

Application of the SPH method in turbulent free-surface flow for simulation of debris accumulation during flood events

Auteur : Santoro, Luca

Promoteur(s) : Archambeau, Pierre

Faculté : Faculté des Sciences appliquées

Diplôme : Master en ingénieur civil physicien, à finalité approfondie

Année académique : 2024-2025

URI/URL : <https://github.com/Rivlow/Master-thesis-free-surface-turbulent-flow-SPH>; <http://hdl.handle.net/2268.2>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.



Academic year 2024-2025

University of Liège - Faculty of applied-sciences

Application of the SPH method in turbulent free-surface flow for simulation of debris accumulation during flood events

Master thesis submitted in fulfilment of the requirements for the
degree of Master in Engineering Physics by Luca Santoro

Author

SANTORO Luca

Supervisors

Prof. Ass.

P. ARCHAMBEAU

Prof. Ass. S. ERPICUM

Jury

Prof. Ass. R. BOMAN

Dr. Ir. L. GOFFIN

Aknowledgment

Rédiger un mémoire n'est jamais simple et demande beaucoup d'énergie, de motivation et de réflexion. Mais en dehors de toutes ces considérations, c'est surtout grâce à mon entourage que j'ai pu arriver jusqu'au bout. C'est pourquoi je tiens à vous remercier tous.

Tout d'abord, merci au Professeur Archambeau qui m'a épaulé tout au long de cette aventure. Il s'est toujours montré disponible et avenant à n'importe quel moment. Nous avons souvent tendance à considérer ces choses comme normales ou acquises mais c'est en réalité très rarement le cas. Je tiens à souligner son point de vue pragmatique et logique des choses qui m'a été d'une aide cruciale dans les moments de doute ou d'incompréhension. Mais ce que je retiendrai surtout, ce sont les bons moments passés pendant les "réunions de quelques minutes maximum".

Ensuite, je tiens à remercier ma famille et en particulier mes parents et mon frère Thomas. Je suis devenu la personne d'aujourd'hui, pour le meilleur comme pour le pire, en majeure partie grâce à vous. Papa, Maman, vous m'avez toujours soutenu depuis le départ et je me rappelle très bien le chemin que nous avons parcouru ensemble, et je ne pourrai jamais assez vous remercier pour cela. Thomas, tu as toujours été un modèle pour moi jusqu'à présent. J'ai appris en avance sur mon âge au travers de tes réussites et difficultés, et tu peux considérer ce travail comme le résumé de ce que tu m'as transmis.

Finalement, j'aimerais remercier deux toutes dernières personnes. On ne choisit pas sa famille biologique, mais on choisit sa famille de coeur pour la vie. Louise et Anaïs, vous occupez une place toute particulière pour moi. Au cours de ces années, nous avons vécu tant de choses ensemble, certaines très chouettes, d'autres un peu moins, mais nous savons que peu importe ce que nous réserve la vie, nous pourrons toujours compter les uns sur les autres. Tous vos bons conseils et ces moments passés ensemble autour d'un verre à râler sur ce qui nous passe par la tête me sont très précieux, et c'est aussi votre présence qui fait la personne que je suis actuellement.

UNIVERSITÉ DE LIÈGE

Applied Sciences

Department of Hydraulics in environmental and civil engineering (HECE)

Master thesis submitted in fulfilment of the requirements for the degree of Master
in Engineering Physics

**Application of the SPH method in turbulent free-surface flow for simulation of
debris accumulation during flood events**

by Luca Santoro

Supervisors:

Prof. Ass. P. ARCHAMBEAU

Prof. Ass. S. ERPICUM

Jury members:

Prof. Ass. R. BOMAN

Dr. Ir. L. GOFFIN

Flood events represent a recurring natural hazard requiring accurate prediction of hydrodynamic forces on structures and debris accumulation patterns. This master thesis evaluates the capability of the Smoothed Particle Hydrodynamics (SPH) method, specifically through the SPLisHSPlasH software implementation, to simulate three-dimensional turbulent free-surface flows with dynamic rigid bodies.

The methodology employs a systematic validation approach through progressively complex configurations. First, turbulent flow behaviour is isolated and analysed in a two-dimensional pipe configuration with Reynolds number $Re = 1.7 \times 10^7$, where the numerical velocity profile correctly fits the analytical solution derived from Generalized Hydraulic Equations. Subsequently, free-surface flow characteristics are examined through a horizontal channel with Reynolds number $Re = 1.8 \times 10^5$ and a parabolic obstacle, correctly capturing hydraulic jump phenomena and head loss distributions despite local pressure instabilities inherent to the SPH formulation.

The final validation integrates both phenomena in a laboratory-scale bridge configuration with $Re = 1.5 \times 10^4$ and with seven floating wood logs. The simulation successfully reproduces obstacle formation at the bridge entrance, with distinct behaviours observed between pressurized and free-surface flow conditions. Quantitative analysis reveals accurate head loss predictions and flow distribution patterns, though with the requirement that particle size remains below one-fifth of the rigid body characteristic dimension to ensure numerical stability.

While certain limitations exist, particularly concerning local pressure calculations and the necessity for manual tuning of physical parameters, this study demonstrates that SPLisHSPlasH can effectively simulate complex hydraulic phenomena relevant to flood engineering applications. The continuous development of the software and capability to handle coupled fluid-structure interactions yield it as a valuable tool for hydraulic engineering analyses, encouraging more systematic adoption in the field.

1	Introduction and motivation	3
1.1	Introduction and motivation	3
1.2	Objectives of this master thesis	4
1.3	Methodology	4
2	Governing equations derivation	5
2.1	Mass conservation law	5
2.2	Linear momentum conservation law	6
2.3	Angular momentum conservation law	6
2.4	Energy conservation law	6
2.5	Constitutive equations: Newtonian fluids	7
2.6	Turbulence in fluid flows	8
2.6.1	Origins of turbulence in the Navier-Stokes equations	9
2.6.2	The turbulent cascade and scale separation	9
2.7	Rigid body dynamics in fluid flows	10
2.7.1	Equations of motion for rigid bodies	10
2.7.2	Hydrodynamic forces and moments	11
2.7.3	Kinematic constraints and boundary conditions	11
2.7.4	Rigidity constraints	12
2.7.5	Application for free surface flows	12
2.8	Complete coupled system	13
3	Mathematical fundamentals of the SPH method	15
3.1	Lagrangian and Eulerian specification	15
3.2	Meshless algorithm	16
3.3	Spatial SPH discretization	16
3.3.1	Approximated integral representation	16
3.3.2	Smoothing kernel conditions	17
4	SPlisHSPlasH software	19
4.1	Kernel used	20
4.2	Neighbourhood research	21
4.2.1	Cell-based methods	21

4.2.2	Z-index sort method	22
4.3	Pressure solver: divergence-free SPH method (DFSPH)	24
4.3.1	Mathematical foundations	24
4.3.2	Algorithm Structure	24
4.4	Time integration scheme	26
4.5	Viscosity implementation	27
4.6	Turbulence representation through vorticity : micropolar model	28
4.7	Air-fluid interaction (drag method)	28
4.7.1	Particle Deformation Model	29
4.7.2	Drag Coefficient Calculation	29
4.7.3	Cross-Sectional Area	30
4.7.4	Final computation	30
4.8	Boundary handling method : Volume Maps	31
4.8.1	Boundary representation	31
4.8.2	Intersection volume	31
4.8.3	Mapping of the intersected volumes	32
4.8.4	Final boundary contribution	32
4.8.5	Pressure force computation	34
4.8.6	Two-way coupling	34
4.9	XSPH numerical dissipation	34
4.9.1	Remark on the physical interpretation of XSPH	35
4.10	Dynamic bodies handling: position-based approach	35
4.10.1	Fluid-solid particles coupling	36
4.10.2	PBD general mathematical description	36
4.10.3	Projective constraint solution	37
4.10.4	Additional effects	38
5	Data treatment procedure	39
5.1	1D analysis (rectangle method)	39
5.1.1	Fixed rectangle	40
5.1.2	Gliding rectangle	40
5.2	2D analysis (grid method)	41
6	Preliminary validation tests	43
6.1	User warning	43
6.2	Two-dimensional turbulent flow analysis	44
6.2.1	Problem statement	44
6.2.2	Theoretical solution: generalized hydraulic equations	45
6.2.3	Numerical results	48
6.2.4	Limitations of the model in the turbulent simulation	55
6.3	Two-dimensional free surface horizontal flow analysis with parabolic obstacle	56
6.3.1	Problem statement	57
6.3.2	Theoretical solution: Bernoulli equations	57
6.3.3	Numerical results	60
6.3.4	Limitations of the model in the free surface simulation	68
7	Final validation test	71
7.1	Problem statement	71
7.1.1	Simulation parameters involved	73
7.1.2	Numerical results (pressurized bridge case: $h_0 = 200$ [mm])	73

7.1.3	Numerical results (free surface flow case: $h_0 = 100$ [mm])	78
7.1.4	Limitations of the two bridge configuration	82
8	Limitations encountered	83
8.1	Spatial resolution	83
8.2	XSPH Coefficient Dependencies	83
8.3	Energy Conservation Issues	84
8.4	Pressure Field Instabilities	84
8.5	Rigid Body Interaction Constraints	84
8.6	Boundary Effects and Domain Constraints	85
9	Conclusion	87
9.1	Complete Summary of This Thesis	87
9.1.1	Mathematical and Physical Framework	87
9.1.2	Fundamentals of the SPH Method	87
9.1.3	Preliminary Validation Tests	88
9.1.4	Final Validation Test	89
9.2	Main Conclusions About the SPlisHSPlasH Software	89
9.3	Future Perspectives	90
A	Additional informations	91
A.1	DFSPH implementation details	91
A.1.1	Formulation of Pressure Calculation	91
A.1.2	Numerical Implementation	92
A.2	GHE velocity profile demonstration	93
A.2.1	Turbulent contribution	94
A.3	Explanation about Levenberg-Marquardt algorithm	95
A.4	Derivation of the 1D model solution for the water height computation	96
B	Technical considerations	99

List of Figures

3.1	Lagrangian versus Eulerian specification. The grid deforms accordingly to the described object in Lagrangian representation whereas it remains undeformed in the Euler one.	16
3.2	SPH Interpolation of $0.5(x + y) \tanh(xy) + \cos(x^2)$ along a representation line s in a two dimensional domain.	18
4.1	Cubic kernel and its derivative with their respective normalisation factor (3D version).	21
4.2	Comparison between standard row-major indexing and Z-curve indexing for a 4×4 grid. The red cell represents the target cell, and blue cells are its spatial neighbours. The Z-curve preserves better spatial locality by keeping neighbouring cells closer in memory index.	23
4.3	Recursive construction of the Z-curve showing how 2×2 blocks are arranged in a Z-pattern, and then these blocks themselves follow a Z-pattern to form larger structures.	23
4.4	Structure of DFSPH algorithm for a single timestep. Key steps: (1-2) Initialization and boundaries, (3) Divergence solver ensuring $\nabla \cdot \mathbf{v} = 0$, (4) Computation of non-pressure forces, (5) Temporary advection, (6-7) Post-advection update and boundaries, (8) Density solver ensuring $\rho = \rho_0$, (9) Final position update with corrected velocities.	25
4.5	Intersection volume of a particle near boundaries. One observes that the intersection volume does have a relationship with the signed distance function. The neighbouring radius is not properly scaled.	33
4.6	Intersection volume of a particle evaluated from interpolated nodes which define the grid cells.	33
4.7	Complete PBD algorithm.	36
5.1	Particles selected inside the rectangle. The rectangle boundaries are shown in red, while particles contributing to the analysis are highlighted.	40
5.2	Applying SPH interpolation for each cells of the grid. The displayed neighbourhood size is not at scale to enhance visualisation.	41
6.1	Two dimensional pipe of length $L = 50$ [m] and diameter $D = 3.2$ [m] with uniform inlet velocity U_0 and expected velocity profile $u(y)$	44

6.2	Mixed velocity profile with $(\delta, \gamma) = (0.1, 0.7)$	47
6.3	Influence of the parameter δ on a almost turbulent velocity profile.	47
6.4	True geometry of the two dimensional channel with emitter object (blue block) and boundaries (grey blocks) considering a larger entry to avoid inlet blocking particles due to non-slip wall condition.	48
6.5	Parameter used to simulate turbulent flow in two dimensional channel.	49
6.6	Final timestep ($t = 30[s]$) of the turbulent flow displaying velocity magnitude field. The disturbed velocity field is partially chaotic, which is an expected behaviour for fully turbulent flow in such a configuration.	50
6.7	Particle distribution inside the rectangle with a width of 10 particles. Showing that particles are indeed uniformly distributed and that statistical data can be extracted from it.	50
6.8	Time derivatives of density and velocity at $25m \pm 10r$ over the whole diameter.	51
6.9	Time evolution and derivative of the mass of a single particle at $25m$	52
6.10	Centreline velocity distribution along the pipe with its mean value and standard deviation.	52
6.11	Spatial derivatives of the velocity vector $\partial \mathbf{u} / \partial x$	53
6.12	Two spatial derivatives of density in the domain $(x, y) \in [1, 46] \times [-D/2, D/2]$	54
6.13	Computation of the mass and volumetric flow rates $(x, y) \in [1, 46] \times [-D/2, D/2]$ in pipe yielding an error about 0.398%.	54
6.14	Fitted velocity profile based on the GHE model with $(\delta, \gamma) = (0.004, 0.8174)$ using raw data.	55
6.15	Fitted velocity profile based on the GHE model with $(\delta, \gamma) = (0.00103, 0.7840)$ using filtered data.	56
6.16	Two dimensional channel of length $L = 8 [m]$ with inlet flow rate Q_o and outlet velocity U_{outlet} . There is a parabolic obstacle in the middle of the channel.	57
6.17	Qualitative behaviour of the water height during the hydraulic jump.	59
6.18	Hydraulic jump location using conjugated height principle.	60
6.19	Parameter used to simulate the free surface flow.	61
6.20	Flow development in the free surface configuration in four steps. The color map is based on the velocity.	62
6.21	Steady state assessment of the free surface flow using the total energy of the system.	63
6.22	Computation of the mass and volumetric flow rates yielding error about 0.779%.	64
6.23	Standard deviation of the velocity $\sigma_u(x)$ across the whole domain. The deviation is almost zero everywhere except in the vicinity of the elliptic obstacle.	64
6.24	Spatially averaged pressure over the whole domain. The mean RMSE is 91.687 [Pa], which demonstrates the expected result.	65
6.25	Local RMSE along the whole horizontal domain. The RMSE is much higher in the upstream region compared to the downstream region.	66
6.26	Comparison of the numerical water height with the analytical one.	67
6.27	Numerical Froude number. The critial value $Fr = 1[-]$ is well captured at extremum of the obstacle at $x = 10 [m]$	67
6.28	Quiver plot in the critical region. It can be observed the existence of two flows. The one near $x = 11.75 [m]$ at the free surface which is stationary. The second one, being the main flow, lies below the former and goes downstream (indicated by the arrow directions).	68
6.29	Numerical total head correctly fitting the expected value and demonstrating a head loss due to the hydraulic jump.	69

6.30	Total energy of the system at a single timestep. One observes that the energy highly decreases in the hydraulic jump and than stabilises just after but with a smaller value than upstream.	69
7.1	Complete set of 2D views of the bridge and flow configuration.	72
7.2	Parameters used to simulate the final flow with bridge.	74
7.3	Snapshots of the bridge configuration at the beginning and the end of the simulation with $h_0 = 200$ [mm].	74
7.4	Snapshots of the bridge configuration with coarse spatial discretization $r = 10$ [mm].	75
7.5	Water height distributions (in the loaded bridge case) through the two bridge openings. The slices have been obtained through the XY axis at $z = \pm 150$ [mm] $\pm 20r$	76
7.6	Streamlines (in XZ plane at $y = 250$ [mm] ± 4 particles) at the end of the simulation.	76
7.7	Total head distribution in plane XY at $z = \pm 150$ [mm] $\pm 20r$. These values have been time-averaged for $t \in [10, 25]$ [s].	77
7.8	Quiver plot XY at $z = \pm 150$ [mm] $\pm 20r$ at $t = 25$ [s]. The particles near the corners of the bridge have a highly reduced velocity compared to the others (indicated by the length of the arrows). The wood logs have an important impact in the upstream region of the bridge.	78
7.9	Snapshots of the bridge configuration at the beginning and the end of the simulation with $h_0 = 100$ [mm]. One observes the wood logs being swept by the current in the downstream region.	79
7.10	Streamlines (in XZ plane at $y = 120$ [mm] ± 4 particles.) at the end of the simulation. The blank areas before the bridge correspond to the wood log locations with particles passing underneath.	79
7.11	Water height distributions (in the free surface flow case) through the two bridge openings. The slices have been obtained through the XY axis at $z = \pm 150$ [mm] $\pm 20r$	80
7.12	Water height distributions (in the free surface flow case) through the two bridge openings. The slices have been obtained through the XY axis at $z = \pm 150$ [mm] $\pm 20r$. The oscillations in the upstream region should not be considered due to the presence of the wood logs there.	81
7.13	Total head distribution (in the free surface flow case) in plane XY at $z = \pm 150$ [mm] $\pm 20r$. These values have been time-averaged for $t \in [10, 25]$ [s].	81

Aknowledgment

Rédiger un mémoire n'est jamais simple et demande beaucoup d'énergie, de motivation et de réflexion. Mais en dehors de toutes ces considérations, c'est surtout grâce à mon entourage que j'ai pu arriver jusqu'au bout. C'est pourquoi je tiens à vous remercier tous.

Tout d'abord, merci au Professeur Archambeau qui m'a épaulé tout au long de cette aventure. Il s'est toujours montré disponible et avenant à n'importe quel moment. Nous avons souvent tendance à considérer ces choses comme normales ou acquises mais c'est en réalité très rarement le cas. Je tiens à souligner son point de vue pragmatique et logique des choses qui m'a été d'une aide cruciale dans les moments de doute ou d'incompréhension. Mais ce que je retiendrai surtout, ce sont les bons moments passés pendant les "réunions de quelques minutes maximum".

Ensuite, je tiens à remercier ma famille et en particulier mes parents et mon frère Thomas. Je suis devenu la personne d'aujourd'hui, pour le meilleur comme pour le pire, en majeure partie grâce à vous. Papa, Maman, vous m'avez toujours soutenu depuis le départ et je me rappelle très bien le chemin que nous avons parcouru ensemble, et je ne pourrai jamais assez vous remercier pour cela. Thomas, tu as toujours été un modèle pour moi jusqu'à présent. J'ai appris en avance sur mon âge au travers de tes réussites et difficultés, et tu peux considérer ce travail comme le résumé de ce que tu m'as transmis.

Finalement, j'aimerais remercier deux toutes dernières personnes. On ne choisit pas sa famille biologique, mais on choisit sa famille de coeur pour la vie. Louise et Anaïs, vous occupez une place toute particulière pour moi. Au cours de ces années, nous avons vécu tant de choses ensemble, certaines très chouettes, d'autres un peu moins, mais nous savons que peu importe ce que nous réserve la vie, nous pourrons toujours compter les uns sur les autres. Tous vos bons conseils et ces moments passés ensemble autour d'un verre à râler sur ce qui nous passe par la tête me sont très précieux, et c'est aussi votre présence qui fait la personne que je suis actuellement.

1.1 Introduction and motivation

Flood events have been a recurring natural hazard throughout human history, occurring when water accumulates beyond the capacity of natural or urban drainage systems. Since the early 20th century, engineers have recognized the critical need to design buildings and infrastructure capable of withstanding flood-induced forces. Traditional engineering approaches rely on simplified calculations and numerous approximations to estimate the pressure loads that structures must endure [1]. While these methods generally provide reasonable order-of-magnitude estimates, they are time-consuming and resource-intensive. Moreover, the prediction of the motion of rolling wooden debris, for instance, has been studied for more than two decades [2].

This master thesis proposes an alternative approach: directly simulating flood flows using computational fluid dynamics to evaluate their impact on building boundaries. This approach requires selecting an appropriate numerical method to solve the fluid dynamic equations. Several Eulerian models have already been used to simulate free surface flows, including methods such as the finite volume method [3] and the finite element method [4], which have already proven their performance.

However, this study chooses the Lagrangian Smoothed Particle Hydrodynamics (SPH) method [5], which raises important questions about computational efficiency and accuracy that this master thesis will address. The SPH method, originally developed in 1977 for astrophysical simulations, has since found widespread application in fluid mechanics, particularly for hydraulic problems. Successful applications in dam break scenarios and water splash simulations [6] as well as free surface flow over a sharp-crested weir [7] have demonstrated the capability of the method to handle free-surface flows accurately. Moreover, turbulence with SPH has already been studied [8] and accurately validated. Finally, the fluid-solid interactions with SPH had been studied, for instance, a decade ago in the context of rigid bodies inside visco-elastic fluid [9].

Nonetheless, all these studies did not provide simulations considering these phenomena altogether. Hence, this thesis investigates whether SPH can effectively simulate turbulent free-surface flows, particularly those occurring during river and canal flooding and clogging.

1.2 Objectives of this master thesis

The primary objective is to evaluate the capability of the SPH method to simulate 3D turbulent free-surface flows with dynamic rigid bodies acting as obstacles. While experimental studies of such flows exist, they are limited by scale constraints. This numerical approach could potentially overcome these limitations.

1.3 Methodology

One will use the SPlisHSPlasH software [10], which offers competitive performance in terms of computation time and provides various modelling options (pressure solvers, kernels, boundary handling methods). Although this software is focused on graphical visualization through its integrated OpenGL preview system, one will assess its capacity to produce physically accurate results.

This master thesis will first describe the mathematical and physical framework in which the equations dictate the laws of physics in Chapter 2. Chapter 3 will then review and fully explain the fundamentals of the SPH method. Chapter 4 will explain the SPlisHSPlasH algorithm logic and detail each method employed in the subsequent simulations.

Chapter 5 discusses the data treatment procedures used to exploit outputs. The performances of the software will then be tested through preliminary validation tests in Chapter 6, with each configuration focused on a unique physical aspect that contributes to the final goal of the master thesis.

After all components have been assessed, the final validation test will be performed in Chapter 7, incorporating all the confidence gained throughout the previous discussions. Chapter 8 will discuss the limitations encountered during the study, and Chapter 9 will present the conclusions and future perspectives.

Governing equations derivation

To effectively analyse the flow of fluid particles, one must first derive the governing equations that describe how each particle behaves in relation to its environment. This chapter derives each fundamental equation that constitutes the mathematical framework of fluid dynamics. The methodology employed to achieve the final form of these governing equations is inspired by the lecture notes of *Professor Dauby* in the context of his course *Irreversibility, Instabilities and Chaos (PHYS0961-1)*.

2.1 Mass conservation law

The conservation of mass states that for any material volume $V(t)$ moving with the fluid, its mass $M(t)$ must remain constant:

$$M(t) = \int_{V(t)} \rho(\mathbf{x}, t) dV = \text{constant} \quad (2.1)$$

where $\rho(\mathbf{x}, t)$ is the local density for all point within the material volume.

Taking the material derivative of this expression:

$$\frac{dM(t)}{dt} = \frac{d}{dt} \int_{V(t)} \rho(\mathbf{x}, t) dV = 0 \quad (2.2)$$

Using Reynolds transport theorem (which relates the time rate of change of an integral over a moving volume to the velocity of the boundary control volume) and the fact that this must hold for any arbitrary volume, one obtains the continuity equation:

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0 \quad (2.3)$$

This is the differential form of mass conservation that one will later discretize using the SPH method. Note that one introduced the D/Dt notation which is the well known *material derivative* equivalent to the *total derivative* but more common in the realm of the fluid dynamic analysis.

2.2 Linear momentum conservation law

The momentum conservation principle states that the rate of change of momentum equals the sum of forces acting on the system. For a material volume, the momentum is:

$$\mathbf{P}(t) = \int_{V(t)} \rho \mathbf{v} dV \quad (2.4)$$

The forces acting on the volume can be split into two categories:

- Body forces: $\int_{V(t)} \rho \mathbf{b} dV$
- Surface forces: $\int_{S(t)} \mathbf{t} dS = \int_{V(t)} \nabla \cdot \boldsymbol{\sigma} dV$

where $\boldsymbol{\sigma}$ is the stress tensor and \mathbf{b} represents body forces per unit mass.

Applying Reynolds transport theorem and equating the rate of change of momentum to the sum of these forces:

$$\rho \frac{D\mathbf{v}}{Dt} = \rho \mathbf{b} + \nabla \cdot \boldsymbol{\sigma} \quad (2.5)$$

For a Newtonian fluid, the stress tensor is related to the rate of strain tensor $\mathbf{E} = \frac{1}{2} (\nabla \mathbf{v} + \nabla \mathbf{v}^T)$. The associated constitutive equations are discussed in more details in Section 2.5 but they finally read:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T) \quad (2.6)$$

where p is the pressure, \mathbf{I} is the identity tensor, and μ is the dynamic viscosity.

Substituting this into equation 2.5 gives us:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}_{ext} \quad (2.7)$$

where \mathbf{f}_{ext} represents external forces per unit volume.

2.3 Angular momentum conservation law

For our type of medium (nonpolar and continuous), the law concerning the rotation of this medium is translated by the symmetry of the stress tensor as followed:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T \quad (2.8)$$

2.4 Energy conservation law

For a complete description of fluid behaviour, it is essential to account for how the internal energy of the medium evolves in relation to the previously defined variables.

Considering a continuum medium (described by a time-changing volume $V(t)$), there exist two types of energy: E_c (kinetic energy) and U (internal energy) within the prescribed medium (which in our case is composed of fluid particles):

$$E_c = \int_{V(t)} \frac{1}{2} \rho |\mathbf{v}|^2 dV \quad (2.9)$$

$$U = \int_{V(t)} \rho u dV \quad (2.10)$$

where u is the specific internal energy (internal energy per unit mass).

The power P supplied to this medium by applied external forces can be characterized by two contributions, one volumic and one surfacic:

$$P = P_V + P_S = \int_{V(t)} \rho \mathbf{b} \cdot \mathbf{v} dV + \int_{S(t)} \mathbf{t} \cdot \mathbf{v} dS \quad (2.11)$$

where \mathbf{b} represents the volumic body forces and \mathbf{t} represents the surfacic forces.

Similarly, the total heat supplied to the medium can be divided into two contributions:

$$\dot{Q} = \dot{Q}_V + \dot{Q}_S = \int_{V(t)} \rho r dV - \int_{S(t)} \mathbf{q} \cdot \mathbf{n} dS \quad (2.12)$$

The heat flux vector \mathbf{q} describes the surface heat supplies through the physical phenomenon of thermal conduction. The volume heat source, defined by its density r (energy per unit mass per unit time), accounts for heat supplied directly within the medium. Note the negative sign before the heat flux vector, which establishes that heat flux entering the medium is considered as positive energy increase (respecting the conventional direction of the normal vector).

One now combines all forms of energy according to the first principle of thermodynamics:

$$\frac{d}{dt} (E_c + U) = P_V + P_S + \dot{Q}_V + \dot{Q}_S \quad (2.13)$$

Rewriting this relation in integral form:

$$\frac{D}{Dt} \int_{V(t)} \rho \left(\frac{1}{2} \|\mathbf{v}\|^2 + u \right) dV = \int_{V(t)} \rho \mathbf{b} \cdot \mathbf{v} dV + \int_{S(t)} \mathbf{t} \cdot \mathbf{v} dS + \int_{V(t)} \rho r dV - \int_{S(t)} \mathbf{q} \cdot \mathbf{n} dS \quad (2.14)$$

Applying the Reynolds transport theorem and the divergence theorem, and noting that this must hold for any arbitrary volume, one obtains the local form of the energy conservation equation:

$$\rho \frac{de}{dt} = \rho \frac{d}{dt} \left(\frac{1}{2} \|\mathbf{v}\|^2 + u \right) = \rho \mathbf{b} \cdot \mathbf{v} + \nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{v}) - \nabla \cdot \mathbf{q} + \rho r \quad (2.15)$$

This equation, in addition to the continuity equation (2.3) and the momentum equation (2.7), forms the complete system of governing equations for fluid dynamics.

2.5 Constitutive equations: Newtonian fluids

Having derived the conservation equations, one must now establish the relationship between the stress tensor and the properties of the fluid motion. This connection is made through constitutive equations that describe the material behaviour. Since one is interested in the modelisation of water, the most common fluid, being the Newtonian fluid, will be described here.

Friction forces occur due to the non zero relative velocities between particles. Hence, the stress tensor may be linked to the velocity gradient of the prescribed fluid.

The rate of strain tensor \mathbf{E} and the vorticity tensor $\boldsymbol{\Omega}$ are defined as the symmetric and antisymmetric components of the velocity gradient tensor ($\nabla \mathbf{v}$):

$$\mathbf{E} = \frac{1}{2} (\nabla \mathbf{v} + \nabla \mathbf{v}^T) \quad (2.16)$$

$$\boldsymbol{\Omega} = -\frac{1}{2} (\nabla \mathbf{v} - \nabla \mathbf{v}^T) \quad (2.17)$$

The physical interpretation of these tensors is fundamental to understanding constitutive equations. The vorticity tensor $\boldsymbol{\Omega}$ characterizes the rotational motion in the neighbourhood of a specific particle. Hence, this neighbourhood moves as a compact block and does not generate friction. Conversely, the rate of strain tensor explicitly translates the relative velocity between a particle and its neighbourhood. Hence \mathbf{E} generates internal frictions such that the viscous stress tensor $\boldsymbol{\sigma}^v$ uniquely depends on it.

If the velocity gradients remain small, one may write the viscous stress tensor as a linear (and homogeneous) expression with respect to the rate of deformation tensor. It is written as:

$$\sigma_{ij}^v = a_{ijkl} V_{kl} \quad (2.18)$$

where a_{ijkl} are the components of a fourth-order tensor that characterizes the viscous properties of the fluid.

For homogeneous materials, a_{ijkl} does not depend on the position but may still vary according to thermodynamic variables (T, ρ).

But considering the symmetry of the stress tensors $\boldsymbol{\sigma}$, one actually has two different symmetries for a_{ijkl} :

- a_{ijkl} must be symmetric with respect to its first two indices ($a_{ijkl} = a_{jikl}$) due to the symmetry of the stress tensor.
- a_{ijkl} can also be assumed symmetric with respect to its last two indices ($a_{ijkl} = a_{ijlk}$) because the rate of deformation tensor is symmetric.

For an isotropic fluid, where material properties are independent of direction, the general tensor a_{ijkl} can be simplified. It is then possible to show that the following constitutive equation is obtained for an isotropic Newtonian fluid:

$$\sigma_{ij}^v = \lambda E_{kk} \delta_{ij} + 2\mu E_{ij} \quad (2.19)$$

where λ and μ are two material parameters that fully define the a_{ijkl} tensor for isotropic media. The parameter μ is particularly significant as it characterizes the resistance of the fluid to shearing deformation (the dynamic viscosity). If one defines the dynamic bulk viscosity as $\kappa = \lambda + \frac{2}{3}\mu$, one can finally read:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \kappa(\nabla \cdot \mathbf{v})\mathbf{I} + 2\mu\hat{\mathbf{E}} \quad (2.20)$$

with $\hat{\mathbf{E}}$ being the deviatoric part of the rate of strain tensor.

Two important special cases are important mention: It is important to note that κ can be neglected without any problems since it already disappears in incompressible fluids and is experimentally almost null in compressible ones.

2.6 Turbulence in fluid flows

Having established the fundamental governing equations for fluid motion, it becomes essential to address one of the most complex and fascinating phenomena in fluid dynamics being the turbulence. This section aims to explain the nature of turbulent flows and their mathematical representation within the framework one has developed. While the discussion focuses on understanding the fundamental characteristics of turbulence, the specific method for numerically handling turbulent flows will be discussed later in Section 4.6.

Turbulent flows represent a regime where the derived Equation (2.7) produces solutions that are inherently chaotic, three-dimensional, and show complex temporal and spatial fluctuations. Understanding this phenomenon requires analysing how the nonlinear terms in our governing equations give rise to this phenomenon.

2.6.1 Origins of turbulence in the Navier-Stokes equations

The transition from laminar to turbulent flow can be understood by analysing the momentum Equation(2.7):

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}_{ext} \quad (2.21)$$

Expanding the material derivative shows the source of complexity:

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}_{ext} \quad (2.22)$$

The nonlinear convective term $\mathbf{v} \cdot \nabla \mathbf{v}$ is the mathematical origin of turbulence. This term couples the velocity field to itself, creating a feedback mechanism where velocity fluctuations can amplify through nonlinear interactions.

To illustrate this mechanism, let us consider a small perturbation $\delta \mathbf{v}$ induced in an initial uniform flow \mathbf{V}_0 :

$$\mathbf{v} = \mathbf{V}_0 + \delta \mathbf{v} \quad (2.23)$$

Substituting this decomposition into the nonlinear term yields:

$$\mathbf{v} \cdot \nabla \mathbf{v} = \mathbf{V}_0 \cdot \nabla \mathbf{V}_0 + \mathbf{V}_0 \cdot \nabla \delta \mathbf{v} + \delta \mathbf{v} \cdot \nabla \mathbf{V}_0 + \boxed{\delta \mathbf{v} \cdot \nabla \delta \mathbf{v}} \quad (2.24)$$

The first term is independent of the perturbation while the second and third terms remain linear in the perturbation. However, the last term is nonlinear in the perturbation $\delta \mathbf{v} \cdot \nabla \delta \mathbf{v}$: the perturbation acts on itself. This self-interaction allows small perturbations to modify their own evolution, potentially leading to exponential growth when viscous effects are too small. When the Reynolds number $Re = \frac{\rho V L}{\mu}$ (with V and L the characteristic velocity and length scales) reaches critical threshold values, these nonlinear interactions overcome the stabilizing effects of viscous diffusion, leading to turbulence.

The Reynolds number effectively represents the ratio of inertial forces to viscous forces. At low Reynolds numbers, viscous effects dominate and flows remain smooth and predictable. However, as the Reynolds number increases, the nonlinear inertial terms become more important, eventually creating instabilities that evolve into the complex, multi-scale phenomenon known as turbulence.

2.6.2 The turbulent cascade and scale separation

Turbulent flows are characterized by the simultaneous presence of motion at different length and time scales. This multi-scale nature arises from what Kolmogorov described as the energy cascade process, where kinetic energy is transferred from large, energy scales down to progressively smaller scales until it is finally dissipated by molecular viscosity.

The range of scales present in a turbulent flow can be estimated through dimensional analysis. The largest scales, characterized by the integral length scale L_0 , are determined by the geometry of the flow domain and contain most of the turbulent kinetic energy. At the other extreme,

the smallest scales—known as the Kolmogorov scales—are determined by the balance between nonlinear energy transfer and viscous dissipation.

The Kolmogorov length scale η can be expressed as:

$$\eta = \left(\frac{\nu^3}{\varepsilon} \right)^{1/4} \quad (2.25)$$

where $\nu = \mu/\rho$ is the kinematic viscosity and ε is the rate of energy dissipation per unit mass. The separation between the largest and smallest scales varies according to the Reynolds number. The ratio of integral to Kolmogorov scales reads:

$$\frac{L_0}{\eta} \sim Re^{3/4} \quad (2.26)$$

This scaling relationship has important implications for numerical simulations. As Reynolds numbers increase, the range of scales that must be resolved grows rapidly. Hence, making direct numerical simulation (DNS) of the complete Navier-Stokes equations is usually avoided.

2.7 Rigid body dynamics in fluid flows

Having established the governing equations for fluid motion and addressed the complexities of turbulent flows, one must now consider the interaction between the fluid and rigid bodies immersed within it. This section derives the fundamental equations governing the motion of rigid objects subject to hydrodynamic forces, establishing the mathematical framework necessary for fluid-structure interaction simulations.

2.7.1 Equations of motion for rigid bodies

The motion of a rigid body is completely described by the evolution of its centre of mass position $\mathbf{x}_c(t)$ and its orientation, represented by a rotation matrix $\mathbf{R}(t)$ or quaternion $\mathbf{q}(t)$ (which are used in numerical equation). The governing equations come from second law of Newton for translational motion and equations of Euler for rotational motion.

For the translational motion of the centre of mass, the governing equation reads:

$$m \frac{d^2 \mathbf{x}_c}{dt^2} = \mathbf{F}_{total} = \mathbf{F}_{fluid} + \mathbf{F}_{ext} \quad (2.27)$$

where m is the mass of the rigid body, \mathbf{F}_{fluid} represents the hydrodynamic forces exerted by the surrounding fluid, and \mathbf{F}_{ext} represents external forces (e.g. gravity).

For the rotational motion the governing equation, in the body-fixed reference frame, reads:

$$\mathbf{I} \frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) = \mathbf{M}_{total} = \mathbf{M}_{fluid} + \mathbf{M}_{ext} \quad (2.28)$$

where \mathbf{I} is the inertia tensor in the body-fixed frame, $\boldsymbol{\omega}$ is the angular velocity vector, \mathbf{M}_{fluid} represents the hydrodynamic moments about the centre of mass, and \mathbf{M}_{ext} represents external torques.

In the inertial reference frame, the rotational equation takes the form:

$$\frac{d(\mathbf{I}_0 \boldsymbol{\omega})}{dt} = \mathbf{M}_{total} \quad (2.29)$$

where $\mathbf{I}_0(t) = \mathbf{R}(t)\mathbf{I}\mathbf{R}^T(t)$ is the time-varying inertia tensor in the inertial frame.

2.7.2 Hydrodynamic forces and moments

The coupling between the fluid and rigid body motion is defined using the hydrodynamic forces and moments, which are evaluated by integrating the fluid stresses over the body surface S_b :

$$\mathbf{F}_{fluid} = \int_{S_b} \boldsymbol{\sigma} \cdot \mathbf{n} dS \quad (2.30)$$

$$\mathbf{M}_{fluid} = \int_{S_b} (\mathbf{x} - \mathbf{x}_c) \times (\boldsymbol{\sigma} \cdot \mathbf{n}) dS \quad (2.31)$$

where \mathbf{n} is the outward unit normal to the body surface and $\boldsymbol{\sigma}$ is the fluid stress tensor.

Substituting Equation (2.6) into Equation (2.30), it becomes:

$$\mathbf{F}_{fluid} = - \int_{S_b} p \mathbf{n} dS + \int_{S_b} \mu (\nabla \mathbf{v} + \nabla \mathbf{v}^T) \cdot \mathbf{n} dS \quad (2.32)$$

The first integral represents the pressure forces (including buoyancy when gravity is present), while the second is about viscous drag and lift forces.

Similarly, substituting Equation (2.6) into Equation (2.31) gives the expanded moment expression:

$$\mathbf{M}_{fluid} = - \int_{S_b} (\mathbf{x} - \mathbf{x}_c) \times (p \mathbf{n}) dS + \int_{S_b} (\mathbf{x} - \mathbf{x}_c) \times [\mu (\nabla \mathbf{v} + \nabla \mathbf{v}^T) \cdot \mathbf{n}] dS \quad (2.33)$$

The first term corresponds to the pressure-induced moment (including buoyancy effects), while the second term represents the viscous moment contribution.

2.7.3 Kinematic constraints and boundary conditions

The interaction between the fluid and rigid body requires the kinematic compatibility at the interface. However, unlike the idealized no-slip condition, real interfaces often show partial slip (e.g. friction). The general boundary condition at the fluid-solid interface can be expressed as a decomposition into normal and tangential components.

For the normal component, the impermeability condition requires:

$$(\mathbf{v}_{fluid} - \mathbf{v}_{body}) \cdot \mathbf{n} = 0 \quad \text{for } \mathbf{x} \in S_b \quad (2.34)$$

where $\mathbf{v}_{body} = \mathbf{v}_c + \boldsymbol{\omega} \times (\mathbf{x} - \mathbf{x}_c)$ represents the velocity of the body surface at point \mathbf{x} .

For the tangential component, the frictional behaviour is characterized by:

$$\boldsymbol{\tau}_{fluid} = -\mu_f \mathbf{v}_{rel,t} \quad (2.35)$$

where $\boldsymbol{\tau}_{fluid}$ is the tangential stress exerted by the fluid, μ_f is the friction coefficient, and $\mathbf{v}_{rel,t}$ is the relative tangential velocity between the fluid and the body surface:

$$\mathbf{v}_{rel,t} = \mathbf{v}_{fluid} - \mathbf{v}_{body} - [(\mathbf{v}_{fluid} - \mathbf{v}_{body}) \cdot \mathbf{n}] \mathbf{n} \quad (2.36)$$

When $\mu_f \rightarrow \infty$, one recovers the no-slip condition where $\mathbf{v}_{rel,t} = \mathbf{0}$. Conversely, $\mu_f = 0$ corresponds to perfect slip with no tangential stress transmission.

2.7.4 Rigidity constraints

The fundamental property distinguishing rigid bodies from deformable ones is the invariance of distances between any two material points within the body. This rigidity constraint can be mathematically expressed as:

$$\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| = \|\mathbf{X}_i - \mathbf{X}_j\| \quad \forall i, j \in \text{body} \quad (2.37)$$

where \mathbf{X}_i and \mathbf{X}_j denote the initial positions of material points in the reference configuration, while $\mathbf{x}_i(t)$ and $\mathbf{x}_j(t)$ are their current positions.

This constraint is automatically satisfied through the kinematic description of rigid body motion:

$$\mathbf{x}(t) = \mathbf{x}_c(t) + \mathbf{R}(t)(\mathbf{X} - \mathbf{X}_c) \quad (2.38)$$

where $\mathbf{R}(t)$ is a rotation matrix satisfying $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ and $\det(\mathbf{R}) = 1$.

The orthogonality constraint on the rotation matrix ensures that distances and angles are preserved during the motion. When using quaternions $\mathbf{q} = (q_0, q_1, q_2, q_3)$ to represent rotations, the unit norm constraint:

$$\|\mathbf{q}\|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \quad (2.39)$$

must be enforced throughout the simulation to maintain rigid body kinematics.

2.7.5 Application for free surface flows

When dealing with free surface flows containing floating or partially submerged rigid bodies, additional complexity arises from the dynamic nature of the wetted surface $S_b(t)$. The hydrodynamic forces must be integrated only over the portion of the body surface in contact with the fluid:

$$\mathbf{F}_{fluid} = \int_{S_b \cap \Omega_f(t)} \boldsymbol{\sigma} \cdot \mathbf{n} dS \quad (2.40)$$

where $\Omega_f(t)$ denotes the time-varying fluid domain.

This introduces the need to track both the free surface evolution and the intersection between the rigid body and the fluid phase, adding another layer of complexity to the numerical implementation. The treatment of such moving boundaries and their interaction with turbulent free surface flows represents one of the primary challenges addressed in this work.

2.8 Complete coupled system

The complete system governing fluid-structure interaction combines the fluid Equations(2.3) and (2.7) with the rigid body Equations(2.27); (2.28) and (2.38), coupled with the hydrodynamic forces (2.32) and the kinematic constraints (2.34). This coupled system can be summarized as:

$$\text{Fluid phase: } \frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0 \quad (2.41)$$

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}_{ext} \quad (2.42)$$

$$\text{Rigid body: } m \frac{d^2 \mathbf{x}_c}{dt^2} = \mathbf{F}_{fluid} + \mathbf{F}_{ext} \quad (2.43)$$

$$\mathbf{I} \frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) = \mathbf{M}_{fluid} + \mathbf{M}_{ext} \quad (2.44)$$

$$\text{Interface: } (\mathbf{v}_{fluid} - \mathbf{v}_{body}) \cdot \mathbf{n} = 0 \quad \text{on } S_b \quad (2.45)$$

$$\tau_{fluid} = -\mu_f \mathbf{v}_{rel,t} \quad \text{on } S_b \quad (2.46)$$

$$\text{Rigidity: } \mathbf{R}^T \mathbf{R} = \mathbf{I}, \quad \det(\mathbf{R}) = 1 \quad (2.47)$$

Mathematical fundamentals of the SPH method

Once the physical background has been fully established, one needs to explain and detail the mathematical tools that will be used to describe the aforementioned physical fluid equations.

As stated in the beginning of this thesis, the equations will be spatially discretized using the so-called *SPH interpolation* which relies on a *Lagrangian representation* and a *meshless configuration*. A complete review of the SPH method has also been proposed by Springel et al. [11]

As such, it is proposed in this chapter to study each of these features in details.

3.1 Lagrangian and Eulerian specification

In classical field theories, the Lagrangian specification of the flow field is a way of looking at fluid motion where the observer follows an individual fluid parcel as it moves through space and time. Whereas, the Eulerian specification of the flow field is a way of looking at fluid motion that focuses on specific locations in the space through which the fluid flows as time passes.

In other words for the Lagrangian representation, the frame of reference does move accordingly with the observed particle. Whereas it does not move at all for the Eulerian representation. These representations can be observed in Figure 3.1.

Mathematically, these two specifications are actually linked through the concept of total/material derivatives. The material derivative is the same as Lagrangian derivative. Let us suppose one has a flow field $\mathbf{u}(\mathbf{x}, t)$, and one also receive a generic field with Eulerian specification $\mathbf{F}(\mathbf{x}, t)$. Hence the total derivative reads:

$$\underbrace{\frac{D\mathbf{F}(\mathbf{x}, t)}{Dt}}_{\text{Lagrangian}} = \frac{\partial \mathbf{F}}{\partial t} + \frac{d\mathbf{x}}{dt} \frac{\partial \mathbf{F}}{\partial \mathbf{x}} = \underbrace{\frac{\partial \mathbf{F}}{\partial t}}_{\text{Eulerian}} + \mathbf{u} \cdot (\nabla \mathbf{F}) \quad (3.1)$$

It clearly shows that the two derivatives (Lagrangian and Eulerian) are linked one to each other. the only difference is the appearance of the non-linear term $\mathbf{u} \cdot (\nabla \mathbf{F})$ that has to be taken into account in every Eulerian methods (such as finite element method, finite volume method, ...).

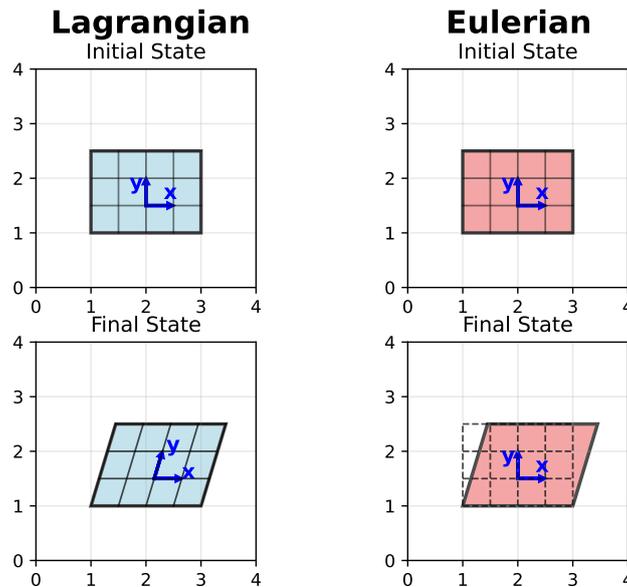


Figure 3.1: Lagrangian versus Eulerian specification. The grid deforms accordingly to the described object in Lagrangian representation whereas it remains undeformed in the Euler one.

So the main advantage of the Lagrangian representation is that the non-linearities, due to convective terms, are implicitly managed through the calculation of the material derivative.

3.2 Meshless algorithm

One of the key features of the SPH method is its meshless nature, which offers several advantages compared to traditional mesh-based methods. In SPH, the continuous fluid domain is discretized into a finite number of particles that carry physical properties (mass, density, pressure, velocity, etc.). Unlike grid-based methods, these particles are not connected by any predefined mesh or grid structure, allowing for a more flexible and adaptable numerical approach.

The absence of a mesh structure provides unique capabilities in handling complex fluid dynamics problems. Since there is no mesh that could become distorted, the gridless methods naturally excel at simulating scenarios involving large deformations, fragmentation, or free surfaces (with this latter being, lately, the point of interest).

3.3 Spatial SPH discretization

The following discussion of smoothing kernels is based on the foundational work by Bender et al. [12] on SPLishSPLasH, to whom full credit is due.

3.3.1 Approximated integral representation

Taking into account the previously presented features, one can now explain the fundamental aspect of the SPH interpolation, which lies in its approach to discretizing functions and partial differential operators. The most important underlying mathematical concept is the *integral representation*, written as:

$$A(\mathbf{x}) = \int_{\Omega} A(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (3.2)$$

where:

- $\delta(\mathbf{x})$ is the Dirac delta function
- Ω is the integration domain
- \mathbf{x}' is the integration variable
- \mathbf{x} is the point where the function is evaluated

It expresses any function as a convolution with the Dirac delta function, as shown in Equation 3.2. This representation is exact and forms the basis for the SPH approximation, where the Dirac delta function is replaced by a smoothing kernel. This smoothing kernel is clearly explicated in the following section

3.3.2 Smoothing kernel conditions

The kernel function, denoted as $W : \mathbb{R}^{2,3} \times \mathbb{R}^+ \rightarrow \mathbb{R}$, must satisfy several properties to ensure a valid approximation. These properties can be categorized into essential and optional conditions: Essential properties for valid approximation $A(\mathbf{x}) \approx (A * W)(\mathbf{x})$:

$$\int_{\mathbb{R}^d} W(\mathbf{r}', h) d\mathbf{v}' = 1 \quad (\text{normalization condition}) \quad (3.3)$$

$$\lim_{h' \rightarrow 0} W(\mathbf{r}, h') = \delta(\mathbf{r}) \quad (\text{Dirac-}\delta \text{ condition}) \quad (3.4)$$

Optional properties that enhance numerical stability and physical meaning:

$$W(\mathbf{r}, h) \geq 0 \quad (\text{positivity condition}) \quad (3.5)$$

$$W(\mathbf{r}, h) = W(-\mathbf{r}, h) \quad (\text{symmetry condition}) \quad (3.6)$$

$$W(\mathbf{r}, h) = 0 \text{ for } |\mathbf{r}| \geq h \quad (\text{compact support condition}) \quad (3.7)$$

The positivity condition ensures exclusively positive weighting and helps meeting physical constraints, such as $\rho \geq 0$. The symmetry condition allows for first-order consistent approximations, while the compact support condition improves computational efficiency by limiting the range of particle interactions. Using this kernel function, a function A and its gradient ∇A can be approximated as:

$$A(\mathbf{x}) \approx \int_{\Omega} A(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \quad (3.8)$$

$$\nabla A(\mathbf{x}) \approx - \int_{\Omega} A(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (3.9)$$

Consistency condition

The accuracy of the kernel approximation can be analysed through a Taylor series expansion. For a continuous approximation (before particle discretization), one can express:

$$(A * W)(\mathbf{x}) = \int_{\Omega} \left[A(\mathbf{x}) + \nabla A | \mathbf{x} \cdot (\mathbf{x}' - \mathbf{x}) + \frac{1}{2} (\mathbf{x}' - \mathbf{x}) \cdot \nabla \nabla A | \mathbf{x} (\mathbf{x}' - \mathbf{x}) + O(|\mathbf{r}|^3) \right] W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \quad (3.10)$$

Due to the normalization condition (3.3) and symmetry property (3.6) of the kernel, one has:

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 1 \quad (3.11)$$

$$\int_{\Omega} (\mathbf{x}' - \mathbf{x}) W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = \mathbf{0} \quad (3.12)$$

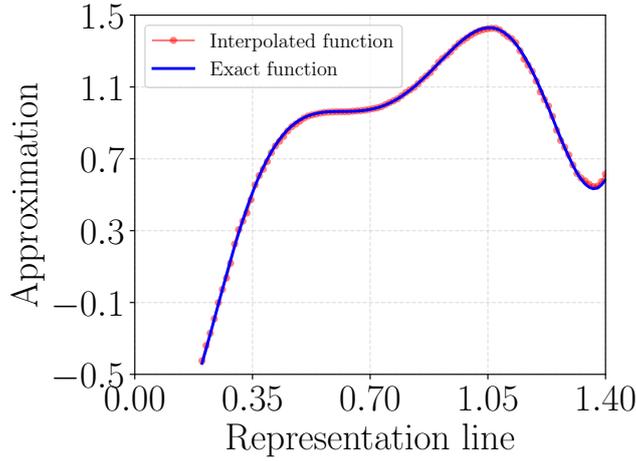


Figure 3.2: SPH Interpolation of $0.5(x + y) \tanh(xy) + \cos(x^2)$ along a representation line s in a two dimensional domain.

This leads to:

$$(A * W)(\mathbf{x}) = A(\mathbf{x}) + O(h^2) \quad (3.13)$$

showing that normalized, symmetric kernels provide at least first-order consistency in the continuous approximation.

However, when applying the particle discretization:

$$\langle A \rangle_i = \sum_j \frac{m_j}{\rho_j} A_j W_{ij} \quad (3.14)$$

the consistency properties are affected. For first-order consistency, the discretized system must satisfy:

$$\sum_j \frac{m_j}{\rho_j} W_{ij} = 1 \quad (0\text{th-order consistency}) \quad (3.15)$$

$$\sum_j \frac{m_j}{\rho_j} (\mathbf{x}_j - \mathbf{x}_i) W_{ij} = \mathbf{0} \quad (1\text{st-order consistency}) \quad (3.16)$$

These conditions are rarely satisfied exactly in practice due to particle disorder and boundary effects. Consequently, the particle discretization typically reduces the order of consistency. However, Bender explained that this loss of formal consistency order does not necessarily translate to poor approximation quality in practice if the discretized equations were, for instance, mass conservative.

Let us consider the following two-dimensional example where a continuous medium has been discretized into a finite number of particles that are uniformly distributed in the domain. If one wishes to express an arbitrary function along a portion of this discretized domain—specifically, one may use kernel interpolation to approximate the values of the function at each point along the line by using the neighbouring particles that constitute the domain. With a sufficient number of particles, one can accurately approximate any function along this line, as illustrated in Figure 3.2. It is worth noting that as the number of particles approaches infinity, one effectively retrieve the properties of a fully continuous medium.

The approximation accuracy remains generally high for well-behaved particle distributions. As explained by Bender, local conservation properties (such as momentum conservation) often prove more important than formal consistency order for stable physical simulations.

In order to manage future fluid simulations using the SPH method, our interest was directed toward the SPlisHSPlasH project for several reasons. First, their open source code has its core available for everyone, allowing us to examine the internal workings of the code. Additionally, the same team develops both mathematical papers about specific aspects of SPH simulation (boundary handling, pressure solver, etc.) and their numerical implementation, providing extensive documentation in both scientific and computational contexts.

The relevance of SPH methods, and particularly SPlisHSPlasH implementations, are demonstrated by the growing diversity of applications in recent literature. These range from geophysical simulations such as three-dimensional lava dynamics modelling [13] to industrial applications including adhesive flow simulation using advanced rheological models [14], and even specialized environmental studies like oil-ice interaction modelling [15]. The method has been used for 3D printing applications with viscoelastic fluids [16].

Furthermore, they offer a wide range of developed tools, including various pressure solvers (DFSPH, WCSPH, PCSPH, etc.), different boundary handling methods (particle-based, density Maps, volume Maps), as well as rigid body simulation (both static and dynamic) with fluid interaction capabilities. The importance of accurate fluid-solid coupling has been increasingly recognized in recent research, with developments in extended partitioned methods for conservative coupling [17], demonstrating the continued relevance of SPlisHSPlasH usage for multiphysics simulation.

The recent optimizations regarding neighbourhood search using a so-called z curve for memory allocation, some AVX optimization, and GPU-supported features are particularly important, especially when results require high particle resolution and thus more computational time. Contemporary research continues to improve the SPH computational efficiency, with studies focusing on GPU optimizations for predictive-corrective particle-based methods [18], efficient non-iterative approaches with variable smoothing length [19], and surface tension acceleration techniques using particle classification [20].

Finally, the team remains very active and enhances their code continuously, which increases confidence in their work.

From this perspectives, it is proposed in this chapter to present the mathematical foundations of their key algorithms, structured along the simulation workflow. Beginning with particle interaction fundamentals (kernel selection and neighbourhood search, Sections 4.1-4.2), one proceeds to core physics (pressure solver and time integration, Sections 4.3-4.4), then turbulence modelling (viscosity and vorticity, Sections 4.5-4.6). Free-surface phenomena follow via the drag method (Section 4.7), leading to boundary interactions (Volume Maps and XSPH stabilization, Sections 4.8-4.9), and conclude with dynamic rigid body coupling (Section 4.10). This progression from local particle dynamics to global system interactions reflects the hierarchical nature of SPH simulations.

4.1 Kernel used

The kernel which one has decided to use is the well known cubic kernel which offers multiple reliable properties. First, a null kernel value is obtained at the end of the radius support (which was not the case for the classical gaussian type), so nay discontinuousness occur from this interpolation. Also, the cubic shape of the kernel offers a smooth transition from the minimal to the maximal distance. Sharper kernel would lead to some instabilities due to the too high transition¹.

The kernel used in the code has two different version whether the simulation is two or three-dimensional.

3D version

The kernel and its gradient read

$$W(\mathbf{r}) = \begin{cases} k_{3D} (6q^3 - 6q^2 + 1) & \text{if } 0 \leq q \leq 0.5 \\ k_{3D} \cdot 2(1 - q)^3 & \text{if } 0.5 < q \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

where $q = \|\mathbf{r}\|/h$.

And

$$\nabla W(\mathbf{r}) = \begin{cases} \frac{l_{3D} \cdot q(3q - 2)}{h^2} \mathbf{r} & \text{if } 0 < q \leq 0.5 \\ -\frac{l_{3D} \cdot (1 - q)^2}{h^2} \mathbf{r} & \text{if } 0.5 < q \leq 1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

with $k_{3D} = \frac{8}{\pi h^3}$ and $l_{3D} = \frac{48}{\pi h^3}$

2D version

The kernel and its gradient read

$$W(\mathbf{r}) = \begin{cases} k_{2D} (6q^3 - 6q^2 + 1) & \text{if } 0 \leq q \leq 0.5 \\ k_{2D} \cdot 2(1 - q)^3 & \text{if } 0.5 < q \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

¹Nevertheless, sharp kernels could be desired in specific applications. As the microfluidic domain with high surface tension contribution. Where adhesion effects have to be high near boundaries but relatively inexistent elsewhere.

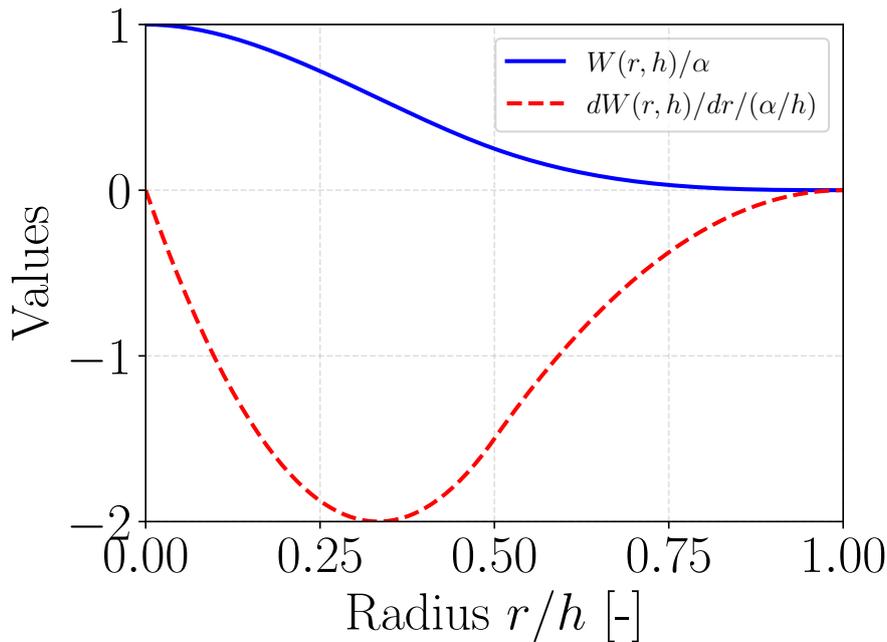


Figure 4.1: Cubic kernel and its derivative with their respective normalisation factor (3D version).

where $q = \|\mathbf{r}\|/h$.

And

$$\nabla W(\mathbf{r}) = \begin{cases} \frac{l_{2D} \cdot q(3q - 2)}{h^2} \mathbf{r} & \text{if } 0 < q \leq 0.5 \\ -\frac{l_{2D} \cdot (1 - q)^2}{h^2} \mathbf{r} & \text{if } 0.5 < q \leq 1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

with $k_{2D} = \frac{40}{7\pi h^2}$ and $l_{2D} = \frac{240}{7\pi h^2}$.

4.2 Neighbourhood research

As introduced in Section 3.3.2 with Equation (3.14), the neighbourhood search constitutes the most computationally intensive part of the algorithm, since the number of interacting particle pairs is significantly larger than the number of particles themselves. It becomes clear that efficient spatial subdivision schemes are crucial for achieving reasonable computational performance.

This section summarises the method involved in the SPlisHSPlasH software, which is explained in more details by Ihmsen et al.[21].

4.2.1 Cell-based methods

The *cell-based* spatial subdivision schemes prove to be the most efficient approach for simulations with uniform support radius h (with h kept constant). The construction cost of a uniform grid is $O(n)$, while any item can be accessed in $O(1)$. This makes it superior to other so-called hierarchical structures like *kd-trees* which have $O(n \log n)$ construction and $O(\log n)$ access costs, respectively.

One assigns each particle i to a cell, indexed by coordinated (k, l, m) , depending on the particle position (x, y, z) :

$$(k, l, m) = \left\lfloor \frac{x - x_{\min}}{h} \right\rfloor, \left\lfloor \frac{y - y_{\min}}{h} \right\rfloor, \left\lfloor \frac{z - z_{\min}}{h} \right\rfloor \quad (4.1)$$

where h denotes the cell size. This precise size allows us to look neighbouring particles only in the same cell and in the 26 (or 8 cells in 2D simulations) adjacent cells .

It is worth noting that the cell size has a direct impact on the number of potential neighbours. Obviously, the smaller the cell size, the smaller the number of potential pairs. However, with cell sizes smaller than h , the number of cells to query increases significantly. This might slow down the neighbourhood query due to a larger number of memory lookups. Hence, the optimal cell size is the influence radius h itself, as smaller cell sizes increase memory transfer costs despite reducing the number of potential pairs to test.

Spatial hashing

Since a particle could be anywhere in a space $\forall(x, y, z) \in \mathbb{R}^3$ that could be arbitrary large, one numerically needs to assign this particle in a cell of finite dimensions (k, l, m) . The function that takes in input (x, y, z) and returns (k, l, m) is called a *hash function*

In their work, the hash function used to create the hash table of size m reads:

$$c = \left[\left(\left\lfloor \frac{x}{d} \right\rfloor \cdot p_1 \right) \text{ xor } \left(\left\lfloor \frac{y}{d} \right\rfloor \cdot p_2 \right) \text{ xor } \left(\left\lfloor \frac{z}{d} \right\rfloor \cdot p_3 \right) \right] \text{ mod } m \quad (4.2)$$

where $p_1 = 73856093$, $p_2 = 19349663$, and $p_3 = 83492791$ are large prime numbers chosen specifically to minimize hash collisions. These numbers are inspired from the paper of Teschner et al. [22].

However, due to the *mod* operation, different cells can be mapped to the same hash cell. This effect is called the *hash collision*, which slows down the neighbourhood research. The number of hash collisions can be reduced by increasing the size of the hash table. Ihmsen found that a hash table size of two times the number of particles is appropriate.

Compact hashing optimization

The major issue about the spatial hashing is that lots of cells are empty, resulting in unnecessary memory allocation. To address this limitation, Ihmsen proposes to use a secondary data structure storing only non-empty cells, the so-called compact hashing approach.

It is important to understand that this compact list of used cells already lowers the memory transfer. However, the hash function is not designed to maintain spatial locality. In other words, particles that are closed in space are not necessarily closed in memory.

4.2.2 Z-index sort method

The Z-index sort method addresses the spatial locality raised just above. Rather than ordering cells dimension by dimension, one employs a space-filling Z-curve that preserves spatial locality through its self-similar recursive block structure.

Instead of using the standard cell indexing:

$$c = k + l \cdot K + m \cdot K \cdot L \quad (4.3)$$

the Z-index sort computes cell indices using bit-interleaving to create a Z-curve ordering. This indexing scheme IS spatially compact, since it follows a recursive Z-pattern that preserves neighbourhood relationships. Thus, particles that are close in space are likely to be close in memory. This method is presented in Figure 4.2 and 4.3.

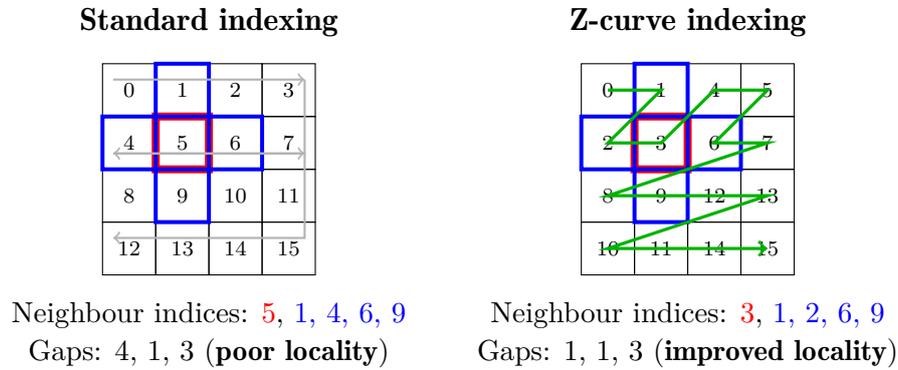


Figure 4.2: Comparison between standard row-major indexing and Z-curve indexing for a 4x4 grid. The red cell represents the target cell, and blue cells are its spatial neighbours. The Z-curve preserves better spatial locality by keeping neighbouring cells closer in memory index.

Recursive Z-pattern construction

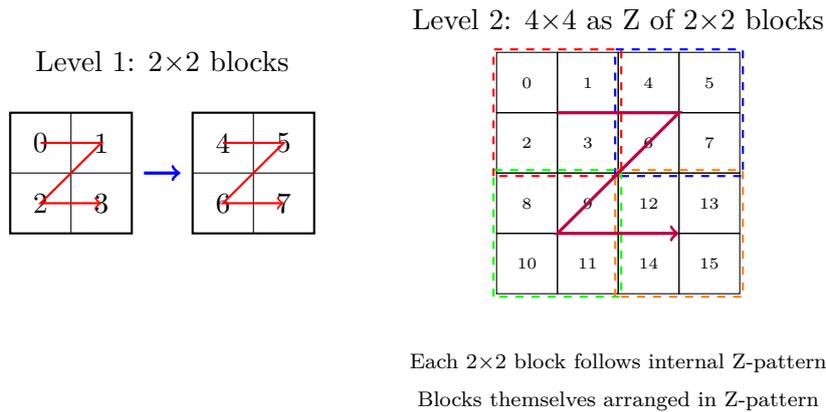


Figure 4.3: Recursive construction of the Z-curve showing how 2x2 blocks are arranged in a Z-pattern, and then these blocks themselves follow a Z-pattern to form larger structures.

4.3 Pressure solver: divergence-free SPH method (DFSPH)

The Divergence-Free Smoothed Particle Hydrodynamics (DFSPH) method represents a significant advancement in particle-based fluid simulation methodology. Developed by Bender and Koschier (2015, 2017), this approach addresses one of the fundamental challenges in computational fluid dynamics: the accurate enforcement of incompressibility constraints in Lagrangian particle systems. Unlike traditional SPH approaches that rely on equation of state formulations or single-phase incompressibility projections, DFSPH implements a two-step projection algorithm that enforces the physical incompressibility of fluids through complementary constraints.

4.3.1 Mathematical foundations

Recalling the two Navier-Stokes equations for incompressible fluids:

$$\frac{D\rho}{Dt} = -\rho\nabla \cdot \mathbf{v} = 0 \quad (4.4)$$

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu\nabla^2 \mathbf{v} + \mathbf{f}_{ext} \quad (4.5)$$

Physically, $\frac{D\rho}{Dt} = 0$ traduces the mass conservation through constant density while $\nabla \cdot \mathbf{v} = 0$ traduces a more geometrical consideration indicating that the outgoing quantity of matter inside an infinitesimal volume is exactly the same as the ingoing quantity of matter.

The key insight of the DFSPH is to actually implement the velocity integration through Equation (4.5) while using Equation (4.4) as a physical constraint by the two interpretation raised above. These complementary constraints enable superior stability and accuracy.

4.3.2 Algorithm Structure

Figure 4.4 shows the main steps of the DFSPH algorithm per timestep. Essentially, it follows a predictive-corrective scheme:

- **Before prediction:** the divergence solver corrects velocities to ensure $\nabla \cdot \mathbf{v} = 0$ (required for viscosity, see Section 4.5).
- **After prediction:** the density solver corrects densities back to ρ_0 .

After divergence correction, the non-pressure forces are applied. Then, positions are temporarily updated using these preliminary velocities which requires recomputing neighbour lists and densities (Step 6), including boundary effects (Step 7).

However, the temporary position update breaks incompressibility ($\rho \neq \rho_0$) and thus need to be reinforced. The density solver (Step 8) fixes this by adjusting velocities to enforce $\rho^{n+1} = \rho_0$. Eventually, the final positions are computed using these corrected velocities (Step 9).

Even though the two solvers are mathematically equivalent (through Equation (4.4)), they address different physical aspects ensuring incompressibility at each steps of the algorithm.

Both use Jacobi iterations (detailed in Section A.1). Iterations stop when residuals meet thresholds. For the density solver, the convergence criterion is:

$$\frac{1}{n} \sum_i |\rho_0 \cdot r_i| \leq \eta \cdot \rho_0 \quad (4.6)$$

where r_i is the residual of particle i , and η is the tolerance (e.g., $\eta = 0.01$ for 1% max density error).

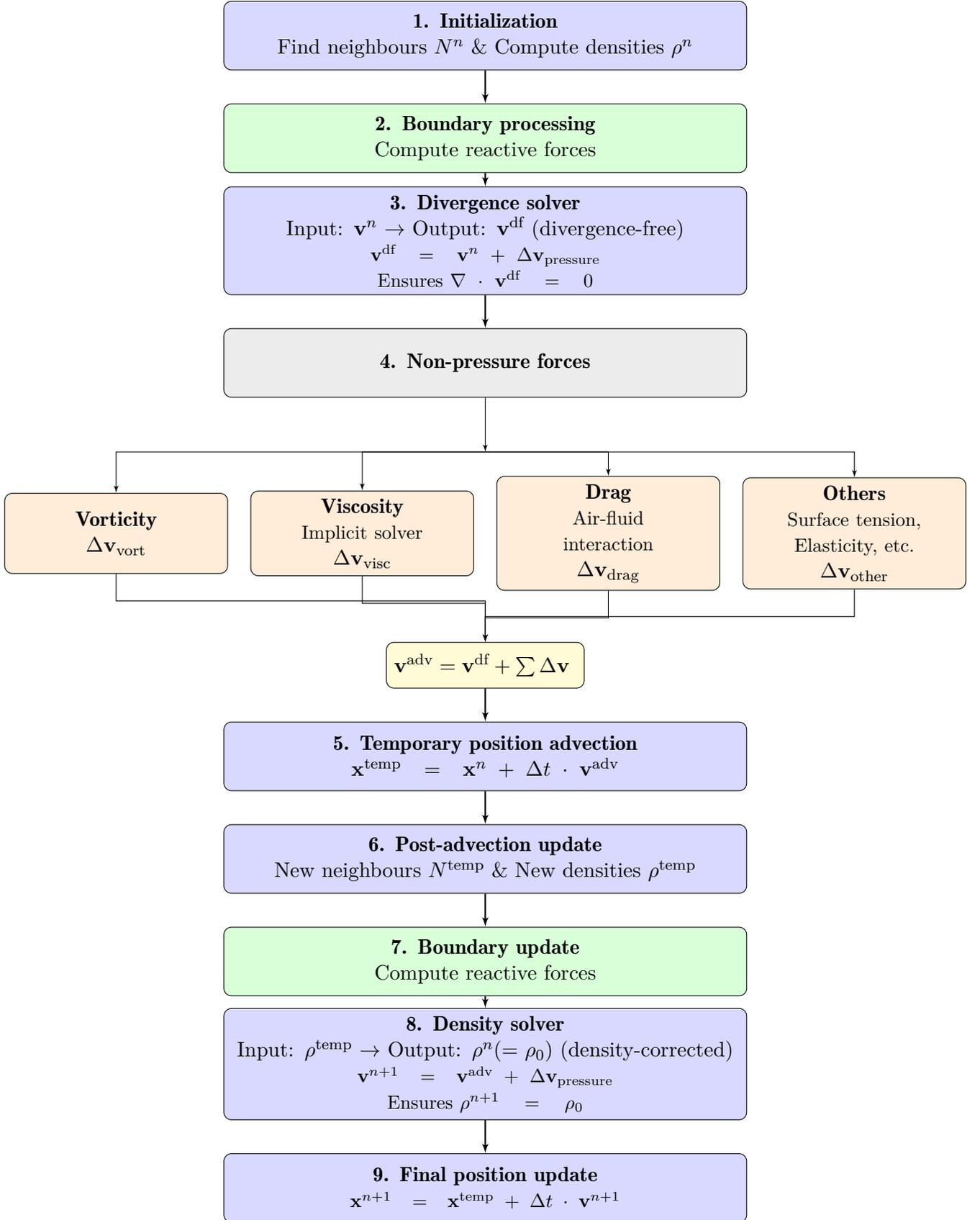


Figure 4.4: Structure of DFSPH algorithm for a single timestep. Key steps: (1-2) Initialization and boundaries, (3) Divergence solver ensuring $\nabla \cdot \mathbf{v} = 0$, (4) Computation of non-pressure forces, (5) Temporary advection, (6-7) Post-advection update and boundaries, (8) Density solver ensuring $\rho = \rho_0$, (9) Final position update with corrected velocities.

4.4 Time integration scheme

The choice of the integration scheme is crucial in order to reach a peak of performance balancing both the accuracy of the outputs and the associated computational time. Two different type of Euler schemes exist in the SPLisHSPlasH software, namely *explicit* and *implicit* Euler scheme. The differences between each other lie in their formulation. Let us assume that one desires to update the position of a particle $x(t)$ with a given timestep Δt based on its velocity $v(t)$, the two numerical schemes then read:

Explicit Euler:

$$v(t + \Delta t) = v(t) + \Delta t a(t) \quad \& \quad x(t + \Delta t) = x(t) + \Delta t v(t)$$

Implicit Euler:

$$v(t + \Delta t) = v(t) + \Delta t a(t + \Delta t) \quad \& \quad x(t + \Delta t) = x(t) + \Delta t v(t + \Delta t)$$

While the first one is evidently the most easy one to compute and the one requiring the less computational time, this scheme may become unstable in certain configuration/simulation. The second one usually offers more stability in the simulation at the cost of solving an algebraic system of equations. Some of the methods presented here below will use either explicit or implicit integration scheme. Nevertheless, the stability of each other is correctly handled thanks to the CFL condition.

CFL condition

For stable simulations across varying flow conditions, adaptive time-stepping based on the CFL condition is essential. One of the CFL method involved the number of iterations inside the pressure solvers to take into account the difficulty of the solver to reach a specific threshold.

$$\Delta t_{n+1} = \min \left(\Delta t_{CFL}, \Delta t_n \cdot \begin{cases} 0.9 & \text{if } iter > 10 \\ 1.1 & \text{if } iter < 5 \\ 1.0 & \text{otherwise} \end{cases} \right) \quad (4.7)$$

where:

$$\Delta t_{CFL} = 0.4 \cdot C_{factor} \cdot \frac{2r}{\sqrt{v_{max}}} \quad (4.8)$$

$$v_{max} = \max_i \|v_i + a_i \Delta t\|^2 \quad (4.9)$$

4.5 Viscosity implementation

As already introduced in Section 2.2, the governing equations, and more specifically the non-pressure forces are influenced by the viscosity through the following variable:

$$\mathbf{f}^{\text{visc.}} = \mu \nabla^2 \mathbf{v} \quad (4.10)$$

which can be interpreted as the velocity spatial diffusion of the fluid up to a scaling factor μ (reminding that this is the dynamic viscosity of the prescribed fluid).

Second derivative computation

In the method developed by Weiler et al. [23], they chose to approximate the Laplacian derivative as follows:

$$\nabla^2 \mathbf{u} = 2(d+2) \sum_j \frac{m_j}{\rho_j} \frac{\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|^2 + 0.01h^2} \nabla W_{ij} \quad (4.11)$$

where d is the spatial dimension and \mathbf{x}_{ij} , \mathbf{v}_{ij} are respectively the displacement and the velocity difference between particle i and j .

As mentioned by the authors, this Laplacian representation is Galilean invariant, vanishes for rigid body rotation and conserves linear and angular momentum.

Time integration consideration

It has already been mentioned that some of the SPLisHSPLasH features were using Euler implicit scheme for the time integration. This was specifically referring to this viscosity method. Indeed the method attempts to solve the following system:

$$\mathbf{v}(t + \Delta t) = \mathbf{v}^{\text{df}} + \Delta t \frac{\mu}{\rho} \nabla^2 \mathbf{v}(t + \Delta t) = \mathbf{v}^{\text{df}} + \Delta t \mathbf{A} \mathbf{v}(t + \Delta t) \quad (4.12)$$

where \mathbf{v}^{df} is the divergence-free velocity and \mathbf{A} the matrix representing the viscosity influence that takes the following form:

$$\mathbf{A}_{ij} = -2(d+2) \frac{\mu \bar{m}_{ij}}{\rho_i \rho_j} \frac{\nabla W_{ij} \mathbf{x}_{ij}^T}{\|\mathbf{x}_{ij}\|^2 + 0.01h^2}, \quad \mathbf{A}_{ii} = - \sum_j \mathbf{A}_{ij} \quad (4.13)$$

where \bar{m}_{ij} is the average mass to get a symmetric and positive definite linear system. One may mention the viscosity solver is applied in the *non-pressure* part of the DFSPH algorithm just after the divergence corrector. This ensures that the velocity field is effectively divergence-free (main assumption for the viscosity solver). The order of operations displayed, in the DFSPH paper from Bener et al.[24], has been modified since the appearance of the paper in 2015, as in their diagram it is mentioned that the density and divergence solvers are applied after the non-pressure contributions.

4.6 Turbulence representation through vorticity : micropolar model

As previously introduced in Section 2.6, turbulence representation in fluid simulation is challenging to handle. Even more due to numerical diffusion inherent in SPH methods, which leads to vorticity loss and damping of turbulent details. To address this issue, one considers here a micropolar material model that introduces additional equations at the particle level to preserve and generate turbulent motion.

The micropolar model introduced by Bender et al. [25] generalizes the classical Navier-Stokes equations by considering that the fluid has a microstructure consisting of rigid, spherical microelements. This model is specifically designed for inviscid fluids such that vorticity is not artificially dissipated, unlike classical SPH simulations.

The key innovation of this method lies in the introduction of a new field variable ω describing the microrotation (angular velocity) at each particle. This microrotation field can exist independently of the linear velocity field and provides a source for new vortices. The governing equations are modified as follows:

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{v} + (\nu + \nu_t)\nabla \times \omega + \frac{\mathbf{f}}{\rho} \quad (4.14)$$

$$\Theta\frac{D\omega}{Dt} = \zeta\Delta\omega + (\nu + \nu_t)(\nabla \times \mathbf{v} - 2\omega) + \frac{\tau}{\rho} \quad (4.15)$$

where ν_t , Θ , ζ , and τ are respectively the kinematic transfer coefficient, the isotropic microinertia coefficient, the kinematic rotational viscosity, and the specific external torque. The coupling term $(\nu + \nu_t)\nabla \times \omega$ in the momentum equation transfers rotational motion into linear motion, while the term $(\nu + \nu_t)(\nabla \times \mathbf{v} - 2\omega)$ in the microrotation equation couples linear and angular motion.

As noted by Bender et al., the classical Navier-Stokes equations are recovered when $\nu_t = -\nu$, $\zeta = 0$, and $\tau = \mathbf{0}$, demonstrating that the micropolar model is indeed a generalization of the standard approach.

4.7 Air-fluid interaction (drag method)

Air-water interaction is usually simulated considering the air having a negligible contribution such that the water may solely being simulated in order to same additional computational time. However, the air is indeed considered as a perfect gaz but its contribution is not inexistent ! To take into account the contribution of air onto the water simulation without actually compute two phases simulation, Gissler et al.[26] proposes a model to compute drag forces in the system that considers correctly air influence. One therefore will summary in this section the key components of the method and present the considered equations and their associated main assumptions.

For modelling the drag forces acting on a fluid particle moving through air (only considering friction and not adhesion), the following quadratic drag equation is used:

$$\mathbf{F}_i^{drag} = \frac{1}{2}\rho_a v_{i,rel}^2 C_{D,i} A_i \quad (4.16)$$

where ρ_a is air density, $v_{i,rel}^2$ is the squared relative velocity vector, $C_{D,i}$ is the drag coefficient, and A_i is the exposed cross-sectional area of particle i .

Also, the relative velocity vector is computed as:

$$\mathbf{v}_{i,rel}^2 = |\mathbf{v}_a - \mathbf{v}_i| \cdot (\mathbf{v}_a - \mathbf{v}_i) \quad (4.17)$$

It is important to understand that both A_i and $C_{D,i}$ vary for each particle and depend on the actual deformation of the associated particle i . Hence, below is explained how the deformations are evaluated.

4.7.1 Particle Deformation Model

The Taylor-Analogy-Breakup (TAB) model, firstly introduced by O'Rourke et al. [27], is used to describe the deformation of fluid particles. This model relates droplet oscillation to a mass-spring system with the following differential equation:

$$\frac{d^2 y_i}{dt^2} = C_F \frac{\rho_a v_{i,rel}^2}{\rho_l L^2} - C_k \frac{\sigma}{\rho_l L^3} y_i - C_d \frac{\mu_l}{\rho_l L^2} \frac{dy_i}{dt} \quad (4.18)$$

where y_i represents the dimensionless deformation (0 = sphere, 1 = disk), ρ_l is liquid density, σ is surface tension, μ_l is liquid viscosity, and L is the droplet radius.

Gissler and al. followed the same reasoning as O'Rourke and involved assumptions to find an analytical solution for Equation 4.18. Assuming constant relative velocities, $y_o = \frac{dy_o}{dt} = 0$ and finally assuming that all particles are always deformed as strongly as possible yields:

$$y_i^{max} = \frac{C_F}{C_k C_b} \cdot We_i \cdot c_{def} = v_{i,rel}^2 \cdot y_{coeff} \quad (4.19)$$

where $We_i = \frac{\rho_a v_{i,rel}^2 L}{\sigma}$ is the Weber number, and c_{def} is a precomputed factor.

4.7.2 Drag Coefficient Calculation

Another paper from Liu and al. [28] had been involved to compute the drag coefficient being considered as an interpolation between a sphere and a disk based on deformation:

$$C_{D,i}^{Liu} = C_{D,i}^{sphere} (1 + 2.632 y_i^{max}) \quad (4.20)$$

where $C_{D,i}^{sphere}$ is calculated based on Reynolds number:

$$C_{D,i}^{sphere} = \begin{cases} \frac{24}{Re_i} (1 + \frac{1}{6} Re_i^{2/3}), & Re_i \leq 1000 \\ 0.424, & Re_i > 1000 \end{cases} \quad (4.21)$$

with $Re_i = \frac{\rho_a |v_{i,rel}| L}{\mu_a}$.

For particles within a cluster, the coefficient is adjusted depending on the number of neighbours:

$$C_{D,i} = \left(1 - \frac{\min(\frac{2}{3} n_{full}, n)}{\frac{2}{3} n_{full}} \right) C_{D,i}^{Liu} + \frac{\min(\frac{2}{3} n_{full}, n)}{\frac{2}{3} n_{full}} \quad (4.22)$$

where n_{full} is the maximum number of neighbours a particle may have.

4.7.3 Cross-Sectional Area

Finally, the cross-sectional area is evaluated based on two consideration. The so-called *unoccluded* and *occluded* surfaces being respectively the areas associated to air contribution and other fluid particles (or rigid boundaries) contribution.

Unoccluded area

First, let us consider the unoccluded part. For a given relative velocity $v_{i,\text{rel}}$ between particle i and air, the initial spherical shape of the particle will be squeezed such that, in a 2D slice in the middle of the particle, the initial circle cross sectional is now more an ellipse. The deformation between the initial radius (from the circle cross section) and the increased radius of the ellipse is calculated from:

$$x_i = C_b L y_i^{\max} \quad (4.23)$$

such that the final modified cross section is calculated as followed:

$$A_i^{\text{droplet}} = \pi (L + C_b L y_i^{\max})^2 \quad (4.24)$$

This is interpolated with the standard area h^2 (good approximation if the considered particle is part of a closed surface of a larger fluid volume) based on neighbour count:

$$A_i^{\text{unoccluded}} = \left(1 - \frac{\min(\frac{2}{3}n_{\text{full}}, n)}{\frac{2}{3}n_{\text{full}}}\right) A_i^{\text{droplet}} + \frac{\min(\frac{2}{3}n_{\text{full}}, n)}{\frac{2}{3}n_{\text{full}}} h^2 \quad (4.25)$$

Occluded area

All particles are partially exposed to air but thus also have an occluded part. To translate this nuanced behaviour, the occluded area is evaluated as:

$$A_i^{\text{occluded}} = \omega_i A_i^{\text{unoccluded}} \quad (4.26)$$

where the occlusion factor w_i is calculated geometrically and depends on both the considered neighbourhood and the actual cross section facing the relative velocity $\mathbf{v}_{i,\text{rel}}$.

To detect if a particle is at the surface, they observes neighbouring particle in a cone in the direction of $\mathbf{v}_{i,\text{rel}}$. Finally, they allows w_i to smoothly vary in the $[0, 1]$ range by considering the cosine between $\mathbf{v}_{i,\text{rel}}$ and the neighbour direction $\mathbf{x}_{i,j}$

$$w_i = 1 - \max_j \left(\frac{\mathbf{v}_{i,\text{rel}} \cdot \mathbf{x}_{ij}}{|\mathbf{v}_{i,\text{rel}}| |\mathbf{x}_{ij}|} \right) \quad (4.27)$$

4.7.4 Final computation

Finally, after evaluating y_i^{\max} and calculated the associated drag coefficient $C_{D,i}$ and cross section A_i , the Equation (4.16) representing the drag force is fully known and may be evaluated for each particle.

4.8 Boundary handling method : Volume Maps

The way boundaries are handled inside a simulation is a crucial step in building confidence in the model. Here, one proposes to delve into the details of the *Volume Maps Approach* developed by [29].

It is worth noting that the aforementioned method is an evolution of their previous model (developed by the same team) called the *Density Maps Approach*, which is itself an evolution of the primary method known as the *Particle-Based Approach*.

4.8.1 Boundary representation

First, this method is based on an implicit boundary representation. This means that the boundaries are not physically described using a set of fluid particles with points (x, y, z) , which is the case for the Particle-Based Approach (i.e., an explicit boundary representation). Rather, a specific function is used to characterize the effects of the boundary, in our case this function is called a *signed distance function* (also known as SDF). For instance, an SDF could be described using the position (x, y, z) of a particle as follows:

$$\phi(x, y, z) \begin{cases} > 0 & \text{if outside of the boundary} \\ = 0 & \text{if on the edge} \\ < 0 & \text{if inside the boundary} \end{cases} \quad (4.28)$$

By doing so, one avoids the storage of the particle coordinates of the boundaries, which is especially beneficial for large-scale simulations.

In our case, if the position of a particle is described by \mathbf{x} and if a boundary and its associated surface are defined by \mathcal{B} and $\partial\mathcal{B}$, the SDF reads:

$$\phi(\mathbf{x}) = \begin{cases} -d(\mathbf{x}, \partial\mathcal{B}) & \text{if } \mathbf{x} \in \mathcal{B} \\ d(\mathbf{x}, \partial\mathcal{B}) & \text{otherwise} \end{cases} \quad (4.29)$$

where

$$d(\mathbf{x}, \partial\mathcal{B}) = \min \|\mathbf{x} - \hat{\mathbf{x}}\| \quad \forall \hat{\mathbf{x}} \in \partial\mathcal{B}$$

is the distance travelled by the particle if it is perpendicularly projected onto the boundary surface.

4.8.2 Intersection volume

Once the concept of SDF is well understood, one may introduce the boundary volume and more specifically the boundary intersection volume. The idea is that, as a particle approaches a boundary, the neighbouring radius (i.e., the smoothing length) will first encounter the boundary, as shown in Figure 4.5. The fraction of the volume (defined by the neighbouring support) that is inside the boundary defines the so-called *intersection volume* which will be used to compute the effect of the boundary on the prescribed particle. This volume is defined as:

$$V_{\mathcal{B}}(\mathbf{x}) = \int_{\mathcal{N}(\mathbf{x})} \gamma^*[\phi(\mathbf{x}')] d\mathbf{x}' \quad (4.30)$$

where $\mathcal{N}(\mathbf{x})$ describes the neighbourhood of the particle and γ^* is a cubic extension function (also called shape function) that creates a smooth transition from the boundary to the fluid

domain and avoids discontinuities:

$$\gamma^*(\phi) = \begin{cases} \frac{C(\phi)}{C(0)} & \text{if } 0 < \phi < r \\ 1 & \text{if } \phi \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.31)$$

where C denotes the cubic spline kernel.

It is important to understand that the intersected volume is then entirely evaluated based on the SDF rather than by actually evaluating a volume integral. It will be shown later how to take into account this computed volume. Nevertheless, it is first proposed to take a look at how this volume is computed in their algorithm.

4.8.3 Mapping of the intersected volumes

It becomes clear that computing at each time step the intersected volume for each particle would lead to an unrealistic amount of time solely spent on evaluating integrals. To prevent this issue, Bender et al. more cleverly introduced a regular spatial grid in the simulation space and thus discretized the domain into cells (and nodes).

Then, before the beginning of the simulation, the intersected volumes are evaluated for each node of the grid using Gauss-Legendre quadrature and are then stored during the whole simulation. Subsequently, if a particle is near a boundary², the algorithm determines the associated cell. Subsequently, the associated nodes encapsulating the particle have their signed distance $d(\mathbf{x}, \partial\mathcal{B})$ extracted. Based on these nodes, the actual particle SDF is obtained through interpolation thanks to the shape function $\gamma^*[\phi(\mathbf{x})]$.

As mentioned by the authors, it is worth noting that each cell has 32 nodes to compute the cubic shape functions of Serendipity type to get a volume map.

As shown in Figure 4.6, the nodes have their precomputed intersection volume values and will be used to interpolate the values for the particles.

4.8.4 Final boundary contribution

There is a subtle but crucial consideration that has not been completely explained up to now. One initially stated, in the DFSPH section, that the reactive forces (due to boundaries) were evaluated. Actually, the authors propose a more clever idea. Indeed, since the intersection volume is already computed, they use this to directly evaluate the density change, rather than adding an additional non-pressure force to the system that would modify the velocity and then finally the density.

Density computation

Once the intersection volume is computed, one needs to add the contribution of this latter to the density of each particle inside the support radius of the boundary. Formally, the total density is computed by an integral that can be decomposed into contributions from the fluid domain \mathcal{F} and the boundary domain \mathcal{B} :

$$\rho(\mathbf{x}) = \int_{\mathcal{N}(\mathbf{x})} \rho(\mathbf{x}') W(\|\mathbf{x} - \mathbf{x}'\|, h) d\mathbf{x}' = \rho_{\mathcal{F}}(\mathbf{x}) + \rho_{\mathcal{B}}(\mathbf{x}) \quad (4.32)$$

²the support radius of any boundary is the same as the one for the particle, namely $h = 4r$ where r is the particle radius.

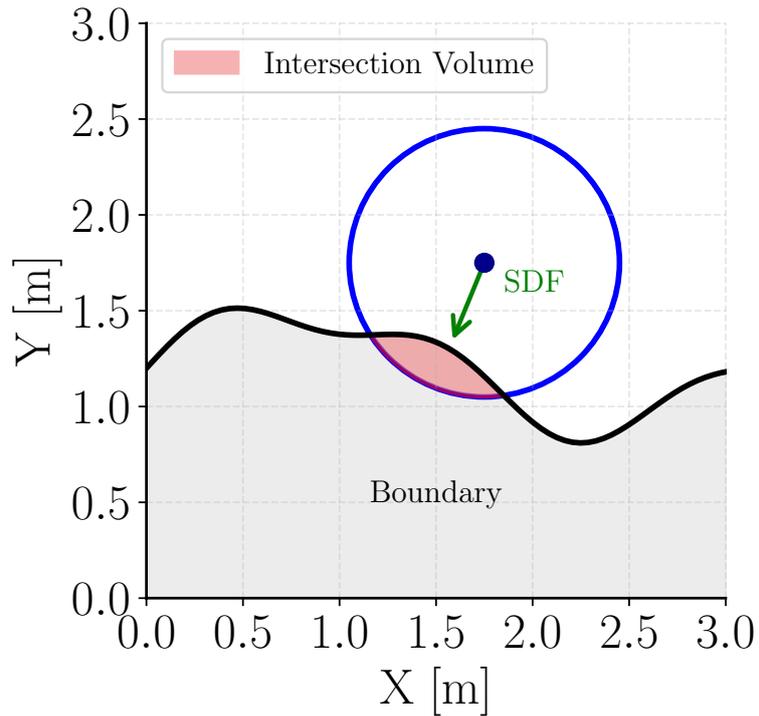


Figure 4.5: Intersection volume of a particle near boundaries. One observes that the intersection volume does have a relationship with the signed distance function. The neighbouring radius is not properly scaled.

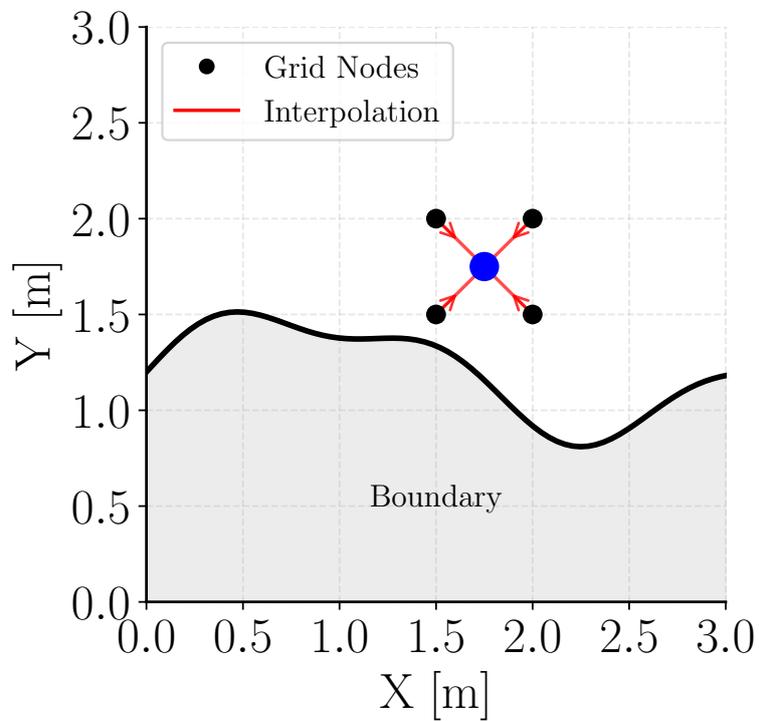


Figure 4.6: Intersection volume of a particle evaluated from interpolated nodes which define the grid cells.

with

$$\rho_{\mathcal{B}}(\mathbf{x}) = V_{\mathcal{B}}(\mathbf{x})\rho_0 W(\|\mathbf{x} - \mathbf{x}^*\|, h) \quad (4.33)$$

such that the total density finally reads:

$$\rho(\mathbf{x}) = \sum_j m_j W(\|\mathbf{x} - \mathbf{x}_j\|, h) + V_{\mathcal{B}}(\mathbf{x})\rho_0 W(\|\mathbf{x} - \mathbf{x}^*\|, h) \quad (4.34)$$

4.8.5 Pressure force computation

Similarly, the pressure force exerted by the boundary on a fluid particle i follows the standard SPH formulation:

$$\mathbf{F}_{i \leftarrow \mathcal{B}}^p = -m_i V_{\mathcal{B}}(\mathbf{x}_i) \rho_0 \left(\frac{p_i}{\rho_i^2} + \frac{\tilde{p}}{\tilde{\rho}^2} \right) \nabla W(\|\mathbf{x}_i - \mathbf{x}^*\|, h) \quad (4.35)$$

where \tilde{p} and $\tilde{\rho}$ are the pressure and density at the boundary, respectively. For standard implementations, they uses $\tilde{p} = p_i$ and $\tilde{\rho} = \rho_0$.

4.8.6 Two-way coupling

Finally, when dealing with dynamic rigid boundaries, the volume Maps approach facilitates two-way coupling between the fluid and the boundary. This is achieved by applying the negative force $\mathbf{F}_{\mathcal{B} \leftarrow i}^p = -\mathbf{F}_{i \leftarrow \mathcal{B}}^p$ at position \mathbf{x}^* on the boundary object.

4.9 XSPH numerical dissipation

For incompressible simulations, numerical instabilities may arise during computations, leading to unphysical behavior. To address this, Monaghan et al.[30] introduced the XSPH coefficient in the velocity time integration, which artificially diffuses momentum:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i + \epsilon \sum_j \frac{m_j}{\rho_j} (\mathbf{v}_i - \mathbf{v}_j) W_{ij} \quad (4.36)$$

Here, ϵ is an empirical coefficient manually tuned to ensure particles move with velocities closer to the neighborhood average, thereby suppressing non-physical fluctuations.

A similar coefficient $\epsilon_{\text{boundary}}$ enforces no-slip conditions at walls for viscous flows.

However, excessively high ϵ values induce artificial clustering in the fluid. Thus, ϵ should be kept *as small as possible*. As explained by Monaghan himself in a paper 10 years after his first work [31], a maximal value $\epsilon = 0.5$ may be used prevent undesired local velocity fluctuations but $\epsilon > 1$ leads to instabilities.

That said, for too low spatial resolution simulations, this term is critical since insufficient particles underestimate density, causing overdetermined pressure gradients and unrealistically high velocities. On the other hand, too high spatial resolution simulations also require specific XSPH value, but this topic is discussed in more details in Section 6.3.3.

4.9.1 Remark on the physical interpretation of XSPH

In his original work, Monaghan associates the ϵ coefficient with dispersion (modifying wave phase velocities) rather than dissipation. However, a mathematical analysis of Equation (4.36) brings another perspective.

Rewriting the SPH summation as a continuous integral:

$$\sum_j \frac{(\mathbf{v}_j - \mathbf{v}_i) W_{ij}}{\rho_j} \rightarrow \int \frac{(\mathbf{v}(\mathbf{r}') - \mathbf{v}(\mathbf{r})) W(|\mathbf{r}' - \mathbf{r}|, h)}{\rho(\mathbf{r}')} d\mathbf{r}',$$

and performing a Taylor expansion of $\mathbf{v}(\mathbf{r}')$ around \mathbf{r} up to second order:

$$\mathbf{v}(\mathbf{r}') \approx \mathbf{v}(\mathbf{r}) + \nabla \mathbf{v} \cdot (\mathbf{r}' - \mathbf{r}) + \frac{1}{2} (\mathbf{r}' - \mathbf{r})^T \cdot \nabla^2 \mathbf{v} \cdot (\mathbf{r}' - \mathbf{r}) + \mathcal{O}(h^3),$$

shows that:

- Constant terms $\mathbf{v}(\mathbf{r})$ and $\mathbf{v}(\mathbf{r}')$ cancel out.
- Linear terms in $(\mathbf{r}' - \mathbf{r})$ vanish due to the kernel symmetry (confirmed since one uses a cubic kernel in this thesis).

The remaining quadratic term is:

$$\int \frac{1}{2} (\mathbf{r}' - \mathbf{r})^T \cdot \nabla^2 \mathbf{v} \cdot (\mathbf{r}' - \mathbf{r}) W d\mathbf{r}'.$$

This term is proportional to the velocity Laplacian $\nabla^2 \mathbf{v}$, which acts as a spatial smoothing operator analogous to artificial viscosity in Navier-Stokes equations.

4.10 Dynamic bodies handling: position-based approach

One of the most remarkable features of SPLisHSPLasH is its implementation of fluid-rigid body interaction handling, which provides an ideal framework for the objectives of this thesis. It is worth noting that the same team behind SPLisHSPLasH has developed *Position Based Dynamics*, a dedicated approach for dynamic body handling. As the name suggests, this method focuses on the displacements of prescribed bodies rather than the forces/torques applied to them. Multiple papers have already been published using this software, such as Akinci et al. [32] for simulating rigid-fluid interactions or Bender et al. [33] who demonstrated the capabilities of the software across numerous examples including soft bodies and cloth simulation.

This section describes the mathematical foundations of the method, with specific emphasis on its application in fluid dynamics. While the PBD software is a stand-alone solution used in computer graphics for simulating clothes, deformable bodies, and more, this section will concentrate on the core principles relevant to fluid applications. One mentions that both first and second order time integration schemes are available for this software.

In SPLisHSPasH algorithm (and also in the majority of known other software), the most relevant variable is the velocity which is processed and updated using Navier-Stokes equations. In this case, the position is solely obtained through time integration but the velocity remains the variable of interest. Conversely, in the PBD algorithm, the important processes variable is actually the position of the objects (on which geometrical constraints will be applied) with the velocity being uniquely the variable involved to communicate between the fluid solver. This conceptual key difference explains the name of their solid solver.

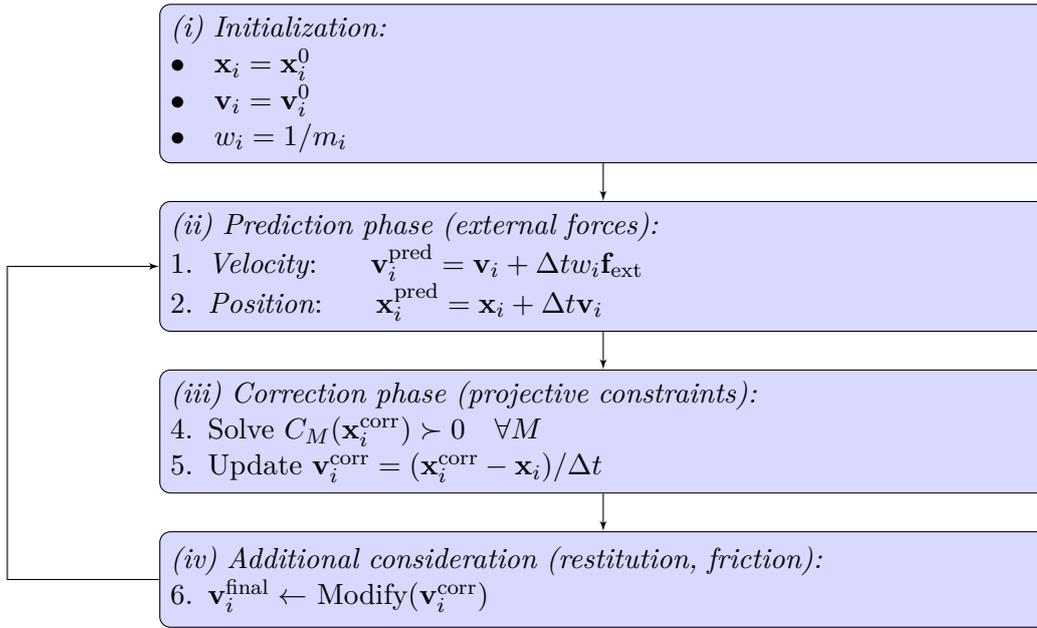


Figure 4.7: Complete PBD algorithm.

4.10.1 Fluid-solid particles coupling

The bridge between the fluid software SPLisHSPlasH and the solid software Position Based Dynamics lies in the consideration of forces and torques. Indeed, as introduced in the previous Section 4.8.6, when a fluid particle i interacts with a boundary (represented by a volume map in our cases), the pressure force exerted by the fluid on the solid follows the principle of action-reaction. Recalling the expression of the pressure force previously established in Equation (4.35):

$$\mathbf{F}_{i \leftarrow \mathcal{B}}^p = -m_i V_{\mathcal{B}}(\mathbf{x}_i) \rho_0 \frac{p_i}{\rho_i^2} \nabla W(|\mathbf{x}_i - \mathbf{x}^*|, h) \quad (4.37)$$

with the corresponding force applied to the rigid body at point \mathbf{x}^* (namely, the closest position from the considered particle) being:

$$\mathbf{F}_{\mathcal{B} \leftarrow i}^p = -\mathbf{F}_{i \leftarrow \mathcal{B}}^p \quad (4.38)$$

This force also generates a torque around the centre of mass of the rigid body:

$$\mathbf{T}_{\mathcal{B} \leftarrow i} = (\mathbf{x}^* - \mathbf{x}_{\text{cm}}) \times \mathbf{F}_{\mathcal{B} \leftarrow i}^p \quad (4.39)$$

Thus, once the external forces/torques are known for each software component, they may compute their fluid/solid particles' evolution independently of each other. Once the positions of every particle in the system are updated, new boundary forces must be considered, and thus the particles' displacements are again modified, and so on.

4.10.2 PBD general mathematical description

Let us now dive deeper into the PBD algorithm. As shown in Figure 4.7, after the bodies have been initialized, the positions and velocities of the bodies are first predicted by considering only the external forces. Then, one needs to take into account the constraints that constitute the solid bodies' interface. Indeed, the particles constituting the bodies are subject to internal

forces that maintain the geometry of the bodies (in the case of rigid bodies). The corrector phase (iii) thus adjusts the values of the predicted positions \mathbf{x}_i such that they satisfy the set of M constraints. Once the corrected positions are obtained, the associated velocities are updated. Finally, if one considers friction of the boundaries or some restitution energy (to produce bouncing effects after collisions), the velocities are further adjusted considering these effects.

4.10.3 Projective constraint solution

The corrective phase requires the solution of a complex system of M constraints with $3N$ unknowns. The three main challenges are the asymmetry of the system, the non-linearities of the constraints, and the potential over-constraining of the system. To solve this system, the non-linear Gauss-Seidel method is used in the code. This iterative method manages each constraint one by one. Firstly, the non-linearities of the constraints are handled by considering a solution in the neighbourhood of the current solution:

$$\mathcal{C}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathcal{C}(\mathbf{x}) + \nabla_{\mathbf{x}}\mathcal{C}(\mathbf{x}) \cdot \Delta\mathbf{x} \succ 0 \quad (4.40)$$

where \succ means either $=$ or \geq . It is important to highlight that the only unknowns in Equation (4.40) are $\Delta\mathbf{x}$ since $\mathcal{C}(\mathbf{x})$ and $\nabla_{\mathbf{x}}\mathcal{C}(\mathbf{x})$ are evaluated before solving the system. The constraints cannot be arbitrary; otherwise, the system could potentially be over-constrained, and no solutions would be obtained. To address this issue, Bender et al. proposed to only consider the relative displacement $\Delta\mathbf{x}$ to be proportional to $\nabla\mathcal{C}(\mathbf{x})$:

$$\Delta\mathbf{x} = \lambda\nabla\mathcal{C}(\mathbf{x}) \quad (4.41)$$

where λ is a Lagrange multiplier, thus giving 3 equations for each constraint. This way, one has $M = 3N$, and the linear system may be solved. It is demonstrated in the Appendix that this assumption preserves both linear and angular momentum. Solving the system this way, the asymmetry behaviour is not problematic since each constraint (row of the matrix system) is solved individually. Also, it is important to mention that the solver updates each particle position between solving each constraint, which means that linearizations are performed again between each constraint to account for the current position.

Contact Constraint

The specific constraints considered in the system above are distance constraint functions ensuring that fluid particles do not travel through the solid boundaries:

$$C(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{n} - \delta \geq 0 \quad (4.42)$$

where \mathbf{n} is the normal at the contact point and δ represents the minimum separation distance labelled *contactTolerance*. The behaviour of the resolution of the constraints above is manually controlled by the *contactStiffnessParticleRigidBody* (rigid-rigid body interactions) or *contactStiffnessRigidBody* (fluid-rigid body interactions) parameters through:

$$\mathbf{x}_i - \mathbf{x}_j = -k_c \frac{C(\mathbf{x}_i, \mathbf{x}_j)}{\nabla_{\mathbf{x}_i}C(\mathbf{x}_i, \mathbf{x}_j)^T \nabla_{\mathbf{x}_i}C(\mathbf{x}_i, \mathbf{x}_j)} \nabla_{\mathbf{x}_i}C(\mathbf{x}_i, \mathbf{x}_j) \quad (4.43)$$

where k_c represents the contact stiffness coefficient.

4.10.4 Additional effects

Coefficient of restitution

The coefficient of restitution e (parameter `restitution` in the code) handles the energy balance for collisions between fluid and rigid bodies. It is expressed mathematically as:

$$\mathbf{v}_{\text{rel}}^+ = -e \cdot \mathbf{v}_{\text{rel}}^- \quad (4.44)$$

where $\mathbf{v}_{\text{rel}}^-$ and $\mathbf{v}_{\text{rel}}^+$ are the relative velocities before and after collision in the normal direction, respectively. This equation may be reformulated as a velocity correction:

$$\Delta \mathbf{v} = -(1 + e) \cdot (\mathbf{v}_{\text{rel}} \cdot \mathbf{n}) \cdot \mathbf{n} \quad (4.45)$$

After resolving position constraints, meaning after phase (iii) from Figure 4.7, the velocity update incorporating restitution becomes:

$$\mathbf{v}_i^{n+1} = \frac{\mathbf{x}_i^{n+1} - \mathbf{x}_i^n}{\Delta t} \cdot e \quad (4.46)$$

where \mathbf{x}_i^{n+1} is the projected position after constraint resolution and \mathbf{x}_i^n is the initial position. This operation thus falls inside phase (iv).

Friction integration

The Coulomb friction model is implemented for fluid-solid and solid-solid interactions, regulating tangential forces:

$$\|\mathbf{f}_T\| \leq \mu \|\mathbf{f}_N\| \quad (4.47)$$

where μ corresponds to the friction coefficient (`friction` in the configuration). The tangential friction force is computed as:

$$\mathbf{f}_T = -\mu \cdot \|\mathbf{f}_N\| \cdot \frac{\mathbf{v}_{\text{rel},T}}{\|\mathbf{v}_{\text{rel},T}\|} \quad (4.48)$$

This complete formulation captures the complex interaction between fluid particles and rigid bodies, accounting for pressure forces, contact constraints, restitution effects, and frictional behaviour at the fluid-solid interface. Hence, the tangential force is directly considered as a normal force. It is important to note that the forces are not directly applied in the system. Rather, they are considered as constraints over the particles' positions. Hence, the friction integration arises in the predictor phase (ii).

Data treatment procedure

Data process is a major consideration in CFD and should be carefully explained beforehand. Hence, this chapter focuses on the philosophy of the methods employed.

Even though the SPH method is based on a Lagrangian approach which naturally allows us to analyse the properties of a given particle both over time and in space, it is proposed here to use an Eulerian approach to process the data.

Since the ultimate objective of this master thesis, namely an accurate simulation of obstacles floating in a turbulent free-surface flow, is commonly encountered in real-life applications, an Eulerian approach is almost always the method employed when extracting experimental data. Indeed, for instance, a volumetric sensor is placed and fixed at a specific location, and the flow passes through it exactly as in the numerical Eulerian approach. One desires to mimic this concept.

5.1 1D analysis (rectangle method)

Later in this thesis, it will be necessary to extract from a two-dimensional (spatial) quantity $f(x, y, t)$ a specific profile/value along a single axis (e.g., x or y). This extraction is essential for comparing with experimental data or analyzing flow development. To do so, one would naively retrieve the quantity at the desired location $f(x^*, y, t)$ (resp. $f(x, y^*, t)$), but it is very unlikely that particles will be found at this specific location. This is due to the Lagrangian nature of SPH where particles move freely according to the flow physics. Hence, one expands the punctual location to a restrained area defined by $[x^* \pm \Delta x/2, y]$ when extracting along the x -axis, or $[x, y^* \pm \Delta y/2]$ when extracting along the y -axis.

Then, every particle within the area is taken into account and the desired quantity is computed over all particles in this region to obtain a representative profile/value at the target location x^* or y^* . This concept is schematically illustrated in Figure 5.1.

From this concept arise two kinds of analysis, each serving different purposes. Let us consider an example where particles are moving through a horizontal pipe such that the longitudinal axis is x and the transversal axis is y . The two methods may be explained individually using the same example.

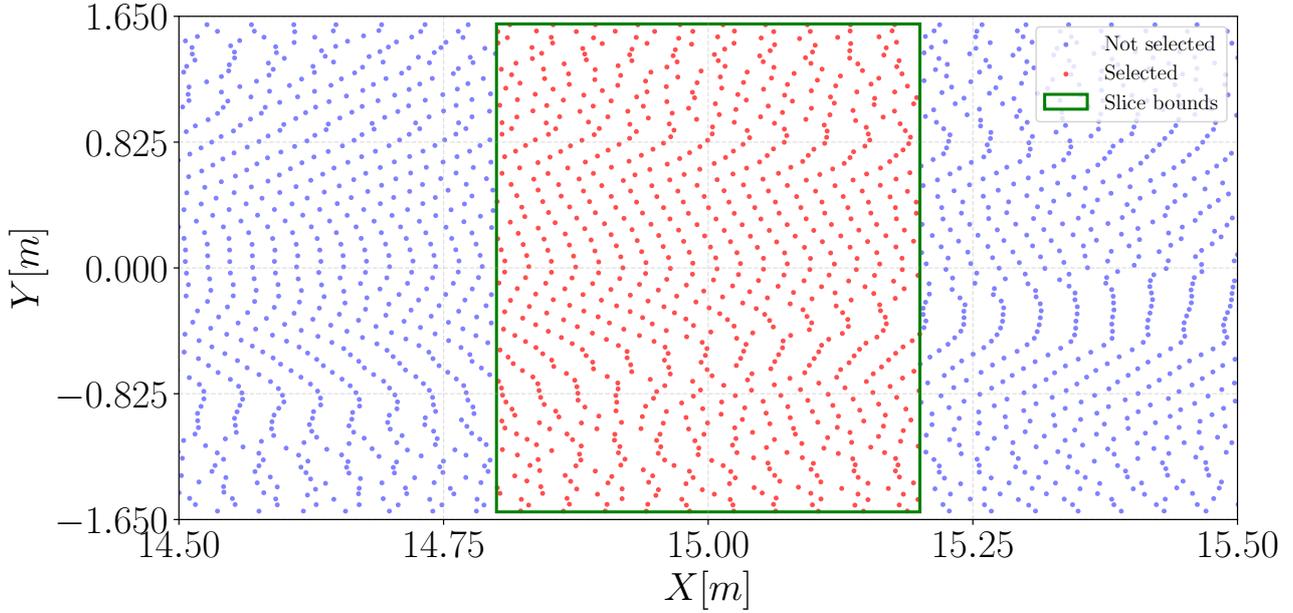


Figure 5.1: Particles selected inside the rectangle. The rectangle boundaries are shown in red, while particles contributing to the analysis are highlighted.

5.1.1 Fixed rectangle

The first method consists of fixing the area at a specific position, either x^* or y^* . Given the horizontal pipe geometry, the area is a rectangle. One would thus obtain a quantity $f(x^*, y, t)$ (resp. $f(x, y^*, t)$) which depends solely on time and the transversal coordinate y (resp. x). In our example, one could retrieve the velocity profile $u(x^* = 15, y, t)$ along the entire diameter at a fixed location. This analysis is therefore chosen when one wants to study how a quantity varies across the transversal direction at a specific longitudinal position.

5.1.2 Gliding rectangle

Keeping the same example, one might want to determine the averaged pressure along the longitudinal pipe direction. To do so, one divides the longitudinal distance x into multiple bins, captures the particles inside each prescribed rectangle, and then performs the data processing separately for each rectangle. For each rectangle, one retrieves a specific value from the framed particles, for example an averaging application, such that the resulting quantity now depends on the longitudinal coordinate x (in contrast to the previous case). This is strictly equivalent to having a single rectangle gliding along the longitudinal axis.

This may be mathematically expressed as a mapping function $\mathcal{A} : \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$\mathcal{A}[f(x, y)]_{(x^*)} = \bar{f}(x^*) \quad \forall x^* \in \mathcal{D}_x \quad (5.1)$$

where \mathcal{D}_x represents the domain of longitudinal positions being analysed.

Note that one may apply either method to analyse instantaneous or time-averaged quantities. Time-averaging is particularly useful for turbulent flows where mean quantities are of interest. To keep consistency with the heart of the SPH method, it is important that the analysed rectangles are at least as large as $4r$. This ensures sufficient particle sampling for statistically meaningful results.

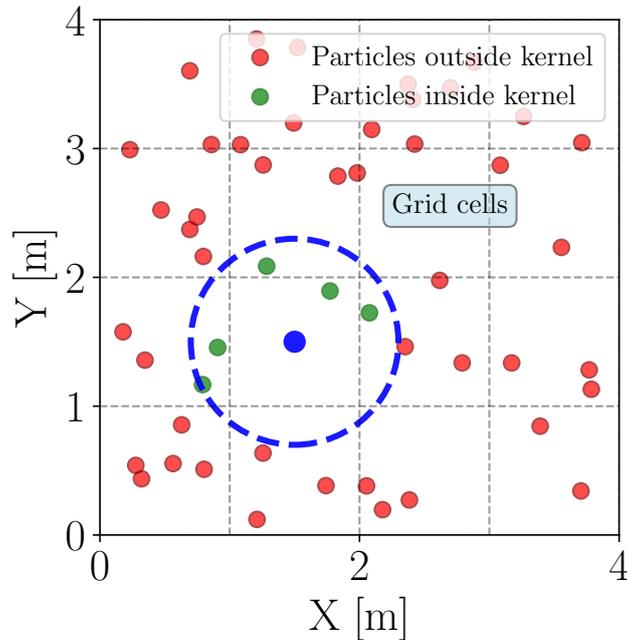


Figure 5.2: Applying SPH interpolation for each cells of the grid. The displayed neighbourhood size is not at scale to enhance visualisation.

5.2 2D analysis (grid method)

Investigating spatial derivative in a single axis is compatible with the previous section using for instance finite difference schemes in combination with the extracted values. However, it becomes impossible to correctly evaluate, for example, the divergence of the velocity of a flow, which requires derivatives in multiple directions simultaneously. Such multi-dimensional operators are crucial for analysing incompressibility, vortex dynamics, and other flow characteristics.

In our Euler approach, it is then necessary to use a grid and evaluate a specific quantity inside each cell. The mapping function used to retrieve a meaningful quantity for each cell is the SPH interpolation with the same neighbour support $h = 4r$, which naturally provides smooth field reconstruction from scattered particle data. By doing so, one may define a grid as fine as desired without losing consistency in the physical analysis. This idea is presented in Figure 5.2.

Even though the grid can be arbitrarily fine, below a threshold value (about the size of a particle spacing), smaller cells would lead to redundant information for neighbouring cells as two adjacent cells would have the exact same particles in their neighbourhood, but the displayed quantity field would solely be more smoothed. Once the field is reconstructed on the grid, standard finite difference operators can be applied to compute any required spatial derivatives.

Preliminary validation tests

Before proceeding further, it is essential to approach the analysis step by step to build confidence with the software and identify its strengths and limitations. To approach our final configuration of flow around obstacles in turbulent free-surface flow, the two main characteristics must first be studied individually. To do so, the current chapter aims to consider two specific configurations where the turbulence and free surface behaviour can be isolated and analysed.

The turbulence in SPH simulation has already been studied multiple times, for instance by [34] where this method has been coupled with $k - \epsilon$ model and successfully granted to be turbulent-accurate. Or also by Hu and al. [35] who used a coarse-grained particle system for turbulence simulation based on SPH approximations to spatially filter the Navier-Stokes equation and also provides promising results.

In the other hand, water interfaces have already been countlessly analysed in the context of free surface simulation articles. One may cite [36] who used a SPH coupled with semi-analytical Smagorinsky model to capture correctly interaction of dambreak flow with porous media. More conventionally, [37] used a SPH- $k - \epsilon$ coupling to model water surface elevations. Nevertheless, these articles solely focus on configuration where the system is *left to itself* with no additional external interactions (such as entering/exiting flow rates).

6.1 User warning

In the SPLisHSPlasH code, there is a 0.8 factor used to slightly reduce the volume of particles and thus their mass. This helps prevent excessive pressure forces at the beginning of the simulation, which can happen if particles are packed too densely.

Nevertheless, this modification will lead to some issues that will be discussed later. To fix this problem, one decides to modify the mass of each particle over time.

In the initial code, the mass of each particles is only evaluated at the initialisation of the simulation and is considered constant. One modifies the code to re-evaluated the mass according to time as follows:

$$m(t) = a(t)\rho_0V \quad \&\mathcal{E} \quad a(t) = 1 - (1 - 0.8)e^{-2t/t^{\text{transition}}} \quad (6.1)$$

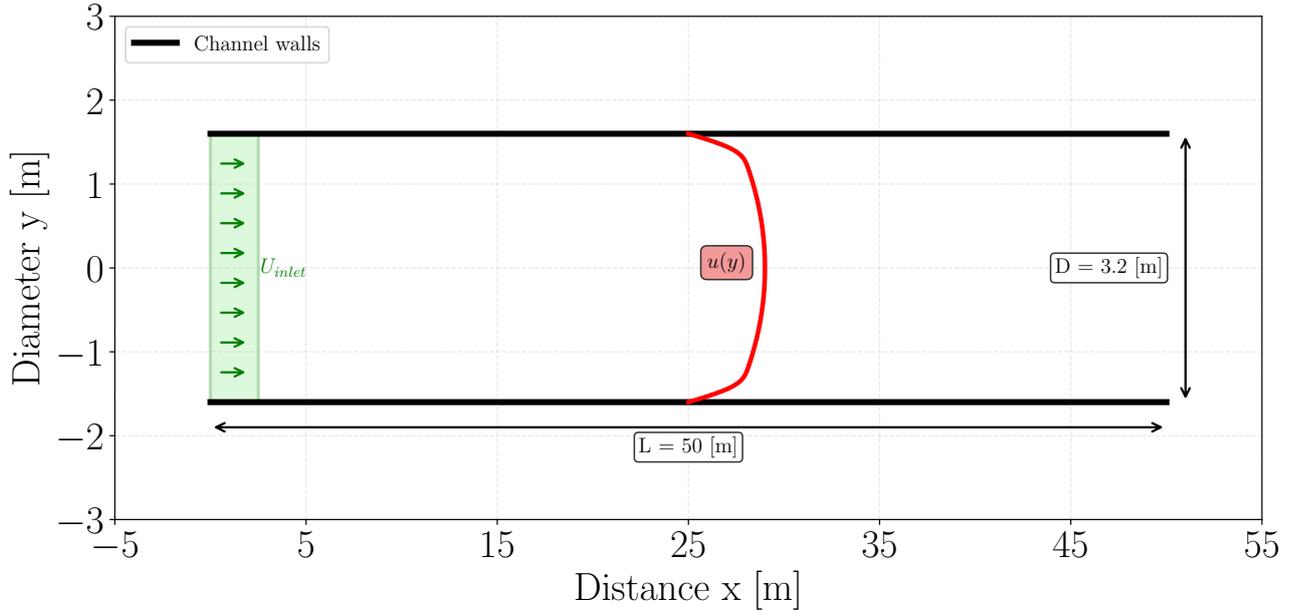


Figure 6.1: Two dimensional pipe of length $L = 50$ [m] and diameter $D = 3.2$ [m] with uniform inlet velocity U_0 and expected velocity profile $u(y)$.

such that the mass reaches its maximal value after approximately 3 seconds.

6.2 Two-dimensional turbulent flow analysis

To makes analysis easier, this section focuses on the turbulent behaviour in a two-dimensional simulation consisting of a pipe geometry with an inlet flow. The two-dimensional results can easily be extrapolated to the three-dimensional domain since the problem is axisymmetric. The specific turbulent metric analysed will be the velocity profile since it is a macro variable that most significantly influences the whole domain (in contrast to the so-called y^+ for instance, which is only important near the walls).

While the laminar flow case yields well-established solutions that are mathematically and easily obtained, the turbulent velocity profile presents clearly additional challenges to obtained its analytical solution. Recent developments by Fedoseyev [38] have introduced a methodological framework for expressing the velocity profile of incompressible fluid flow analytically.

6.2.1 Problem statement

The considered pipe configuration has a diameter $D = 3.2$ [m] with its axial length $L = 50$ [m]. The two boundary conditions consist of an uniform inlet velocity $U_0 = 5$ [m/s] and outlet pressure $p_{\text{out}} = 0$ [Pa]. presented in Figure 6.1. The non-slip wall condition is also involved here.

In this configuration, the characteristic Reynolds is given by $Re = \frac{U_0 D}{\nu} = \frac{5 \times 3.2}{10^{-6}} = 1.7 \times 10^7$. Typically in pipe geometries, the flow is considered fully-turbulent for $Re \gtrsim 4000$ yielding no doubts about the turbulent behaviour of our simulation.

6.2.2 Theoretical solution: generalized hydraulic equations

The approach of Fedoseyev represents the velocity profile as a linear superposition of laminar and turbulent components through application of the Generalized Hydraulic Equations (GHE). The following discussion tries to summarize the theoretical framework of Fedoseyev but also provides additional informations concerning the derivation of the final solutions.

The GHE, which extend the classical Navier-Stokes equations, are derived from the Boltzmann transport equation (from theory of gaz) dictate how the particle velocity distribution functions should behave, denoted as f :

$$\frac{Df}{Dt} = J \quad (6.2)$$

Fedoseyev proposes to add a third time scale associated with finite-dimension interacting particles, an additional term emerges in the Boltzmann transport equation, yielding to:

$$\frac{Df}{Dt} - \frac{D}{Dt} \left(\tau \frac{Df}{Dt} \right) = J, \quad (6.3)$$

where τ represents the *mean free path*, e.g. the mean time between particle collisions. This additional term is proportionality to the Knudsen number¹.

After some tedious manipulations, the so-called GHE, explicitly formulated in [39], may be written as:

$$\nabla \cdot \mathbf{u} = \tau^* \left\{ 2 \frac{\partial}{\partial t} (\nabla \cdot \mathbf{u}) + \nabla \cdot (\mathbf{u} \nabla) \mathbf{u} + \nabla^2 p - \nabla \cdot \mathbf{F} \right\} \quad (6.4)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \nabla) \mathbf{u} - Re^{-1} \nabla^2 \mathbf{u} + \nabla p - \mathbf{F} = \tau^* \left\{ 2 \frac{\partial}{\partial t} (\nabla p) + \nabla^2 (p \mathbf{u}) + \nabla (\nabla \cdot (p \mathbf{u})) \right\} \quad (6.5)$$

where the parameters are defined as follows:

- \mathbf{u} denotes the dimensionless velocity vector
- p represents the dimensionless pressure
- $Re = U_0 L / \nu$ corresponds to the Reynolds number
- U_0 is the characteristic velocity scale (here, the uniform inlet velocity)
- L represents the hydrodynamic length scale (here, the channel diameter)
- $\nu = \mu / \rho$ denotes the kinematic viscosity
- \mathbf{F} represents the body force vector
- dimensionless $\tau = \tau^* L^{-1} V_0$

The value of τ^* constitutes a material property not determinable a priori, Fedoseyev provided insights regarding its selection in his analysis of results.

In deriving Equations (6.4) and (6.5), Fedoseyev made some additional assumptions (slow flow variation; $\tau = \text{constant}$; negligible third order fluctuation; steady state) and applying finally pressure boundary condition:

$$\nabla (p - \mathbf{F}) \cdot \mathbf{n} = 0, \quad (6.6)$$

¹The Knudsen number, $Kn = \lambda / L$ represents the ratio of molecular mean free path length to a characteristic physical length scale.

where \mathbf{n} denotes the wall-normal unit vector.

Eventually, the simplified GHE are obtained:

$$\nabla \cdot \mathbf{u} = \tau \nabla^2 p^* \quad (6.7)$$

$$\mathbf{u} \nabla \cdot \mathbf{u} = -\nabla p^* + \frac{1}{Re} \nabla^2 \mathbf{u} \quad (6.8)$$

For the two-dimensional case, the velocity profile can be expressed as:

$$\mathbf{u}(x, y) = u(x, y)\mathbf{e}_x + v(x, y)\mathbf{e}_y \quad (6.9)$$

The fundamental assumptions underlying this analysis can be summarized as:

- Steady state $\partial t = 0$
- Incompressible flow conditions: $\rho = \text{constant}$
- Negligible body forces (here gravity): $\mathbf{F} \approx \mathbf{0}$
- Uniform pressure gradient: $\nabla p = \text{constant}$
- Fully-developed flow conditions: $\partial \mathbf{u} / \partial x = \mathbf{0}$

The derivation of the laminar and turbulent profile has already been done by Fedoseyev. Nevertheless, some typos and inaccuracies were present in this pre print paper. Hence, the full demonstration has been redone and is presented in Section A.2.

As stated in the beginning, the actual velocity profile is written as a linear combination of the turbulent and laminar solution as:

$$u(y) = \gamma u(y)^{\text{turbulent}} + (1 - \gamma) u(y)^{\text{laminar}} \quad (6.10)$$

$$= U_0 \left[\gamma \left(1 - e^{\left(1 - e^{\frac{y}{\delta}}\right)} \right) + (1 - \gamma) 4y(L - y)/L^2 \right] \quad (6.11)$$

This is important to notice and remind that the two only unknown parameters are γ and δ . The parameter δ is directly linked to the time parameter τ^* and would thus require experiments to be assessed. Nevertheless, given the concordance with this analytical profile and the experiments from Fedoseyev, it is proposed here to fit the parameters depending on our results. Additionally, one will physically study the influence of δ on the flow.

Parameters δ physical meaning Given the definition of the δ parameters (derived in the associated Annexe section):

$$\delta = \frac{\sqrt{\tau^* \nu}}{L} \quad (6.12)$$

represents the ratio between a characteristic diffusion length $\sqrt{\tau^* \nu}$ and the characteristic length of the flow domain L . This parameter captures the relative importance of viscous diffusion and molecular relaxation processes in the flow.

One can observe in Figure 6.3 the influence of δ on the velocity profile for a given γ . For this analysis, γ is kept close to unity (0.8) to focus on the turbulent characteristics of the flow. As shown in Figure 6.3a, for small values of δ , the velocity profile evolves sharply from the boundary to the centre of the pipe, indicating stronger velocity gradients near the walls. This

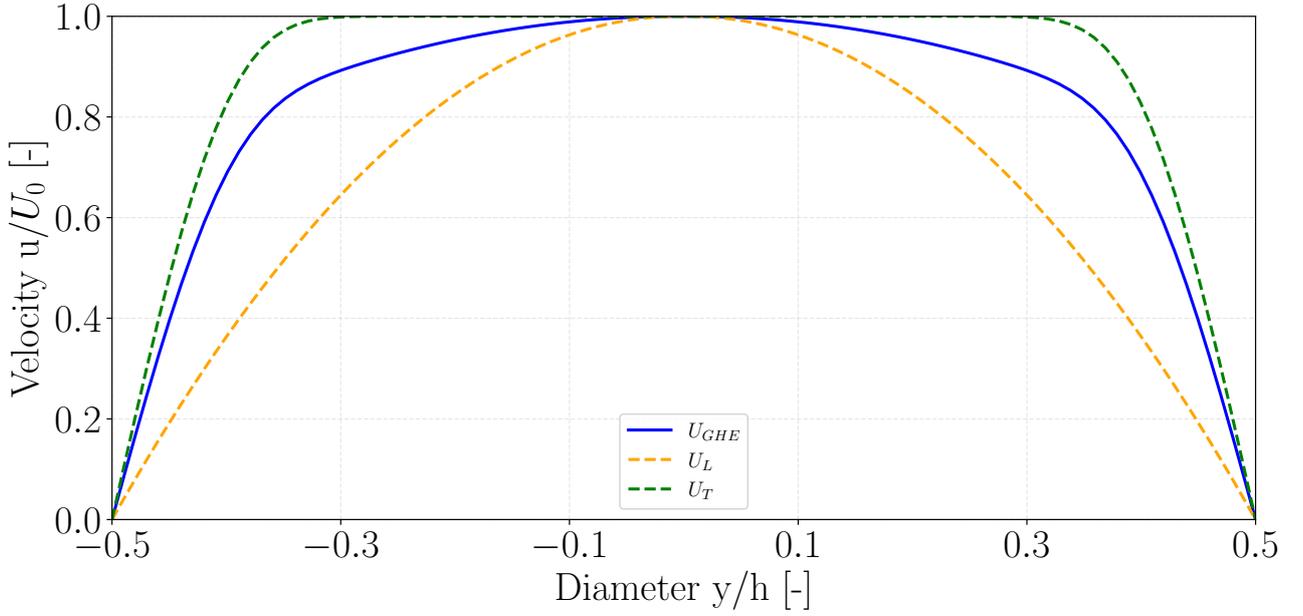
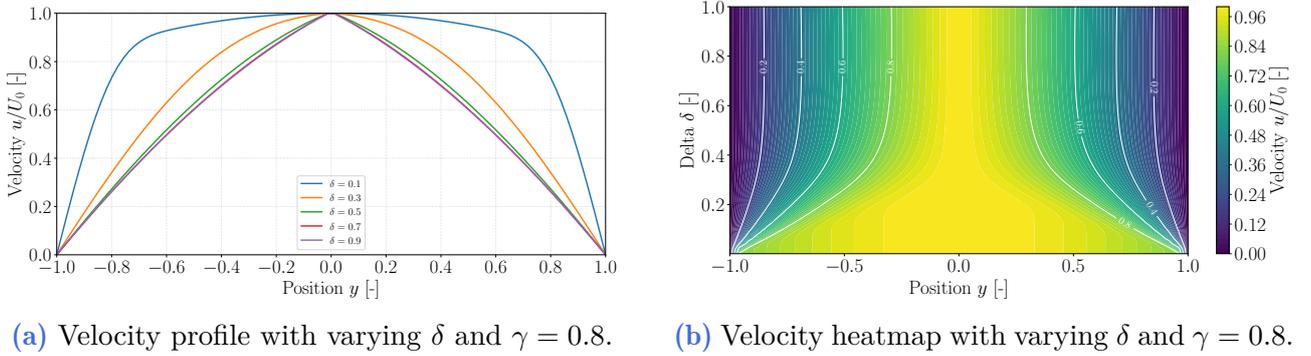


Figure 6.2: Mixed velocity profile with $(\delta, \gamma) = (0.1, 0.7)$.



(a) Velocity profile with varying δ and $\gamma = 0.8$.

(b) Velocity heatmap with varying δ and $\gamma = 0.8$.

Figure 6.3: Influence of the parameter δ on a almost turbulent velocity profile.

behaviour is physically consistent since $\delta \propto \sqrt{\tau^*}$, where smaller relaxation times correspond to more frequent particle interactions, characteristic of highly turbulent flows.

Figure 6.3b further illustrates this phenomenon: with small δ values, the flow reaches its near-maximum velocity very close to the boundary. Since $\delta \propto \sqrt{\nu}$, smaller values indicate that viscous effects are relatively less significant compared to the domain size, resulting in a reduced influence of boundaries on nearby fluid particles. Conversely, higher values of δ indicate more significant momentum diffusion throughout the flow, leading to smoother velocity transitions from the boundaries toward the centre.



Figure 6.4: True geometry of the two dimensional channel with emitter object (blue block) and boundaries (grey blocks) considering a larger entry to avoid inlet blocking particles due to non-slip wall condition.

6.2.3 Numerical results

Beforehand, it is important to note that the fluid described by the Equations(6.7) and (6.8) is not perfectly incompressible due to the additional dissipation term in the former equation. Even though Fedoseyev explained that this error on the continuity equation exists, the error committed remains substantial and the fluid could still be considered as incompressible.

Modified geometry

The geometry considered for the simulation differs slightly from the one involved in the theoretical solution, as shown in Figure 6.4. Particles are injected into the channel using a *particles emitter* (blue object) which releases particles in the x direction at a specific and uniform velocity U_0 .

However, the no-slip wall boundary condition would prevent some particles (those near the walls) from entering the channel properly, causing them to be blocked at the entrance and effectively creating an obstacle. To prevent this issue, a wider entrance (with a slightly greater diameter $D^* = 3.94$ [m]) was introduced so that the blocked particles at the boundaries would not withstand the main flow. Alternatively, a more brutal approach would be to increase the number of particles such that the blocking particles would be too small to influence the main flow.

Hence, the horizontal pipe begins at $x = 0$. Also, to prevent any undesired boundary effects, the dimensions involved in the future analyses are:

$$(x_0, y_0) = (0, -D/2) \quad \& \quad (x_1, y_1) = (45, D/2)$$

Simulation parameters	Value	Units
particleRadius	0.02	m
timeStepSize	0.001	s
cfFactor	0.4	-
cfMaxTimeStepSize	0.005	s
maxIterations	200	-
maxError	0.05	-
maxIterationsV	200	-
maxErrorV	0.05	-
viscoMaxIter	500	-
viscoMaxError	0.05	-
Flow parameters	Value	Units
stopAt	30	s
Fluid parameters	Value	Units
viscosity	10^{-6}	m^2/s
viscosityBoundary	2	m^2/s
inertiaInverse	0.5	m^{-2}
xsph	0.01	-
xsphBoundary	0.7	-
drag	1.0	-
vorticity	0.02	m^2/s
viscosityOmega	0.1	m^2/s

Figure 6.5: Parameter used to simulate turbulent flow in two dimensional channel.

Simulation parameters involved

The following set of parameters have been chosen and are displayed in Table 6.5. the viscosity of the boundaries are voluntary extremely high to simulate a non-slip wall condition. Also, the number of iteration required for the density and divergence solver may be that low since the number of particle is high (about 130 000).

The desired *stopAt* is chosen to be proportional to the advection time. The needed time for an inlet particle to reach the end of the pipe is given by:

$$t_{\text{advection}} = \frac{L}{U_0} \approx \frac{45}{5} = 9 \text{ [s]}$$

where one arbitrary decides to use $\text{stopAt} = 3t_{\text{advection}} \approx 30 \text{ [s]}$.

At the end of the simulation, here is depicted in Figure 6.6 the heat map of the velocity magnitude for all particles.

Steady state assumption

One initially made numerous assumptions about the flow and fluid. the first clear check is to verify the steady state assumption. To do so, one uses the so-called *fixed rectangle method* for a fixed area of the pipe defined by:

$$(x_0, y_0) = (25 - 5r, -D/2) \quad \& \quad (x_1, y_1) = (25 + 5r, D/2)$$

the width of the rectangle is chosen ensuring at least 80 particles inside and uniformly distributed, as shown in Figure 6.7.

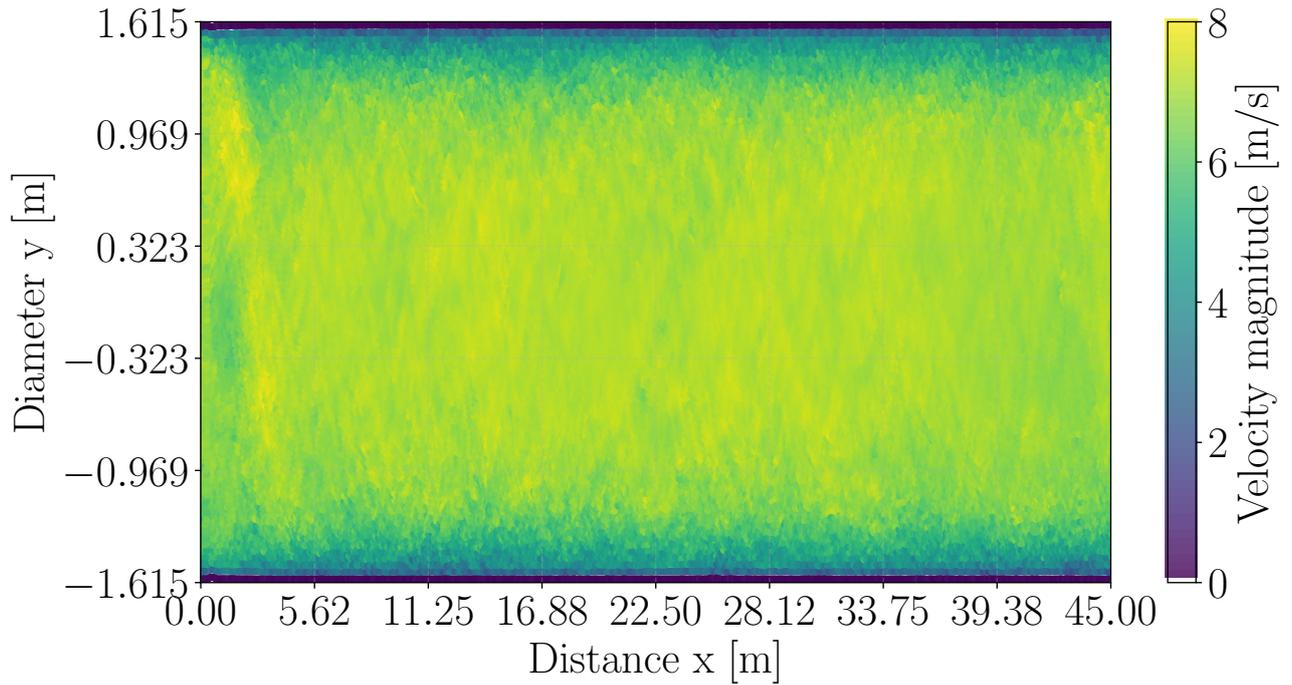


Figure 6.6: Final timestep ($t = 30[s]$) of the turbulent flow displaying velocity magnitude field. The disturbed velocity field is partially chaotic, which is an expected behaviour for fully turbulent flow in such a configuration.

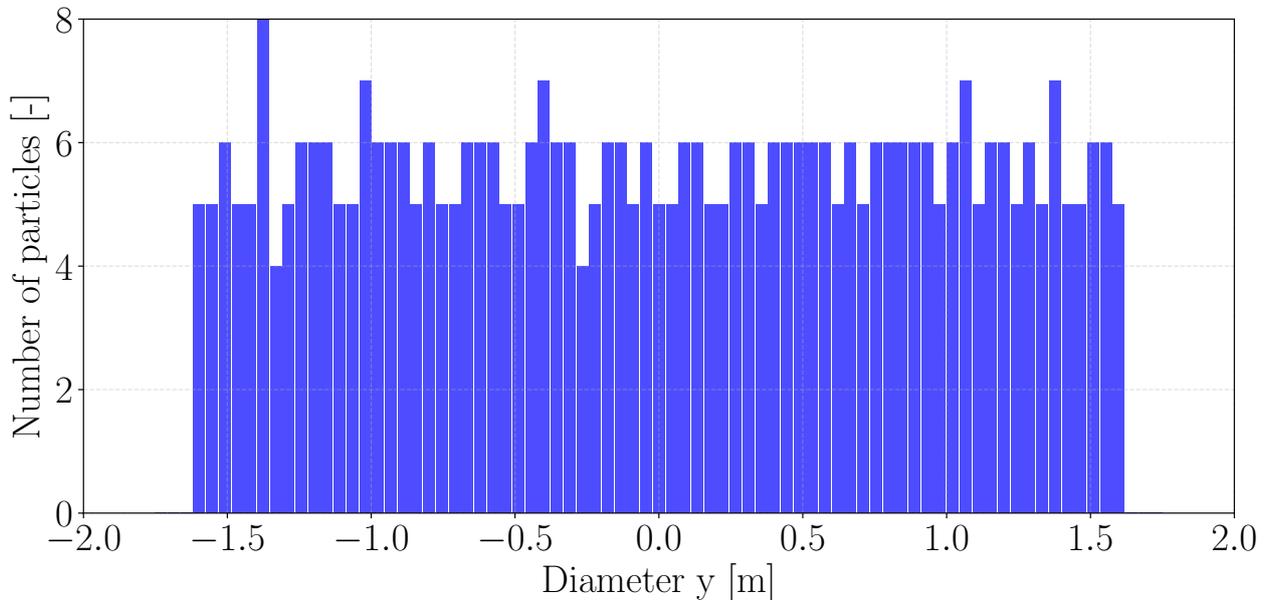


Figure 6.7: Particle distribution inside the rectangle with a width of 10 particles. Showing that particles are indeed uniformly distributed and that statistical data can be extracted from it.

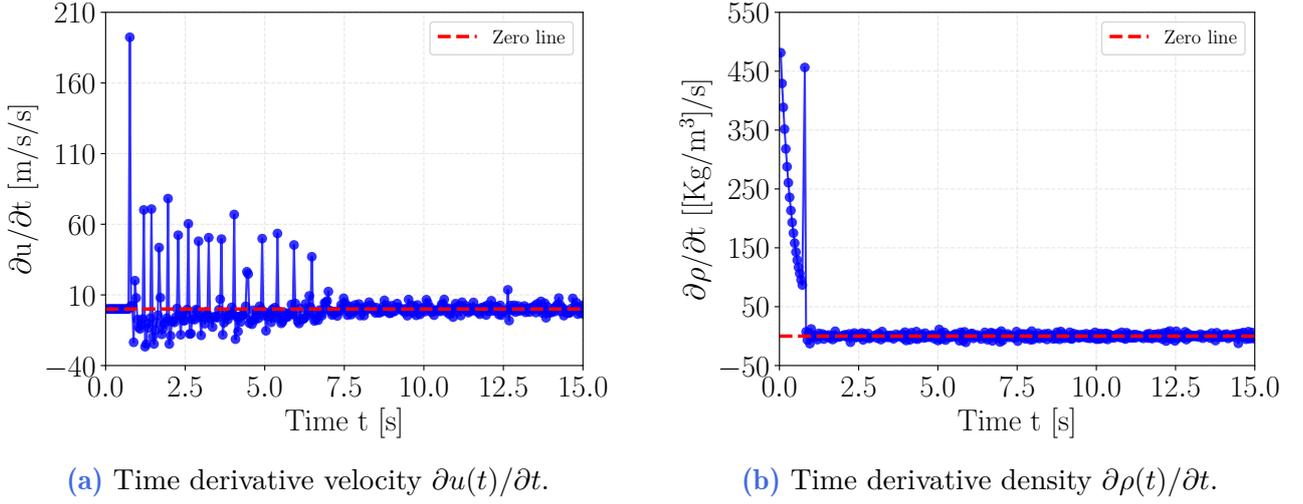


Figure 6.8: Time derivatives of density and velocity at $25\text{m} \pm 10\text{r}$ over the whole diameter.

The time derivative can be approximated using explicit Euler scheme:

$$\frac{df(t)}{dt} \approx \frac{f(t^{n+1}) - f(t^n)}{t^{n+1} - t^n}$$

and should decrease over time. In Figure 6.8 are depicted the approximated time derivative of the density and velocity variables. It has to be pointed at that the one used the mean velocity as pertinent variable rather than the whole velocity profile.

From the figures above, it can be observed that the steady state is reached for:

$$t_{\text{steady state}} \geq 10[s]$$

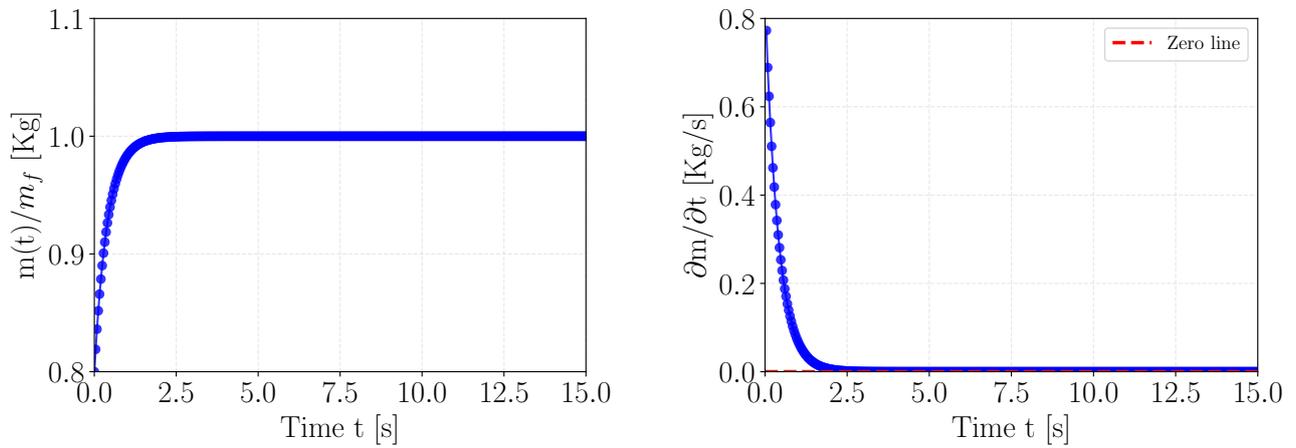
for both variables and all futures data will be analysed after this value. A notable remark about density is that its steady is reached after only few seconds while the velocity needs a longer time to reach a slightly perturbed steady state. One may assess that the DFSPH works greatly about the density variables. It is nonetheless expected for the velocity to be disturbed as such given the turbulence behaviour of the fluid.

One also mentions the time variation of the mass. As explained earlier, it has been modified to reach its final value after some time. the time evolution and derivative of the mass of a unique particle are depicted in Figure 6.8 where the results are in agreement with the desired behaviour.

Fully developed flow assumption

The second main assumption is the fully developed. It is proposed to analyse the spatial derivative of the velocity variable $\frac{\partial \mathbf{u}}{\partial x}$ onto the whole domain. In 6.11 are depicted the two spatial derivative velocities.

One observes in Figure 6.11a that the x -derivative has obviously slightly higher values at the entrance with a highest value about 2 (in absolute value). The fluctuations inside the derivatives can be associated to the turbulent flow which inherently bring disturbance in the flow. Nevertheless, these fluctuations are more than 3 times smaller (~ 0.7) than the ones at the entrance.



(a) Time variation of the mass $m(t)$.

(b) Time derivative of the mass $\partial m(t)/\partial t$.

Figure 6.9: Time evolution and derivative of the mass of a single particle at 25m.

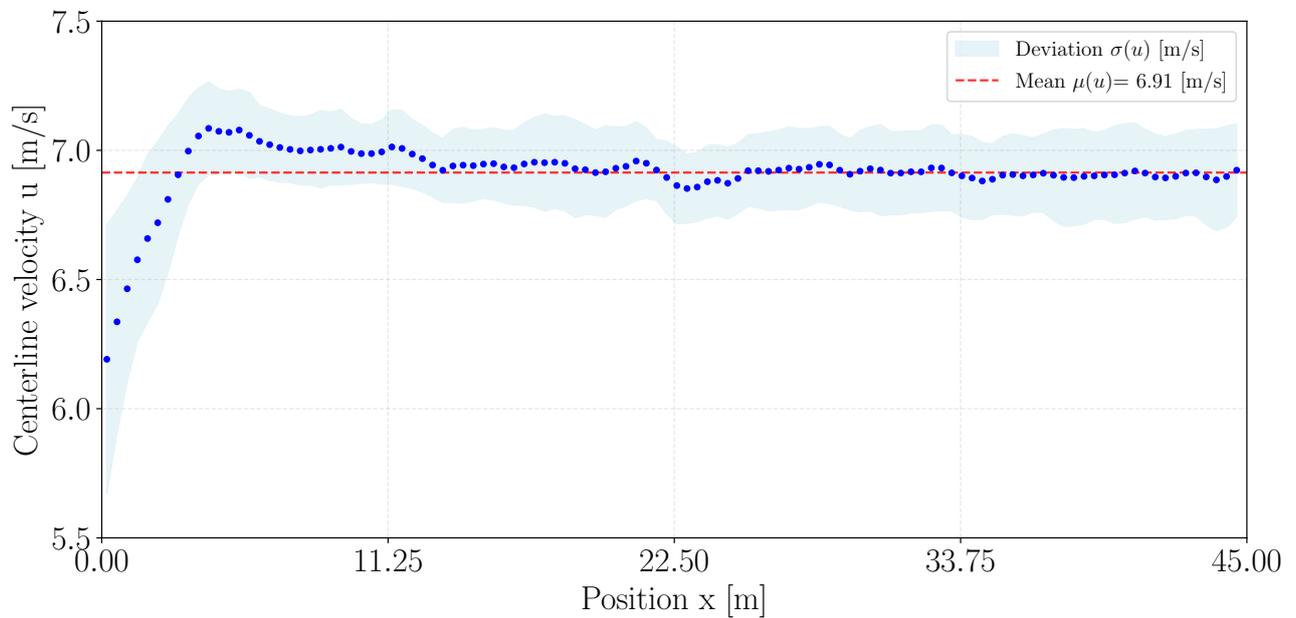
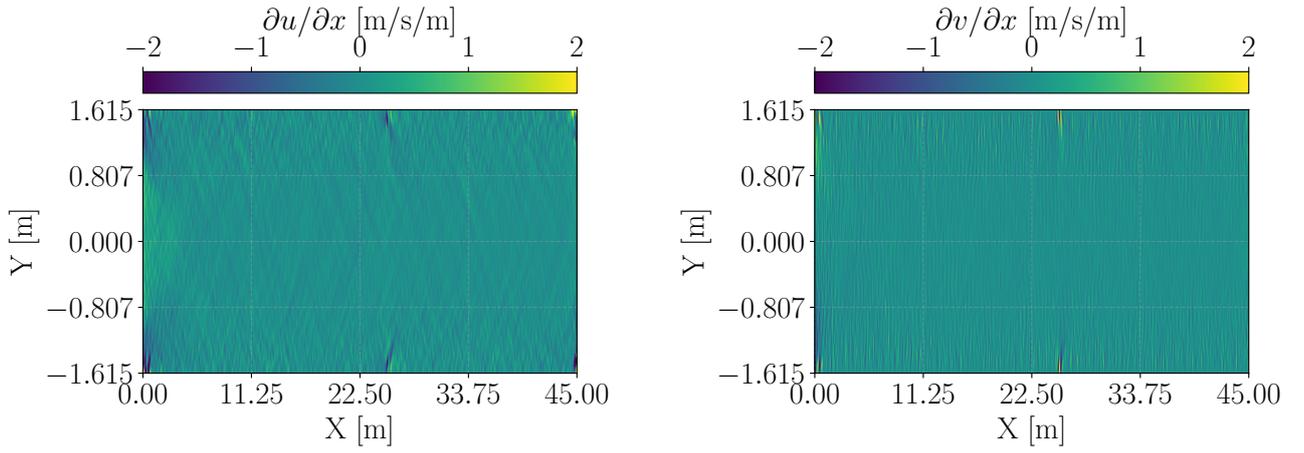


Figure 6.10: Centreline velocity distribution along the pipe with its mean value and standard deviation.



(a) Spatial derivative of the longitudinal velocity $\partial u/\partial x$.

(b) Spatial derivative of the transversal velocity $\partial v/\partial x$.

Figure 6.11: Spatial derivatives of the velocity vector $\partial \mathbf{u}/\partial x$.

Another manner to assess fully development is to analyse the centreline velocity that should be constant once the assumption is verified. In Figure 6.10 is depicted the centreline velocity with its according mean value and standard deviation.

From the discussion above, one considers the flow to be fully developed in the region:

$$x_{\text{fully developed}} \in [25, 45] \text{ [m]}$$

without any doubt. Thus the further velocity analysis will be performed in this region.

Incompressibility assumption

Finally the last main assumption is the incompressibility of the fluid. The *local approach* consists in determining the total time derivative of the density. Since one already pointed out that the density has reached its steady state after 2-3 seconds, if the spatial derivatives are also zero, the material derivative would also be zero and the incompressibility assumption would be verified.

Restricting our analysis in the fully developed domain to avoid the inlet/exit disturbance effects, the two partial derivatives are evaluated and displayed in Figure 6.12. One highlights the fact that all the quantities have been averaged over time using $t > t_{\text{steady state}}$.

Since the total derivative of the density reads:

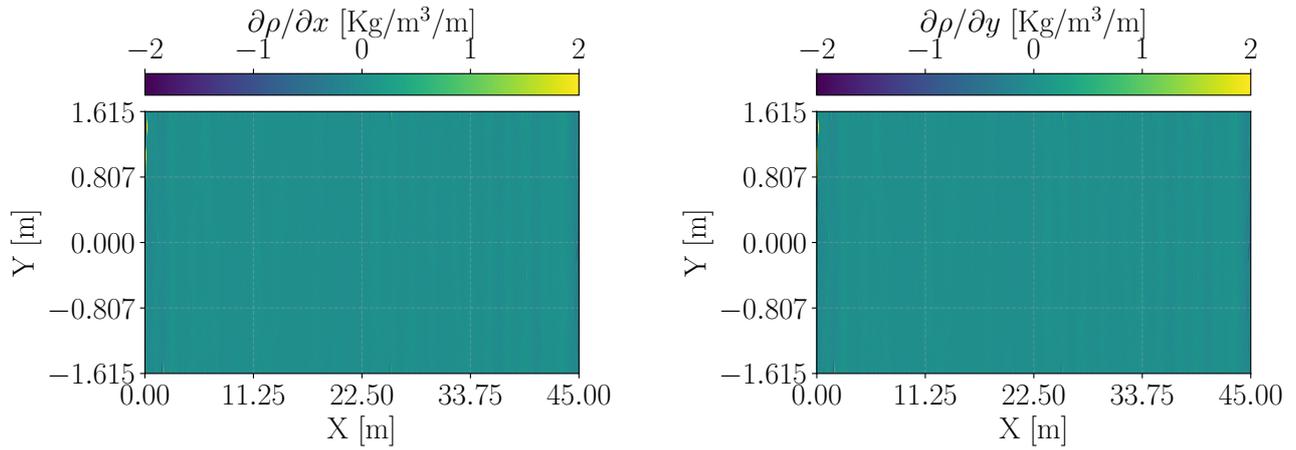
$$\frac{d\rho(x, y, t)}{dt} \equiv \frac{\partial \rho}{\partial t} + \frac{\partial \rho}{\partial y} \frac{dy}{dt} + \frac{\partial \rho}{\partial x} \frac{dx}{dt} \approx 0 + 0 + 0 \quad (6.13)$$

One assesses that, at least in the prescribed domain, the flow is locally incompressible.

Another way of proving incompressibility is to evaluate the flow rate over the channel and observe if it is conserved. This method is known as a *global approach*. One evaluates both the mass and volume flow rate, respectively defined as:

$$Q_v = \int_{-D/2}^{D/2} u(y) dy \quad (6.14)$$

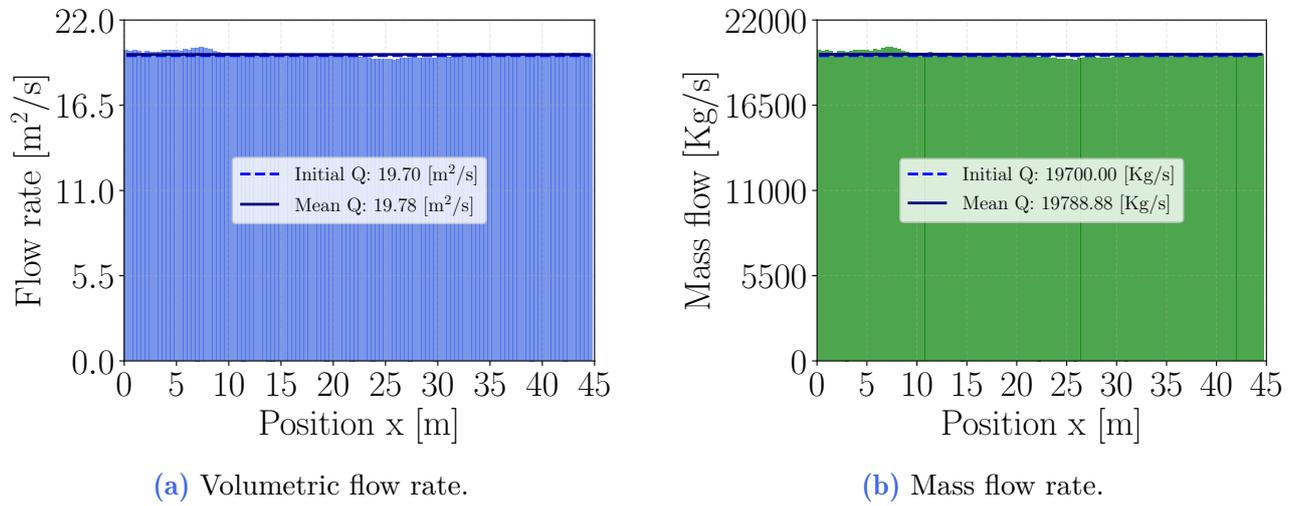
$$Q_m = \int_{-D/2}^{D/2} \rho u(y) dy \quad (6.15)$$



(a) Spatial derivative of the density $\partial\rho/\partial x$ in longitudinal axis.

(b) Spatial derivative of the density $\partial\rho/\partial y$ in transversal axis.

Figure 6.12: Two spatial derivatives of density in the domain $(x, y) \in [1, 46] \times [-D/2, D/2]$.



(a) Volumetric flow rate.

(b) Mass flow rate.

Figure 6.13: Computation of the mass and volumetric flow rates $(x, y) \in [1, 46] \times [-D/2, D/2]$ in pipe yielding an error about 0.398%.

Given that the inlet velocity profile is uniform and given that the initial density is known, the two references values are easily obtained:

$$Q_v^{\text{init}} = DU_0 \quad (6.16)$$

$$Q_m^{\text{init}} = D\rho_0 U_0 \quad (6.17)$$

$$(6.18)$$

Finally the two flow rates are displayed in Figure 6.13. In this area, the numerical errors on the flow rates is 0.398% which far more than acceptable.

Hence, from both the local and global point of view, the fluid may be considered as incompressible such that all the main assumptions have been validated.

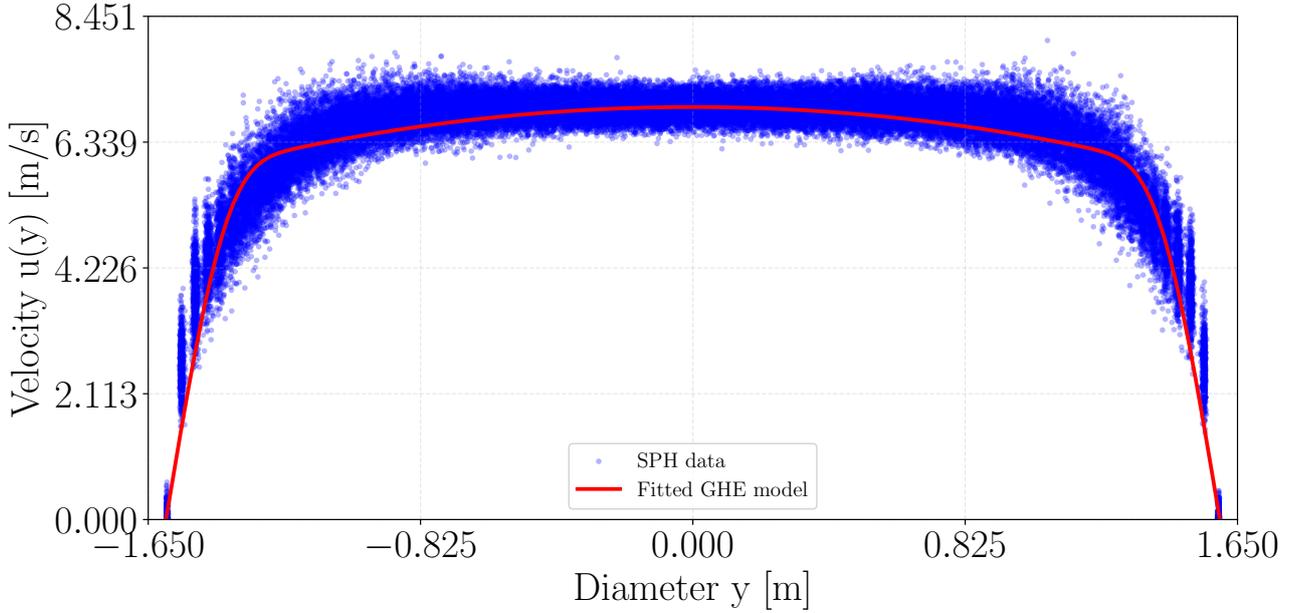


Figure 6.14: Fitted velocity profile based on the GHE model with $(\delta, \gamma) = (0.004, 0.8174)$ using raw data.

Numerical velocity profile

Two approaches have been used to assess the accuracy of the simulation to reproduce the desired velocity profile. The first one including every particles within the location $x = 25 \pm 4r$ between the time $t \in [15, 30]$ [s] to observe the raw behaviour of the outputs, depicted in Figure 6.14. By doing so, one notices that some particles far from the centre actually reach higher values than the centred one, which should not theoretically occurs. Due to this, the velocity U_0 has to be manually tuned by the central particles rather than the true maximal velocity value. However, the fitted GHE model is completely framed within the cloud data points which is a good indicator.

In the other hand, the second approach is focused on a more smooth and clean output treatment by time averaging the transversal coordinates y and velocity $u(y)$ in order to retrieve exactly maximal velocity value at the centre. As observed in Figure 6.15, this behaviour is correctly captured but at the cost of slightly well captured fitting model with also slightly different optimal fitting parameters.

As introduced in the section above, the parameters (δ, γ) used to fit the curves are of course manually fitted since δ depends on the fluid physical properties and given that Fedoseyev did not provide experimental information about δ_{water} . In the same way, the γ parameter only indicate the turbulent proportion within the flow and has thus to be modified according to our turbulent regime depending on the Reynolds number. The fitting method used to obtain such velocity profiles is called *Levenberg-Marquardt algorithm* and is explained in more details in Section A.3 in the Annexe.

6.2.4 Limitations of the model in the turbulent simulation

Due to the nature of SPH simulations, it is not possible to refine spatial resolution only near the boundaries. Hence, to reproduce additional turbulent metrics as y^+ , a much higher number of particles should be added which would directly highly increases the simulation time. Hence, only global metrics as the velocity profile in the channel can be evaluated.

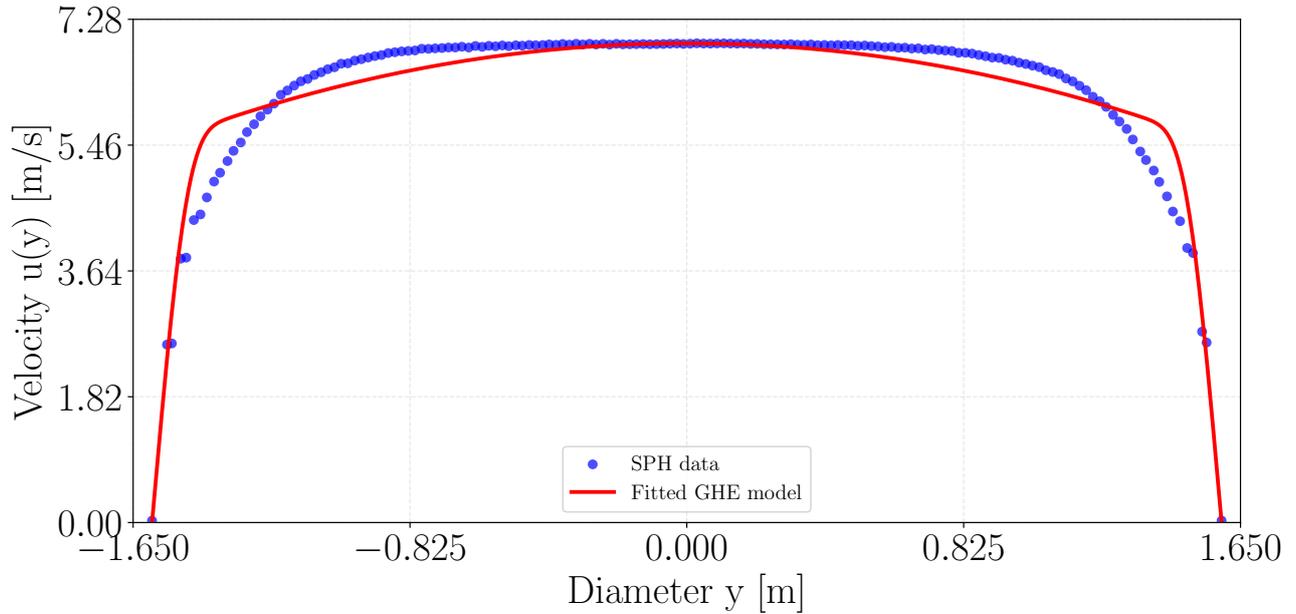


Figure 6.15: Fitted velocity profile based on the GHE model with $(\delta, \gamma) = (0.00103, 0.7840)$ using filtered data.

6.3 Two-dimensional free surface horizontal flow analysis with parabolic obstacle

Up to now, the turbulence behaviour is perfectly (at least at a macro scale) captured by the software in the case of channel pipe. Nevertheless, one only considers the case of rigid solid boundary-fluid interaction. Whereas, flood events occur in nature where fluid-air interaction is ubiquitous and has of course to be studied in detail. Let us now dive into the realm of the free surface flows to see if the software is able to capture the complex shape of the water-air interface. It is important to note that the air only brings a constant atmospheric pressure contribution to the system. Hence, not considering air particles (and thus leaving "void" at the water interface) would only bring null pressure contribution to the system. It is thus possible to study free surface flow behaviour without computing a multi-phase simulation (which would require additional computing time).

This section investigates free surface flow behaviour in a straight horizontal channel with an elliptic obstacle. A constant inlet volume flow rate and downstream outlet velocity are imposed. As with the turbulence study, the two-dimensional analysis is directly applicable to the three-dimensional domain. Indeed, many 3D hydraulic problems are considered as 2D problems with appropriate adjustment factors with confidence. The variable to be analysed will be the water height, as it is the most important parameter in free surface flow.

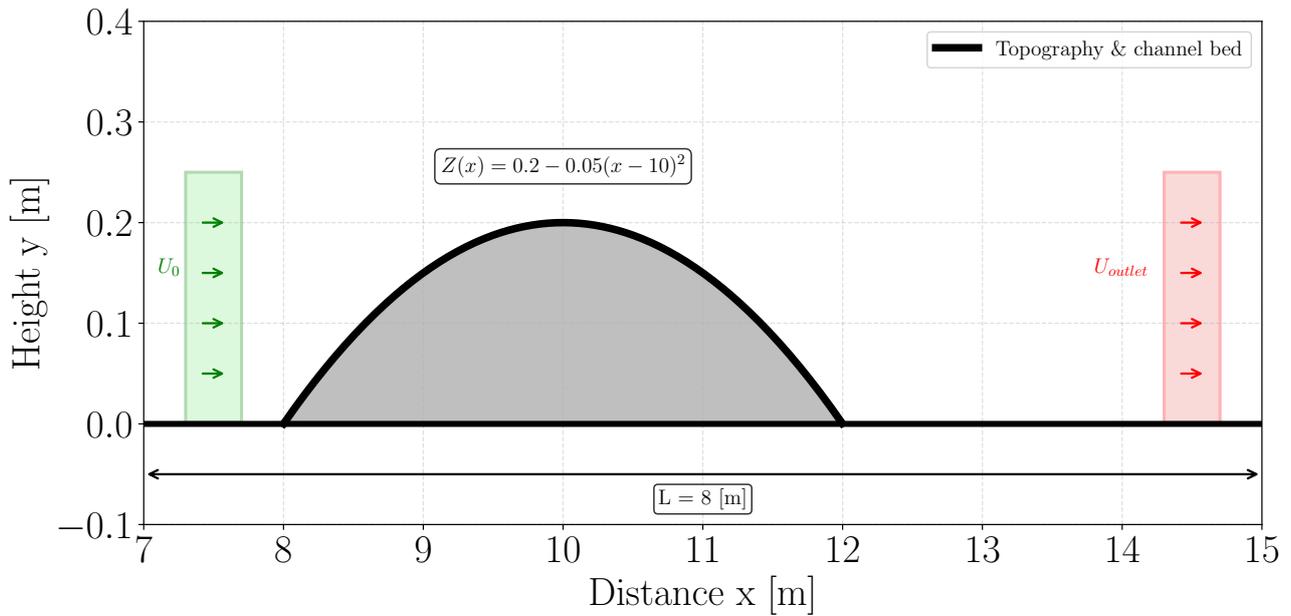


Figure 6.16: Two dimensional channel of length $L = 8$ [m] with inlet flow rate Q_o and outlet velocity U_{outlet} . There is a parabolic obstacle in the middle of the channel.

6.3.1 Problem statement

The considered case is a two dimensional flow with horizontal ground and a parabolic shape obstacle in the middle of the domain with the obstacle having the equation:

$$z_b(x) = 0.2 - 0.05(x - 10)^2$$

The inlet flow rate is $Q_0 = 0.18$ [m²/s] and the outlet water height $h_{\text{outlet}} = 0.33$ [m] obtained through velocity boundary using mass conservation. The configuration is presented in Figure 6.16.

However, since the software only allows us to specify an inlet velocity, one creates at the entrance a vertical emitter of size $L = 0.5$ [m] with an uniform velocity $U_0 = 0.36$ [m/s] such that the vertical integration this latter gives us the desired inlet flow rate Q_0 .

The actual configuration discussed in the thesis of Professor Archambeau is along the distance $x \in [0, 25]$ [m] but one restricts our analysis nearby the obstacle in $x \in [7, 15]$ [m] to reduce the total computational time.

6.3.2 Theoretical solution: Bernoulli equations

The following analytical solution for the present configuration has been derived by Professor Archambeau in his PhD thesis [40] and one will base the accuracy of the model upon his results. While most of the equations are provided and clearly explained in his thesis, one proposes in the Annexe at Section A.4 some additional derivation steps for anybody who might not completely understand the operations performed.

It is important to note that the final derived solution is a 1D solution without taking vorticity into account. Hence, one already mention that the 2D results should adequately be discussed ultimately.

The main assumptions to obtain an analytical solution are the following ones:

- Incompressible fluid
- Steady state $\partial t = 0$
- Rectangular cross section
- Inviscid fluid $\nu = 0$
- Slip wall condition $u(x, y = y_{\text{wall}}) \neq 0$
- Hydrostatic pressure² $p \propto \rho g y$
- Uniform vertical velocity $u = u(x)$

Considering these assumptions, the general Bernoulli relationship reads:

$$\frac{\partial}{\partial x} \left(u(x)q(x) + g \frac{h(x)^2}{2} \right) = gh \frac{\partial z_b(x)}{\partial x} \quad (6.19)$$

With all calculations done, this latter equation is written as follows:

$$[1 - Fr^2] \frac{dh(x)}{dx} = \frac{dz_b(x)}{dx} \quad (6.20)$$

As already mentioned in [40], given the elliptic obstacle shape, one has $\frac{dz_b(x=10)}{dx} = 0$ such that the L.H.S. of Equation (6.20) is also zero at location $x = 10$ leading to two possibilities. Either $\frac{dh(x=10)}{dx} = 0$ or $Fr^2 = 1$. Whereas the first solution is obviously ignored, the second one indicates a transition to a critical Froude number yielding a supracritical flow such that the water height at this location is:

$$Fr^2 = \frac{u_{\text{cr}}^2}{gh_{\text{cr}}} = \frac{q^2}{gh_{\text{cr}}^3} = 1 \Rightarrow h_{\text{cr}} = \left(\frac{q^2}{g} \right)^{1/3} = 0.1489 \text{ [m]} \quad (6.21)$$

Also, considering the outlet water height condition, one easily evaluates the outlet Froude number using q :

$$Fr_{\text{outlet}} = \sqrt{\frac{q^2}{gh_{\text{outlet}}^3}} = 0.3 \text{ [-]} \quad (6.22)$$

From this value one understands that a transition from supracritical to subcritical flow will occur somewhere between the top of the obstacle and the outlet. This particular behaviour leads to the well-known *hydraulic jump* physical effect.

Qualitatively, the water height distribution will follow the trend shown in Figure 6.17.

From the inlet region to the hydraulic jump location (regions I and II), the conservation of head allows us to equate the inlet head and the head at $x = 10$:

$$H_{\text{inlet}} = H_{\text{cr}} = z_b(x = 10) + h_{\text{cr}} + \frac{q^2}{2gh_{\text{cr}}^2} = 0.4234 \text{ [m]} \quad (6.23)$$

With an analogous reasoning, the head conservation between the outlet and the end of the hydraulic jump (region III) may be applied:

$$H_{\text{outlet}} = H_{\text{cr}} = z_b(x_{\text{outlet}}) + h_{\text{outlet}} + \frac{q^2}{2gh_{\text{outlet}}^2} = 0.3452 \text{ [m]} \quad (6.24)$$

²with y being equal to 0 at the surface of the liquid

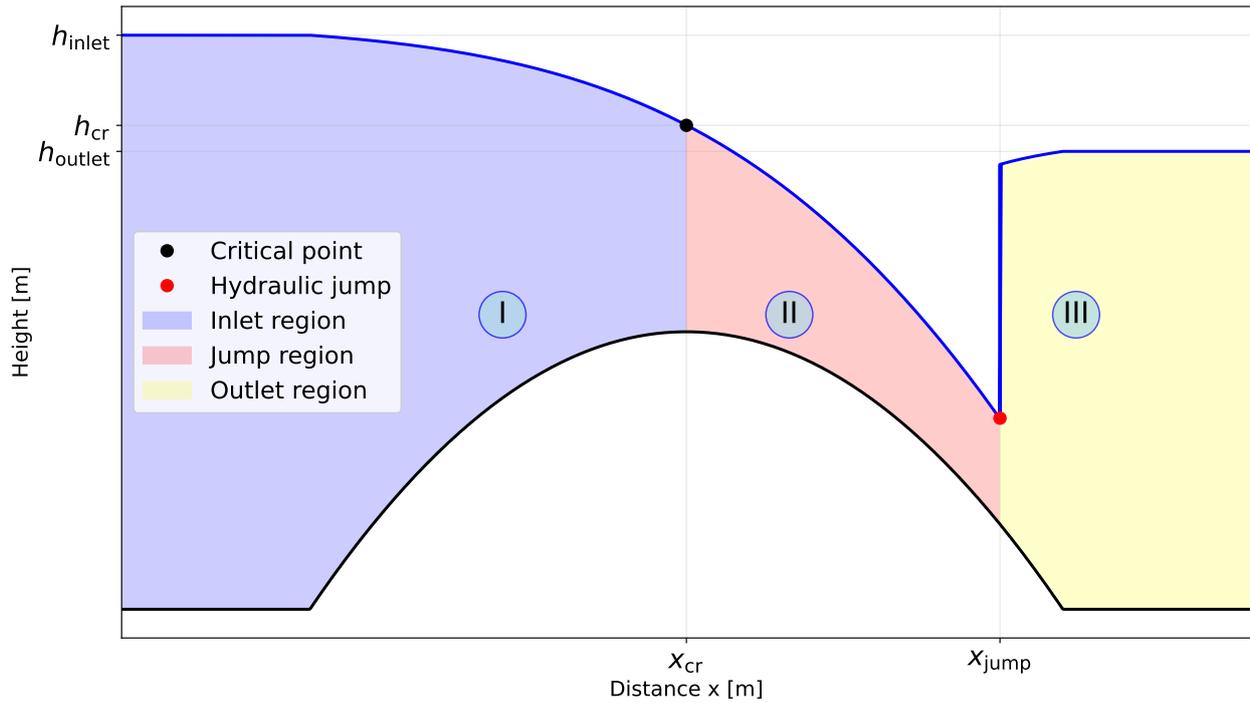


Figure 6.17: Qualitative behaviour of the water height during the hydraulic jump.

One thus has two equations where the unknown is $h(x)$. Nevertheless, it is important to note they actually are third-order polynomials which naturally have three solutions (either real or not). Among them, one is always negative and should not be taken into account, the two other solutions depend on whether the flow is subcritical or supercritical. Hence, the water height solutions in region I are obviously the ones for subcritical flow whereas the water heights inside region II are supercritical solutions.

Finally, one needs to precisely locate the hydraulic jump location x_{jump} . This is easily done using the *conjugated heights*. The idea is to associate to the inlet water height another value calculated by the following equation:

$$h_{\text{conj}}(x) = \frac{h_I(x)}{2} \left(\sqrt{1 + 8Fr(x)^2} - 1 \right) \quad (6.25)$$

such that the hydraulic jump is the x-coordinate of the intersection point between this conjugated height and the outlet water height, as shown in Figure 6.18.

Remark about the theoretical model

It is important to notice that the theoretical model is a simple 1D model which does not take into account nor vorticity and recirculation. Therefore the future outputs should carefully be discussed considering this limitation.

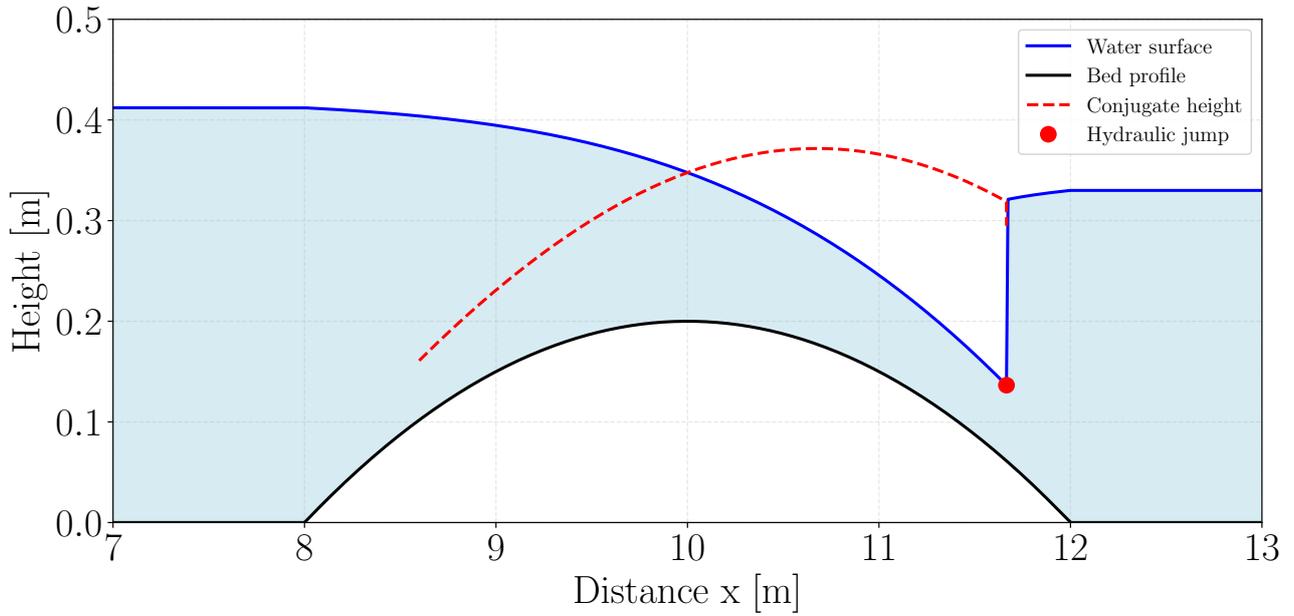


Figure 6.18: Hydraulic jump location using conjugated height principle.

6.3.3 Numerical results

Simulation parameters involved

In Table 6.19 are displayed the parameters used for the simulation. To simulate inviscid fluid and slip wall conditions, one naturally sets to 0 all viscosity parameters.

Notably, while the $\epsilon_{\text{boundary}}$ needs to be zero, a small value for the ϵ is not forbidden and is actually preferred. Indeed, the simulation reaches its steady state more quickly with a higher value of x_{sph} . The small numerical dissipation is too small to modify substantially the results. Additionally, it has been noticed that simulations with more than 100 000 particles need $\epsilon \approx 0.5$ to produce realistic behaviour. The problem is that the micropolar vorticity model creates and preserves all type of vortex scales and thus preserve micro scale vortex (that should normally be dissipated). It has been difficult to find a precise value for the vorticity viscosity factor that both ensures large vortex scales and reduces small vortex scales. Thereby, $\epsilon \approx 0.5$ helps to dissipates small vortices (which are not present in simulations involving only 20 000 to 30 000 particles).

In contrast to the turbulent flow simulation, the vorticity plays a major role in this particular case such that the associated parameters had to be carefully chosen. Indeed, using the same parameter values as before would lead to completely unphysical behaviour where angular bulk displacements would completely dominate linear bulk displacements (which is for sure not the case in real conditions).

Finally, the *stopAt* parameter is chosen based on a characteristic time. Since, the geometry is initially empty one needs to first consider the time for inlet particle to reach the exit section. Then, the time for downstream hydraulic information to propagate upstream at speed $c - V$ (with $c = \sqrt{gh}$ the gravitational wave celerity) and establish the steady hydraulic jump configuration:

$$t_{\text{charac}} = t_{\text{fill}} + t_{\text{wave}} = \frac{L_{\text{upstream}}}{U_0} + \frac{L_{\text{downstream}}}{c - U_0} = \frac{8}{0.36} + \frac{3}{\sqrt{g \times 0.33} - 0.36} \approx 25 \text{ [s]}$$

Simulation parameters	Value	Units
particleRadius	0.004	m
timeStepSize	0.001	s
cfFactor	0.6	-
cfMaxTimeStepSize	0.01	s
maxIterations	100	-
maxError	0.05	-
maxIterationsV	100	-
maxErrorV	0.05	-
viscoMaxIter	500	-
viscoMaxError	0.05	-
Flow parameters	Value	Units
stopAt	100	s
Fluid parameters	Value	Units
viscosity	0	m ² /s
viscosityBoundary	0.0	m ² /s
inertiaInverse	1	m ⁻²
xsph	0.04	-
xsphBoundary	0.0	-
drag	1.0	-
vorticity	0.02	m ² /s
viscosityOmega	0.1	m ² /s

Figure 6.19: Parameter used to simulate the free surface flow.

By security, one uses $stopAt = 4 \times t_{charac} = 100$ [s].

Flow development

Since the simulation begins with no particles at all, the flow has to establish itself:

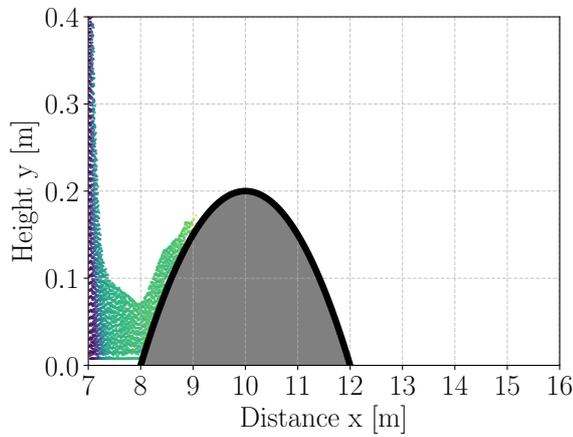
1. First, the particles fill the upstream section of the flow.
2. When sufficient particles are in the inlet section, they begin to pass through the obstacle.
3. The particles then travel the whole downstream section and endure the boundary condition at the very end ($x = 25$ [m]). The outlet water height reaches its final value and the downstream section will influence the upper stream sections.
4. Finally, the downstream flow reaches the obstacle and the hydraulic jump begins to appear.

These four steps are depicted in Figure 6.20.

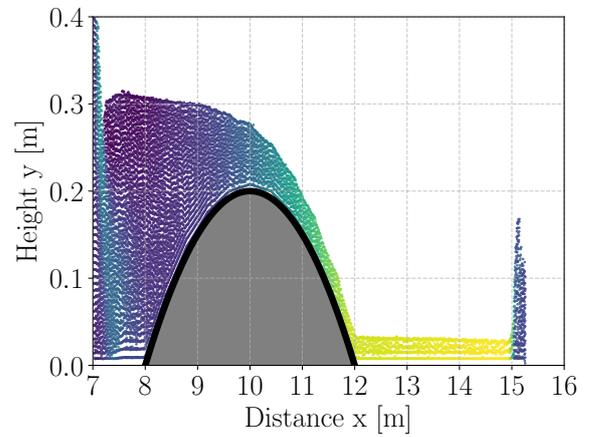
Steady state assumption

Once again, it is mandatory to assess the steady state of each variable studied. However, the use of a volume of control (e.g. a local analysis), as performed in the previous turbulent section, is not relevant. One easily understands that the time derivative of, for example, the velocity will always fluctuate in the vicinity of the hydraulic jump.

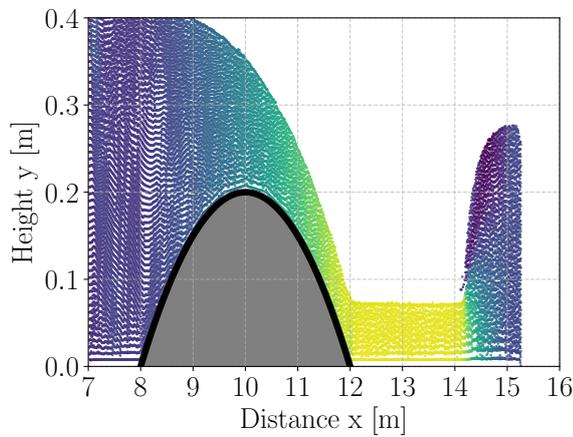
Hence, it is proposed to study the total energy of the system and determine its time convergence. To do so, one neglects the friction (despite the applied XSPH coefficient) and the internal energy



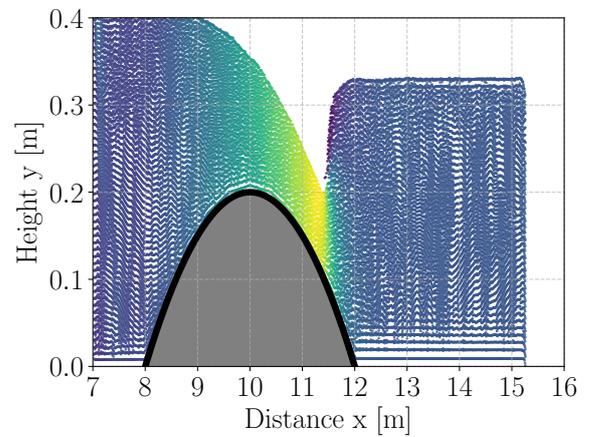
(a) First step ($t = 1[s]$). Arriving particles first fill the upstream region until reaching the top of the obstacle.



(b) Second step ($t = 5[s]$). Flow begins to mount the obstacle and go to the downstream region.



(c) Third step ($t = 15[s]$). The flow endures the outlet boundary condition and transmits the information upstream.



(d) Fourth step ($t = 100[s]$). Finally, the outlet boundary condition information has been transmitted to the obstacle.

Figure 6.20: Flow development in the free surface configuration in four steps. The color map is based on the velocity.

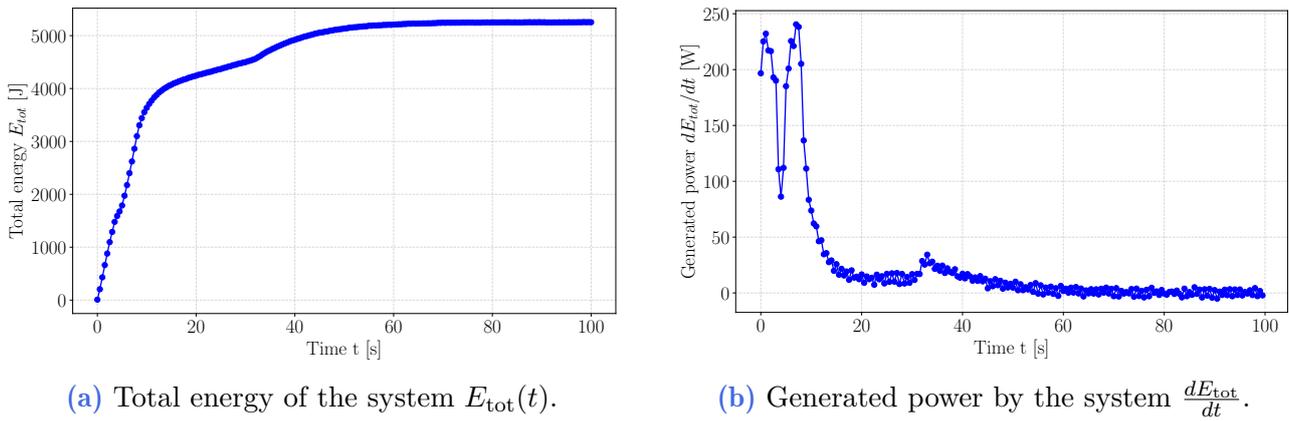


Figure 6.21: Steady state assessment of the free surface flow using the total energy of the system.

which is small compared to the kinetic and potential energies. One observes in Figure 6.21 that the energy and its time derivative both reach a steady state after 80 [s].

Incompressibility assumption

Given the free surface flow characteristics, it is even more important here to analyse this major assumption. One immediately focuses on the *global approach* and analyse the volume and mass flow rates. Indeed, it is expected that particles at the air-water interface will have a smaller density due to neighbour deficiency³. Hence, the spatial derivative of the density $\frac{\partial \rho}{\partial y}$ would never be small.

With that being said, let us represent in Figure 6.22 the two flow rates. It can be observed that both flow rates fluctuate more than in the turbulent case. Nevertheless, the mean errors remain substantially small (about 0.77%), which is totally acceptable considering the free surface contribution.

It is noted that the numerical volume/mass flow rates are both slightly above the initial value in the vicinity of the hydraulic jump. This could be associated with the difficulty of the DFSPH algorithm in correcting pressure values against the high velocity gradients in this area.

Uniform velocity profile

The uniform velocity assumption is mandatory to find the solution for the 1D-model, hence the simulation should also follow this behaviour. It is proposed to use the *rectangle method* and compute the standard deviation of the velocity for all slices. Since the velocity should theoretically not vary with height, its mean value is the most representative metric, and thus the standard deviation is the perfect metric to analyse. This metric is presented in Figure 6.23. *If you read this, I owe you a drink.* One observes that the aforementioned assumption is verified almost everywhere in the domain. The only area that does not exhibit uniformity is the area near the elliptic obstacle, which is expected. It is more reasonable to assess that the assumption is not realistic in this area. One concludes that this assumption is thus verified in the simulation.

³It should be recalled that SPH interpolation is based on the number of neighbouring particles. If there are too few neighbours, the interpolated value is expected to be smaller than it should be.

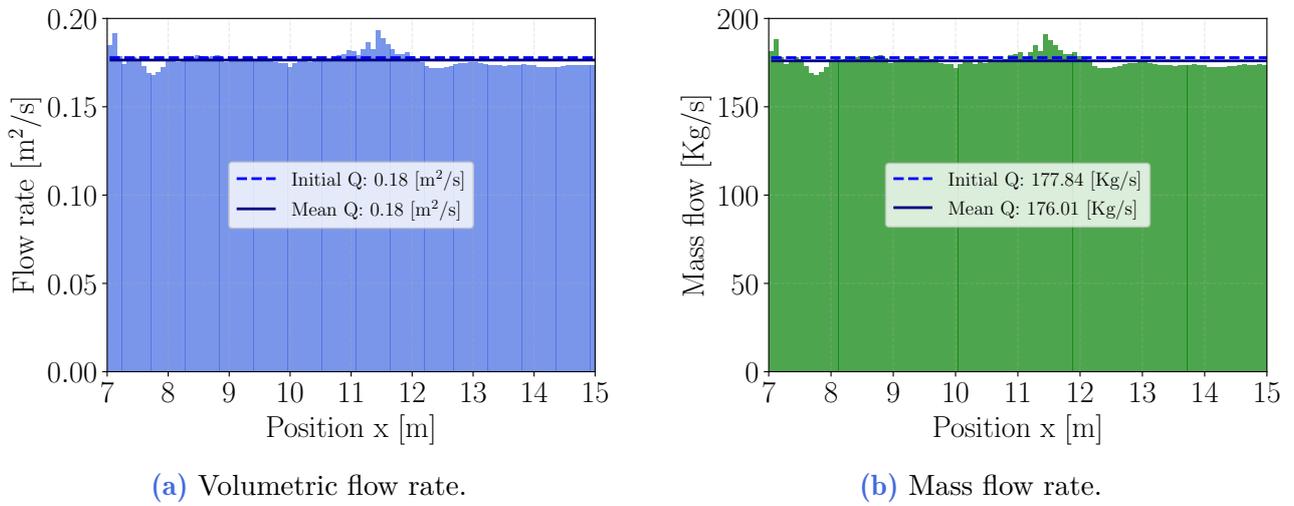


Figure 6.22: Computation of the mass and volumetric flow rates yielding error about 0.779%.

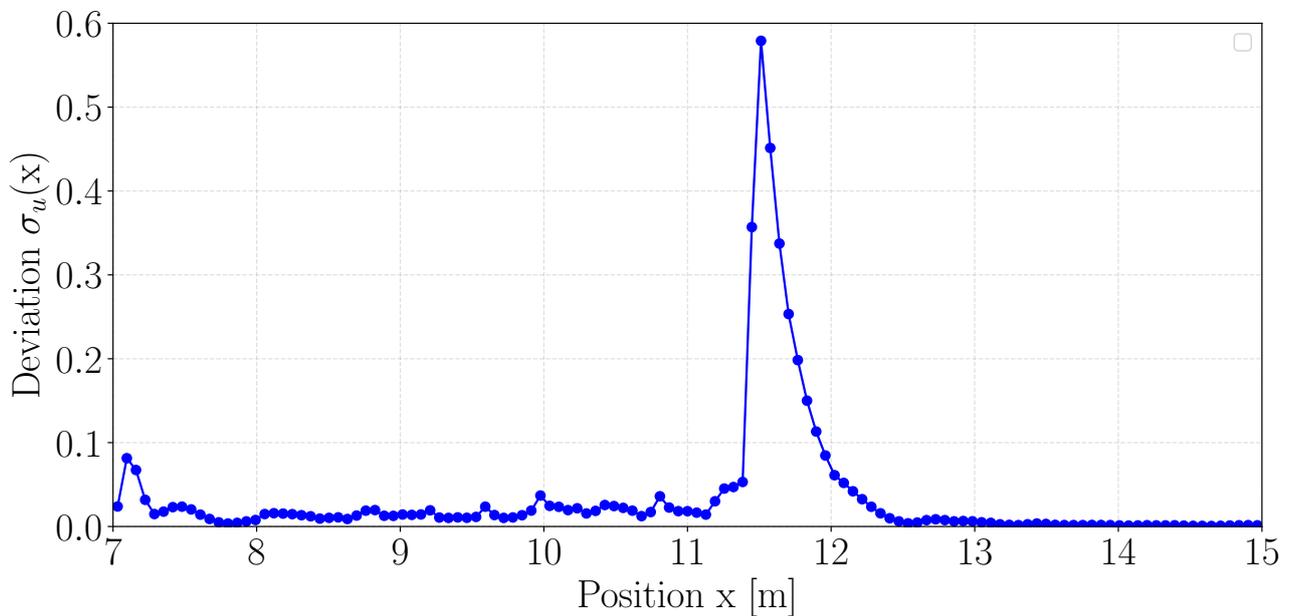


Figure 6.23: Standard deviation of the velocity $\sigma_u(x)$ across the whole domain. The deviation is almost zero everywhere except in the vicinity of the elliptic obstacle.

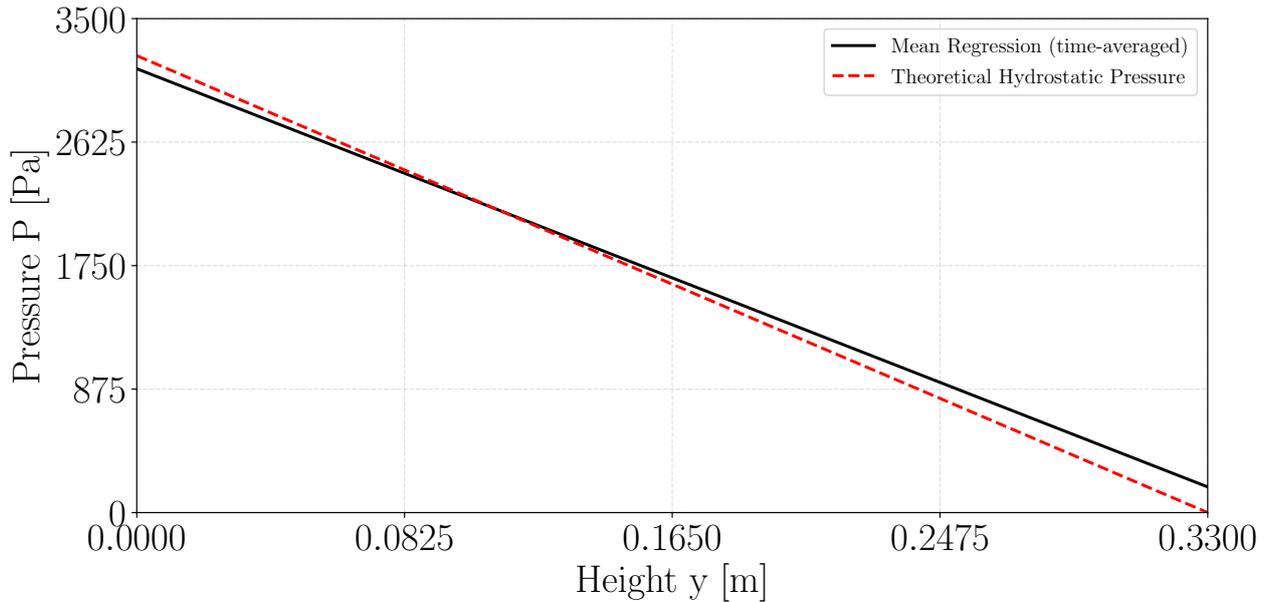


Figure 6.24: Spatially averaged pressure over the whole domain. The mean RMSE is 91.687 [Pa], which demonstrates the expected result.

Hydrostatic pressure

In this configuration, the pressure is assumed to vary linearly with respect to the height of the fluid particle. One proposes to determine whether this assumption is verified in the simulation.

Once again, one uses the so-called *rectangle method* to evaluate the pressure for all slices of the sampled domain. Then, the pressure was time-averaged over 5 seconds to smooth the results. Unlike the velocity profile, the standard deviation is not an accurate metric to use since the pressure at each position varies with the height. Rather, a metric that can vary according to a specific function, the *Root Mean Square Error* (RMSE), is perfectly adapted here.

The first proposed quantity to observe is the spatially averaged pressure over the whole domain in Figure 6.24.

Another quantity is the local pressure for each slice of the domain. One may compute the RMSE along the horizontal channel and observe its trend. In Figure 6.25, it is notable that the RMSE is quite high in the upstream section, whereas it diminishes significantly in the downstream section.

Even though the downstream RMSE is expected to be small due to the imposed uniform outlet velocity boundary condition, it is surprising that the results are quite moderate for the upstream regions since it has already been proven that the velocity was indeed uniform in both regions.

Numerical water height

Since all theoretical assumptions has been verified from a numerical point of view, one may now take care of the most important variable in this context being the water height. In Figure 6.26 is illustrated the comparison of the numerical and analytical solution.

The numerical upstream/downstream water height does perfectly match the numerical solution with an average error below 1%. However the water height nearby the hydraulic jump does not completely fit the analytical one. Which is expected reminding that the 1D model does not take into account recirculation. Also, the 1D model indicates an abrupt jump just after

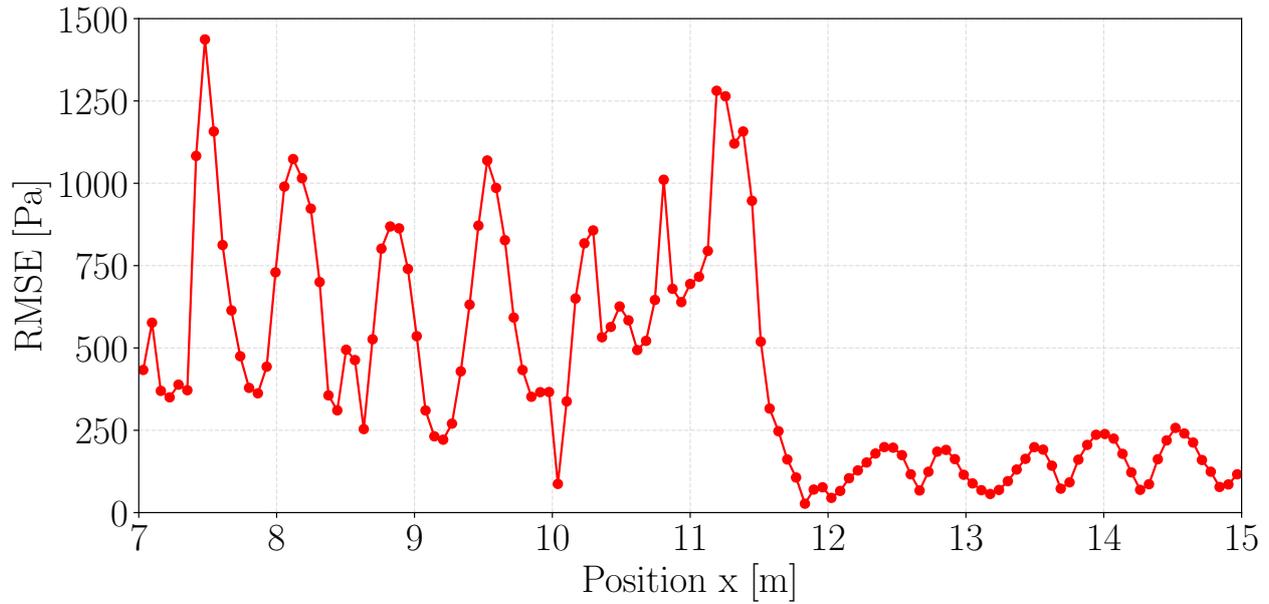


Figure 6.25: Local RMSE along the whole horizontal domain. The RMSE is much higher in the upstream region compared to the downstream region.

the hydraulic jump, which is obviously not physically realistic. Finally, the positioning of the hydraulic jump is slightly shifted to the right with a relative difference of 2.27% w.r.t. the theoretical model.

Numerical Froude number

As stated in the associated theoretical section, the flow endures a transition from sub to supra critical, meaning a Froude number varying from smaller until greater value than one with the critical point being the extremum of the elliptic obstacle. It could be interesting to evaluate the numerical Froude number based on the computed water height. It is remarkable how the critical Froude number is perfectly placed at the extremum of the obstacle, as shown in Figure 6.27. The impact of the hydraulic jump is present here since the Froude number goes from almost a value of two to again a smaller value than one, indicating the presence of the hydraulic jump.

One may notice that the Fr_{SPH} is smaller than the model one in the vicinity of the hydraulic jump. As already mentioned, the more plausible cause for this is the DFSPH hardly adjusting pressure due to the high velocity gradients.

Total head conservation

Another interesting variable to analyse is the total head. From Bernoulli's Equation (6.23), the upstream head is conserved. One observes that the numerical head correctly fits the theoretical values in Figure 6.29 before and after the hydraulic jump. The head loss due to the hydraulic jump is also present and correctly captured. A particular behaviour is observed just after the hydraulic jump in the region $x \in 11.5 \pm dx$: the downstream total head seems to slightly increase, but it does not actually. In fact, there are two distinct flows after the hydraulic jump: the main one, which dives beneath the water surface and goes downstream and will later diffuse,

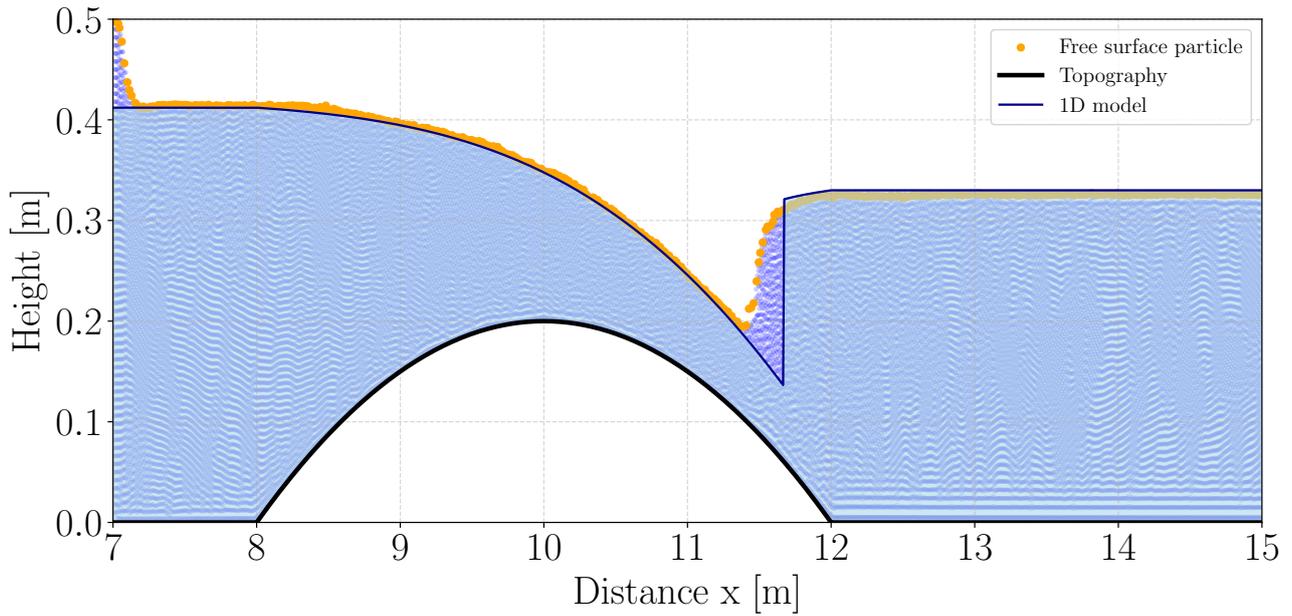


Figure 6.26: Comparison of the numerical water height with the analytical one.

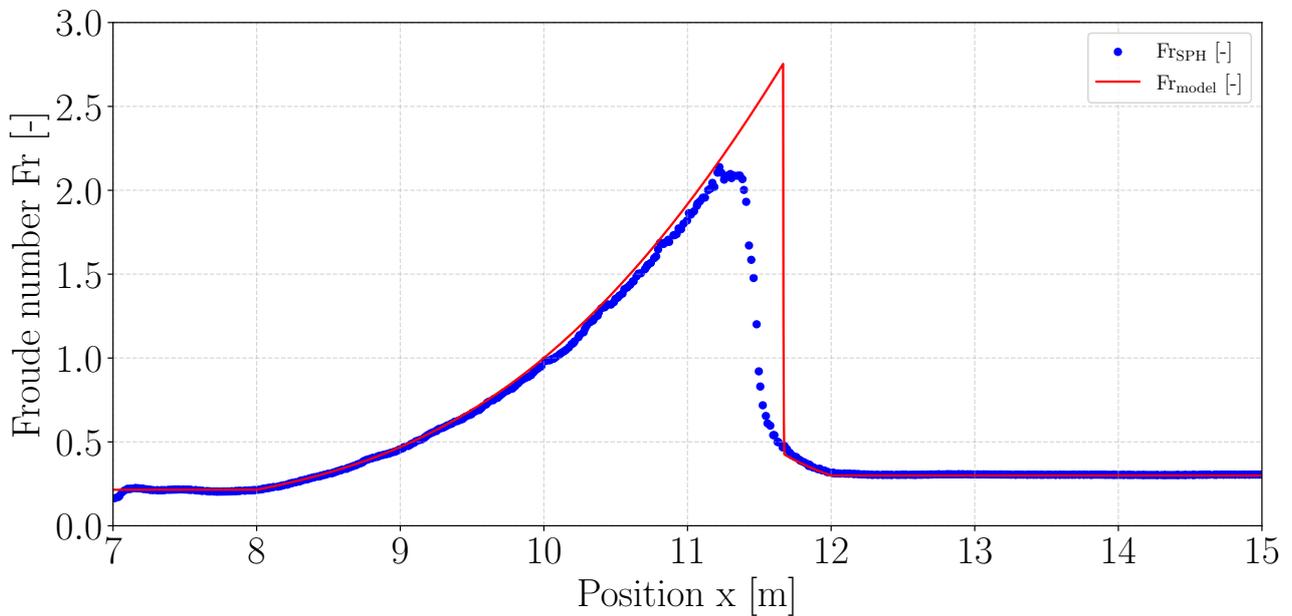


Figure 6.27: Numerical Froude number. The critical value $Fr = 1[-]$ is well captured at extremum of the obstacle at $x = 10$ [m]

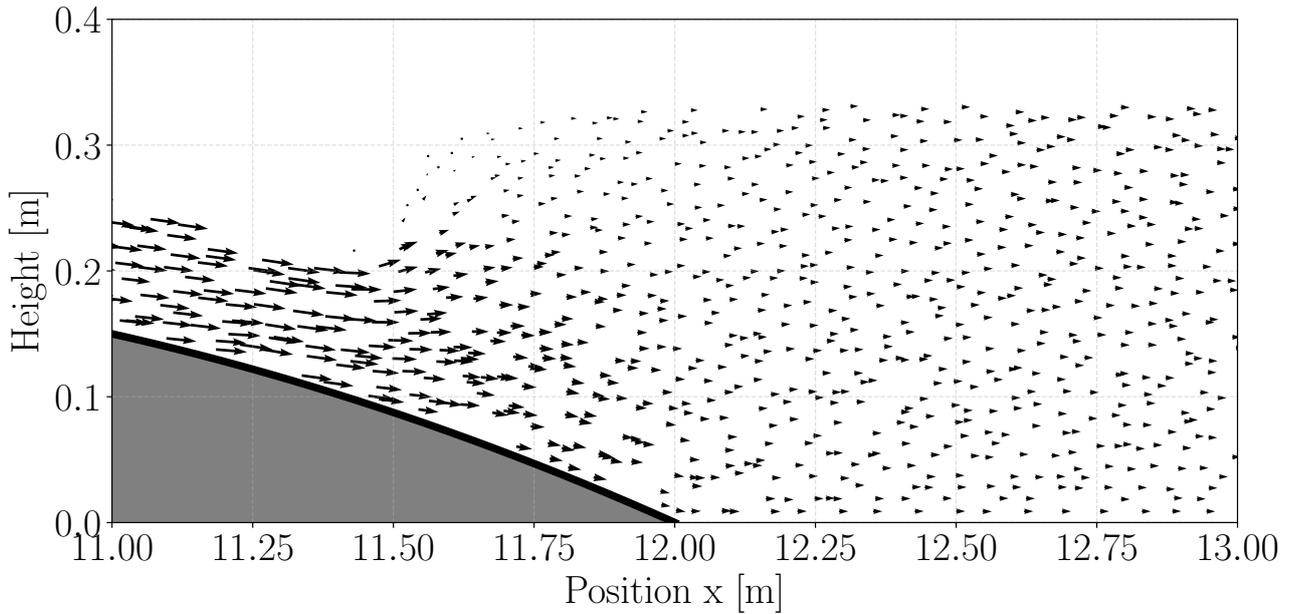


Figure 6.28: Quiver plot in the critical region. It can be observed the existence of two flows. The one near $x = 11.75$ [m] at the free surface which is stationary. The second one, being the main flow, lies below the former and goes downstream (indicated by the arrow directions).

and the secondary one, which is above and initially goes upstream (in the opposite direction)⁴ but remains stationary after reaching the steady state. This behaviour is observed in Figure 6.28. The way the total head is evaluated is by using the averaged velocity on a slice of the domain to reduce the impact of this secondary flow

6.3.4 Limitations of the model in the free surface simulation

After analysing outputs of the free surface simulations, several good results had been obtained. However, three major limitations of the software should be discussed.

The first one is about the XSPH coefficient. As already mentioned in Section 6.3.3, for an increased number of particles, the ϵ factor also has to be increased to attenuate potential instabilities and also smoothed area with high velocity gradients. As a consequence, systems with high spatial resolution can not be considered as inviscid.

Then, the software does not consider energy conservation. A manner of seeing that is to observe the total energy over the whole domain at a specific timestep which is presented in Figure 6.30. One sees that the total energy per slice is not conserved at all along the domain, which is important to consider while locally analysing data.

The third one has already been discussed and is the difficulty of the software to reproduce locally hydrostatic pressure by itself in the case of a uniform flow.

⁴This flow arises from the outlet boundary condition, previously discussed, that transmits information upstream

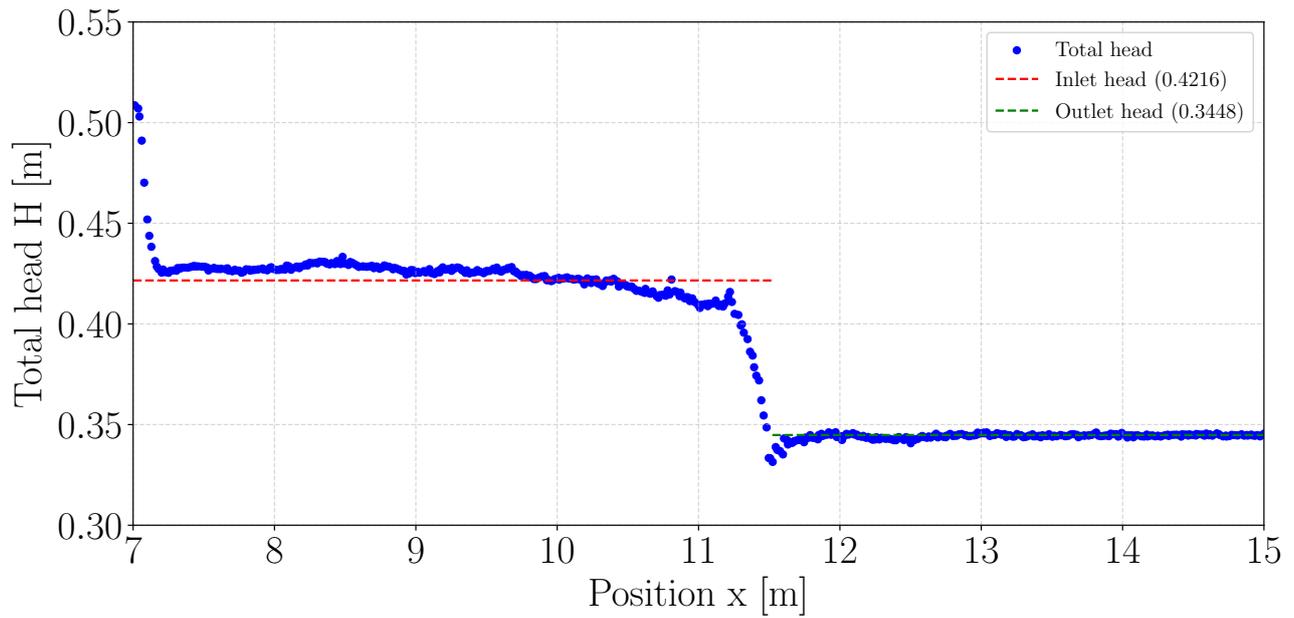


Figure 6.29: Numerical total head correctly fitting the expected value and demonstrating a head loss due to the hydraulic jump.

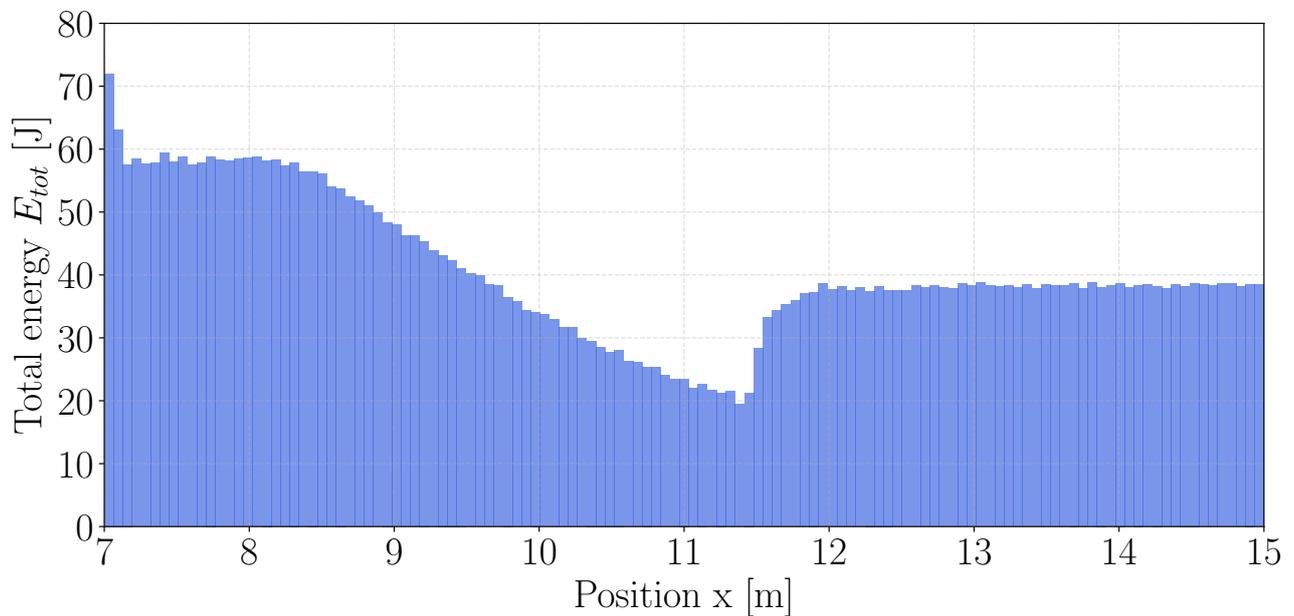


Figure 6.30: Total energy of the system at a single timestep. One observes that the energy highly decreases in the hydraulic jump and then stabilises just after but with a smaller value than upstream.

Final validation test

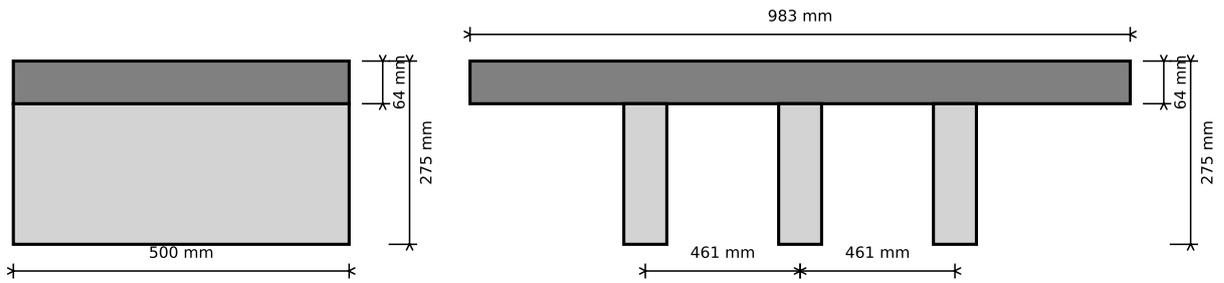
After examining the performance of the software in separate simulations and since the simulation results have been proven to gain confidence in the software, it is proposed to gather them all and also directly consider multiple dynamic body interactions. It is proposed to simulate a laboratory configuration and more specifically the one from the Engineering Hydraulics Laboratory of the University of Liège. The idea of this configuration was brought by the master thesis of Florence Dütz [41] who was specifically studying the experimental formation of obstacles with wood logs.

The purpose of this simulation is to obtain two different configurations. One being a loaded bridge, meaning that the underside of the bridge is pressurized, in order to see if any obstacle formation occurs in such a case. The other case being a free surface flow under the bridge. The differences between these two case will be highlighted and discussed.

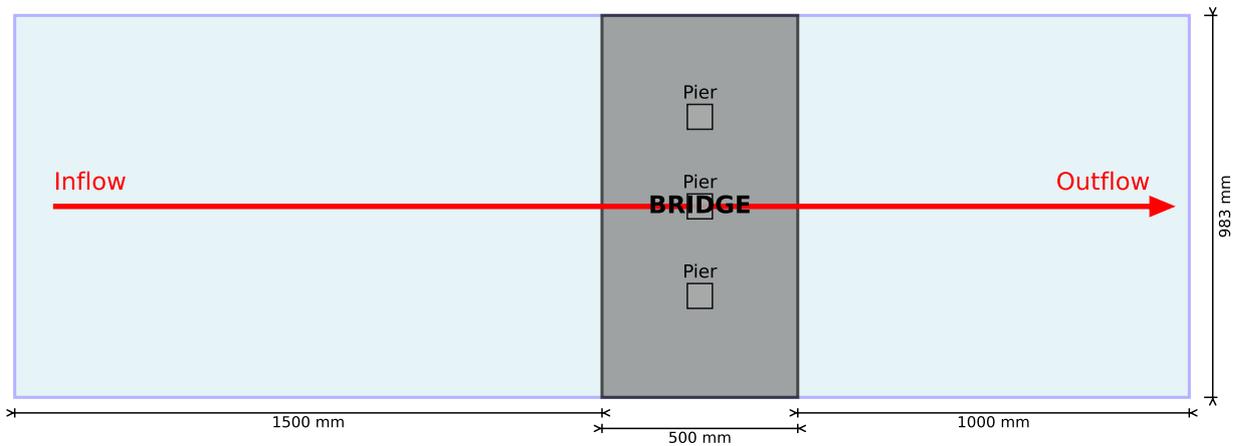
7.1 Problem statement

The geometry of the bridges and the channel is presented in the Figure 7.1. One simulates an inlet flow with a water height $h_0 = a \times 0.1$ [m]. The variable $a = 1$ for the free surface situation and $a = 2$ for the pressurized case. The inlet velocity is $U_0 = 0.075$ [m/s] with a two-dimensional emitter of size $(L_y, L_z) = (2 \times h_0, 1)$ such that the inlet flow rate is $Q_0 = (ah_0 \times L_z) \times U_0 = (7.5, 15) \times 10^{-3}$ [m³/s]. The considered boundary conditions will be discussed depending on the configuration.

Seven rigid cylinders of length L and diameter D respectively equal to 0.4 [m] and 0.035 [m] with a density of 800 [kg/m³] are chosen. The density of wood is usually lower than this, but one specifically uses this value to get a higher wetted wood surface to visually enhance the results.



(a) Side and front view of the bridges



(b) Top view of the bridge inside the horizontal channel and the prescribed flow direction.

Figure 7.1: Complete set of 2D views of the bridge and flow configuration.

7.1.1 Simulation parameters involved

Most of the parameters are common to the previous preliminary cases except that now one considers the PBD simulation parameters already discussed in Sections 4.10.2 and 4.10.4.

The boundary viscosity has been chosen to reproduce the non-slip wall conditions. The vorticity parameters have been tuned to obtain small recirculation (that should be induced by the wood logs) while avoiding non-physical vortices caused by the flow itself. The simulation time has been chosen based on the initial velocity U_0 and the distance between the inlet position and the bridge:

$$t_{\text{charac.}} = \frac{L}{U_0} = \frac{1.5}{0.75} = 2 \text{ [s]}$$

To ensure capturing any obstacle formation, one uses $\text{stopAt} = 12.5 \times t_{\text{charac.}} = 25 \text{ [s]}$.

The particle radius has been chosen to be about 8-10 times smaller than the wood log diameter. Then, the restitution coefficient has been set to 0 to avoid any elastic rebound. If this value is strictly positive, some unrealistic rebounds occur within the simulation. Additionally, one uses a friction coefficient of 0.8. A value of 1 would lead to stacks of wood logs moving as a single mass, while a value of 0 leads to complete slip solid boundary conditions.

Regarding the so-called *contact* parameters, the parameter *contactTolerance* is equal to the particle radius because smaller values lead to particles passing through the solid bodies, whereas greater values lead to premature repulsion. Finally, the *contactStiffnessRigidBody* and *contactStiffnessParticleRigidBody* are respectively set to 100 and 0.1 to reproduce strong repulsion for solids between themselves but smoother repulsion for fluid-solid interactions.

7.1.2 Numerical results (pressurized bridge case: $h_0 = 200 \text{ [mm]}$)

One proposes first to analyse the loaded bridge configuration since it is the most novel contribution compared to the previous results. The free surface case will be discussed in the following section.

In this case, the outlet velocity U_{outlet} is chosen according to the flow rate such that the outlet water height is equal to $h_{\text{outlet}} = h_0$.

Obstacle formation

After some time, the wood logs do indeed block the entrance of the bridge, as depicted in Figure 7.9. Qualitatively, one observes that the software can simulate obstacle formations and even reproduce cases where these obstacles pile up on top of one another.

Also, one mentions that too coarse spatial resolution (for instance $r = 10 \text{ [mm]}$) leads to unrealistic behaviour where wood logs undergo a kind of backward repulsion, as shown in Figure 7.4. This can easily be explained by the particles being too large and thus creating highly overestimated pressure forces at the solid body interfaces. One can also mention that with such a poorly discretized simulation, the tuning of the PBD parameters does not allow prevention of this behaviour. Hence, a first limitation arises, yielding a criterion about the ratio between the particle size and the solid characteristic length:

$$\frac{2r}{D} \lesssim 0.2$$

Simulation parameters	Value	Units
particleRadius	0.005	m
timeStepSize	0.0005	s
cffFactor	0.6	-
cffMaxTimeStepSize	0.01	s
maxIterations	2000	-
maxError	0.05	-
maxIterationsV	1000	-
maxErrorV	0.05	-
viscoMaxIter	500	-
viscoMaxError	0.05	-
Flow parameters	Value	Units
stopAt	25	s
Fluid parameters	Value	Units
viscosity	10^{-6}	m^2/s
viscosityBoundary	1	m^2/s
inertiaInverse	1	m^{-2}
xsph	0.08	-
xsphBoundary	0.01	-
drag	0.5	-
vorticity	0.04	m^2/s
viscosityOmega	0.08	m^2/s
restitution	0	-
friction	0.8	-
contactTolerance	0.005	-
contactStiffnessRigidBody	100	-
contactStiffnessParticleRigidBody	0.1	-

Figure 7.2: Parameters used to simulate the final flow with bridge.



(a) Initial timestep: the wood logs are perpendicular to the flow, ensuring contact between them and the bridge.

(b) Final timestep: one observes the partial recirculation in front of the wood logs as well as the drag effect at the exit of the bridge.

Figure 7.3: Snapshots of the bridge configuration at the beginning and the end of the simulation with $h_0 = 200$ [mm].

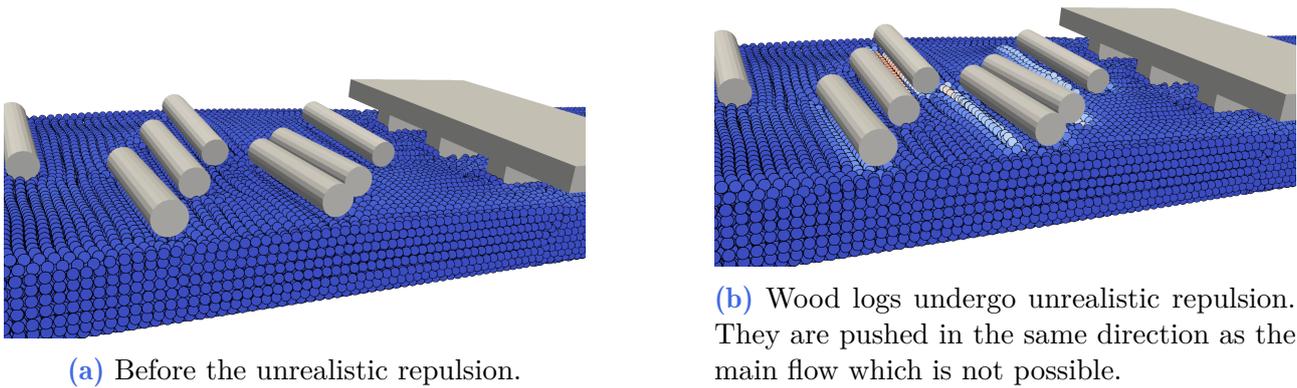


Figure 7.4: Snapshots of the bridge configuration with coarse spatial discretization $r = 10$ [mm].

Pressurized bridge assumption

Before going any further, one needs to assess whether the underside of the bridge is pressurized since it was the desired configuration. Figure 7.5 depicts the water height distribution through the two bridge openings. It can be noticed that, fortunately, the two water levels are identical in the downstream and upstream regions far from the bridge. Also, the two distributions differ near the bridge entrance, which is explained by the asymmetric stacking of wood logs forming the obstacles.

One observes that the fluid particles correctly adhere to the underside of the bridge deck, creating pressurized flow.

Obstacle influence on the flow

While visual and qualitative results give confidence in the software, it is more interesting to obtain different quantitative outputs to analyse. It is first proposed to analyse the modified flow distribution due to the wood logs. To do so, one observes the streamlines at the final timestep at $t = 25$ [s].

Figure 7.6 shows the upstream particles trying to move forward while avoiding the wood logs. Also, one notes in the downstream regions that the three piers influence the outgoing flow with high recirculation just after them. Most notably, the formation of recirculation zones upstream of the rectangular obstacle shows the complex interaction between the incoming flow and the bridge structure. These vortices, characterized by closed streamline loops, indicate regions of reversed flow that contribute to the accumulation of debris mentioned in the simulation description.

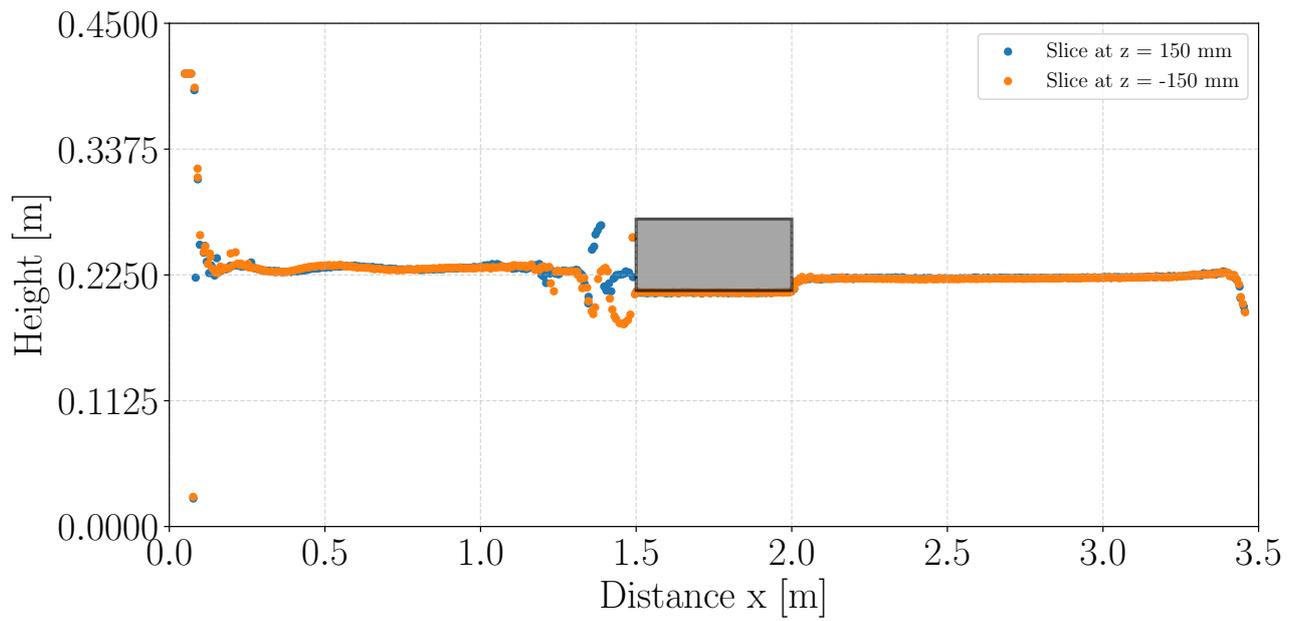


Figure 7.5: Water height distributions (in the loaded bridge case) through the two bridge openings. The slices have been obtained through the XY axis at $z = \pm 150$ [mm] $\pm 20r$.

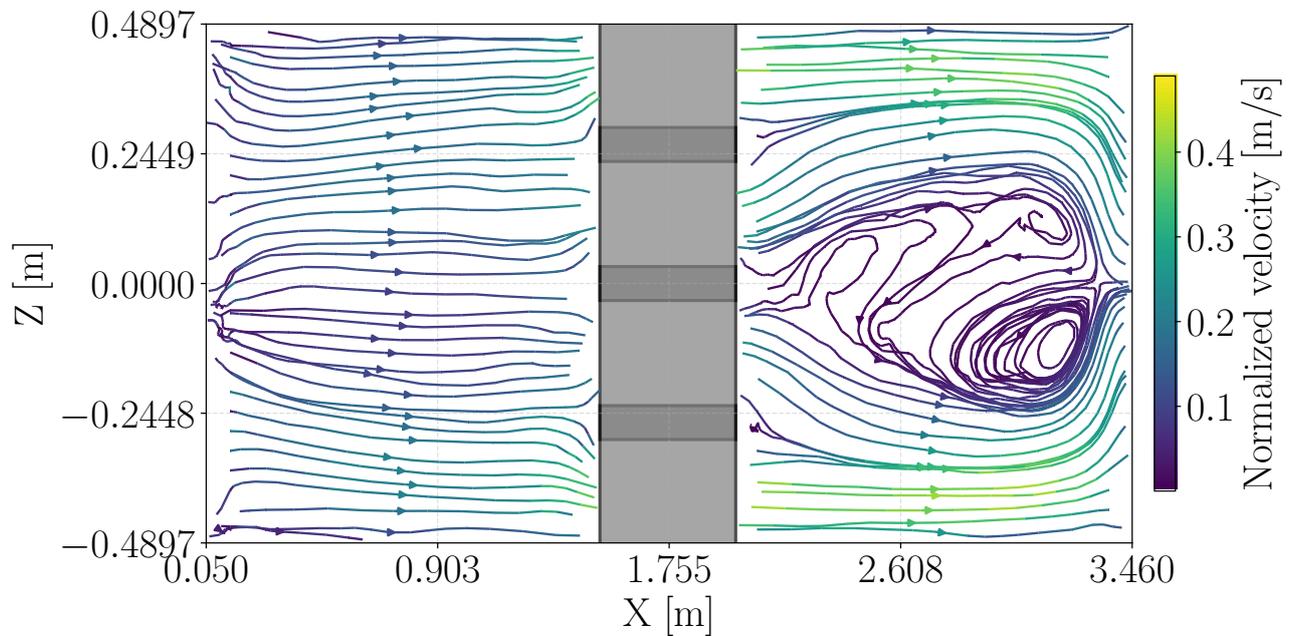


Figure 7.6: Streamlines (in XZ plane at $y = 250$ [mm] ± 4 particles) at the end of the simulation.

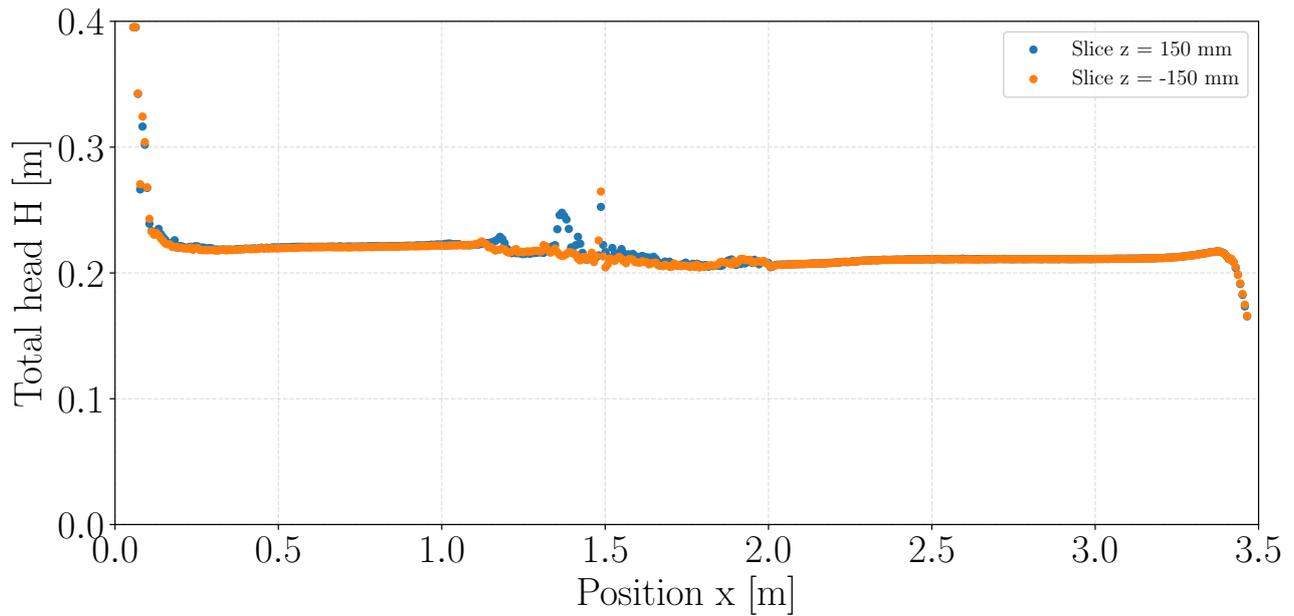


Figure 7.7: Total head distribution in plane XY at $z = \pm 150$ [mm] $\pm 20r$. These values have been time-averaged for $t \in [10, 25]$ [s].

Total head evaluation

In this specific configuration, the evaluation of the total head needs to be reconsidered to take into account the pressurized flow. Up to now, one considered the Bernoulli equation as:

$$H = z + \frac{u^2}{2g} + \frac{p}{\rho g} \quad (7.1)$$

while considering $p = p_{\text{atm.}} = 0$ (with relative pressure). However, this is no longer true, and one needs to take into account the actual pressure of the fluid particles pushing against the deck.

With this consideration, the total head distribution is presented in Figure 7.7. Due to the local pressure instabilities already discussed in Section 6.3.4, the pressure contribution has to be carefully processed. One mitigates these noisy oscillations by time-averaging the total head distribution over 15 seconds.

Physically, the fluid should undergo two different head losses, with the first one at the bridge entrance and the other at its exit. Indeed, the first head loss is caused by a sudden contraction (increased by wood logs) where turbulence, and thus high dissipation zones, occur. The second head loss is due to a sudden expansion which, for the same reason, creates turbulence and some recirculation that dissipates a fraction of the energy of the flow. This behaviour is highlighted in Figure 7.8.

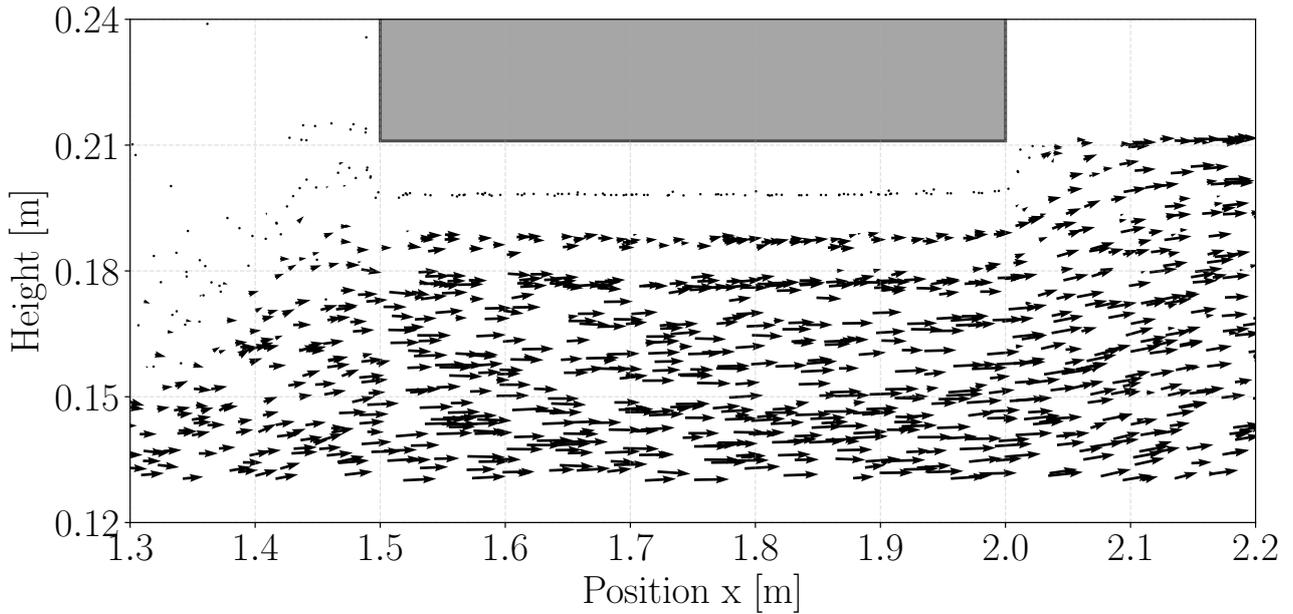


Figure 7.8: Quiver plot XY at $z = \pm 150$ [mm] $\pm 20r$ at $t = 25$ [s]. The particles near the corners of the bridge have a highly reduced velocity compared to the others (indicated by the length of the arrows). The wood logs have an important impact in the upstream region of the bridge.

7.1.3 Numerical results (free surface flow case: $h_0 = 100$ [mm])

In the present case, no outlet boundary conditions have been imposed, such that the fluid is *left to itself*. This configuration can be seen as a portion of a much larger rectangular channel.

Obstacle formation

Even though this new configuration is almost identical to the previous one, it is interesting to analyse how the water height influences the flow dynamics. Figure 7.9b shows that most of the wood logs have passed through the bridge, unlike in the previous case. This occurs because the absence of the deck as an obstacle provides the wood logs with a larger available cross-sectional area. Consequently, they are pushed toward the sides of the bridge piers and follow the main streamlines rather than accumulating at the entrance.

Obstacle influence on the flow

In this case, Figure 7.10 depicts a slightly different behaviour. The streamlines in the upstream region indicate that the main flow splits into two preferred paths through the outer regions of the bridge openings. This pattern makes sense compared to the previous case, where all the wood logs prevented this type of flow separation.

Furthermore, the higher recirculation observed in the downstream region is explained by the wood logs being constrained within the simulation domain boundaries, whereas fluid particles are not. The wood logs cannot leave the simulation domain and thus bounce against the domain edges, creating artificial flow disturbances. Therefore, only the upstream streamlines can be physically interpreted with confidence.

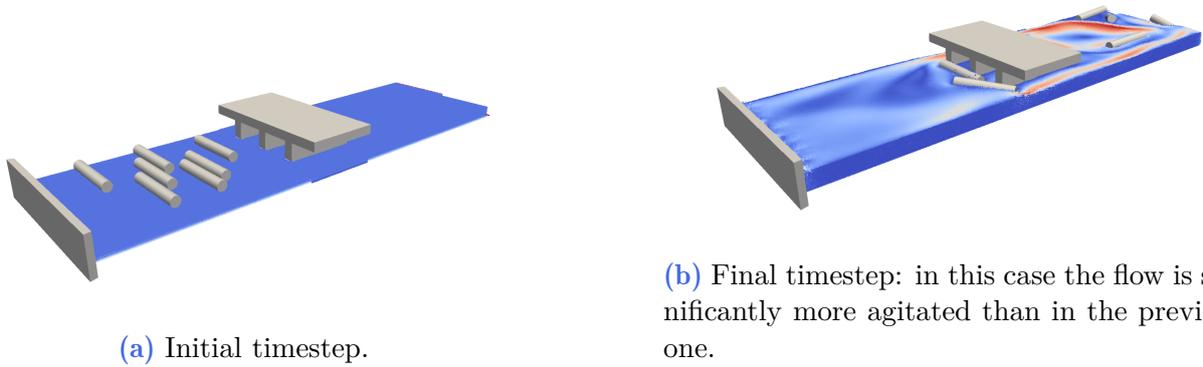


Figure 7.9: Snapshots of the bridge configuration at the beginning and the end of the simulation with $h_0 = 100$ [mm]. One observes the wood logs being swept by the current in the downstream region.

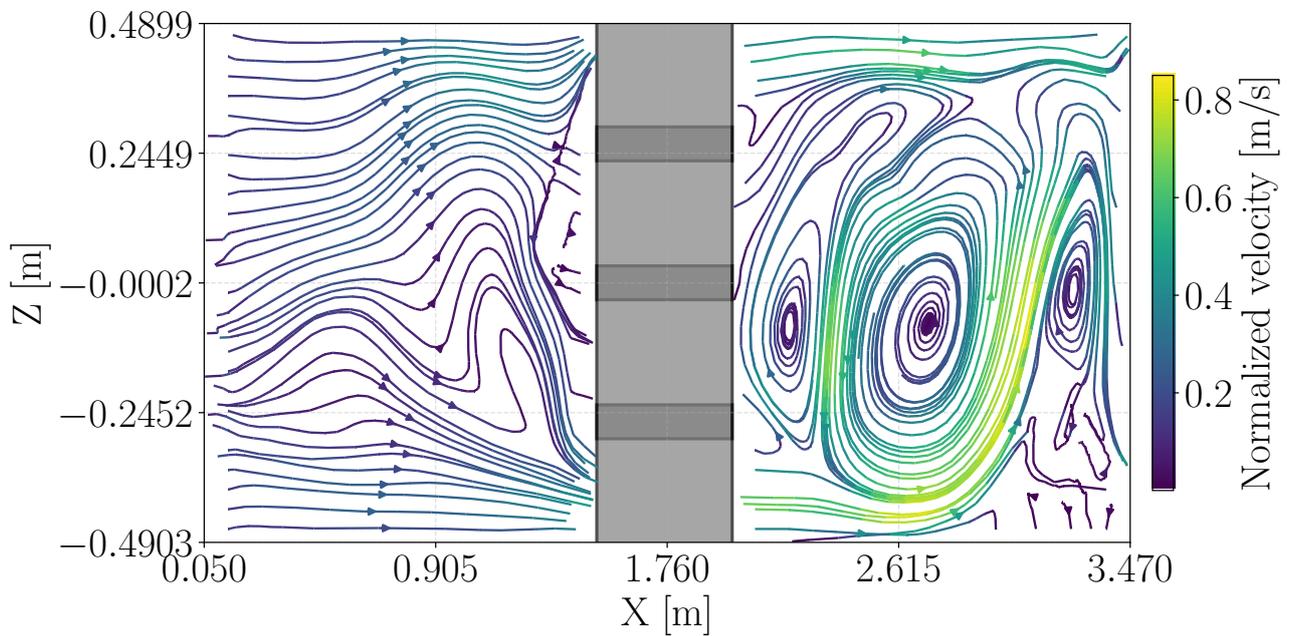


Figure 7.10: Streamlines (in XZ plane at $y = 120$ [mm] ± 4 particles.) at the end of the simulation. The blank areas before the bridge correspond to the wood log locations with particles passing underneath.

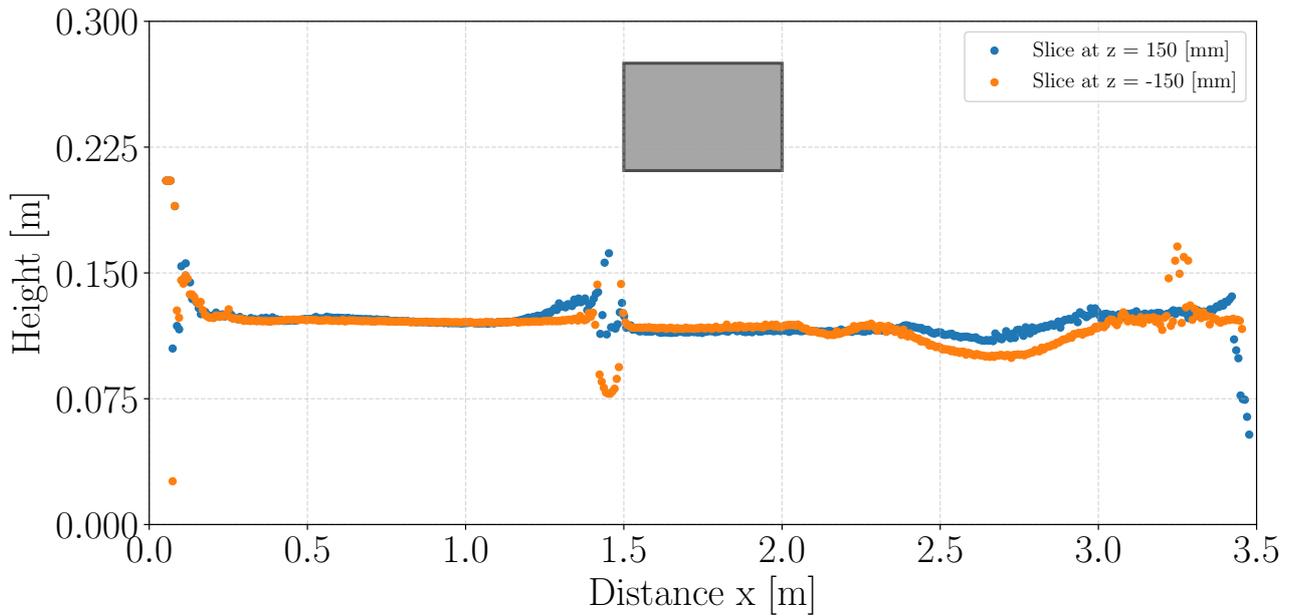


Figure 7.11: Water height distributions (in the free surface flow case) through the two bridge openings. The slices have been obtained through the XY axis at $z = \pm 150$ [mm] $\pm 20r$.

Free surface behaviour

Figure 7.11 shows the expected free surface behaviour. Away from the influence of the wood logs, the water height remains nearly uniform throughout the domain. Note that the region $x \in [2.5, 3]$ [m] may not be analysed correctly due to the issues discussed in the previous section. Additionally, no hydraulic jumps are expected in this flow configuration, and Figure 7.12 confirms this expectation since the Froude number (at slices $z = \pm 150$ [mm]) remains below unity throughout the domain. The observed asymmetries between the two slices are consistent with the asymmetric nature of the flow itself, caused by the irregular distribution of wood logs. The Froude number fluctuations in the downstream region should not be over-interpreted due to the previously mentioned instabilities.

Total head evaluation

Due to the wood logs accumulated at the bridge entrance, a head loss should be observed resulting from the sudden contraction they create. Indeed, this is exactly what is observed in Figure 7.13. This total head distribution follows the same trend as in the second preliminary case discussed earlier in Figure 6.29: a constant head at its highest value in the upstream region, followed by a sudden head loss, and finally stabilization downstream of the obstruction.

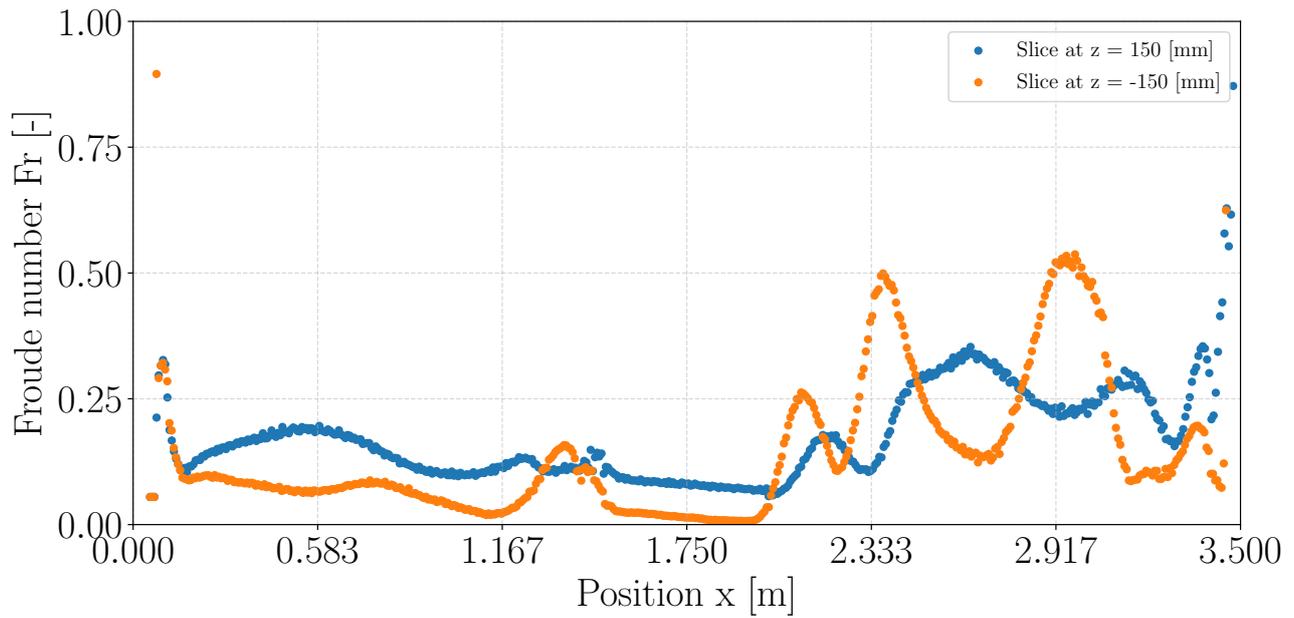


Figure 7.12: Water height distributions (in the free surface flow case) through the two bridge openings. The slices have been obtained through the XY axis at $z = \pm 150$ [mm] $\pm 20r$. The oscillations in the upstream region should not be considered due to the presence of the wood logs there.

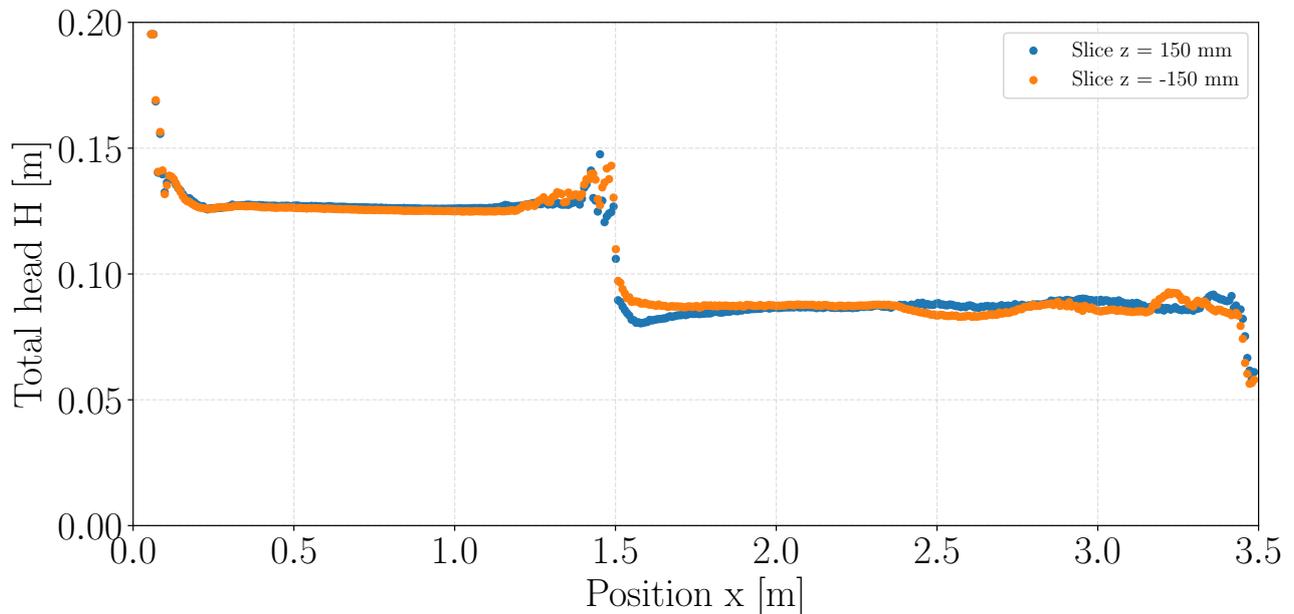


Figure 7.13: Total head distribution (in the free surface flow case) in plane XY at $z = \pm 150$ [mm] $\pm 20r$. These values have been time-averaged for $t \in [10, 25]$ [s].

7.1.4 Limitations of the two bridge configuration

Despite several successful results, such as the desired obstacle formation, typical total head distribution behaviour, and main recirculation patterns, some hydraulic features were not captured due to insufficient spatial resolution. For instance, smaller recirculation zones at the bridge exit (in the vicinity of the piers) were not resolved. Reducing the particle radius would significantly increase computational time.

Furthermore, as shown in Figure 7.7, particle pressure underneath the bridge remains substantially unstable, and data processing in these regions should be interpreted with caution.

Additionally, as observed in Figure 7.10, simulations where rigid bodies become trapped in the downstream region should be discussed carefully to prevent biased analyses.

Finally, the behaviour of the *restitution* and *friction* parameters has not been studied and needs further investigation to analyse simulation stability. While higher values of these parameters could potentially yield more physically realistic results, particularly for solid-solid interactions, excessively high values could quickly lead to unrealistic behaviour.

Limitations encountered

Some limitations of the previous chapters have been identified and mentioned. These limitations come either from the characteristics of the SPH method itself; the implementation of the software or from the configurations. This section summarizes the key limitations encountered across the preliminary validation tests and the final validation configuration.

8.1 Spatial resolution

One of the most significant limitations encountered is about spatial resolution constraints that affect the capture of near-boundary phenomena. Due to the nature of SPH simulations, it is not possible to refine spatial resolution only near the boundaries without significantly increasing the total computational cost. This limitation impacts the ability to reproduce additional turbulent metrics such as y^+ near wall regions, which would require a much higher number of particles and thus highly increased simulation time.

Consequently, only global metrics such as the velocity profile in channels can be evaluated with confidence. This constraint becomes particularly important when trying to capture smaller recirculation zones at complex geometries, such as those observed near bridge exits in the vicinity of piers in Figure 7.6 and 7.10.

8.2 XSPH Coefficient Dependencies

Another important limitation concerns the dependency on the XSPH coefficient for simulation stability. As demonstrated in Section 6.3.3, certain simulations involving more than 100,000 particles require $\epsilon \approx 0.5$ to produce realistic behaviour and prevent instabilities. This requirement creates a contradiction: while the initial idea is to simulate inviscid flows, systems with high spatial resolution cannot be considered truly inviscid due to the artificial numerical dissipation introduced by the XSPH term.

This artificial viscosity becomes necessary because the micropolar vorticity model creates and preserves vortex scales at all levels, including microscale vortices that should normally be dissipated. The difficulty in finding precise values for vorticity viscosity factors that both

ensure large vortex scale preservation and reduce small vortex scales necessitates reliance on the XSPH coefficient for stability.

8.3 Energy Conservation Issues

The software shows fundamental limitations regarding energy conservation, as demonstrated in Figure 6.30. The total energy per slice is not conserved along the domain, representing a significant concern for local data analysis. This non-conservation behaviour is even more highlighted in regions with high velocity gradients, such as hydraulic jumps, where energy losses are expected but should be physically associated to real dissipation mechanisms rather than numerical artifacts.

8.4 Pressure Field Instabilities

Local pressure evaluations also present difficulties throughout all the configurations tested. While spatially averaged pressure values prove reasonable agreement with theoretical values (as shown in Figure 6.24), local pressure fields show significant instabilities. This limitation is particularly important in:

- Hydrostatic pressure reproduction in uniform flows, where local RMSE values remain substantially high in upstream regions (Figure 6.25)
- Pressure calculations underneath bridge decks in pressurized configurations, where substantial instabilities persist despite time-averaging procedures
- Regions with high velocity gradients, where the DFSPH algorithm struggles to adjust pressure values appropriately

These pressure instabilities thus need careful interpretation of other local quantities that rely on these latter.

8.5 Rigid Body Interaction Constraints

The simulation of dynamic rigid bodies introduces geometric constraints that limit the involved configurations. A threshold has been found regarding the ratio between particle size and solid body characteristic dimensions:

$$\frac{2r}{D} \lesssim 0.2 \tag{8.1}$$

When this criterion is violated, unrealistic behaviours such as backward repulsion of wood logs occur (Figure 7.4). This limitation directly impacts the achievable spatial resolution for simulations involving small-scale debris or detailed geometric features.

Furthermore, the behaviour of restitution and friction parameters has not been systematically studied and requires further investigation for simulation stability. While higher values of these parameters could potentially yield more physically realistic results, particularly for solid-solid interactions, excessively high values quickly lead to unrealistic behaviour.

8.6 Boundary Effects and Domain Constraints

Simulations where rigid bodies become trapped in downstream regions present interpretation difficulties, as demonstrated in Figure 7.10. These bodies cannot leave the simulation domain and thus create artificial flow disturbances that bias the analysis.

This master thesis has evaluated the capability of the SPH method, specifically through the SPLisHSPlasH software, to simulate turbulent free-surface flows with dynamic rigid bodies. Through systematic validation and progressive complexity, one has demonstrated both the strengths and limitations of this particle-based approach for hydraulic engineering applications.

It is proposed here to summarize the journey undertaken throughout this master thesis, focusing on the most relevant chapters.

9.1 Complete Summary of This Thesis

9.1.1 Mathematical and Physical Framework

Chapter 3 established the mathematical and physical framework upon which this master thesis is built. Beginning from first principles, Sections 2.1, 2.2, 2.3 and 2.5 derived the well-known Navier-Stokes equations, with Equations(2.3) and (2.7) dictating how a Newtonian and incompressible fluid should behave.

The origins of turbulence in flows were then explained in Section 2.6, demonstrating how to assess the importance of this phenomenon using the Reynolds number through the scale separation relationship in Equation (2.26). Once the fluid particles were completely described, attention turned to rigid body dynamics in Section 2.7, which are governed by translational and rotational motion equations, respectively given by Equations(2.27) and (2.28).

Finally, the coupling between fluid particles and rigid bodies was established through hydrodynamic forces and moments, as expressed in Equations(2.32) and (2.33).

9.1.2 Fundamentals of the SPH Method

Chapter 4 captured the essence of the SPH method by explaining each of its core characteristics. First, the Lagrangian representation was explained in Section 3.1 and compared to the Eulerian approach through Equation (3.1).

The meshless feature of the method was briefly introduced in Section 3.2.

Section 3.3.2 then delved into the core of the method—the SPH interpolation—by utilizing the original exact integral representation from Equation (3.2), which involves a Dirac delta function. This integral was shown to be approximated by a kernel function to approximate any function or its gradient through Equations (3.8) and (3.9). The discretized version of this kernel approximation was then discussed in Equation (3.14), which provides a good approximation even though first and second-order consistency are not perfectly satisfied, provided the approximated equations remain properly mass conservative.

9.1.3 Preliminary Validation Tests

Once the theoretical foundation was established, Chapter 6 introduced the first preliminary cases where turbulence and free surface behaviour were analysed separately.

Section 6.1 highlighted a necessary modification to the code, ensuring that particles achieve their complete volume after some time, rather than only 80% of their theoretical volume.

The turbulent case was first examined using a two-dimensional pipe configuration, as described in Figure 6.1, with dimensions $(L, D) = (50, 3.2)$ [m] and an inlet velocity $U_0 = 5$ [m/s]. This configuration was particularly valuable since an analytical solution for the velocity profile had been derived using the Generalized Hydraulic Equations (GHE), as discussed in Section 6.2.2, yielding the velocity profile in Equation (6.11). The main assumptions of the flow and analytical solution—namely steady state, fully developed flow, and incompressible fluid—were successfully validated in Section 6.2.3. Specifically, steady state was found to be reached for $t > 10$ [s], fully developed flow was confirmed for $x \in [25, 45]$ [m], and the incompressibility assumption was verified from both local and global perspectives. Finally, the numerical velocity profile retrieved at $x = 25$ [m] for $t \in [15, 30]$ [s], presented in Figures 6.14 and 6.15, was found to fit the raw data perfectly and the time-averaged velocity profile quite accurately. The major limitation of this configuration is that capturing other metrics such as y^+ near the boundaries is not possible with such spatial resolution. However, this experiment is considered successful given the validation of key assumptions and the accurate reproduction of the velocity profile.

Subsequently, the free-surface flow configuration was investigated, as depicted in Figure 6.16. This involved a two-dimensional horizontal channel of length $L = 8$ [m] with an elliptic obstacle at $x \in [9, 11]$ [m] described by the equation $z_b(x) = 0.2 - 0.05(x - 10)^2$, an inlet flow rate $Q_0 = 0.18$ [m²/s], and an outlet water height $h_{\text{outlet}} = 0.33$ [m]. In this configuration, a hydraulic jump was expected to occur in the vicinity of the obstacle. An analytical solution for the water height, described by the Bernoulli relationship in Equation (6.20), was obtained using a 1D model. The main assumptions—steady state, incompressibility, and uniform velocity profile—were successfully validated. Hydrostatic pressure was confirmed when spatially averaged across the entire domain (Figure 6.24), although local pressure results showed greater uncertainty, as emphasized in Figure 6.25.

Nevertheless, the software correctly captured the hydraulic jump and accurately reproduced the 1D model in both upstream and downstream regions, with slight differences near the hydraulic jump itself, as presented in Figure 6.26. These discrepancies were expected since the reference model does not account for vorticity and recirculation effects. Additionally, Figure 6.27 demonstrated the capability of the software to reproduce the expected Froude number distribution along the domain, and Figure 6.29 showed the correct head loss due to the hydraulic jump.

Based on these two preliminary tests, one can confidently assess that the software is capable of reproducing the desired hydraulic phenomena.

9.1.4 Final Validation Test

Chapter 7 focused on the ultimate objective: a 3D free-surface turbulent configuration with dynamic rigid bodies.

The configuration, presented in Figure 7.1, consisted of uniform flow in a horizontal channel with a bridge supported by three piers, with seven rigid cylinders of dimensions $(L, D) = (0.4, 0.035)$ [m] placed upstream to potentially create obstacle formations at the bridge entrance. Two distinct scenarios were investigated: a pressurized bridge case and a free surface flow case.

Pressurized Bridge Configuration

The desired obstacle formation was correctly captured, as illustrated in Figure 7.3b. However, the importance of choosing an appropriate ratio between particle size and dynamic rigid body dimensions about at least one-fifth of the rigid body characteristic dimension is required. Otherwise, instabilities such as those shown in Figure 7.4b may occur.

The assumption of a pressurized bridge was validated by analysing the water height distribution between the two bridge spans, as depicted in Figure 7.5. The influence of the wood logs on the fluid was captured through streamlines analysis, with Figure 7.6 showing significant recirculation in both upstream and downstream regions.

The head distribution between spans was evaluated and discussed in Figure 7.7. The two expected head losses were correctly captured and attributed to sudden contraction and expansion effects, although some pressure instabilities under the bridge deck were observed due to inherent characteristics of the SPH solver.

Free Surface Flow Configuration

In contrast to the pressurized case, the free surface configuration demonstrated highly different debris behaviour. As shown in Figure 7.9b, most wood logs passed through the bridge rather than accumulating at the entrance, due to the larger available cross-sectional area without the deck constraint. The streamlines analysis in Figure 7.10 revealed that the main flow split into two preferred paths through the outer regions of the bridge openings, creating a fundamentally different flow pattern compared to the pressurized case.

The free surface behaviour was successfully captured, with Figure 7.11 confirming the expected uniform water height distribution away from the wood log influence zones. The absence of hydraulic jumps was verified through Froude number analysis in Figure 7.12, which remained below unity throughout the domain. The total head evaluation, presented in Figure 7.13, correctly demonstrated the expected head loss pattern due to the sudden contraction created by the accumulated wood logs at the bridge entrance in free surface configuration.

9.2 Main Conclusions About the SPLisHSPlasH Software

From all considerations presented in this master thesis, it should be clear that this software is perfectly capable of simulating 3D turbulent free-surface flows with confidence. Given the increasing necessity to simulate such flow types and the continuous improvement of this software, I strongly believe that SPLisHSPlasH should be used more systematically in hydraulic engineering applications in the coming years.

9.3 Future Perspectives

Obviously, the physical influences of drag, vorticity, and viscosity have been somewhat roughly tuned for each configuration using a trial-and-error methodology that, while occasionally fun, is not systematic. A more rigorous analysis of these contributions should be undertaken. For example, in physical cases where only vorticity is involved, examining how the flow evolves while modifying the associated parameters might reveal relationships between these parameters and particle radius.

Another promising direction would be to investigate different object types such as parallelepiped or spherical rigid bodies with varying sizes to observe how interactions between objects of different scales behave in fluid simulations.

From another perspective, the PBD software allows simulation of deformable objects and even hair-like structures. It would be very interesting to investigate obstacle formation with 'thin' pieces of wood that could bend, to determine whether the software can reproduce this behaviour. Ultimately, incorporating vegetation through "hair-like" objects near flow boundaries could provide surprisingly interesting insights into how the prescribed flow would be modified by such interactions.

A.1 DFSPH implementation details

A.1.1 Formulation of Pressure Calculation

The DFSPH algorithm evaluates pressure values by solving a linear system:

$$\sum_j a_{ij} \frac{p_j}{\rho_j^2} = s_i \quad (\text{A.1})$$

where a_{ij} represents the interaction coefficients between particles, p_j denotes the pressure at particle j , and s_i is the source term based on the specific incompressibility constraint.

The elements of the coefficient matrix a_{ij} describe pressure propagation through the fluid. The diagonal element a_{ii} , representing the self-influence, is particularly important:

$$a_{ii} = -\frac{1}{\rho_i^2} \left(\sum_j V_j \|\nabla W_{ij}\|^2 + \left\| \sum_j V_j \nabla W_{ij} \right\|^2 \right) \quad (\text{A.2})$$

This diagonal element is used to compute the relaxation factor:

$$\alpha_i = \frac{1}{\sum_j V_j \|\nabla W_{ij}\|^2 + \left\| \sum_j V_j \nabla W_{ij} \right\|^2} = -\frac{1}{a_{ii} \rho_i^2} \quad (\text{A.3})$$

where V_j is the volume of particle j and ∇W_{ij} is the kernel gradient.

The source term s_i differs between the two constraints. For the divergence-free constraint, this term reads:

$$s_i = -\frac{D\rho_i}{Dt} = -\rho_i \nabla \cdot \mathbf{v}_i = -\rho_i \sum_j V_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij} \quad (\text{A.4})$$

For the constant density constraint, it takes a different form:

$$s_i = 1 - \frac{\rho_i^{adv}}{\rho_0} \quad (\text{A.5})$$

where ρ_i^{adv} is the advected density (i.e., the density influenced solely by advection):

$$\rho_i^{adv} = \frac{\rho_i}{\rho_0} + \Delta t \sum_j V_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij} \quad (\text{A.6})$$

A.1.2 Numerical Implementation

Rather than explicitly assembling and solving the matrix system a_{ij} , DFSPH uses a matrix-free implementation that directly calculates $\sum_j a_{ij} \frac{p_j}{\rho_j^2}$ based on physical interpretations. The key insight is that particles influence each other through pressure forces, leading to pressure accelerations:

$$\mathbf{a}_i^p = - \sum_j V_j \left(\frac{p_i}{\rho_i^2} + \frac{\rho_{0j} p_j}{\rho_0 \rho_j^2} \right) \nabla W_{ij} \quad (\text{A.7})$$

where ρ_{0j} is the reference density of the neighbouring particle's phase.

The matrix-vector product is then evaluated through the density change caused by these pressure accelerations:

For the **constant density solver**:

$$\sum_j a_{ij} \frac{p_j}{\rho_j^2} = h^2 \sum_j V_j (\mathbf{a}_i^p - \mathbf{a}_j^p) \cdot \nabla W_{ij} \quad (\text{A.8})$$

For the **divergence-free solver**:

$$\sum_j a_{ij} \frac{p_j}{\rho_j^2} = h \sum_j V_j (\mathbf{a}_i^p - \mathbf{a}_j^p) \cdot \nabla W_{ij} \quad (\text{A.9})$$

This approach significantly reduces memory requirements for large-scale simulations.

Jacobi Iteration

Both solvers use a modified Jacobi iteration with under-relaxation to improve stability:

$$\frac{p_i^{k+1}}{\rho_i^2} = \max \left(\frac{p_i^k}{\rho_i^2} - 0.5 \cdot \alpha_i \cdot \left(s_i - \sum_j a_{ij} \frac{p_j^k}{\rho_j^2} \right), 0 \right) \quad (\text{A.10})$$

The factor of 0.5 improves stability, and the maximum operation enforces non-negative pressure values. The relaxation factor α_i is precomputed for each particle during initialization and represents the inverse of the diagonal matrix element scaled appropriately.

Also, DFSPH employs a warm start strategy by initializing pressure values from previous time steps:

For the **constant density solver**:

$$\frac{p_i}{\rho_i^2} = 0.5 \cdot \min \left(\frac{p_i^{prev}}{\rho_i^2}, c \right) \cdot \frac{1}{h^2} \quad (\text{A.11})$$

For the **divergence-free solver**:

$$\frac{p_i}{\rho_i^2} = 0.5 \cdot \min \left(\frac{p_i^{prev}}{\rho_i^2}, c \right) \cdot \frac{1}{h} \quad (\text{A.12})$$

where c is a small constant and h is the time step size. The scaling factors account for the time step size dependency in each solver.

A.2 GHE velocity profile demonstration

Laminar contribution

The laminar contribution is well known and is a simple parabolic profile. Indeed, an analytical solution is obtained uniquely from the momentum equation.

In this particular case, one may use an additional assumption:

- The flow is only one-dimensional (by symmetry) : $v = 0 \Leftrightarrow \mathbf{u} = u\mathbf{e}_x$

such that the Equations (6.7) and (6.8) (projected on the \mathbf{e}_x axis) read:

$$\frac{\partial u}{\partial x} = \tau \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right) \quad (\text{A.13})$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (\text{A.14})$$

Nevertheless, the non-linear terms from Equation (A.14) disappear thanks to the fully-developed ($\partial_x u = 0$) and one directional flow assumption ($v = 0$). Hence, this latter Equation becomes:

$$\frac{\partial^2 u(y)}{\partial y^2} = \frac{d^2 u}{dy^2} = Re \frac{\partial p}{\partial x} \Leftrightarrow u(y) = \frac{Re \frac{\partial p}{\partial x}}{2} y^2 + C_1 y + C_2 \quad (\text{A.15})$$

where the boundary conditions considered are the **non-slip wall condition**, namely $\mathbf{u}|_{y=\{0,L\}} = \frac{d\mathbf{u}}{dy}|_{y=\{0,L\}} = \mathbf{0}$, which gives the final laminar solution:

$$\mathbf{u}(y)^{\text{laminar}} = 4U_0 y(L - y)/L^2 \quad (\text{A.16})$$

where $U_0 = \frac{1}{2} Re \frac{\partial p}{\partial x}$

It is important to note in this specific case that the continuity equation has not been used in the obtention of the velocity profile. Which will be the case for the turbulent contribution.

A.2.1 Turbulent contribution

The turbulent profile is a bit harder to get. Indeed the previous assumption of no vertical velocity v is not valid any more and has thus to be taken into account. Hence, the momentum equation has to be considered in both e_x and e_y axes such that one adds to the Equations (A.13) and (A.14) another one:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \tau \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right) \quad (\text{A.17})$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (\text{A.18})$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (\text{A.19})$$

Recalling the fully-developed flow in the x direction and the horizontal pressure gradient to be constant, this set of equation becomes:

$$\frac{\partial v}{\partial y} = \tau \frac{\partial^2 p}{\partial y^2} \quad (\text{A.20})$$

$$v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \frac{\partial^2 u}{\partial y^2} \quad (\text{A.21})$$

$$\underbrace{v \frac{\partial v}{\partial y}}_{\approx 0} = -\frac{\partial p}{\partial y} + \frac{1}{Re} \frac{\partial^2 v}{\partial y^2} \quad (\text{A.22})$$

where the non-linear term $v \frac{\partial v}{\partial y}$ is assumed to be small and thus neglected. Otherwise an analytical solution can not be obtained.

By derivating Equation (A.22) w.r.t. y and injecting the result in Equation (A.20), one obtains a third order ode for v :

$$\frac{\tau}{Re} \frac{\partial^3 v}{\partial y^3} - \frac{\partial v}{\partial y} = 0 \Rightarrow v(y) = A + B e^{\pm \sqrt{\frac{Re}{\tau}} y} \quad (\text{A.23})$$

And finally applying the boundary conditions $v(y=0) = 0$, it yields:

$$v(y) = A \left(1 - e^{\pm \sqrt{\frac{Re}{\tau}} y} \right) \quad (\text{A.24})$$

Also, recalling the two definition of the Reynolds number Re and timescale τ :

$$\frac{Re}{\tau} = \frac{U_0 L}{\frac{\nu}{\tau^* U_0}} = \frac{L^2}{\tau^* \nu} \equiv \delta^{-2} \Leftrightarrow \delta = \frac{\sqrt{\tau^* \nu}}{L} = \sqrt{\frac{\tau}{Re}} \quad (\text{A.25})$$

$$\Rightarrow v(y) = A \left(1 - e^{\pm \frac{\delta}{y}} \right) \quad (\text{A.26})$$

At this point the two sign in the exponential should be considered. Nevertheless, Alex had already pointed out that the negative sign solution leads to an unbounded solution and is thus not acceptable. Hence, one only focuses on the positive sign solution. Injecting Equation (A.26) into Equation (A.21) leads to:

$$A \left(1 - e^{\frac{\delta}{y}} \right) \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \frac{\partial^2 u}{\partial y^2} \quad (\text{A.27})$$

$$\frac{1}{Re} \frac{\partial^2 u}{\partial y^2} - A \left(1 - e^{\frac{\delta}{y}} \right) \frac{\partial u}{\partial y} = \frac{\partial p}{\partial x} \quad (\text{A.28})$$

Given that this differential equation is linear in u , one may obtain a homogeneous and particular solution. For the homogeneous solution (e.g. $\partial p/\partial x = 0$):

$$\frac{1}{Re} \frac{\partial^2 u}{\partial y^2} - A \left(1 - e^{\frac{\delta}{y}}\right) \frac{\partial u}{\partial y} = 0 \quad (\text{A.29})$$

$$\frac{\partial^2 u/\partial y^2}{\partial u/\partial y} = ReA \left(1 - e^{\frac{\delta}{y}}\right) \quad (\text{A.30})$$

Reminding the property of the logarithm derivative:

$$\frac{d \ln f(x)}{dx} = \frac{1}{f(x)} \frac{df(x)}{dx} \quad (\text{A.31})$$

The Equation (A.30) is rewritten as:

$$\frac{d(\ln(\partial u/\partial y))}{dy} = ReA \left(1 - e^{\frac{\delta}{y}}\right) \quad (\text{A.32})$$

$$\ln(\partial u/\partial y) = ReA \left(y - \delta e^{\frac{\delta}{y}}\right) + c_1 \quad (\text{A.33})$$

$$\frac{\partial u}{\partial y} = e^{\left[ReA \left(y - \delta e^{\frac{\delta}{y}}\right)\right]} \underbrace{e^{c_1}}_{\equiv C_1} \quad (\text{A.34})$$

Then, by defining the change of variable $t = e^{y/\delta} \Rightarrow dy = \delta \frac{dt}{t}$, the Equation (A.34) reads:

$$u(y) = \int C_1 e^{\left[ReA\delta \left(\frac{y}{\delta} - e^{\frac{y}{\delta}}\right)\right]} dy = \int C_1 \delta \frac{e^{[ReA\delta(\ln t - t)]}}{t} dt \quad (\text{A.35})$$

Imposing the arbitrary constant $A = \frac{1}{Re\delta}$ gives us a solution:

$$u(y) = \int C_1 \delta \frac{e^{[(\ln t - t)]}}{t} dt = C_1 \delta \int \frac{te^{-t}}{t} dt = -C_1 \delta e^{-t} + c_2 = -C_1 \delta e^{-e^{\frac{y}{\delta}}} + c_2 \quad (\text{A.36})$$

Finally, the integration constants are obtained through the two boundary conditions $u(y = 0) = 0$ and $u(y \rightarrow +\infty) = U_0$ lead to:

$$C_1 = \frac{V_0 e}{\delta} \quad \& \quad C_2 = U_0$$

which gives:

$$u(y)^{\text{turbulent}} = U_0 \left(1 - e^{\left(1 - e^{\frac{y}{\delta}}\right)}\right) \quad (\text{A.37})$$

A.3 Explanation about Levenberg-Marquardt algorithm

The Levenberg-Marquardt algorithm is a numerical optimization method for solving nonlinear least squares problems that combines both gradient descent and Gauss-Newton methods.

For a set of observed data points (x_i, y_i) and a parametric model function $f(x|\mathbf{p})$. In the GHE model the vector $\mathbf{p} = (\delta, \gamma)$ and the data points $(x_i, y_i) = (y_i^{\text{sph}}, u_i^{\text{sph}})$. One aims to find the parameter vector \mathbf{p} that minimizes the following quantity:

$$S(\mathbf{p}) = \sum_{i=1}^m [y_i - f(x_i|\mathbf{p})]^2 \quad (\text{A.38})$$

Starting with an initial parameter estimate \mathbf{p}_0 , the algorithm iteratively computes updates:

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{q}_k \quad (\text{A.39})$$

The update vector \mathbf{q}_k is determined by solving the damped normal equations:

$$\left[\mathbf{J}^T \mathbf{J} + \lambda \cdot \text{diag}(\mathbf{J}^T \mathbf{J}) \right] \mathbf{q} = \mathbf{J}^T (y - f(\mathbf{p})) \quad (\text{A.40})$$

where:

- J is the Jacobian matrix with elements $J_{ij} = \frac{\partial f_i}{\partial p_j}$
- λ is the damping parameter (non negative scalar)

The damping parameter λ is adjusted at each iteration:

$$\text{If } S(p_k + q_k) < S(p_k) : \lambda_{k+1} = \frac{\lambda_k}{\nu}, \text{ accept update} \quad (\text{A.41})$$

$$\text{If } S(p_k + q_k) \geq S(p_k) : \lambda_{k+1} = \nu \cdot \lambda_k, \text{ reject update} \quad (\text{A.42})$$

where $\nu > 1$.

It is worth to mention that if $\lambda \rightarrow 0$, the algorithm approaches the Gauss-Newton method and if $\lambda \rightarrow \infty$ the algorithm approaches gradient descent with small steps.

A.4 Derivation of the 1D model solution for the water height computation

Reminding the initial Bernoulli equation:

$$\frac{\partial}{\partial x} \left(u(x)q + g \frac{h(x)^2}{2} \right) = gh(x) \frac{\partial z_b(x)}{\partial x} \quad (\text{A.43})$$

One keeps in mind that the uniform velocity profile assumption is obviously not truly verified in the vicinity of the obstacle. But this assumption is mandatory such that one may write:

$$q = \int_{h(x)} u(x) dy = u(x)h(x)$$

and Equation (A.43) is transformed as:

$$\frac{\partial}{\partial x} \left(u(x)^2 h(x) + g \frac{h(x)^2}{2} \right) = gh \frac{\partial z_b(x)}{\partial x} \quad (\text{A.44})$$

Next, one apply the x derivative on the L.H.S:

$$2u(x) \frac{\partial u(x)}{\partial x} h(x) + u(x)^2 \frac{\partial h(x)}{\partial x} + \frac{g}{2} 2h(x) \frac{\partial h(x)}{\partial x} \quad (\text{A.45})$$

The term $\frac{\partial u(x)}{\partial x}h(x)$ is modified using the incompressible fluid assumption:

$$\begin{aligned} q &= u(x)h(x) = \text{cst} \\ \Rightarrow \frac{\partial q}{\partial x} &= \frac{\partial [u(x)h(x)]}{\partial x} = \frac{\partial u(x)}{\partial x}h(x) + \frac{\partial h(x)}{\partial x}u(x) = 0 \\ \Rightarrow \frac{\partial u(x)}{\partial x}h(x) &= -\frac{\partial h(x)}{\partial x}u(x) \end{aligned}$$

Then, injecting this result in Equation (A.45) yields:

$$\begin{aligned} & -2u(x)^2 \frac{\partial h(x)}{\partial x} + u(x)^2 \frac{\partial h(x)}{\partial x} + gh(x) \frac{\partial h(x)}{\partial x} \\ &= -u(x)^2 \frac{\partial h(x)}{\partial x} + gh(x) \frac{\partial h(x)}{\partial x} \\ &= [-u(x)^2 + gh(x)] \frac{\partial h(x)}{\partial x} \end{aligned}$$

Injecting this latter in Equation (A.43) and dividing by $gh(x)$ leads to:

$$\frac{1}{gh(x)} [-u(x)^2 + gh(x)] \frac{\partial h(x)}{\partial x} = \frac{gh(x)}{gh(x)} \frac{\partial z_b(x)}{\partial x} \quad (\text{A.46})$$

$$\Rightarrow \left[-\frac{u(x)^2}{gh(x)} + 1 \right] \frac{\partial h(x)}{\partial x} = \frac{\partial z_b(x)}{\partial x} \quad (\text{A.47})$$

Finally, one involves the Froude number definition $Fr = \frac{u}{\sqrt{gh}}$ to the desired relationship. Also, one drops the partial derivative notation:

$$\left[1 - Fr(x)^2 \right] \frac{dh(x)}{dx} = \frac{dz_b(x)}{dx} \quad (\text{A.48})$$

Technical considerations

For the interested reader, here are all the relevant information concerning both the software and hardware specifications.

Hardware components

Component	Specification
Laptop	ASUS ROG Zephyrus G16 (GU605MI)
RAM	32 GB
CPU	Intel Core Ultra 7 155H @ 3.80 GHz (22 logical processors)
GPU 1	NVIDIA GeForce RTX 4070 Laptop GPU
GPU 2	Intel Arc Graphics
OS	Windows 11

Softwares utilised

Software	Version/Details
Build System	CMake 3.31.7
Programming Language	C++ with Python 3.12.4
GPU Computing	NVIDIA CUDA Toolkit 12.8
ParaView version	ParaView 5.13.3
SPH Framework	Modified SPlisHSPlasH

Data location



Access to all data :

[https://github.com/Rivlow/
Master-thesis-free-surface-turbulent-flow-SPH](https://github.com/Rivlow/Master-thesis-free-surface-turbulent-flow-SPH)

This repository contains all the data, scripts and results mentioned in this submission.

Bibliography

- [1] R. B. Haehnel and S. F. Daly, “Maximum impact force of woody debris on floodplain structures,” *Journal of Hydraulic Engineering*, vol. 130, no. 2, pp. 112–120, 2004.
- [2] D. Bocchiola, M. C. Rulli, and R. Rosso, “Flume experiments on wood entrainment in rivers,” *Advances in Water Resources*, vol. 29, no. 8, pp. 1182–1195, 2006.
- [3] E. Bresciani, P. Davy, and J.-R. de Dreuzy, “A finite volume approach with local adaptation scheme for the simulation of free surface flow in porous media,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 36, no. 13, pp. 1574–1591, 2012.
- [4] M. J. Kazemzadeh-Parsi, “Numerical flow simulation in gated hydraulic structures using smoothed fixed grid finite element method,” *Applied Mathematics and Computation*, vol. 246, pp. 447–459, 2014.
- [5] R. A. Gingold and J. J. Monaghan, “Smoothed particle hydrodynamics: theory and application to non-spherical stars,” *Monthly Notices of the Royal Astronomical Society*, vol. 181, no. 3, pp. 375–389, 1977.
- [6] L. Goffin, “Development of a didactic sph model,” Master’s thesis, University of Liège, June 2013. Master’s Thesis.
- [7] M. Lodomez, S. Erpicum, B. Dewals, M. Piroton, and P. Archambeau, “Comparison between experimental and sph models over a sharp-crested weir,” in *5th IAHR International Junior Researcher and Engineer Workshop on Hydraulic Structures*, 2014.
- [8] S. Adami, X. Hu, and N. Adams, “Simulating three-dimensional turbulence with sph,” tech. rep., Citeseer, 2012. Technical Report.
- [9] M. R. Hashemi, R. Fatehi, and M. T. Manzari, “Sph simulation of interacting solid bodies suspended in a shear flow of an oldroyd-b fluid,” *Journal of Non-Newtonian Fluid Mechanics*, vol. 166, no. 21-22, pp. 1239–1252, 2011.
- [10] J. Bender *et al.*, “SPliHSPlasH Library,” 2024. MIT License.
- [11] V. Springel, “Smoothed particle hydrodynamics in astrophysics,” *Annual Review of Astronomy and Astrophysics*, vol. 48, pp. 391–430, 2010.
- [12] D. Koschier, J. Bender, B. Solenthaler, and M. Teschner, “Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids,” *arXiv preprint arXiv:2009.06944*, 2020.

- [13] I. S. Starodubtsev, Y. V. Starodubtseva, I. A. Tsepelev, and A. T. Ismail-Zadeh, “Three-dimensional numerical modeling of lava dynamics using the smoothed particle hydrodynamics method,” *Journal of Volcanology and Seismology*, vol. 17, pp. 1–15, 2023.
- [14] M. Röhler, V. Kumar, C. Richter, and G. Reinhart, “A sph simulation approach using the carreau model for the free surface flow of adhesives,” in *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2018.
- [15] H. R. Abbasia and R. Lubbad, “A numerical model for the simulation of oil–ice interaction,” *Physics of Fluids*, vol. 33, 2021.
- [16] U. Berdica, Y. Fu, Y. Liu, E. Angelidis, and C. Feng, “Mobile 3d printing robot simulation with viscoelastic fluids,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [17] M. Akbay, N. Nobles, V. Zordan, and T. Shinar, “An extended partitioned method for conservative solid-fluid coupling,” *ACM Transactions on Graphics*, vol. 37, no. 4, 2018.
- [18] S. Carensac, N. Pronost, and S. Bouakaz, “Optimizations for predictive–corrective particle-based fluid simulation on gpu,” *The Visual Computer*, vol. 39, pp. 1–14, 2023.
- [19] M. Li, H. Li, W. Meng, J. Zhu, and G. Zhang, “An efficient non-iterative smoothed particle hydrodynamics fluid simulation method with variable smoothing length,” *Visual Computing for Industry, Biomedicine, and Art*, vol. 6, no. 1, 2023.
- [20] F. Zorilla, M. Ritter, J. Sappl, W. Rauch, and M. Harders, “Accelerating surface tension calculation in sph via particle classification and monte carlo integration,” in *Computer Graphics and Visual Computing (CGVC)*, 2019.
- [21] M. Ihmsen, N. Akinici, M. Becker, and M. Teschner, “A parallel sph implementation on multi-core cpus,” *Computer Graphics Forum*, vol. 30, no. 1, pp. 99–112, 2011.
- [22] M. Teschner, B. Heidelberger, M. Müller, D. Pomeranets, and M. H. Gross, “Optimized spatial hashing for collision detection of deformable objects,” in *Vision, Modeling, and Visualization*, pp. 47–54, 2003.
- [23] M. Weiler, D. Koschier, M. Brand, and J. Bender, “A physically consistent implicit viscosity solver for sph fluids,” *Computer Graphics Forum*, vol. 37, no. 2, pp. 145–155, 2018.
- [24] J. Bender and D. Koschier, “Divergence-free smoothed particle hydrodynamics,” in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 147–155, 2015.
- [25] J. Bender, D. Koschier, T. Kugelstadt, and M. Weiler, “A micropolar material model for turbulent sph fluids,” in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 1–8, 2017.
- [26] C. Gissler, S. Band, A. Peer, M. Ihmsen, and M. Teschner, “Approximate air-fluid interactions for sph,” in *VRIPHYS*, pp. 29–38, 2017.
- [27] P. J. O’Rourke and A. A. Amsden, “The tab method for numerical calculation of spray droplet breakup,” tech. rep., SAE Technical Paper, 1987.
- [28] A. B. Liu, D. Mather, and R. D. Reitz, “Modeling the effects of drop drag and breakup on fuel sprays,” *SAE Transactions*, pp. 83–95, 1993.
- [29] J. Bender, T. Kugelstadt, M. Weiler, and D. Koschier, “Volume maps: An implicit boundary representation for sph,” in *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pp. 1–10, 2019.
- [30] J. J. Monaghan, “Smoothed particle hydrodynamics,” *Annual Review of Astronomy and Astrophysics*, vol. 30, pp. 543–574, 1992.

- [31] J. J. Monaghan, “Sph compressible turbulence,” *Monthly Notices of the Royal Astronomical Society*, vol. 335, no. 3, pp. 843–852, 2002.
- [32] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner, “Versatile rigid-fluid coupling for incompressible sph,” *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–8, 2012.
- [33] J. Bender, M. Müller, M. A. Otaduy, M. Teschner, and M. Macklin, “A survey on position-based simulation methods in computer graphics,” *Computer Graphics Forum*, vol. 33, no. 6, pp. 228–251, 2014.
- [34] D. Violeau and R. Issa, “Numerical modelling of complex turbulent free-surface flows with the sph method: an overview,” *International Journal for Numerical Methods in Fluids*, vol. 53, no. 2, pp. 277–304, 2007.
- [35] X. Y. Hu and N. A. Adams, “A sph model for incompressible turbulence,” *Procedia IUTAM*, vol. 18, pp. 66–75, 2015.
- [36] G. Pahar and A. Dhar, “Modeling free-surface flow in porous media with modified incompressible sph,” *Engineering Analysis with Boundary Elements*, vol. 68, pp. 75–85, 2016.
- [37] S. Shao, “Incompressible sph simulation of wave breaking and overtopping with turbulence modelling,” *International Journal for Numerical Methods in Fluids*, vol. 50, no. 5, pp. 597–621, 2006.
- [38] A. Fedoseyev, “Analytical solution for turbulent flow in channel,” *arXiv preprint arXiv:2306.04038*, 2023.
- [39] A. I. Fedoseyev and B. V. Alexeev, “Generalized hydrodynamic equations for viscous flows-simulation versus experimental data,” in *AIP Conference Proceedings*, vol. 1487, pp. 241–247, AIP, 2012.
- [40] P. Archambeau, *Contribution to the modelling of flood genesis and propagation*. PhD thesis, University of Liège, 2006. Ph.D. Dissertation.
- [41] F. Dütz, “Effect on flow conditions of floating debris accumulation at bridges,” Master’s thesis, University of Liège, 2023.