Politecnico di Torino

Dipartimento di ingegneria Civile Ambientale

Corso di Laurea Magistrale in Ingegneria Civile Idraulica

Tesi di Laurea Magistrale

SPH modeling for Dam-Break waves of irrigation tanks

Relatore

prof. Davide Poggi

Corelatore

prof. Edoardo Daly

Candidato

Marco De Castro

Luglio 2019

"Alla mia famiglia"

Abstract

The present thesis focuses on the use of the Smoothed Particle Hydrodynamics (SPH) method to model the propagation of a dam-break wave and its interaction with the area downstream of the dam body. In particular, the Explicit Incompressible SPH (EISPH) scheme was used; this is a variation of the SPH algorithm, where pressure fluctuations and computational effort are largely reduced. This algorithm has been used and validated in two-dimensional simulations, but it is somewhat unexplored in three-dimensional. For this purpose, as the first test, the confrontation with a simulation of a dam-break wave carried out with the two-dimensional EISPH scheme and one carried out by Shao & Lo is performed, in order to verify the correct functioning of the algorithm. Then a comparison with a simple analytical solution of a Dam-Break wave, given by Ritter's solution, was useful to understand the correctness of the wave profile and of the order of magnitude of the wavefront celerity. Completed these two tests, helpful for the validation of the algorithm and code, experiments from the published literature were simulated specifically, the case of a solid parallelepiped placed downstream of a breach in the dam body and hit by the dam-break wave caused by the sudden opening of this breach was simulated. The difference in shape between the wave in the physical and in the numerical model was investigated, paying attention to the variation of water depth between them, in time and space, using the transformation function between grey tones and water depth derived by the authors, realizing a punctual comparison with the EISPH simulation at the same time steps. In order to increase the resolution of the numerical model, which means increase the number of particles present in the domain decreasing their radius, without increasing the computational cost too much, the symmetry of the problem was exploited, realizing only half the domain, thus having half of the necessary particles and a contained computational effort, evaluating again the difference in water depth distribution between the two models.

The thesis showed that EISPH might be a useful method to simulate complex hydrodynamic problems.

Contents

Abstract **i**

List of Figures **iv**

List of Tables **iv**

1 introduction **1**

2 Literature Review **5**
2.1 Analytical studies 5
2.1.1 Ritter's Analytical Solution 5
2.2 Case Studies 8
2.2.1 Dam-Break Flows: Acquisition of Experimental Data through an Imaging Technique and 2D Numerical Modeling [F. Aureli, A. Maranzoni, P. Mignosa, C. Ziveri] 8
2.2.2 Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface [S. Shao, E. Lo] 16
2.2.3 Comparison between WCSPH, δ-SPH and EISPH 18

3 Methods **25**
3.1 Equations 25
3.1.1 Constitutive Equation 25
3.1.2 Continuity Equation 26
3.1.3 Momentum Equation 26
3.2 SPH approximation 27
3.2.1 Essential formulations 27
3.3 EISPH 32
3.3.1 Numerical schemes 32
3.3.2 Implementation 35

4 Applications **45**
4.1 2D Dam-break 45
4.1.1 Comparison with a 2D EISPH model and Shao & Lo Results 45
4.1.2 Comparison with Ritter's analytical solution 51
4.2 3D Dam-Break: Comparison with F. Aureli, A. Maranzoni, P. Mignosa, and C. Ziveri experiment[1] 57
4.2.1 Full Domain: 144.000 real particles 57
4.2.2 Half Domain: 576.000 real particles 68

5 Conclusions and future perspective **85**

List of Figures

2.1 Ritter's original sketch . . . 7
2.2 Facility . . . 8
2.3 Facility scheme . . . 9
2.4 Transformation function grey scale - depth . . . 10
2.5 Test 1 . . . 12
2.6 Test 4 . . . 13
2.7 Frequency distribution . . . 14
2.8 Water level series . . . 15
2.9 ISPH patterns . . . 17
2.10 ISPH leading edge evolution . . . 17
2.11 ISPH isobar . . . 18
2.12 Comparsion hydrostatic . . . 20
2.13 Comparison leading edge . . . 21
2.14 Comparison dam-break impact . . . 22
2.15 Computational time . . . 23
2.16 Wet bed . . . 24
3.1 Cubic Spline kernel function . . . 30
3.2 Support domain . . . 30
3.3 Boundary particles . . . 37
3.4 No slip condition . . . 38
3.5 Free slip condition . . . 38
3.6 All-pair search . . . 42
3.7 Tree search algorithm . . . 42
3.8 Linked-list search algorithm . . . 43
4.1 Selected domain . . . 46
4.2 Particle configurations after collapse of dam at different times. Red dots correspond to Shao & Lo computations while purple dots correspond to the 2D-EISPH. . . . 47
4.3 Wavefront comparison . . . 48
4.4 Pressure distribution comparison after collapse of dam at time $t = 0,05$s for 3D-EISPH (above) and Shao & Lo (below) simulations. . . . 49
4.5 Pressure distribution comparison after collapse of dam at time $t = 0,05$s for 3D-EISPH (above) and Shao & Lo (below) simulations. . . . 50
4.6 initial distribution . . . 51
4.7 Wavefront celerity comparison . . . 53
4.8 Wavefront position comparison . . . 53
4.9 Wave profile comparison . . . 54
4.10 Wave profile evolution . . . 55
4.11 Water depth evolution . . . 56
4.12 Full domain initial configuration . . . 59
4.13 Full domain comparison . . . 60
4.14 Full domain overlapped snapshot . . . 61
4.15 Water level time series full domain simulation . . . 62
4.16 Grid cell main dimensions . . . 65
4.17 Interpolation grid - Full domain . . . 68

4.18 Frequency distribution 69
4.19 Half domain front view representation 71
4.20 Full domain (left) and half domain (right) up view representation at time $t = 0,05$ s. Real (color proportional to v_y), virtual (black) and ghost (grey) particles are represented. The colorbar indicates the transversal velocity v_y. 71
4.21 Full domain (left) and half domain (right) up view representation at time $t = 0,15$ s. Real (color proportional to v_y) and virtual (black) and ghost (grey) particles are represented. The colorbar indicates the transversal velocity v_y. 72
4.22 Half domain initial configuration 73
4.23 Half domain comparison 74
4.24 Half domain overlapped snapshots 75
4.25 Full and half domain comparison 77
4.26 Water level time series half domain simulation 78
4.27 Interpolation grid - 576.000 particle simulation 79
4.28 Frequency distribution 82

List of Tables

1 Variance of the results obtained repeating several time the process. 64
2 Main statistics value of the deviation $h_{SPH} - h_{exp}$ 64
3 Absolute, Relative and Cumulative frequency distribution for every $h_{SPH} - h_{exp}$ classes 65
4 Main statistics value of the deviation $h_{SPH} - h_{exp}$ 79
5 Absolute, Relative and Cumulative frequency distribution for every $h_{SPH} - h_{exp}$ classes 79

1 introduction

The dam-break is a phenomenon that occurs when there is a sudden collapse of a part of the dam body, with consequent release of water and formation of a flood wave that, spreading downstream, can lead to disastrous effects in terms of economic damages and loss of life.

Most dam-break problems are due to the existence of many farm dams (there are at least 735,000 farm dams only in Australia) all around the globe[2]. This type of dams is accompanied by poor, or inexistent, planning and bad management, leading to high potential failure with possible severe consequences.

For example, in China the Shimantan and Banquia dams failed in 1975 because of the cumulative failure of 60 smaller upstream dams, resulting in the death of 230,000. In Italy, the Stava dam near Trento failed in 1985 and, while releasing only 180 ML of tailings material, it killed 268 people and caused serious environmental damages. In the United States, the Kelly Barnes Lake dam, only 8 m high, failed in 1977 killing 39 people; the 8 m Lake Lawn dam in Colorado, which stored only 830ML, failed in 1982 killing 3 people and causing USD 31 million in damage despite warnings and evacuation; the 5 m Evans and Lockwood dams, which held only 89 ML and 39 ML of water respectively, both collapsed in a cascade manner in 1989, killing 2 people; and the 13 m Kaloko farm dam in Hawaii overtopped and failed due to a blocked spillway in 2006 resulting in the deaths of 7 people and widespread environmental damage. In Indonesia the Situ Gintung earthen dam, only 10 m high failed by overtopping in 2009 killing around 100 people and causing widespread damage in Jakarta [2].

It is therefore possible to understand how important the study of this phenomenon is, with particular attention to flood wave propagation and how it interacts with the area downstream of the dam. In fact, there are a lot of studies about the dam-break phenomenon, from simplified analytic solutions, like Ritter's dam break wave, to numerous experiments of simple dam-break scenarios or with the presence of obstacles downstream[1].

During the years, many numerical methods have been developed to study and simulate the Dam-break phenomenon. Finite elements methods, like *Marker And Cell* (MAC) or *Volume Of Fluid* (VOF), are mostly used in this kind of problems. However, their need for meshes to solve the Navier-Stokes equations makes them not always suitable for problems involving free surfaces, where the water curvature, near obstacles and edges, becomes very high.

Smoothed particle hydrodynamics (SPH) is a meshfree Lagrangian method introduced independently by Lucy and Gingold and Monaghan[3], to solve gas dynamics problems in astrophysics.

The move from astrophysics problems to terrestrial problems involving incompressible fluids was initiated by a specific interest in archeology and how a tsunami may have destroyed the Minoan civilization in Crete [4].

Since then, given its flexibility in adapting to complex geometry, it became really useful into numerous branches of computational physics, to simulate a large range of physical problems, including shock waves, solitary waves, shallow water and dam-break problems, which is the main argument of the present document.

The main concept behind SPH is to approximate the numerical solutions of physical partial differential equations through the use of a large enough number of fluid particles mimicking a continuum domain. The decomposed computational domain does not need to hold the connectivity among the particles with a computational mesh, from this it follows that the SPH is a mesh-free method. Each particle carries its physical properties such as mass, pressure, and velocity that are obtained using constitutive relations and interpolation with its computational neighboring particles.

The first use of SPH for fluid flow is due to Monaghan[3], who limited fluid compressibility by imposing a large speed of sound to the equation of state. This approach is known as *Weakly Compressible SPH (WCSPH)* and, despite having the advantage of not requiring any treatment of the free surface and a fast computational time, its inaccuracy in simulating pressure caused other methods to be developed. Since then, many modifications to WCSPH have been suggested to reduce these pressure fluctuations. For example, one of the first to do so is the X-SPH (where the X term stands for unknown factor). In this type of model, the position of particles is artificially smoothed, while the density of these is renormalized after a certain number of steps.

Another way to reduce pressure fluctuations is to employ a diffusive term in the continuity equation. The model which does this is called δ-SPH.

In the same period of time, an incompressible scheme was developed. It is called ISPH and it requires the solution of the Poisson equation, which results in an implicit calculation of pressure; implicit schemes like the IISPH are thus more complex to implement than the Weakly Compressible SPH, and the time required to run a simulation is bigger.

A compromise between reducing pressure fluctuation and, at the same time, the computational

effort, with a simplified algorithm, is represented by the Explicit Incompressible Smoothed Particle Hydrodynamics method (EISPH).
Like the IISPH, these schemes use an approximated solution for the Poisson equation, but with less computational time.

The present thesis focuses on the use of the three-dimensional Smoothed Particle Hydrodynamics method to model a complex Dam-Break scenario with the presence of obstacles. In particular, the EISPH scheme is used to simulate the experiment from F. Aureli, A. Maranzoni, P. Mignosa, C. Ziveri [1] where an obstacle is placed in front of a breach in the dam body.

2 Literature Review

2.1 Analytical studies

2.1.1 Ritter's Analytical Solution

The motion of a flood wave can be described through the De Saint-Venant equations. These equations describe the state of an incompressible mass of fluid on a one-dimensional motion, under the hypotheses of hydrostatic distribution of pressures and uniform distribution of velocities in each cross-section.

The first equation is called *Continuity Equation*, that forces the satisfaction of the principle of conservation of mass.

The second equation is the *Dynamic Equation*, which imposes the balance of forces acting on the mass of fluid in motion. These are formulated as

$$\begin{cases} \frac{\partial \Omega}{\partial t} + \frac{\partial (u\Omega)}{\partial x} = 0; \\ \frac{1}{g}\frac{\partial u}{\partial t} + \frac{u}{g}\frac{\partial u}{\partial x} + \frac{\partial h}{\partial x} = i_f - J, \end{cases} \tag{2.1}$$

where:

- x is the abscissa, with $x = 0$ at the dam body;
- t is the time, with $t = 0$ at the dam collapse;
- Ω is the outflow area;
- u is the average velocity on a given section;
- h is the water depth on a given section from the channel bed;
- i_f is the channel bed slope;
- J is the term that represents the load losses per unit of width.
- g is the gravity acceleration.

For a rectangular channel, this equations can be rewritten as a function of the velocity u and the water depth h, as follows.

$$A = bh \tag{2.2}$$

$$\begin{cases} \frac{\partial h}{\partial t} + h\frac{\partial v}{\partial x} + u\frac{\partial h}{\partial x} = 0; \\ \frac{1}{g}\frac{\partial u}{\partial t} + \frac{u}{g}\frac{\partial u}{\partial x} + \frac{\partial h}{\partial x} = i_f - J, \end{cases} \tag{2.3}$$

where b is the channel width.

The easiest solution to the De Saint Venant equations is given by Ritter who considered that the load losses J are equal to the channel bed slope i_f, which is equal to 0 (that means a frictionless condition). Ritter also considers a rectangular channel with a horizontal bottom with smooth bed and walls; the dam runs perpendicular to bed and breaks outright. The initial conditions are given as

$$h(x, t = 0) = \begin{cases} h_0 & if \quad x < 0 \\ 0 & if \quad x > 0 \end{cases} \tag{2.4}$$

$$u(x, t = 0) = 0 \tag{2.5}$$

where h_0 is the initial water depth upstream the dam body, which corresponds the abscissa $x = 0$.

The solution given by Ritter is written as:

$$h = \frac{1}{9g}\left(2\sqrt{gh_0} - \frac{x}{t}\right)^2 \tag{2.6}$$

$$u = \frac{2}{3}\left(\sqrt{gh_0} + \frac{x}{t}\right) \tag{2.7}$$

Defined in the interval

$$-1 < \frac{x}{t\sqrt{gh_0}} < 2 \tag{2.8}$$

Outside of this interval the quantities h and u are not affected by the wave.

The solution for the wave shows the following properties: its profile is parabolic with a tangent to the channel bottom at its front; at the dam site the water height is constant in time with

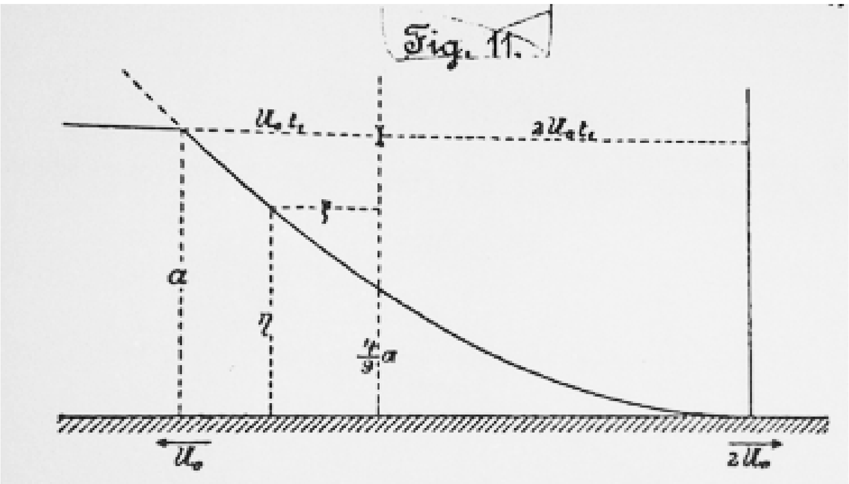


Figure 2.1: Original sketch of the Dam-Break wave analytical solution by Ritter

$h(x = 0, t) = \frac{4}{9}h_0$, so in every moment the wave profile pass through this point, as shown in figure 4.10 further on.

The velocity is linear along the channel with the value $-\sqrt{gh_0}$ at the negative front and $2\sqrt{gh_0}$ at the positive front. Also the velocity is constant at the dam position, where it value is $u(x = 0, t) = \frac{2}{3}\sqrt{gh_0}$.

In figure 2.1 the original parabolic profile sketched by Ritter can be observed.

2.2 Case Studies

In this section, some case studies concerning the Dam-Break phenomenon will be discussed.

2.2.1 Dam-Break Flows: Acquisition of Experimental Data through an Imaging Technique and 2D Numerical Modeling [F. Aureli, A. Maranzoni, P. Mignosa, C. Ziveri]

This experiment from Aureli, Maranzoni, Mignosa and Ziveri [1] consisted of four tests concerning rapidly varying flows induced by the sudden removal of a sluice gate. The aim of those tests was to obtain further experimental data useful for the validation of numerical codes. Since experimental data are usually acquired through a limited number of gauges, which makes it difficult to reconstruct the overall features of the flow field especially in the presence of wetting and drying fronts and moving shocks, the authors applied an imaging technique capable of providing spatially distributed information. They added a coloring agent to the water, backlighted the opalescent bottom of the facility and took photos of the area of interest. These pictures, acquired in grey tones, were calibrated to show the water depth as a function of the grey scale. The experimental data were then compared with the numerical results obtained by a two-dimensional MUSCL-Hancock finite volume numerical model which is based on the shallow water approximations.

The facility for the tests is shown in figure 2.2 with a schematic view in figure 2.3 and it was set up at the Hydraulic Laboratory of Parma University. It was composed by a tank divided between two parts: the first one (the smaller) functioning as a reservoir and the second one (the larger) designed to receive the flood wave after the sudden removal of a gate placed in the middle of the dividing wall. The gate was moved by a pneumatic piston whose total stroke (0.20 m) was released in about 0.08 s.

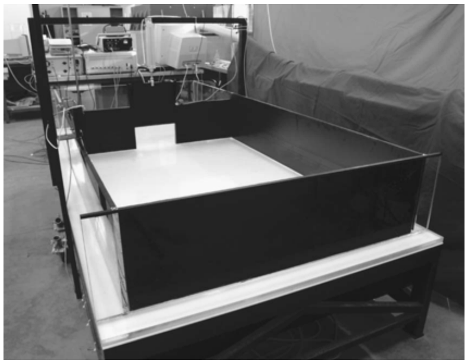


Figure 2.2: Picture of the experimental facility (sourced from Aureli, Maranzoni, Mignosa, Ziveri[1]).

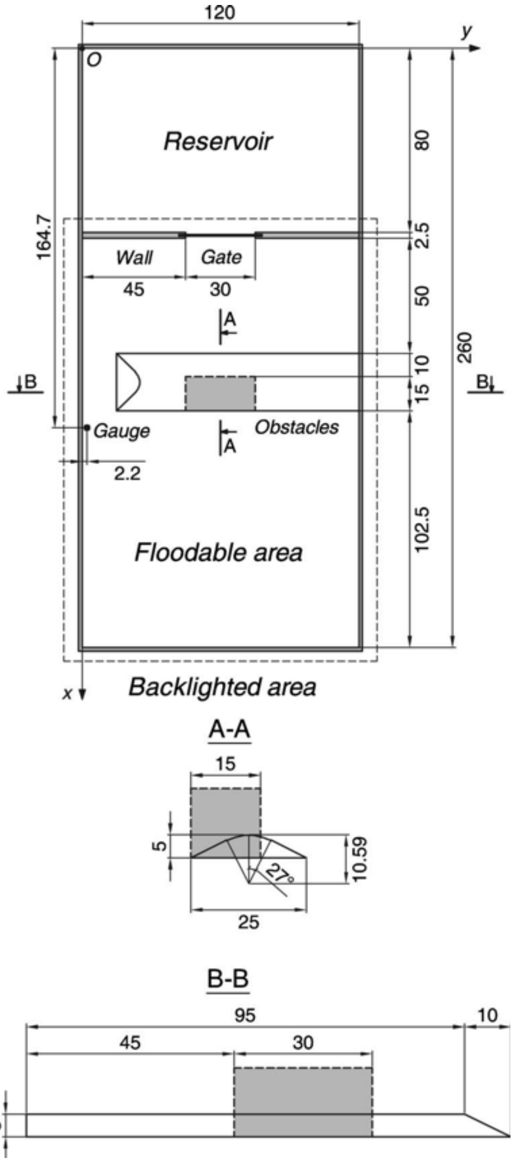


Figure 2.3: Main dimensions (in cm) of the experimental facility (sourced from Aureli, Maranzoni, Mignosa, Ziveri[1]).

The first test consisted of a sudden release of water which freely expanded in a dry area and then reflected by the lateral walls. The second test, similar to the first one, had a wet bed downstream the gate instead of a dry one. In the last two tests, authors put an object in the dry bed, submersible and asymmetric for test number three and not submersible and symmetric for the fourth one.

The imaging acquisition was performed by a Canon EOS 10D camera (SRL type) which allows to acquire images in RAW (native) in a 16-bit RGB color space and then merged into an 8-bit grey space. This was useful because as the intensity of the transmitted light decreases with water thickness, they were able to convert the data acquired by the camera sensor into quantitative information about water depth and to obtain the water depth distribution in the whole area of interest avoiding any disturbance to the flow field. Before the tests, they did a calibration increasing the water depth in the tank from 0 to 15 cm, from which they derived a transformation function between grey tones (from 0 to 255) and water depths (from 0 to 15 cm) for each pixel as shown in figure 2.4.

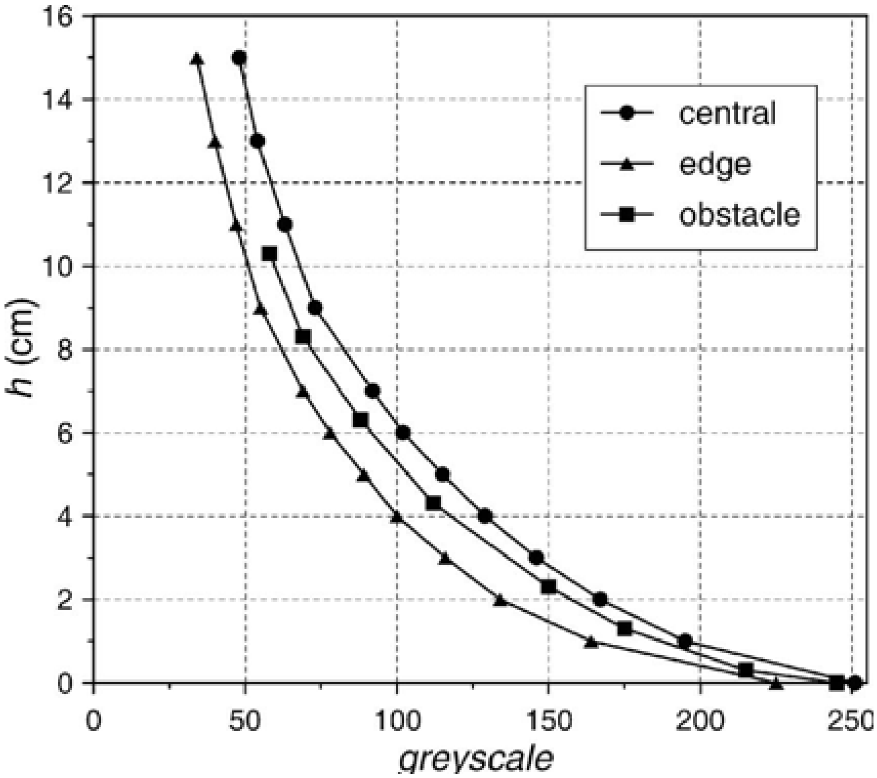


Figure 2.4: Examples of transformation functions grayscale—water depth obtained by the calibration preceding Test 3 (sourced from Aureli, Maranzoni, Mignosa, Ziveri[1]).

The four tests were then modeled through a finite volume numerical that solves the 2D shallow water problem using their integral form. The computational domain was described using discretization with square-cells of 5 mm sides length, while the Manning coefficient that was chosen was set for the bottom at 0.007 s m -1/3 .

For the purpose of this thesis, the results of test 1 and 4 of the present analysis will be presented below.

Figure 2.5 and 2.6 compare some snapshots of experimental and computed water depths in the floodable area at different times respectively for Tests 1 and 4. The first numerical frame shows some disagreements with the experimental one. Two reasons for this disagreement can be explained: the first is due to the gate removal time (0.08 s); the second is that, immediately after the release, high curvatures occur in the water surface near the breach site, so that shallow water equations cannot correctly represent the phenomenon. In the second frame the position of the wavefront on the dry bottom (Test 1, t=1.16 s, figure 2.5), and the horseshoe-shaped hydraulic jump in front of the squat obstacle (Test 4, t=0.75 s, figure 2.6) can be seen.

In the third frame, the shocks caused by the impact of the water against the lateral walls are distinctly formed, whereas in the fourth one a hydraulic jump moving upstream is clearly recognizable. In Tests 1, moreover, a vortex structure develops, characterized by low water depths in the center (they can be observed in the experiments, too).

In the last frame, the observed and simulated 2D structures present in the flow field after the impact against the downstream wall can be compared. For Test 4, in the region close to the wall, the numerical reflected wave differs from the experimental one significantly both in shape and celerity.

Figure 2.7 provides a global quantitative comparison between experimental and numerical results. In the figure the relative frequency distribution of the deviation $h_{num} - h_{exp}$ between numerical and experimental water depths is represented, together with the main sample statistics.

Figure 2.8 shows the scatter plot of the water depth values obtained by the authors through an ultrasonic transducer and the imaging technique at the gauge point shown in Figure 2.3. The data derived from the imaging technique were averaged on a $1,5$ cm X 1.5 cm area as the sensor of the ultrasonic device was 10 mm wide, its distance from the water surface was in the range of 10–20 cm, and the emission angle diverges by 3.5.

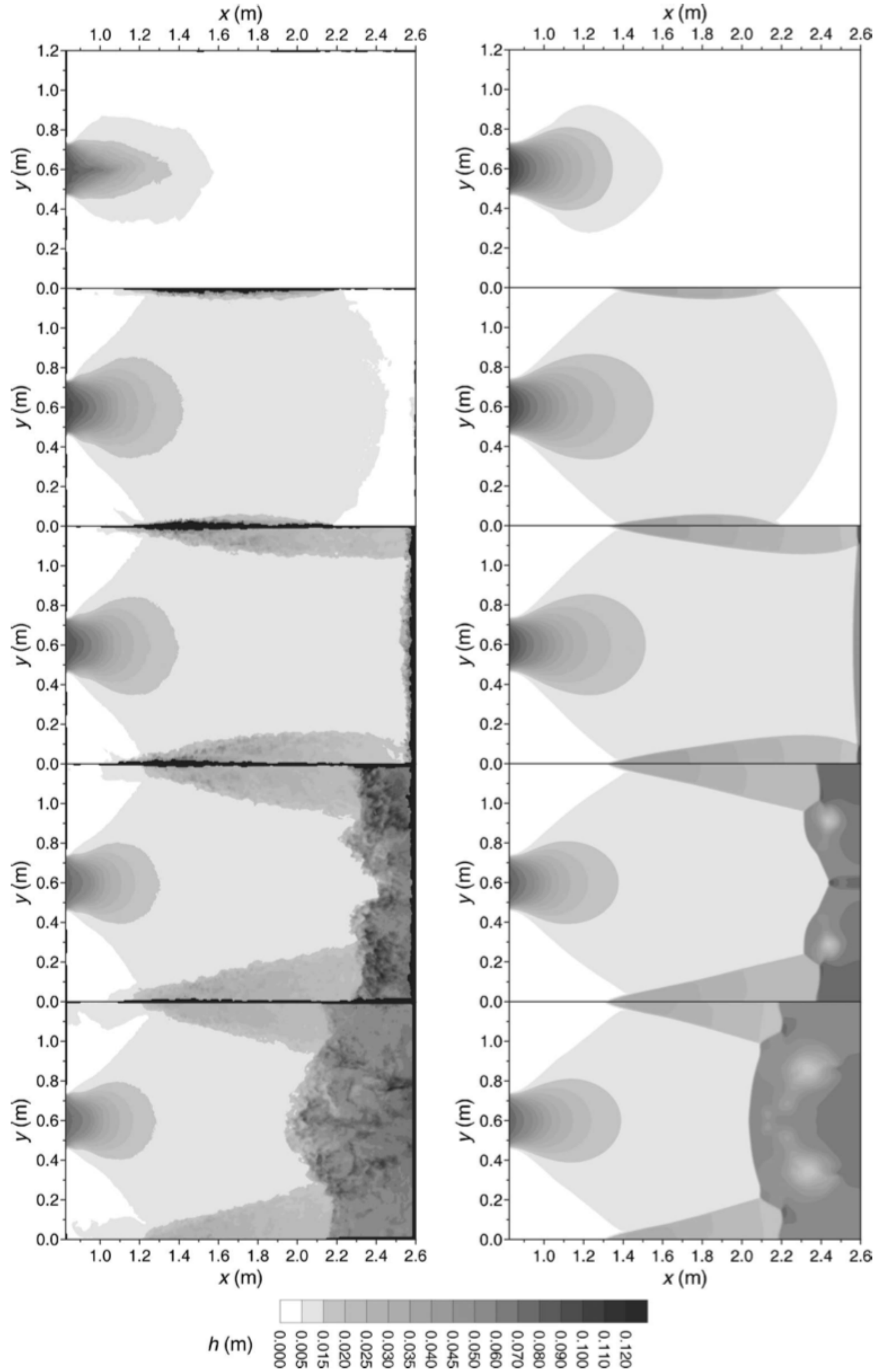


Figure 2.5: Experimental (left) and numerical (right) water depths for Test 1 at t=0.46 s; t=1.16 s; t=1.51 s; t=2.22 s; and t=2.92 s (sourced from Aureli, Maranzoni, Mignosa, Ziveri[1]).

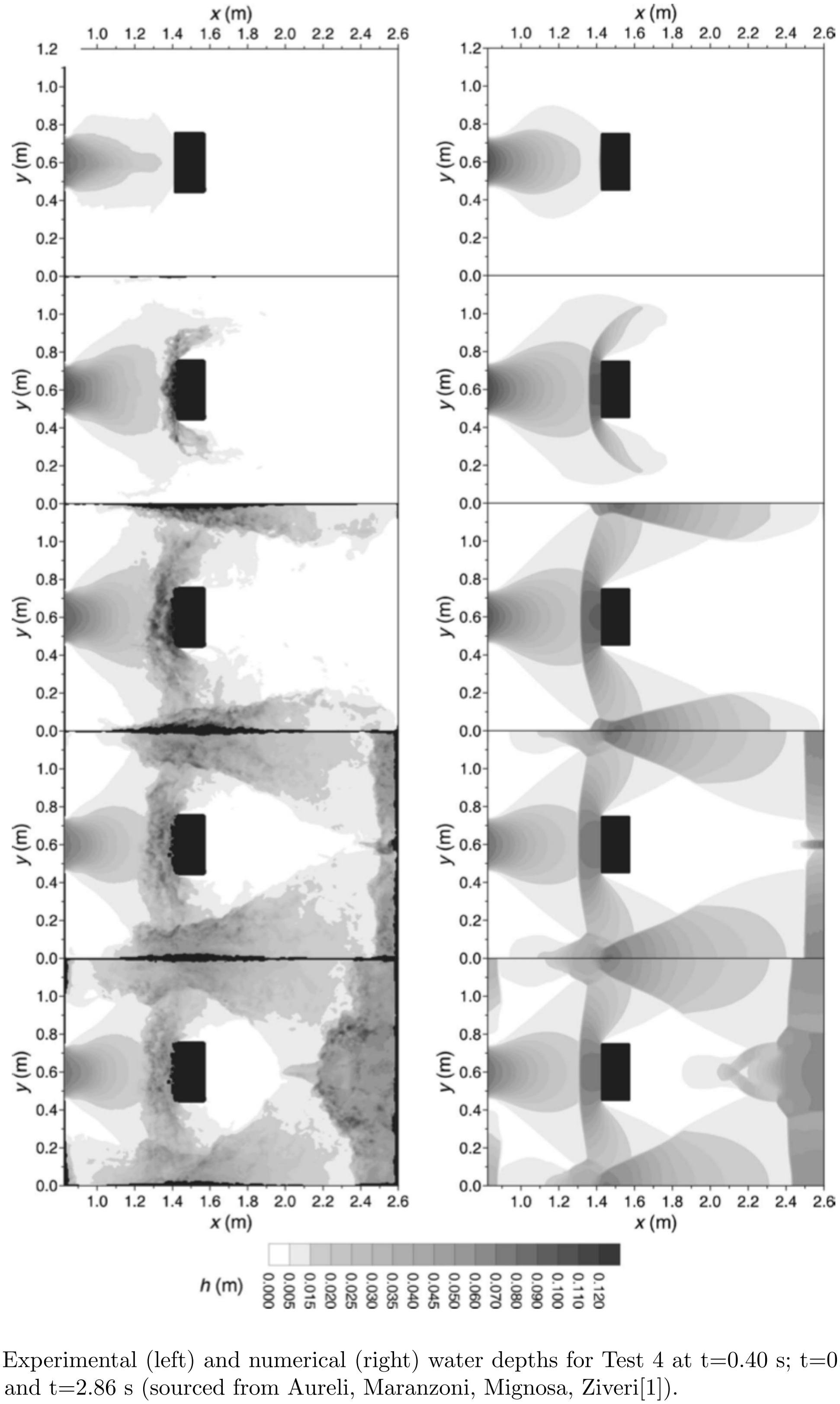


Figure 2.6: Experimental (left) and numerical (right) water depths for Test 4 at t=0.40 s; t=0.75 s; t=1.45 s; t=2.16 s; and t=2.86 s (sourced from Aureli, Maranzoni, Mignosa, Ziveri[1]).

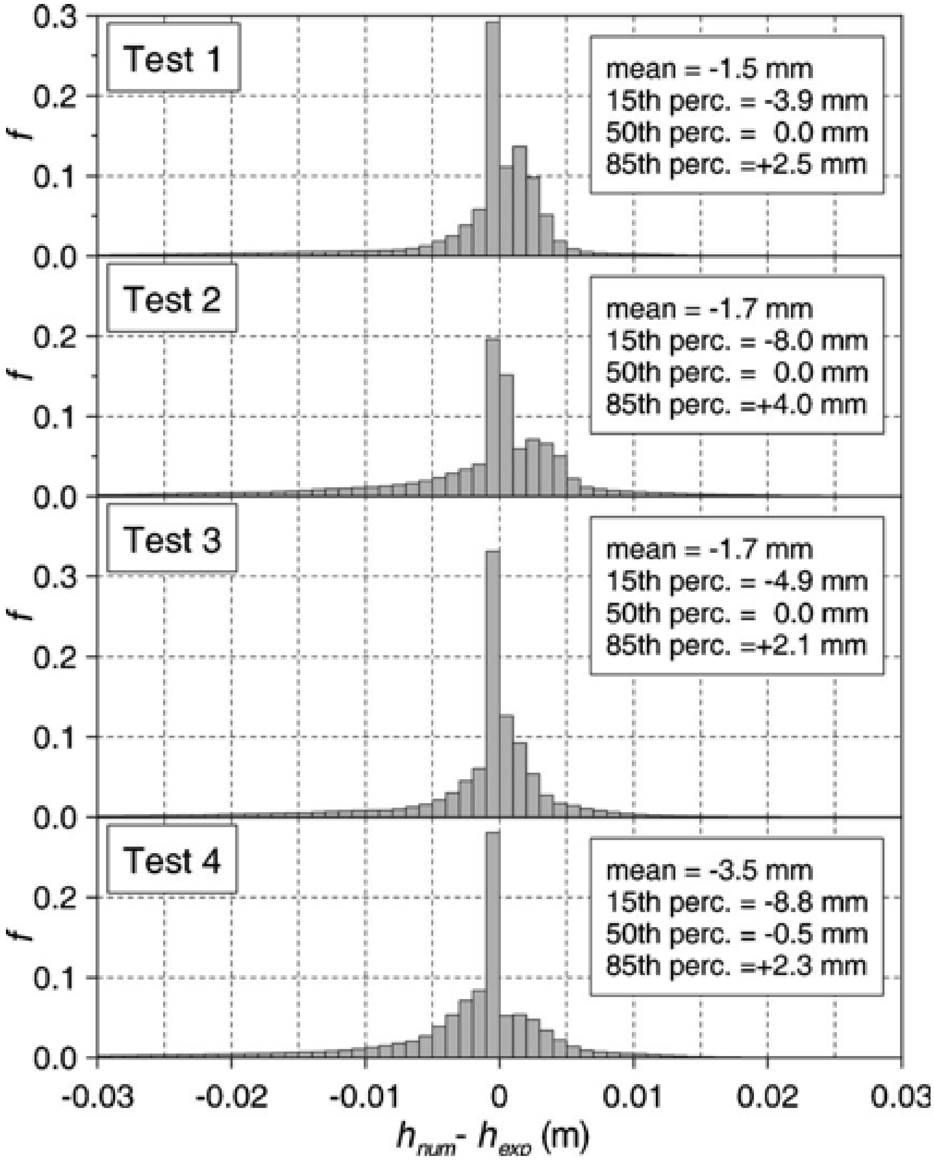


Figure 2.7: Relative frequency distribution f and main statistics of the deviation $h_{num} - h_{exp}$ between numerical and experiment (sourced from Aureli, Maranzoni, Mignosa, Ziveri [1]).

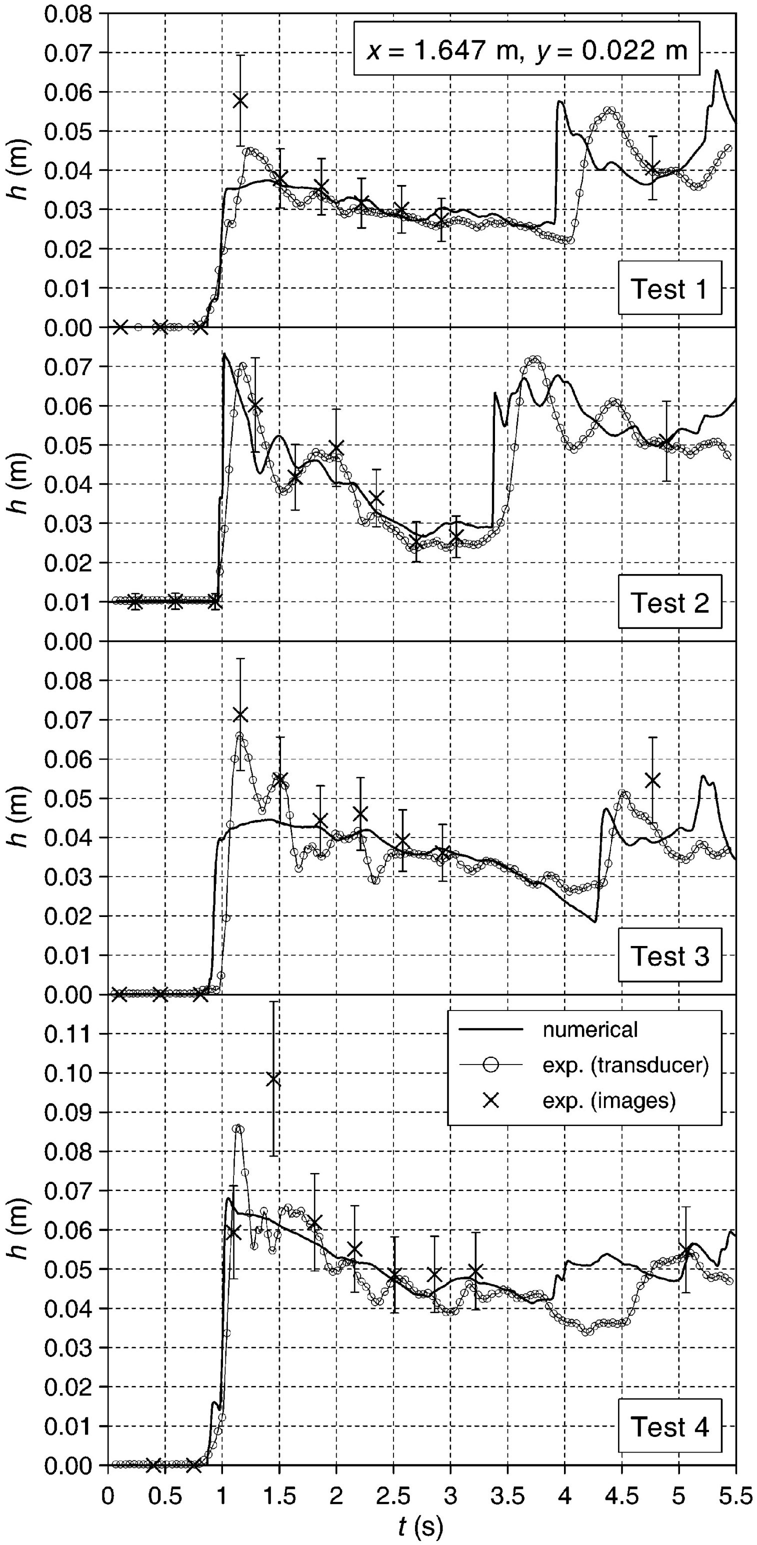


Figure 2.8: Experimental and numerical water level time series (sourced from Aureli, Maranzoni, Mignosa, Ziveri[1]).

2.2.2 Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface [S. Shao, E. Lo]

This simulation from S. Shao and E. Lo [5] is focused on the usage of an *Incompressible Smoothed Particle Hydrodynamics* (ISPH) method to simulate Newtonian and Non- Newtonian dam-break cases with free surfaces, and their comparison with two finite elements methods (FEM):

- The *Marker And Cell* (MAC) method;
- The *Volume Of Fluid* (VOF) method.

These are two of the most suitable methods for simulating free surface flows. The former uses marker particles to define the free surface while the latter solves a transport equation for the volume fraction of the fluid. Their disadvantage is that they require a fixed Eulerian grid to solve Navier-Stokes equations. The ISPH method, on the other hand, since it is a Lagrangian method, does not require such a grid, as explained in Section 1. As said above, they made two simulations, using first a Newtonian fluid and then a Non-Newtonian fluid, but for the purpose of the present thesis, only the former will be discussed. This two-dimensional simulation consisted of a rectangular column of still water kept in that position by a sluice gate and confined between two vertical walls, whit a horizontal terrain. The water column is $a = 0,1$ m wide and $H = 0,2m$ high with a particle spacing of $0,005$ m which leads to a total number of particles $N = 800$ and a computation time step of $0,0005$s. They adopted a *no-slip* boundary condition (explained in section 3.3.2). At the beginning of the computation, the sluice gate (dam) is instantaneously removed and the water is allowed to flow out along a dry horizontal bed. They compared the particle configuration for different times with the results obtained with VOF method. This is represented in figure 2.9 that shows the simulated flows patterns are very similar between the two methods.

Figure 2.10 shows a comparison of the time variation for the position of the leading edge from the dam site with experiments from Martin and Moyce, as well as the simulation performed using MAC method and another one with VOF method. The graphic represents the relationship between the normalized leading edge position $X = x/H$ and the normalized time $T = t\sqrt(g/H)$ (where g is the gravity acceleration) and it can be seen that this two are in good agreement for the different method.

The pressure isobars of the dam break flow at time $t = 0,05$ and $0,18$s , are given in figure 2.11, where it can be seen that the sudden removal of the dam causes an adjustment of the pressure field

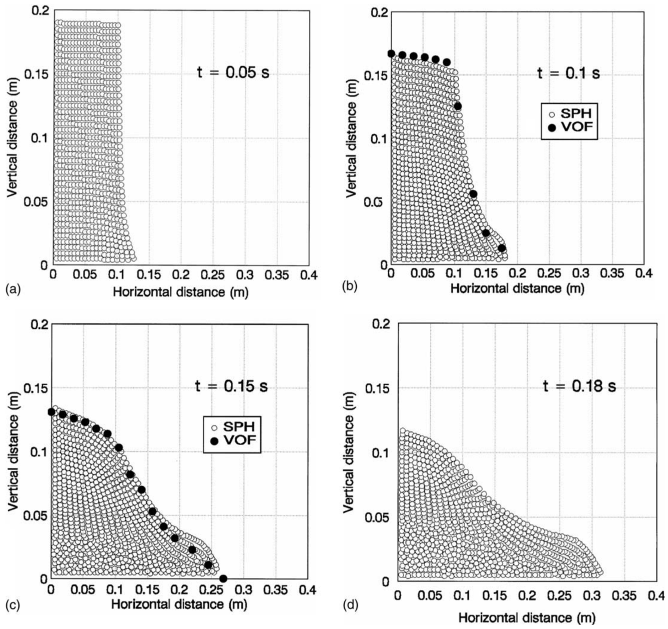


Figure 2.9: Particle configurations after collapse of dam at time (a) $t = 0,05$ s, (b) $t = 0,1$ s, (c) $t = 0,15$ s and (d) $t = 0,18$ s. Black dots correspond to computations by VOF method (sourced from S. Shao & E. Lo [5]).

since the fluid is incompressible with infinite sound speed. At time $t = 0,05$s pressure distribution inside the fluid deviates significantly from hydrostatic and the maximum pressure is about half of the initial hydrostatic value. At time $t = 0,18$s the pressure distribution is almost hydrostatic and the pressure value starts to increase.

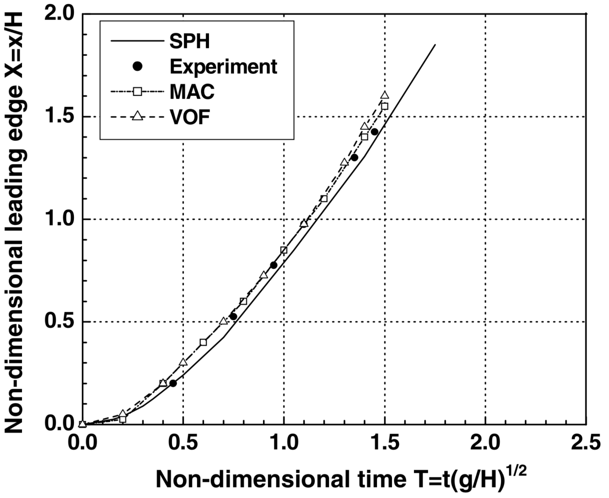


Figure 2.10: Relationship between the non-dimensional leading edge and time after dam-break (sourced from S. Shao & E. Lo[5]).

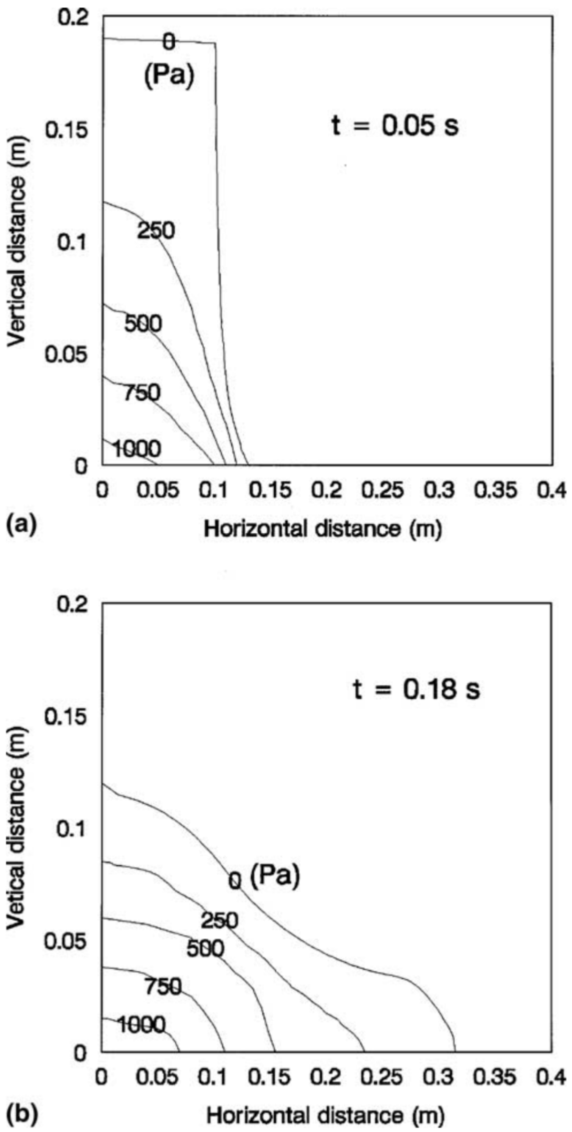


Figure 2.11: Pressure distribution after collapse of dam at time (a) $t = 0,05s$ and (b) $t = 0,18s$ (sourced from S. Shao & E. Lo[5]).

In conclusion, it can be seen how this numerical model is very effective in dealing with large deformation of free surfaces, presenting values in good agreement with experimental and different numerical data

2.2.3 Comparison between WCSPH, δ-SPH and EISPH

Four test cases were simulated by [3] to compare the performances of these three numerical schemes. These four cases were two-dimensional and they were run with the same hardware: an Intel(R)

Core(TM) i7-3770@ 3.4 GHz processor, 32GB of memory and 64-bit Windows OS.

First case: still water The aim of this test was to measure the pressure trend inside a 1m tank (2D) filled with 0.3m of still water and compare it with analytical values. The initial distance between particles was $\Delta x = \Delta xy = 0.01m$ with a smoothing length $h = 1.32\Delta x$, and time step equal to 10^{-4}s. The authors had some issues imposing the initial conditions in WCSPH, so they imposed that the initial density varied vertically and was calculated using the following equation with a hydrostatic pressure distribution

$$P = \frac{\rho_0 c_0^\gamma}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^\gamma - 1\right] \tag{2.9}$$

where c_0 is the speed of sound and $\gamma = 7$

They also added a damping term in the momentum equation (3.5). Considering these adjustments, the simulation was still very unstable. In order to solve so, they imposed a constant density, ρ_0, and constant pressure equal to zero in the initial conditions. Doing so, the simulation was stable after 30 seconds of large pressure fluctuations. These problems weren't found in the other two schemes (δ-SPH and EISPH).

Figure 2.12 shows the pressure distribution of the three different schemes and the analytical solution, after 30s of numerical simulation. As can be seen, EISPH and δ-SPH gave better results than WCSPH, with smoother pressure distribution.

Second case: Dam-Break leading edge This test compares the position of the leading edge with the experiment of a dam-break over a dry bed realized by [6]. In this experiment, they created a dam-break with an initial water volume equal to 0.06 m wide and 0.12 m high. The initial distance between the particles was $\Delta x = \Delta y = 0.005$, the smoothing length, h , was $1.32\Delta x$ and the time step was $10^{-4}s$ for WCSPH and δ-SPH and $5 \cdot 10^{-4}s$ for the EISPH scheme. As they did in the previous test, for the WCSPH the initial pressure was set equal to zero, while for the other two schemes, the pressure as initialized by solving the Poisson equation at t=0. The initial density for both WCSPH and EISPH was set to 1000 kg m^{-3} whilst, in the δ-SPH scheme, it was adjusted to

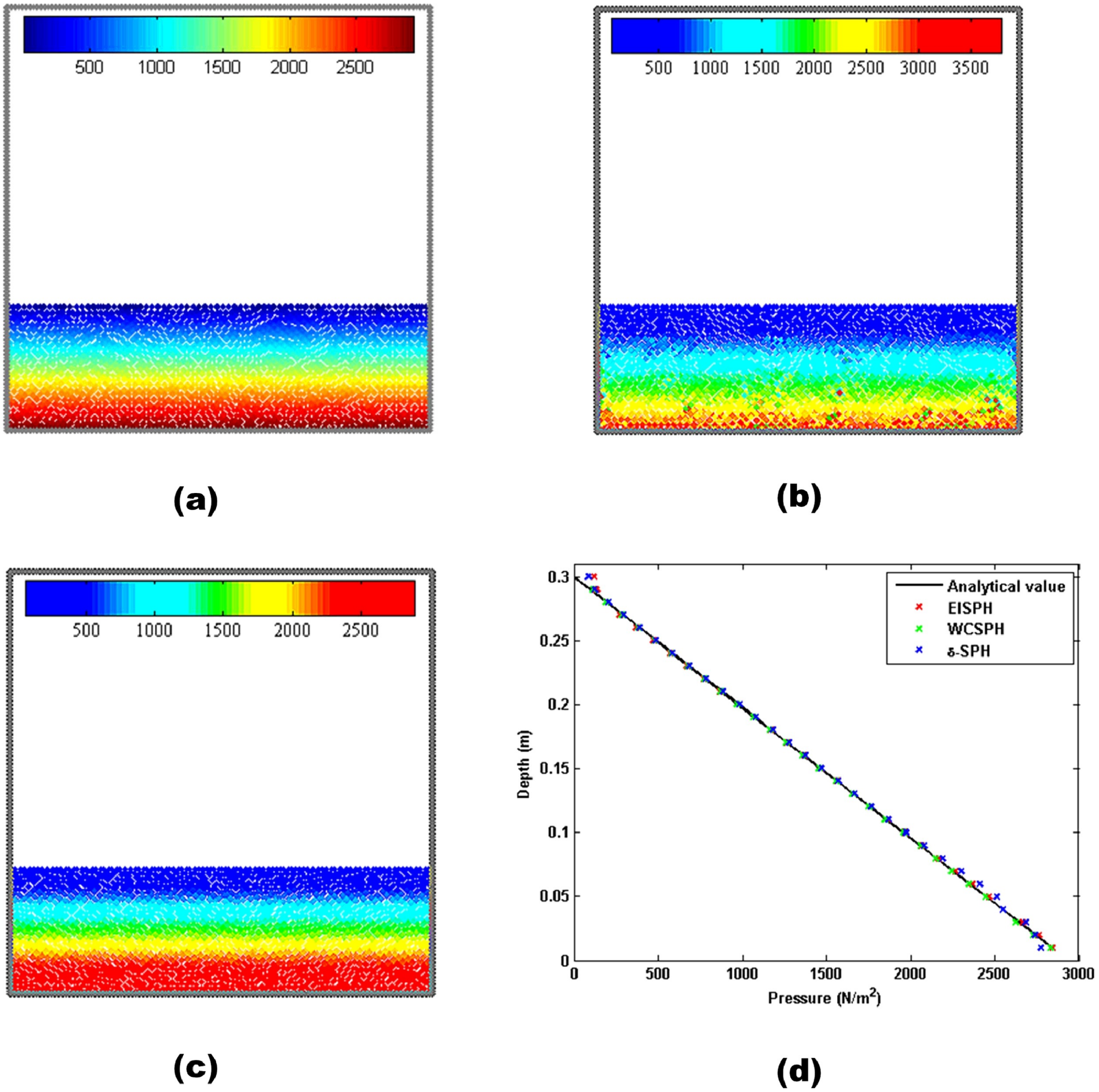


Figure 2.12: Pressure distribution in a tank with still water calculated with (a) EISPH, (b) WCSPH and (c) δ-SPH. (d) Shows pressure profiles along the middle section after 30 s from the start of the simulation (sourced from Nomeritae, E. Daly, S. Grimaldi, and H. H. Bui[3]).

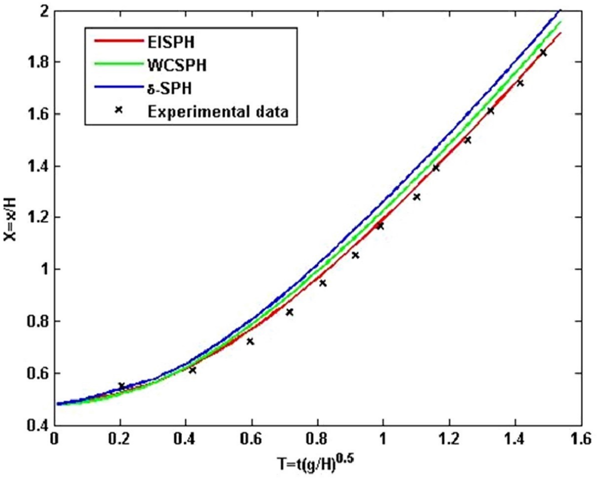


Figure 2.13: Comparison between measured and simulated leading edge position over time (sourced from Nomeritae, E. Daly, S. Grimaldi, and H. H. Bui[3]).

account for the pressure distribution using the following equation:

$$P = c_0^2(\rho - \rho_0) \tag{2.10}$$

The results in fig 2.13 show the position of the leading edge over time of the three schemes and the experimental data. It can be seen that all of the schemes reproduced well the measured data, but EISPH is the one closest to them.

Third case: Dam-Break impact This test focused on the comparison of EISPH and δ-SPH schemes simulating the experiments presented by [7] of a dam break where the water hit a vertical wall. The experiment consisted in a volume of water, instantaneously released by the removal of a gate, which generate a wave that travels towards the opposite side of the tank where it hit a wall, as shown in 2.14a. The pressure, in the experiment, was measured by two probes (P1 and P2) with 90 mm diameter located at 160 and 420 mm from the bottom of the tank respectively; the modeled pressure at the bottom left corner was also used here to compare the models and is referred to as P0. The initial distance between particles was $\Delta x = \Delta y = 0.006$ m for a total of 20,000 particles, $h = 1.32\Delta x$ and time step was $10^{-4}s$ for both EISPH and δ-SPH. The initial pressure and density for both schemes were set as in the previous test.

In the firsts steps of the simulation, the results with the two schemes appear to be the same,

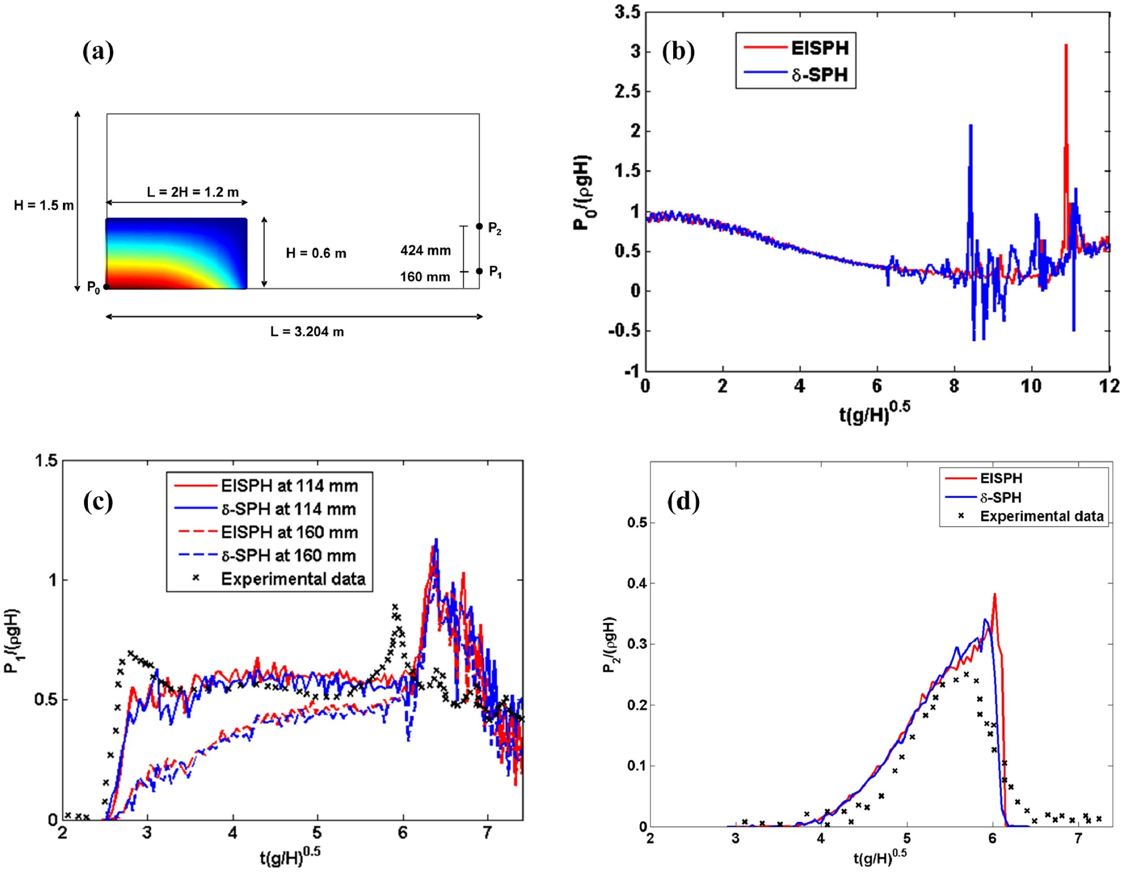


Figure 2.14: (a) Buchner experiment scheme. Comparison of pressure between EISPH and δ-SPH at P0 (b), P1 at 114 mm and 160 mm (c), and P2 (d) (sourced from Nomeritae, E. Daly, S. Grimaldi, and H. H. Bui[3]).

with δ-SPH that has more pressure fluctuations, as shown in 2.14b. The modeled pressure measured at the center of the probe P1 (160mm) shows an underestimated first peak with respect to the one observed in the experiments. So the authors, as suggested by Greco(2001) and Adami et al. (2012), simulated the pressure at the bottom of the sensor P1 (140mm) obtaining better results, as shown in 2.14c. The pressure path at the probe P2 is similar for both schemes to the measured one. In this test, the authors showed another key factor, apart from the lower pressure fluctuation, which leads the EISPH scheme to be more appropriate; they quantified and compared the computational time for running 4.2 s of simulation of physical time with 7200, 11250 and 20000 particles with $\Delta x = \Delta y = 0.01, 0.008, 0.006$ respectively, for both schemes. It can be seen by results showed in 2.15 that the EISPH computational time was about a fifth of the δ-SPH one.

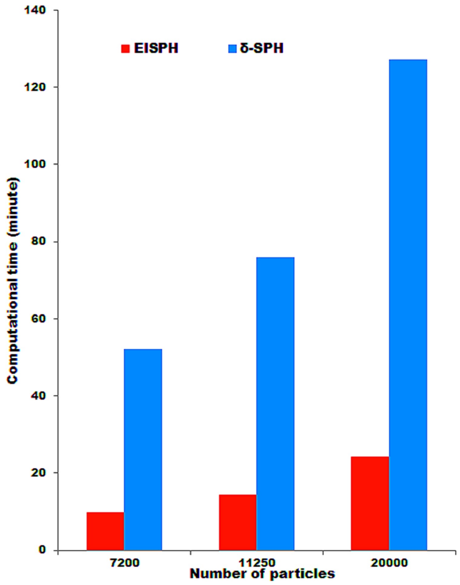
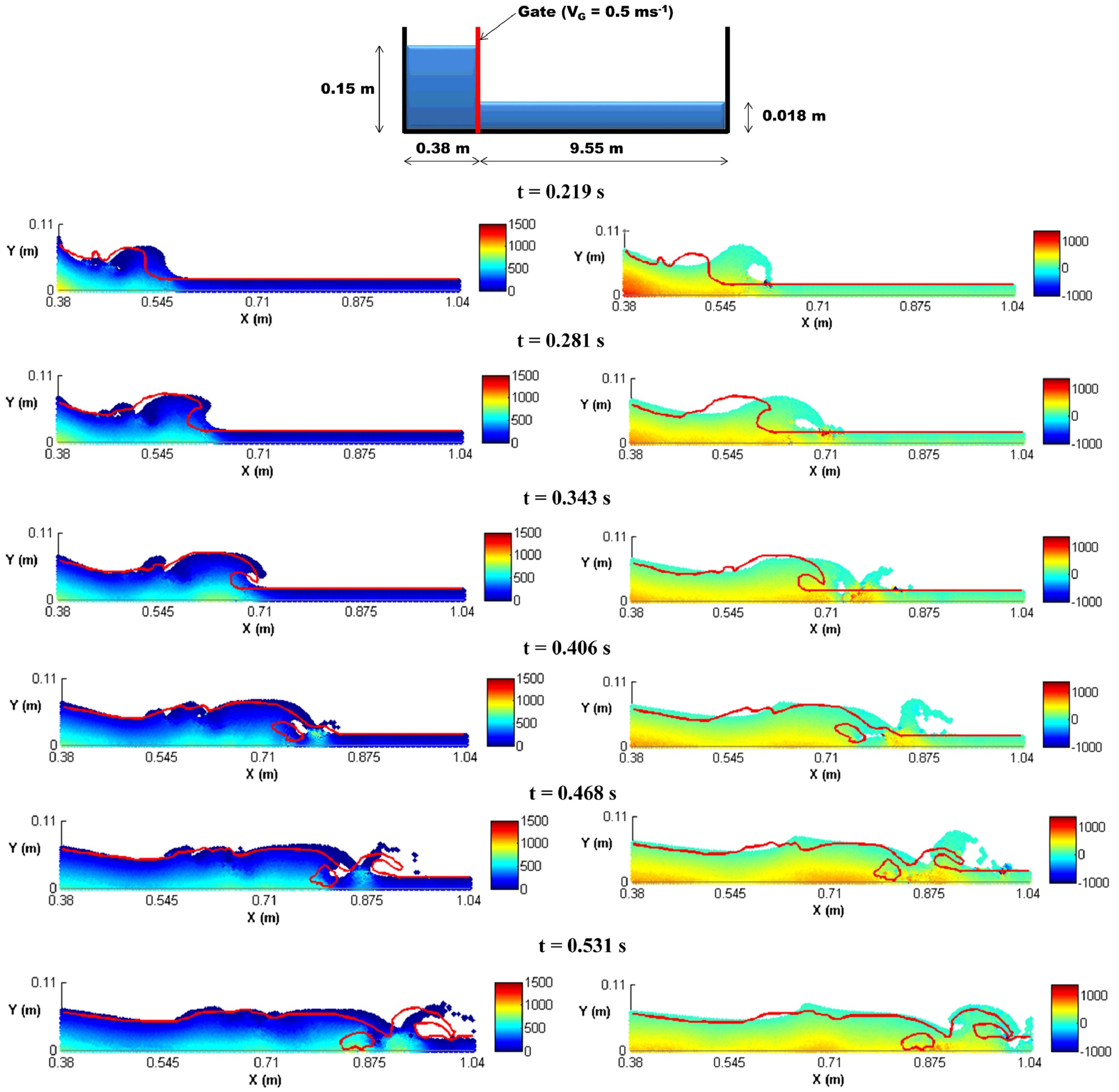


Figure 2.15: Comparison of computational time between EISPH and δ-SPH (sourced from Nomeritae, E. Daly, S. Grimaldi, and H. H. Bui[3]).

Fourth case: dam-break on wet bed In this test, the authors referred to the experiments of Jànosi et al. (2004) of a dam-break wave over a volume of standing water positioned downstream of the dam. The water volume upstream of the dam was 0.38 m by 0.15 whilst the water downstream was 0.018 m high. In the simulation, the initial distance between particle was $\Delta x = \Delta y = 0.002m$, the smoothing length $h = 1.2\Delta x$ and the time step was $10^{-}4$. The initial pressure was hydrostatic for both schemes and the density has been set as in the previous cases.

The results of this test are shown in 2.16 and it can be seen that δ-SPH produced a faster wave compared to EISPH one and also a negative pressure for some particles near the surface. This problem wasn't encountered in the EISPH scheme because they prevented it by imposing the pressure to be positive.



(a) (b)

Figure 2.16: Comparison between (a) EISPH and (b) δ-SPH for the dam-break wave evolution over a wet bed based on the experiments by Jànosi et al. (2004) represented by the red line. The coloured-bar represents pressure values (sourced from Nomeritae, E. Daly, S. Grimaldi, and H. H. Bui[3]) .

3 Methods

3.1 Equations

The basic governing equations of fluid dynamics are based on the following three fundamental physical laws of conservation:

1) conservation of mass
2) conservation of momentum
3) conservation of energy

Different forms of equations can be employed to describe the fluid flows, depending on the specific circumstances [8]. There are two approaches for describing the physical governing equations:

Eulerian description;
Lagrangian description.

The Eulerian description is a spatial description, whereas the Lagrangian description is a material description. The fundamental difference between these two descriptions is that the Lagrangian description employs the total time derivative as the combination of local derivative and convective derivative. In conjunction with the Lagrangian nature of the SPH method, the governing equations in Lagrangian form will be employed in this section.

The three main equations are the **constitutive equation**, which is related to the nature of the fluid, the **continuity equation**, which defines the conservation of mass, and the **momentum equation**, which is the Newton's law applied to fluids. The energy equation will not be used in this thesis.

3.1.1 Constitutive Equation

The first equation defines the properties of the material. The constitutive equation relates to the density of the material under consideration to some mechanical and thermal quantities. In the case of water, it is usually assumed the density varies with pressure and temperature, so that the constitutive equation in general reads

$$\rho = \rho\left(p, T\right). \tag{3.1}$$

If water is assumed incompressible, as suitable in many practical applications, the constitutive equation simply states that the density is constant:

$$\rho = const. \tag{3.2}$$

3.1.2 Continuity Equation

The continuity equation states the principle of conservation of mass. In fluid dynamics, the continuity equation states that the rate at which mass enters a system is equal to the rate at which mass leaves the system plus the accumulation of mass within the system. The differential form of the continuity equation is:

$$\frac{D\rho}{Dt} = -\rho\nabla\cdot\vec{u} = -\rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right). \tag{3.3}$$

For an incompressible fluid, the continuity equation becomes:

$$\nabla\cdot\vec{u} = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial v} + \frac{\partial w}{\partial z} = 0, \tag{3.4}$$

which means that if a particle stretches in one direction, it needs to compress in the other two directions so that the volume occupied by the particle remains constant.

3.1.3 Momentum Equation

The momentum equation is a statement of Newton's Second Law and relates the sum of the forces (i.e., volume forces (e.g., weight) and surface forces (e.g., generated by pressure)) acting on an element of fluid to its acceleration or rate of change of momentum (i.e., d(mu)/dt). For a compressible, Newtonian fluid, the momentum equation can be written as:

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \vec{g} + \nu\nabla^2\vec{u} + \frac{1}{3}\nu\nabla\left(\nabla\cdot\vec{u}\right). \tag{3.5}$$

If the fluid is incompressible, the last term of Eq. (3.5) is zero.

Eq. (3.5) is written in a vectorial form and therefore is the combination of three equations along with different directions. For example, along the direction $x\vec{i}$, the momentum equation reads

$$\frac{Du}{Dt} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + g_x + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) + \frac{1}{3}\nu\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right). \tag{3.6}$$

3.2 SPH approximation

Since hydrodynamic problems are in the form of partial differential equations (PDE) of field variables (e.g., density, velocity and energy), which do not have an analytical solution, the results obtained with the SPH method are approximations of the real solutions of sets of differential equations. These approximations are obtained by considering the domain of the equations as an ensemble of particles that interact with each other, which leads to the term *Lagrangian method*.

The first thing one needs is to discretize the problem domain where the PDEs are defined. Next, a method is needed to provide an approximation for the values of the field functions and their derivatives at any point. The function approximation is then applied to the PDEs to produce a set of ordinary differential equations (ODE) in a discretized form with respect only to time. This set of discretized ODE can then be solved using one of the standard integration routines of the conventional finite difference method.

3.2.1 Essential formulations

The SPH method can be resumed in two key steps:

- Integral representation of fields of functions (Kernel approximation)
- Particle approximation

In the first step, the integration of the multiplication of an arbitrary function and a smoothing kernel function gives the kernel approximation in the form of integral representation of the function. The concept of integral representation of a function f(x) used in the SPH method starts from the following identity:

$$f(\vec{x}) = \int_{\Omega} f(\vec{x}')\delta(\vec{x} - \vec{x}')d\Omega, \tag{3.7}$$

where Ω is the volume that contains x, and $\delta(\vec{x} - \vec{x}')$ is the Dirac function define as

$$\delta(\vec{x} - \vec{x}') = \begin{cases} 1 & \vec{x} = \vec{x}'; \\ 0 & \vec{x} \neq \vec{x}'. \end{cases} \tag{3.8}$$

The kernel approximation consists by replacing the Dirac function with the so-called *kernel function* $W(\vec{x} - \vec{x}', h)$, obtaining

$$f(\vec{x}) \approx \int_{\Omega} f(\vec{x}')W(\vec{x} - \vec{x}', h)d\Omega. \tag{3.9}$$

In the kernel function (hereinafter kernel for short), h is the smoothing length defining the influence area of the smoothing function W. The term *kernel approximation* derives from the fact that as long as that the kernel function is not equal to the Dirac function, the integral in the equation (3.9) can only be an approximation.

In the SPH convention, the kernel approximation operator is marked by the angle bracket <>, and therefore equation (3.9) is rewritten as

$$< f(\vec{x}) >= \int_{\Omega} f(\vec{x}')W(\vec{x} - \vec{x}', h)d\Omega. \tag{3.10}$$

The smoothing function W is usually chosen to be an even function and it should also satisfy a number of conditions. The first one is the *unity condition* that states

$$\int_{\Omega} W(\vec{x} - \vec{x}', h)d\Omega = 1. \tag{3.11}$$

The second condition is the *Dirac function property* that is observed when the smoothing length approaches zero

$$\lim_{h \to 0} W(\vec{x} - \vec{x}', h) = \delta(\vec{x} - \vec{x}'). \tag{3.12}$$

The third and last condition is the *compact condition*, which states

$$W(\vec{x} - \vec{x}', h) = 0 \quad when |\vec{x} - \vec{x}'| > kh. \tag{3.13}$$

where k is a constant related to the smoothing function for point x, and defines the effective area of the smoothing function. This effective area is called the support domain for the smoothing function of point x (or the support domain of that point). Using this compact condition, integration over the entire problem domain is localized as integration over the support domain of the smoothing function. Therefore, the integration domain Ω can be, and usually is, the same as the support domain. The most commonly used kernel function which satisfy all these requests is the cubic splines, defined as

$$W(|r|, h) = \frac{\sigma}{h^\nu} \begin{cases} 1 - \frac{3}{2}\frac{|r|^2}{h^2} + \frac{3}{4}\frac{|r|^3}{h^3} & 0 \leq \frac{|r|}{h} \leq 1; \\ \frac{1}{4}\left(2 - \frac{|r|}{h}\right)^2 & 1 < \frac{|r|}{h} \leq 2; \\ 0 & \frac{|r|}{h} > 2, \end{cases} \tag{3.14}$$

where ν is the number of dimensions (1, 2 or 3) and σ is equal to $2/3$, $10/(7\pi)$ or $1/\pi$) respectively when the number of dimensions is 1, 2 or 3.

The support domain of the kernel function is $2h$. Figure 3.1 shows the cubic spline kernel function and its derivative.

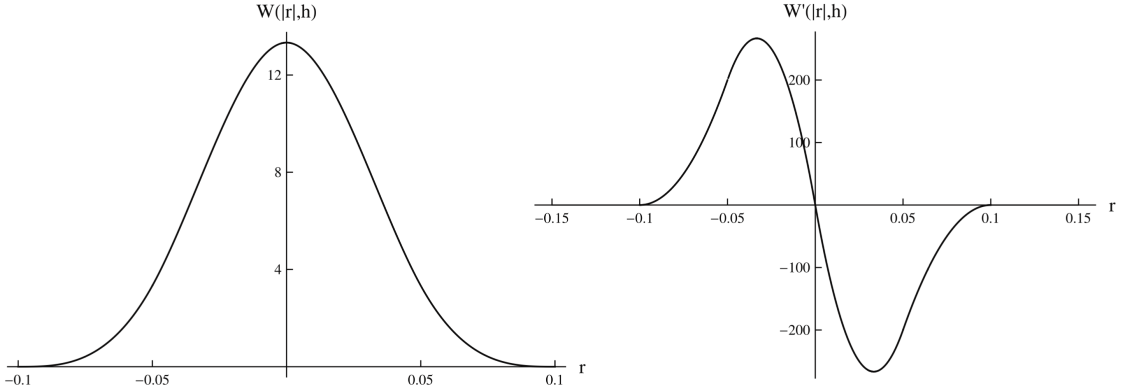


Figure 3.1: Cubic Spline kernel function (left) and its derivative (right).

For the SPH method, there's also the need to define the integral approximation of the spatial derivative $\nabla \cdot f(\vec{x})$. This is obtained simply by substituting the $f(x)$ with the with $\nabla \cdot f(\vec{x})$ in the equation (3.10), which gave

$$< \nabla \cdot f(\vec{x}) >= \int_{\Omega} [\nabla \cdot f(\vec{x}')] W(\vec{x} - \vec{x}', h) d\Omega. \tag{3.15}$$

Using the definition of derivative and the divergence theorem, for those points whose support domain is inside the problem domain, equation (3.15) become as follow:

$$< \nabla \cdot f(\vec{x}) >= \int_{\Omega} [f(\vec{x}')] \cdot \nabla W(\vec{x} - \vec{x}', h) d\Omega. \tag{3.16}$$

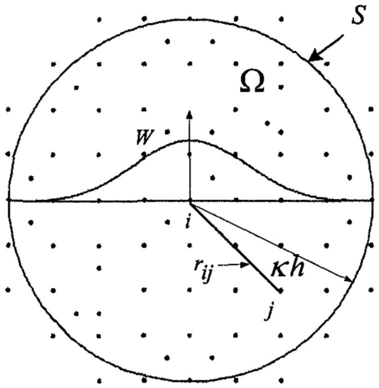


Figure 3.2:
2D scheme of the particle approximations using particles within the support domain of the smoothing function W for particle i. The support domain is circular with a radius of kh.

Since the system is represented by a finite number of particles that carry individual mass and occupy individual space, the next step is to go from a continuous description to a discretized one. This is achieved by the *particle approximation.*

Space is so discretized by particles occupying a defined volume and having a certain mass. The volume of the j-th particle is $dV_j \approx \Delta V_j = m_j/\rho_j$ and thus the integral in equation (3.10) can be approximated as a summation over the volume of the domain. Since the approximation is performed using the kernel function, which has a limited support, the integral, and thus the summation, can be limited to the support of the kernel function (Figure 3.2).

For a generic function, the particle approximation reads:

$$< f(\vec{x}) >= \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\vec{x}_j) W(\vec{x} - \vec{x}', h), \tag{3.17}$$

which can be also be written as

$$< f(\vec{x}) >= \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\vec{x}_j) W_{ij}, \tag{3.18}$$

where

$$W_{ij} = W(\vec{x}_i - \vec{x}_j, h) = W(r_{ij}, h) = W(-r_{ij}, h), \tag{3.19}$$

where the second equality derives from the fact that W is even. Equation (3.17) states that the value of a function at particle i is approximated using the average of those values of the function at all the particles in the support domain of particle i weighted by the smoothing function.

The same thing can be one for the derivative of a function, simply by substituting summation to integral, as follow

$$< \nabla \cdot f(\vec{x}) >= -\sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\vec{x}_j) \cdot \nabla_j W_{ij}, \tag{3.20}$$

where

$$\nabla_j W_{ij} = \frac{\vec{x}_i - \vec{x}_j}{|\vec{x}_i - \vec{x}_j|} \frac{\partial}{\partial r_{ij}} W_{ij}, \tag{3.21}$$

the negative sign in equation (3.20) can be removed taking $\nabla_i W_{ij}$ with respect to the particle i.

3.3 EISPH

3.3.1 Numerical schemes

The equations described in section 3.1 can be discretised as:

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^{N} m_j(\vec{u}_i - \vec{u}_j) \cdot \nabla_i kiolp W_{ij}, \tag{3.22}$$

$$\frac{D\vec{u}_i}{Dt} = -\sum_{j=1}^{N} m_j \left(\frac{P_i}{\rho_i} + \frac{P_j}{\rho_j}\right) \cdot \nabla_i W_{ij} + \nu\nabla^2\vec{u} + \vec{g}. \tag{3.23}$$

The approximation of the viscosity is written as:

$$\nu\nabla^2\vec{u} = \sum_{j=1}^{N} m_j \frac{8(\nu_i + \nu_j)(\vec{u}_i - \vec{u}_j) \cdot \vec{x}_{ij}}{(\rho_i + \rho_j)|\vec{x}_{ij}|^2 + \eta^2} \nabla_i W_{ij}. \tag{3.24}$$

η is a small parameter included to ensure that the denominator remains different from zero.

Using a prediction correction method, the integration for equations (3.22) and (3.23) can be obtained. For doing so, intermediate position ($\vec{x}_*$) and velocity ($\vec{u}_*$) are needed. They can be calculated as

$$\vec{x}_* = \vec{x}^n + \vec{u}_*\Delta t, \tag{3.25}$$

$$\vec{u}_* = \vec{u}^n + (\nu\nabla^2\vec{u} + \vec{g})\Delta t, \tag{3.26}$$

where $\vec{x}^n$ and $\vec{u}^n$ are position and velocity at time t^n.

This two equation are used to calculate an intermediate value of density ρ_*, which needs to be corrected to enforce the incompressibility.

In the correction step, the intermediate velocity is updated by accounting for the pressure gradient:

$$\vec{u}^{n+1} = \vec{u}_* - \frac{1}{\rho_*}\nabla P^{n+1}\Delta t, \tag{3.27}$$

where $\vec{P}^{n+1}$ and $\vec{u}^{n+1}$ are pressure and velocity at time t^{n+1} bringing $\vec{u}_*$ to the first member and dividing by Δt, equation (3.27) become

$$\frac{\vec{u}^{n+1} - \vec{u}_*}{\Delta t} = -\frac{1}{\rho_*}\nabla P^{n+1}. \tag{3.28}$$

The divergence of equation (3.28) read

$$\nabla \cdot \left(\frac{\vec{u}^{n+1} - \vec{u}_*}{\Delta t}\right) = \nabla \cdot \left(-\frac{1}{\rho_*}\nabla P^{n+1}\right). \tag{3.29}$$

The condition to enforce the incompressibility is

$$\nabla \cdot \left(\frac{\vec{u}^{n+1} - \vec{u}_*}{\Delta t}\right) = 0. \tag{3.30}$$

Thus, combining, equation (3.29) and (3.30), one obtains

$$\nabla \cdot \left(\frac{1}{\rho_*}\nabla P^{n+1}\right) = \frac{1}{\Delta t}\nabla \cdot \vec{u}^{n+1}. \tag{3.31}$$

Useful to solve equation (3.27).

The Laplacian operator in equation (3.31) can be approximated as

$$\nabla \cdot \left(\frac{1}{\rho_*} \nabla P^{n+1} \right) = \sum_{j=1}^{N} m_j \frac{8}{(\rho_{i*} + \rho_{j*})^2} \frac{\left(P_i^{n+1} P_j^{n+1} \right) \vec{x}_{ij} \cdot \nabla_i W_{ij}}{|\vec{x}_{ij}|^2 + \eta^2}. \tag{3.32}$$

So, considering equation (3.32), becomes

$$\sum_{j=1}^{N} m_j \frac{8}{(\rho_{i*} + \rho_{j*})^2} \frac{\left(P_i^{n+1} P_j^{n+1} \right) \vec{x}_{ij} \cdot \nabla_i W_{ij}}{|\vec{x}_{ij}|^2 + \eta^2} = \frac{1}{\Delta t} \nabla \cdot \vec{u}^{n+1}, \tag{3.33}$$

which, for simplicity, can be written as

$$P_i^{n+1} = \frac{B_i + \sum_{j=1}^{N} A_{ij} P_j^{n+1}}{\sum_{j=1}^{N} A_{ij}}, \tag{3.34}$$

with

$$A_{ij} = m_j \frac{8}{(\rho_{i*} + \rho_{j*})^2} \frac{\vec{x}_{ij} \cdot \nabla_i W_{ij}}{|\vec{x}_{ij}|^2 + \eta^2}, \tag{3.35}$$

and

$$B_i = \frac{1}{\Delta t} \nabla \cdot \vec{u}^{n+1}. \tag{3.36}$$

If the increment of time is small enough, one can be assumed that

$$P_j^{n+1} = P_j^n, \tag{3.37}$$

so that can be write

$$P_i^{n+1} = \frac{B_i + \sum_{j=1}^{N} A_{ij} P_j^{n}}{\sum_{j=1}^{N} A_{ij}}. \tag{3.38}$$

The pressure so calculated is then used to calculate the velocity $\vec{u}^{n+1}$ with equation (3.27). Thus the position can be calculated as

$$\vec{x}^{n+1} = \vec{x}^{n} + \frac{\vec{u}^{n} + \vec{u}^{n+1}}{2}\Delta t. \tag{3.39}$$

Note that the intermediate density ρ_* in equation (3.38) is obtained with equation (3.22), while the density is imposed to its original value ρ_0 before the next time step.

3.3.2 Implementation

The implementation of the EISPH scheme takes place by means of a code written in the *Fortran90* language. In this code, the initial distribution of the particles and the boundary conditions are defined and all the equations are solved.

Initialization Here the initial conditions are defined by creating the so-called *Real Particles*: these are the particles that compose the fluid in the domain and take part in the SPH approximation. In order to impose the initial conditions, the first thing one needs is to define the fluid domain shape and dimensions, which means build the spatial configuration of the REAL particles and then give a diameter to them, which will define the number of these within the simulation.

For each real particle is then assigned:

- An integer number that defines the nature of the particle (usually for the real particles, the number 2 is assigned);
- The initial density;
- The mass;
- The initial pressure distribution;
- The smoothing length.

Boundary conditions Near the boundary, a *particle deficiency* emerges since the integral is truncated here[8]. For the particles which are located in this region, only particles inside the boundary contribute to the summation of the particle interaction, and there isn't a contribution from outside since there are no particles beyond the boundary, as demonstrated in figure 3.3. This means that if a real particle is moving towards the boundary, it doesn't encounter anything that reflects its motion back to the domain. So, in order to avoid this, the other two kinds of particles are created to treat the boundary. The first kind is called " *Virtual Particles*". These particles are located on the boundary and they produce a repulsive force to the real particles which are located near the boundary, and thus to prevent these particles from unphysically penetrating through the boundary. So, If a boundary particle is the neighboring particle of a real particle that is approaching the boundary, a force is applied pair wisely along the centreline of these two particles, and it's calculated as follows.

$$BF_{ij} = \begin{cases} c^2 \left[\left(\frac{r_0}{r_{ij}} \right)^{p_1} - \left(\frac{r_0}{r_{ij}} \right)^{p_2} \right] \frac{1}{r_{ij}} & if \quad \frac{r_0}{r_{ij}} \leq 1; \\ 0 & if \quad \frac{r_0}{r_{ij}} > 1, \end{cases} \tag{3.40}$$

- r_0 is the *cutoff distance* and it's a fraction of the particle diameter dx, dy or dz (in most of the application they are the same) . For this thesis, the value for r_0 is set as

$$r_0 = 0,4dx. \tag{3.41}$$

- p_1 and p_2 are dimensionless constants that need to be calibrated in order to give the right value to the repulsive force. For the 3D EISPH simulations used here, their value is $p_1 = 4$ and $p_2 = 2$.

- c^2 is a problem dependent parameter and is chosen to be in the same scale as the square of the largest velocity.

The repulsive force for every particle is then added to the external force term in the momentum equation 3.23. The position and physical variables for the virtual particles is initialized at the beginning of the simulation and do not evolve in the simulation process. To get the impenetrability condition stronger, the second kind of boundary particles is generated and it's called " *Ghost Particles*". These ones are constructed in the following way: when a real particle i is located within a distance of kh_i from the boundary (where k is a certain constant and h is the smoothing length), a ghost particle is placed symmetrically on the outside of the boundary and the following characteristics are given to it:

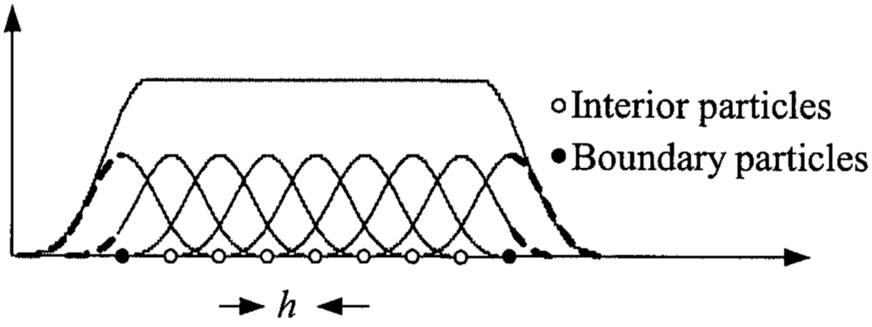


Figure 3.3: *SPH kernel and particle approximations for interior and boundary particles[8].*

- same pressure as the real particle i;
- same density as the real particle i;
- same diameter as the real particle i;
- a certain velocity in the perpendicular and in the tangential direction.

The last one is the key of the boundary condition. When the real particle is moving toward the boundary with a velocity it needs that the symmetrical ghost particle has an opposite *perpendicular velocity* to it. The value of this velocity determines the type of impact that the real particle makes on contact with the boundary; in most of application the perpendicular velocity given to the ghost particle is equal and opposite to the real particle's one. For example, for a three-dimensional case with an x, y, z axes system, if a real particle is moving with a velocity $V = (\vec{v_r}, \vec{u_r}, \vec{w_r})$ in a perpendicular direction to the xy wall, the generated ghost particle will have a perpendicular velocity $w_g = -w_r$.

The tangential velocity of the ghost particle determines the resistance applied to the real particle. In order to have a *no-slip* boundary condition, this velocity needs to be equal in value and opposite in direction to the tangential velocity of the real particle as in figure 3.4. Giving the same direction instead, one would have a *free slip* condition as shown in figure 3.5. Playing with the value of this speed it's possible to set the resistance at the boundary.

Ghost particles can both be applied to treat solid boundaries and free surfaces. The numerical tests have shown that this treatment of the boundary is very stable and effective. It not only improves the accuracy of the SPH approximation in the boundary region but also prevents the

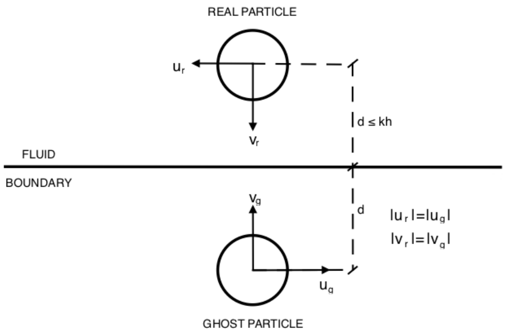


Figure 3.4: *"No-slip" boundary condition.*

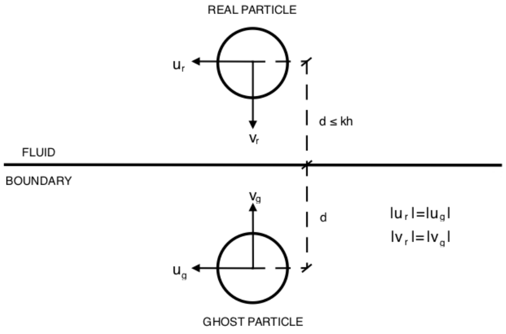


Figure 3.5: *"Free-slip" boundary condition.*

unphysical particle penetration outside the solid boundary [8].

However, a difficulty arises when one wants to create an impenetrable surface with a geometrically complex shape or that is inside the domain (such as an obstacle, a sluice gate, etc.): in order to define the anti-penetration repulsive force to the virtual particles on the surface, one needs to create the normal vectors to such surface first; this is easy for linear geometry like a square (or a cube in the three-dimensional case), but it can become really difficult for a complex figure. Plus, if the surface is surrounded by fluid, like a sluice gate, the generated ghost particles need a special treatment to interact only with the fluid on the other side of the surface and not with the fluid present in their same side, this would cost errors in the simulations.

This leads to another type of particles, called "*object particles*" which are realized like the virtual particles, but unlike these, they interact with the real particles like they were also real particles, but without changing position during the simulation. This approach needs a bigger computational effort than the one with virtual and ghost particles, so it's preferred to create only obstacles within the domain. In this thesis both the approaches were used: virtual and ghost particles to create the boundaries and object particles to create the obstacles and the sluice gates.

Time Integration As an explicit hydrodynamic method, the discrete EISPH equations can be integrated with different methods. One of these is the *Leap-Frog (LF)*, predictor-corrector which is an algorithm that requires low memory storage for the computation and has the efficiency for one force evaluation per step. In some cases where the smoothing lengths become very small, the time step can become very small to be prohibitive. In these cases, a better choice would be the *Runge-Kutta (RK)* integrator with adaptive time-step. The disadvantage with this method is that it needs two force evaluations per step and thus is computationally expensive.

In particular, this code used for this thesis implements the LF algorithm. In this integration methods, the equations for updating velocity and position are

$$v_{n+1/2} = v_{n-1/2} + a_n \Delta t, \tag{3.42}$$

$$x_{n+1} = x_n + v_{n+1/2} \Delta t, \tag{3.43}$$

$$a_n = F(x_n). \tag{3.44}$$

where:

- x_n is the position at step n;
- $v_{n+1/2}$ is the velocity at step $n + 1/2$;
- a_n is the acceleration at step n;
- Δt is the size of each time step

The numerical stability of the LF algorithm is governed by the *Courant-Friedrich-Levy (CFL)* condition, which states that the computational domain of dependence in a numerical simulation should include the physical domain of dependence, or the maximum speed of numerical propagation must exceed the maximum speed of physical propagation[8]. This CFL condition requires the time step to be proportional to the smallest spatial particle resolution, which in SPH applications is represented by the smallest smoothing length h:

$$\Delta t_1 \leq C_{cour} \left(\frac{h}{V_{max}} \right), \tag{3.45}$$

where V_{max} is the maximum velocity in the computation. C_{cour} is the Courant coefficient and is chosen to be equal to $0,5$.

In the EISPH scheme, another expression for estimating time step when considering viscous diffusion is taken into account

$$\Delta t_2 \leq 0.5 \frac{h^2}{\nu} = 0.5 \frac{h^2}{\mu} \rho, \tag{3.46}$$

where $\nu = \mu/\rho$ is the kinetic viscosity.

So the minimum time step that needs to be taken in order to have a stable LF algorithm is

$$\Delta t = min(\Delta t_1, \Delta t_2). \tag{3.47}$$

Particle Interaction: Nearest neighbouring particle searching (NNPS) As explained in section 3.2.1, the smoothing function has a compact support domain of dimension kh; this means the only a finite number of particles are within this support domain of the concerned particle and are used in the particle approximations. These particles are generally referred to as *nearest neighbouring particles (NNP)* for the concerned particle. So the process of finding the nearest particles is commonly referred to as *nearest neighbouring particle searching (NNPS).* The particularity of the NNPS for SPH in respect of a classic grid-based numerical method, where the position of neighbour grid-cells are well defined once the grids are given, is that the nearest neighbouring particles for a given particle can vary with time.

An SPH code usually uses one of the following NNPS approaches:

1. *All-pair search algorithm*;
2. *Tree search algorithm*;
3. *Linked-list search algorithm.*

1. The All-pair search approach is a really easy and direct method. For a given particle i, the distance r_{ij} from every particle j in the problem domain is calculated; if $r_{ij} < kh$, particle j is found belonging to the support domain of particle i, and if the smoothing length is symmetrical, then particle i is also within the support domain of particle j. Thus i and j are a couple of NNP.

Is easy to understand though, that, since this searching is carried out for every particle in the domain $i = 1, 2, ..., N$ and for everyone of these, $j = 1, 2, ..., N$ particles are searched, leading to a computational effort proportional to N^2, which is too much, since the NNPS is needed at every time step. So the All-pair search algorithm is inadequate for problems that involve a certain number of particles.

2. The Tree search algorithm is suggested for problems with variable smoothing lengths. It involves creating ordered trees according to the particle positions. Once the tree structure is created, it can be used efficiently to find the nearest neighbouring particles. For a given particle i, a square (a cube in a three-dimensional case) with the side of $2kh$ is used to enclose the particle, which is located at the center of the square. At each level, the algorithm checks if the volume of the search square overlaps the volume represented by the current node in the tree structure. If not, it discontinues the descent down on that particular path. If yes, it continues the tree descent and proceeds down to the next level repeatedly until the current node represents a particle. Then it checks if the particle is within the support domain of the given particle i. If yes, it's recorded as a NNP. This algorithm is very efficient and robust especially for a large number of particles of variable smoothing lengths and its computational effort proportional to $NlogN$[8].

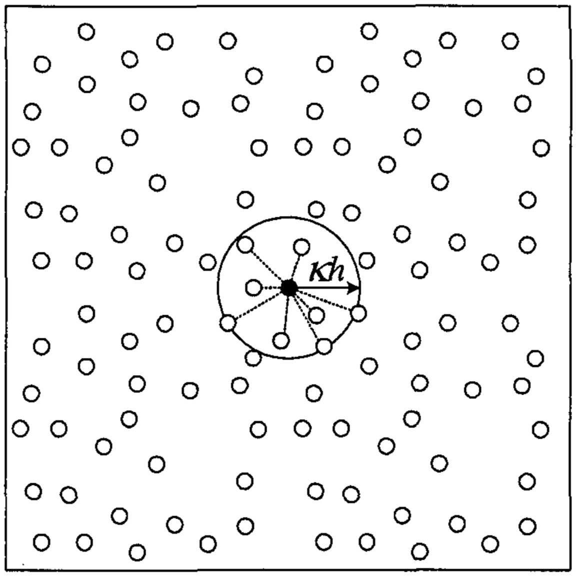


Figure 3.6: *All-pair search algorithm for searching the nearest neighbouring particles in two-dimensional space[8].*

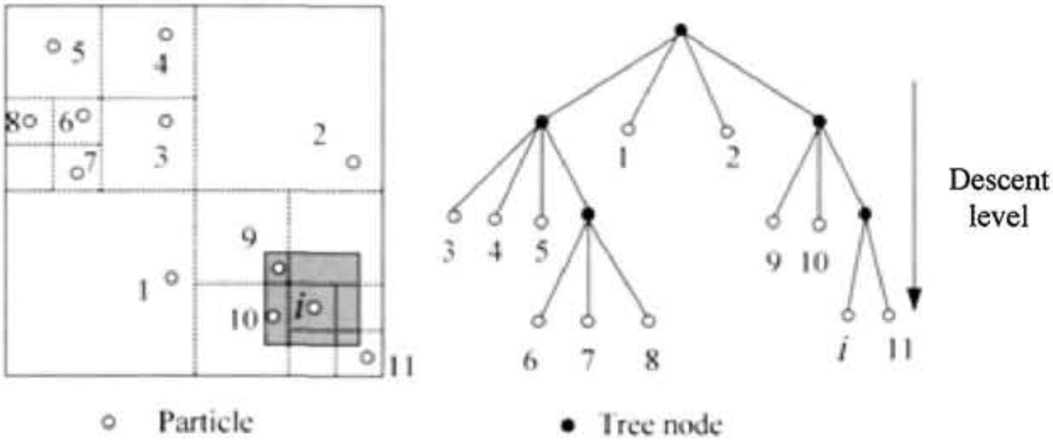


Figure 3.7: *Tree structure and tree search algorithm in two-dimensional space[8].*

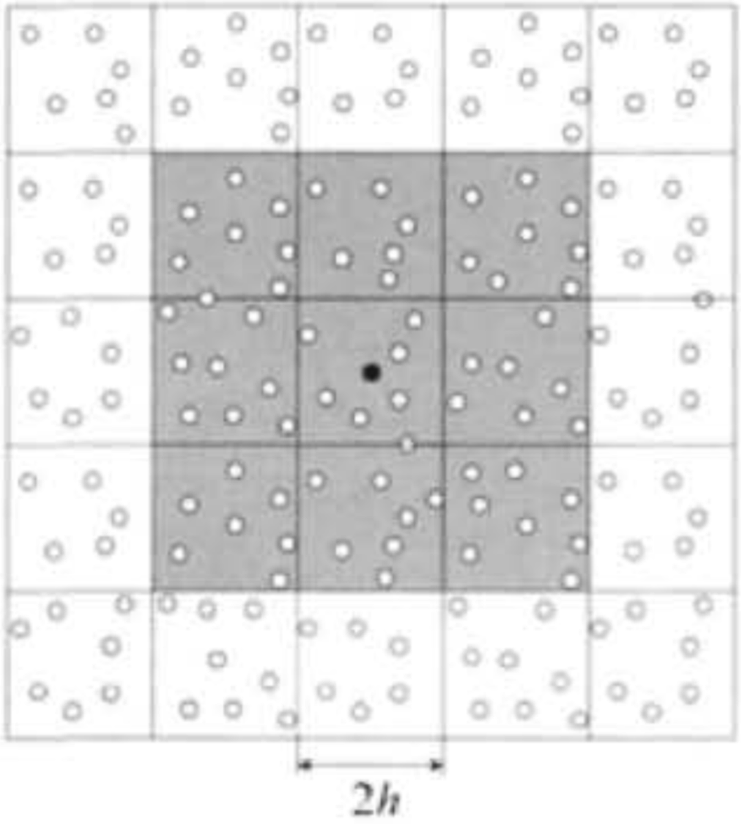


Figure 3.8: *Linked-list algorithm for searching the nearest neighbouring particles in a two-dimensional space* $(k = 2)$*[8].*

3. The Linked-list search algorithm is suitable for problems that involve a spatially constant smoothing length.

In the implementation of the Linked-list algorithm, a temporary mesh is overlaid on the problem domain, as shown in figure 3.8, and its spacing is selected to match the dimension of the support domain, equals to kh. Then for a given particle i, its nearest neighbouring particles can only be in the same grid cell or in those immediately adjacent to it. Thus, the search is confined only over 3, 9 or 27 cells for one-, two- or three-dimensional space, respectively, if $k = 2$. All the particles are assigned to cells and identified through linked-lists so the computational effort can be limited since the NNPS process is only necessary for a certain group of particles and if the average number of particles per cell is sufficiently small, the complexity of the Linked-list search algorithm is proportional to N.

For the EISPH model used in this thesis, the Linked-list search algorithm is used.

4 Applications

4.1 2D Dam-break

In order to evaluate the reliability of the EISPH scheme, a comparison with some established cases is needed. All of the following cases are a two-dimensional representation of a three-dimensional model obtained selecting a central portion of the domain and interpolating the values of the variables (as pressure, density, etc.) in the nearing of the selected area, as shown in figure 4.1.

All cases were run on a 2.6GHz 6-core 8th-generation Intel Core i7 processor with Turbo Boost up to 4.3GHz, 16GB 2400MHz DDR4 memory and macOS Mojave.

4.1.1 Comparison with a 2D EISPH model and Shao & Lo Results

The first step to understand if the algorithm works is to compare it with validated experiments or simulations. In order to do so, the simulations from Shao & Lo [5], involving a ISPH model of a Dam-Break, and the 2D EISPH from Nomeritae, E. Daly, S. Grimaldi, and H. H. Bui [3], where they compared their results to Shao & Lo's ones, will be exploited.

The model has an initial three-dimensional particle configuration of $a = 0,1$, $b = 0,1$ and $H = 0,2$ m respectively in x, y, and z (longitudinal, transverse and vertical) direction as shown in figure 4.1. The initial distance between particles has been chosen to be the same as the above-mentioned simulations: 0,005m, which leads to a total number of real particles equal to 16000. The timestep is equal to $1.0e10-4$ s in order to respect the CFL condition and the boundary is set to "free-slip" condition. To compare the three-dimensional wave with those two cases, the portion included in the interval $0.045 < y < 0.055$ has been taken has shown in figure 4.1, where the pressure distribution is hydrostatic and is calculated as:

$$p_i = \rho_i g(H - z_i) \quad [Pa], \tag{4.1}$$

where the subscription indicated the generic particle i, g is the gravity acceleration and ρ is the density. At the beginning of the computation, the fluid is kept in its position by a vertical wall (dam) which is instantaneously removed and the water is allowed to flow out along a dry horizontal bed. Figure 4.2 shows the particle configuration for different timesteps $t = 0,05,\ 0,10,\ 0,15$ and $0,18$s compared with the Shao & Lo and 2D-EISPH simulation. It can be seen how similar are the three simulations to each other; the only difference is in the leading edge of the Shao & Lo simulation because they used a no-slip boundary condition which leads to a slower wave.

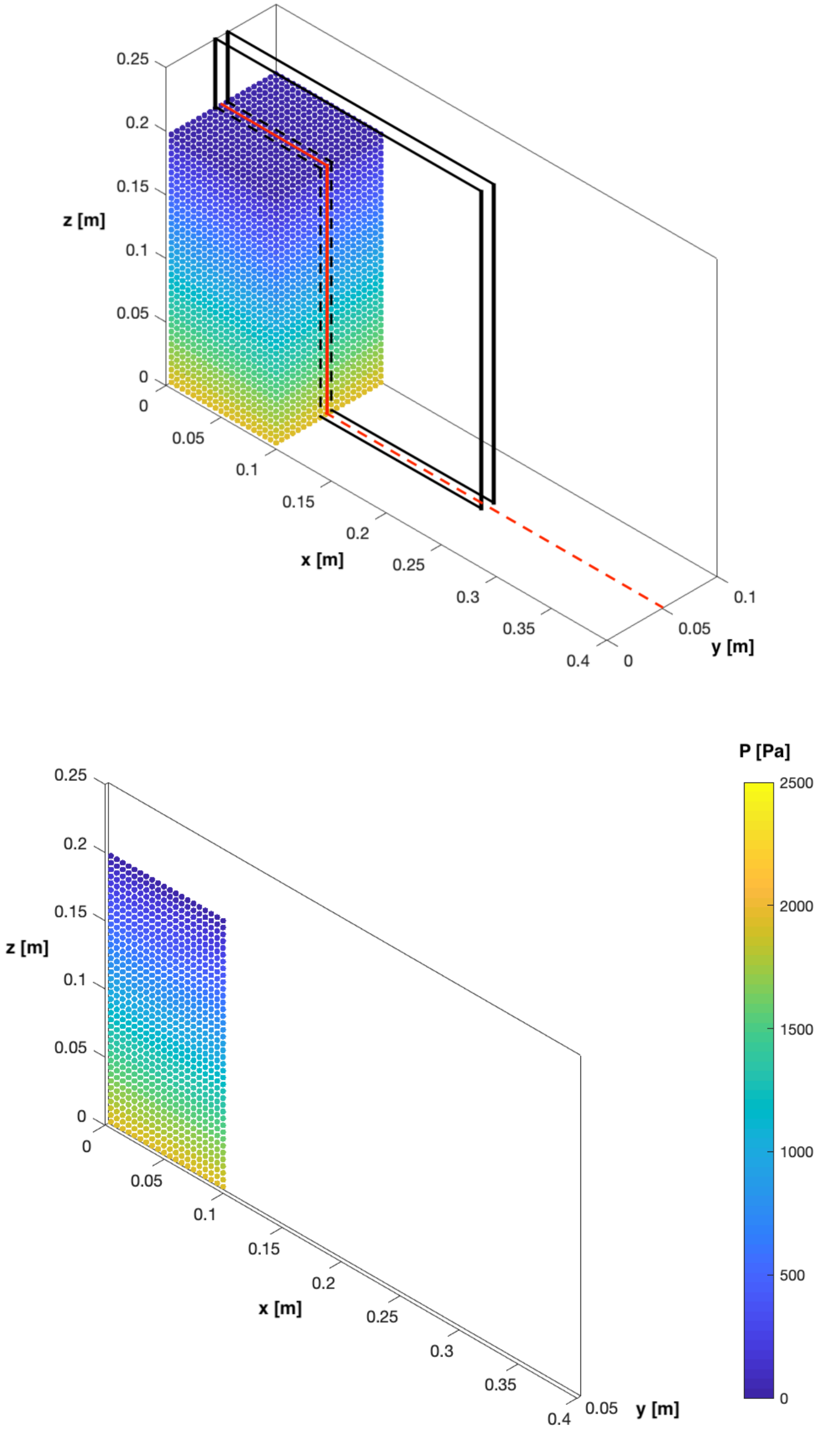


Figure 4.1: Initial spatial configuration: (above) Full domain, (below) Selected domain. The colorbar indicates the pressure distribution in [Pa].

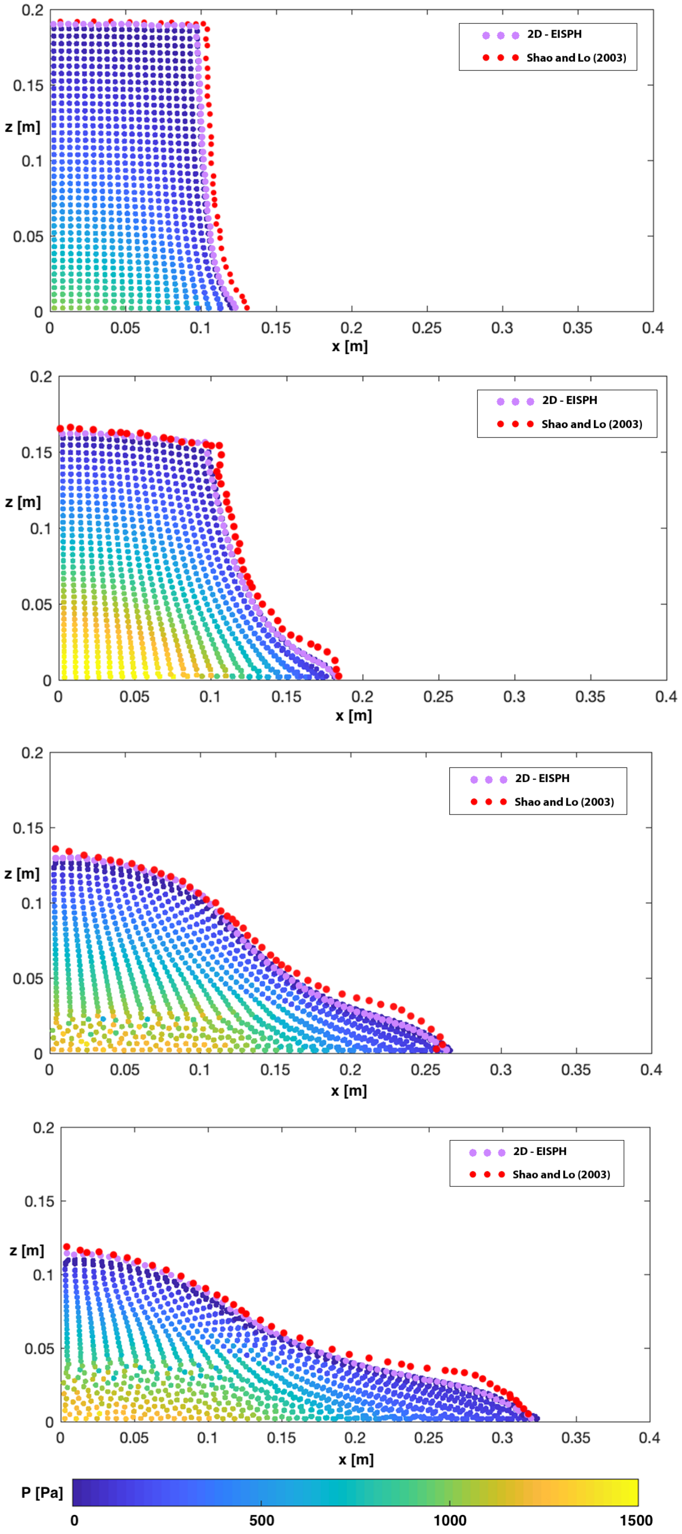


Figure 4.2: Particle configurations after collapse of dam at different times. Red dots correspond to Shao & Lo computations while purple dots correspond to the 2D-EISPH.

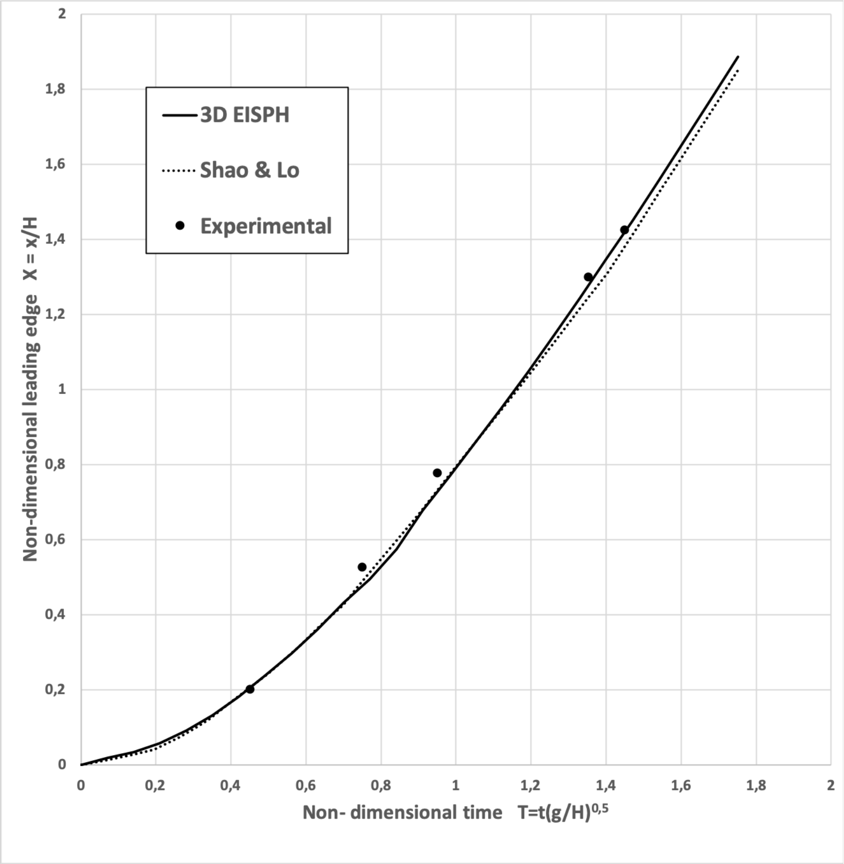


Figure 4.3: Relationship between the non-dimensional leading edge and time after dam-break.

The relation between the normalized time $T = t(g/H)^{0,5}$ and leading edge $X = x/H$ (note that the origin of X is defined at the dam site) graphed in figure 4.3, shows good agreement among data, in particular, the 3D-EISPH model seems approximate better the experimental data gave by the authors.

The pressure isobars of the dam-break flow for the 3D-EISPH and Shao & Lo model, at time $t = 0,05$ and $0,18$ s are given in figure 4.4 and 4.5. This figure shows that the 3D-EISPH simulation, in agreement with Shao & Lo' ones, has a sudden adjustment of the pressure field since the fluid is incompressible with infinite sound speed, caused by instantaneous removal of the dam. The pressure distribution inside the fluid deviates significantly from hydrostatic and the maximum pressure is about half of the initial hydrostatic value. At $t = 0,18$ s, the values of the pressure gradually increase, reaching a nearly hydrostatic distribution.

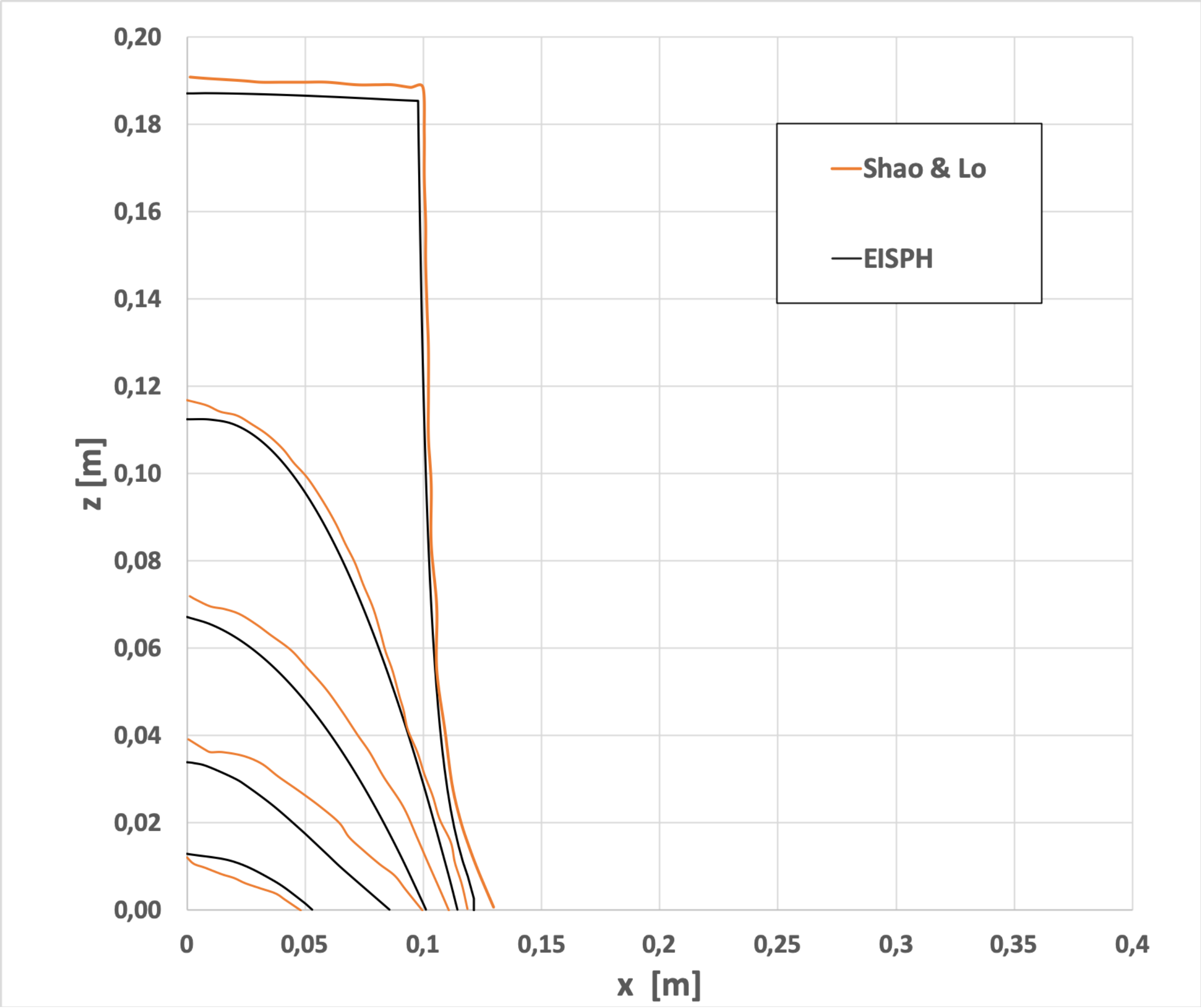


Figure 4.4: Pressure distribution comparison after collapse of dam at time $t = 0,05$s for 3D-EISPH (above) and Shao & Lo (below) simulations.

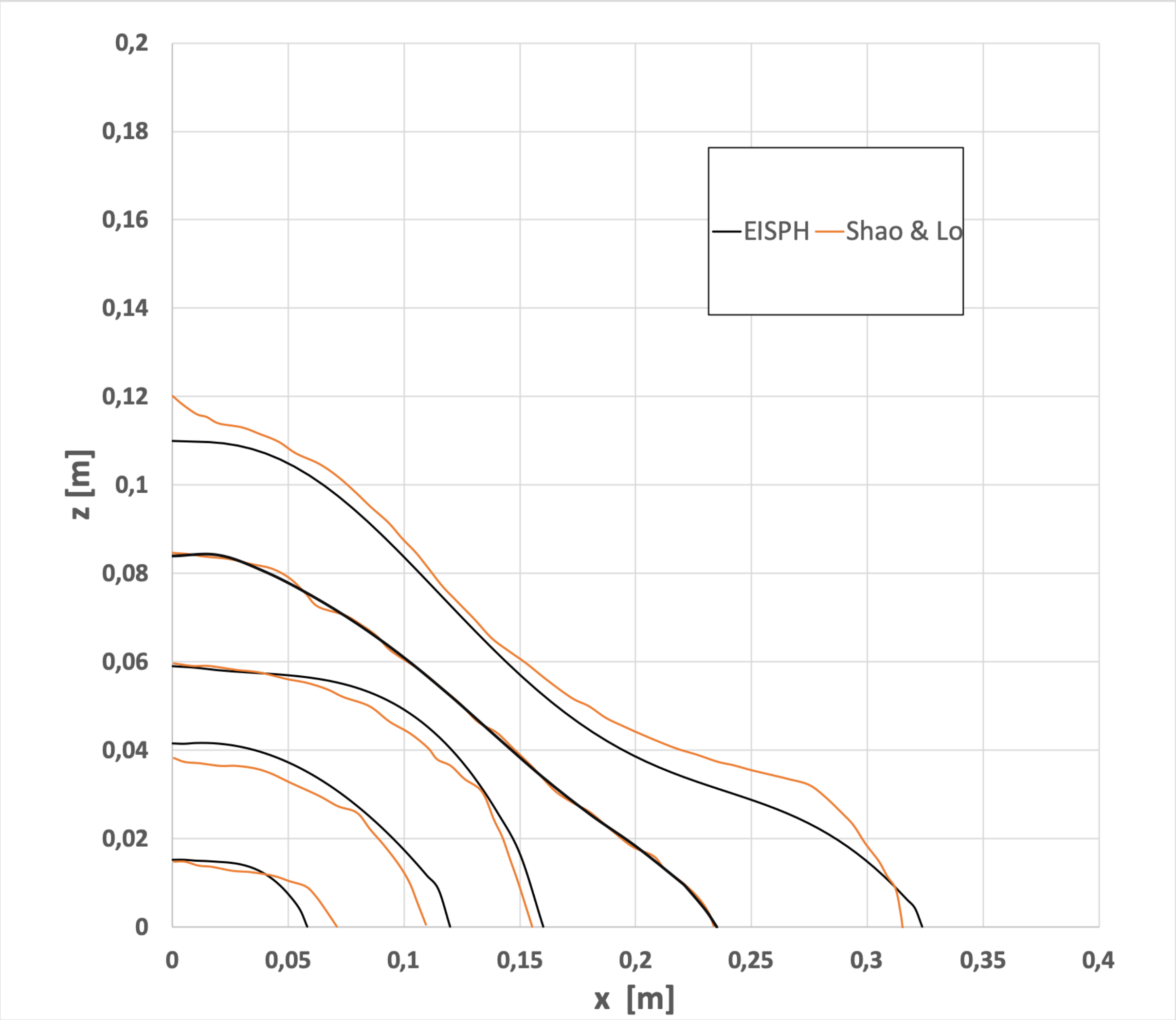
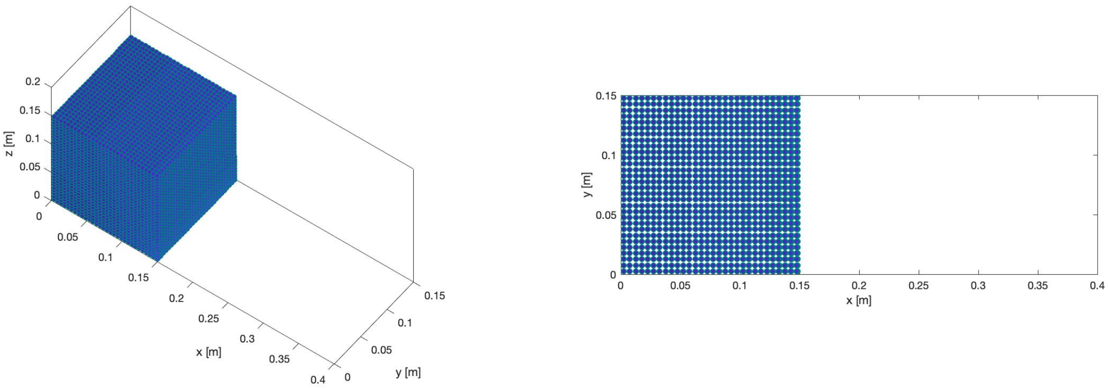


Figure 4.5: Pressure distribution comparison after collapse of dam at time $t = 0,05$s for 3D-EISPH (above) and Shao & Lo (below) simulations.

4.1.2 Comparison with Ritter's analytical solution

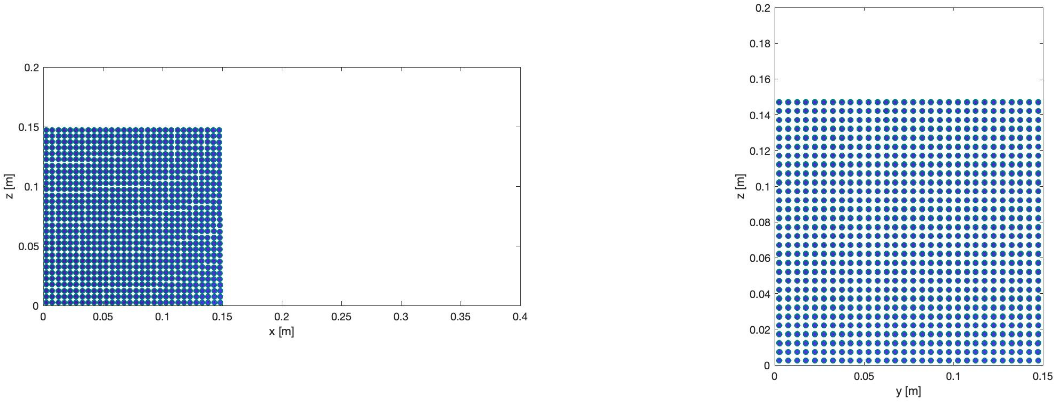
The Ritter's solution, though ideal, hence frictionless, is widely used to test numerical solutions. However, the real wave structure observed experimentally differs in a major portion of the wave profile, including the positive and negative fronts, therefore, this comparison serves only to have a qualitative idea about wavefront propagation.

This solution is compared with a simple case of 3D dam-break wave generated by the sudden release of a 0.15 m x 0.15 m volume of still water. The initial distance between particles is $\Delta x = \Delta y = \Delta z = 0.005m$, with a smoothing length, h, equal to $1.32\Delta x$; the time step is $10^{-4}s$. The initial particle distribution is shown in figure 4.6.



(a) Perspective view

(b) up view



(c) lateral view

(d) front view

Figure 4.6: initial distribution

As said before, in order to compare the model with the analytical solution, one needs to take a central portion of the model and interpolate the particles' velocity. The interpolation has been made by taking, for every particle i on the free surface, the average of the nearest particles j, as shown in the following equation

$$\vec{u}_i(x,y,z) = \frac{\sum_{j=0}^{N} \vec{u}_j(x,y,z)}{N} \quad for \quad 0.07 < y < 0.085m, \tag{4.2}$$

the particles on the free surface were selected looking for those that had a pressure equal to 0.

The results proposed below show how in the initial step the model differs from Ritter's solution; this is due to the frictionless condition that Ritter assumed in his formulations which makes the wave faster than the simulated one. Looking at figure 4.7, that shows the evolution of the wavefront celerity for both formulations over time, it can be easily seen how Ritter's wavefront celerity is constant with a value equal to $2,4m/s$ while the EISPH's one starts from zero and grows in a logarithmic way until it reaches a constant celerity equals to $2,25m/s$. For this reason, the wavefront position in the numerical model is distanced from the analytic one in proportion to the difference in speed between the two, which, decreasing over time up to a constant value of $2,4-2,25 = 0.15m/s$, causes this spacing to increase by about 15 cm every second. This is shown in figure 4.8 where the orange line represents the Ritter's wavefront solution and its linear over time because the first derivate (which is equal to the celerity) is constant and the blue one represents the model's solution, which tends to an oblique asymptote with an inclination corresponding to the regime celerity.

Another observation that must be made, concerns the wave profile. From the analytical equation for the wave water depth, equation 2.6, it can be seen that it has a parabolic shape, while the numerical wave has an inflection near the wavefront. Also this is due to the resistance present in the SPH model which tends to brakes the wave. A comparison between the two approaches is shown in figures 4.9 for different time steps: $t = 0,01s$, $t = 0,1s$ and $t = 0,2s$.

Looking at the evolution of the wave profile for both methods, another difference can be seen: in the Ritter's solution, the negative wave spread upstream with a constant celerity equal to $-\sqrt{gh_0} = -1,2m/s$ (where g is the gravity acceleration and h_0 is the initial water depth); the water upstream this negative wave keeps its original depth until it's reached by this one. This happens because of one the assumptions made by Ritter: apart from the frictionless condition, he assumed an infinite tank. This is why in the EISPH model the profile of the water who is still in the

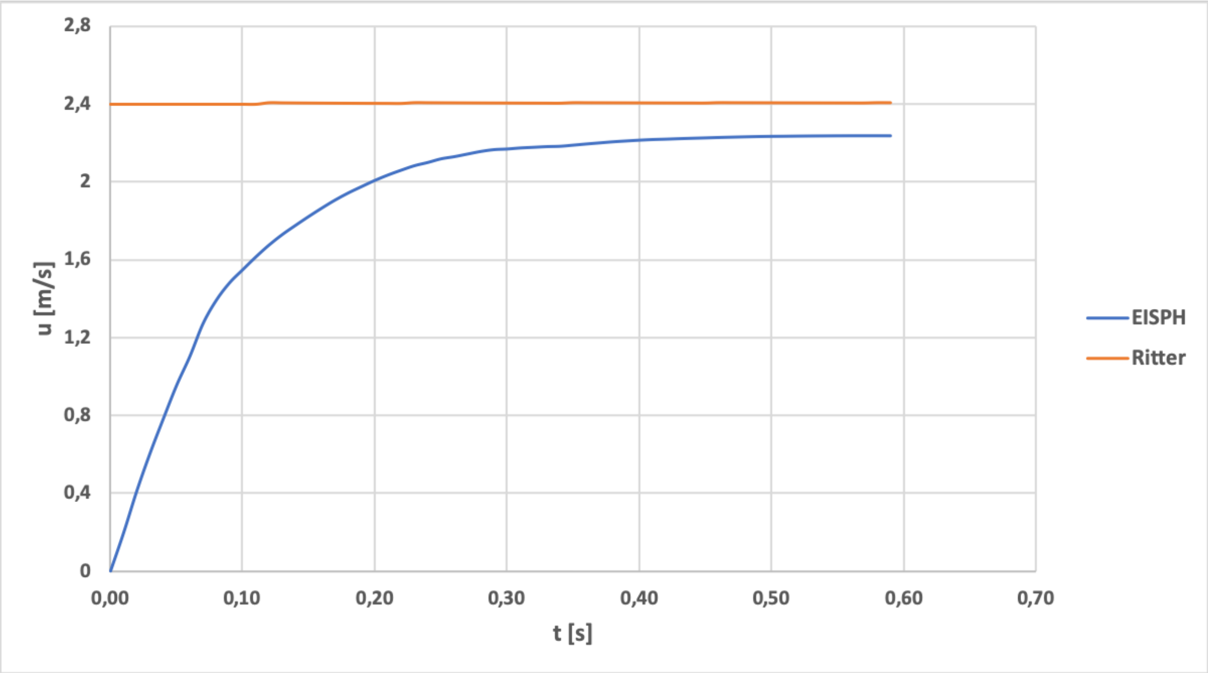


Figure 4.7: *Wavefront celerity evolution for Ritter's solution and EISPH model, over time*

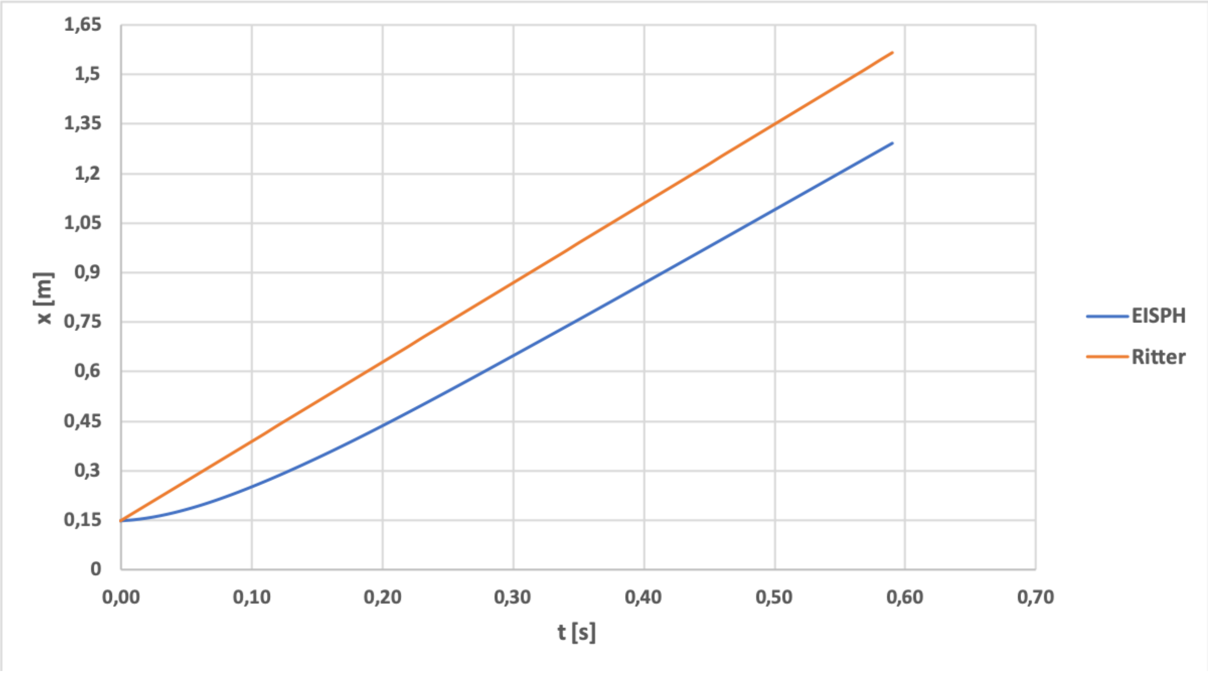


Figure 4.8: *Wavefront position evolution for Ritter's solution and EISPH model, over time*

tank, bend instead of keeping a constant deep. It is also impossible to have a discontinuity point on a real fluid, with a sudden change of direction.

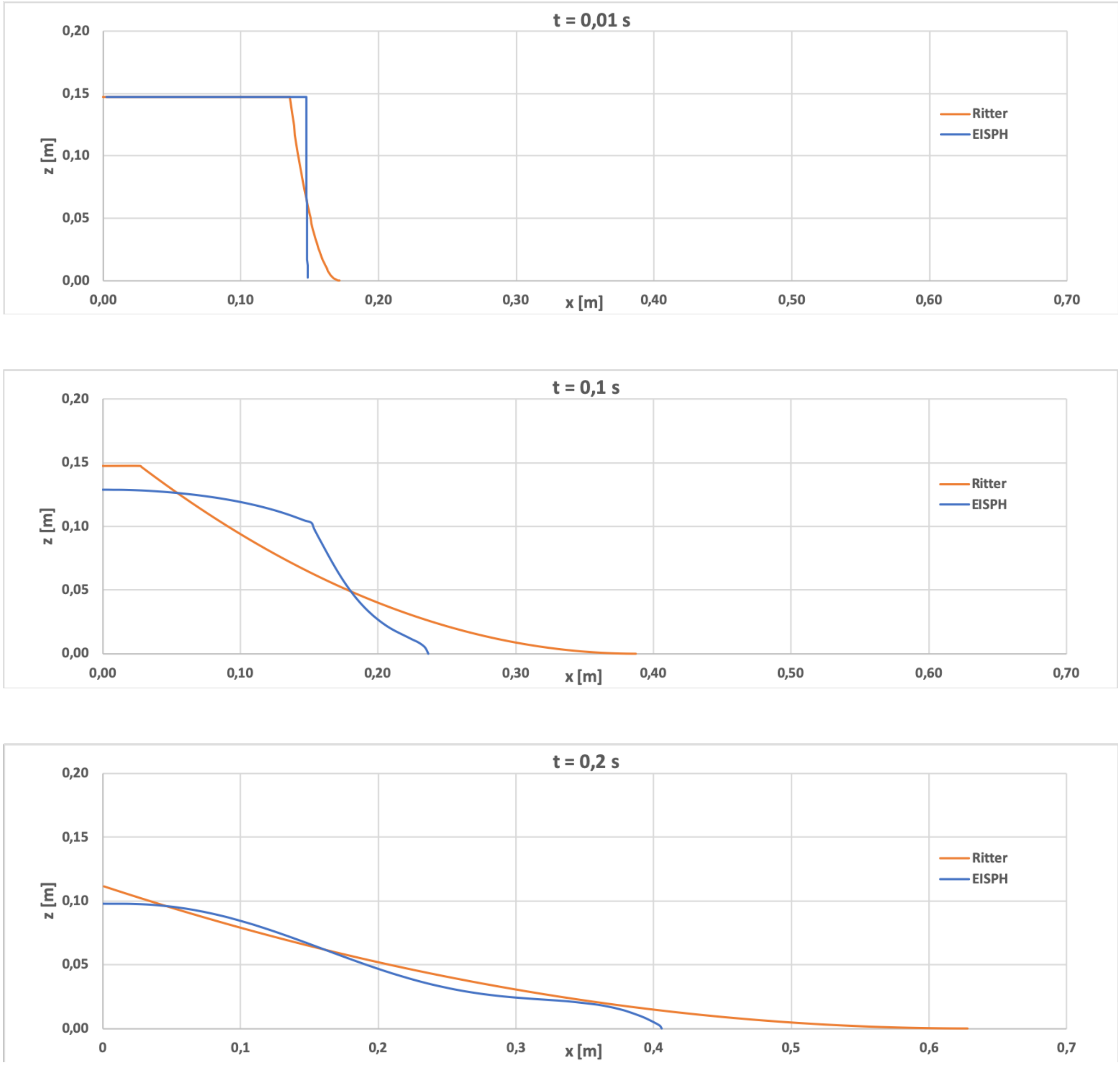


Figure 4.9: *Wave profile comparison between Ritter's solution and EISPH model, for* $t = 0,01s$*,* $t = 0,1s$ *and* $t = 0,2s$

The infinite tank condition has influence also on the water depth lowering speed, which is faster in the numerical model. This can be seen in figure 4.10 where the model and Ritter wave profile evolutions are shown. It's also useful to plot the water depth evolution over time for both approaches to have an easy look at how different is the loss of depth, this is shown in figure 4.11

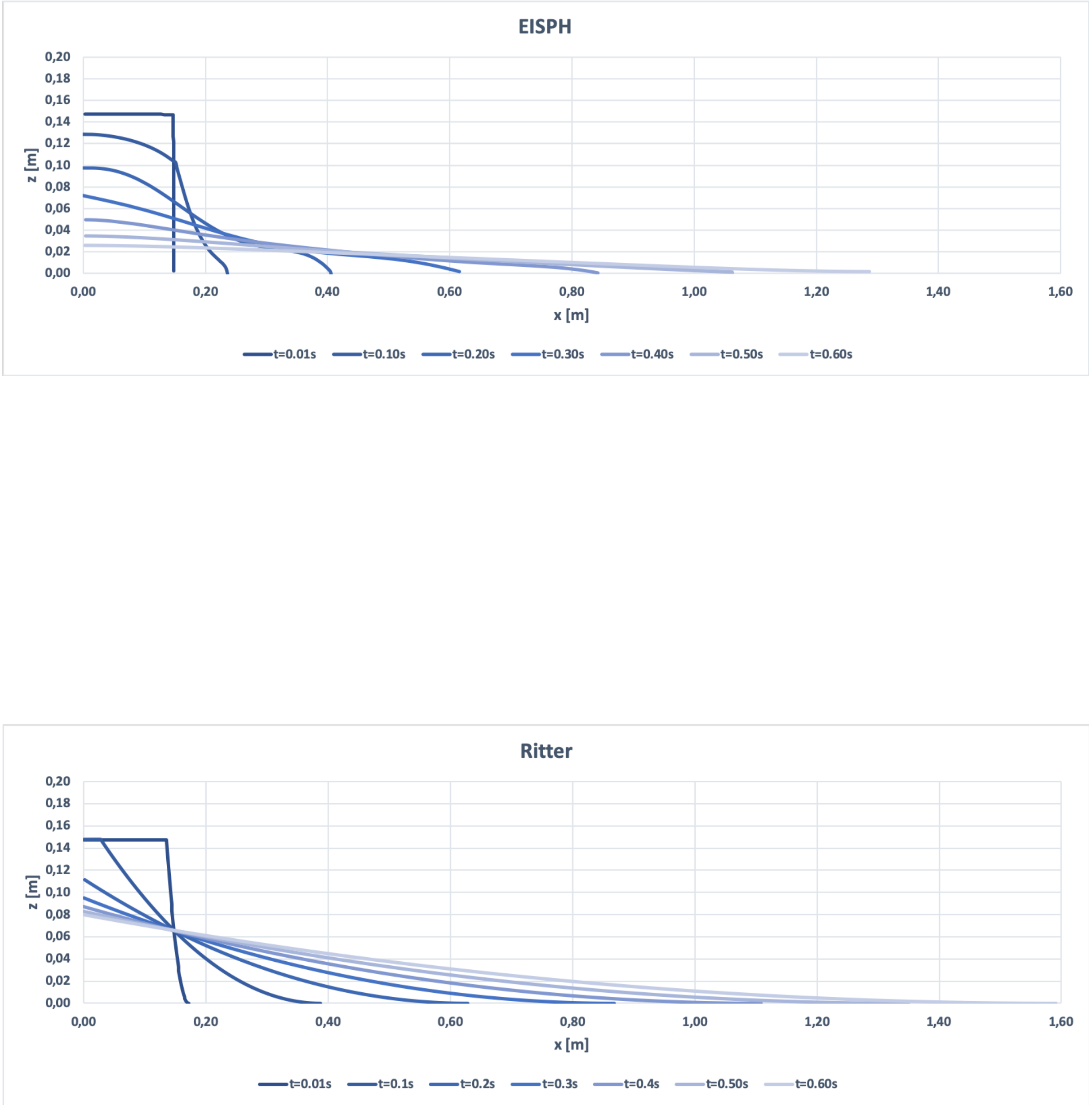


Figure 4.10: *Wave profile evolution for the EISPH model and the Ritter's solution, for different time steps*

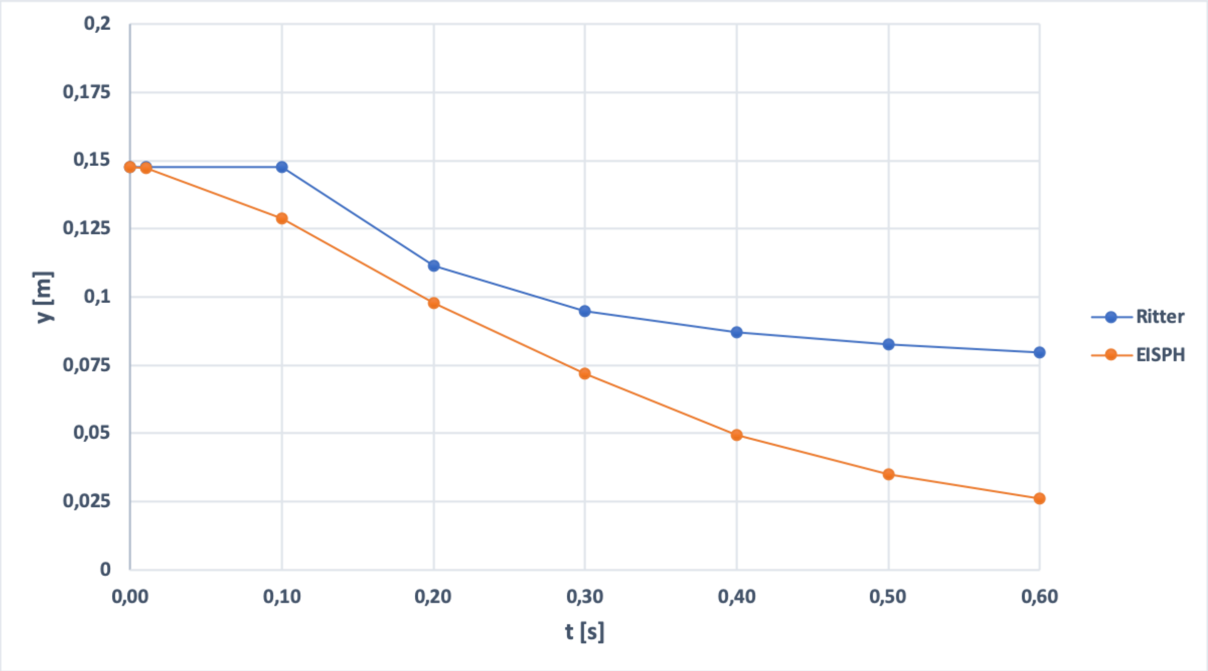


Figure 4.11: *Wave water depth evolution for the EISPH model and the Ritter's solution at the origin* $x = 0m$

4.2 3D Dam-Break: Comparison with F. Aureli, A. Maranzoni, P. Mignosa, and C. Ziveri experiment[1]

What has been done above served to validate the EISPH scheme and to verify its correct functioning, involving simple cases of Dam-Break without any obstacle and with linear geometry.

What one needs to understand now is if this numerical method is applicable to a real experiment, with non-linear geometry and the presence of obstacles. In order to do so, the experiment from F. Aureli, A. Maranzoni, P. Mignosa, and C. Ziveri[1] , presented in section 2.2.1, will be exploited.

The simulation consists of a rectangular tank divided between a reservoir and a floodable area by a gate representing the dam body, as in the real experiment. Two kinds of simulations were built: one with an initial particle distance of $0,01$ m, representing the full experiment domain and one with an initial particle distance of $0,005$ m representing only half of the experiment domain, given its symmetry, as explained further on.

All cases were run on a 2.6GHz 6-core 8th-generation Intel Core i7 processor with Turbo Boost up to 4.3GHz, 16GB 2400MHz DDR4 memory and macOS Mojave.

4.2.1 Full Domain: 144.000 real particles

The initial particle configuration is shown in figure 4.12 and consists in a $H = 0,15$m high, $a = 0,8$ m long and $b = 1,2$ m wide parallelepiped of real particles, simulating the water in the dam. Their initial distance, as written above, is constant in every direction with a value of $dx = dy = dx = 0,01$m leading to a total amount of real particles equal to $N = 144.000$. The boundary conditions on the wall of the tank are created using a combination of virtual particles equipped with a repulsive force and ghost particles, as explained in section 3.3.2, giving a free-slip condition. The distance between virtual particles is set to $dx/2 = 0,005$m on a single layer, while the ghost's one is identical to the real particle's one. The dam body that encloses the water is built with three layers of object particles with a distance of $dx/3 = 0,0033$m to ensure no penetration from the real particles. The dam height is $HD = 0,2$ m with a thickness of $c = 0,025$ m, and it presents a $B = 0,3$ m wide breach on the center. In front of this breach, within a distance of $i = 0,6$m, there is an obstacle with a parallelepiped shape, placed symmetrically with respect to the longitudinal axis, $d = 0,15$ m long, $e = 0,3$m wide and $HO = 0,25$ m high in order not be submersible, as in the original experiment. The obstacle is also made by object particles and it's placed in a floodable area $f = 1.775$ m long and $l = 1.2$ m wide.

Taking into account the CFL condition, the timesteps is set to $\Delta t = 1,0 \cdot 10^{-04}$ s. The amount of particle involved in this simulation is: 144.000 real particle, 185.000 virtual particles, and 41.392 object particles. Which leads to a total of 370.392 particles.

Figure 4.13 compares five snapshots of experimental and EISPH water depths in the floodable area at different times. It can be seen how the simulation replicates quite faithfully the shape of the dam-break wave in the experiment. In the first frame, the EISPH wave shape appears to be more wide open and rounded than the real one, but this can be attributed to the fact that the grey scale can only show water with a minimum depth of $0,005$ m, so if the fluid level is under this value, it is given the same color as the background. On the other hand, the particles in the numerical model have a diameter of $0,01$ m so there isn't an erasing due to the depth.

In the second frame, after the impact on the obstacle, a horseshoe-shaped hydraulic jump is formed in the model as in the experiment. A noteworthy fact, which remains valid for the rest of the simulation, is that the angle that forms between the body of the dam and the jet from the breach in the simulation and the one in the experiment are very similar; the same thing for the angle between obstacle longitudinal face and the water. This can be more appreciated in figure 4.14 where the snapshots are overlapped, step by step.

In the third step, corresponding to $t = 1,45$s, is clearly visible the formation of lateral hydraulic jumps due to the impact of the particles on the walls.

In the fourth one, the particles that have reached the downstream wall, forme a jump that moves upstream and, in the last frame, the particles reach the dam wall and they starts two symmetrical vortexes in that area.

Given this, there is a consideration to be made: although visually the images overlap well, this remains a completely qualitative verification, as it is difficult to understand well what is the difference in height between the experiment and simulation. The next step is to measure quantitatively how much the model differs from reality and to do this, the height measurements taken by the authors thanks to an ultrasonic transducer placed at a known measuring point (see section 2.2.1) will be exploited. Since the available height data belong to point particles, it becomes almost impossible to determine the maximum height of a given point in space, because there should always be at least one particle at that point. To overcome this problem, the simplest and most

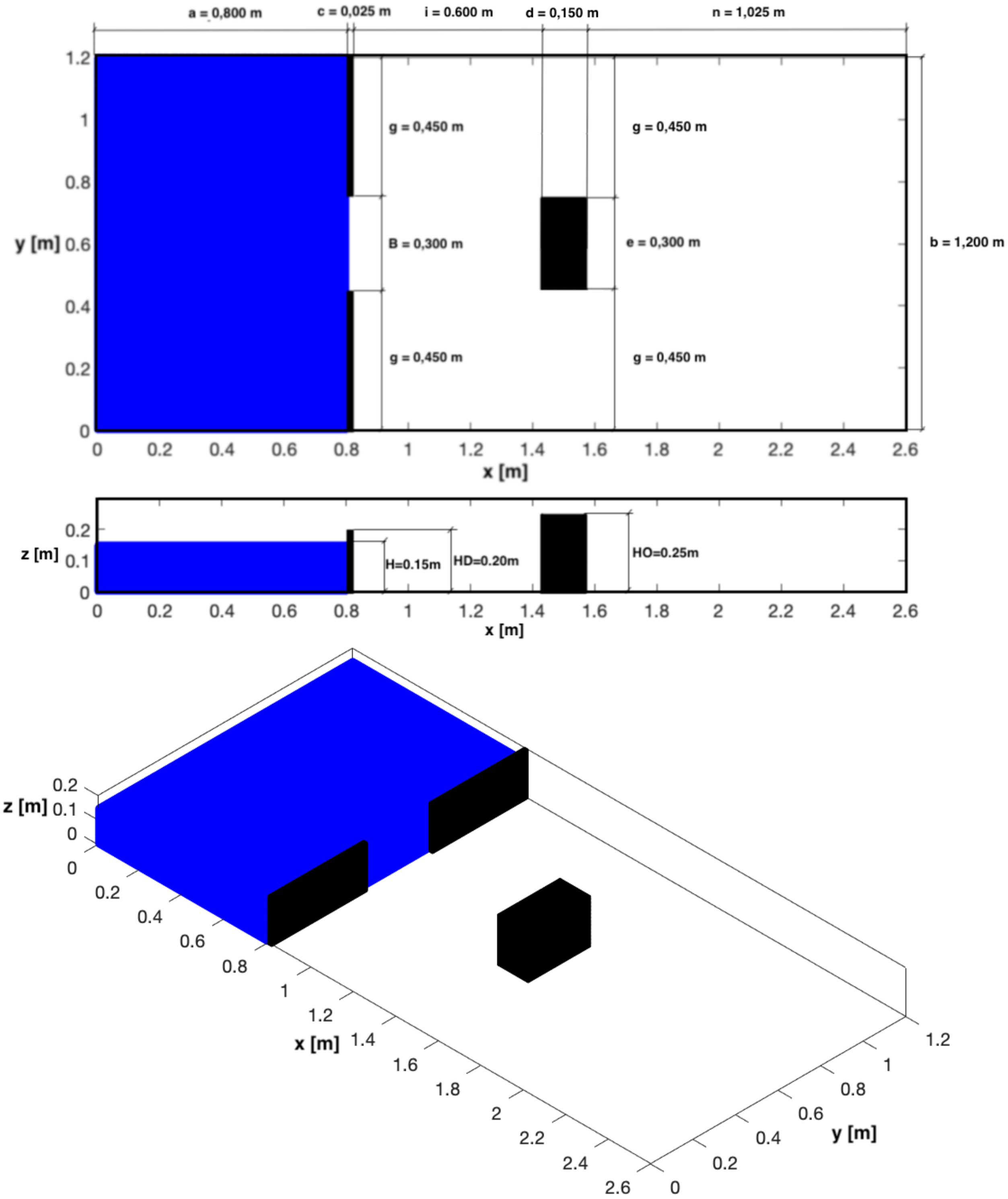


Figure 4.12: Simulation facility scheme and initial particle configuration, with main dimension

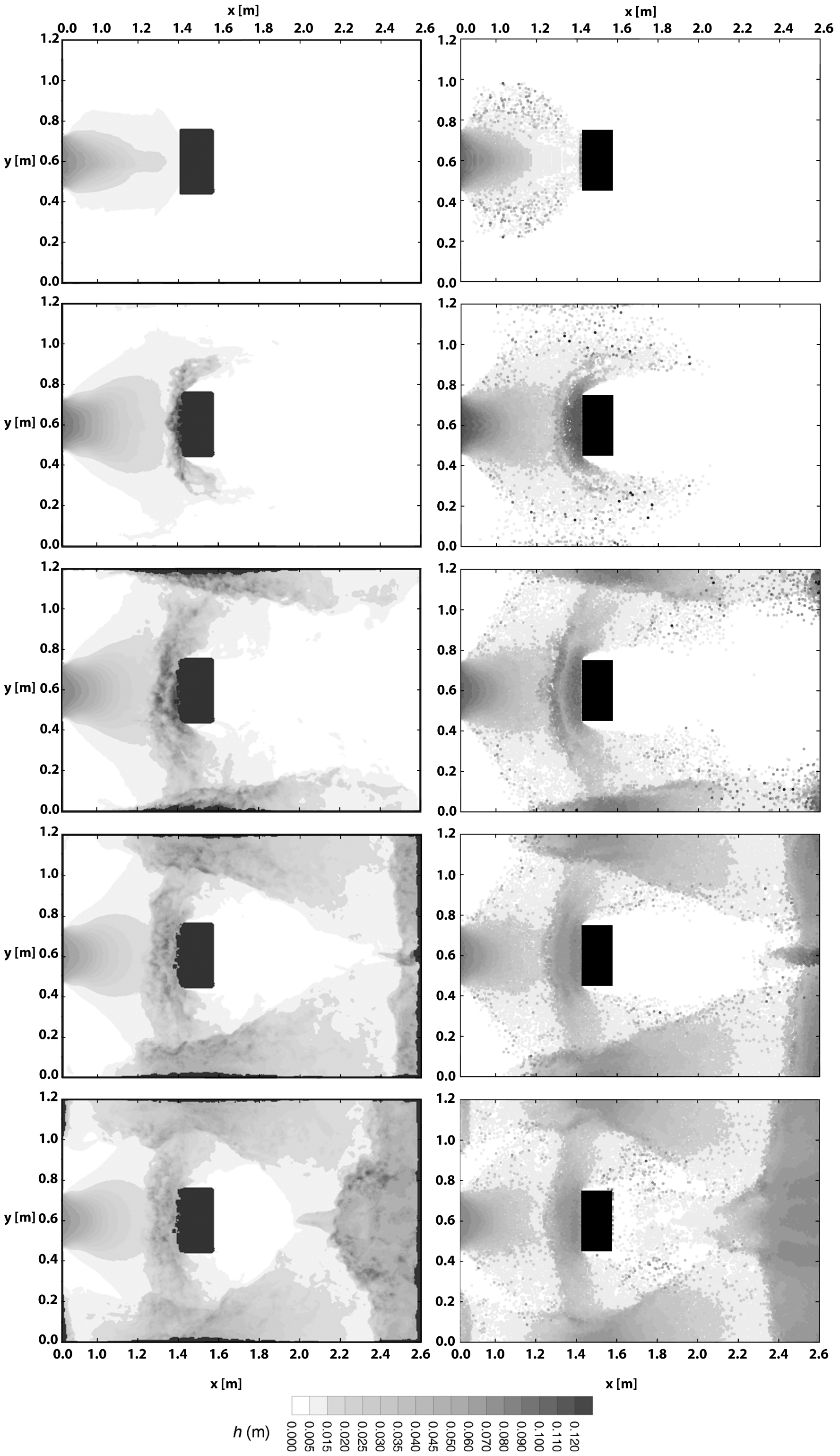


Figure 4.13: Comparison between experimental (left) and EISPH simulation (right) at time $t = 0,40$ s; $t = 0,75$ s; $t = 1,45$ s; $t = 2,16$ s; $t = 2,86$ s. Full domain: 144.000 real particles.

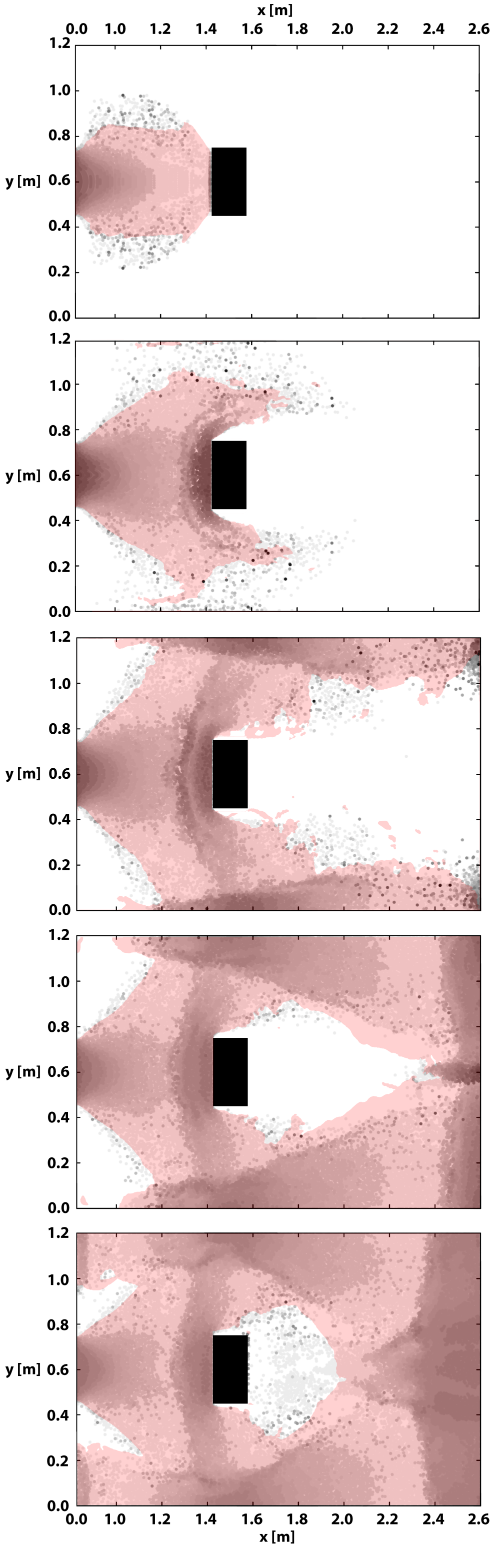


Figure 4.14: Overlapped snapshot at time $t = 0,40$ s; $t = 0,75$ s; $t = 1,45$ s; $t = 2,16$ s; $t = 2,86$ s. Red area is the experiment water profile. Full domain: 144.000 real particles.

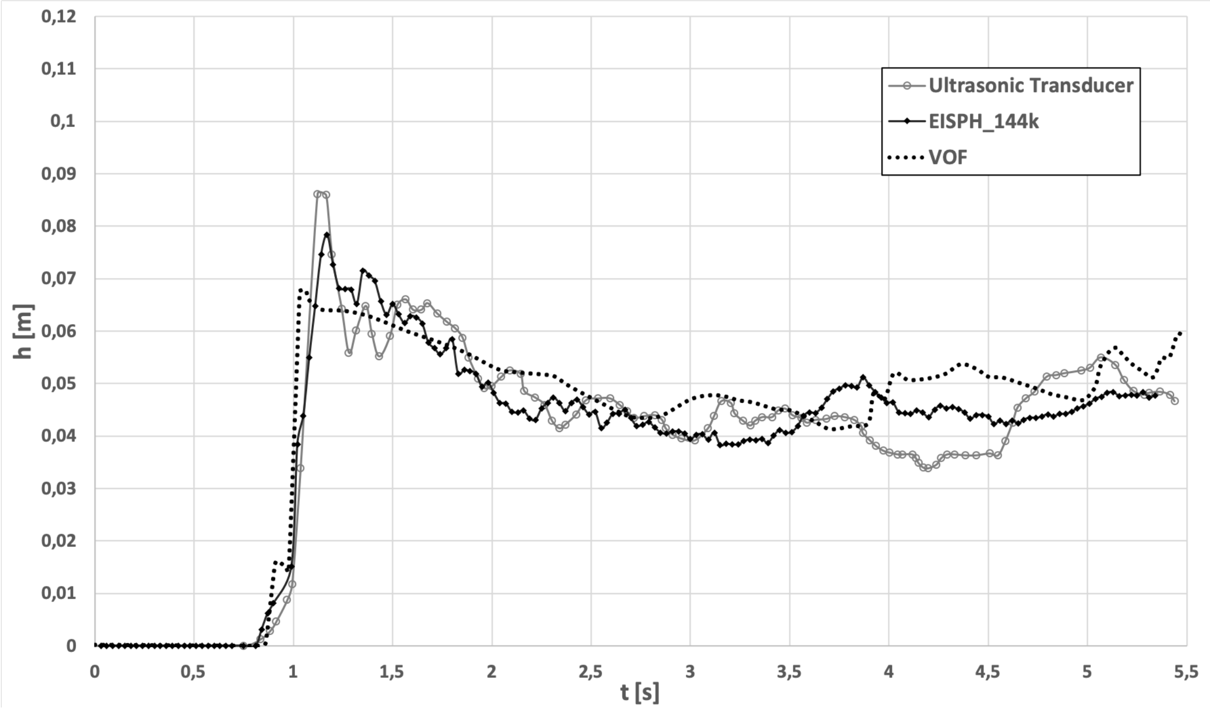


Figure 4.15: Water level time series for EISPH simulation and ultrasonic transducer at $x = 1,674$ and $y = 0,022$ m. Full domain: 144.000 real particles.

effective solution is to refer to a portion of a control volume around the point of interest, calibrated in order to always have at least one particle inside it and then recording, for each time step, the greatest height among the particles that occur in that space. For this simulation, the so calibrated control volume is a parallelepiped with the same x and y side length of $1,5dx = 0,015$ m and a height equal to $0,2$ m, pointed at the gauge point used by the authors located at $x = 1,647$ m $0,022$m. The acquired heights are shown in figure 4.15 in comparison with those obtained with the ultrasonic transducer. It can be seen how the two are similar most of the time. The biggest difference is in the last two seconds of simulation, where they seem to be in opposition, but then, in the final, they lead back to the same value. The same comparison is made also with the results obtained from a VOF numerical modelling at the same gauge point in [1]. This one does not have a peak at the beginning while in the last part of the series is much higher than the EISPH end experimental values; it is also more linear, without any oscillation.

This last graphic gives an idea of how works the simulation but is still a punctual evaluation. In order to have an overall measurement of the difference between the two, this evaluation must be extended to the whole domain. Here, however, a problem arises: the data provided by the authors are limited to those seen so far, ie the frames and water heights measured with the ultrasonic transducer at the gauge point. But, considering the grayscale images of the experiment, one can think of a way to extend the height measurement to the whole domain. In fact, building a grid of points on the images for each time step and measuring the RGB value (which, being in shades of grey, R,B, and G values will be equal to each other) for each point of this grid, one can convert the latter into water height, using the greyscale-water depth transformation function provided by the authors, shown in figure 2.4. Then, by applying the same grid to the numerical simulation, one can obtain the measurements of the height of the particles in the same points (or better, regions), as done for the gauge point. Operationally speaking, the process is inverse: the grid is built starting from the numerical model because it must have cells large enough to always have particles inside it (obviously for the simulation, the grid is three-dimensional, since the component z is also present). A cell size identical to that used for the comparison with the transducer was chosen, with the side equal to $lx = ly = 1,5dx = 0,015$ m, therefore of a square shape, marked in the figure 4.16. The maximum height value of the particles is calculated for each cell, in the instant investigated, and is saved in a matrix with the same dimensions as the grid. After that, the grid is applied to the frame corresponding to the same instant and the RGB value corresponding to the center of gravity of the cell is taken, saving it in a matrix as before. Here a clarification is needed: inside the frames, there are darker areas near the edges and the obstacle, which, as described by the authors [1], are due to the presence of air bubbles trapped inside the water, that gives rise to anomalous light reflections. Therefore in these areas the water depth value, obtained with the technique described above, is inappropriate. To overcome this problem, in those areas a value of water depth equal to that of the nearest water with a correct grey value is assigned, increased by 10%. Another problem due to these dark areas is that their RGB = [25,25,25] is identical to the obstacle one. In order not to consider the latter in the calculations, given that in that area there can be no particles, a green rectangle is superimposed on the obstacle in the frames so as to have values of R, G and B different from each other (can be used any color that is not a grey), thus allowing to impose a simple condition for not considering the obstacle and imposing a value of water height equals to zero in that area. The overlap of the grid and the green rectangle to a general frame is shown in figure 4.17.

At this point the difference between the two matrices is made, thus calculating the deviation between the numerical model and the experiment. Gathering together the differences for the time steps investigated $t = 0.40$ s, $t = 0.75$, $t = 1.45$ s, $t = 2.16$ and $t = 2.86$, $47,200$ values are obtained which, divided into 60 classes (the same number used by the authors in their analysis) with a constant interval equal to the difference between the maximum value and the minimum value divided by 60, allow to realize a frequency distribution that is useful to evaluate quantitatively as the simulation approaches the real model.

Since the application of the grid to the frames is achieved by means of the MatLab® *grab function*, in which the user points to the extremes of the figure defining the geometric coordinates of the domain, it is understandable how this method is susceptible to the precision of the user himself. However, as is shown in table 1, repeating several times the process, the results are very similar, with a small variance, so it can be concluded that the used method has a good accuracy and the influence of the users precision doesn't affect very much the final distribution (obviously the user must choose carefully the extremes of the snapshots).

The mediated results of this comparison are shown in table 3 where the classes are represented by the mean value of the classes interval deviation, while the absolute, relative and cumulative frequency distribution are graphed at figure 4.18. The main mediated statistics value of the deviation $h_{SPH} - h_{exp}$ are shown in table 2.

Table 1: Variance of the results obtained repeating several time the process.

	1st Rep	**2nd Rep**	**3rd Rep**	**4th Rep**	**Standard deviation [mm]**
Max [mm]	$+111,12$	$+111,22$	$+111,16$	$+111,14$	$0,037$
Min [mm]	$-130,00$	$-130,00$	$-130,00$	$-130,00$	$0,000$
Mean [mm]	$-2,16$	$-2,13$	$-2,15$	$-2,15$	$0,011$
Median [mm]	$-1,69$	$-1,68$	$-1,66$	$-1,70$	$0,015$
15th perc. [mm]	$-18,43$	$-18,41$	$-18,44$	$-18,41$	$0,013$
85th perc. [mm]	$+8,40$	$+8,38$	$+8,41$	$+8,40$	$0,011$

Table 2: Main statistics value of the deviation $h_{SPH} - h_{exp}$

Max	$+111,16$	[mm]
Min	$-130,00$	[mm]
Mean	$-2,15$	[mm]
Median	$-1,68$	[mm]
15th perc.	$-18,42$	[mm]
85th perc.	$+8,40$	[mm]

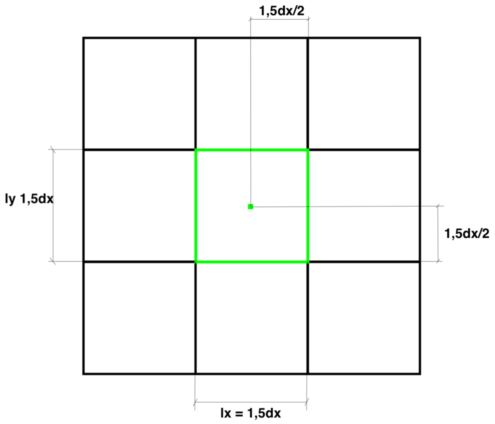


Figure 4.16: Grid cell main dimensions.

Table 3: Absolute, Relative and Cumulative frequency distribution for every $h_{SPH} - h_{exp}$ classes

$h_{SPH} - h_{exp}$	$h_{SPH} - h_{exp}$	Absolute Frequency	Relative Frequency	Cumulative Frequency
[m]	[mm]	[-]	[%]	[-]
-0,12792	-127,92	73	0,00155	0,00155
-0,12377	-123,77	0	0,00000	0,00155
-0,11962	-119,62	156	0,00331	0,00485
-0,11547	-115,47	0	0,00000	0,00485
-0,11132	-111,32	22	0,00047	0,00532
-0,10717	-107,17	3	0,00006	0,00538
-0,10302	-103,02	0	0,00000	0,00538
-0,09887	-98,87	38	0,00081	0,00619
-0,09472	-94,72	1	0,00002	0,00621
-0,09057	-90,57	42	0,00089	0,00710
-0,08641	-86,41	12	0,00025	0,00735
-0,08226	-82,26	22	0,00047	0,00782

-0,07811	-78,11	129	0,00273	0,01055
-0,07396	-73,96	36	0,00076	0,01131
-0,06981	-69,81	191	0,00405	0,01536
-0,06566	-65,66	19	0,00040	0,01576
-0,06151	-61,51	415	0,00879	0,02456
-0,05736	-57,36	18	0,00038	0,02494
-0,05321	-53,21	8	0,00017	0,02511
-0,04906	-49,06	824	0,01746	0,04256
-0,04490	-44,90	652	0,01381	0,05638
-0,04075	-40,75	528	0,01119	0,06756
-0,03660	-36,60	437	0,00926	0,07682
-0,03245	-32,45	58	0,00123	0,07805
-0,02830	-28,30	653	0,01383	0,09189
-0,02415	-24,15	817	0,01731	0,10919
-0,02000	-20,00	1184	0,02508	0,13428
-0,01585	-15,85	1964	0,04161	0,17589
-0,01170	-11,70	403	0,00854	0,18443
-0,00755	-7,55	758	0,01606	0,20049
-0,00339	-3,39	5164	0,10941	0,30989
0,00076	0,76	21838	0,46267	0,77256
0,00491	4,91	2094	0,04436	0,81693
0,00906	9,06	1857	0,03934	0,85627
0,01321	13,21	1034	0,02191	0,87818
0,01736	17,36	920	0,01949	0,89767
0,02151	21,51	884	0,01873	0,91640
0,02566	25,66	847	0,01794	0,93434
0,02981	29,81	620	0,01314	0,94748
0,03396	33,96	459	0,00972	0,95720
0,03812	38,12	459	0,00972	0,96693

0,04227	42,27	599	0,01269	0,97962
0,04642	46,42	321	0,00680	0,98642
0,05057	50,57	161	0,00341	0,98983
0,05472	54,72	134	0,00284	0,99267
0,05887	58,87	118	0,00250	0,99517
0,06302	63,02	57	0,00121	0,99638
0,06717	67,17	50	0,00106	0,99744
0,07132	71,32	29	0,00061	0,99805
0,07547	75,47	31	0,00066	0,99871
0,07963	79,63	30	0,00064	0,99934
0,08378	83,78	12	0,00025	0,99960
0,08793	87,93	9	0,00019	0,99979
0,09208	92,08	3	0,00006	0,99985
0,09623	96,23	3	0,00006	0,99992
0,10038	100,38	0	0,00000	0,99992
0,10453	104,53	3	0,00006	0,99998
0,10868	108,68	0	0,00000	0,99998
0,11283	112,83	1	0,00002	1,00000
0,11698	116,98	0	0,00000	1,00000

As it can be seen from the frequency distribution graph and table, most of the deviation values are located around $0,76$ mm, which mean that the EISPH model and the experiment are quite similar to each other. Furthermore, the left branch of the distribution is a little discontinuous and has some peaks around -12 mm, due to residual edge effects. The right branch, on the other hand, is smoother and is monotonically decreasing because, since the model is not represented by RGB values, the water depth is more certain.

As said above, and it can also be seen in figure 4.13, the areas in the experiment, where the water level is under $0,005$ m, aren't represented. However, some grey areas detached from the main flow are recognizable; this means the area interested by the presence of water is larger than the

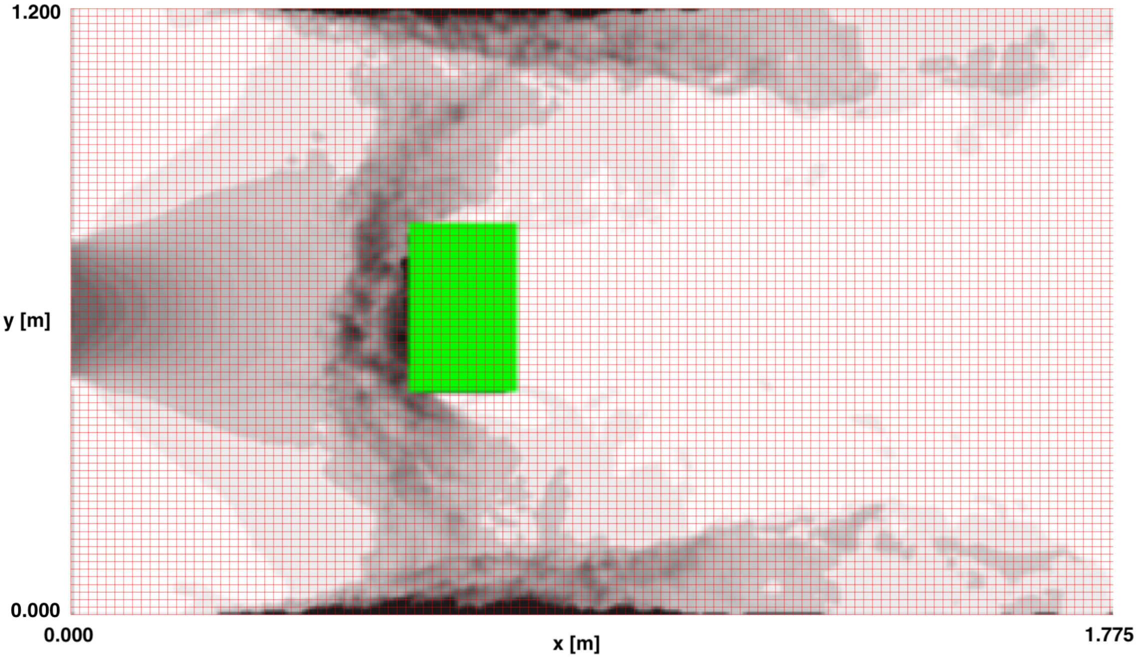


Figure 4.17: Interpolation grid - Full domain.

one represented by the photograms. In the EISPH simulation, on the contrary, every particle is represented, since their diameter is greater than $0,005$m. This means that the particle diameter chosen for this simulation is insufficient to represent all the water in the domain, which leads to the next paragraph.

4.2.2 Half Domain: 576.000 real particles

In order to increase the model resolution, the particle diameter has been halved to the value of $0,005$ m. This means that the number of real particles presents in the domain becomes eight times bigger than the previous simulation, for a total of 1.152.000 real particles. Since this algorithm is not parallelized, so all the simulation is computed by just one CPU core, the time required to perform the entire simulation would be very large. Here comes in help the symmetry of the problem; how it can be seen from the previous figures, the experiment domain is totally symmetrical, and so it is the dam-break wave cause by the opening of the breach. So, instead of representing the entire tank, this simulation si performed for only half of the domain.

In order to have right results, is really important the choice of the boundary conditions at the

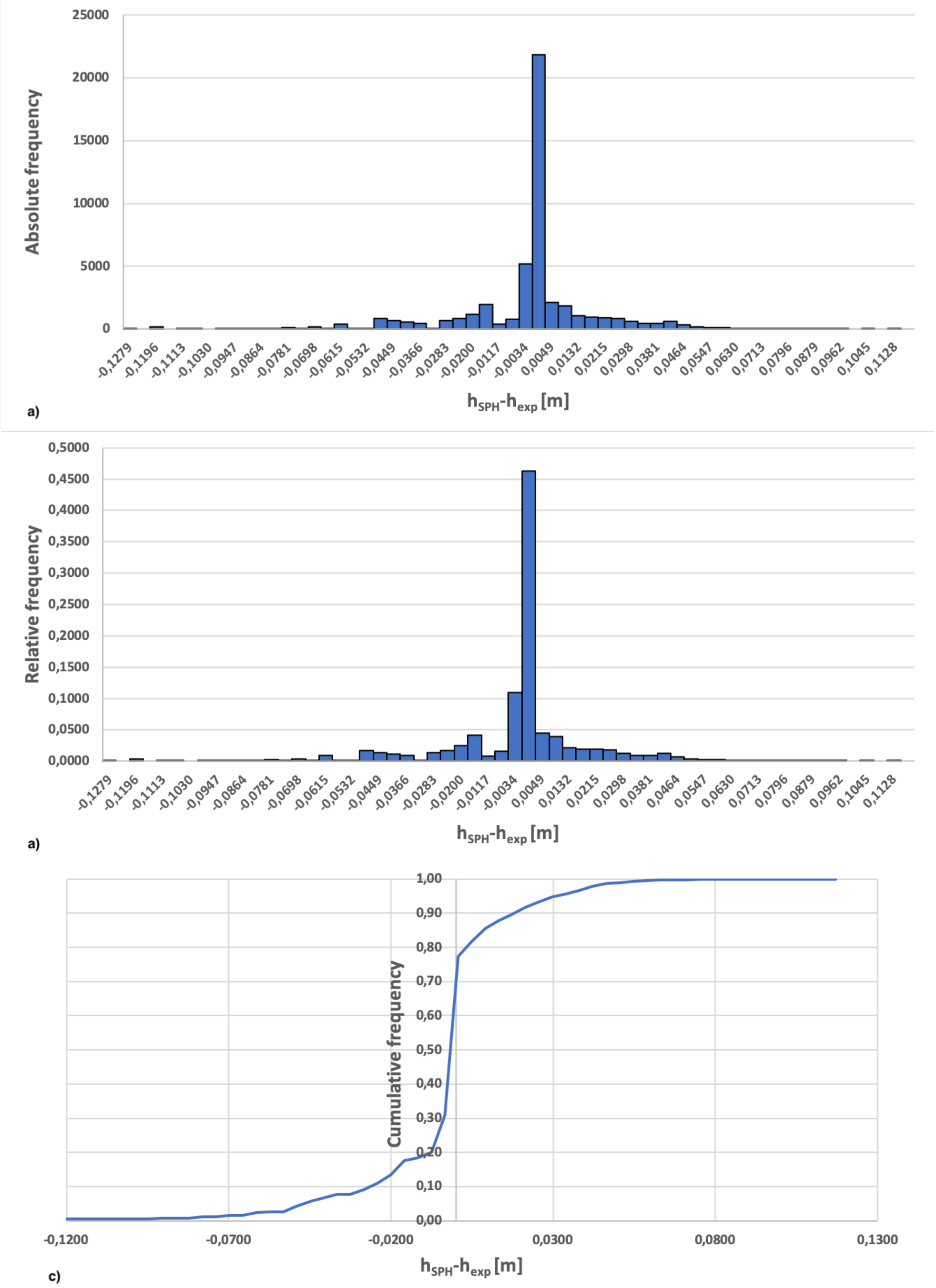


Figure 4.18: Absolute (a), Relative (b) and Cumulative (c) frequency distribution of the deviation $h_{SPH} - h_{exp}$ between numerical and experiment.Full domain: 144k real particles

symmetry wall. There are two ways to create this boundary: using only object particles or using virtual particles equipped with the repulsive force plus ghost particles with a free-slip condition. The former appear to give better results but, considering the dimensions of the tank, this approach would require a very large computational time, which would make the division of domain useless. The latter gives less weight on computing time, but it gives also problem near the corners: the anti-penetration action in that zone, works only with the presence of ghost particles that act with a direction of 45 degrees to the sides (figure 4.19). This means that the real particles near the corners suffer from a force that pushes them towards the inside of the domain (which does not happen at the center of the whole domain case). An example of this phenomenon is shown in figures 4.20b and 4.21, where a full-domain case is represented by half, with the extreme longitudinal axis of symmetry superimposed and a case of half-domain of the same simulation. The grey particles are the ghost ones, the black one are the virtual ones and the remain particles are the real ones, represented by a color proportional to the transversal velocity v_y. It can be seen that in the half-domain case, the real particles near the up edge are moving toward the inside of the domain, while in the full-domain case, they move tangentially to the x axis. However, over time, the difference between this two simulations become smaller, and is almost eliminated if the tank dimension and the number of particles increase, because the influence of the ghost particles affects a much smaller region respect the size of the domain for this reason, this approach is adopted to built the simulation.

The initial particle configuration for the half-domain simulation is shown in figure 4.22 and consists in a $H = 0,15$m high, $a = 0,8$ m long and $b = 0,6$ m wide parallelepiped of real particles, simulating the water in the dam. Their initial distance, as written above, is constant in every direction with a value of $dx = dy = dx = 0,005$m leading to a total amount of real particles equal to $N = 576.000$. The boundary conditions on the wall of the tank are created using a combination of virtual particles equipped with a repulsive force and ghost particles, as explained in section 3.3.2, giving a free-slip condition. The distance between virtual particles is set to $dx/2 = 0,0025$m on a single layer, while the ghost's one is identical to the real particle's one. The dam body that encloses the water is built with five layers of object particles with a distance of $dx/3 = 0,0017$m to ensure no penetration from the real particles. The dam height is $HD = 0,2$ m with a thickness of $c = 0,025$ m, and it presents a $B = 0,15$ m wide breach on the center. In front of this breach, within a distance of $i = 0,6$m, there is an obstacle with a parallelepiped shape, placed at the edge of the

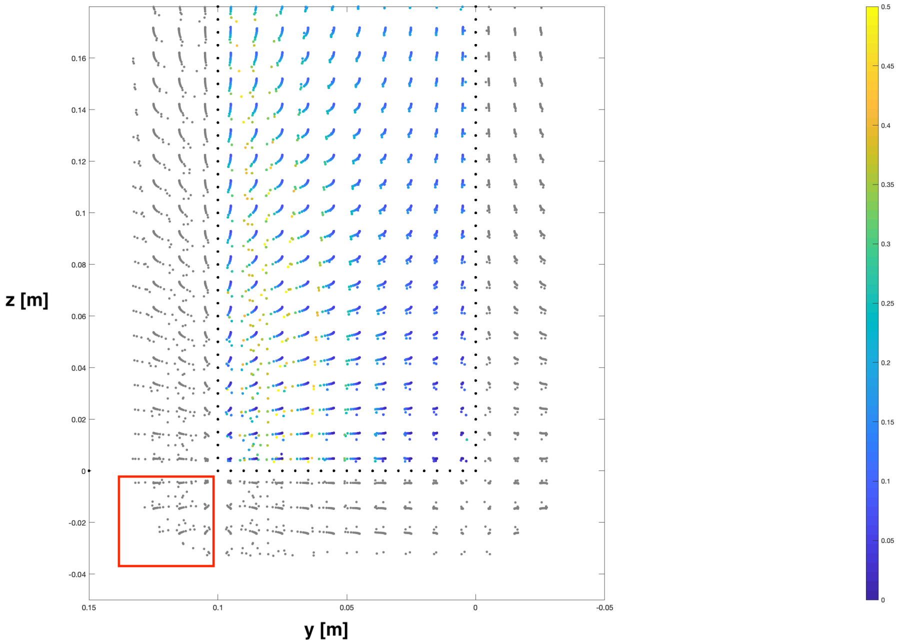
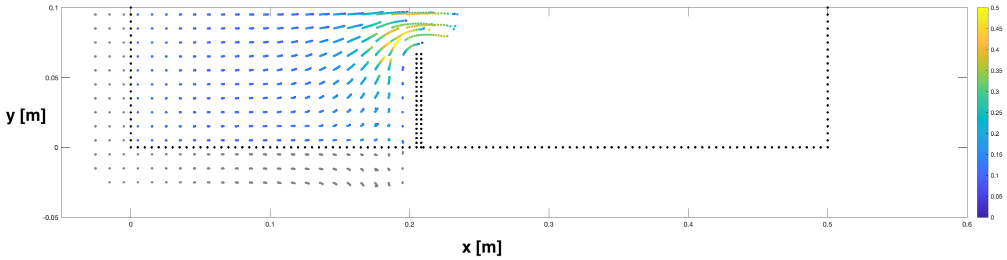
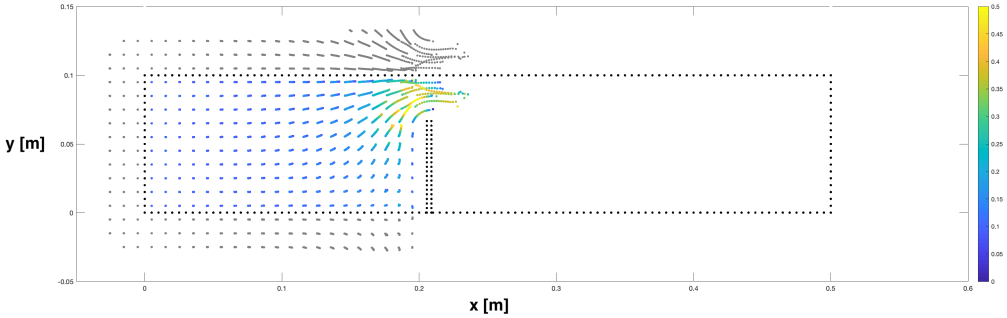


Figure 4.19: Half domain front view representation at time $t = 0,05$ s. Real (color proportional to v_y), virtual (black) and ghost (grey) particles are represented. The colorbar indicates the transversal velocity v_y.

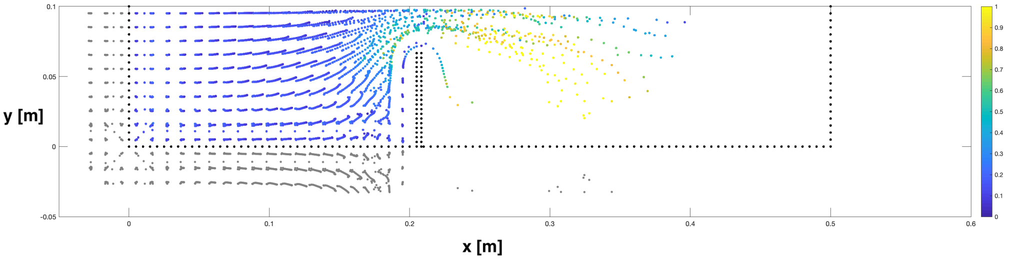


(a) Full domain

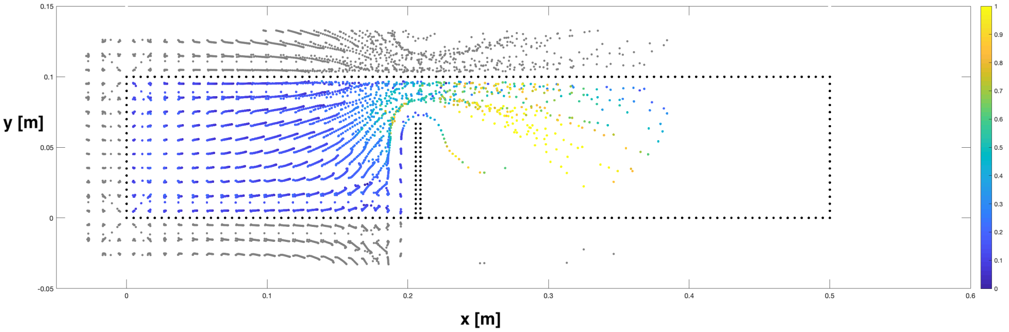


(b) Half domain

Figure 4.20: Full domain (left) and half domain (right) up view representation at time $t = 0,05$ s. Real (color proportional to v_y), virtual (black) and ghost (grey) particles are represented. The colorbar indicates the transversal velocity v_y.



(a) Full domain



(b) Half domain

Figure 4.21: Full domain (left) and half domain (right) up view representation at time $t = 0,15$ s. Real (color proportional to v_y) and virtual (black) and ghost (grey) particles are represented. The colorbar indicates the transversal velocity v_y.

tank, $d = 0,15$ m long, $e = 0,15$m wide and $HO = 0,25$ m high in order not be submersible, as in the original experiment. The obstacle is also made by object particles and it's placed in a floodable area $f = 1.775$ m long and $l = 0,6$ m wide. Taking into account the CFL condition, the timesteps is set to $\Delta t = 1,0 \cdot 10^{-04}$ s. The amount of particle involved in this simulation is: 576.000 real particle, 454.400 virtual particles, and 152.876 object particles. Which leads to a total of 1.183.276 particles.

In figure 4.23 five snapshots of experimental and EISPH water depths in the floodable area at different times are compared. To make possible and easier the comparison, a mirrored half is added at the simulation snapshot. Unlike before, the simulations snapshots are taken discarding the particles with a depth lower than 0, 005. To appreciate more how similar are the simulation and the experiment, an overlap between the two has been made in figures 4.24.

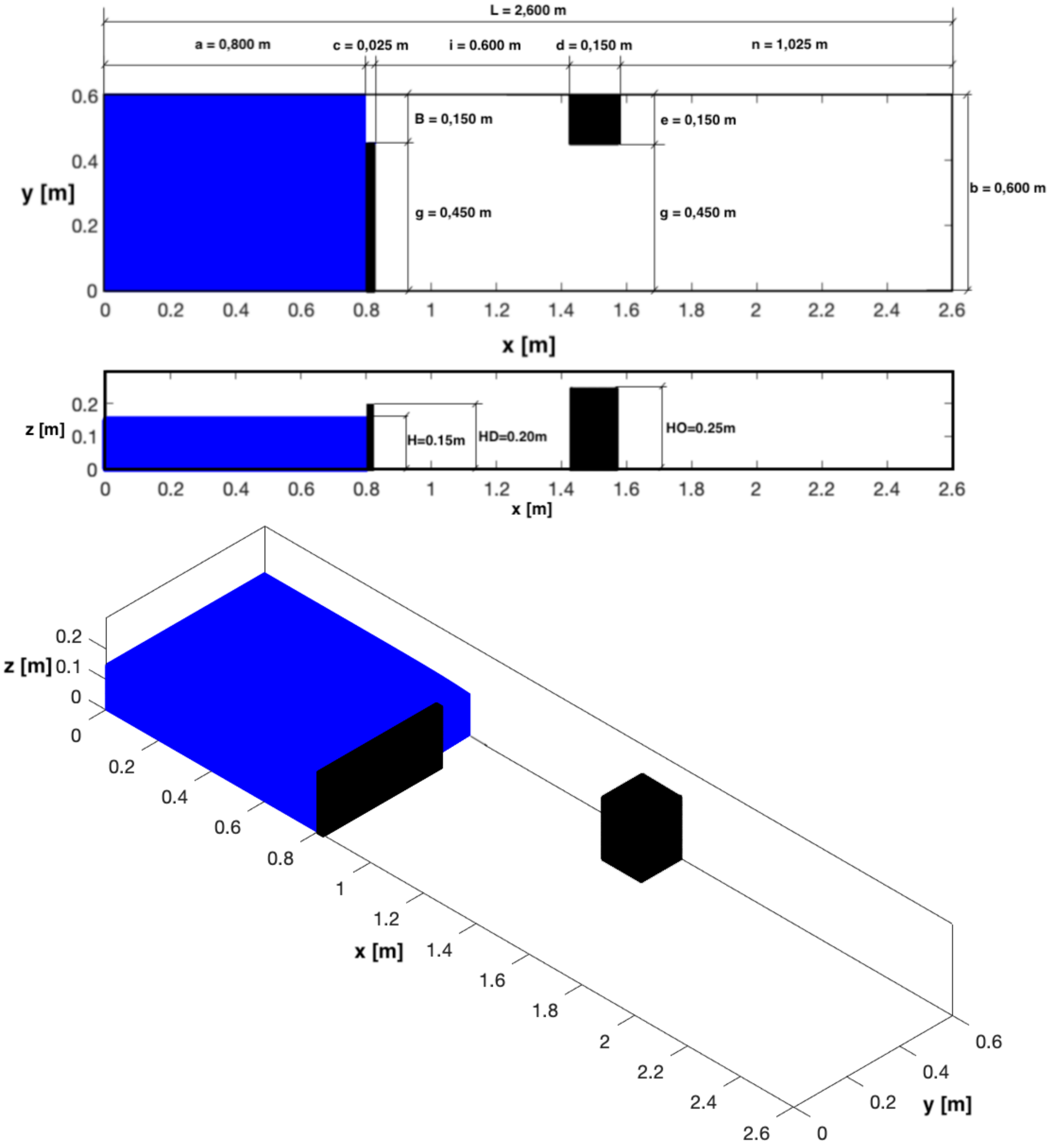


Figure 4.22: Simulation facility scheme and initial particle configuration, with main dimension for the half domain simulation.

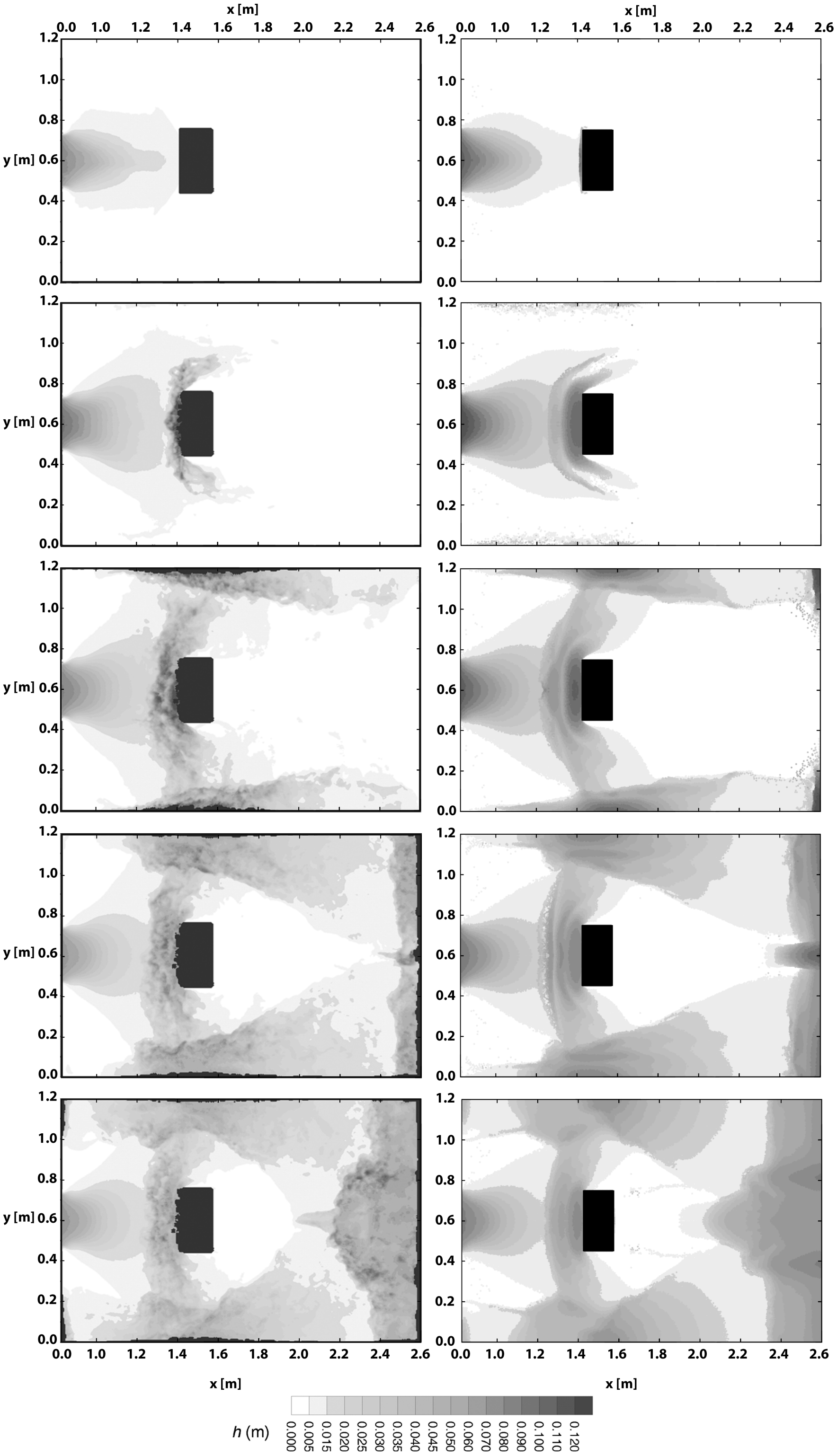


Figure 4.23: Comparison between experimental (left) and EISPH simulation (right) at time $t = 0,40$ s; $t = 0,75$ s; $t = 1,45$ s; $t = 2,16$ s; $t = 2,86$ s. Half domain: 576.000 real particles.

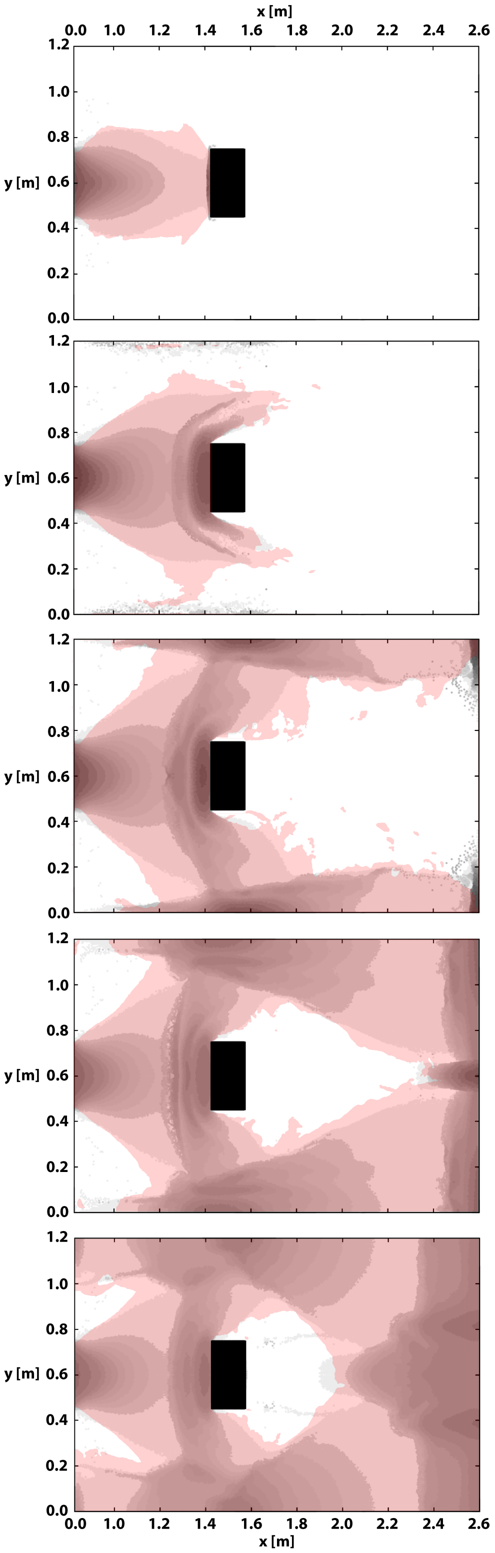


Figure 4.24: Overlapped snapshot at time $t = 0,40$ s; $t = 0,75$ s; $t = 1,45$ s; $t = 2,16$ s; $t = 2,86$ s. Red area is the experiment water profile. Half domain: 576.000 real particles

In order to understand better the difference between the half-domain and the full domain simulations, figure 4.25 compares the two of them, considering all the particles in the domain, the ones under $0,005$m too. It can be seen how in this last simulation, more areas are filled with water with smoother curves and much less detached particles.

As done for the 144,000 particle simulation, also for this one the particle height at the gauge point was compared to the water level measured with the ultrasonic transducer in the experiment. The results of this comparison are shown in figure 4.26. Although the level in the simulation closely follows that of the experiment, the simulation with 144.000 particles seemed to copy better the initial peak and the oscillations; however, it must be said that in the latter, since it is computationally less heavy, the output data have been saved every 200 timesteps, ie every 0.02 s of simulation; in the current simulation instead, the weight of each output file is much greater, resulting in a slowest computation, so they were saved every 500 timesteps, equivalent to 0.05 s of simulation. For this reason, some information about the initial peak and some oscillations have been lost.

The same comparison is made also with the results obtained from a VOF numerical modelling at the same gauge point in [1], with the same considerations of Section 4.2.1. The analysis is extended to the whole domain by creating a grid and applying it on the simulation (with the addition of the mirrored half) field and on the experiment snapshots, at the same timesteps, as done for the first simulation. The grid in this case, since the particles are smaller, is composed by cells of size $lx = ly = 1,5dc = 0,0075$ m (and $lz = 0,20$ m for the simulation). The grid is visible in figure 4.27.

Gathering together the height deviations for the time steps investigated $t = 0.40$ s, $t = 0.75$, $t = 1.45$ s, $t = 2.16$ and $t = 2.86$ s, 188.800 values are obtained which, divided into 60 classes (the same number used by the authors in their analysis) with a constant interval equal to the difference between the maximum value and the minimum value divided by 60, allow to realize a frequency distribution that is useful to evaluate quantitatively as the simulation approaches the real model. Since the sensibility to the user accuracy doesn't affect very much the final results (as long as he carries out the process carefully), as demonstrated in the previous simulation, only one repetition of the process has been made and the final distribution is shown in table 5 where the classes are represented by the mean value of the classes interval deviation, while the absolute, relative and cumulative frequency distribution are graphed at figure 4.28. The main statistics values of the deviation $h_{SPH} - h_{exp}$ are shown in table 4.

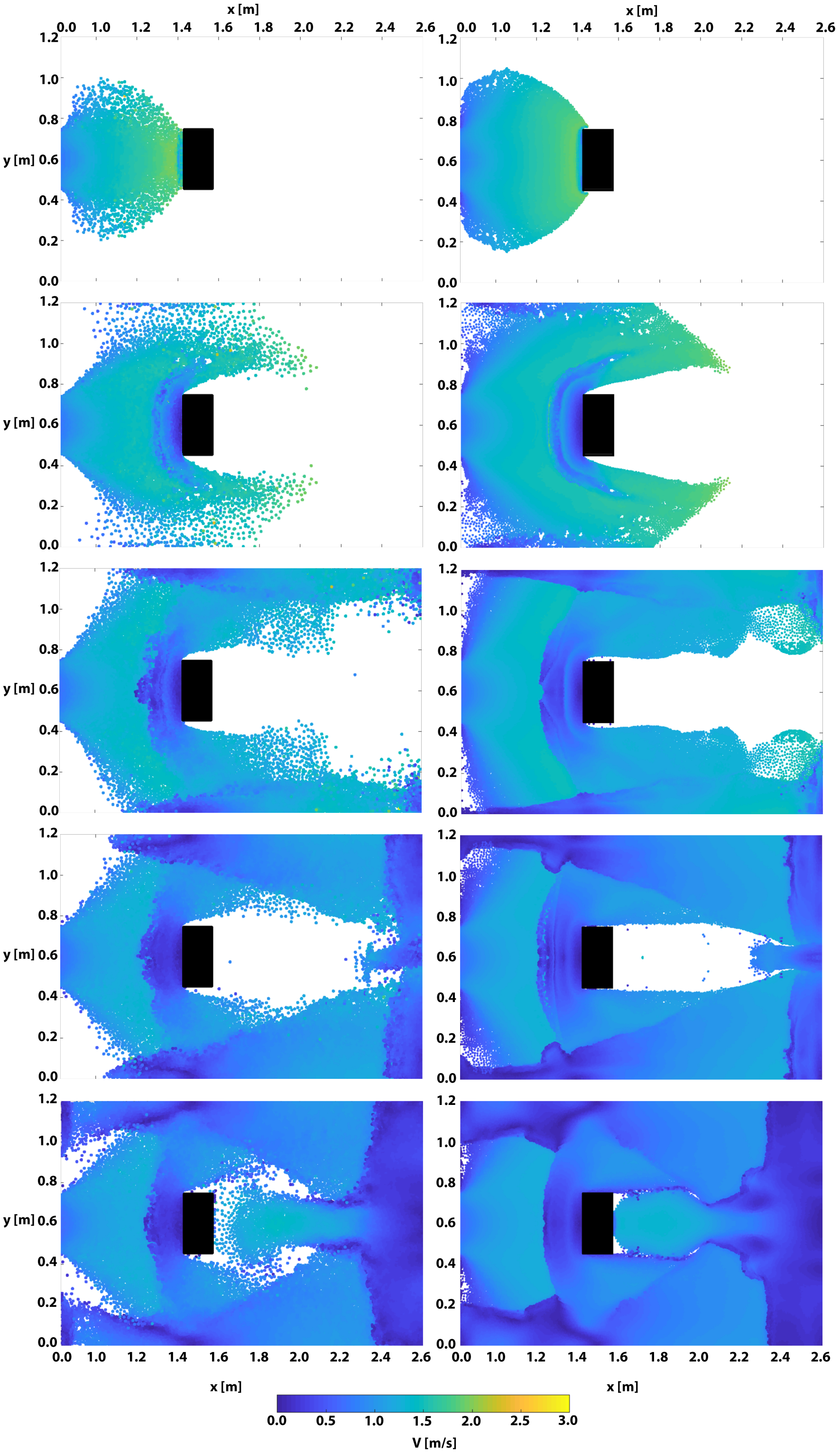


Figure 4.25: Comparison between 144.000 (left) and 576.000 (right) real particles simulations. The colorbar represent the velocity modulus of the particles

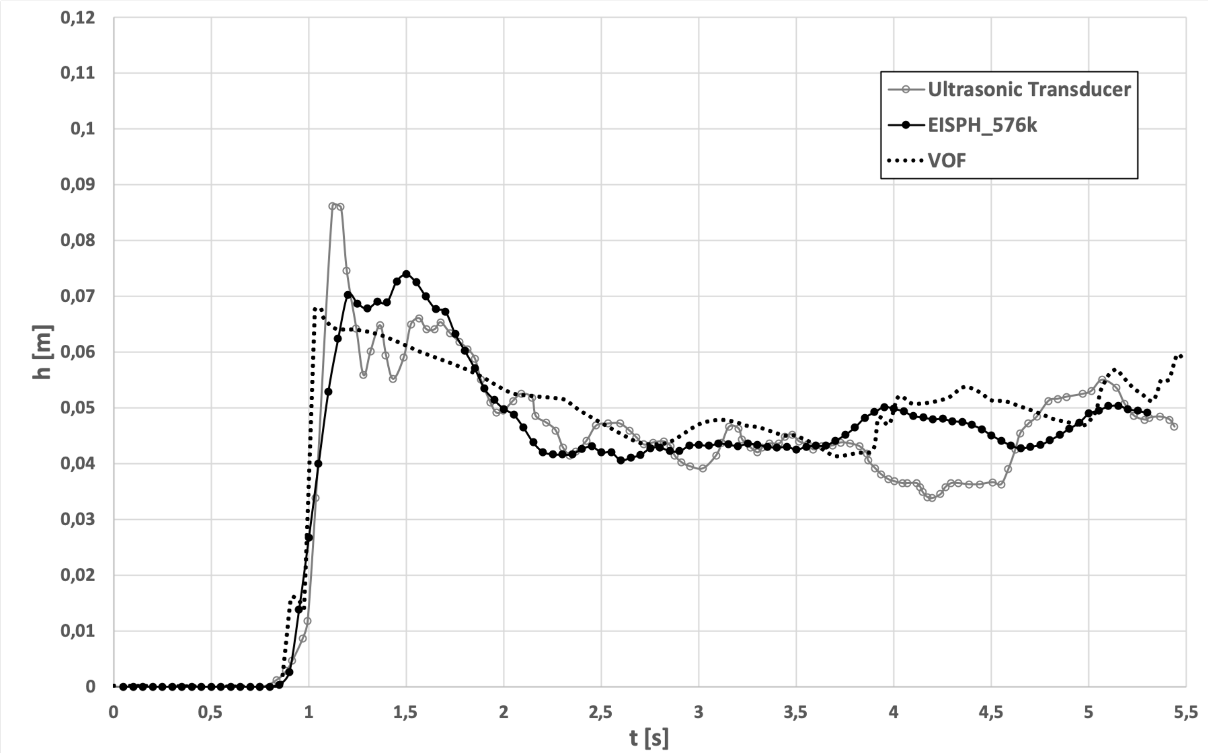


Figure 4.26: Water level time series for EISPH simulation and ultrasonic transducer. Half domain: 576.000 real particles.

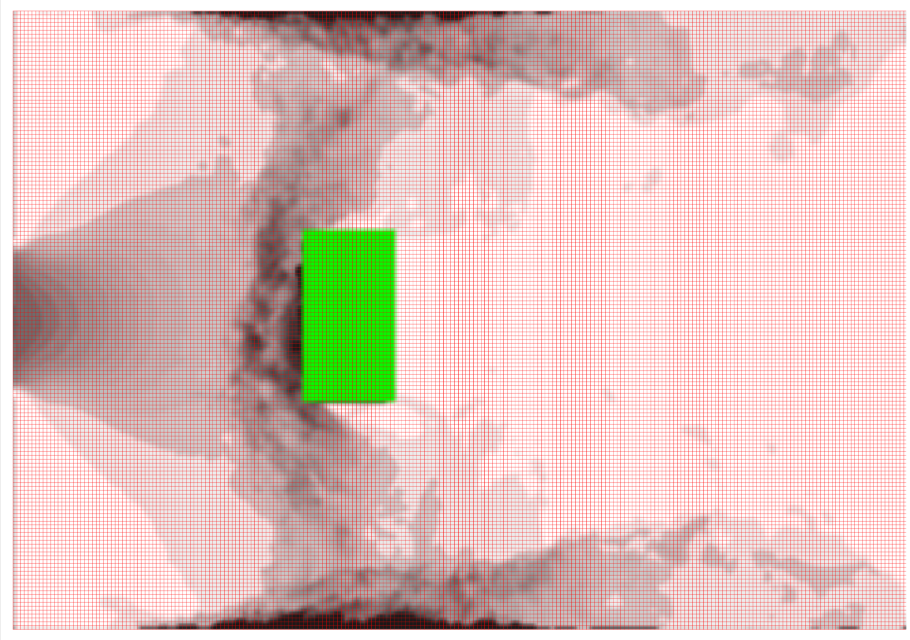


Figure 4.27: Interpolation grid - 576.000 particle simulation

Table 4: Main statistics value of the deviation $h_{SPH} - h_{exp}$

Max	$108,61$	[mm]
Min	$-107,67$	[mm]
Mean	$0,58$	[mm]
Median	$-2,63$	[mm]
15th perc.	$-4,89$	[mm]
85th perc.	$2,37$	[mm]

Table 5: Absolute, Relative and Cumulative frequency distribution for every $h_{SPH} - h_{exp}$ classes

$h_{SPH} - h_{exp}$	$h_{SPH} - h_{exp}$	Absolute Frequency	Relative Frequency	Cumulative Frequency
[m]	[mm]	[-]	[%]	[-]
-0,10587	-105,87	4	0,00002	0,00002
-0,10226	-102,26	17	0,00009	0,00011
-0,09866	-98,66	8	0,00004	0,00015
-0,09505	-95,05	21	0,00011	0,00026
-0,09145	-91,45	41	0,00022	0,00048
-0,08784	-87,84	71	0,00038	0,00086
-0,08424	-84,24	45	0,00024	0,00110
-0,08063	-80,63	50	0,00026	0,00136
-0,07703	-77,03	71	0,00038	0,00174
-0,07342	-73,42	56	0,00030	0,00203
-0,06982	-69,82	53	0,00028	0,00231
-0,06621	-66,21	88	0,00047	0,00278
-0,06261	-62,61	72	0,00038	0,00316
-0,05900	-59,00	89	0,00047	0,00363
-0,05540	-55,40	124	0,00066	0,00429
-0,05180	-51,80	134	0,00071	0,00500
-0,04819	-48,19	121	0,00064	0,00564
-0,04459	-44,59	148	0,00078	0,00642
-0,04098	-40,98	194	0,00103	0,00745

-0,03738	-37,38	270	0,00143	0,00888
-0,03377	-33,77	368	0,00195	0,01083
-0,03017	-30,17	411	0,00218	0,01301
-0,02656	-26,56	533	0,00282	0,01583
-0,02296	-22,96	682	0,00361	0,01944
-0,01935	-19,35	1089	0,00577	0,02521
-0,01575	-15,75	1754	0,00929	0,03450
-0,01214	-12,14	2320	0,01229	0,04679
-0,00854	-8,54	3760	0,01992	0,06671
-0,00493	-4,93	14420	0,07638	0,14308
-0,00133	-1,33	105565	0,55914	0,70222
0,00227	2,27	22780	0,12066	0,82288
0,00588	5,88	16825	0,08912	0,91199
0,00948	9,48	7026	0,03721	0,94921
0,01309	13,09	3398	0,01800	0,96720
0,01669	16,69	2026	0,01073	0,97793
0,02030	20,30	1161	0,00615	0,98408
0,02390	23,90	705	0,00373	0,98782
0,02751	27,51	418	0,00221	0,99003
0,03111	31,11	254	0,00135	0,99138
0,03472	34,72	150	0,00079	0,99217
0,03832	38,32	126	0,00067	0,99284
0,04193	41,93	93	0,00049	0,99333
0,04553	45,53	87	0,00046	0,99379
0,04914	49,14	94	0,00050	0,99429
0,05274	52,74	76	0,00040	0,99469
0,05635	56,35	82	0,00043	0,99513
0,05995	59,95	135	0,00072	0,99584
0,06355	63,55	111	0,00059	0,99643

0,06716	67,16	94	0,00050	0,99693
0,07076	70,76	55	0,00029	0,99722
0,07437	74,37	51	0,00027	0,99749
0,07797	77,97	63	0,00033	0,99782
0,08158	81,58	111	0,00059	0,99841
0,08518	85,18	136	0,00072	0,99913
0,08879	88,79	59	0,00031	0,99944
0,09239	92,39	32	0,00017	0,99961
0,09600	96,00	23	0,00012	0,99974
0,09960	99,60	31	0,00016	0,99990
0,10321	103,21	14	0,00007	0,99997
0,10681	106,81	5	0,00003	1,00000

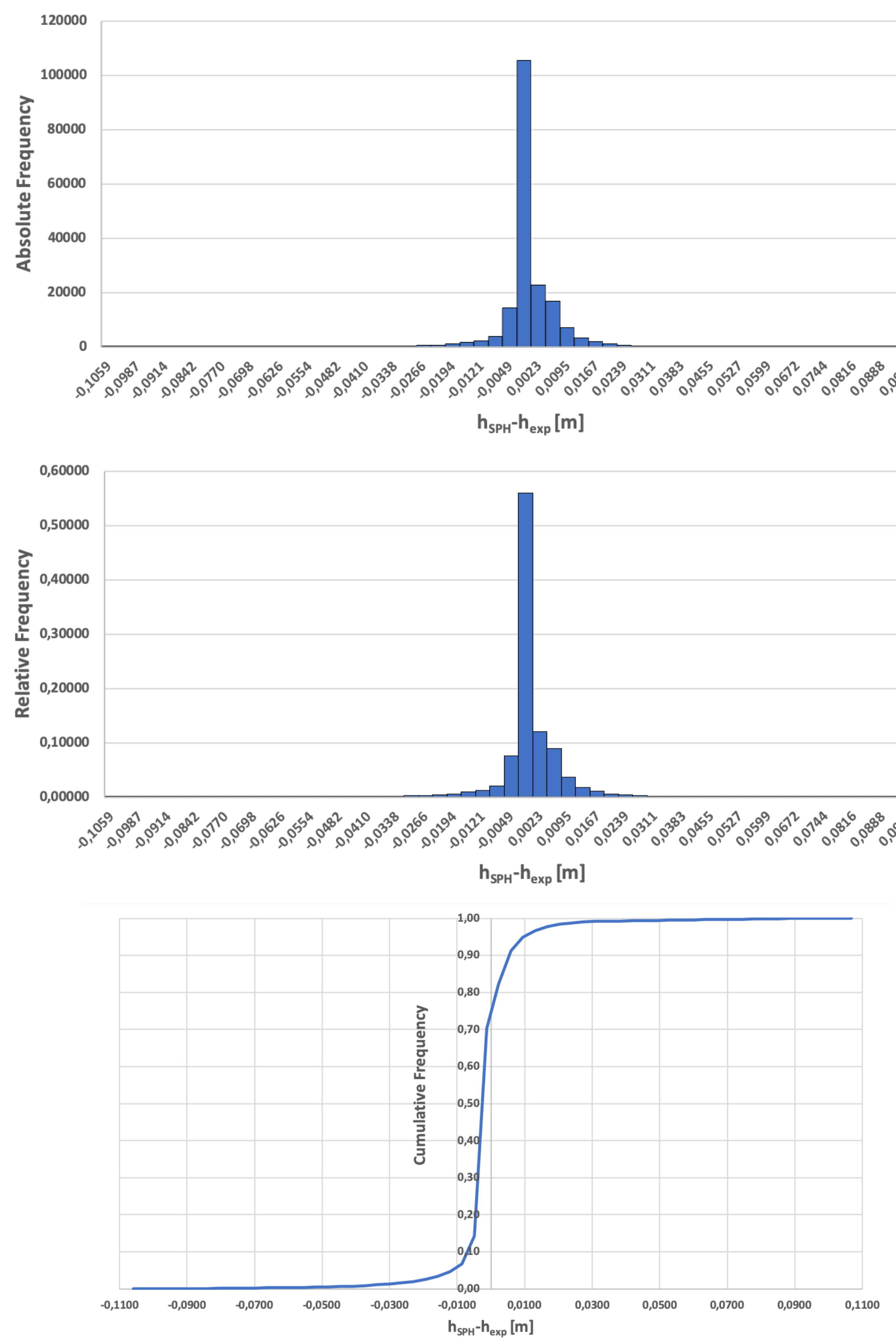


Figure 4.28: Absolute (a), Relative (b) and Cumulative (c) frequency distribution of the deviation $h_{SPH} - h_{exp}$ between numerical and experiment.Full domain: 144k real particles

As the frequency distribution shows, the accuracy of this simulation is greater than the precedent. The values of the 15th and 85th percentiles are closer to the center of the distribution so the 70% of $h_{SPH} - h_{exp}$ values are within a range of $-4,89$ and $2,37$ mm while in the simulation with 144.000 particles the same percentage is within a range of $-18,43$ and $8,40$ mm, so the latter is more open and less accurate. Another difference is that, with 576.000 real particles, the frequency distribution is smoother, without peaks far from the center and the values around the extremes of the interval are almost totally absent; the left branch, here, is monotonically increasing while the right one is monotonically decreasing. It's thus easily to see how increasing the number of real particles, while at the same time decreasing their diameter, leads to a more accurate simulation, which is able to simulate with a good precision a real experiment like the one compared here.

5 Conclusions and future perspective

The present thesis has focused on the use of the three-dimensional *Explicit Incompressible Smoothed Particle Hydrodynamics (EISPH)* scheme to model Dam-Break waves generated by the collapse of irrigation tanks and their propagation in the floodable areas downstream the dam body. As the first test, to evaluates the working condition of the algorithm, a comparison with two dam-break simple cases were performed. The first of them was a two-dimensional EISPH simulation of a free dam-break with free-slip boundary conditions, while the second, performed by Shao & Lo, was the same case as the first, but built with an ISPH scheme with no-slip boundary conditions. As results, the three-dimensional EISPH was able to give the same results as the other two methods, showing a similar wave profile and pressure isobars for different time steps.

As the second test, the 3D EISPH has been compared with the analytical solution of a dam-break wave found by Ritter, which has a friction-less boundary condition and an infinite water tank. The two waves were quite different in the firsts steps since the dam collapse, with the EISPH wave resulting to be slower than the Ritter's one, which spreads downstream with a constant celerity equals to $2\sqrt{gh_0} = 2,4m/s$, while the former starts from 0 to finally reach a constant value of $2,24m/s$. Going forward in the simulation, the EISPH wave tends to converge to the analytic solution, in shape and celerity, despite the leading edge of the latter being more forward, due to the initial velocity.

The third and main test involved a simulation of a complex dam-break problem realized by F. Aureli, A. Maranzoni, P. Mignosa, and C. Ziveri in their experiments. Two models were created: the first one using 144.000 real particles with a diameter of $0,01$ m representing the whole domain; the second, using 576.000 real particles with a diameter of $0,005$ m and representing only half of the domain, exploiting the symmetry of the problem and save half of the requested particles.

The first simulation showed similar water profiles to the experiment; however, the particles weren't enough in number to represent all the water in the domain, especially where it has a level under $0,005$ m. The comparison with the ultrasonic transducer yet showed good results, apart from the tail of the series where the simulation seemed to be in opposition with the measures. To extend the analysis of the water level deviation to all the domain, the grayscale of the experiment snapshots was exploited, applying a grid first on the simulation domain, recording the maximum particle height, and then on the photograms, recording the RGB value for every cell and converting it into water height. The difference between this two was then calculated, realizing a frequency distribution by separating the values in 60 classes. This last analysis provided good results, where the 70% of the samples (47.200) was within a range from $-18,43$ to $8,40$ millimeters, showing however some peaks far from the target value, near -12 millimeters and a very discontinue left branch.

All these defects have been greatly improved with the second simulation. The water profiles suited really well with the experiment one, having less detached particles. Also, the comparison with the ultrasonic transducer measurements gave good results, although it seemed to follow it worse than the previous simulation, lacking the first peak and some oscillations. This was probably the timestep recording value fault, which was of $0,5$s, greater than the $0,2$ s used for the first simulation, and so it hasn't been able to show those points. The most effective improvement is shown in the frequency distribution. In this case, the 70% of the samples (188.800) was within a range from $-4,89$ to $2,37$ millimeters and the 95th percentile presented a value of 9 millimeters. The pick of the distribution counted 105.565 samples in a range between $-3,13$ and $0,47$ millimeters. The frequency distribution was also smoother than the previous one, without peaks far from the target value and with the left branch monotonically increasing while the right one was monotonically decreasing.

This last simulation showed how powerful the SPH algorithm can be, especially in those cases were high curvature of the fluid are present, like in the nearing of the dam body and the obstacle. The suitability to most of the problematic is outstanding and its simplicity to create the problem configuration make it one of the best solutions for Dam-Break simulations. However, its major weakness is the computational effort required to complete a simulation when the number of particles begins to be consistent. For example, the time required to carry out $5,4$ seconds of the 144.000 particle simulation, has been almost 27 hours, while the 576.000 one required almost 300 hours of computation. For this reason, many SPH schemes are being improved by make them run on multi-cores, or, even better, using clusters of GPUs instead of the CPUs, which leads to computational efforts several times lower than before. In conclusion, the EISPH scheme appears to be one of the best existing algorithms, talking about simplicity and stability, so the next improvement that needs to be carried on is the parallelization on GPU, making it become one of the first choices in fluid dynamic simulations.

References

[1] F. Aureli, A. Maranzoni, P. Mignosa, and C. Ziveri, “An image processing technique for measuring free surface of dam-break flows,” *Journal of Hydraulic Engineering*, 2008.

[2] J. D. Pisaniello, R. L. Burritt, and J. Tingey-Holyoak, “Dam safety management for sustainable farming businesses and catchments,” *Elsevier Ltd.*, 2010.

[3] Nomeritae, E. Daly, S. Grimaldi, and H. H. Bui, “Explicit incompressible sph algorithm for free-surface flow modelling: A comparsion with weakly compressible schemes,” *Elsevier Ltd.*, 2016.

[4] S. Shahriari and L. Kadem, *Numerical Methods and Advanced Simulation in Biomechanics and Biological Processes.* ACADEMIC PRESS, 2018.

[5] S. Shao and E. Y. Lo, “Incompressible sph method for simulating newtonian and non-newtonian flows with a free surface,” *Elsevier Science Ltd.*, 2003.

[6] J. Martin and W. Moyce, “Part iv. an experimental study of the collapse of liquid columns on a rigid horizontal plane,” *Royal Society*, 1952.

[7] B. Buchner, “Green water on ship-type offshore structures,” *Bas Buchner*, 2002.

[8] M. B. LIU, G. R. LIU, and Z. ZONG, “An overview on smoothed particle hydrodynamics,” *International Journal of Computational Methods*, 2008.

[9] J. Monaghan, “Smoothed particle hydrodynamics and its diverse applications,” *Annu. Rev. Fluid Mech.*, vol. 44, 2012.

[10] L. E.-S., C. Moulinec, and D. Xu R., “Comparisons of weakly compressible and truly incompressible algorithms for the sph mesh free particle method,” *Journal of Computational Physics*, vol. 227, 2008.

[11] E.-S. Lee, “Truly incompressible approach for computing incompressible flow in sph and comparisons with the traditional weakly compressible approach,” *Ph.D. thesis, University of Manchester*, 2007.

[12] A. Rafiee and K. Thiagarajan, “An sph projection method for simulating fluid-hypoelastic structure interaction,” *Computer Methods in Applied Mechanics and Engineering*, vol. 198, 2009.

[13] D. Barcarolo, D. Touze, and F. Vuyst, “Incompressible smoothed particle hydrodynamics: proposition and validation of a fully-explicit algorithm,” *Seventh International SPHERIC Workshop, SPHERIC*, 2012.

[14] L. Lucy, “A numerical approach to the testing of the fission hypothesis,” *Astron. J.*, vol. 82, 1977.

[15] J. M. R.A. Gingold, “Smoothed particle hydrodynamics: theory and application to non-spherical stars,” *Mon. Not. R. Astron. Soc.*, vol. 181, 1977.

[16] S. Cummins and M. Rudman, "An sph projection method," *Journal of Computational Physics*, vol. 152, 1999.

[17] D. A. Barcarolo, "Improvement of the precision and the efficiency of the sph method: theoretical and numerical study," *Ecole Centrale de Nantes*, 2013.

[18] W. W. . R. Marsooli, "A depth-averaged 2d shallow water model for breaking and non-breaking long waves affected by rigid vegetation," *Journal of Hydraulic Research*, 2012.

[19] S. Hosseini, M. Manzari, and S. Hannani, "A fully explicit three-step sph algorithm for simulation of non-newtonian fluid flow," *International Journal of Numerical Methods for Heat and Fluid Flow*, 2007.

[20] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annual Reviews*, 1992.

[21] F. Aureli, A. Maranzoni, P. Mignosa, and C. Ziveri, "An image processing technique for measuring free surface of dam-break flows," *Springer*, 2010.

[22] W. Wu and R. Marsooli, "A depth-averaged 2d shallow water model for breaking and non-breaking long waves affected by rigid vegetation," *Journal of Hydraulic Research*, 2012.

[23] J. D. Pisaniello, "A comparative review of environmental policies and laws involving hazardous private dams: 'appropriate' practice models for safe catchments," *William & Mary Environmental Law and Policy Review*, vol. 35, 2011.

[24] J. L. Tingey-Holyoak and A. S. John D. Pisaniello, Roger L. Burritt, "Incorporating on-farm water storage safety into catchment policy frameworks: International best practice policy for private dam safety accountability and assurance," *ELSEVIER*, 2012.

[25] S. Marrone, M. Antuono, A. Colagrossi, G. C. D. L. Touzé, and G. Graziani, "d-sph model for simulating violent impact flows," *ELSEVIER*, 2010.

[26] T.-R. Wu, T.-H.-N. Vuong, J.-W. Lin, C.-R. Chu, and C.-Y. Wang, "Three-dimensional numerical study on the interaction between dam-break wave and cylinder array," *Journal of Earthquake and Tsunami*, vol. 12, no. 2, 2018.

[27] H. von Häfen, M. N. Goseberg, J. Stolle, and M. I. Nistor, "Gate-opening criteria for generating dam-break waves," *Journal of Hydraulic Engineering*, 2019.

[28] G. B. Whitham, "The effect of hydraulic resistance in the dam-break problem," 1955.