^{n+1} \cdot \boldsymbol{n}_v = \frac{\rho}{\delta t} (\tilde{\boldsymbol{v}}_v^{n+1} - \boldsymbol{v}_v^{wall}) \cdot \boldsymbol{n}_v \tag{3.68}$$

The same applies to the segments since their velocity is calculated as the average of the velocities of the vertex particles directly linked to it (see (3.30)). Finally, the boundary term of the Laplacian operator applied to the pressure (with the hydrostatic correction, see equation (3.24)) can be written

as:

$$L_a^{bound}\{p_b + \rho g z_b\} = -\frac{2\rho}{\gamma_a}\sum_{s\in\mathcal{S}}\left(\frac{\tilde{\boldsymbol{v}}_s^{n+1} - \boldsymbol{v}_s^{wall}}{\delta t} + \boldsymbol{g}\right)\cdot\boldsymbol{\nabla}\gamma_{as} \tag{3.69}$$

One can check on a simple case that this pressure wall-boundary condition is physical. Let us consider the case of a fluid at rest with a free-surface in a rectangular tank. Following the steps of the projection method, the following holds:

$$\tilde{\boldsymbol{v}}_s^{n+1} = \delta t\boldsymbol{g} \tag{3.70}$$

because the velocity at the initial time n is equal to zero. Then the condition imposed on the pressure gradient at the wall is:

$$\boldsymbol{\nabla}\tilde{p}_s^{n+1}\cdot\boldsymbol{n}_s = \rho\boldsymbol{g}\cdot\boldsymbol{n}_s \tag{3.71}$$

which is the expected non-homogeneous boundary condition under gravity at rest. Thus we see that the condition (3.68) provides the exact pressure condition in order to balance gravity forces on a horizontal bed. This condition is non-homogeneous in many cases since the right-hand side depends on viscous and external forces through $\tilde{\boldsymbol{v}}^{n+1}$. The same pressure wall boundary condition is prescribed in the rotational projection scheme proposed by Hosseini & Feng [51] for SPH, although their formulation is less accurate due to the use of ghost particles for the boundary modelling.

3.4 Free-surface conditions

Recall that the imposition of free-surface conditions on $\boldsymbol{v}$, k, ϵ and T does not require any treatment since it is a homogeneous Neumann condition (see Table 1.2 and section 2.5). However, it is necessary to impose the Dirichlet condition on the pressure at the free-surface, which requires a tracking of the free-surface particles. In this work the free-surface detection is done in a similar way as in [72] (see section 2.5). However, the free-surface detection cannot be done with a classical SPH divergence (2.160) in the framework of the USAW boundary conditions, due to the kernel truncation close to the walls. Instead, the position divergence is computed through:

$$D_a^{\gamma,-}\{\boldsymbol{r}_b\} = -\frac{1}{\gamma_a\rho_a}\sum_{b\in\mathcal{P}} m_b\boldsymbol{r}_{ab}\cdot\nabla w_{ab} + \frac{1}{\gamma_a\rho_a}\sum_{s\in\mathcal{S}}\rho_s\boldsymbol{r}_{as}\cdot\nabla\gamma_{as} \tag{3.72}$$

which is the counterpart of (2.160) in the framework of the USAW boundary conditions. The boundary term in (3.72) avoids tracking particles that are close to the walls. Since γ_a is equal to one and there is no boundary term far away from the walls (even close to the free-surface), this position divergence gets lower values than the expected ones (2 in 2-D, 3 in 3-D) when approaching the free-surface. Then, a particle is considered as belonging to the latter when $D_a^{\gamma,-}\{\mathbf{r}_b\} \leq 1.5$ in 2-D and when $D_a^{\gamma,-}\{\mathbf{r}_b\} \leq 2.4$ in 3-D, as in [71].

The free-surface particle tracking is crucial for the stability of the ISPH simulations and the walls'

impermeability. Indeed, when particles with few neighbours are not identified as belonging to the free-surface, the Laplacian matrix is ill-conditioned. On the other hand, when a free-surface particle moves towards a non-immersed wall at high velocity it crosses it, since its pressure and the wall's pressure are set to zero. A modification of the free-surface detection is proposed in this work to fix the latter issue: a particle is identified as belonging to the free-surface only if it is sufficiently far from a wall. Thus, for each particle $a \in \mathcal{F}$ fulfilling the above-mentioned position divergence criterion, a test is performed to check whether it will cross the wall at the next time-step (with a small margin), which is done through the following criterion:

$$\boldsymbol{r}_{av} + \delta t \left(\boldsymbol{v}_a \cdot \frac{\boldsymbol{r}_{av}}{r_{av}^2} \right) < \frac{h q_{max}}{8} \tag{3.73}$$

If the latter relation holds, the free particle a and the vertex particle v are not identified as free-surface particles. This technique was tested on the triangular wedge case (Section 4.2.1.4).

It was observed on the free-surface test-cases presented in section 4.2 that the free-surface detection has important effects on the simulation behaviour. Attempts were made to work on the dynamic pressure in the whole time-scheme (3.19), in order to increase the accuracy of the results and to get rid of the hydrostatic correction (3.24). Though, this requires further investigation since it did not work with our formulation, maybe due to the lack of accuracy of the present free-surface detection.

3.5 Open boundaries

The imposition of inflow/outflow boundaries is done through a similar technique as that proposed by Kassiotis *et al.* [59] (see section 2.6). The open boundary is represented through a set of vertex particles and segments (see the Figure 2.8 for the definition of the vertex particles and segments). The set of vertex particles (resp. segments) belonging to an open boundary is denoted by $\mathcal{V}_{i/o}$ (resp. $\mathcal{S}_{i/o}$). The set of vertex particles (resp. segments) belonging to an inflow boundary is denoted by $\mathcal{V}_i$ (resp. $\mathcal{S}_i$). The set of vertex particles (resp. segments) belonging to an outflow boundary is denoted by $\mathcal{V}_o$ (resp. $\mathcal{S}_o$).

There are two main requirements for the imposition of open boundaries in the ISPH model proposed here: an algorithm to let particles enter and leave the domain, and the correct imposition of open boundary conditions on the fields. Regarding the algorithm for particles creation/destruction, the technique mentioned in section 2.6 is used. The idea is to let the masses of the vertex particles $v \in \mathcal{V}_{i/o}$ evolve over time as a function of the desired ingoing/outgoing mass flux through the segments $s \in \mathcal{S}_{i/o}$ directly connected to v. The vertex particles are then used to create/delete fluid particles, which is done by setting a minimum and a maximum value for their mass. This process is described in section 3.5.1. On the other hand, the imposition of open boundary conditions on the fields is done in a similar way as for wall boundaries: Dirichlet conditions are imposed at the vertex particles whereas the Neumann conditions are imposed through the segments by setting the

boundary terms of the Laplacian operators. We will see how this is done in section 3.5.2.

3.5.1 Particles creation/destruction

As evoked above, the mass of inlet/outlet vertex particles varies as a consequence of ingoing/outgoing mass fluxes. The mass evolution should not introduce any perturbations in the flow, so care must be taken that it evolves smoothly. The time-derivative of the mass, denoted by $\dot{m}_v^n$, is determined by the Eulerian velocity $\boldsymbol{u}_s$ imposed at the open boundaries:

$$\forall v \in \mathcal{V}_{i/o}, \quad \dot{m}_v^n = \frac{1}{N_{sv}} \sum_{s \in \mathcal{N}_{sv}} \rho S_s (\boldsymbol{u}_s - \boldsymbol{v}_s) \cdot \boldsymbol{n}_s \tag{3.74}$$

with $\mathcal{N}_{sv}$ the set of segments s directly connected to v, N_{sv} its size, S_s the surface of segment s (or length in 2-D). The vertex particles are then used to create/delete fluid particles, which is done by setting a minimum and a maximum value for their mass, more precisely $\pm 0.5 m_{ref}$ on a plane boundary, with m_{ref} the mass of a free particle. At an inlet, the mass flux (3.74) is positive and the mass of a vertex v increases until it reaches the higher threshold $+0.5 m_{ref}$. Then, a free particle is created at that location while m_{ref} is subtracted to m_v, so that m_v goes down to $-0.5 m_{ref}$. This process was illustrated in section 2.6 in the Figure 2.11. In this way the mass variation is smooth regarding space and time. At an outlet, the mass flux is negative and when a free particle crosses a segment to get out of the domain it is deleted and its mass is distributed onto the vertices directly linked to the segment, a weight $\beta_{a,v}$ being associated to each of these vertices. An illustration of the notations and of the fraction of segment area $\beta_{a,v}$ attributed to a vertex is provided on Figure 3.2. Let $\boldsymbol{p}_i$ be the projection of $\boldsymbol{r}_{av_i}$ on s (see Figure 3.2): $\boldsymbol{p}_i = \boldsymbol{r}_{av_i} - (\boldsymbol{r}_{av_i} \cdot \boldsymbol{n}_s)\boldsymbol{n}_s$ (v_i being one of the vertices linked to s). Then the coefficient $\beta_{a,v}$ is computed as:

- in 2-D, for v_0 and v_1 connected to s:

$$\begin{aligned} \beta_{a,v_0} &= \frac{\boldsymbol{p}_1 \cdot \boldsymbol{r}_{v_0 v_1}}{|\boldsymbol{r}_{v_0 v_1}|^2} \\ \beta_{a,v_1} &= \frac{\boldsymbol{p}_0 \cdot \boldsymbol{r}_{v_1 v_0}}{|\boldsymbol{r}_{v_0 v_1}|^2} = 1 - \beta_{a v_0} \end{aligned} \tag{3.75}$$

- in 3-D, for v_0, v_1 and v_2 connected to s:

$$\begin{aligned} \beta_{a,v_0} &= \frac{\frac{1}{2}[\boldsymbol{p}_2 \times \boldsymbol{r}_{v_2 v_1}] \cdot \boldsymbol{n}_s}{\frac{1}{2}[\boldsymbol{r}_{v_0 v_1} \times \boldsymbol{r}_{v_0 v_2}] \cdot \boldsymbol{n}_s} \\ \beta_{a,v_1} &= \frac{\frac{1}{2}[\boldsymbol{p}_0 \times \boldsymbol{r}_{v_0 v_2}] \cdot \boldsymbol{n}_s}{\frac{1}{2}[\boldsymbol{r}_{v_0 v_1} \times \boldsymbol{r}_{v_0 v_2}] \cdot \boldsymbol{n}_s} \\ \beta_{a,v_2} &= \frac{\frac{1}{2}[\boldsymbol{p}_1 \times \boldsymbol{r}_{v_1 v_0}] \cdot \boldsymbol{n}_s}{\frac{1}{2}[\boldsymbol{r}_{v_0 v_1} \times \boldsymbol{r}_{v_0 v_2}] \cdot \boldsymbol{n}_s} \end{aligned} \tag{3.76}$$

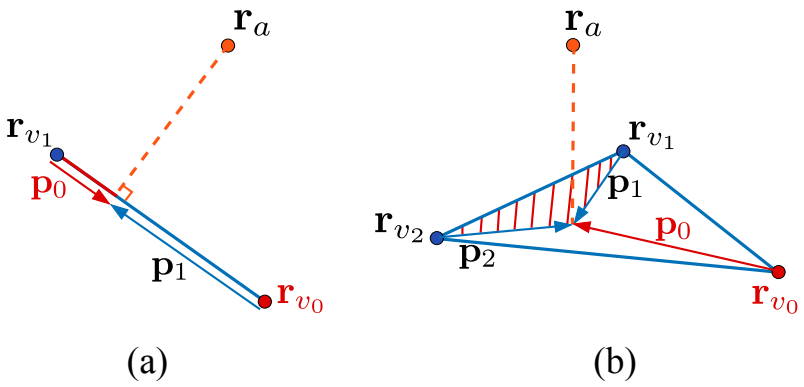


Figure 3.2: Open boundary technique based on the USAW boundary conditions: sketch of the notations for the computation of the fraction of segment area $\beta_{a,v}$ (represented in red) attributed to a vertex at the outlet [60]. (a) 2-D case; (b) 3-D case. The coefficient associated to the vertex v_0 is proportional to the red area so that the largest amount of mass goes to the closest vertex particle.

In this way the largest amount of mass is attributed to the closest vertex particle to the point where a fluid particle is destroyed, which minimises mass displacement. At an outflow, the mass variation is thus smooth in time but not exactly in space, although care was taken to distribute the mass as close as possible to the point where the particle is deleted.

The following mass evolution equation is thus solved $\forall v \in V_{i/o}$:

$$m_v^{n+1} = m_v^n + \delta t \dot{m}_v^n + \delta m_v^n \tag{3.77}$$

with δm_v^n the mass variation due to particle creation/destruction and $\dot{m}_v^n$ the mass flux corresponding to the imposed velocity at the open boundary. This mass update is performed for each half time-step of the particles position update. In the end, gathering this with equations (3.14),

(3.19) and (3.27), the total system to be solved reads:

$$
\begin{cases}
\dfrac{k_a^{n+1}-k_a^n}{\delta t} = \mathbb{P}_a^n + \mathbb{G}_a^n - \epsilon_a^n \dfrac{k_a^{n+1}}{k_a^n} + \dfrac{1}{\rho} L_a^\gamma\{\mu_{k,b}^n, k_b^n\} \\
\dfrac{\epsilon_a^{n+1}-\epsilon_a^n}{\delta t} = \dfrac{\epsilon_a^n}{k_a^n}\left(C_{\epsilon_1}\mathbb{P}_a^n + C_{\epsilon_3}\mathbb{G}_a^n - C_{\epsilon_2,Y,a}^n \epsilon_a^{n+1}\right) + \dfrac{1}{\rho} L_a^\gamma\{\mu_{\epsilon,b}^n, \epsilon_b^n\} \\
\dfrac{\tilde{\boldsymbol{v}}_a^{n+1/2}-\boldsymbol{v}_a^n}{\delta t} = \dfrac{1}{\rho}\boldsymbol{L}_a^\gamma\{\mu_{E,b}^{n+1}, \boldsymbol{u}_b^n\} - [\beta(T_a^n - T_0) - 1]\,\boldsymbol{g} \\
L_a^\gamma\{\tilde{p}_b^{n+1}\} = \dfrac{\rho}{\delta t} D_a^{\gamma,-}\{\tilde{\boldsymbol{v}}_b^{n+1/2}\} \\
\dfrac{\boldsymbol{v}_a^{n+1/2}-\tilde{\boldsymbol{v}}_a^{n+1/2}}{\delta t} = -\dfrac{1}{\rho}\boldsymbol{G}_a^{\gamma,+}\{\tilde{p}_b^{n+1}\}\boldsymbol{r}_a^{n+1/2} = \boldsymbol{r}^n + \dfrac{\delta t}{2}\boldsymbol{v}_a^{n+1/2} \\
\forall v \in \mathcal{V}_{i/o},\ \ m_v^{n+1/2} = m_v^n + \dfrac{\delta t}{2}\dot{m}_v^n + \delta m_v^{n+1/2} \\
T_a^{n+1} = T_a^n + \delta t L_a^\gamma\{K_{E,b}^{n+1}, T_b^n\} \\
\boldsymbol{r}_a^* = \boldsymbol{r}_a^{n+1/2} + \delta\boldsymbol{r}_a \ \text{ with } \ \delta\boldsymbol{r}_a = -C_{\text{shift}} h^2 \boldsymbol{G}_a^\gamma\{1\} \\
(T_a, k_a, \epsilon_a)^{n+1} \leftarrow (T_a, k_a, \epsilon_a)^{n+1} + \boldsymbol{G}_a^{\gamma,-}\{(T_b, k_b, \epsilon_b)^{n+1}\} \cdot \delta\boldsymbol{r}_a \\
\boldsymbol{v}_a^{n+1} = \boldsymbol{v}_a^{n+1/2} + \boldsymbol{G}_a^{\gamma,-}\{\boldsymbol{v}_b^{n+1/2}\} \cdot \delta\boldsymbol{r}_a \\
\boldsymbol{r}_a^{n+1} = \boldsymbol{r}_a^* + \dfrac{\delta t}{2}\boldsymbol{v}_a^{n+1} \\
\forall v \in \mathcal{V}_{i/o},\ \ m_v^{n+1} = m_v^{n+1/2} + \dfrac{\delta t}{2}\dot{m}_v^{n+1/2} + \delta m_v^{n+1}
\end{cases}
\tag{3.78}
$$

This model with the associated set of boundary conditions is referred to as ISPH-USAW in Chapters 4 and 5.

3.5.2 Imposition of the inflow/outflow boundary conditions

Two types of open boundaries must be treated, as was described in Chapter 1. At an inflow, a Dirichlet condition is imposed on the velocity, the temperature, k and ϵ. Besides, a homogeneous Neumann condition is imposed on the pressure. At an outflow, a Dirichlet condition is imposed on the pressure and a homogeneous Neumann condition is imposed on the velocity, the temperature, k and ϵ. Note that the distinction is done in terms of what is imposed on the fields but the algorithm makes it possible to have particles leaving the domain through an inflow boundary, which is necessary in case of a prescribed recirculation close to an inlet. In other words, inlet and outlet conditions can be handled by any open boundary at the same time. In the same way, the particles may enter the domain through an outflow boundary. As for wall boundary conditions, the Dirichlet conditions are imposed at the vertex particles, whereas the Neumann conditions are imposed through the segments.

3.5.3 Inflow boundaries

At an inflow boundary, the Dirichlet conditions on T, k and ϵ are imposed at the vertex particles and a homogeneous Neumann condition is imposed at the segments on those fields. The Dirichlet condition on the Eulerian velocity $\boldsymbol{u}$ is imposed at the vertex particles in the correction step of the projection method. The homogeneous Neumann condition on $\boldsymbol{u}$ is then imposed at the segments. The homogeneous Neumann condition on the pressure is imposed through the pressure Laplacian in the pressure Poisson equation. Moreover, the pressure of vertex particles belonging to an inlet is not computed through the Poisson equation. Instead, it is extrapolated from the surrounding fluid particles so that a homogeneous Neumann is imposed. Thus, the inflow boundary conditions imposed during the time-scheme read:

$$\forall v \in \mathcal{V}_i,\ \forall s \in \mathcal{S}_i,\quad \begin{cases} k_v^{n+1} = k_v^{inflow} \\ \epsilon_v^{n+1} = \epsilon_v^{inflow} \\ \left(\dfrac{\partial p}{\partial \boldsymbol{n}}\right)_s^{n+1} = 0 \\ \boldsymbol{u}_v^{n+1} = \boldsymbol{u}_v^{inflow} \\ T_v^{n+1} = T_v^{inflow} \end{cases} \tag{3.79}$$

The compatible inflow conditions read:

$$\forall v \in \mathcal{V}_i,\ \forall s \in \mathcal{S}_i,\quad \begin{cases} \left(\dfrac{\partial k}{\partial \boldsymbol{n}}\right)_s^{n+1} = 0 \\ \left(\dfrac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_s^{n+1} = 0 \\ p_v^{n+1} = \dfrac{1}{\alpha_v} \displaystyle\sum_{b \in \mathcal{F}} V_b p_b^{n+1} w_{vb} \\ \left(\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)_s^{n+1} \cdot \boldsymbol{n}_s = \dfrac{\boldsymbol{u}_{as}}{\delta r_{as}} \cdot \boldsymbol{n}_s \\ \left(\dfrac{\partial T}{\partial \boldsymbol{n}}\right)_s^{n+1} = 0 \end{cases} \tag{3.80}$$

The values of the fields at the segments of the inflow is deduced from a mean of the directly linked vertex particles. The Neumann conditions in equations (3.79) and (3.80) are imposed in the boundary terms of the Laplacian operators ((3.15), (3.23), (3.22), (3.25)). In equation (3.80), the fourth line corresponds to an estimation of the normal velocity gradient through a linear interpolation.

3.5.4 Outflow boundaries

At an outflow boundary, the homogeneous Neumann condition on T, k and ϵ is imposed at the segments in the boundary terms of the Laplacian operator applied to those fields. A compatible Dirichlet is deduced through an interpolation on the surrounding free particles. The Dirichlet on the pressure is imposed at the vertex particles. It can be either a fixed pressure value (like the hydrostatic pressure) or a radiative condition (see equation (1.29)). In the latter case, the outflow pressure condition reads, $\forall v \in \mathcal{V}_o$, $\forall s \in \mathcal{S}_o$:

$$\begin{cases} p_s^{n+1} = p_s^{outflow} = p_s^n - \dfrac{C\delta t}{\alpha_s} \displaystyle\sum_{b \in \mathcal{F}} V_b \frac{p_b^n - p_s^n}{\delta r_{sb}} w_{sb} \\ p_v^{outflow} = \dfrac{1}{N_{sv}} \displaystyle\sum_{s \in \mathcal{N}_{sv}} p_s^{outflow} \end{cases} \tag{3.81}$$

where δr_{sb} is defined as in (3.36) and α_s as in (3.46).

The Neumann condition imposed on the pressure is obtained through a linear interpolation of the surrounding free particles pressure. Thus, the outflow boundary conditions imposed during the time-scheme read:

$$\forall v \in \mathcal{V}_o,\ \forall s \in \mathcal{S}_o, \quad \begin{cases} \left(\dfrac{\partial k}{\partial \boldsymbol{n}}\right)_s^{n+1} = 0 \\ \left(\dfrac{\partial \epsilon}{\partial \boldsymbol{n}}\right)_s^{n+1} = 0 \\ p_v^{n+1} = p_v^{outflow} \\ \left(\dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)_s^{n+1} = 0 \\ \left(\dfrac{\partial T}{\partial \boldsymbol{n}}\right)_s^{n+1} = 0 \end{cases} \tag{3.82}$$

The compatible outflow boundary conditions read:

$$\forall v \in \mathcal{V}_o,\ \forall s \in \mathcal{S}_o, \quad \begin{cases} k_v^{n+1} = \dfrac{1}{\alpha_v} \displaystyle\sum_{b \in \mathcal{F}} V_b k_b^{n+1} w_{vb} \\ \epsilon_v^{n+1} = \dfrac{1}{\alpha_v} \displaystyle\sum_{b \in \mathcal{F}} V_b \epsilon_b^{n+1} w_{vb} \\ \left[\left(\dfrac{\partial (p + \rho g z)}{\partial \boldsymbol{n}}\right)_a^{n+1} + \left(\dfrac{\partial (p + \rho g z)}{\partial \boldsymbol{n}}\right)_s^{n+1}\right] \cdot \boldsymbol{n}_s = 2 \left[\dfrac{p_a^{n+1} - p_s^{outflow}}{\delta r_{as}} + \rho \boldsymbol{g} \cdot \boldsymbol{n}_s\right] \\ \boldsymbol{u}_v^{n+1} \cdot \boldsymbol{n}_v = \dfrac{1}{\alpha_v} \displaystyle\sum_{b \in \mathcal{F}} V_b \boldsymbol{u}_b^{n+1} \cdot \boldsymbol{n}_v w_{vb} \\ T_v^{n+1} = \dfrac{1}{\alpha_v} \displaystyle\sum_{b \in \mathcal{F}} V_b T_b^{n+1} w_{vb} \end{cases} \tag{3.83}$$

3.6 Solving the pressure Poisson equation

In the framework of the USAW boundary conditions, the pressure Poisson equation with hydrostatic correction (see equation (3.24)) reads:

$$\frac{2}{\gamma_a}\sum_{b\in\mathcal{P}}V_b\frac{p_{ab}+\rho g z_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab}-\frac{1}{\gamma_a}\sum_{s\in\mathcal{S}}\left[\boldsymbol{\nabla}p_a+\boldsymbol{\nabla}p_s+2\rho\boldsymbol{g}\cdot\boldsymbol{n}_s\right]\cdot\boldsymbol{\nabla}\gamma_{as}=\frac{\rho}{\delta t}D_a^{\gamma,-}\{\tilde{\boldsymbol{v}}_b^{n+1}\} \tag{3.84}$$

Taking the boundary conditions described in sections 3.3, 3.4 and 3.5 into account (equation (3.69) and 3rd lines of (3.79) and (3.83)), this equation becomes:

$$\frac{2}{\gamma_a}\left[\begin{array}{l}\displaystyle\sum_{b\in\mathcal{P}}V_b\frac{p_{ab}+\rho g z_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab}\\ \displaystyle-\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{i/o}}\rho\left(\frac{\tilde{\boldsymbol{v}}_s^{n+1}-\boldsymbol{v}_s^{wall}}{\delta t}+\boldsymbol{g}\right)\cdot\boldsymbol{\nabla}\gamma_{as}\\ \displaystyle-\sum_{s\in\mathcal{S}_i}\rho\boldsymbol{g}\cdot\boldsymbol{\nabla}\gamma_{as}\\ \displaystyle-\sum_{s\in\mathcal{S}_o}\left(\frac{p_a-p_s^{outflow}}{\delta r_{as}}+\rho\boldsymbol{g}\cdot\boldsymbol{n}_s\right)|\boldsymbol{\nabla}\gamma_{as}|\end{array}\right]=\frac{\rho}{\delta t}D_a^{\gamma,-}\{\tilde{\boldsymbol{v}}_b^{n+1}\} \tag{3.85}$$

First, let us keep in the left-hand side only the terms involving the unknown dynamic pressures:

$$\begin{aligned}\frac{2}{\gamma_a}\left[\begin{array}{l}\displaystyle\sum_{b\in\mathcal{P}}V_b\frac{p_{ab}+\rho g z_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab}\\ \displaystyle-\sum_{s\in\mathcal{S}_o}\frac{p_a+\rho g z_{as}}{\delta r_{as}}|\boldsymbol{\nabla}\gamma_{as}|\end{array}\right]&=\frac{\rho}{\delta t}D_a^{\gamma,-}\{\tilde{\boldsymbol{v}}_b^{n+1}\}+\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{i/o}}\rho\left(\frac{\tilde{\boldsymbol{v}}_s^{n+1}-\boldsymbol{v}_s^{wall}}{\delta t}\right)\cdot\boldsymbol{\nabla}\gamma_{as}\\ &+\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}\setminus\mathcal{S}_o}\rho\boldsymbol{g}\cdot\boldsymbol{\nabla}\gamma_{as}\\ &-\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}_o}\frac{p_s^{outflow}}{\delta r_{as}}|\boldsymbol{\nabla}\gamma_{as}|\end{aligned} \tag{3.86}$$

The lines of the matrix corresponding to Dirichlet particles (particles on which a Dirichlet condition is imposed) are removed, as in section 2.5. Indeed, there is no need to solve the system for these particles. Besides, the product of the columns corresponding to Dirichlet particles with the unknown pressure vector is known and passed to the right-hand side. In the end the system to be solved does not involve the free-surface particles $e\in\mathcal{E}$ and inlet/outlet particles anymore and they

appear in the right-hand side:

$$\frac{2}{\gamma_a}\left[\begin{array}{l}\displaystyle\sum_{b\in\mathcal{P}\setminus(\mathcal{E}\cup\mathcal{V}_{i/o})} V_b\frac{p_{ab}+\rho g z_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab}\\ \displaystyle+\sum_{b\in\mathcal{E}\cup\mathcal{V}_{i/o}} V_b\frac{p_a+\rho g z_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab}\\ \displaystyle-\sum_{s\in\mathcal{S}_o}\frac{p_a+\rho g z_{as}}{\delta r_{as}}|\boldsymbol{\nabla}\gamma_{as}|\end{array}\right] = \frac{\rho}{\delta t}D_a^{\gamma,-}\{\tilde{\boldsymbol{v}}_b^{n+1}\}+\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}\setminus\mathcal{S}_o}\rho\boldsymbol{g}\cdot\boldsymbol{\nabla}\gamma_{as}$$
$$+\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{i/o}}\rho\left(\frac{\tilde{\boldsymbol{v}}_s^{n+1}-\boldsymbol{v}_s^{wall}}{\delta t}\right)\cdot\boldsymbol{\nabla}\gamma_{as}$$
$$-\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}_o}\frac{p_s^{outflow}}{\delta r_{as}}|\boldsymbol{\nabla}\gamma_{as}|$$
$$+\frac{2}{\gamma_a}\sum_{b\in\mathcal{E}\cup\mathcal{V}_{i/o}}V_b\frac{p_b^{imposed}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab} \tag{3.87}$$

where $p_b^{imposed}$ denotes either p_b^{inflow}, $p_b^{outflow}$ or 0 (when $b\in\mathcal{E}$). This equation corresponds to a linear system:

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{B} \tag{3.88}$$

where $\boldsymbol{x}$ is the unknown vector of all particles dynamic pressures: $x_a = p_a^*$, $\boldsymbol{B}$ is the vector of right-hand side values at all particles:

$$\begin{aligned}B_a \;&= \frac{\rho}{\delta t}D_a^{\gamma,-}\{\tilde{\boldsymbol{v}}_b^{n+1}\}+\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}\setminus\mathcal{S}_o}\rho\boldsymbol{g}\cdot\boldsymbol{\nabla}\gamma_{as}+\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{i/o}}\rho\left(\frac{\tilde{\boldsymbol{v}}_s^{n+1}-\boldsymbol{v}_s^{wall}}{\delta t}\right)\cdot\boldsymbol{\nabla}\gamma_{as}\\ &-\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}_o}\frac{p_s^{outflow}}{\delta r_{as}}|\boldsymbol{\nabla}\gamma_{as}|+\frac{2}{\gamma_a}\sum_{b\in\mathcal{E}\cup\mathcal{V}_{i/o}}V_b\frac{p_b^{imposed}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab}\end{aligned} \tag{3.89}$$

$\boldsymbol{A}$ is a sparse matrix corresponding to the discrete Laplacian operator:

$$\left\{\begin{array}{l}A_{aa}=\dfrac{2}{\gamma_a}\displaystyle\sum_{b\in\mathcal{P}}V_b\frac{\boldsymbol{r}_{ab}}{r_{ab}^2}\cdot\boldsymbol{\nabla}w_{ab}-\frac{2}{\gamma_a}\sum_{s\in\mathcal{S}_o}\frac{|\boldsymbol{\nabla}\gamma_{as}|}{\delta r_{as}}\\ A_{ab}=-\dfrac{2}{\gamma_a}V_b\dfrac{\boldsymbol{r}_{ab}}{r_{ab}^2}\cdot\boldsymbol{\nabla}w_{ab}\end{array}\right. \tag{3.90}$$

The Laplacian matrix is non-symmetric because of the term involving outlet segments and because V_b is not constant with the USAW boundary conditions: the volume of the vertex particles is lower than that of the free particles.

To solve system (3.88), the linear solvers GMRES [125] and Bi-CGSTAB [153] were implemented on a CPU sequential code. The Bi-CGSTAB algorithm was also implemented on a GPU massively parallel code (see section 3.9). In both cases, it was chosen not to use a library for the matrix inversion, but to implement the algorithm, in order to avoid the matrix construction and storage which requires much memory. Indeed, the solver is meant to invert only the matrix given by (3.90).

Table 3.1: Non-preconditioned Bi-CGSTAB algorithm for the resolution of $\boldsymbol{Ax} = \boldsymbol{B}$ [153].

1. $\boldsymbol{x}_0$ is an initial bet; $\boldsymbol{r}_0 = \boldsymbol{B} - \boldsymbol{A}\boldsymbol{x}_0$;
2. $\hat{\boldsymbol{r}}_0$ is an arbitrary vector that satisfies
3. $(\hat{\boldsymbol{r}}_0, \boldsymbol{r}_0) \neq 0$, for instance, $\hat{\boldsymbol{r}}_0 = \boldsymbol{r}_0$
4. $\rho_0 = \alpha = \omega_0 = 1$;
5. $\boldsymbol{v}_0 = \boldsymbol{p}_0 = 0$;
6. for $i = 1, 2, 3, ...,$
7. $\rho_i = (\hat{\boldsymbol{r}}_0, r_{i-1}); \beta = (\rho_i/\rho_{i-1})(\alpha/\omega_{i-1})$;
8. $\boldsymbol{p}_i = \boldsymbol{r}_{i-1} + \beta(\boldsymbol{p}_{i-1} - \omega_{i-1}\boldsymbol{v}_{i-1})$;
9. $\boldsymbol{v}_i = \boldsymbol{A}\boldsymbol{p}_i$;
10. $\alpha = \rho_i/(\hat{\boldsymbol{r}}_0, \boldsymbol{v}_i)$;
11. $\boldsymbol{s} = \boldsymbol{r}_{i-1} - \alpha\boldsymbol{v}_i$;
12. $\boldsymbol{t} = \boldsymbol{A}\boldsymbol{s}$;
13. $\omega_i = (t, s)/(t, t)$;
14. $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \alpha\boldsymbol{p}_i + \omega_i\boldsymbol{s}$;
15. if $\boldsymbol{x}_i$ is accurate enough: quit;
16. $\boldsymbol{r}_i = \boldsymbol{s} - \omega_i\boldsymbol{t}$;
17. end;

This is particularly important in the GPU code where memory storage and memory access (in particular transfers of data between blocks) are the most limiting actions in terms of performance. Let us consider the case of the Bi-CGSTAB algorithm, given in Table 3.1. The result of each matrix-vector product ($\boldsymbol{A}\boldsymbol{x}_0$, $\boldsymbol{A}\boldsymbol{p}_i$, $\boldsymbol{A}\boldsymbol{s}$) is known:

$$\boldsymbol{A}\boldsymbol{x} = \frac{2}{\gamma_a}\left[\sum_{b\in\mathcal{P}} V_b \frac{x_{ab}}{r_{ab}^2}\boldsymbol{r}_{ab}\cdot\boldsymbol{\nabla}w_{ab} - \sum_{s\in\mathcal{S}_o}\frac{x_a}{\delta r_{as}}|\boldsymbol{\nabla}\gamma_{as}|\right] \tag{3.91}$$

Thus, each matrix-vector product ($\boldsymbol{A}\boldsymbol{x}_0$, $\boldsymbol{A}\boldsymbol{p}_i$, $\boldsymbol{A}\boldsymbol{s}$) is stored in an array through the call of a function that directly provides the result, without having to store the matrix. Note that in the GPU code, all the vector-vector operations (dot products, norms computations, etc.) are performed using the Cublas library, which is optimised for this purpose.

In the case of confined flows, if no Dirichlet condition is imposed the system has an infinity of solutions, and the matrix $\boldsymbol{A}$ is not invertible. It is made invertible by adding a small perturbation through a slight reinforcement of the diagonal terms A_{aa} and by imposing the mean of the right-hand side components to be zero.

3.7 Numerical stability

To our knowledge, there is no published theoretical study of ISPH numerical stability. Making a count of all the variables involved in the discrete system of equations, it appears that the maximum

time-step size for numerical stability is a function of 7 variables:

$$\delta t_{max} = \phi\left(\rho, \nu, K, \beta g, v_{max}, h, \Delta T\right) \tag{3.92}$$

where v_{max} is the maximum velocity of the flow, h the smoothing length and $\Delta T = T_h - T_c$ with T_h the highest and T_c the lowest temperature in the flow. The system counts 8 variables with 4 units, but the density is the only one involving a mass unit. The Vaschy-Buckingham theorem states it is determined by 4 dimensionless numbers:

$$\begin{aligned} \frac{\delta t_{max} v_{max}}{h} &= \psi\left(\frac{v_{max} h}{\nu}, \frac{\nu}{K}, \frac{\beta g \Delta T h^3}{\nu^2}\right) \\ &= \psi\left(Re_0, Pr, Gr_0\right) \end{aligned} \tag{3.93}$$

where Re_0 is a numerical Reynolds number, Pr is the Prandlt number and Gr_0 is a numerical Grashoff number. To obtain more details about the function ψ, it would be necessary to make a Neumann stability analysis as the one presented in section 2.3.1.3. Though, obtaining a formula like (2.98) for the stability domain is made difficult by the additional heat equation. Anyway, such an analysis does not take the presence of walls and of a free-surface into account, so it would still be necessary to use the empirical conditions (2.131), which is what is done here:

$$\delta t = \min\left(C_{CFL}\frac{h}{v_{max}}, C_{visq}\frac{h^2}{\nu}\right) \tag{3.94}$$

The value of $C_{visq} = 0.125$ is the same as for WCSPH schemes (see section 2.3.1.3), and the CFL number C_{CFL} was taken as 0.2 in the present work, based on crude numerical tests. Recall that several values of CFL number in ISPH are found in the literature, ranging from 0.1 to 0.4 (see section 2.3.2.4).

Following [148], in turbulent conditions one may use the same kind of condition but replacing ν by $\max(\nu_T)$ and Pr by Pr_T.

Note that no stability condition relative to the temperature was used in this work (*i.e.* no influence of Pr and Gr_0 in (3.94)), which was not a problem in the cases presented in section 4.3, except at very high Grashoff numbers on the case of a turbulent flow in a differentially heated rectangular cavity presented in section 4.3.2.2. Further investigation would give a more general stability condition of the form:

$$\delta t = \min\left(C_{CFL}\frac{h}{v_{max}}, C_{visq}\frac{h^2}{\nu}, C_{diff}\frac{h^2}{K}, C_{Gr}\frac{\beta g \Delta T h^2}{v_{max}^3}\right) \tag{3.95}$$

3.8 Computation of the kernel renormalisation factor and its gradient

As explained in section 2.4.1.2, computing γ_a through a governing equation (2.137) leads to an additional time-step size restriction that proved to prevail on the CFL condition quite often in ISPH, thus destroying the advantage of using v_{max} instead of the speed of sound in the CFL condition (see section 2.3.2.4). It is thus advantageous to compute γ_a through an analytical equation in our model. Here a technique is proposed to analytically compute γ_a in 2-D (section 3.8.1). Then, the computation of γ_a and its gradient in 3-D are briefly dealt with in section 3.8.2.

Recall that γ_a is defined as:

$$\gamma_a = \int_{\Omega_a} w(|\boldsymbol{r}_a - \boldsymbol{r}'|) d\boldsymbol{r}' \tag{3.96}$$

while its gradient is defined as:

$$\boldsymbol{\nabla}\gamma_a = \int_{\partial\Omega\cap\Omega_a} w(|\boldsymbol{r}_a - \boldsymbol{r}'|)\boldsymbol{n}(\boldsymbol{r}')d\Gamma' = \sum_{s\in\mathcal{S}} \int_{\partial\Omega_s\cap\Omega_a} w(|\boldsymbol{r}_a - \boldsymbol{r}'|)\boldsymbol{n}(\boldsymbol{r}')d\Gamma' = \sum_{s\in\mathcal{S}} \boldsymbol{\nabla}\gamma_{as} \tag{3.97}$$

3.8.1 Computation of the kernel renormalisation factor and its gradient in 2-D

The computation of $\boldsymbol{\nabla}\gamma_a$ in 2-D is done through the analytical formula proposed by Ferrand *et al.* [35], which serves to compute the $\boldsymbol{\nabla}\gamma_{as}$ terms appearing in equation (3.97). These terms are the ones involved in the differential operators (2.145), (2.146) and (2.151). On the other hand, a technique to compute γ_a through an analytical formula is proposed here and was used for all the test-cases presented in Chapter 4. It follows the idea proposed by Feldman and Bonet [33], which consists in writing γ_a as a boundary integral by applying the Gauss theorem to (3.96):

$$\gamma_a = -\int_{\partial\Omega} \mathbf{W}\left(\left|\mathbf{r}_a - \mathbf{r}'\right|\right) \cdot \mathbf{n}\left(\mathbf{r}'\right) d^{n-1}\Gamma\left(\mathbf{r}'\right) \tag{3.98}$$

where $\mathbf{W}$ is defined as:

$$w\left(q\right) = \nabla \cdot \mathbf{W}\left(q\right) \tag{3.99}$$

Since w is a radial function, so is $\mathbf{W}$:

$$\mathbf{W}\left(q\right) = -\varphi\left(q\right)\tilde{\mathbf{r}} \tag{3.100}$$

where $q = \dfrac{\tilde{r}}{h}$ and $\tilde{\mathbf{r}} = \boldsymbol{r}_a - \boldsymbol{r}'$. In polar coordinates $w(q) = \nabla \cdot \boldsymbol{W}(q)$ reads:

$$w(q) = -\frac{1}{\tilde{r}}\frac{\partial}{\partial\tilde{r}}\left[\tilde{r}^2\varphi(q)\right] \tag{3.101}$$

Multiplying by q and integrating gives:

$$\varphi(q) = -\frac{1}{q^2}\int_q^{q_{max}} q' w(q') dq' \tag{3.102}$$

The calculations were done for the $5th$ order Wendland kernel (2.10), which yields:

$$\varphi\left(q\right) = \frac{1}{2\pi h^2 q^2}\left(1-\frac{q}{2}\right)^5\left(1+\frac{5q}{2}+2q^2\right) \quad \text{for } q \leqslant 2 \tag{3.103}$$

Then we have :

$$\gamma_a = \int_{\partial\Omega} \varphi\left(q\right)\tilde{\mathbf{r}}\cdot\mathbf{n}\left(\tilde{\mathbf{r}}\right)d^{n-1}\Gamma\left(\tilde{\mathbf{r}}\right) \tag{3.104}$$

As pointed out in [33], the function $h^2\varphi\left(q\right)$ presents a singularity in $q=0$, so that the Gauss theorem invoked to obtain (3.98) is only valid if the integration is done on $\partial\Omega\cup\partial\Omega_\epsilon$, with Ω_ϵ a small sphere of radius ϵh centred on $q=0$. By decreasing ϵ to zero, it is possible to show that the integral over $\partial\Omega_\epsilon$ is equal to 1. Thus:

$$\gamma_a = 1 - \sum_s \tilde{\gamma}_{as} \tag{3.105}$$

with:

$$\tilde{\gamma}_{as} = -\mathbf{n}_s \cdot \int_s \varphi\left(\frac{\tilde{r}}{h}\right)\tilde{\mathbf{r}}d^{n-1}\Gamma\left(\tilde{\mathbf{r}}\right) \tag{3.106}$$

Recall $\mathbf{n}_s$ is the inward unit normal on segment s. Let $\mathbf{t}_s$ be the unitary vector tangential to s (see Figure 3.3). We have $\tilde{\mathbf{r}} = \mathbf{r}_0 - \mathbf{r}_a + y\mathbf{t}_s$ so (3.106) can be simplified to give:

$$\tilde{\gamma}_{as} = \mathbf{n}_s\cdot\mathbf{r}_{a0}\int_s \varphi\left(\frac{\tilde{r}}{h}\right)d^{n-1}\Gamma\left(\tilde{\mathbf{r}}\right) \tag{3.107}$$

where $\mathbf{r}_{a0} = \mathbf{r}_a - \mathbf{r}_0$ and $\mathbf{r}_0$ is the orthogonal projection of $\mathbf{r}_a$ on the segment s. Let y be the coordinate along $\mathbf{t}_s$, $r_{a0} = |\mathbf{r}_{a0}|$ the distance from the integration point to the segment and $q_{a0} = r_{a0}/h$. We define:

$$\zeta_i = \min\left(\frac{1}{2}\sqrt{\tilde{q}_{a0}^2 + \frac{y_i^2}{h^2}}, 1\right) \tag{3.108}$$

for $i=1,2$, with $\tilde{q}_{a0} = \min\left(q_{a0}, 2\right)$. y spans the interval $[y_1 = \boldsymbol{r}_{av_1}\cdot\boldsymbol{t}_s; y_2 = \boldsymbol{r}_{av_2}\cdot\boldsymbol{t}_s]$. It is then found that:

$$\tilde{\gamma}_{as} = \text{sign}\left(\mathbf{n}_s\cdot\mathbf{r}_{a0}\right)\frac{1}{4\pi}\left[\text{sign}\left(y_2\right)\psi\left(q_{a0},\zeta_2\right) - \text{sign}\left(y_1\right)\psi\left(q_{a0},\zeta_1\right)\right] \tag{3.109}$$

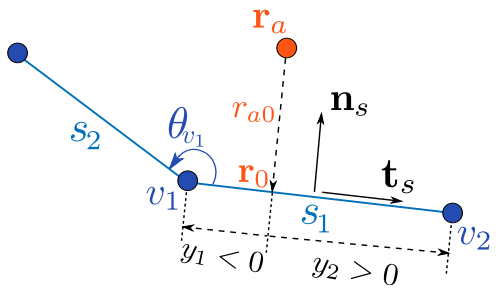


Figure 3.3: Notations for the calculation of $\tilde{\gamma}_{as}$.

with:

$$\begin{aligned}\psi(q,\zeta) &= q\sqrt{\zeta^2-\tfrac{q^2}{4}}\left[\begin{array}{l}-\frac{4}{3}\zeta^5+7\zeta^4-\left(\frac{5}{12}q^2+14\right)\zeta^3\\+\frac{7}{3}\left(q^2+5\right)\zeta^2-\frac{1}{4}\left(\frac{5}{8}q^2+21\right)q^2\zeta\\+\frac{7}{6}q^4+\frac{35}{6}q^2-7\end{array}\right]\\&-\left(\tfrac{5}{8}q^2+21\right)\tfrac{q^5}{16}\operatorname{arg\,cosh}\tfrac{2\zeta}{q}+2\arctan\sqrt{\tfrac{4\zeta^2}{q^2}-1}\end{aligned} \tag{3.110}$$

Let us now consider the particular case where a is located exactly on the straight line driven by the segment s. The limit of $\tilde{\gamma}_{as}$ when q_{a0} tends to zero (the y_i remaining different from zero) is:

$$\lim_{q_{a0}\longrightarrow 0}\tilde{\gamma}_{as} = \frac{1}{4}\left[\text{sign}\left(y_2\right)-\text{sign}\left(y_1\right)\right] \tag{3.111}$$

If the integration point is located inside the segment, y_1 and y_2 have opposite signs and y_2 is positive according to our notations, so we find $\tilde{\gamma}_{as} = 1/2$, as expected. On the contrary, if the point is located outside the segment, y_1 and y_2 have the same sign, and we obtain $\tilde{\gamma}_{as} = 0$. Thus, for a point located on a straight wall, (3.109) gives the expected result: $\gamma_a = 1/2$.

The case where the integration point is located at the intersection of two segments corresponds to a singularity. Let us consider a point a belonging to the segment s_1 and getting closer of one of its extremities $\mathbf{r}_v$, in the direction of segment s_2, which makes an internal angle θ_v with s_1 at the point $\mathbf{r}_v$ (see Figure 3.3). Let us assume that the lengths of the two segments are large enough so that only the segments s_1 and s_2 have a contribution. What we saw before shows that $\tilde{\gamma}_{as_1} = 1/2$ for any value of the distance $r_{av} > 0$. Making r_{av} tend to zero we obtain:

$$\tilde{\gamma}_{vs_2} = \frac{1}{2}-\frac{\theta_v}{2\pi} \tag{3.112}$$

Finally:

$$\begin{aligned}\gamma_v &= 1-\left(\tilde{\gamma}_{vs_1}+\tilde{\gamma}_{vs_2}\right)\\&= \frac{\theta_v}{2\pi}\end{aligned} \tag{3.113}$$

Table 3.2: Summary of the various cases for the calculation of $\tilde{\gamma}_{as}$ in 2-D.

Particle position	$\tilde{\gamma}_{\mathbf{as}}$
Near a wall	eqn (3.109)
On a segment	$\frac{1}{2}$
On a vertex v	$\frac{1}{2}\left(1-\frac{\theta_v}{2\pi}\right)$

which is the expected result. If the shape of the wall changes quickly close to the vertex particle v, other positive or negative contributions can appear in (3.105), but there is no singularity problem. In Table 3.2 the techniques used to compute $\tilde{\gamma}_{as}$ in all the situations are summarised.

It was checked that the computed results perfectly match the theoretical values of γ_a in cases of a straight infinite wall and of an arbitrary angle.

3.8.2 Computation of the kernel renormalisation factor and its gradient in 3-D

In 3-D, γ_a is computed through the governing equation proposed by Ferrand *et al.* [35] (equation (2.137), repeated here):

$$\frac{d\gamma_a}{dt} = \boldsymbol{\nabla}\gamma_a \cdot (\boldsymbol{v}_a - \boldsymbol{v}^{wall}) \tag{3.114}$$

On the other hand, $\boldsymbol{\nabla}\gamma_a$ is computed through an analytical formula proposed in [150] and slightly modified here so as to avoid numerical issues due to discontinuities in the original formula. The idea is to apply the Gauss theorem to the definition of $\boldsymbol{\nabla}\gamma_a$ so as to compute it as a sum of integrals over the edges of the segments, which are triangles in 3-D (note that they could be quadrangles). The method is explained in Appendix A, which results in the formula (A.14) for the computation of $\boldsymbol{\nabla}\gamma_a$, that was used in the 3-D cases presented in Chapter 5. Note that a technique to analytically compute γ_a in 3-D was proposed in [150], which could further improve the results on the 3-D cases.

3.9 Parallelisation in a GPU framework

Initially, the developments presented in this Chapter were introduced in a sequential code called Spartacus-2D developped at EDF, that was based on a WCSPH formulation using a first-order symplectic scheme with USAW wall boundary conditions. The ISPH model was introduced in this code, starting from a Bi-CGSTAB linear solver provided by Lee *et al.*. A GMRES solver was also implemented which showed better convergence rates compared to Bi-CGSTAB, although no quantitative comparisons are provided here. However, as evoked in section 2.9 the massive parallelisation of the

code is necessary in order to bring the method to an industrial level. This was done in a code called Sphynx, derived at EDF from the open-source code GPUSPH [45] but with simplifications regarding the available options [4]. Compared to GPUSPH, Sphynx does not allow to represent solid objects which movements can be affected by the flow, and it does not provide all the boundary conditions, time-scheme and kernel options. Besides, contrary to GPUSPH it is not able to run on multiple GPUs. On the other hand, Sphynx was developed in order to implement the USAW boundary conditions in a GPU framework. It also includes a treatment for inlet/outlet boundary conditions in WCSPH following [60]. The present ISPH model was implemented in this code, as well as the $k-\epsilon$ turbulence model and the buoyancy model. Besides, the inlet/outlet boundary conditions for ISPH presented in section 3.5 were implemented.

As GPUSPH, Sphynx is working in single precision in order to spare memory and be able to perform bigger simulations. The whole algorithm is computed on the GPU, data being exchanged with the CPU at the beginning of the simulation and for every output of a result file. As in GPUSPH, a Cartesian grid of cell size the kernel support is used in order to compute the neighbour list faster. The latter is stored in a table after each particle displacement. In order to get rid of numerical precision issues (enhanced by the use of single precision), the particles positions are stored relatively to their cell position (homogeneous precision, see [46]). The absolute positions are only recovered for visualisation and post-processing purposes. It was measured on the lid-driven cavity test-case with 2.5×10^5 particles that the computational time was divided by about a factor 50 compared to the CPU sequential code.

[4] GPUSH can be downloaded from http://www.gpusph.org/

Chapter 4

Validation on 2-D cases

Ce Chapitre traite de la validation du modèle présenté au Chapitre 3 sur des cas 2-D. Le modèle est comparé à d'autres modèles SPH, ainsi qu'aux Volumes Finis dans le cas d'écoulements confinés et à la méthode Volume of Fluid dans le cas d'écoulements à surface libre, avec des résultats très satisfaisants. Dans un premier temps on s'intéresse à la validation sur des écoulements isothermes. En régime laminaire, le cas de la cavité entraînée a été testé, les résultats obtenus montrant que le modèle améliore effectivement les résultats par rapport à des modèles SPH existants. On vérifie la capacité du modèle à prédire les forces exercées sur les objets sur un cas d'écoulement autour de cylindres et sur une rupture de barrage sur un obstacle. Un cas à surface libre présentant des parois plus complexes et mobiles est également présenté. La formulation des conditions aux frontières ouvertes est testée sur un canal de Poiseuille laminaire, sur une rupture de barrage schématique coupée et sur un cas de propagation de vague. Le modèle $k - \epsilon$ est testé sur un écoulement de Poiseuille turbulent dans un canal infini et sur un cas schématique de passe à poissons. Dans un deuxième temps, on s'intéresse à la validation sur des écoulements non-isothermes. Deux cas laminaires d'écoulements dans des cavités carrées chauffées différentiellement sont présentés, le deuxième présentant une paroi supérieure mobile. Ensuite, deux cas d'écoulements non-isothermes turbulents sont considérés : un écoulement dans un canal et un écoulement dans une cavité rectangulaire chauffée différentiellement.

This Chapter focuses on the validation of the buoyant incompressible SPH model described in Chapter 3 with USAW boundary conditions in 2-D (system (3.78) and the subsequent boundary conditions). The latter is referred to as ISPH-USAW in what follows. First, in the section 4.1 our notations are introduced. Then, the validation on 2-D isothermal flows is presented in the section 4.2. The ISPH algorithm itself, without USAW boundary conditions, is relatively well established [76], so that we do not present any validation on cases without walls in this work. In the section 4.2.1, bounded isothermal laminar flows are considered, for which reference results are widely available in the literature. This includes free-surface and confined flows, as well as flows with inlet/outlet boundaries. In the section 4.2.2, two isothermal confined turbulent flows are considered, one of them being a 2-D turbulent Poiseuille channel flow, which is the standard case for validation of the $k-\epsilon$ model. Finally, the section 4.3 focuses on the validation on 2-D non-isothermal confined laminar and turbulent flows. The $5th$ order Wendland kernel (2.10) was used for all the simulations with a smoothing length $h = 2\delta r$ (recall δr is the initial interparticular spacing). In all the simulations the reference density of the fluid is $\rho = 1000\ kg.m^{-3}$. The results obtained with ISPH-USAW are compared to mesh-based methods and to other SPH models. Comparisons with FV are provided for most confined cases, whereas comparisons with Volume of Fluid (VoF) are provided for most free-surface cases. The FV results were obtained with the *Code_Saturne* open-source software [8] and the VoF results with the OpenFOAM open-source software [140]. It seemed important to compare the results of the new model to reference methods, which is why so many confined cases were tested.

4.1 Nomenclature

In all cases the characteristic length of the flow is denoted by L and the characteristic velocity is denoted by U. The isothermal flows are characterised by the Reynolds number:

$$Re = \frac{UL}{\nu} \tag{4.1}$$

The non-isothermal flows are characterised by three other dimensionless numbers: the Prandlt number, the Grashoff number and, in case a non-zero heat flux is imposed through a wall, a bulk Nusselt number. The Prandlt number is defined as:

$$Pr = \frac{\nu}{K} \tag{4.2}$$

The Grashoff number is defined as:

$$Gr = \frac{\beta g \Delta T L^3}{\nu^2} \tag{4.3}$$

The Nusselt number is defined as:

$$Nu = \frac{LQ_0^T}{K\Delta T} \tag{4.4}$$

where $\Delta T = T_h - T_c$ with T_h defined as the highest temperature and T_c the lowest temperature of the flow. On the other hand, Q_0^T is an imposed heat flux through a wall (usually zero, except in section 4.3.1.1).

In what follows, the dimensionless variables are identified with a $+$ superscript. Our 2-D coordinates are denoted by (x, z) and y represents the distance to a wall in section 4.2.2.1. The components of the velocity field $\boldsymbol{v}$ are denoted by (v_x, v_z). Unless stated otherwise, the dimensionless variables are defined by:

$$\begin{cases} x^+ = \dfrac{x}{L}, & z^+ = \dfrac{z}{L}, & \boldsymbol{v}^+ = \dfrac{\boldsymbol{v}}{U}, & \nu_T^+ = \dfrac{\nu_T}{LU}, & t^+ = \dfrac{t}{\sqrt{gH}} \\ k^+ = \dfrac{k}{U^2}, & \epsilon^+ = \dfrac{\epsilon L}{U^3}, & T^+ = \dfrac{T}{T_h}, & p^+ = \dfrac{p}{\rho U^2/2} & \end{cases} \tag{4.5}$$

where H is a reference water height.

4.2 Validation on isothermal 2-D cases

4.2.1 Laminar flows

4.2.1.1 Lid-driven cavity

The lid-driven cavity test-case is classical in fluid dynamics and is much used to validate numerical models. It consists of a square closed cavity of size L (the characteristic length of the flow) whose lid slides laterally at a constant velocity U (the characteristic flow velocity), driving the fluid under the effect of the viscosity. For Reynolds numbers lower than about 7500 [112], it reaches a steady-state after some time. Then, it is possible to compare the results between different computational fluid dynamics (CFD) codes. In particular, the SPH results were compared to the ones obtained by Ghia *et al.* [38] with a multigrid simulation method, and to the ones obtained with *Code_Saturne*, a widely validated code based on FV [8]. The FV simulations were always done with 512×512 cells. Three Reynolds numbers (defined through (4.1)) were considered here: 100, 400 and 1000. A representation of the results obtained with the present ISPH-USAW model with 240×240 particles and FV after time-convergence for a Reynolds number of 1000 is presented Figure 4.1, which qualitatively shows that the two CFD codes give very similar results. Simulations on this test-case showed that the impermeability of the walls is granted by the ISPH-USAW model.

For the Reynolds number 100, we compared ISPH-USAW results to Yildiz *et al.*'s results [157] based on an ISPH model with the multiple boundary tangent method (ISPH-MBT). A discretisation of 120×120 particles was used in both methods. The velocity profiles in $x^+ = 1/2$ and $z^+ = 1/2$ are shown in Figure 4.2, where the same quality of results was obtained with both ISPH models compared to Ghia *et al.* and to FV results. We could not compare pressure results since there were

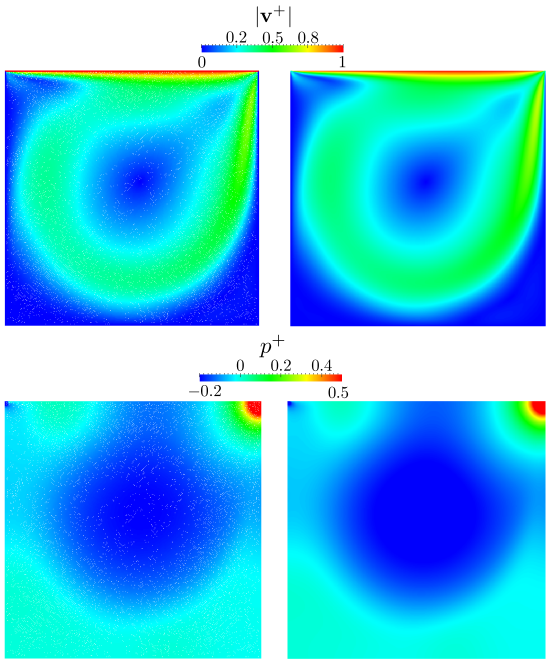


Figure 4.1: Lid-driven cavity case for $Re = 1000$: comparison of the results obtained after convergence with ISPH-USAW (left) and with FV (right).

none available in [157].

For the Reynolds number 400, we compared ISPH-USAW results to WCSPH using USAW boundary conditions (WCSPH-USAW). A discretisation of 200×200 particles was used in both methods. For WCSPH-USAW the numerical speed of sound was taken equal to $10U$, and a background pressure was imposed, without which cavities appear in the flow, in agreement with [72]. Besides, a Ferrari density correction (2.99) was applied, as adapted to WCSPH-USAW by Mayrhofer *et al.* [94]. The dimensionless velocity profiles are shown on the left side of Figure 4.3, where the same quality of results was obtained with ISPH-USAW and WCSPH-USAW compared to Ghia *et al.* and to FV. The dimensionless pressure profiles in $z^+ = 1/2$ and on the diagonal of the cavity, defined as that between the bottom-left and the top-right corners, are shown on the right side of Figure 4.3. It appears that WCSPH-USAW results are far inferior to ISPH-USAW results in terms of pressure prediction, even with a Ferrari density correction.

For the Reynolds number 1000, we compared our ISPH-USAW results to WCSPH-USAW and to the results obtained by Xu *et al.* [155] using ISPH with a classical ghost particles technique (ISPH-GP). A discretisation of 240×240 particles was used in all methods. The dimensionless velocity profiles are shown on the Figure 4.4, where the same quality of results was obtained with both ISPH models compared to Ghia *et al.* and to FV. The velocity results obtained with WCSPH-USAW are slightly inferior to the two ISPH models. Both ISPH models are much better than WCSPH in terms of pressure prediction, as can be seen in Figure 4.6. Finally, the computational time with ISPH-USAW was smaller than with WCSPH-USAW as shown in Table 4.1, and FV performed faster.

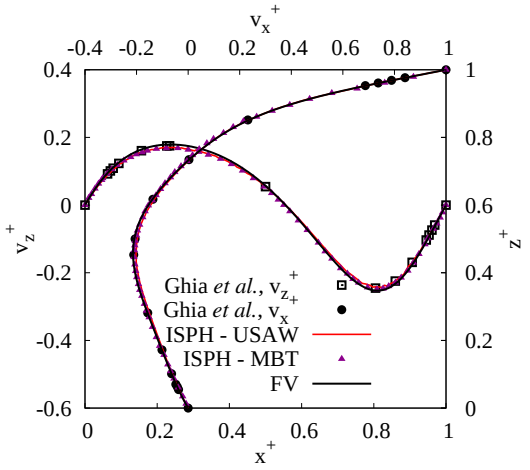


Figure 4.2: Lid-driven cavity for $Re = 100$. Dimensionless velocity profiles in $x^+ = 1/2$ and $z^+ = 1/2$. Comparison between ISPH-USAW, ISPH-MBT [157], FV and the results of Ghia *et al.* [38].

For the three Reynolds numbers ISPH-USAW results are in good agreement with the ones obtained with FV and by Ghia *et al.* in terms of velocity and pressure, which shows that the boundary conditions are imposed satisfactorily for laminar flows. It is expected that ISPH-MBT and ISPH-GP perform well on this test-case where the geometry is simple. Though, no convergence study was presented in the two latter works, so that the rate of convergence of those models is not known.

To quantify the error made with our ISPH model compared to the FV method, convergence studies were performed at a Reynolds number of 1000 where the results obtained with FV on a cavity discretised by 512×512 cells were taken as a reference. The L_2 error was calculated based on the values of the horizontal velocity field obtained by the ISPH method and by FV at all particles positions, through:

$$L_2 = \sqrt{\frac{1}{V_{tot}} \sum_{b \in \mathcal{P}} V_b \left(\frac{v_{x,b}^{sol} - v_{x,b}^{ref}}{v_{max}} \right)^2} \tag{4.6}$$

where $V_{tot} = \sum_{b \in \mathcal{P}} V_b$ is the total volume of the computational domain, v_x^{sol} is the horizontal velocity obtained by the ISPH model, v_x^{ref} is the horizontal velocity obtained with FV and $v_{max} = U$ is the maximum velocity of the flow. The results of the convergence study are shown on the Figure 4.5, where it appears that the rate of convergence of ISPH-USAW is close to 2, whereas WCSPH-USAW presents a convergence order less than one and an error about 10 times higher than ISPH-USAW. Note that to our knowledge there are no theoretical results concerning the convergence rate of the ISPH method.

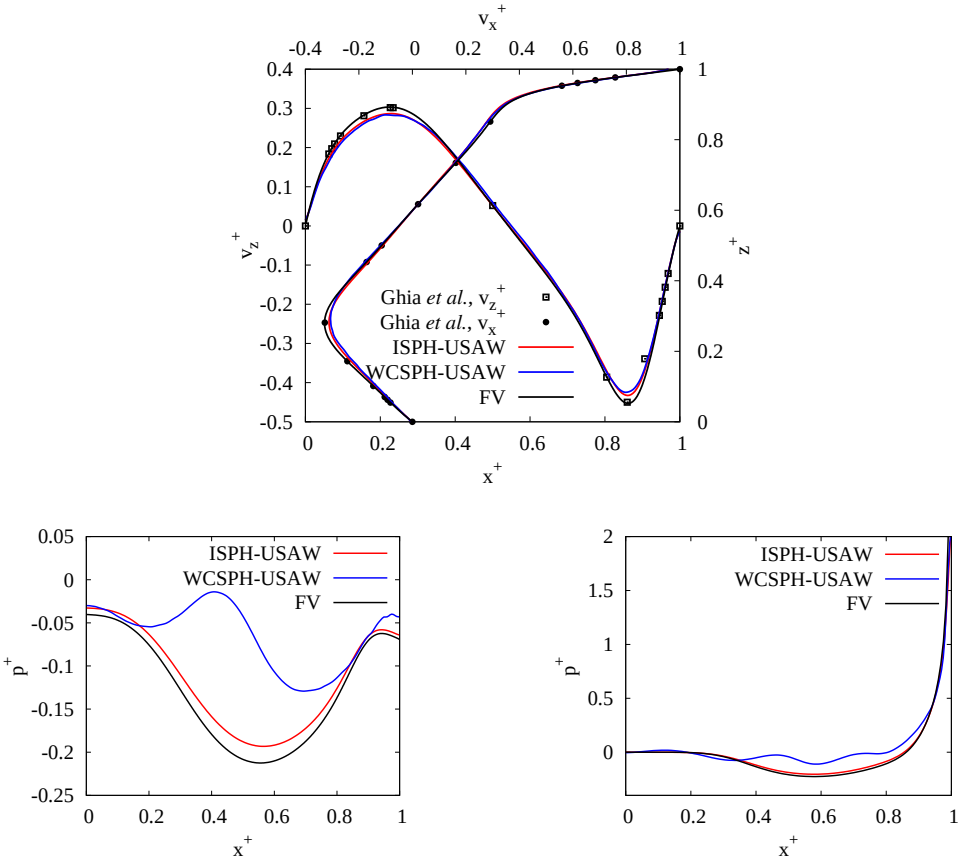


Figure 4.3: Lid-driven cavity for $Re = 400$. Dimensionless velocity profiles (top), pressure profiles in $z^+ = 1/2$ (bottom-left) and pressure profiles on the diagonal (bottom-right). Comparison between FV, WCSPH-USAW and ISPH-USAW. Velocity results are also compared to Ghia *et al.*'s results [38].

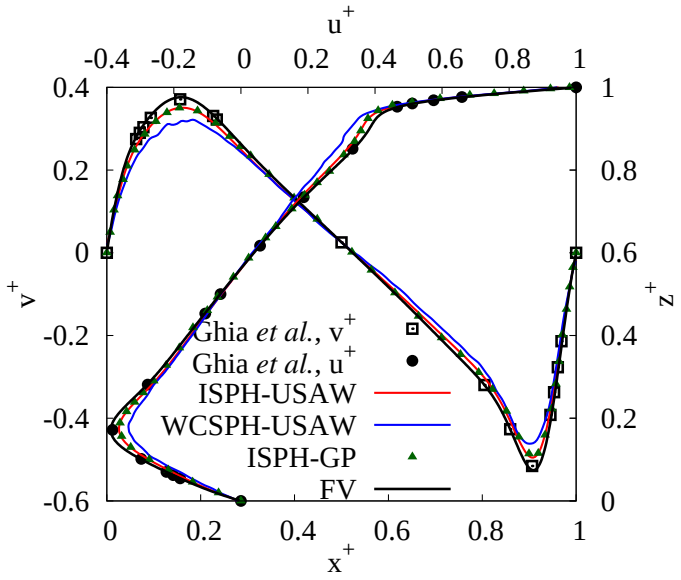


Figure 4.4: Lid-driven cavity for $Re = 1000$. Comparison of the dimensionless velocity profiles in $x^+ = 1/2$ and $z^+ = 1/2$ with ISPH-USAW, ISPH-GP [155], WCSPH-USAW, FV and the results of Ghia *et al.* [38].

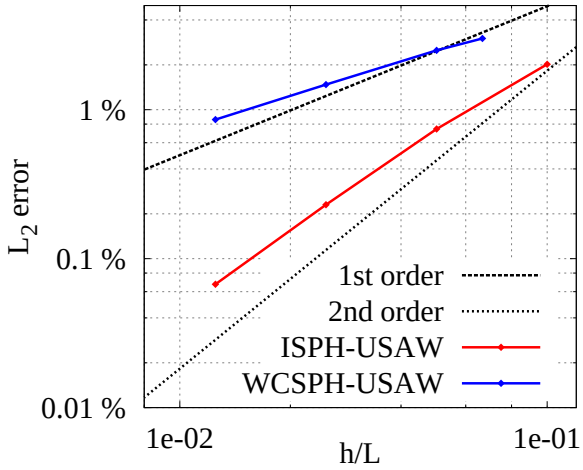


Figure 4.5: Lid-driven cavity for $Re = 1000$. Convergence studies with ISPH-USAW and WCSPH-USAW.

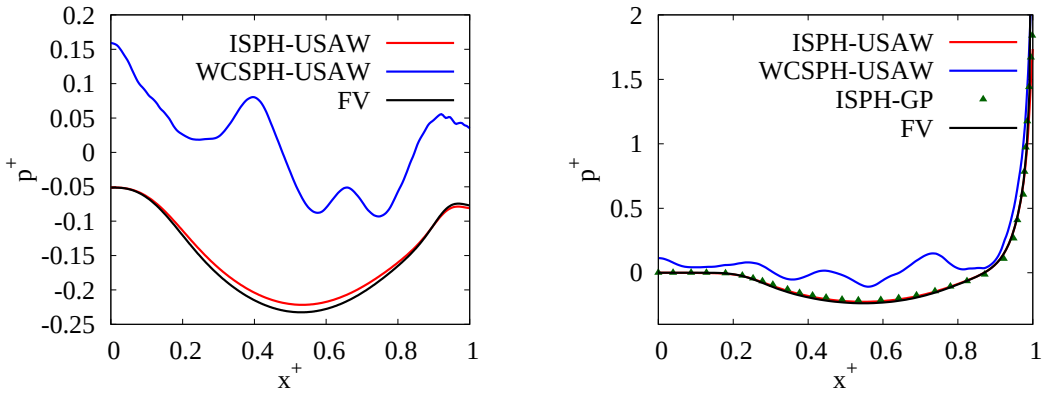


Figure 4.6: Lid-driven cavity for $Re = 1000$. On the left: dimensionless pressure profiles in $z^+ = 1/2$. On the right: dimensionless pressure profiles on the diagonal. Comparison between ISPH-USAW, ISPH-GP [155], WCSPH-USAW and FV.

The simulations at $Re = 1000$ were also run on the GPU code with a discretisation similar to that of the FV simulation (500×500 particles). The Figure 4.7 shows the results obtained with ISPH-USAW and WCSPH-USAW compared to FV and to Ghia *et al.*'s results [38] regarding the velocity field. The results quality is clearly higher with ISPH-USAW than with WCSPH-USAW.

4.2.1.2 Infinite array of cylinders in a channel

The second confined laminar flow considered in this work consists of a very viscous flow around an infinite array of cylinders confined in a channel. This case was chosen in order to check that ISPH-USAW can accurately predict hydrodynamic forces on walls. The problem considered in this work is the same as in [78] and [30]. Figure 4.8 shows a sketch of the geometry. All the distances are made dimensionless by the radius of the cylinders L. Its dimensionless height H^+ is set to 4 and a cylinder is placed at its half-height: $z^+ = 2$. Periodic boundary conditions are applied along the x-direction so that an infinite array of cylinders is represented. The inter-cylinder distance is set through the length of the channel. Various inter-cylinder dimensionless distances l^+ were considered, ranging from $l^+ = 2.5$ up to $l^+ = 35$. The fluid considered presents a dynamic viscosity $\nu = 10^{-4} m^2 s^{-1}$. The value of the average flow velocity in the unobstructed channel is imposed as $U = 1.2 \times 10^{-4} m s^{-1}$, which produces a Reynolds number $Re = 2.4 \times 10^{-2}$. A body force $\boldsymbol{F} = F\boldsymbol{e}_x$ is dynamically applied to the fluid in order to obtain the desired value of U and the simulations are run until a steady-state is reached. The formula used to compute the longitudinal body force is the one proposed in [94] (here we drop the particle labels):

$$F^n = F^{n-1} + \frac{U - 2\tilde{v}^{n-1} + \tilde{v}^{n-2}}{\delta t} \tag{4.7}$$

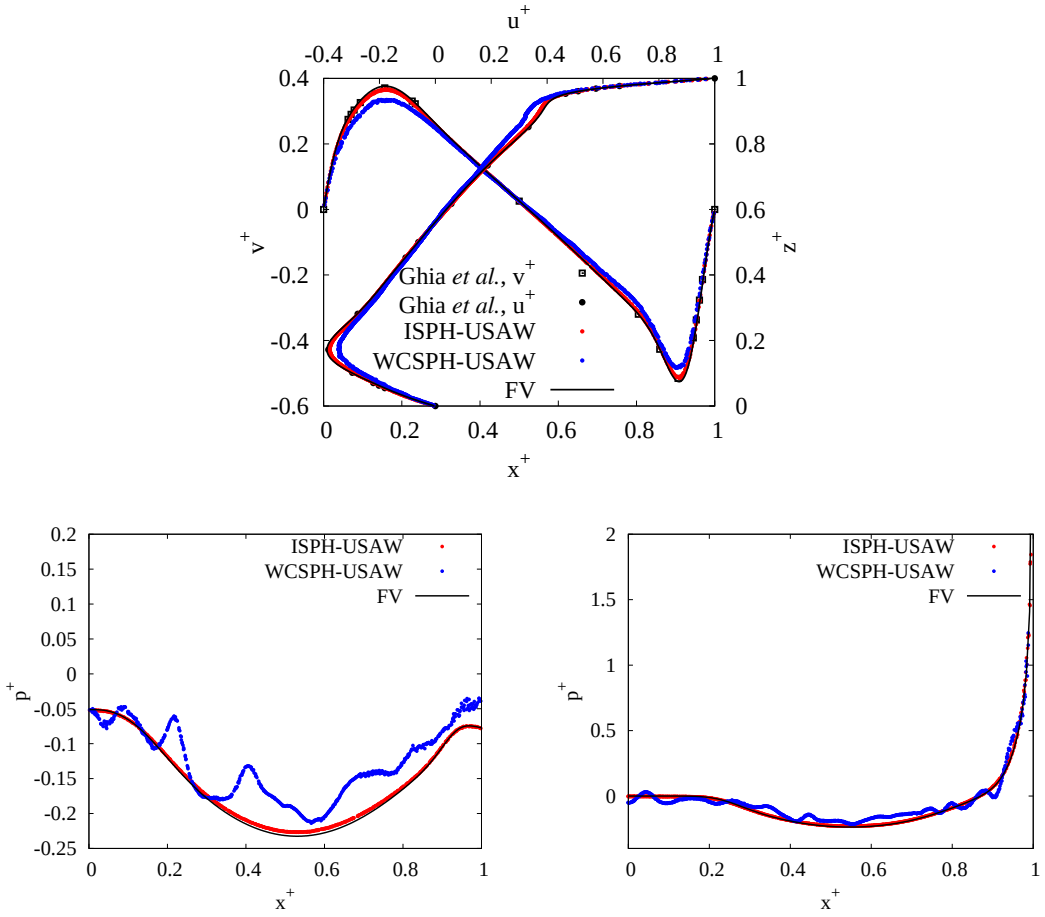


Figure 4.7: Lid-driven cavity for $Re = 1000$. Dimensionless velocity profiles (top), pressure profiles in $z^+ = 1/2$ (bottom-left) and pressure profiles on the diagonal (bottom-right). Comparison between FV, WCSPH-USAW and ISPH-USAW. The discretisation used in the SPH simulations is 500×500 particles. Velocity results are also compared to Ghia *et al.*'s results [38].

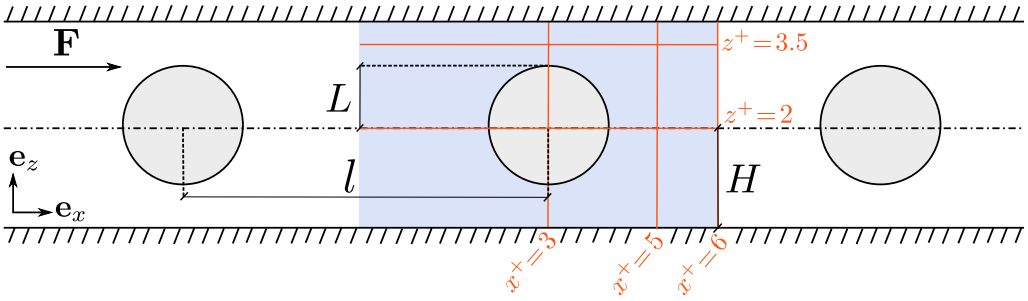


Figure 4.8: Infinite array of cylinders in a channel: sketch of the geometry with $l^+ = 6$ [30, 78]. The orange lines correspond to the profiles plotted in Figure 4.10 ($x^+ = 3, 5, 6$ and $z^+ = 2, 3.5$).

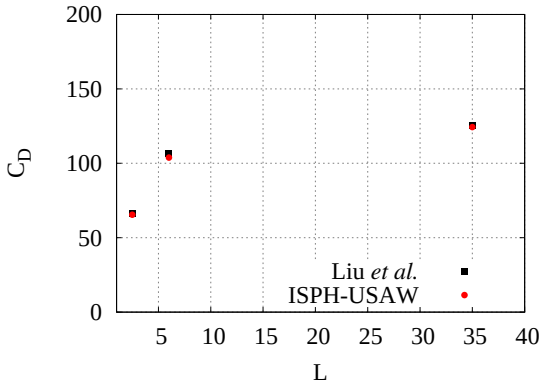


Figure 4.9: Infinite array of cylinders in a channel: dimensionless drag force as a function of the inter-cylinder distance. Comparison between ISPH-USAW and the results obtained by Liu *et al.* [78].

where $\tilde{v}^n$ is the average longitudinal flow velocity in the unobstructed channel at time n, computed as:

$$\tilde{v}^n = \frac{1}{N_c^n} \sum_{a \in \mathcal{F} \cup \Omega_c} v_x^n \tag{4.8}$$

where Ω_c is a slice of the channel located at $x^+ = l^+$ of width equal to the initial interparticular spacing δr, and N_c is the number of fluid particles located in this slice at time n.

The total drag force per unit length acting on the cylinder, F_D, was computed for several values of l^+. This force is oriented along the x-direction and was computed as:

$$F_D = \sum_{s \in \Gamma} \left(-p_s \mathbf{n}_s + \mu \left[\boldsymbol{\nabla} \boldsymbol{u}_s + \boldsymbol{\nabla} \boldsymbol{u}_s^T \right] \right) \cdot \mathbf{e}_x S_s \tag{4.9}$$

where Γ is the boundary of the cylinder, S_s is the length of the segment s and the gradient of velocity at the segments was computed as:

$$\boldsymbol{\nabla} \boldsymbol{u}_s \doteq \frac{1}{2} \sum_{i=1,2} \boldsymbol{G}_{v_i}^{\gamma,-} \{\boldsymbol{u}_b\} \tag{4.10}$$

where the v_i are the vertices linked together by segment s and $\boldsymbol{G}_a^{\gamma,-}$ is defined by (2.144). For the following comparisons, the dimensionless drag coefficient is defined as $C_D = \frac{F_D \rho}{\nu U}$ [30]. Figure 4.9 shows the values of C_D obtained with ISPH-USAW compared with the results of Liu *et al.* [78] for several lengths of the channel. Their results were obtained with a Finite Elements Method (FEM). The agreement is good for the three values of L considered.

Let us now focus on the case where $l^+ = 6$. A comparison of velocity profiles at $x^+ = 3$, $x^+ = 5$, $x^+ = 6$, $z^+ = 2$ and $z^+ = 3.5$ was done with results obtained by Ellero *et al.* [30] where they used the Immersed Boundary Method (IBM) [97, 98] and with WCSPH using mirror particles to model boundaries (WCSPH-MP). For the SPH simulations, a discretisation of 120 particles along

the height of the channel was used. We observe that the ISPH-USAW velocity profiles match quite well the ones obtained with IBM (see Figure 4.10). Ellero *et al.* obtained slightly better velocity profiles with WCSPH-MP, which can be explained by the fact that they used a ratio $h/\delta r = 4.5$, whereas we took it equal to 2. With $l^+ = 6$, Liu *et al.* obtained $C_D = 106.77$ using periodic boundary conditions along the x-direction. This value was taken as a reference and the relative error compared to the SPH results was calculated for several discretisations[1], using a fixed ratio $h/\delta r = 2$. The results of this convergence study are presented on the right-hand side of Figure 4.11, where WCSPH-USAW and ISPH-USAW are compared. With ISPH-USAW, an order of convergence of 1.39 ± 0.03 was obtained, while with WCSPH-USAW it was only of 0.94 ± 0.04. Note that Ellero *et al.* obtained an order of convergence of about 0.94 with WCSPH-MP. Though, in their simulations C_D converged towards a higher value than the one obtained by Liu *et al.*, as can be seen on the left side of Figure 4.11. They attributed this to the fact that the discretisation error becomes predominant for lower resolutions but it does not seem to be a relevant explanation since we did not observe this phenomenon in our simulations. Nevertheless, our results show that the pressure prediction is more accurate with ISPH-USAW than with WCSPH-USAW.
Note that for this test-case the numerical stability is conditioned by the viscous force, so that the time-step is the same with WCSPH and ISPH. Thus, computational times are higher with the latter. They are presented in Table 4.1. To reduce computational times at low Reynolds numbers with ISPH a solution would be to treat the viscous term implicitly, as was presented in [141] for example.

4.2.1.3 Laminar plane Poiseuille flow with inlet/outlet

In order to test the performance of the open boundaries algorithm, a laminar plane Poiseuille flow was modelled. The half-height of the channel is the characteristic length of the flow, L, and inflow/outflow conditions are applied at the extremities of the channel. The width of the channel is equal to $\frac{L}{4}$. The maximum velocity of the flow is used as reference velocity U. It is imposed through the prescription of the theoretical velocity profile at the inlet:

$$\boldsymbol{v}^+ = \left[1 - (z^+ - z_0^+)^2\right] \boldsymbol{e}_x \tag{4.11}$$

where z_0^+ is the dimensionless vertical coordinate of the channel centre. The Reynolds number is set to 10. At the outlet, the pressure is imposed equal to zero. The simulation is run during $70s$ of physical time, which corresponds to about 2×10^5 iterations with an initial dimensionless interparticular space of $\delta r^+ = 10^{-2}$. The results obtained with ISPH-USAW are presented in the Figure 4.12, where the horizontal dimensionless velocity v_x^+ is plotted as a function of z^+ for all the particles. The agreement with the theoretical parabolic profile is excellent, which shows the inflow/outflow conditions are properly imposed on this case.

[1] $\delta r^+ = (0.25, 0.2, 0.17, 0.11, 0.083, 0.07, 0.06, 0.048, 0.042, 0.036)$

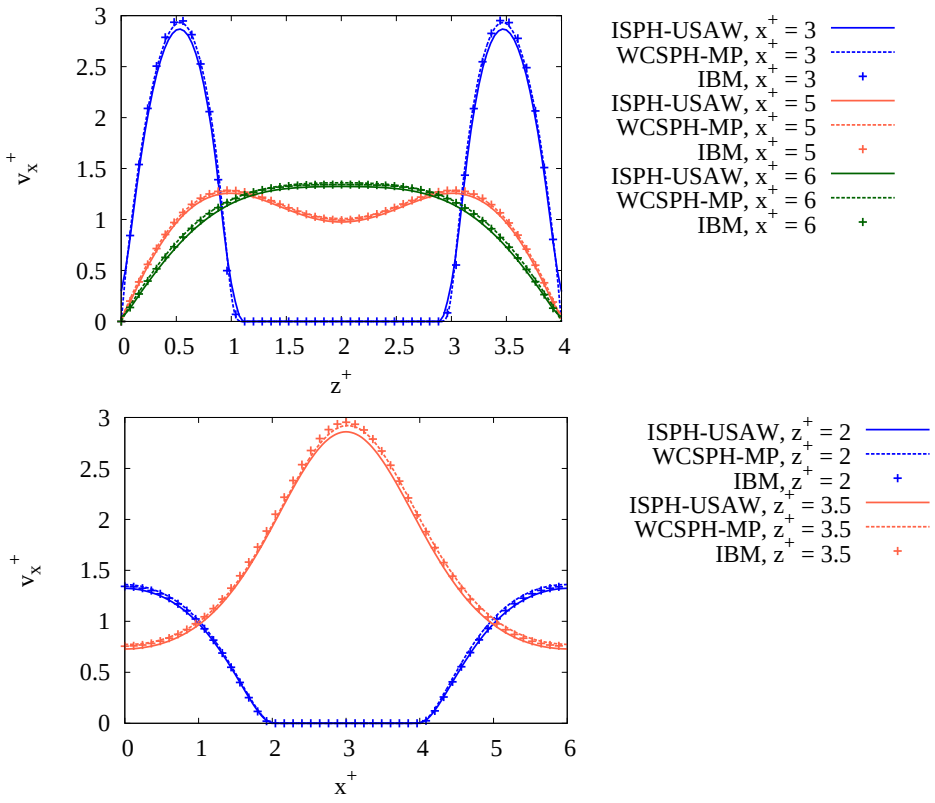


Figure 4.10: Infinite array of cylinders in a channel: velocity profiles for the case $l^+ = 6$. Comparison between ISPH-USAW, WCSPH-MP and IBM [30].

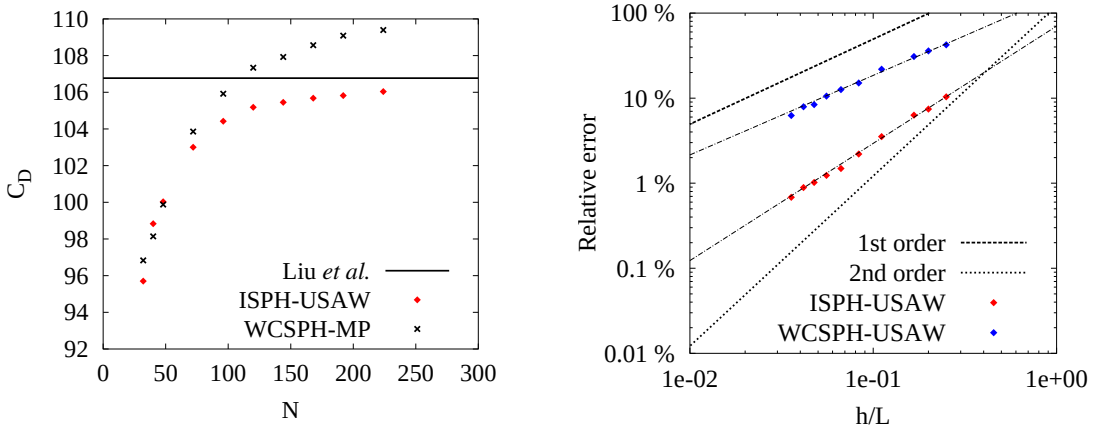


Figure 4.11: Infinite array of cylinders in a channel ($l^+ = 6$). On the left: evolution of the drag coefficient C_D as a function of the discretisation. On the right: relative error in C_D as a function of the discretisation, the results of Liu *et al.* [78] being chosen as a reference.

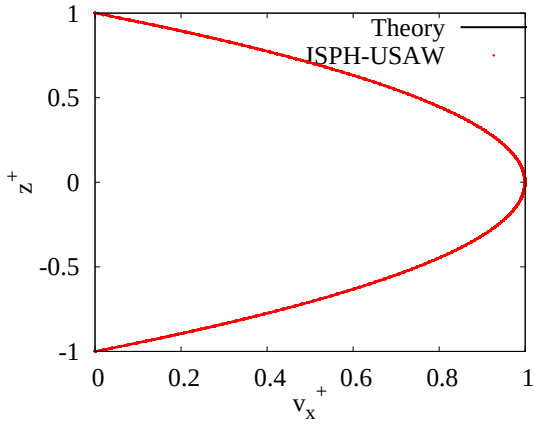


Figure 4.12: Laminar plane Poiseuille channel flow with inflow/outflow conditions. Comparison of the dimensionless velocity profile obtained with ISPH-USAW with $\delta r^+ = 10^{-2}$ and the theoretical parabolic profile. The two curves fit almost perfectly.

4.2.1.4 Dam-break over a wedge

This case was simulated in order to check that our new ISPH-USAW model can accurately represent violent free-surface flows. It consists of a schematic 2-D dam-break in a 2 meters long and 1 meter high pool, presenting a triangular wedge in the bottom. The geometry is the same as in [35]. The initial interparticular spacing for the simulations with ISPH and WCSPH was taken equal to $10^{-2}m$ and the kinematic viscosity to $10^{-2}m^2s^{-1}$. In the case of the WCSPH method, a Ferrari density correction was used (2.99) and the numerical speed of sound was taken equal to $20ms^{-1}$. The results obtained with ISPH and WCSPH with 5.881×10^3 particles were compared to the ones obtained with VoF, with 6.322×10^3 cells. Although in VoF the simulations were done for a two-phase (air + water) model, which limits the extent of the comparison with the single-phase SPH models, this comparison is useful to check the accuracy of our method. The results obtained with VoF were considered as a reference against which the ones obtained with SPH were compared. Figure 4.13 shows a qualitative comparison of the results obtained with VoF and ISPH-USAW. The dimensionless time t^+ was defined as in equation (4.5) with H is the initial fluid depth ($H = 1m$). The two methods give similar results. Differences appear between the models that can be due to the two-phase nature of VoF, while the SPH models are single-phase. Moreover, in the visualisation of VoF results, the free surface is considered as the locations where the volume fraction is 0.5, which can explain some of the differences appearing in Figure 4.13 at early times. Important differences of behaviour appear from the moment when the jet impacts the wall, which has the effect to capture air inside the fluid in the two-phase VoF simulation, which does not happen with SPH. In Figure 4.13, one can observe that a consequent number of particles remain stuck to the walls during the SPH simulation. For example, this can be seen quite well at time $t^+ = 3.13$. This is due to the high viscosity of the fluid considered here. Furthermore, particle clumping is observed at the free-surface, which is well visible on the jet. This is due to the switch off for the diffusion

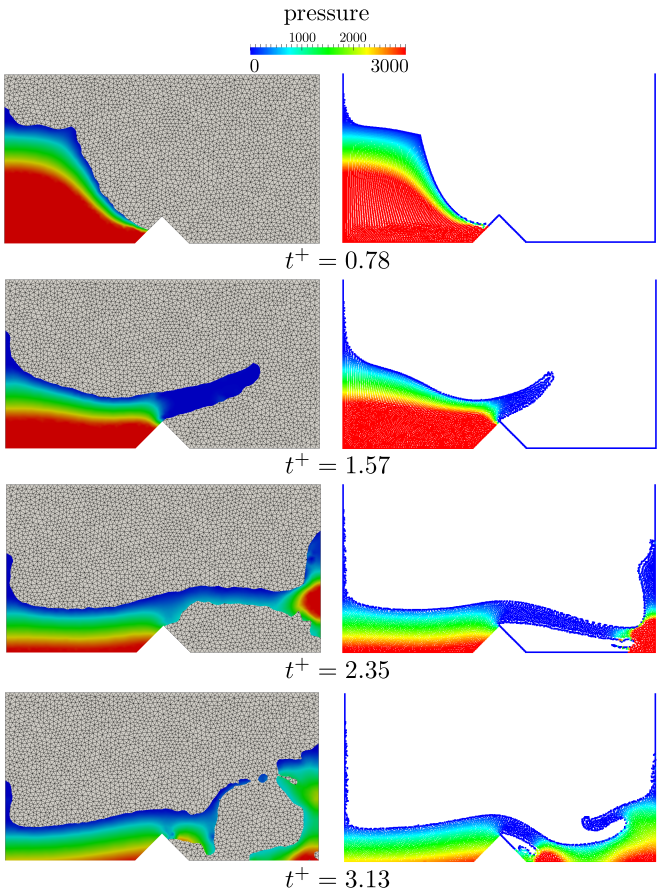


Figure 4.13: Dam-break over a wedge. Comparison of the free-surface shapes and pressure fields obtained with VoF (left) and ISPH-USAW (right) at different times.

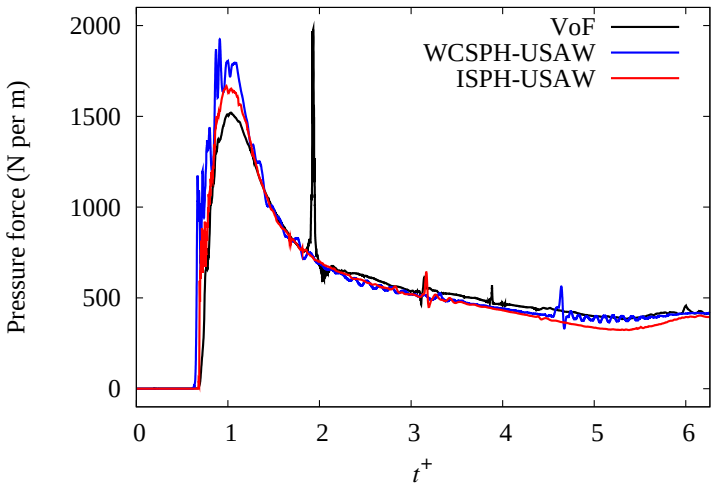


Figure 4.14: Dam-break over a wedge. Comparison of the evolution of the pressure force applied on the left-side of the wedge between VoF (6322 cells), ISPH-USAW (5881 particles) and WCSPH-USAW (5881 particles).

shift close to the free-surface as mentioned in section 3.2.2. In order to quantitatively compare the different methods, the evolution of the pressure force applied on the left side of the wedge during the simulation is plotted, as in [35]. This normal force F was computed by integrating the pressure on the left side of the wedge, Γ, according to:

$$F = \sum_{s \in \mathcal{S} \cup \Gamma} p_s S_s \tag{4.12}$$

where S_s is the surface of the segment s. In this case all the surfaces of the segments are equal to δr. The results obtained with ISPH-USAW, WCSPH-USAW and VoF are compared in Figure 4.14. The sharp peaks that appear on the VoF curve correspond to the collapse of trapped air bubbles, which hampers the convergence of the linear solver. The three methods give similar results. However, the evolution of the value of the force is smoother with ISPH-USAW than with WCSPH-USAW. Besides, the prediction of the maximum value of the force is closer to the one obtained by VoF with ISPH-USAW than with WCSPH-USAW. When the pressure maximum occurs, the effect of air is likely to be small, so that ISPH probably predicts that maximum better than WCSPH.

On the other hand, simulations of this test case showed that the impermeability of the walls is granted by the ISPH-USAW model even in the presence of strong impact of the water on a solid wall. For the latter, the computational time was smaller than for WCSPH-USAW, as shown in Table 4.1. VoF presented higher computational time than the two SPH models, which also happened in the next test case (Section 4.2.1.5).

4.2.1.5 Water wheel

A water wheel case is now considered in order to show that the new ISPH-USAW model is able to represent flows where complex free-surface shapes and complex moving wall boundaries are involved. Figure 4.15 shows the problem geometry. The wheel radius L is the characteristic length. The wheel turns counterclockwise at $\pi/2\ rad.s^{-1}$, driving the fluid. The viscosity was set to $10^{-2}m^2s^{-1}$. Thus, the Reynolds number is about 300 and it is possible to assume that the flow is laminar. The latter is periodic along x, presents a free-surface and a horizontal bottom along

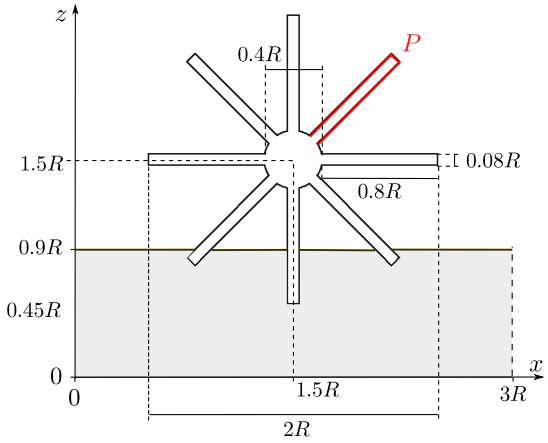


Figure 4.15: Water wheel test-case: sketch of the geometry.

$z^+ = 0$. The dimensionless time t^+ was defined as in equation (4.5) with H the initial water height ($H = 0.9m$). As for the dam-break case, the results obtained with ISPH-USAW are compared to the VoF two-phase model. A comparison with WCSPH-USAW is also presented. The free-surface shapes and velocity fields obtained at $t^+ = 66$ with the ISPH-USAW and the VoF method are depicted in Figure 4.16. The simulation counted 8×10^4 cells with VoF and 3×10^4 particles with ISPH-USAW. The Figure shows strong wetting of the wheel-arms in the VoF simulation whereas for the ISPH-USAW simulation the arms out of the water are dry except for very few individual water particles. This discrepancy is due to the post-treatment with OpenFOAM: as in Section 4.2.1.4 the free-surface is considered as the locations where the volume fraction is 0.5, which gives the impression that there is water on the paddles. This is a drawback of the VoF method where the free-surface is fuzzy. A quantitative comparison was done by comparing the time evolution of the pressure force applied on the bucket P (in red in Figure 4.15) obtained with the three methods. The results are presented in Figure 4.17, where we present smoothed results for the sake of readability, since they were very noisy with the three methods. With ISPH-USAW and VoF this is explained by the fact that it is hard for the pressure solver to converge. With VoF this is due to the rotating mesh, while with ISPH-USAW it is due to the few particles wetting the wheel arms when they are above the free-surface. Although some differences appear due to the fact that we are comparing a single-phase model with a two-phase one, ISPH-USAW and VoF results are in reasonable agreement. On the other hand, with WCSPH-USAW the pressure peaks present much greater amplitudes. The

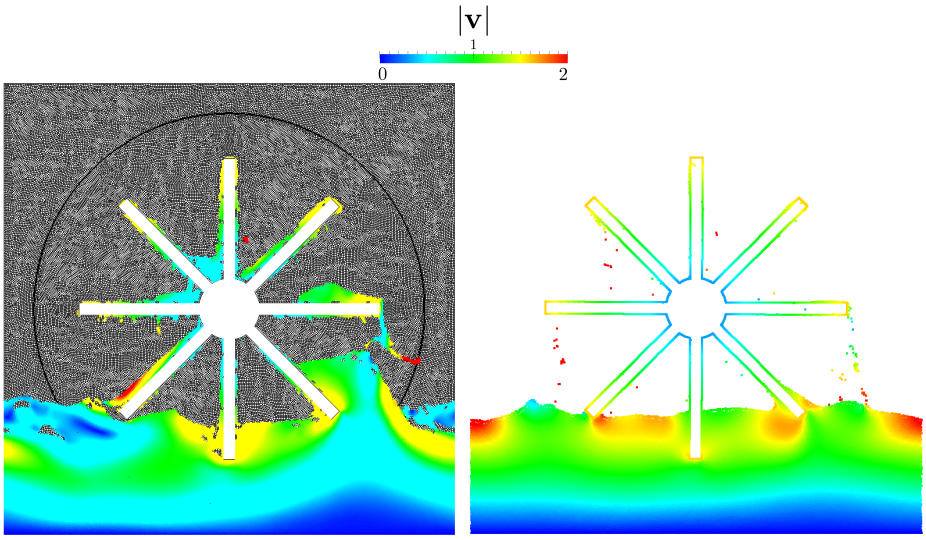


Figure 4.16: Water wheel test-case. Comparison of the free-surface shapes and velocity fields between VoF (left) and ISPH-USAW (right) at $t^+ = 66$.

amplitude of the pressure force peaks is slightly higher with ISPH-USAW than with VoF because of the presence of air trapped between the wheel and the fluid. The air pockets provide an additional pressure on the wheel, but they also reduce the water level beneath it, which in the end reduces the force due to water on the paddle. In spite of this, the results obtained with ISPH-USAW are quite satisfactory and show that the new model is robust and accurate, even with complex walls. Besides, the computational time was lower with ISPH-USAW than with WCSPH-USAW and VoF performed slower than the two SPH models, as shown in Table 4.1 (all codes running on one CPU). The very high computational time exhibited by VoF on this case is due to the difficulty the pressure solver had to converge.

4.2.1.6 Schematic dam-break with an outflow condition

This case consists of a schematic dam-break on a flat bottom, which was cut so as to test the outlet formulation, and check that the fluid leaves the domain without reflections. The outlet boundary is the left-wall (at each time) in Figure 4.18. The height of the fluid column at the initial time is $H = 1m$. The viscosity of the fluid was set to $10^{-2}m^2s^{-1}$. A zero-pressure is imposed at the outlet. The initial interparticular space was taken equal to $\delta r = 6 \times 10^{-3}m$. Figure 4.18 shows the velocity field shape at several dimensionless times, the latter being defined as in (4.5). The fluid correctly leaves the domain without reflections at the outlet. The free-surface shape of the same non-cut dam-break simulated with ISPH-USAW is provided and appears in black in the Figure. The agreement is quite good between the two simulations. Differences appear after some time, which is expected since imposed pressure a the outlet differs from the pressure in the non-cut simulation.

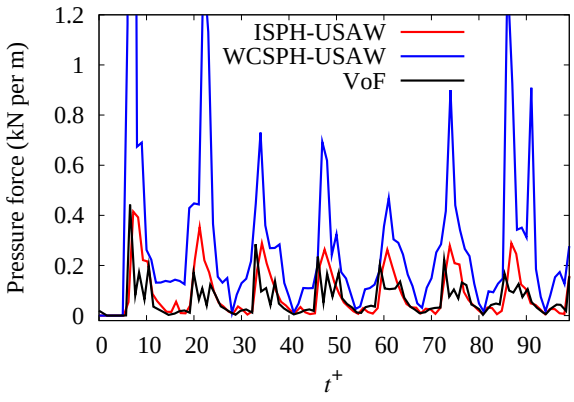


Figure 4.17: Water wheel test-case. Evolution of the smoothed pressure force magnitude applied on the bucket P. Comparison between VoF, WCSPH-USAW and ISPH-USAW.

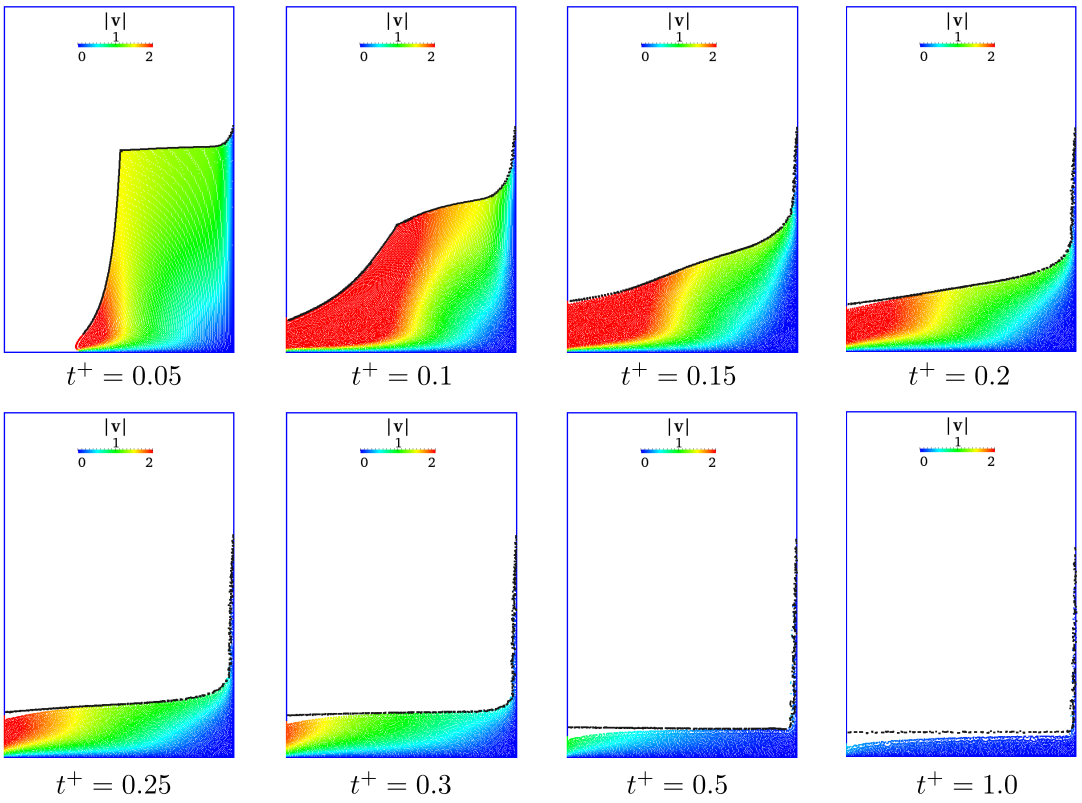


Figure 4.18: Dam-break with an outflow condition: velocity field shape and comparison with the free-surface shape of a non-cut dam-break (black dots).

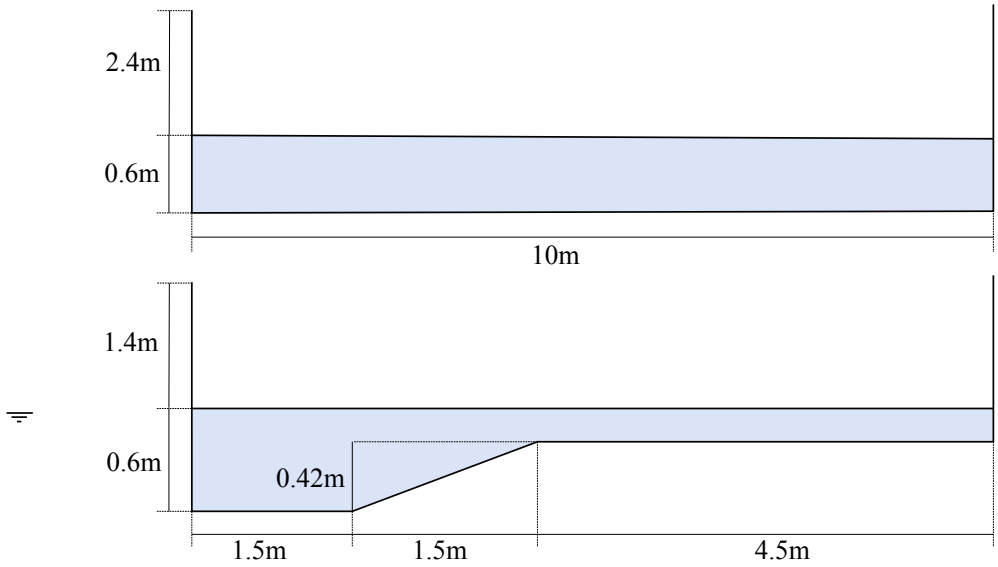


Figure 4.19: Sketch of the geometries of the two cases of solitary wave propagation.

4.2.1.7 2-D solitary waves

Two cases of solitary waves are presented here, one on a flat bottom and the other on a slope where the wave breaks before leaving the domain. Figure 4.19 shows the geometry of the two cases. The incoming free-surface elevation is prescribed as a solitary wave (solution to the Korteveg-De Vries equation) [24]:

$$\eta(x,t) = A\text{sech}^2[k(x - Ct - x_0)] \tag{4.13}$$

where η is the free-surface elevation compared to a reference water level H, A is the wave amplitude, $k = \sqrt{\frac{3A}{4H^3}}$ is the wave number and $C = \sqrt{g(A+H)}$ is the wave celerity. In both cases the wave amplitude is $A = \frac{H}{2}$. x_0 is the initial position of the wave, equal to $x_0 = x_{inlet} - \frac{4}{k}$ here. At the inlet (left boundary in Figure 4.19), the water height H_t is used to impose a linearised velocity profile:

$$\begin{cases} H_t(t) = H + \eta(x_{inlet}, t) \\ u_x(z,t) = C\dfrac{\eta(x_{inlet},t)}{h(t)} \\ u_z(z,t) = \dfrac{z}{H_t(t)}\dfrac{\partial \eta}{\partial t}(x_{inlet},t) \end{cases} \tag{4.14}$$

with x_{inlet} the horizontal coordinate of the inlet. At the outlet (right boundary in Figure 4.19), the pressure is imposed through the Orlanski radiative boundary condition (equation (3.81)).

Figures 4.20 and 4.21 shows the propagation of the solitary wave on a flat bottom with a dynamic molecular viscosity of $10^{-2}m^2s^{-1}$ and $10^{-6}m^2s^{-1}$ respectively (for the latter no turbulence model was used). The dimensionless time t^+ is defined as in (4.5). The colours correspond to the magnitude of the velocity field obtained with ISPH-USAW and the black lines to the analytical solution

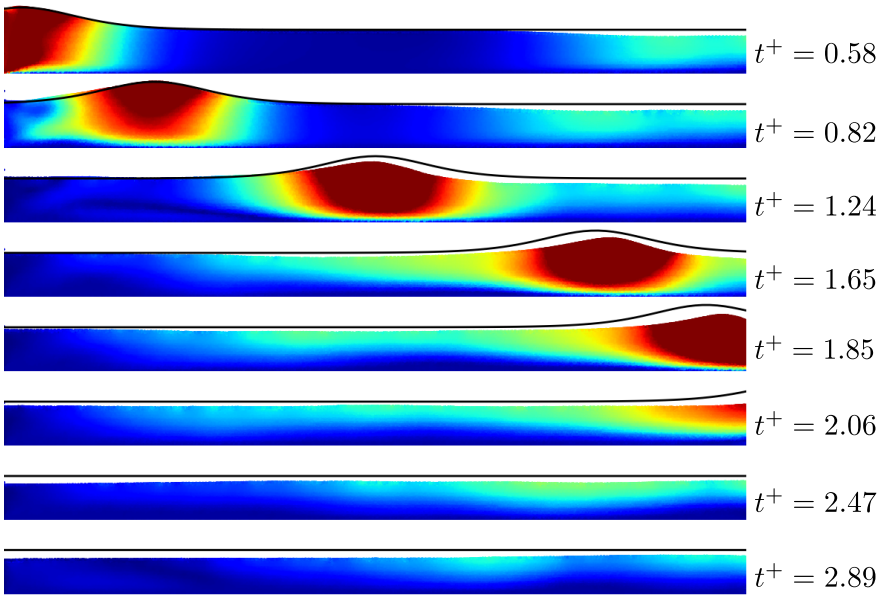


Figure 4.20: Propagation of a solitary wave on a flat bottom with $\nu = 10^{-2}m^2s^{-1}$: the colours correspond to the velocity magnitude obtained with ISPH-USAW and the black lines to the analytical solution of equation (4.13). This simulation was run with 63206 particles ($\delta r = 0.01m$).

of equation (4.13). These simulations were run with 63206 particles ($\delta r = 0.01m$). The wave enters correctly the domain and goes out smoothly. The agreement with the analytical solution is satisfactory, considering that the latter does not account for viscous effects and is not even a solution of Euler's equations. However, the water level drops slightly, initiating from the outlet, which shows that the outflow condition on the pressure still needs to be improved. Besides, some discrepancies in the velocity field appear in the low-viscosity simulation along the bottom wall (this is especially visible at $t^+ = 2.89$).

Figures 4.22 and 4.23 show the propagation of the solitary wave on a slope with a dynamic molecular viscosity of $10^{-2}m^2s^{-1}$ and $10^{-6}m^2s^{-1}$ respectively (for the latter no turbulence model was used). The colours correspond to the pressure field obtained with ISPH-USAW. These simulations were run with 30315 particles ($\delta r = 0.01m$). This time the water level decrease is less visible (although it still happens), and the breaking wave leaves the domain apparently without reflections.

4.2.2 Turbulent flows

Two validation cases were performed to assess the performance of the $k - \epsilon$ model in the SPH incompressible formalism. Let us recall that since we use a model based on the RANS formalism, only the mean quantities of the flows are modelled, which proves sufficient in many industrial studies. A more accurate model would need, *e.g.* LES, but this is not the purpose of the present work.

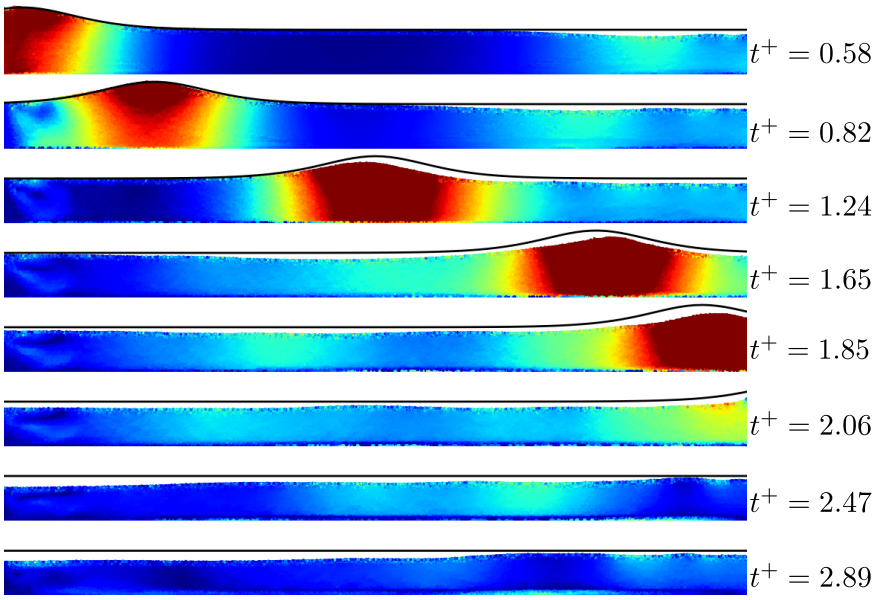


Figure 4.21: Propagation of a solitary wave on a flat bottom with $\nu = 10^{-6} m^2 s^{-1}$: the colours correspond to the velocity magnitude obtained with ISPH-USAW and the black lines to the analytical solution of equation (4.13). This simulation was run with 63206 particles ($\delta r = 0.01m$).

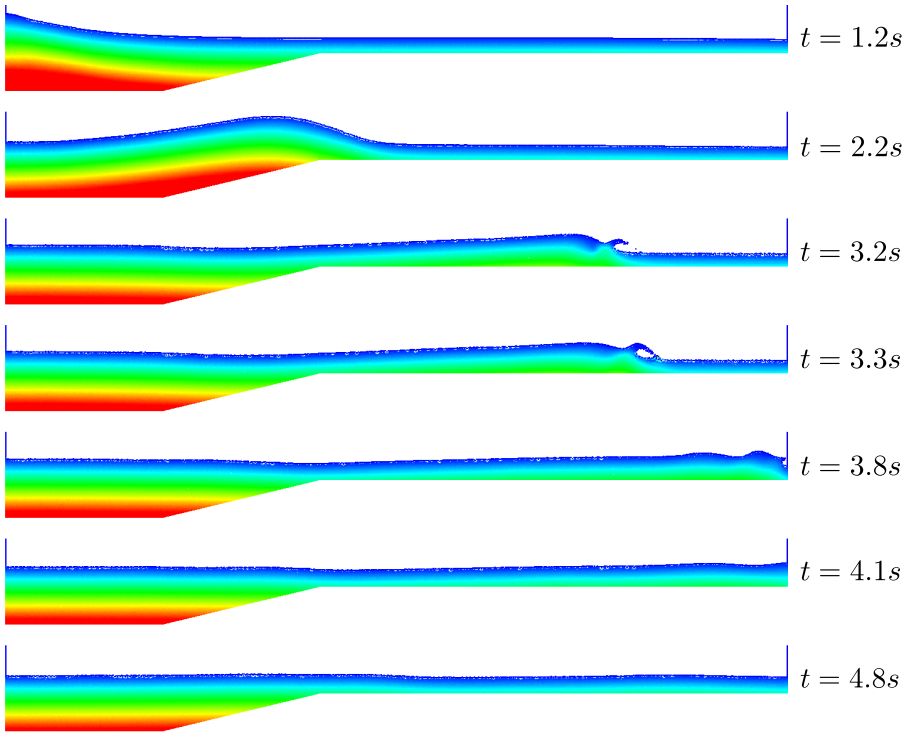


Figure 4.22: Propagation of a solitary wave on a slope: pressure field obtained with ISPH-USAW with 30315 particles ($\delta r = 0.01m$) and $\nu = 10^{-2} m^2 s^{-1}$.

Table 4.1: Computational times of the various models on several test-cases. The calculations were performed on 1 CPU (Intel® Xeon® Processor E5504, 4M Cache, 2.00 GHz, 4.80 GT/s Intel® QPI).

Model	**Number of cells/particles**	**Time**
Lid-driven cavity ($Re = 1000$, $60s$ of physical time)		
FV	512×512	38 h
ISPH-USAW	200×200	31 h
WCSPH-USAW	200×200	32 h
Infinite array of cylinders ($80s$ of physical time)		
ISPH-USAW	12.659×10^3	10h00
WCSPH-USAW	12.659×10^3	1h30
Dam-break over a wedge ($2s$ of physical time)		
VoF	6.322×10^3	> 1h
ISPH-USAW	5.881×10^3	20 min
WCSPH-USAW	5.881×10^3	30 min
Water wheel ($30s$ of physical time)		
VoF	$\approx 8 \times 10^4$	5 days
ISPH-USAW	$\approx 3 \times 10^4$	15 h
WCSPH-USAW	$\approx 3 \times 10^4$	18.5 h
Fish-pass ($20s$ of physical time)		
FV	$\approx 2.5 \times 10^4$	26 h
ISPH-USAW	$\approx 6 \times 10^4$	76 h
WCSPH-USAW	$\approx 6 \times 10^4$	55 h

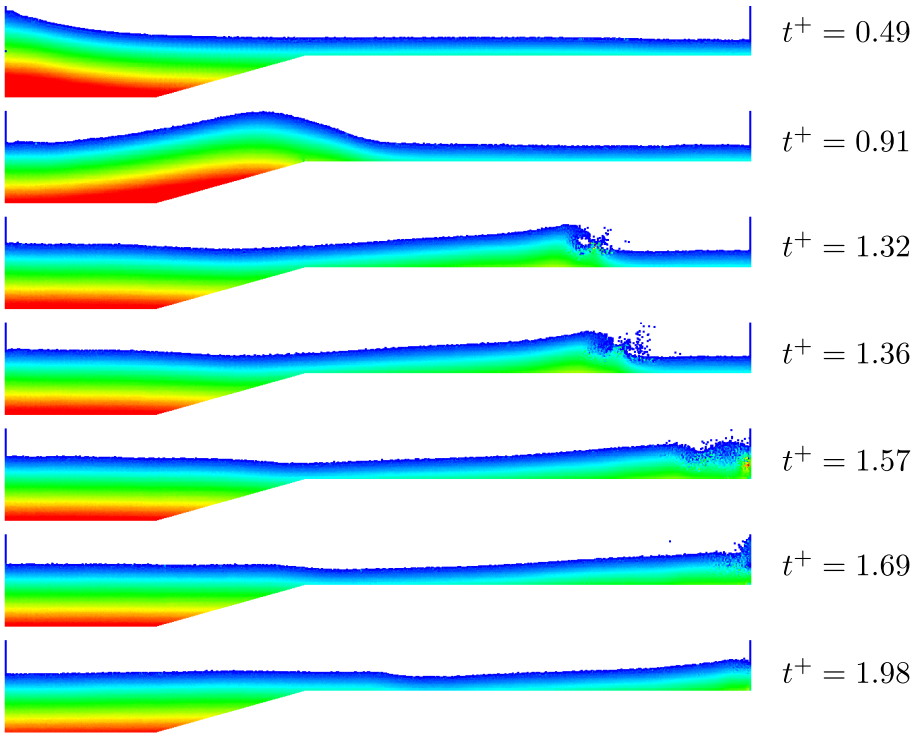


Figure 4.23: Propagation of a solitary wave on a slope: pressure field obtained with ISPH-USAW with 30315 particles ($\delta r = 0.01m$) and $\nu = 10^{-6}m^2s^{-1}$.

4.2.2.1 Turbulent channel flow

In order to test the performance of the $k-\epsilon$ model associated to ISPH, a turbulent Poiseuille channel flow was modelled. The half-height of the channel is the characteristic length of the flow, L, and periodic conditions are applied along the horizontal in the x-direction. The friction velocity u_* is set to $1ms^{-1}$ by imposing a horizontal volumetric force of constant magnitude, $F = 1.0m.s^{-2}$[2]. At the initial time, the particles are aligned along horizontal lines and they remain so during the simulation, even after $100s$ of physical time (about 60000 iterations), with either ISPH-USAW or WCSPH-USAW. The dimensionless variables are the ones of equation (4.5) with $U = u_*$. Besides, the dimensionless distance to the lower wall is defined as:

$$y^+ = \frac{yu_*}{\nu} \tag{4.15}$$

where y is the distance to the lower wall. The friction Reynolds number defined through (4.1) with $U = u_*$:

$$Re_* = \frac{u_* L}{\nu} \tag{4.16}$$

[2]The friction velocity can be calculated by writing a balance of the forces and is equal to $\sqrt{fL} = 1\ m.s^{-1}$.

The friction Reynolds number is equal to the dimensionless vertical coordinate at the centre of the channel, and was taken equal to 640, so that the molecular viscosity of the fluid was taken equal to $1.5625 \times 10^{-3} m^2.s^{-1}$. The results presented below were obtained with an initial interparticular spacing of $5 \times 10^{-2} m$.

The results obtained with ISPH-USAW are presented in Figures 4.24 and 4.25, where the profiles of dimensionless velocity, turbulent kinetic energy k^+ and dissipation rate ϵ^+ are plotted along the lower half of the channel. A comparison is presented with Direct Numerical Simulation (DNS) results obtained by Kawamura *et al.* [1, 61] and with a FV $k - \epsilon$ model. No comparison with WCSPH-USAW is presented since in this case it perfectly matches ISPH-USAW. The results ob-

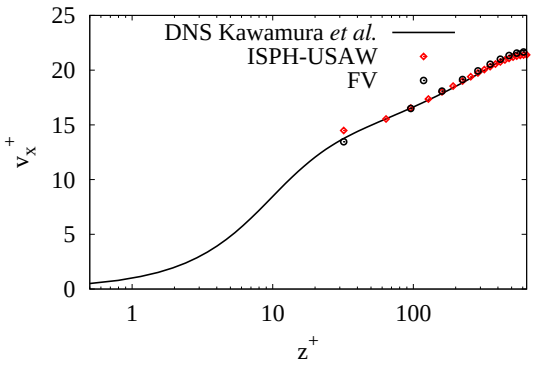


Figure 4.24: Turbulent Poiseuille channel flow at $Re_* = 640$. Comparison of the dimensionless velocity profiles obtained by ISPH-USAW, FV (both with the $k - \epsilon$ model) and DNS.

tained with ISPH-USAW match very well the FV ones and are very close to the DNS, although the velocity near the viscous sub-layer is slightly overestimated. To our knowledge, this is the first time a RANS $k - \epsilon$ model is validated with the SPH method, reaching the same accuracy as FV. It is

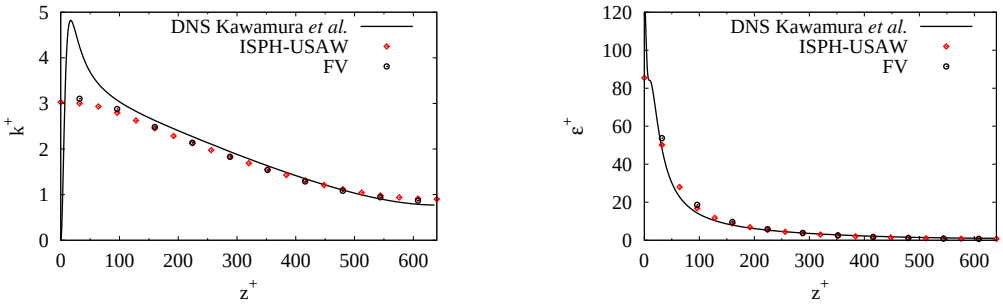


Figure 4.25: Turbulent Poiseuille channel flow at $Re_* = 640$. Comparison of the profiles of dimensionless turbulent kinetic energy (left) and dissipation rate (right) obtained by ISPH-USAW, FV (both with the $k - \epsilon$ model) and DNS.

noteworthy that the viscous sublayer is not meant to be reproduced by the $k-\epsilon$ turbulence model, which explains why the turbulent kinetic energy profile obtained with DNS is different from the ones obtained with FV and ISPH-USAW close to the wall.

4.2.2.2 Schematic fish-pass

Let us now consider another turbulent case, more complex and closer to reality: a water flow through a schematic periodic fish-pass system, which is the one considered in [35, 147]. It consists of a series of pools communicating through vertical slots. When the number of pools is high enough, the flow can be considered as periodic and it is sufficient to study one of them. Experimental results [137] showed that the mean flow is approximately parallel to the bottom of the pool, the latter being inclined of an angle $I \approx 0.1\ rad$ compared to the horizontal. Thus, the flow was modelled in two dimensions (top-viewed) and the variations along the vertical were neglected. The effect of gravity was not taken into account and the free-surface behaviour was not represented. Thus, this flow does not represent the real one, since turbulence is a three dimensional phenomenon and the free-surface cannot remain perfectly horizontal. For a complete description of the geometry of the fish-pass, see [147]. In our simulations the flow was driven by a constant body force along the x axis of magnitude $1.885\ m.s^{-2}$. The characteristic length is the size of the slot, $L = 0.3m$ and the characteristic velocity in the fluid U is close to $1m.s^{-1}$. The molecular viscosity of the fluid is $\nu = 10^{-6}m^2.s^{-1}$ so the Reynolds number is between 10^5 and 10^6. The results obtained with ISPH-USAW were compared to the ones obtains with FV and with WCSPH-USAW. In all cases the RANS equations were solved using a $k-\epsilon$ model. The SPH simulations were done with $58,823$ particles while the simulations with FV were done with 24632 cells. A qualitative comparison of the results obtained with ISPH-USAW and FV after $20s$ of physical time is presented in Figure 4.26. At that moment, the flow has converged to a nearly steady-state. A quantitative comparison of the three methods was done by comparing velocity, pressure, turbulent kinetic energy and dissipation rate profiles at sections P_1, P_2 and P_3 plotted in Figure 4.26. The four Figures 4.27, 4.28, 4.29 and 4.30 show that ISPH-USAW improves the prediction of all quantities in comparison to WCSPH-USAW, especially for pressure and near-wall velocity. Note that the results obtained with WCSPH-USAW are sensitive to the imposed value of background pressure: high values of the latter lead to inaccurate results. Its value was set equal to $5.10 \times 10^4 Pa$ for this test-case, so as to avoid the formation of voids in the flow. It was checked that velocity and pressure fields are accurately predicted at the wall when compared to FV results by plotting them along the bottom-left part of the wall (profile P_4 in Figure 4.26). The results are shown in Figure 4.31, where we see that ISPH-USAW improves a lot the distribution of wall pressures. The differences observed between the two SPH models and FV can be due to slight differences in the imposition of boundary conditions in the $k-\epsilon$ model. In this test case, WCSPH performed faster than ISPH and FV performed faster than the SPH models (see Table 4.1). In summary, the new ISPH-USAW model makes it possible to accurately represent turbulent flows presenting complex

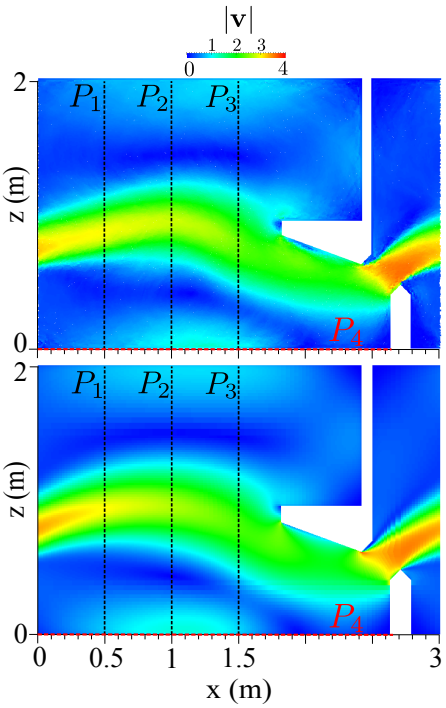


Figure 4.26: Fish-pass after $20s$. Comparison of the results obtained with ISPH-USAW (top) and FV (bottom).

wall boundaries, while such flows are very hard to model using ghost or mirror particles, due to the accuracy required regarding the imposition of a non-homogeneous Neumann boundary condition on p and ϵ.

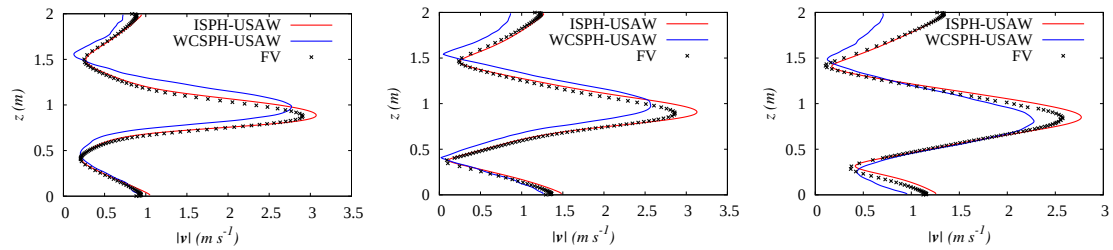


Figure 4.27: Fish-pass after $20s$. Mean velocity profiles on P_1 (left), P_2 (middle) and P_3 (right) obtained with FV, ISPH-USAW and WCSPH-USAW.

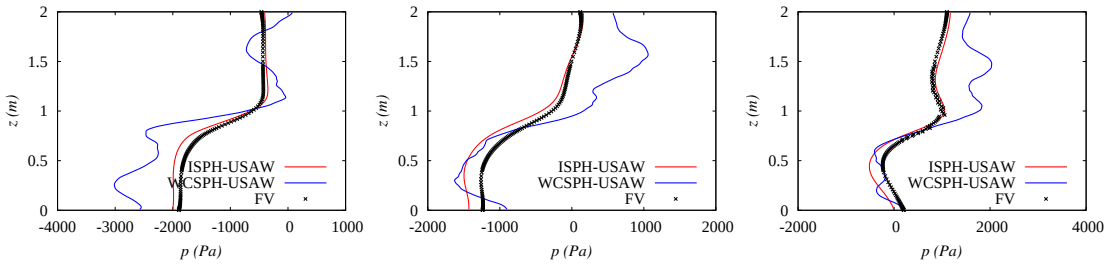


Figure 4.28: Fish-pass after $20s$. Pressure profiles on P_1 (left), P_2 (middle) and P_3 (right) obtained with FV, ISPH-USAW and WCSPH-USAW.

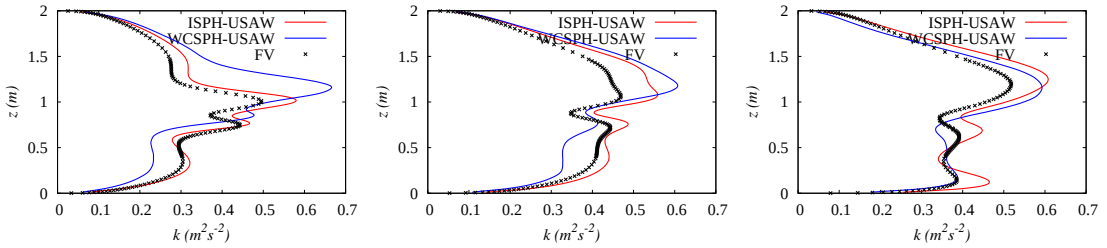


Figure 4.29: Fish-pass after $20s$. Turbulent kinetic energy profiles on P_1 (left), P_2 (middle) and P_3 (right) obtained with FV, ISPH-USAW and WCSPH-USAW.

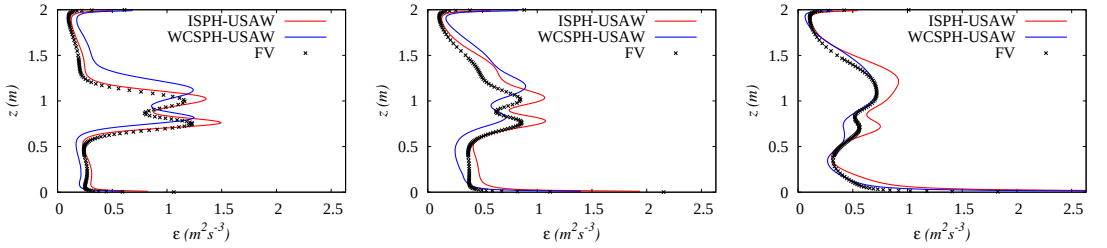


Figure 4.30: Fish-pass after $20s$. Dissipation rate profiles on P_1 (left), P_2 (middle) and P_3 (right) obtained with FV, ISPH-USAW and WCSPH-USAW.

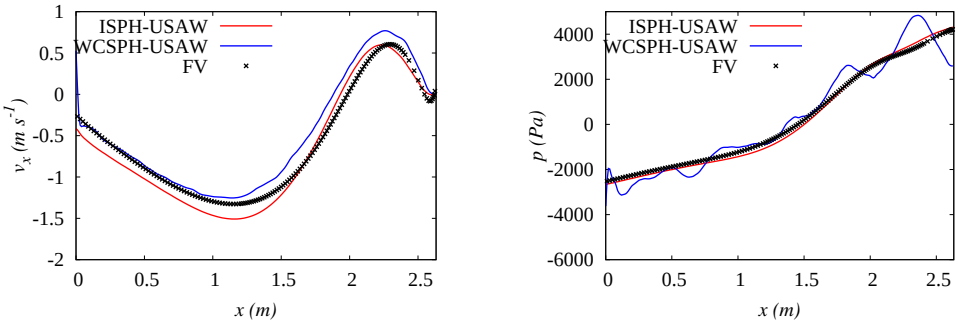


Figure 4.31: Fish-pass after $20s$. Velocity and pressure profiles on profile P_4 obtained with FV, ISPH-USAW and WCSPH-USAW.

4.3 Validation on non-isothermal 2-D cases

4.3.1 Laminar flows

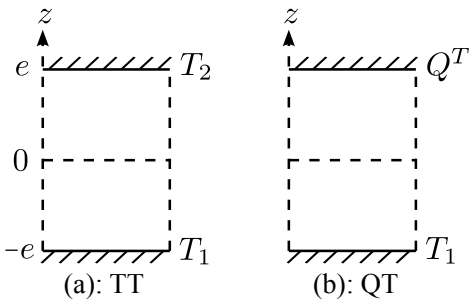
4.3.1.1 Laminar plane Poiseuille flow

In order to check that the boundary conditions on the temperature are properly imposed by the method described in section 3.3.3, two configurations of a 2D laminar plane Poiseuille flow were tested. A schematic description of their geometries is provided in Figure 4.32. The half-height of the channel is the characteristic length L. In the first case (denoted TT), constant temperatures $T_1 = T_c$ and $T_2 = T_h$ are imposed on the lower and upper walls, respectively. Recall that T_h denotes the highest temperature of the flow whereas T_c the lowest. In the second case (denoted QT), a constant heat flux Q^T is imposed through the upper wall, while the lower wall remains isothermal at the temperature T_1. The flow, with bulk velocity U, is driven by a constant volumic force. In the two cases, the Prandlt number (equation (4.2)) was set to 1 and the Reynolds number to 50. In the QT case, the bulk Nusselt number (equation (4.4)) was set to 0.5 and the Grashoff number (equation (4.3)) was set to 196, while in the TT case the Grashoff number was set to 98 and there is no bulk Nusselt number.

Note that for this case the dimensionless temperature and dynamic pressure were not defined as in (4.5), but as:

$$\begin{cases} T^+ = \dfrac{T - T_1}{\Delta T} \\ p^{*^+} = \dfrac{p^*}{\rho\beta\Delta T g L} \end{cases} \tag{4.17}$$

where $p^* = p + \rho g z$. Table 4.2 shows the theoretical expressions of the dimensionless temperature and dynamic pressure as functions of Re, Gr, Nu and of the dimensionless coordinates for the two cases. In both cases the velocity field is that of the ordinary plane Poiseuille flow. The simulations



(a): TT (b): QT

Figure 4.32: Laminar plane Poiseuille flow: sketch of the two configurations.

Table 4.2: Laminar plane Poiseuille flow: theoretical solutions for T^+ and p^{*^+}.

Case (a): TT	**Case (b): QT**
$T^+ = \frac{1}{2}\left(1 + z^+\right)$	$T^+ = -Nu\left(1 + z^+\right)$
$p^{*^+} = \frac{z^{+2}}{4} - 2\frac{Re}{Gr}x^+$	$p^{*^+} = z^+\left[\frac{1}{2} - Nu\left(1 + \frac{z^+}{2}\right)\right] - \frac{2Re}{Gr}x^+$

were done with 902 particles ($\delta r^+ = 0.05$). Figure 4.33 shows the vertical profiles of dimensionless temperature and dynamic pressure obtained with ISPH-USAW. It can be observed that an excellent agreement with the theory is obtained in both cases. In the case QT, the vertical variation of the fields does not depend on the Grashoff number. Thus, different values of the heat flux should yield the same results, which was checked with ISPH-USAW and gave similar errors between the model and the theory in all cases (with a Grashoff number up to 1960). With the chosen discretisation, the maximum relative error was of the order of 1%, on the temperature and on the pressure. These results show that the imposition of Dirichlet and Neumann boundary conditions on the temperature is properly done with our method.

4.3.1.2 Differentially heated square cavity

The second non-isothermal validation case consists of a laminar flow in a differentially heated square cavity of size L the characteristic length of the flow, that was studied in [134] with the SPH method. The left and right walls are isothermal, the right wall at temperature T_c and the left wall at temperature T_h. The upper and lower walls are adiabatic. The molecular Prandlt number is $Pr = 0.71$ and three values of the Rayleigh number $Ra = Pr \times Gr$ were tested, *i.e.* 10^3, 10^4 and 10^5. The characteristic velocity of the flow is given by $U = \frac{\nu}{L}$. A discretisation of 160×160 particles was used for the SPH simulations. Figure 4.34 shows the shape of the dimensionless temperature and velocity fields (defined through (4.5)) after convergence for $Ra = 10^5$. For the FV simulation, a discretisation of 512×512 cells was used. Figure 4.35 shows the dimensionless velocity and temperature profiles in $x^+ = 1/2$ and $z^+ = 1/2$ for $Ra = 10^5$. The SPH results

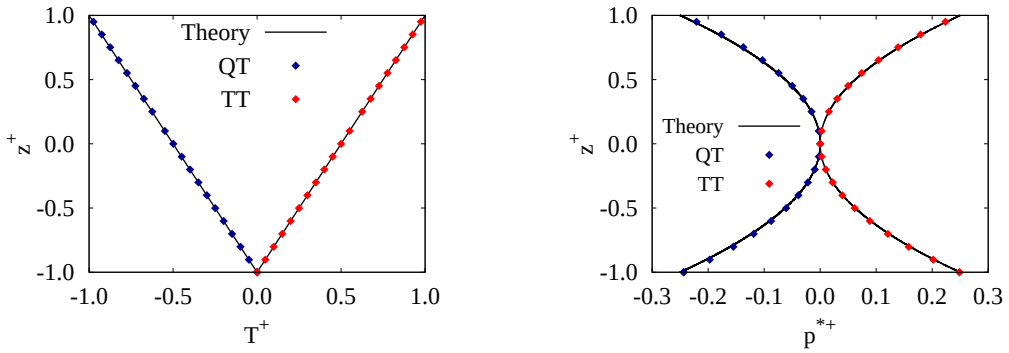


Figure 4.33: Laminar plane Poiseuille flow: profiles of dimensionless temperature (left) and dynamic pressure (right) on the vertical section of the channel obtained with ISPH-USAW, compared to the theoretical solutions of Table 4.2.

are compared to FV and to the ones obtained by Wan *et al.* [154] by discrete singular convolution. Excellent agreement was obtained with both methods. The same quality of results was obtained with ISPH-USAW for $Ra = 10^3$ and $Ra = 10^4$.

The local Nusselt number measures the ratio of convective over conductive heat transfer across the boundary. For a wall segment s, it is defined as $Nu_s = L|\left(\frac{\partial T}{\partial \boldsymbol{n}}\right)_s \cdot \boldsymbol{n}_s|/\Delta T$ and computed according to:

$$Nu_s = \frac{L}{\Delta T}|\mathbf{G}_s^{\gamma,-}\{T_b\} \cdot \boldsymbol{n}_s| \tag{4.18}$$

where $\boldsymbol{G}_a^{\gamma,-}$ is defined by (2.144). Figure 4.36 shows the evolution of Nu_s along the cold wall for the three values of Rayleigh number. It appears that the behaviour is globally well predicted. However, a discrepancy occurs on the top of the curve $Ra = 10^5$, which corresponds to the top right corner of the flow in Figure 4.34, where temperature gradients are rather high. It is a consequence of a lack of accuracy of the $\mathbf{G}^{\gamma,-}$ SPH gradient operator, used to compute Nu_s, since our temperature profiles are still in very good agreement with FV in this area. Note that using a first-order consistent gradient, $\boldsymbol{G}_a^{\gamma,-,1}$ (2.158) did not improve the results.

4.3.1.3 Differentially heated lid-driven cavity

A differentially heated lid-driven cavity at $Re = \frac{UL}{\nu} = 1000$ was tested, L being the size of the cavity and U the velocity of the lid. The flow is driven by the shear force resulting from the lid motion and by the buoyancy force. The upper and lower walls are isothermal, their temperatures being of T_c and T_h respectively. The molecular Prandlt number was set to 1 and the Grashoff number to 10^4 . A discretisation of 500×500 particles was used for the SPH simulation. Figure 4.37 shows the shape of the temperature and velocity fields after convergence. The results are compared to FV using a discretisation of 512×512 cells. Figures 4.38 and 4.39 show the dimensionless velocity and temperature profiles along $x^+ = 1/2$ and $z^+ = 1/2$. Very good agreement is obtained

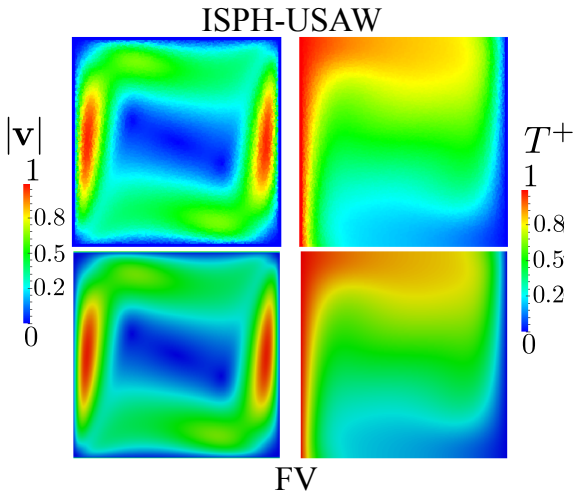


Figure 4.34: Differentially heated square cavity at $Ra = 10^5$. Shape of the temperature (right) and velocity (left) fields obtained with ISPH-USAW (top) and FV (bottom) after convergence.

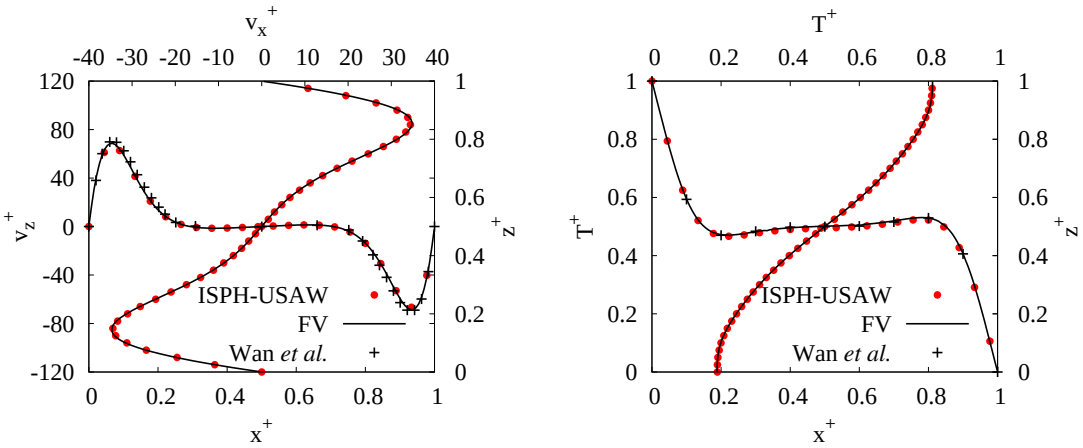


Figure 4.35: Differentially heated square cavity at $Ra = 10^5$. Profiles of velocity (left) and temperature (right) in $x^+ = 1/2$ and $z^+ = 1/2$ obtained with ISPH-USAW and FV after convergence. The horizontal profiles are also compared to the ones obtained by Wan *et al.* [154] with the discrete singular convolution method.

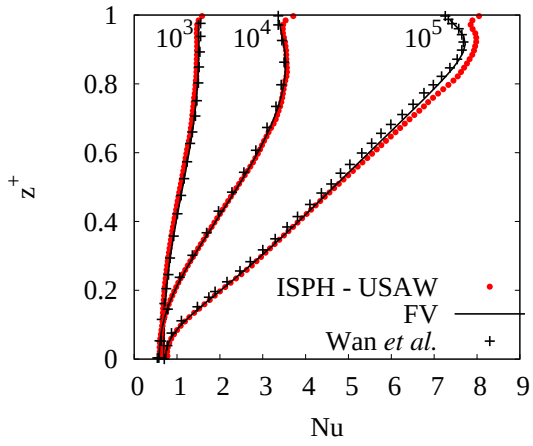


Figure 4.36: Differentially heated square cavity. Evolution of the Nusselt number along the cold wall of the cavity at $Ra = 10^3$, 10^4 and 10^5. Comparison of the results obtained with ISPH-USAW, FV and discrete singular convolution [154] after convergence.

with FV for the velocity and the temperature fields. Figure 4.40 shows the repartition of local Nusselt number (computed according to (4.18)) along the upper and lower walls of the cavity. The formula (4.18) was used to compute the Nusselt number in the SPH simulation, and this time due to the fine discretisation used in the SPH model the agreement with FV is very good, although there is a small underestimation of Nu with ISPH-USAW near the upper-left corner.

4.3.1.4 Lock-exchange

This validation case consists of a symmetric lock-exchange flow in a rectangular cavity of height $2L$ and width $30L$. This case was studied in [37] with the SPH method. All lengths are made dimensionless by L, the half-height of the cavity. The flow consists of two fluids at temperatures T_h (on the right) and T_c (on the left) separated at $t = 0$ at the half-width of the domain. The dimensionless numbers describing the flow are $Gr = 1.25 \times 10^6$ and $Pr = 1$. For this test-case, the dimensionless time is defined through: $t^+ = \frac{tU}{L}$ with $U = \sqrt{\beta \Delta T g L}$. Figure 4.41 shows the shape of the temperature field obtained with ISPH-USAW at $t^+ = 10$. The SPH simulation was done with a discretisation of 1500×100 particles. Figure 4.42 shows the temperature contours obtained with ISPH-USAW at several instants, compared to the ones obtained by Härtel *et al.* [44] through a 2-D Direct Numerical Simulation (DNS) with a mixed spectral/spectral-element discretisation in space together with finite differences in time. The shape and velocity of the front are well reproduced by the present SPH model. It should be noted that the results shown in Figure 4.42 were obtained with a symmetric operator for the pressure gradient (2.144), which better reproduced the vortices at the interface of the two fluids compared to the DNS results. For all other test-cases in the present thesis, an antisymmetric operator (2.145) was used since it conserves linear momentum, but no significant differences were observed when using a symmetric operator.

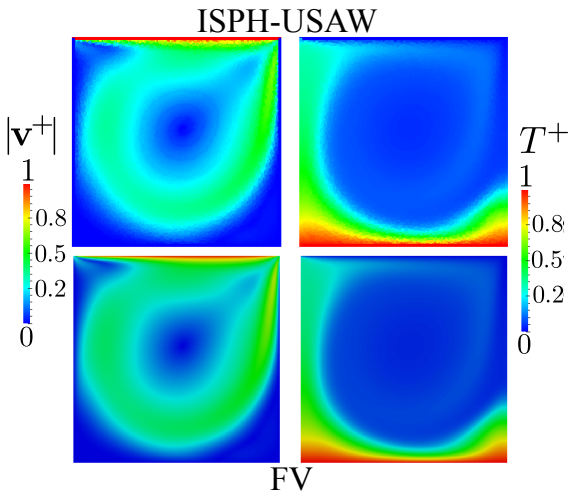


Figure 4.37: Differentially heated lid-driven cavity. Shape of the temperature (right) and velocity (left) fields obtained with ISPH-USAW (top) and FV (bottom) after convergence.

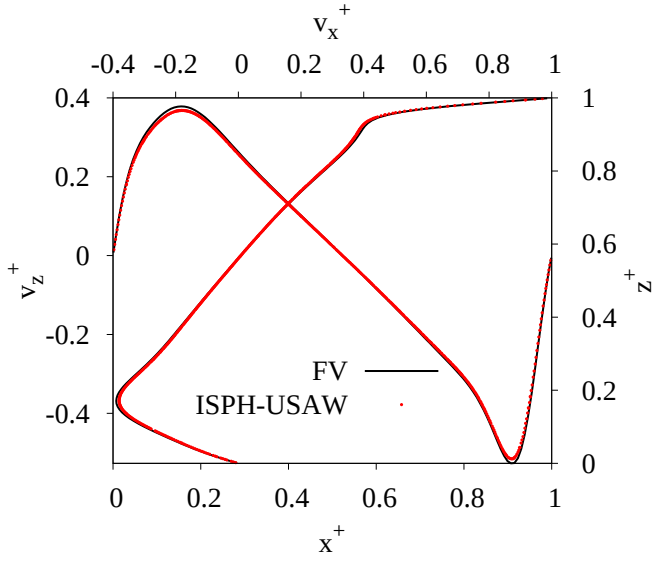


Figure 4.38: Differentially heated lid-driven cavity. Profiles of velocity in $x^+ = 1/2$ and $z^+ = 1/2$ obtained with ISPH-USAW and FV after convergence.

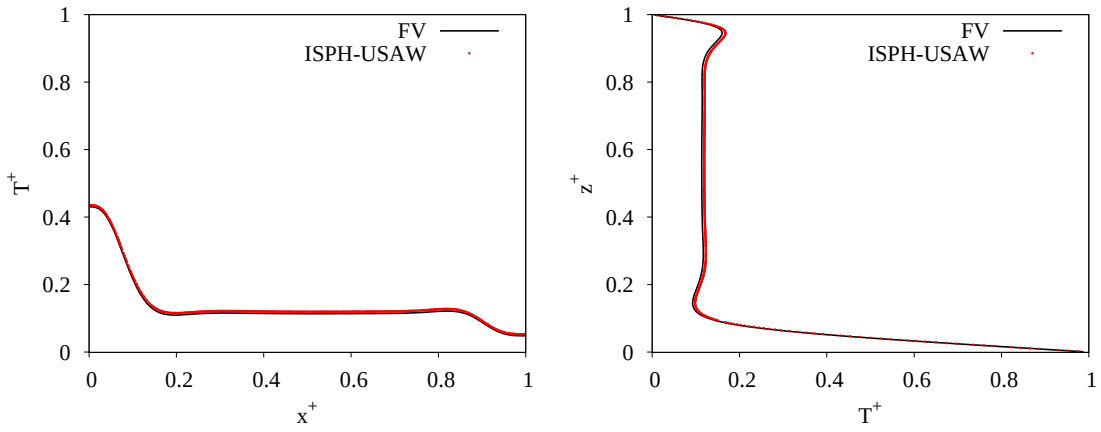


Figure 4.39: Differentially heated lid-driven cavity. Profiles of temperature in $z^+ = 1/2$ (left) and $x^+ = 1/2$ (right) obtained with ISPH-USAW and FV after convergence.

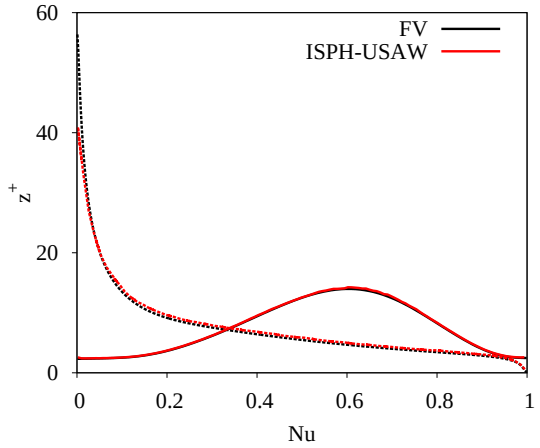


Figure 4.40: Differentially heated lid-driven cavity. Evolution of the Nusselt number along the lid (dash lines) and the lower wall (solid lines). Comparison of the results obtained with ISPH-USAW and FV after convergence.

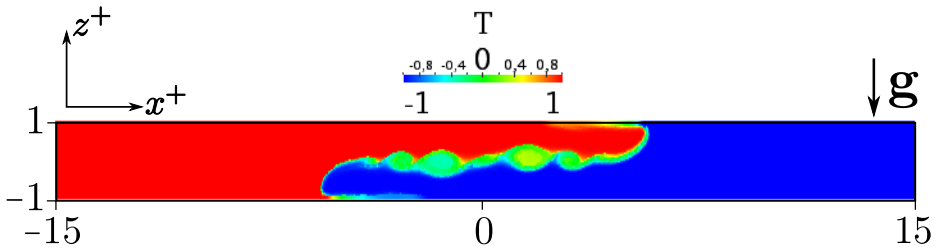


Figure 4.41: Lock-exchange: shape of the temperature field obtained with ISPH-USAW at $t^+ = 10$.

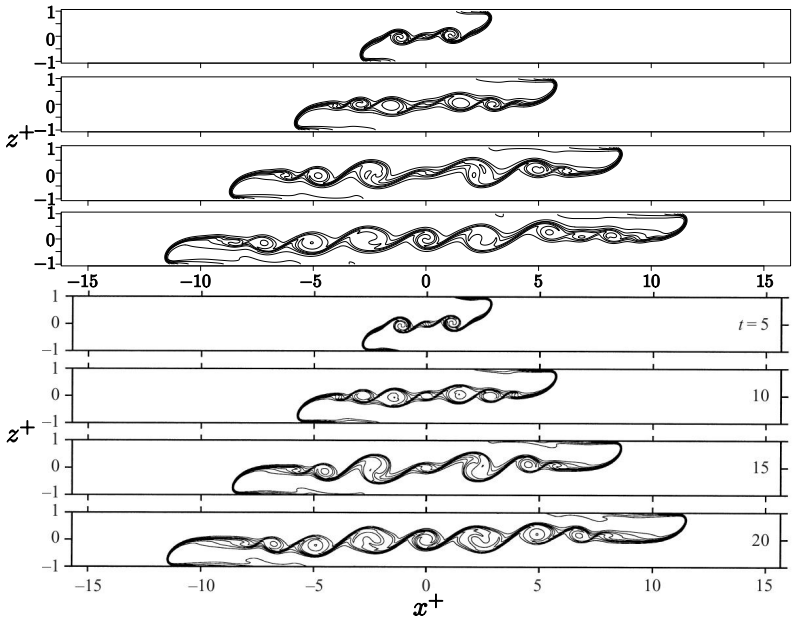


Figure 4.42: Lock-exchange: temperature contours at $t^+ = 5, 10, 15, 20$. Top: ISPH-USAW. Bottom: Härtel *et al.* [44].

4.3.2 Turbulent flows

The validation of the $k - \epsilon$ turbulence model with buoyancy was done on two classical validation cases: a turbulent plane Poiseuille flow with two isothermal walls and a rectangular differentially heated cavity.

4.3.2.1 Turbulent plane Poiseuille flow with two isothermal walls

This case consists of a turbulent flow between two parallel, infinite vertical isothermal walls. The temperature of the left wall is set to T_h and that of the right wall to T_c, with $T_h > T_c$. The flow is thus vertical (and invariant along z), driven by a prescribed pressure gradient (the friction velocity is imposed and is the characteristic flow velocity U) and by the temperature difference between the walls. The buoyancy force acts upwards near the hot wall and downwards near the cold wall. The friction Reynolds number, $Re_* = \frac{u_* L}{\nu}$ is set to 150, where L is the half-width of the channel and u_* is the friction velocity, used to defined the dimensionless variables, and which was set to $1ms^{-1}$ through the imposition of an upward volumic force:

$$f = \frac{u_*^2}{L} - g\beta \frac{\sum_{b\in\mathcal{F}} V_b(T_b - T_0)}{\sum_{b\in\mathcal{F}} V_b} \tag{4.19}$$

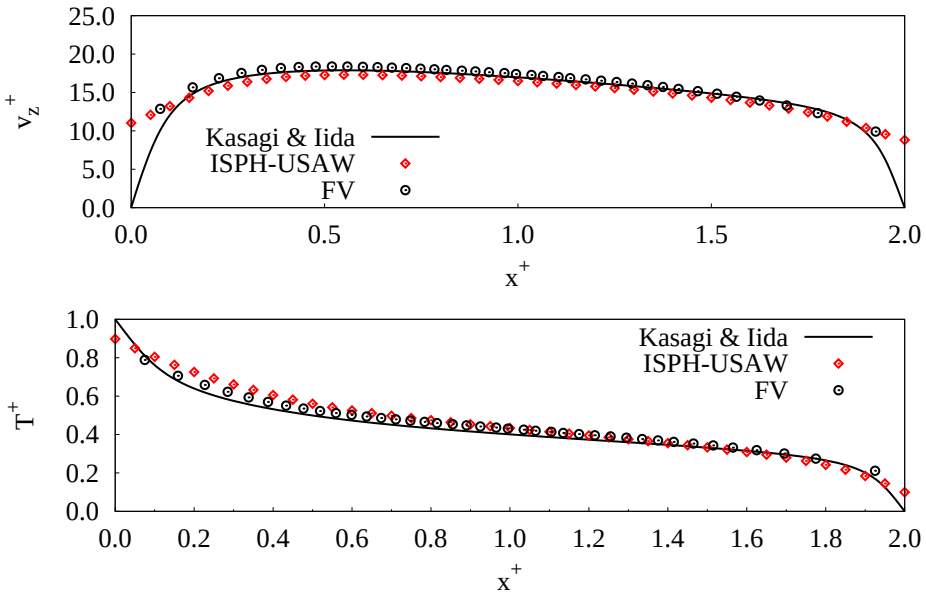


Figure 4.43: Turbulent plane Poiseuille flow. Profiles of velocity (top) and temperature (bottom) after convergence. Comparison of ISPH-USAW and FV with DNS results provided by Kasagi & Iida [58].

where $T_0 = \frac{T_h+T_c}{2}$. The molecular Prandlt number is set to 0.71 and the Grashoff number to $Gr = 9.6 \times 10^5$. The results obtained with SPH are compared to FV and to DNS data provided by Kasagi & Iida [58]. Figures 4.43 and 4.44 show the profiles of velocity, temperature, turbulent kinetic energy and dissipation rate as functions of x^+. Good agreement is observed between the FV and SPH results, although some discrepancies on k and ϵ are visible close to the walls. They seem to be due to the differences in the imposition of the wall boundary conditions on those fields between FV and ISPH-USAW.

4.3.2.2 Differentially heated rectangular cavity in turbulent regime

The last 2-D validation case consists of a differentially heated rectangular cavity of aspect ratio 4. The geometry of the case is described in Figure 4.45. The left and right walls are isothermal at temperature T_h and T_c respectively. The upper and lower walls are adiabatic. The reference length L is the height of the cavity. The temperatures are made dimensionless by $\Delta T = T_h - T_c$. The value of the molecular Prandtl number is 0.71 and that of the Rayleigh number Ra is 6.4×10^8. In the SPH simulation, a discretisation of 50×200 particles was used. The SPH results are compared to DNS results provided by Trias *et al.* [142] and to FV. For the FV simulation the same discretisation than in SPH was used. A comparison of the shape of the temperature field after convergence between SPH and FV is provided in Figure 4.45. Figure 4.46 shows the temperature profiles along

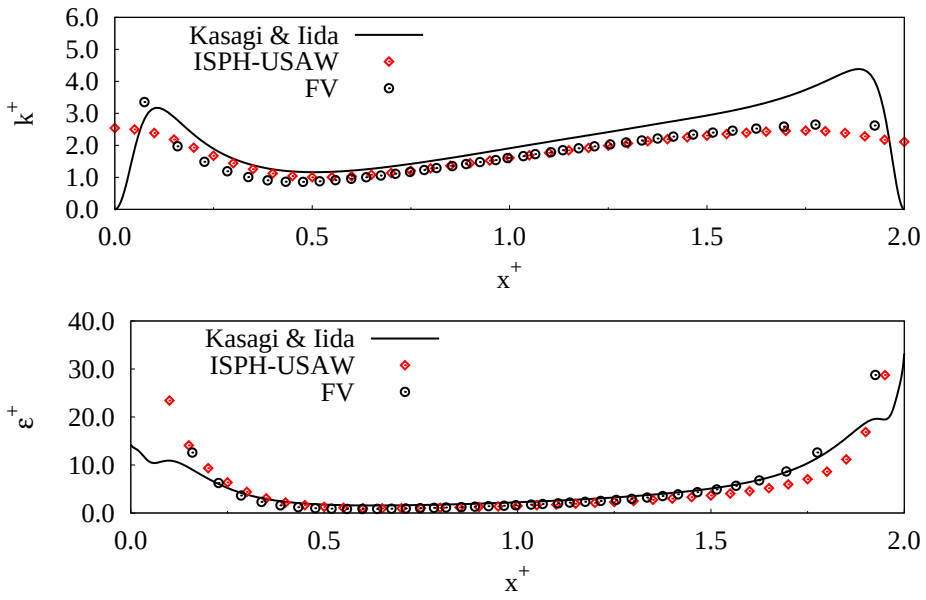


Figure 4.44: Turbulent plane Poiseuille flow. Profiles of turbulent kinetic energy (top) and dissipation rate (bottom) after convergence. Comparison of ISPH-USAW and FV with DNS results provided by Kasagi & Iida [58].

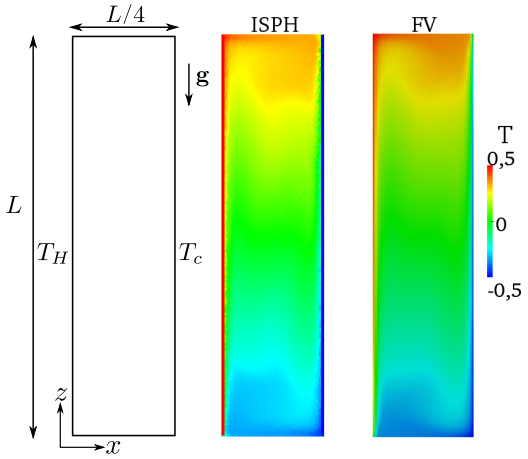


Figure 4.45: Differentially heated rectangular cavity. Sketch of the case (left) and shape of the temperature field after convergence with ISPH-USAW (middle) and FV (right).

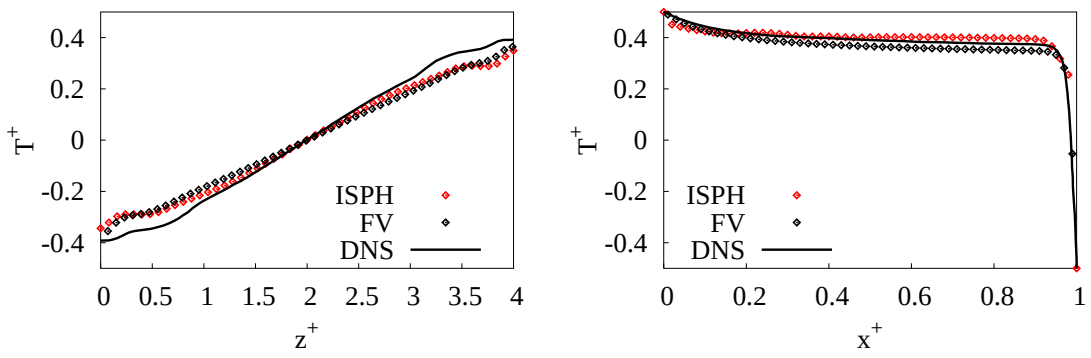


Figure 4.46: Differentially heated rectangular cavity. Profiles of temperature after convergence along $x^+ = 0.125$ (left) and $z^+ = 1$ (right). Comparison of ISPH-USAW and FV with DNS results provided by Trias *et al.* [142].

$x^+ = 0.125$ and $z^+ = 1$. A satisfactory agreement is observed between ISPH-USAW and FV. Both methods present reasonable agreement with the DNS results, although some differences are observed, which were expected since a 2D RANS $k - \epsilon$ model is not meant to perfectly reproduce 3D DNS data.

Chapter 5

3-D cases: validation and preliminary application results

Ce Chapitre traite de la validation du modèle présenté au Chapitre 3 sur des cas 3-D. Des résultats préliminaires sur un cas d'application sont également présentés. Dans un premier temps, un cas d'écoulement laminaire dans un tuyau à section circulaire avec des frontières ouvertes est présenté. On considère ensuite une rupture de barrage sur un obstacle. Enfin, on propose un cas d'écoulement dans des tuyaux connectés présentant des températures différentes.

This Chapter focuses on the validation of the buoyant incompressible SPH model described in Chapter 3 with USAW boundary conditions in 3-D. The latter is referred to as ISPH-USAW, as in the previous Chapter. The notations used in this Chapter are nearly exactly the same as in Chapter 4, except for the coordinates system which is now (x, y, z), the horizontal, transverse and vertical coordinates. Two validation cases in 3-D are presented in what follows. The first case is an isothermal laminar flow in a circular pipe with inflow/outflow conditions, for which a theoretical solution is known. The second case is a schematic dam-break over an obstacle, for which comparisons are provided with a WCSPH model that uses the USAW boundary conditions (WCSPH-USAW). Comparisons are also provided with VoF results, that were obtained with the OpenFOAM open-source software [140]. The last section of this Chapter presents preliminary results on a simple application case that consists of a flow through two connected pipes at different temperatures. For this case, inflow/outflow conditions are prescribed at the extremities of the two pipes. The $5th$ order Wendland kernel (2.10) was used for the simulations with a smoothing length $h = 1.3\delta r$ (recall δr is the initial interparticular spacing). In all the simulations the reference density of the fluid is $\rho = 1000\ kg.m^{-3}$. Note that the 3-D geometries were generated with a GPU pre-processing software called Crixus[1] [91] that takes a skin-mesh of the boundary as input and creates an SPH geometry from it, filling it with particles where required. This pre-processing tool computes the masses of the particles and stores the connectivity between segments and vertices as well.

5.1 Laminar flow in a circular pipe with inflow/outflow boundaries

This case consists of a laminar flow through a 3-D pipe with a circular cross-section. Inflow and outflow boundaries are imposed at the extremities of the pipe. The reference length of the flow L is the radius of the cross-section, it serves to make all lengths dimensionless. The dimensionless length of the pipe is equal to 4. The reference velocity of the flow U is the maximum velocity in the pipe, set to $1ms^{-1}$ by imposing the theoretical dimensionless velocity at the inlet:

$$\boldsymbol{v}^+ = \left[1 - (y^+ - y_0^+)^2 + (z^+ - z_0^+)^2\right] \boldsymbol{e}_x \tag{5.1}$$

where $(y_0^+, z_0^+) = (0, 0)$ are the dimensionless transverse and vertical coordinates of the cross-section centre. At the outlet, a zero-pressure is imposed. The Reynolds number is set to 10. The dimensionless time is defined as $t^+ = \frac{tU}{L}$. The simulation is run until $t^+ = 35$, which corresponds to about 1.4×10^5 iterations with an initial dimensionless interparticular space $\delta r^+ = 0.3$. Figure 5.1 shows the shape of the velocity field in the pipe at $t^+ = 35$.

Figure 5.2 shows velocity and pressure profiles in the pipe at $t^+ = 35$. On the left, the horizontal dimensionless velocity v_x^+ obtained with ISPH-USAW is plotted as a function of z^+ along the

[1] Code available at https://github.com/Azrael3000/Crixus.

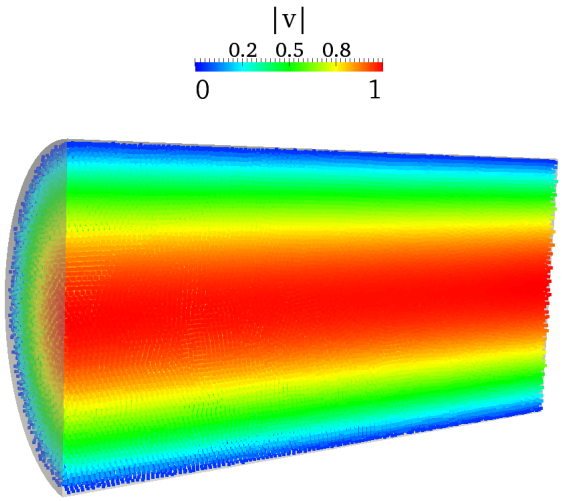


Figure 5.1: Laminar flow in a 3-D circular pipe with inflow/outflow conditions. Shape of the velocity profile obtained with ISPH-USAW using $\delta r^+ = 0.3$ at $t = 35s$.

vertical profile at the centre of the channel ($x^+ = 2$, $y^+ = 0$). Note that the same results were obtained on a vertical profile in $y^+ = 0$ but at $x^+ = 3.9$ instead of $x^+ = 2$. Good agreement with the theoretical parabolic profile (5.1) is obtained. On the right of Figure 5.2, the dimensionless pressure p^+ obtained with ISPH-USAW is plotted as a function of x^+ along the horizontal profile at the centre of the channel. The agreement with the theoretical linear pressure distribution along the channel is good, the latter being given by:

$$p^+ = \frac{8}{Re}\left(4 - x^+\right) \tag{5.2}$$

where $Re = \frac{UL}{\nu}$. A small discrepancy close to the inflow boundary appears, where a homogeneous Neumann condition is imposed on the pressure. Nevertheless, the quality of the results shows that the 3-D ISPH-USAW model is performing well, even with inflow/outflow conditions.

5.2 Dam-break over an obstacle

This case consists of a 3-D schematic dam-break over an obstacle and is used to assess the capability of the 3-D formulation to reproduce free-surface flows. The geometry is provided as the second SPHERIC validation test case [55], with pressure and water-depth probes located at the same positions. Figures 5.3 and 5.4 show a sketch of the geometry and the dimensions of the problem with the location of the pressure and water height probes.

A viscosity of $\nu = 10^{-2} m^2 s^{-1}$ is used so that the flow remains laminar. There are still issues regarding the modelling of low-viscosity flows with a free-surface: the impermeability of the walls is not granted everywhere. Note that this is also the case regarding low-viscosity flows with in-

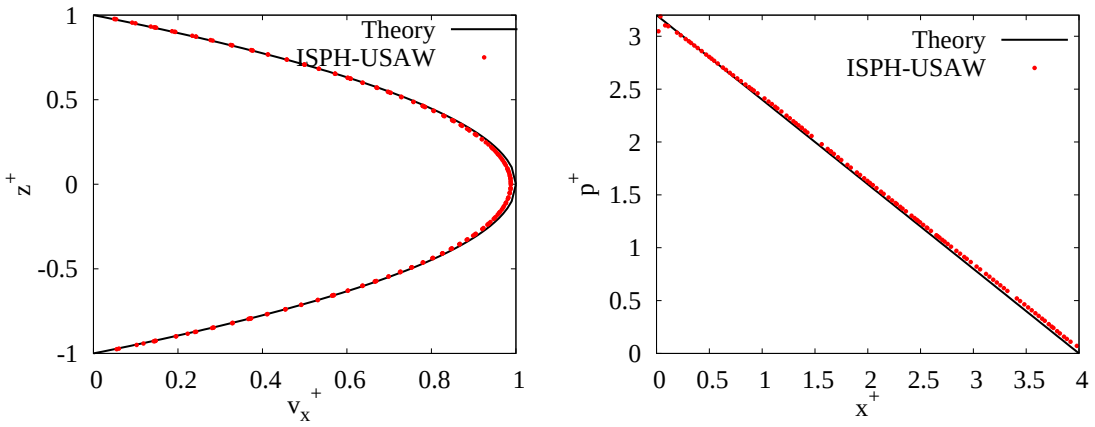


Figure 5.2: Laminar flow in a 3-D circular pipe with inflow/outflow conditions. Dimensionless velocity profile along the vertical line at the centre of the channel (left) and dimensionless pressure profile along the horizontal line at the centre of the channel (right). Comparison of the results obtained with ISPH-USAW using $\delta r^+ = 0.3$ with the theoretical velocity and pressure profiles.

let/outlet in 3-D. This is why a high viscosity was used, thus preventing a comparison with the available experiments results on this case. Instead, comparisons are presented with VoF results.

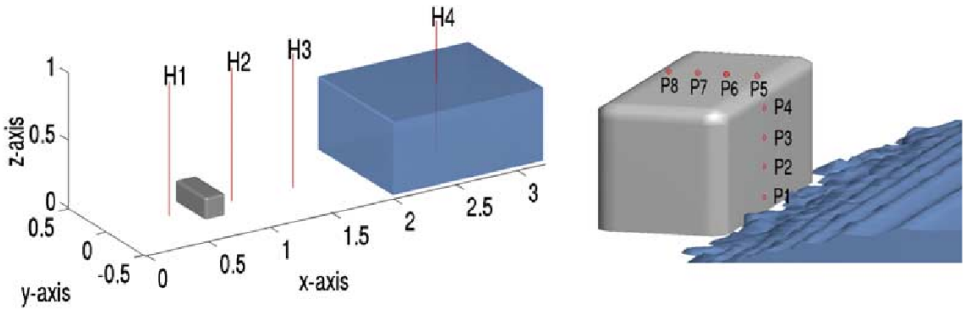


Figure 5.3: Schematic 3-D dam-break over an obstacle. Sketch of the geometry [55].

Figure 5.5 shows the shape of the free-surface and velocity field obtained with ISPH-USAW with $\delta r = 0.02m$ at several dimensionless times. The dimensionless time is defined as $t^+ = \frac{t}{\sqrt{gH}}$, with $H = 0.55m$ the initial water height. On the bottom-left picture of Figure 5.5, it is visible that a consequent number of particles remains stuck to the walls during the SPH simulation. This is due to the high viscosity of the fluid considered here. Note that the walls impermeability is ensured during the simulation.

Figures 5.6 and 5.7 show the time-evolution of the pressure at probes P2 and P5, and of the water depth at probes H2 and H4 (see Figures 5.3 and 5.4). Comparisons are provided with a WCSPH model using the USAW boundary conditions, that was run with a massively parallel CPU code with $\delta r = 0.0183m$ [91]. A Ferrari density correction was used for the WCSPH simulation and the speed of sound was taken as $40ms^{-1}$. In the WCSPH simulation, $\boldsymbol{\nabla}\gamma_a$ was computed analytically

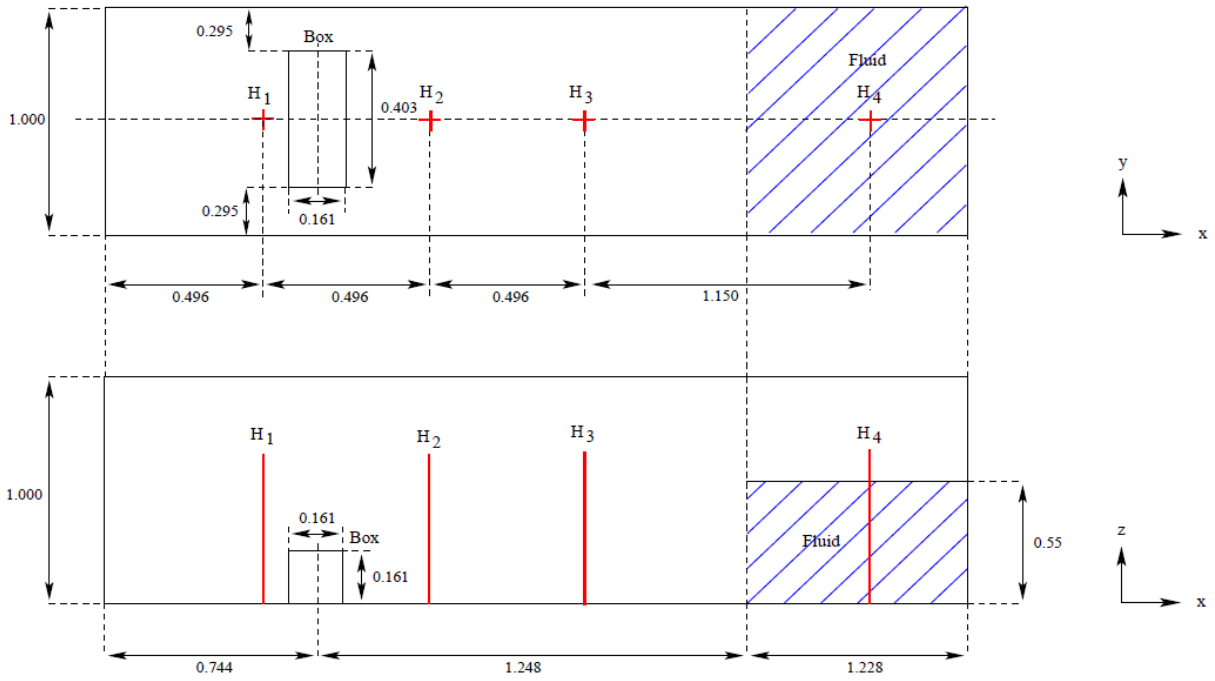


Figure 5.4: Schematic 3-D dam-break over an obstacle. Case dimensions and location of the pressure and water height probes [55].

through the formula proposed in [92], which was shown to yield the same results as the simpler formula used for the ISPH simulation (see Appendix A) [150]. γ_a was computed through a dynamic governing equation in the two SPH models. Comparisons between the SPH results and VoF are provided. Recall that in VoF the simulations are done for a two-phase (air + water) model, which limits the extent of the comparison with the single-phase SPH models. Besides, in the visualisation of VoF results, the definition of the free-surface is not obvious and it is considered as the locations where the volume fraction of fluid is 0.5. Though, this comparison is useful to check the accuracy of the 3-D model. The results obtained with VoF are considered as a reference against which the ones obtained with SPH are compared. The three methods give similar results, although some differences appear between the models, which can be due to the two-phase nature of VoF, while the SPH models are single-phase. In particular, air happens to be trapped inside the fluid close to the obstacle in the VoF simulation. This can explain the differences in the water level between the SPH models and VoF at probe H2, which is close to the obstacle, while the effect of air is less visible at probe H4 which is far from the obstacle (see Figure 5.6). The new ISPH model better predicts the pressure peak that occurs when the flow hits the obstacle compared to WCSPH (see Figure 5.7).

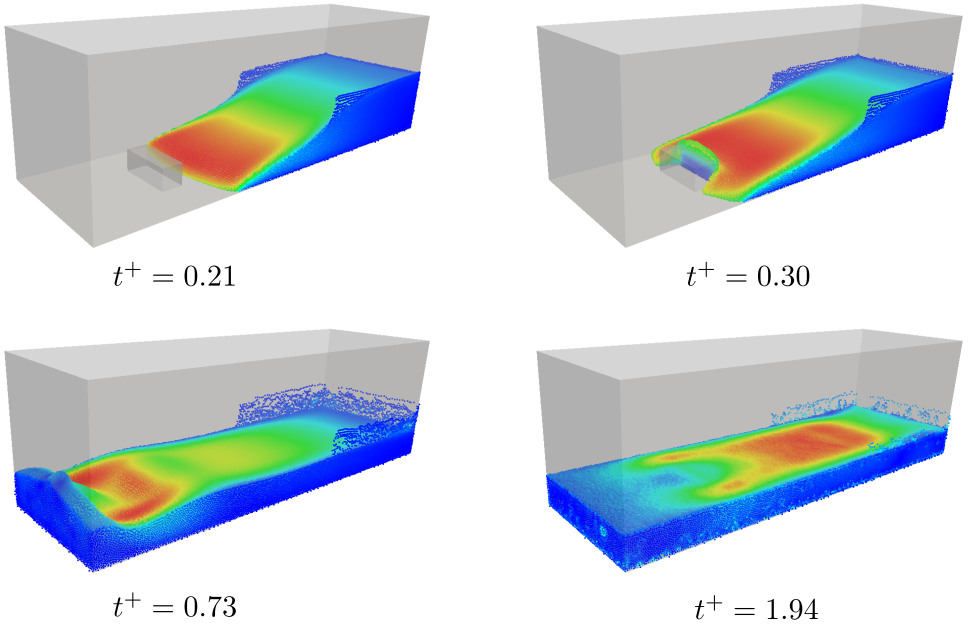


Figure 5.5: Schematic 3-D dam-break over an obstacle. Shape of the free-surface and velocity field obtained with ISPH-USAW with $\delta r = 0.02m$ at several dimensionless times.

5.3 Connected pipes case

This case is a preliminary application case in 3-D that consists of two connected pipes at different temperatures. The geometry is described in the Figure 5.8. The reference length L is the diameter of the larger pipe. At the initial time, fluid at the temperature T_h is placed in a horizontal pipe with zero velocity and pressure and fluid at the temperature T_c is placed in a smaller inclined pipe connected to the first one with zero velocity and pressure. As time goes by, fluid at the temperature T_h is injected through the left extremity of the horizontal pipe and fluid at the temperature T_c is injected at the highest extremity of the inclined pipe. The velocity is imposed at these inflow boundaries. In the horizontal pipe, it is imposed through:

$$\boldsymbol{v} = U\left[1 - (y - y_0)^2 - (z - z_0)^2\right]\boldsymbol{n}_0 \tag{5.3}$$

where y_0 and z_0 are the transverse and vertical coordinates of the centre of the big pipe cross-section at the inlet, and $\boldsymbol{n}_0 = \boldsymbol{e}_x$ is the unit normal vector to that cross-section. U is the reference velocity of the flow and was set to $0.5ms^{-1}$. The Reynolds number based on U and L was set to 10. On the other hand, in the inclined pipe the inlet velocity is imposed through

$$\boldsymbol{v} = \frac{U}{2}\left(1 - (x - x_1)^2 - (y - y_1)^2\right)\boldsymbol{n}_1 \tag{5.4}$$

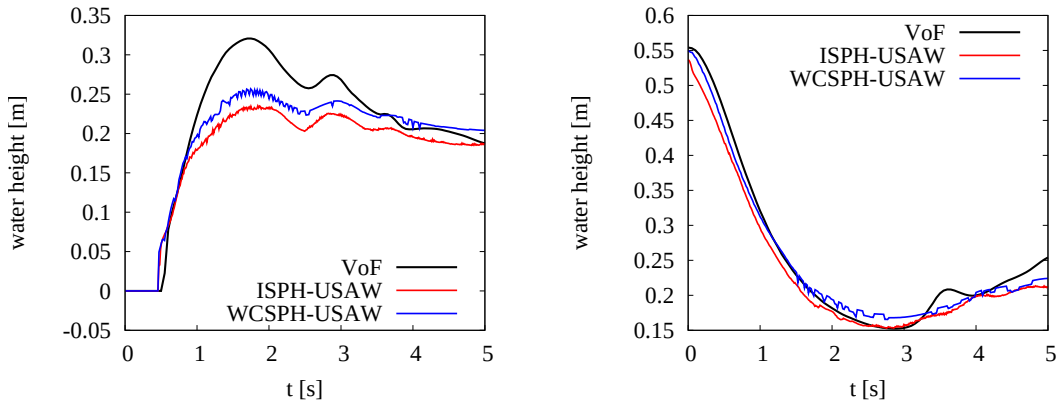


Figure 5.6: Schematic 3-D dam-break over an obstacle. Time-evolution of the water depth at probes number H2 and H4. Comparison between ISPH-USAW, WCSPH-USAW [91] and VoF.

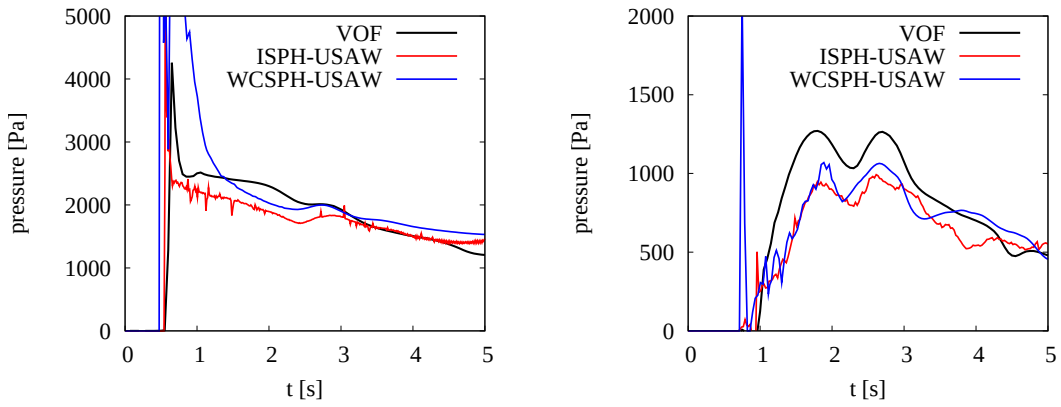


Figure 5.7: Schematic 3-D dam-break over an obstacle. Time-evolution of the pressure at probes number P2 and P5. Comparison between ISPH-USAW, WCSPH-USAW [91] and VoF.

where x_1 and y_1 are the horizontal and transverse coordinates of the centre of the small pipe cross-section at this inlet, and $\boldsymbol{n}_1 = \left(-\frac{1}{\sqrt{3}}, 0, -\frac{2}{\sqrt{3}}\right)$ is the unit normal vector to that cross-section. An outflow boundary condition is imposed at the right extremity of the horizontal pipe: the pressure is imposed to zero and a homogeneous Neumann condition is imposed on the temperature. A homogeneous Neumann condition is also imposed on the temperature at solid walls. The Grashoff number (4.3) was set to 0.162, and the Prandlt (4.2) number to 69[2]. Figure 5.9 shows the shape of the temperature field at several times during the simulation.

The application of the new ISPH model to more complex 3-D flows is an ongoing work. Though, the 3-D model is not as robust as the 2-D one, which seems closely related to the accuracy and robustness of the computation of γ_a. Further work is thus required regarding its computation in

[2]Actually the parameters of water are used: $\beta = 2.07 \times 10^{-4} K^{-1}$ and $K = 1.43 \times 10^{-4} m^2 s^{-1}$, but the viscosity was set to $10^{-2} m^2 s^{-1}$ otherwise the flow is unstable, which is due to the inlet/outlet formulation (this was evoked in section 5.2).

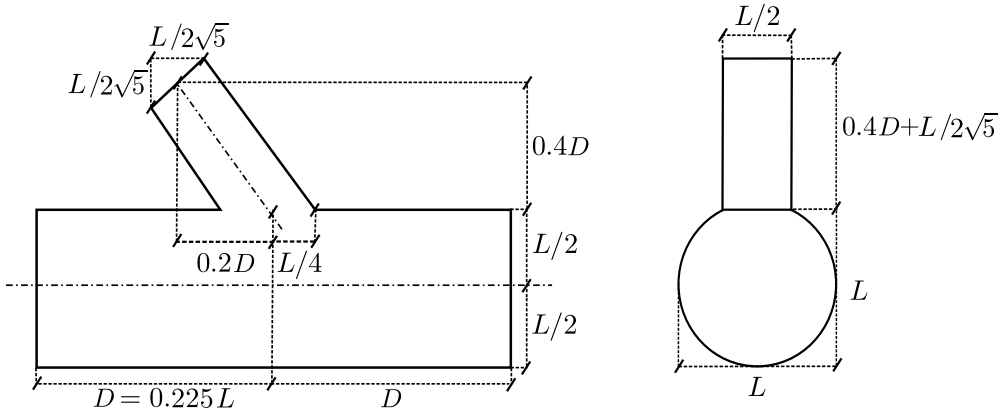


Figure 5.8: Laminar flow in two connected pipes at different temperatures. Sketch of the geometry.

3-D. Besides, the issues regarding low viscosity flows with the inlet/outlet formulation and with free-surfaces require investigation. Nevertheless, the possibilities of applications of the new model are promising.

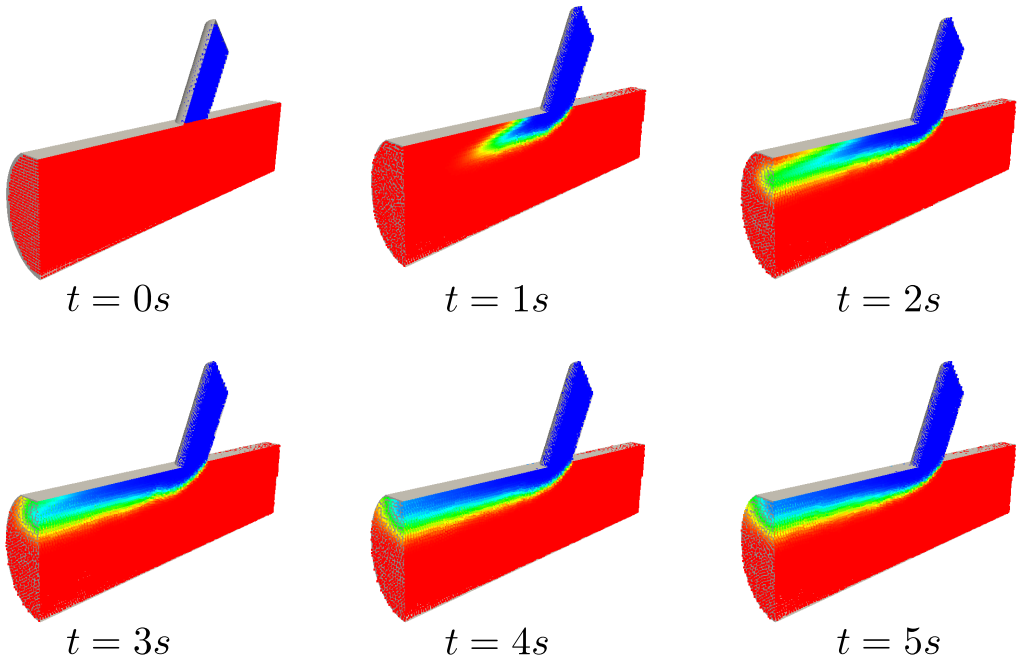


Figure 5.9: Laminar flow in two connected pipes at different temperatures. Shape of the temperature field at several times.

Conclusions

Dans cette thèse un nouveau modèle SPH incompressible est proposé, où la technique des conditions aux limites semi-analytiques est utilisée pour la représentation de parois et de frontières ouvertes. Une amélioration majeure par rapport aux modèles SPH incompressibles existants est l'imposition exacte d'une condition de Neumann non-homogène consistente sur la pression au niveau des parois. Un modèle de turbulence $k-\epsilon$ et un modèle de scalaires actifs ont été intégrés, ce dernier reposant sur l'approximation de Boussinesq. Les interactions entre la flottabilité et la turbulence sont modélisées. L'utilisation de conditions aux limites semi-analytiques permet d'imposer précisément les conditions aux limites adaptées sur les champs. Plusieurs cas-tests en 2-D sont proposés pour la validation du modèle. Les résulats montrent que le modèle permet de représenter précisément des écoulements laminaires et turbulents confinés ou à surface libre. Avec les conditions aux limites utilisées, il est possible d'appliquer le modèle à des problèmes à géométrie complexe. De manière générale, on obtient une bonne correspondance entre les résultats du nouveau modèle et des méthodes à maillages. De plus, les résultats obtenus avec le nouveau modèle sont plus précis que ceux obtenus avec un modèle SPH quasi-incompressible, en particulier sur la pression, et sont obtenus en des temps de calcul similaires. Deux cas de validation en 3-D sont présentés. Des améliorations du modèle $k-\epsilon$ présenté ici sont possibles, et il serait intéressant de développer d'autres modèles de turbulence comme un modèle bas-Renolds. Le comportement de la surface libre reste pourtant à améliorer, ce qui pourrait permettre de résoudre le problème de fuites à travers les parois en faible viscosité avec une surface libre. Le modèle est destiné à être implémenté dans le logiciel GPUSPH, ce qui devrait permettre l'application à des cas d'application réels grâce au multi-GPU et à la possiblité de représenter des objets mobiles dans l'écoulement.

5.1. Achievements of this work

In this thesis a new ISPH (Incompressible SPH) method is proposed, in which solid boundaries are modelled through the unified semi-analytical wall (USAW) boundary conditions. One major improvement compared to a classical ISPH model is the exact imposition of a non-homogeneous Neumann wall boundary condition on the pressure field to solve the pressure Poisson equation, which makes it possible to prescribe the impermeability condition on solid walls. In order to treat industrial or environmental flows, a $k-\epsilon$ turbulence closure and a model for active scalars effects were introduced in the new ISPH model. The buoyancy model is based on the Boussinesq approximation. The interactions between buoyancy and turbulence are modelled. On the other hand, a technique to represent open boundaries in this formalism is proposed, the open boundary conditions being accurately prescribed on the fields. The use of USAW boundary conditions made it possible to accurately prescribe arbitrary boundary conditions on T, k and ϵ.

Various 2-D test-cases are presented to show that ISPH-USAW is able to accurately model complex laminar and Reynolds-averaged turbulent flows, even with complex geometries. Convergence studies on a lid-driven cavity are presented, the velocity field obtained with FV being taken as a reference, that show a convergence rate close to 2 for the new model. This indicates that the wall boundary conditions are satisfactorily imposed on the pressure. The accuracy of the $k-\epsilon$ turbulence model combined to ISPH-USAW was checked on a turbulent plane Poiseuille flow where an excellent agreement between our results and DNS and FV results is observed. Besides, our results are in good agreement with the ones obtained with FV in the case of the fish-pass. The buoyancy model is tested in laminar and turbulent regimes and good agreement is obtained with FV in all cases. Several cases presenting open boundaries are also presented, including two cases of propagation of a solitary wave, which show the inflow/outflow conditions are properly imposed.

In general, the results obtained with the proposed ISPH-USAW model are better than with WCSPH-USAW, especially regarding the pressure prediction, and are obtained in most cases with a similar computational time. To achieve this reduction of computational time in ISPH-USAW, the wall renormalisation factor γ_a is computed through an analytical formula, extending the method proposed by Feldman and Bonet [33] to our wall discretisation in 2-D. It should be noted that with the USAW boundary conditions it is possible to apply the ISPH method to complex geometries, not easy to handle with the traditional SPH wall treatments like ghost particles. Besides, the model proposed here is advantageous compared to mesh-based methods for the simulation of flows presenting complex free-surface shapes and/or involving moving solid bodies.

Finally, two laminar 3-D validation cases are proposed, one of them with inflow/outflow conditions, and preliminary results on a simple application case are presented. The application of the new model to real-life cases is close to hand.

5.2. Perspectives

In order to improve the present work, we now suggest possible further developments.

- Turbulence modelling:
  Possible improvements concerning the turbulence modelling could be to take the deviatoric part of $\boldsymbol{S}$ into account, to have a varying turbulent Prandlt number, to use a harmonic mean of the viscosities in the viscous term instead of an arithmetic mean. On the other hand, the implementation of other RANS models would be interesting, in particular a low-Reynolds model. Further validation could include a turbulent jet and comparisons with experimental results. Moreover, the Neumann condition on the temperature could be adapted in case of turbulence in order to include the temperature wall function.

- Boundary conditions:
  A stable formulation for the analytical computation of γ_a in 3-D has not been reached yet. Further work is required in order to avoid numerical issues due to discontinuities in the formula proposed in [150]. Using an analytical formula to compute γ_a in 3-D could help reduce computational times, increase the simulations stability and their accuracy.

- Inlet/outlet formulation:
  The technique proposed here for inflow/outflow conditions still requires improvements, since it proved unstable with low viscosity flows (without using a turbulence model). Besides, it is still necessary to validate the inflow/outflow formulation with the $k-\epsilon$ model, since it was not done in the present work. Finally, the radiative condition on the pressure, although it lets the fluid leave the domain without visible reflections, leads to a lowering of the free-surface level, which is problematic.

- Projection method:
  Concerning the projection method, a higher order scheme like a rotational projection scheme could be implemented instead of the Chorin-type scheme proposed here. Then, care must be taken that the viscous term must be treated implicitly in order to impose consistent wall boundary conditions on the pressure.

- Low viscosity flows :
  There are still issues regarding the simulation of low-viscosity free-surface flows: the walls impermeability is not always ensured. This seems closely related to the free-surface detection algorithm but may also be linked to the projection method used here. On the other hand, as mentioned above low viscosity flows with inflow/outflow boundaries are unstable.

- Free-surface flows :
  More advanced techniques for free-surface detection could be used instead of the one chosen

here, like the one proposed in [86]. Moreover, further validation could be done, for example based on the 2-D and 3-D cases considered in [85].

- Particle shifting :
  An improvement could be to use a constant density projection scheme to stabilise the simulations, instead of the particle shift used here. Then, care must be taken regarding the treatment of the boundary conditions. The free-surface shape is incorrectly predicted due to the disabling of the particle shift in its vicinity, and such an approach could solve this issue. Another possibility would be to apply only a tangential shift close to the free-surface, and thus build the shift so as it reduces anisotropy but not heterogeneity in the particle distribution.

- 3-D flows:
  The validation on 3-D flows should be pushed further, and the application to more complex industrial cases should be done.

- Multi-GPU:
  The developments proposed here are meant to be included in the open-source code GPUSPH. One aim is to run 3-D simulations with this model on multiple GPUs. Besides, GPUSPH includes a module for interactions with moving bodies in the flow, which opens more possibilities of applications.

These suggestions do not pretend to be exhaustive.

Appendix A

Analytical computation of $\nabla\gamma_a$ in 3-D with the Wendland kernel

In this Appendix the method to compute the gradient of γ_a through an analytical formula in 3-D is explained. Note that using an analytical formula for $\boldsymbol{\nabla}\gamma_a$ instead of an approximate one greatly improved the results on 3-D simulations with the new ISPH model. Recall that $\boldsymbol{\nabla}\gamma_a$ is defined by:

$$\begin{aligned}\boldsymbol{\nabla}\gamma_a &= \int_{\partial\Omega} w(q)\boldsymbol{n}d\Gamma \\ &= \sum_{s\in\mathcal{S}}\int_s w(q)\boldsymbol{n}_s d\Gamma \\ &= \sum_{s\in\mathcal{S}}\boldsymbol{\nabla}\gamma_{as}\end{aligned} \tag{A.1}$$

where $q = \frac{|\boldsymbol{r}_a - \boldsymbol{r}'|}{h} = \frac{|\tilde{\boldsymbol{r}}|}{h}$. $d\Gamma$ is an elementary surface of the boundary. The idea is to apply the Gauss theorem to the second line of (A.1) so as to obtain $\boldsymbol{\nabla}\gamma_{as}$ as a sum of integrals over the edges of segment s, which is a triangle in 3-D (see Figure A.1), although the boundary discretisation could also be done with quadrangles or other polygons. This reads:

$$\int_s w(q)\boldsymbol{n}_s d\Gamma = \int_{\partial s} \boldsymbol{\chi}(p)\cdot\boldsymbol{n}_{\partial s}dp \tag{A.2}$$

where p is a 2-D dimensionless coordinate in the segment's plane, taking the projection σ of the particle a on s as an origin. On the other hand, $\boldsymbol{\chi}$ is defined through $w(q) = \nabla\cdot\boldsymbol{\chi}(p)$ and $\boldsymbol{n}_{\partial s}$ is

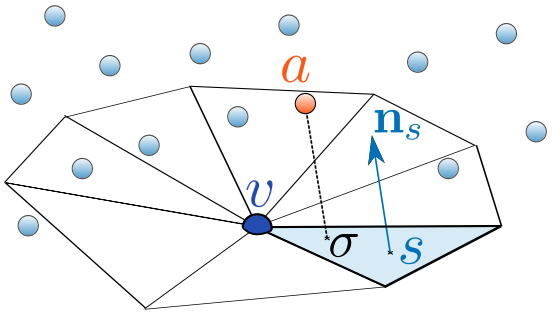


Figure A.1: 3-D sketch for the computation of the gradient of γ_a in 3-D. $\boldsymbol{\nabla}\gamma_a$ is computed as a sum of integrals over the segments s.

the outward normal to the edges of the segment (see Figure A.2). $\boldsymbol{\chi}$ is a radial function since w is, so it can be written as $\boldsymbol{\chi}(p) = -\phi(p)\boldsymbol{p}$ where $\boldsymbol{p}$ is the dimensionless 2-D vector in the segment's plane. As in the case of the analytical computation of γ_a in 2-D (see section 3.8.1), ϕ presents a singularity in $p = 0$, so that the application of the Gauss theorem can only be done up to a small disc s_ϵ of centre σ and radius ϵ, which boundary is denoted by ∂s_ϵ (see Figure A.2):

$$\begin{aligned}\int_s w(q)\boldsymbol{n}_s d\Gamma &= -\int_{\partial s}\phi(p)\boldsymbol{p}\cdot\boldsymbol{n}_{\partial s}dp \\ &\quad -\lim_{\epsilon\to 0}\int_{\partial s_\epsilon}\phi(p)\boldsymbol{p}\cdot\boldsymbol{n}_{\partial s}dp\end{aligned} \tag{A.3}$$

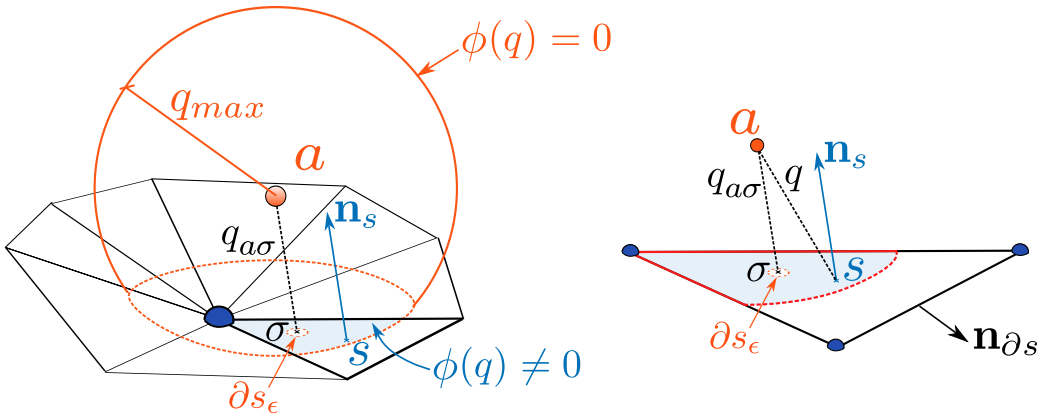


Figure A.2: 3-D and 2-D notations for the computation of the gradient of γ_a in 3-D. ϕ is equal to zero everywhere outside the orange sphere in 3-D. $\boldsymbol{\nabla}\gamma_{as}$ is computed as an integral over the red line but the contribution of the dashed red lines is equal to zero.

With the same method as in section 3.8.1, but using 3-D spherical coordinates [150] we get:

$$\phi(p) = \frac{3}{16h^3p^2}\left(1-\frac{q}{2}\right)^5\left(2+5q+4q^2\right) \tag{A.4}$$

and:

$$h^2\lim_{\epsilon\to 0}\int_{\partial s_\epsilon}\phi(p)\boldsymbol{p}\cdot\boldsymbol{n}_{\partial s}dp = -\frac{3}{8h}\left(1-\frac{q_{a\sigma}}{2}\right)^5\left(2+5q_{a\sigma}+4q_{a\sigma}^2\right) \tag{A.5}$$

Thus, we have:

$$\begin{aligned}|\boldsymbol{\nabla}\gamma_{as}| = h^2\int_s w(q)d\Gamma \quad &= -\frac{3}{16\pi h}\int_{\partial s}\frac{1}{p^2}\left(1-\frac{q}{2}\right)^5\left(2+5q+4q^2\right)\boldsymbol{p}\cdot\boldsymbol{n}_{\partial s}dp \\ &+\frac{3\delta_{\sigma s}}{8h}\left(1-\frac{q_{a\sigma}}{2}\right)^5\left(2+5q_{a\sigma}+4q_{a\sigma}^2\right)\end{aligned} \tag{A.6}$$

where $\delta_{\sigma s} = 1$ if σ is inside segment s and $\delta_{\sigma s} = 0$ otherwise. The integral in the first line of (A.6) can be computed as a summation of integrals over the segment's edges, the latter being denoted by e. The algebraic dimensionless distances along the edges are denoted by l. For example, the algebraic dimensionless distance between a vertex v_0 and the projection of σ on the edge e composed of the vertices v_0 and v_1 is equal to:

$$l_{ev_0} = \frac{\boldsymbol{v}_1-\boldsymbol{v}_0}{|\boldsymbol{v}_1-\boldsymbol{v}_0|}\cdot\boldsymbol{r}_{av_0} \tag{A.7}$$

Then, the three-dimensional distances q may be clipped to 2 since ϕ was chosen so as to be zero when $q \geq 2$. The clipped three-dimensional distances are denoted by q^c. The corresponding clipped 2-D and 1-D coordinates can be deduced through the Pythagorean theorem and are denoted by p^c and l^c. Then, the integral in the first line of (A.6) only needs to be computed up to the clipped quantities, as illustrated by the right sketch of Figure A.2 and Figure A.3. Then, $\boldsymbol{\nabla}\gamma_{as}$ is given

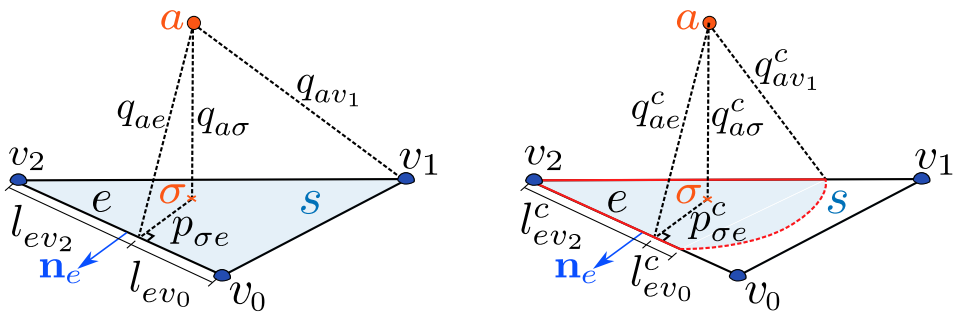


Figure A.3: Definition of the 3-D, 2-D and 1-D coordinates for the computation of the gradient of γ_a in 3-D. On the left the triangle is completely contained in the sphere of interaction of particle a, but not on the right where the clipped quantities differ from the non-clipped ones. In this sketch l_{ev_0} is higher than 0 whereas l_{ev_2} is lower than zero.

by [150]:

$$\boldsymbol{\nabla}\gamma_{as} = \frac{1}{h}\boldsymbol{n}_s\left[2\pi\delta_{\sigma s}W(q^c_{a\sigma}) + \sum_{e\in s}\left(F_{3D}(q^c_{av_1}, q^c_{a\sigma}, q^c_{ae}, p^c_{\sigma e}, l^c_{ev_1}) - F_{3D}(q^c_{av_0}, q^c_{a\sigma}, q^c_{ae}, p^c_{\sigma e}, l^c_{ev_0})\right)\right] \tag{A.8}$$

with $W(q_{a\sigma})$ given by:

$$\frac{3}{16\pi}\left(1-\frac{q_{a\sigma}}{2}\right)^5\left(2+5q_{a\sigma}+4q^2_{a\sigma}\right) \tag{A.9}$$

In equation (A.8), the three-dimensional distances q are always positive, but the 2-D distance $p_{\sigma e}$ is algebraic and may be negative: $p_{\sigma e} = \boldsymbol{n}_e \cdot \boldsymbol{r}_{av_1}$ where $\boldsymbol{n}_e$ is the normal to edge e, oriented outwards. On the other hand, F_{3D} is the integral of the second line of (A.6) and is given by:

$$\begin{aligned}
F_{3D}\quad (q_{av}, q_{a\sigma}, q_{ae}, p_{\sigma e}, l_{ev}) = \frac{1}{4096\pi}\,(&\\
&-24(64+7q^2_{a\sigma}(-16+5q^2_{a\sigma}(4+q^2_{a\sigma})))\arctan\frac{l_{ev}}{p_{\sigma e}}\\
&+96q^5_{a\sigma}(28+q^2_{a\sigma})\arctan\frac{q_{a\sigma}l_{ev}}{qp_{\sigma e}}\\
&+2l_{ev}p_{\sigma e}\left[\begin{array}{l}
3q^4_{a\sigma}(-420+29q^2)+p^4_{\sigma e}(-420+33q)\\
+2q^2_{a\sigma}(-210(8+q^2)+756q+19l^2_{ev}q)\\
+4(336+l^4_{ev}(-21+2q)+28p^2_{av}(-5+3q))\\
+2p^2_{\sigma e}(420(-2+q)+6q^2_{a\sigma}(-105+8q)+l^2_{ev}(-140+13q))
\end{array}\right]\\
&+6\mathrm{sgn}(l_{ev})(5p^6_{\sigma e}+21p^4_{\sigma e}(8+q^2_{a\sigma})+35p^2_{\sigma e}q^2_{a\sigma}(13+q^2_{a\sigma})+35q^4_{a\sigma}(24+q^2_{a\sigma}))\mathrm{acosh}\frac{l_{ev}}{q_{ae}}\\
&)
\end{aligned} \tag{A.10}$$

Equation (A.8) already makes it possible to compute $\boldsymbol{\nabla}\gamma_a$, but there is an issue with this formula when implementing it in a code. Indeed, $\delta_{\sigma s}W(q_{a\sigma})$ and F_{3D} are discontinuous functions, which is a problem when it comes to the numerical computation of $\boldsymbol{\nabla}\gamma_{as}$. The discontinuity in $W(q_{a\sigma})$

comes from the $\delta_{\sigma s}$ term, which passes from 0 to 1 when the projection of a on the segment moves outside of s.

The discontinuities in F_{3D} stem from the two arctan functions in the first two lines of (A.10). Indeed, arctan passes from $\frac{\pi}{2}$ to $-\frac{\pi}{2}$ when its argument changes sign. Though, it is possible to show that the discontinuities of the arctan terms of F_{3D} and of $\delta_{\sigma s}W(q_{a\sigma})$ compensate each other and can thus be cleverly rearranged so as to write $\boldsymbol{\nabla}\gamma_{as}$ as a continuous function. Here it is considered that the vertices are always positioned counterclockwise when $\boldsymbol{n}_s$ is oriented towards us, which determines the signs of the arctan. First, let us note that $2\pi\delta_{\sigma s}W(q^c_{a\sigma})$ can be re-written as:

$$2\pi\delta_{\sigma s}W(q^c_{a\sigma}) = W(q^c_{a\sigma})\sum_{e\in s}\left(\arctan\frac{l_{ev_1}}{p_{\sigma e}} - \arctan\frac{l_{ev_0}}{p_{\sigma e}}\right) \tag{A.11}$$

Then, writing $F_{3D} = F_{3D} - W(q_{a\sigma})\arctan\frac{l_{ev}}{p_{\sigma e}} + W(q_{a\sigma})\arctan\frac{l_{ev}}{p_{\sigma e}}$ makes it possible to rearrange (A.10). After factorisation, this yields:

$$\begin{aligned}
F_{3D}\quad (q_{av}, q_{a\sigma}, q_{ae}, p_{\sigma e}, l_{ev}) = &-W(q_{a\sigma})\arctan\frac{l_{ev}}{p_{\sigma e}} + \frac{1}{4096\pi}\,(\\
&+96q^5_{a\sigma}(28+q^2_{a\sigma})\left(\arctan\frac{q_{a\sigma}l_{ev}}{qp_{\sigma e}} - \arctan\frac{l_{ev}}{p_{\sigma e}}\right)\\
&+2l_{ev}p_{\sigma e}\left[\begin{array}{l}
3q^4_{a\sigma}(-420+29q^2) + p^4_{\sigma e}(-420+33q)\\
+2q^2_{a\sigma}(-210(8+q^2)+756q+19l^2_{ev}q)\\
+4(336+l^4_{ev}(-21+2q)+28p^2_{av}(-5+3q))\\
+2p^2_{\sigma e}(420(-2+q)+6q^2_{a\sigma}(-105+8q)+l^2_{ev}(-140+13q))
\end{array}\right]\\
&+6\mathrm{sgn}(l_{ev})(5p^6_{\sigma e}+21p^4_{\sigma e}(8+q^2_{a\sigma})+35p^2_{\sigma e}q^2_{a\sigma}(13+q^2_{a\sigma})+35q^4_{a\sigma}(24+q^2_{a\sigma}))\mathrm{acosh}\frac{l_{ev}}{q_{ae}}\\
&)
\end{aligned} \tag{A.12}$$

Now, the difference of the arctan terms in the second line of (A.12) is continuous because $\frac{l_{ev}}{p_{\sigma e}}$ and $\frac{q_{a\sigma}l_{ev}}{qp_{\sigma e}}$ always change signs together so their difference is zero. The only remaining discontinuities comes from the first line of (A.12) and from the first line of (A.8). Indeed, $\mathrm{acosh}\frac{l_{ev}}{q_{ae}}$ tends to infinity when q_{ae} tends to zero but that the last line of (A.10) is in fact continuously expendable in zero since it tends to zero as $x\log x$ when x tends to zero. Thus, a clip inside the acosh resolves this problem.

Thus, let us write F_{3D} as:

$$F_{3D}(q_{av}, q_{a\sigma}, q_{ae}, p_{\sigma e}, l_{ev}) = -W(q_{a\sigma})\arctan\left[\frac{l_{ev}}{p_{\sigma e}}\right] + H_{C0}(q_{av}, q_{a\sigma}, q_{ae}, p_{\sigma e}, l_{ev}) \tag{A.13}$$

with H_{C0} a continuous function defined by (A.12) (except the first term). Now, (A.8) can be rewritten as:

$$\boldsymbol{\nabla}\gamma_{as} = \frac{1}{h}\boldsymbol{n}_s \left[\begin{array}{l} W(q^c_{a\sigma}) \displaystyle\sum_{e\in s} \left[\begin{array}{l} \arctan\left[\dfrac{l_{ev_1}}{p_{\sigma e}}\right] - \arctan\left[\dfrac{l^c_{ev_1}}{p^c_{\sigma e}}\right] \\ -\arctan\left[\dfrac{l_{ev_0}}{p_{\sigma e}}\right] - \arctan\left[\dfrac{l^c_{ev_0}}{p^c_{\sigma e}}\right] \end{array} \right] \\ + \displaystyle\sum_{e\in s} \left(H_{C0}(q^c_{av_1}, q^c_{a\sigma}, q^c_{ae}, p^c_{\sigma e}, l^c_{ev_1}) - H_{C0}(q^c_{av_0}, q^c_{a\sigma}, q^c_{ae}, p^c_{\sigma e}, l^c_{ev_0}) \right) \end{array} \right] \tag{A.14}$$

All the terms of equation (A.14) are now continuous. This formula was used for the computation of $\boldsymbol{\nabla}\gamma_a$ in the 3-D cases presented in Chapter 5.

Bibliography

[1] Abe, H., Kawamura, H., and Matsuo, Y. Direct numerical simulation of a fully developed turbulent channel flow with respect to Reynolds number dependence. *ASME Journal of Fluids Engineering 123*, 2 (2001), 382–393.

[2] Abramowitz, M., and Stegun, I. A. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover Publications, New-York, 1972. (10th edition).

[3] Altomare, C., Crespo, A. J. C., Rogers, B. D., Dominguez, J. M., Gironella, X., and Gómez-Gesteira, M. Numerical modelling of armour block sea breakwater with smoothed particle hydrodynamics. *Computers & Structures 130* (2014), 34–45.

[4] Amada, T., Imura, M., Yasumuro, Y., Manabe, Y., and Chihara, K. Particle-based fluid simulation on the GPU. In *Proc. ACM Workshop on General-purpose Computing on Graphics Processors* (2003).

[5] Amicarelli, A., Marongiu, J.-C., Leboeuf, F., Leduc, J., and Caro, J. SPH truncation error in estimating a 3D function. *Computers & Fluids 44*, 1 (2011), 279–296.

[6] Antuono, M., Colagrossi, A., and Marrone, S. Numerical diffusive terms in weakly-compressible SPH schemes. *Computer Physics Communications 183*, 12 (2012), 2570–2580.

[7] Antuono, M., Colagrossi, A., and Marrone, S. On the use of numerical diffusive terms in weakly-compressible SPH schemes. In *Proc. 7th international SPHERIC workshop* (2012), pp. 200–207.

[8] Archambeau, F., Méchitoua, N., and Sakiz, M. *Code_Saturne*: a Finite Volume code for the computation of turbulent incompressible flows - industrial applications. *International Journal on Finite Volumes 1* (2004), 1–62.

[9] Aris, R. *Vectors, Tensors and the Basic Equations of Fluid Mechanics.* Prentice-Hall Inc.: Engelwood Cliffs, NJ, 1962.

[10] Basa, M., Quinlan, N. J., and Lastiwka, M. Robustness and accuracy of SPH formulations for viscous flow. *International Journal for Numerical Methods in Fluids 60*, 10 (2009), 1127–1148.

[11] Beyer, W. H. *CRC Standard Mathematical Tables, 28th ed.* Boca Raton, FL: CRC Press, 1987.

[12] Bonet, J., and Lok, T. S. L. Variational and momentum preservation aspects of smoothed particle hydrodynamics formulations. *Computer Methods in applied mechanics and engineering 180* (1999), 97–115.

[13] Bredberg, J. On the wall boundary condition for turbulence models. Tech. rep., Chalmers University of Technology, Göteborg, Sweden, 2000.

[14] Chmielewski, M., and Gieras, M. Three-zonal wall function for k-epsilon turbulence models. *Computational Methods in Science and Technology 19*, 2 (2013), 107–114.

[15] Chorin, A. J. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation 22* (1968), 745–762.

[16] Colagrossi, A., Antuono, M., Souto-Iglesias, A., and Le Touzé, D. Theoretical analysis and numerical verification of the consistency of viscous smoothed-particle-hydrodynamics formulations in simulating free-surface flows. *Physical Review E 84*, 2 (2011).

[17] Connolly, A., and Iannucci, L. A modified Godunov SPH method for materials with strength. In *Proc. 7th international SPHERIC workshop* (2012).

[18] Crespo, A. J. C., Dominguez, J. M., Barreiro, A., Gómez-Gesteira, M., and Rogers, B. D. GPUs, a new tool of acceleration in CFD: Efficiency and reliability on smoothed particle hydrodynamics methods. *PLoS ONE 6*, 6 (2011), e20685.

[19] Cummins, S. J., and Rudman, M. An SPH projection method. *Journal of Computational Physics 152*, 2 (1999), 584–607.

[20] Dalrymple, R. A., and Rogers, B. D. Numerical modeling of water waves with the SPH method. *Coastal Engineering 53*, 2-3 (2006), 141–147.

[21] De Leffe, M. *Modélisation d'écoulements visqueux par méthode SPH en vue d'application à l'hydrodynamique navale.* PhD thesis, Ecole Centrale de Nantes, 2011. (in French).

[22] De Leffe, M., Le Touzé, D., and Alessandrini, B. Normal flux method at the boundary for SPH. In *Proc. 4th international SPHERIC workshop* (2009), pp. 149–156.

[23] De Padova, D., Mossa, M., Sibilla, S., and Torti, E. 3D SPH modelling of hydraulic jump in a very large channel. *Journal of Hydraulic Research 51*, 2 (2013), 158–173.

[24] Dean, R. G., and Dalrymple, R. A. *Water wave mechanics for engineers and scientists, volume 2 of Advanced Series on Ocean Engineering.* World Scientific, Singapore, 1991.

[25] Dehnen, W., and Aly, H. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. *Monthly Notices of the Royal Astronomical Society 0* (2012), 1–15.

[26] Domínguez, J. M., Crespo, A. J. C., Gómez-Gesteira, M., and Rogers, B. D. Simulating more than 1 billion sph particles using gpu hardware acceleration. In *Proc. 8th international SPHERIC workshop* (2013).

[27] Domínguez, J. M., Crespo, A. J. C., Valdez-Balderas, D., Rogers, B. D., and Gómez-Gesteira, M. New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters. *Computer Physics Communications 184*, 8 (2013), 1848–1860.

[28] EDF R&D. *Code_Saturne* 3.0.0 theory guide. http://code-saturne.org/cms/sites/default/files/theory-3.0.pdf. Accessed: 2013-10-16.

[29] EDF R&D. Code_Saturne 3.0.0 Theory Guide, 2013. http://code-saturne.org/cms/sites/default/files/theory-3.0.pdf.

[30] Ellero, M., and Adams, N. A. SPH simulations of flow around a periodic array of cylinders confined in a channel. *International Journal for Numerical Methods in Fluids 86* (2011), 1027–1040.

[31] Farahani, R. J., Dalrymple, R. A., Hérault, A., and Bilotta, G. Turbulent coherent structures under breaking water waves. In *Proc. 7th international SPHERIC workshop* (2012).

[32] Fatehi, R., and Manzari, F. T. Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. *Computers & Mathematics with Applications 61*, 2 (2011), 482–498.

[33] Feldman, J., and Bonet, J. Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems. *International Journal for Numerical Methods in Engineering 72* (2007), 295–324.

[34] Ferrand, M., an A. Joly, C. K., Violeau, D., Leroy, A., Morel, F.-X., and Rogers, B. D. Semi-analytical conditions and riemann solver for open boundaries in sph. *Journal of Computational Physics* (2015). to be submitted.

[35] Ferrand, M., Laurence, D. R., Rogers, B. D., Violeau, D., and Kassiotis, C. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. *International Journal for Numerical Methods in Fluids 71* (2013), 446–472.

[36] Ferrari, A., Dumbser, M., Toro, E. F., and Armanini, A. A new 3D parallel SPH scheme for free surface flows. *Computers & Fluids 36*, 6 (2009), 1203–1217.

[37] Ghasemi, V. A., Firoozabadi, B., and Mahdinia, M. 2D numerical simulation of density currents using the SPH projection method. *European Journal of Mechanics - B/Fluids 38* (2013), 38–46.

[38] Ghia, U., Ghia, K. N., and Shin, C. T. High-Re solutions for incompressible flow using the Navier-Stokes equations and multigrid method. *Journal of Computational Physics 48* (1982), 387–411.

[39] Giles, M. B. Nonreflecting boundary conditions for Euler equation calculations. *AIAA Journal 28*, 12 (1990), 2050–2058.

[40] Guermond, J. L., Minev, P., and Shen, J. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering 195*, 44-47 (2006), 6011–6045.

[41] Guimet, V., and Laurence, D. A linearised turbulent production in the k-ϵ model for engineering applications. In *Proc. Vth International Symposium on Engineering Turbulence Modelling and Measurements* (2002), pp. 157–166. Majorqua (Spain).

[42] Harada, T., Koshizuka, S., and Kawaguchi, Y. Smoothed particle hydrodynamics on GPUs. In *Proc. Comput. Graph. Intl.* (2007), pp. 63–70.

[43] Harlow, F. H., and Welch, J. E. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids 8* (1965), 2182–2189.

[44] Härtel, C., Meiburg, E., and Necker, F. Analysis and direct numerical simulation of the flow at a gravity-current head. Part 1. Flow topology and front speed for slip and no-slip boundaries. *Journal of Fluid Mechanics 418* (2000), 189–212.

[45] Hérault, A., Bilotta, G., and Dalrymple, R. A. SPH on GPU with CUDA. *Journal of Hydraulic Research 48* (2010), 74–79.

[46] Hérault, A., Bilotta, G., and Dalrymple, R. A. Achieving the best accuracy in an SPH implementation. In *Proc. 9th international SPHERIC workshop* (2014).

[47] Hérault, A., Vicari, A., and Del Negro, C. A SPH thermal model for the cooling of a lava lake. In *Proc. 3rd international SPHERIC workshop* (2008), pp. 143–148.

[48] Hérault, A., Vicari, A., Negro, C. D., Bilotta, G., and Rustico, E. GPU-LAVA: SPH lava flow simulation on CUDA. In *Proc. 6th international SPHERIC workshop* (2011).

[49] Hervouet, J.-M. *Hydrodynamics of Free Surface Flows: Modelling with the Finite Element Method.* John Wiley, 2007.

[50] Hockney, R. W., and Eastwood, J. W. *Computer simulation using particles.* McGraw-Hill, New-York, 1981.

[51] Hosseini, S. M., and Feng, J. J. Pressure boundary conditions for computing incompressible flows with SPH. *Journal of Computational Physics 230* (2011), 7473–7487.

[52] Hu, X. Y., and Adams, N. A. An incompressible multi-phase SPH method. *Journal of Computational Physics 227* (2007), 264–278.

[53] Issa, R., Moulinec, C., Latino, D., Violeau, D., Biddiscombe, J., and Thibaud, G. Modelling a plunging breaking solitary wave with eddy-viscosity turbulent SPH models. In *Proc. 3rd international SPHERIC workshop* (2008).

[54] Issa, R., and Violeau, D. Modelling a plunging breaking solitary wave with eddy-viscosity turbulent SPH models. *Computers, Materials, & Continua 8*, 3 (2008), 151–164.

[55] Issa, R., and Violeau, D. Spheric test-case # 2, 2014.

[56] Issa, R., Violeau, D., Lee, E.-S., and Flament, H. *Modelling nonlinear water waves with RANS and LES SPH models, in Advances in Numerical Simulation of Nonlinear Water Waves*, vol. 11. Editor, Q. W. Ma and World Scientific Publishing Co, 2010, ch. 14.

[57] Johnson, G. R., Stryk, R. A., and Beissel, S. R. Sph for high velocity impact computations. *Computational Methods for Applied Mechanics and Engineering 139* (1996), 347–373.

[58] Kasagi, N., and Iida, O. Progress in direct numerical simulation of turbulent heat transfer. In *Proc. 5th ASME/JSME Joint Thermal Engineering Conference* (San Diego, California, 1999).

[59] Kassiotis, C., Ferrand, M., Violeau, D., Rogers, B. D., Stansby, P. K., and Benoit, M. Coupling SPH with a 1-D Boussinesq-type wave model. In *Proc. 6th international SPHERIC workshop* (2011), pp. 241–247.

[60] Kassiotis, C., Violeau, D., and Ferrand, M. Semi-analytical conditions for open boundaries in smoothed particle hydrodynamics. In *Proc. 8th international SPHERIC workshop* (2013).

[61] Kawamura, H., Abe, H., and Shingai, K. DNS of turbulence and heat transport in a channel flow with different Reynolds and Prandtl numbers and boundary conditions. In *Turbulence, Heat and Mass Transfer 3 (Proc. of the 3rd International Symposium on Turbulence, Heat and Mass Transfer)* (2000), Y. Nagano and K. Hanjalic and T. Tsuji, pp. 15–32.

[62] Kolb, A., and Cuntz, N. Dynamic particle coupling for GPU-based fluid simulation. In *Proc. 18th Symposium on Simulation Technique* (2005), pp. 722–727.

[63] Kolmogorov, A. N. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 434*, 1890 (1991), 9–13.

[64] Koshizura, S., Nobe, A., and Oka, Y. Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids 26* (1998), 751–769.

[65] Kulasegaram, S., Bonet, J., Lewis, R. W., and Profit, M. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. *Computational Mechanics 33* (2004), 316–325.

[66] Kuzmin, D., Mierka, O., and Turek, S. *On the Implementation of the k-epsilon Turbulence Model in Incompressible Flow Solvers Based on a Finite Element Discretization.* Ergebnisberichte angewandte Mathematik. Tech. University of Dortmund, 2007.

[67] Lastiwka, M., Basa, M., and Quinlan, N. J. Permeable and non-reflecting boundary conditions in SPH. *International Journal for Numerical Methods in Fluids 61*, 7 (2009), 709–724.

[68] Launder, B. E., Reece, G. J., and Rodi, W. Progress in the development of a reynolds stress turbulent closure. *Journal of Fluid Mechanics 68* (1975), 537–566.

[69] Launder, B. E., and Spalding, D. B. *Mathematical models of turbulence.* London: Academic Press, 1972.

[70] Launder, B. E., and Spalding, D. B. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering 3* (1974), 269–289.

[71] Lee, E.-S. *Truly incompressible approach for computing incompressible flows in SPH and comparisons with the traditional weakly compressible approach.* PhD thesis, University of Manchester, Manchester, UK, 2007.

[72] Lee, E.-S., Moulinec, C., Xu, R., Violeau, D., Laurence, D., and Stansby, P. K. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. *Journal of Computational Physics 227* (2008), 8417–8436.

[73] Lee, E.-S., Violeau, D., Issa, R., and Ploix, S. Application of weakly compressible and truly incompressible SPH to 3-D water collapse in waterworks. *Journal of Hydraulic Research 48* (2008), 50–60.

[74] Lee, E.-S., Violeau, D., Issa, R., Ploix, S., and Marc, R. Simulating a real dam spillway flow with 3-D SPH. In *Proc. 4th international SPHERIC workshop* (2009), pp. 339–345.

[75] Libersky, L. D., Petschek, A. G., Carney, T. C., Hipp, J. R., and Allahdadi, F. A. High strain lagrangian hydrodynamics: a three-dimensional SPH code for dynamic material response. *Journal of Computational Physics 109*, 1 (1993), 67–75.

[76] Lind, S., Xu, R., Stansby, P. K., and Rogers, B. D. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics 231*, 4 (2012), 1499–1523.

[77] Lions, P.-L. *Mathematical Topics in Fluid Mechanics. Volume 1: Incompressible Models.* Clarendon Press, Oxford Lecture Series in Mathematics and Its Applications 3, 1996.

[78] Liu, A. W., Bornside, D. E., Armstrong, R. C., and Brown, R. A. Viscoelastic flow of polymer solutions around a periodic, linear array of cylinders: comparisons of predictions for microstructure and flow fields. *Journal of Non-Newtonian Fluid Mechanics 77* (1998), 153–190.

[79] Lo, E., and Shao, S. Simulation of near-shore solitary waves mechanics by an incompressible SPH method. *Applied Ocean Research 24* (2002), 275–286.

[80] Lucy, L. B. A numerical approach to testing the fission hypothesis. *The Astronomical Journal 82*, 12 (1977), 1013–1924.

[81] Macià, F., González, L. M., Cercos-Pita, J. L., and Souto-Iglesias, A. A boundary integral sph formulation - consistency and applications to isph and wcsph. *Progress of Theoretical Physics 128*, 3 (2012), 439–462.

[82] Mahmood, O., Violeau, D., Kassiotis, C., Rogers, B. D., and Ferrand, M. Absorbing inlet/outlet boundary conditions for 2D SPH turbulent free-surface flows. In *Proc. 7th international SPHERIC workshop* (2012), pp. 296–302.

[83] Marongiu, J.-C. *Méthode Numérique Lagrangienne pour la simulation d'écoulements à surface libre – Application aux turbines Pelton.* PhD thesis, Ecole Centrale de Lyon, 2007. (in French).

[84] Marongiu, J.-C., Parkinson, E., Lais, S., Leboeuf, F., and Leduc, J. Application of SPH-ALE method to pelton hydraulic turbines. In *Proc. 5th international SPHERIC workshop* (2010).

[85] Marrone, S., Antuono, M., Colagrossi, A., Colicchio, G., Le Touzé, D., and Graziani, G. δ-SPH model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering 200*, 13-16 (2011), 1526–1542.

[86] Marrone, S., Colagrossi, A., Le Touzé, D., and Graziani, G. Fast free-surface detection and level-set function definition in SPH solvers. *Journal of Computational Physics 229*, 10 (2010), 3652–3663.

[87] Marsden, J. E., and West, M. Discrete mechanics and variational integrators. *Acta Numerica 10*, 1 (2001), 357–514.

[88] Maruzewski, P., Le Touzé, D., Oger, G., and Avellan, F. SPH high-performance computing simulations of rigid solids impacting the free-surface of water. *Journal of Hydraulic Research 48* (2010), 126–134.

[89] Matteo, T. D., Springel, V., and Hernquist, L. Energy input from quasars regulates the growth and activity of black holes and their host galaxies. *Nature 433* (2005), 604–607.

[90] Mayrhofer, A. Large eddy simulation with smoothed particle hydrodynamics: Part 1: Improvement and extension of the unified semi-analytical wall boundary conditions. Tech. rep., University of Manchester, 2011.

[91] Mayrhofer, A. *Large Eddy Simulation with Smoothed Particle Hydrodynamics: an investigation into wall boundary conditions and turbulent flows.* PhD thesis, University of Manchester, Manchester, UK, 2013.

[92] Mayrhofer, A., Ferrand, M., Kassiotis, C., Violeau, D., and Morel, F.-X. Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3-d. *Numerical Algorithms* (2014).

[93] Mayrhofer, A., Laurence, D., Rogers, B. D., Violeau, D., and Ferrand, M. Direct numerical simulation of 3-D turbulent wall bounded flows with SPH. In *Proc. 8th international SPHERIC workshop* (2013), pp. 130–138.

[94] Mayrhofer, A., Rogers, B. D., Violeau, D., and Ferrand, M. Study of differential operators in the context of the semi-analytical wall boundary conditions. In *Proc. 7th international SPHERIC workshop* (2012), pp. 149–156.

[95] Mayrhofer, A., Rogers, B. D., Violeau, D., and Ferrand, M. Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions. *Computer Physics Communications 184*, 11 (2013), 2515–2527.

[96] Menter, F. R. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal 32*, 8 (1994), 1598–1605.

[97] Meyer, M., Devesa, A., Hickel, S., Hu, X. Y., and Adams, N. A. A conservative immersed interface method for large-eddy simulation of incompressible flows. *Journal of Computational Physics 229* (2010), 6300–6317.

[98] Meyer, M., Hickel, S., and Adams, N. A. Assessment of implicit large-eddy simulation with a conservative immersed interface method for turbulent cylinder flow. *International Journal of Heat and Fluid Flow 31* (2010), 368–377.

[99] Mohammadi, B., and Pironneau, O. *Analysis of the k-epsilon turbulence model.* France: Editions MASSON, 1993.

[100] Monaghan, J. SPH without a tensile instability. *Journal of Computational Physics 159*, 2 (2000), 290–311.

[101] Monaghan, J., and Gingold, R. Shock simulation by the particle methos SPH. *Journal of Computational Physics 52*, 2 (1983), 374–389.

[102] Monaghan, J. J. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics 30* (1992), 543–574.

[103] Monaghan, J. J. Smoothed particle hydrodynamics. *Reports on Progress in Physics 68* (2005), 1703–1759.

[104] Morris, J. P., Fox, P. J., and Zhu, Y. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics 136*, 1 (1997), 214–226.

[105] Moulinec, C., Issa, R., Marongiu, J.-C., and Violeau, D. Parallel 3-D SPH simulations. *Computing Modeling Engineering Science 25*, 3 (2008), 133–148.

[106] Moulinec, C., Issa, R., Violeau, D., Marongiu, J.-C., and Leboeuf, F. Parallel 3-D SPH simulations over periodic hills. In *Proc. Int. Conf. Comput. and Experimental Eng. and Sciences, IInd symposium on meshless methods* (2007).

[107] Nezu, I., and Nakawaga, H. Turbulence in open channel. In *IAHR Monograph* (1993), Balkema, Ed.

[108] Oger, G., Doring, M., Alessandrini, B., and Ferrant, P. Two-dimensional SPH simulations of wedge water entries. *Journal of Computational Physics 213*, 2 (2006), 803–822.

[109] Oger, G., Doring, M., Alessandrini, B., and Ferrant, P. An improved SPH method: Towards higher order convergence. *Journal of Computational Physics 225*, 2 (2007), 1472–1492.

[110] Oger, G., Leroy, C., Jacquin, E., Le Touzé, D., and Alessandrini, B. Specific pre/post treatments for 3-D SPH applications through massive HPC simulations. In *Proc. 4th international SPHERIC workshop* (2009), pp. 27–29.

[111] Orlanski, I. A simple boundary condition for unbounded hyperbolic flows. *Journal of Computational Physics 21* (1976), 251–269.

[112] Peng, Y.-F., Shiau, Y.-H., and Hwang, R. R. Transition in a 2-D lid-driven cavity flow. *Computers & Fluids 32* (2003), 337–352.

[113] Pope, S. B. A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics 72* (1975), 331–340.

[114] Pope, S. B. *Turbulent flows*. Cambridge: Cambridge University Press, 2000.

[115] Price, D. J. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics 231* (2012), 759–794.

[116] Quinlan, G. D., and Tremaine, S. Symmetric multistep methods for the numerical integration of planetary orbits. *The Astronomical Journal 100* (1990), 1694–1700.

[117] Quinlan, N. J., Basa, M., and Lastiwka, M. Truncation error in mesh-free particle methods. *International Journal for Numerical Methods in Engineering 66* (2006), 2064–2085.

[118] Randles, P. W., and Libersky, L. D. Smoothed particle hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering 139*, 1–4 (1996), 375–408.

[119] Rannacher, R. On chorin's projection method for the incompressible navier-stokes equations. In *The Navier-Stokes Equations II — Theory and Numerical Methods*, vol. 1530 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1992, pp. 167–183.

[120] Rodi, W. Turbulence models and their applications in hydraulics. In *IAHR monograph* (Brookfield, Rotterdam, 2000).

[121] Rogers, B. D., and Dalrymple, R. A. SPH modeling of tsunami waves. *Advanced numerical models for simulating tsunami waves and runup 10* (2008), 75–100.

[122] Rogers, B. D., Dalrymple, R. A., Stansby, P. K., and Laurence, D. Development of a parallel sph code for free-surface wave hydrodynamics. In *Proc. 2nd SPHERIC International Workshop* (2007), pp. 111–114.

[123] Russell, P. A., and Abdallah, S. Dilation-free solutions for the incompressible flow equations on nonstaggered grids. *AIAA Journal 35*, 3 (1997), 585–586.

[124] Rustico, E., Hérault, A., and Bilotta, G. Multi-GPU, multi-node SPH implementation with arbitrary domain decomposition. In *Proc. 9th SPHERIC International Workshop* (2014), pp. 127–133.

[125] Saad, Y., and Schultz, M. H. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *Journal on Scientific and Statistical Computing 7*, 3 (1986), 856–869.

[126] Sani, R. L., Gresho, P. M., Lee, R. L., and Griffiths, D. F. The cause and cure (?) of the spurious pressures generated by certain FEM solutions of the incompressible navier-stokes equations: Part 1. *International Journal for Numerical Methods in Fluids 1*, 1 (1981), 17–43.

[127] Schwaiger, H. F. An implicit corrected SPH formulation for thermal diffusion with linear free surface boundary conditions. *International Journal for Numerical Methods in Engineering 75*, 6 (2008), 647–671.

[128] Schwartz, L. *Théorie des distributions 1-2*. Hermann, Paris, 1950-1951. (in French).

[129] Shao, S., and Lo, E. Y. M. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free-surface. *Advanced Water Ressources 26* (2003), 787–800.

[130] Shepard, D. A two dimensional function for irregularly spaced data. In *ACM National Conference* (1968).

[131] Smagorinsky, J. General circulation experiments with the primitive equations i. the basic experiment. *Monthly Weather Review 91*, 3 (1963), 99–164.

[132] Souto-Iglesias, A., Macià, F., González, L. M., and Cercos-Pita, J. L. On the consistency of MPS. *Computer Physics Communications 184*, 3 (2013), 732–745.

[133] Springel, V. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society 364*, 4 (2005), 1105–1134.

[134] Szewc, K., Pozorski, J., and Tanière, A. Modeling of natural convection with Smoothed Particle Hydrodynamics: Non-Boussinesq formulation. *International Journal of Heat and Mass Transfer 54*, 23-24 (2011), 4807–4816.

[135] Tait, P. G. Report on some of the physical properties of fresh water and sea water. *Rept. Sci. Results Voy. H.M.S. Challenger, Phys. Chem. 2* (1888), 1–76.

[136] Takeda, H., Miyama, S. M., and Sekiya, M. Numerical simulation of viscous flow by Smoothed Particle Hydrodynamics. *Progress of Theoretical Physics 92*, 5 (1994), 939–960.

[137] Tarrade, L., Texier, A., David, L., Pineau, G., and Larinier, M. Experimental approach to adapt the turbulent flow in the vertical slot fishways to the small fish species. *Journal Hydrobiologia 1* (2008), 177–188.

[138] Temam, R. Une méthode d'approximation des solutions des équations de Navier–Stokes. *Bulletin de la Société Mathématique de France 98* (1968), 115–152. (in French).

[139] Temam, R. Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires II. *Archive for Rational Mechanics and Analysis 33* (1969), 377–385. (in French).

[140] The OpenFOAM Foundation, 2013. http://www.openfoam.org/index.php.

[141] Trask, N., Maxey, M., Yang, K., Hu, X. Y., and Xu, J. Accuracy and performance of implicit projection methods for transient viscous flows using SPH. In *Proc. 8th international SPHERIC workshop* (2013).

[142] Trias, F. X., Gorobets, A., Soria, M., and Oliva, A. Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Rayleigh numbers up to 1011 – Part I: Numerical methods and time-averaged flow. *International Journal of Heat and Mass Transfer 53*, 4 (2010), 665–673.

[143] Vacondio, R., Rogers, B. D., Stansby, P. K., and Mignosa, P. SPH modeling of shallow flow with open boundaries for practical flood simulation. *Journal of Hydraulic Engineering 138*, 6 (2012), 530–541.

[144] Vacondio, R., Rogers, B. D., Stansby, P. K., and Mignosa, P. Shallow water sph for flooding with dynamic particle coalescing and splitting. *Advances in Water Resources 58* (2013), 10–23.

[145] Vila, J. P. On particle weighted methods and smooth particle hydrodynamics. *Mathematical Models and Methods in Applied Sciences 9*, 2 (1999), 161–209.

[146] Violeau, D. Dissipative forces for Lagrangian models in computational fluid dynamics and application to Smoothed Particle Hydrodynamics. *Physical Review E 80*, 3 (2009).

[147] Violeau, D. *Fluid Mechanics and the SPH method.* Oxford University Press, 2012.

[148] Violeau, D., and Issa, R. Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview. *International Journal for Numerical Methods in Fluids 53* (2007), 277–304.

[149] Violeau, D., and Leroy, A. On the maximum time step in weakly compressible SPH. *Journal of Computational Physics 256* (2014), 388–415.

[150] Violeau, D., Leroy, A., and Mayrhofer, A. Exact computation of SPH wall renormalising integrals in 3-D. In *Proc. 9th international SPHERIC workshop* (2014).

[151] Viollet, P.-L. *Mécanique des fluides à masse volumique variable.* Presses de l'Ecole Nationale des Ponts et Chaussées, 1997. (in French).

[152] Viollet, P.-L., Chabard, J.-P., Esposito, P., and Laurence, D. *Mécanique des fluides appliquée.* Presses de l'Ecole Nationale des Ponts et Chaussées, 2002. (in French).

[153] Vorst, H. A. V. D. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *Journal on Scientific and Statistical Computing 13*, 2 (1992), 631–644.

[154] Wan, D. C., Patnaik, B. S. V., and Wei, G. W. A new benchmark quality solution for the buoyancy-driven cavity by discrete singular convolution. *Numerical Heat Transfer, Part B: Fundamentals 40*, 3 (2001), 199–228.

[155] Xu, R., Stansby, P. K., and Laurence, D. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of Computational Physics 228*, 18 (2009), 6703–6725.

[156] Yap, C. J. *Turbulent heat and momentum transfer in recirculating and impringing flows.* PhD thesis, University of Manchester, Manchester, UK, 1987.

[157] Yildiz, M., Rook, R. A., and Suleman, A. SPH with the multiple boundary tangent method. *International Journal for Numerical Methods in Engineering 77* (2009), 1416–1438.

[158] Zhao, J. *Development of a fast SPH model for non linear shallow water flows: application to coastal flooding and dam breaking.* PhD thesis, Ecole Centrale de Nantes, 2012.