Particle Based Methods: Mini-project

SPH simulation of a dam break - Verification, Validation and Parameters studies

by Barde Paul, n^{o} 223400

Ecole polytéchnique fédérale de Lausanne
Mechanichal engineering
Particle based methods
Dr Mark L.Sawley
May 29, 2016

Contents

Abstract

Complex free-surface flows are commonly resolved using the Lagrangian Smooth Particle Hydrodynamics (SPH) method. In this paper we propose the simulation of a dam break and the resulting flow impact on a rectangular obstacle. This is done following the test-case proposed by ERCOFTAC in [1] and using the Opensource SPH code `DualSPHysics` [2]. Thanks to the experimental measurements provided by [1], a validation is carried out. A verification from a comparison with a Volume of Fluid simulation produced by [1] is also completed. The obtained results are also confronted with another SPH simulation of the same test case. This SPH simulation has been undertaken by [3] as part of the validation procedure of the `DualSPHysics` code.

The purpose of this work is to produce a simulation of quality, in the limit of the provided computational resources, which results are properly verified and validated. Such a reliable simulation can then be used to investigate the effects of the obstacle geometry on the wave impact.

Due to time constraint, the presented work does not treat the theoretical implementation, the physical model or the use of advanced visualization. Focus is granted to the case study application and the production of a numerical simulation.

1 Introduction

1.1 General scope of the study

During the past decade, densely populated coastal area have been hit by tsunamis. Such events are among the deadliest natural disaster. One can cite for example the 2004 Indian Ocean Tsunami or the 2011 Japan Tsunami. Coastal structures need therefore to be designed accordingly in order to sustain tsunamis' impact, reduce over-topping and floods. Obstacles can be placed ahead of these structures to reduce the energy and the magnitude of the tsunami or even have it circumvent the populated areas, such structures are called sea walls. This mini-project is a first step toward the investigation of the optimal design for such apparatus. Starting from a simplified problem, with only one rectangular obstacle impacted by a tsunami like wave load, we design a reliable simulation to predict wave heights and pressure on the obstacle's surface. Once this simulation validated and verified, it will then possible to increase the complexity of the problem. One could consider a more detailed obstacle geometry and a beach behind this see wall on which buildings-like structures stand. Then, the impact of the wave, once it has overcome the see-wall, on these on-shore structures can be quantified. A systematic set of simulations will enable the investigation of the optimal design to prevent tsunami induced damages.

1.2 Quick overview of the SPH method

The SPH method is a meshless particle based method. The fluid is discretized as a set of distributed particles of finite size. They have their individual quantities such as position, velocity, mass, density, pressure, etc. Each particle follows the equations of motion and conservation, moreover, a particle interacts with its environment by interpolating the properties of the particles surrounding it (such as pressure). This method is known to produce accurate results for flows with large surface deformations and interacting with complex geometries. In addition, no special treatment is required to account for these large surface deformations. The dam break wave impact of this work is therefore preferably resolved with SPH. Nevertheless, the method posses a large number of numerical and physical parameters that must be tuned accordingly to properly model the flow. Therefore, the design of an accurate simulation may require an significant amount of time (and trials) in order to get to the proper set of parameters and capture the flow's features of interest. The work provided here would be the first step of such a study: the first selection of a parameter set and the comparison of the resulting simulation to the desired output. After repeating this routine several times, one gets a simulation that efficiently captures the physic of the problem. This simulation design can then be used to investigate different topological configurations of a similar problem.

The aim of this paper is not to reach such an accurate design but to show the first steps of this iterative process. It could therefore serve as a starting point for a more elaborated work or project.

2 Simulation design

2.1 Problem definition

In this section the experimental setup, on which the simulation is based, is defined. It has been assessed [4] that tsunami-wave like hydrodynamic loading on structures can be reproduced by high velocity dam-break bores. The Kleefsman's dambreaking test-case carried out here is inspired from this analogy. The following paragraph is extracted from [1].

> «In Kleefsman's experimental set up, a large tank of $3.22\times1\times1$ m is used with an open roof. The right part of the tank is first closed by a door. Behind the door, 0.55 m of water is waiting to flow into the tank when the door is opened. This is done by releasing a weight, which almost instantaneously pulls the door up. In the tank, a box has been placed [...]. During the experiment, measurements have been performed on water heights, pressures and forces. As shown in Fig. 2.1, four vertical height probes have been used: one in the reservoir and the other three in the tank. The box was also covered by eight pressure sensors, four on the front of the box and four on the top.»

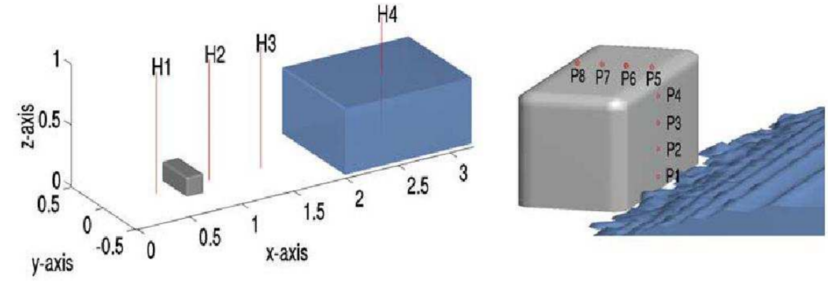


Figure 2.1: Experimental set-up and measurement points

A more precise blueprint of the domain with all the dimensions is given in appendix A.

In this work the quantities of interest are all the surface elevations H_1, H_2, H_3 and H_4 and two pressure probes, P_1 and P_3, on the front side of the obstacle where the impact occurs.

2.2 Technique employed

In this section the parameters and the processes of the simulation are explained. The simulation and the post-processing of the results is done thanks to the open-source software `DualSPHysics` [2]. The geometry is defined following the blueprints Fig. A.1 and Fig. A.2. For the boundaries and the obstacle we define a lattice of 2 (2 particles per point) and a lattice of 1 for the fluid. Moreover, the obstacle is drawn as a solid (particles at the faces and inside). This should prevent the fluid particles to escape the computational domain through the boundaries, especially during the wave impact when the velocities and the forces are at their apex. To model the water at rest, awaiting behind the retractable wall, we defined a cubic volume of water. The water is initially at rest but subject to gravity, hence it starts flowing into the tank at the first time step of the simulation. This is quasi equivalent to the experimental set-up as we will see from the flow features of the simulation debuts.

Constants definitions: Here are displayed the constants of the model. The one modified from their default value are discussed.

gravity	**cflnumber**	**hswl**	**coefsound**	**coefficient**	**gamma**	**rhop0**	**eps**
$-9.81\boldsymbol{e_z}$	0.2	auto	20	1.2	7	1000	0.5

Table 2.1: Constant definition in `DualSPHysics`

`rhop0` corresponds to water density, `gamma` is the coefficient of Trait's state equation that appears in the weakly compressible formulation of `DualSPHysics`. To account for the ocean properties it is recommended to set it to 7 [5]. Similarly, for dam break hydrodynamic impact simulations a `coefficient` value of 1.2 is recommended. The minimum allowed time-step DtMin is set to 1.0×10^{-5}s, this is a way to limit the time elapsed during the simulation but might produce a simulation that ignores and does not capture some high velocity features of the flow.

Parameters definition: Here are displayed the numerical parameters of the solver. The one modified from their default value are discussed.

Time-stepping algorithm	**VerlvetSteps**	**Kernel**	**Viscosity coef**	**Shepard filter**	**Delta-SPH coeff**
Velocity -Verlvet	40	Cubic Spline	Artificial 0.01	No	0.1

Table 2.2: Parameters definition in `DualSPHysics`

It as also been chosen not to exclude particles based on their density values. For more details the readers is refereed to appendix B where both `.bat` and `.xml` files of the simulation are reported.

In order to account for the computational resources limitations it has been decided to make a first simulation with less than 200'000. To fulfill this condition the initial inter-particle distance is set to $dp = 0.0183$m which results in a total of 197'040 particles. It should be emphasized that dp is proportional to the spatial resolution of our system, hence the geometry we defined is accurate up to $\sim dp$. This means that the dimensions defined by the blueprints are not exactly matched. For example, the height of the water column at time $t = 0$ should be 0.55 m, but one gets 0.5325 m if a `<fillbox>...</fillbox>` is used and 0.565 m if a `<boxfill>...</boxfill>` is chosen. Indeed, the `fillbox` fills the defined domain with fluid but without exceeding it whereas the `boxfill` draws a box of the indicated dimensions and then replaces it with fluid particles. The `fillbox` option is chosen in this work since the experimental measurements of [1] start at 0.54m instead of 0.55m.

Later on, a simulation with 1 million particles is carried out with the same set of coefficients in order to asses the effect of the particle size. The inter-particle distance is hence $dp = 0.0098$m and results in 1'023'096 particles.

3 Results and discussion

The physical quantities such as the pressure and the wave heights are interpolated from the simulation's output thanks to `DualSPHysics`'[2] `Measuretool` script. Similarly, the free-surface visualization is computed with `DualSPHysics`' `IsoSurface` script. The visual rendering is obtained with the Opensource software `Paraview` [6].

3.1 Simulation's direct output

200k particles

- DTs adjusted to DtMin: 0
- Excluded particles: 900
- Excluded particles due to Velocity: 0
- Percentage of excluded particles: 0.45%

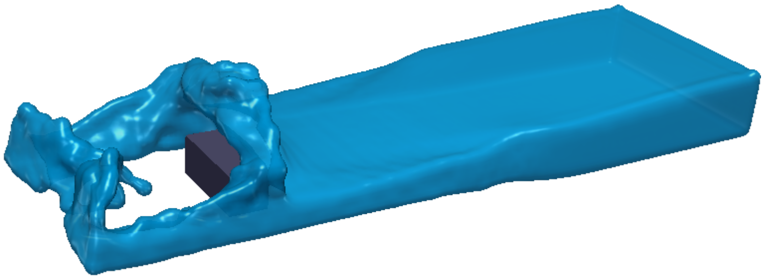
The number of particles exiting the domain through the boundaries is negligible. Moreover, since the particles are rather large, their dynamic is rather slow and the minimal time step is enough to capture it. We will see that this results in a reduced and coarse "splash" when the water hits the obstacle (see Fig. 3.1 a)).

1M particles:

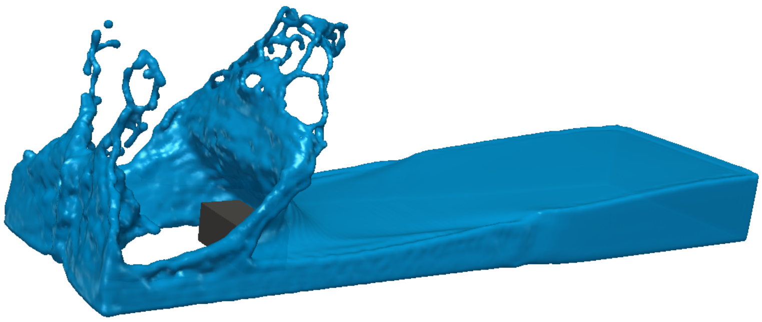
- DTs adjusted to DtMin: 2'859
- Excluded particles: 31'687
- Excluded particles due to Velocity: 743
- Percentage of excluded particles: 3.7%

Here again the number of excluded particles is rather low. But in this case the minimal time step is not small enough to capture the fast dynamics of some particles. Some of them are therefore excluded due to their high velocity that the solver cannot resolve with the granted time step. A particle is excluded when it travels beyond 0.9 times the cell size during one time step (velocity too large compared to the time step)[5].

Now the particles are smaller and their individual dynamics are faster, we get a beautiful and realistic "splash" like it is seen in Fig. 3.1 b). This highlights the necessity to have small particles to capture small scales (and high velocity) feature of the flow such as splashes, droplets and jets. The result is visually much more realistic.



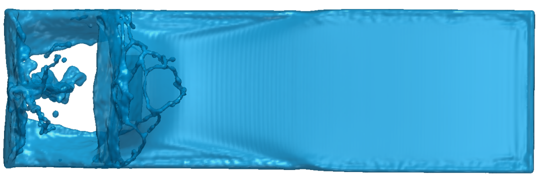
a) 200k particles



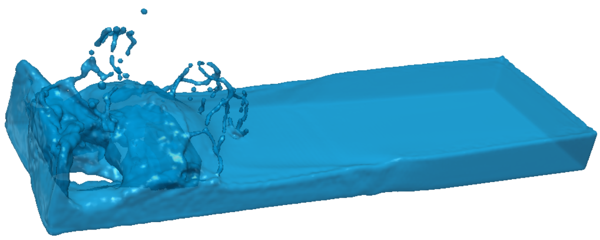
b) 1M particles

Figure 3.1: Visual comparison of water "splash" depending on the number of particles

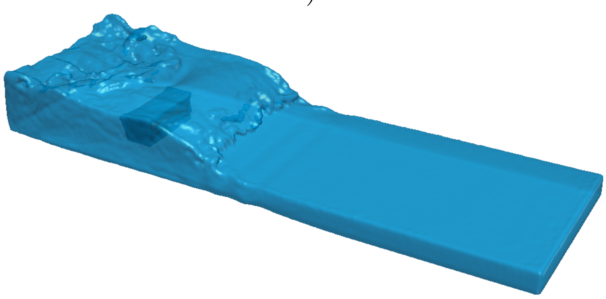
Flow description



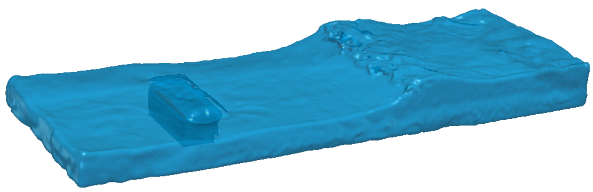
a)



b)



c)



d)

Figure 3.2: Flow configuration at different stages of the simulation. 1M particles

Fig. 3.2:

- a) shows the top view of the flow just after the impact, we see that the flow bypasses the obstacle. Part of the water slides on it from above and produces a "splash" that goes up to 1m. The rest of the fluid avoids the obstacle by the sides; this fraction hits the rear wall and comes back toward the obstacle like a jet. The fluid ahead of the obstacle flows rapidly and is influenced by the lateral walls: this gives rise to two oblique wrinkles on the sides of the tank.

- In b) we have the moment when the jet and the falling splash meet and collapse on the obstacle. At this moment the major part of the water is being accumulated around the obstacle, the incoming wave is being reflected.

- c) displays the resulting reflected wave, which will be called "first reflected wave" from now on. This wave starts its propagation on an adverse current; it resembles a broken and foamy wave incoming on a beach.

- Finally, in d) when the flow has come back to the initial wall and been reflected we are left with the "second incoming wave" that looks like a breaking shallow water wave.

3.2 Qualitative validation and verification

For the validation we used the experimental data of [1]. Regarding the verification we used the VOF solution from [1] and the SPH simulation of [3]. The following Fig. 3.3 displays on the left column the current simulation, on the middle column [3] simulation and on the right column [1] experiment.

It can be assessed from this visual comparison that our simulation captures the main flow features of the experiment. However the resolution is not accurate enough to accurately capture the spays of small scale lengths present in the experiment. Similarly, the high vorticity structures filled with bubbles and foam is less complex in our SPH simulation. This is due to the fact that we do not model the air, required to produce foam. In addition, we can add that the splash we obtain at the impact is more curved and directed backward, toward the reservoir.

The SPH simulation of [3] is more accurate and seems to have a higher resolution. This is unexpected since they also used 1 million particles. However, they capture more accurately the spays and the reflecting wave at $t = 2$s. This might be due to different numerical parameters (which will be discussed later on) but it may also result from a more refined visualization of the flow surface. One could also argue that the spray particle lacking in our simulation were the one excluded due to their high velocity. A new simulation with a smaller time step would answer this question.

One can also look at the snapshots provided by [1] for the VOF simulation in Fig. 3.1. Clearly, the VOF simulation seems less realistic than the SPH ones. First of all it predicts at $t = 0.4$s a surface with overestimated wrinkles and a fragmented water front. Nothing of this magnitude is observed in the experiment or the SPH simulation. Secondly, the splash is coarsely resolved and does not displays the backward curvature. This emphasize the propensity of SPH method to resolve this kind of flow compared to Eulerian method like VOF which require an overly fine mesh to capture small scale structures.

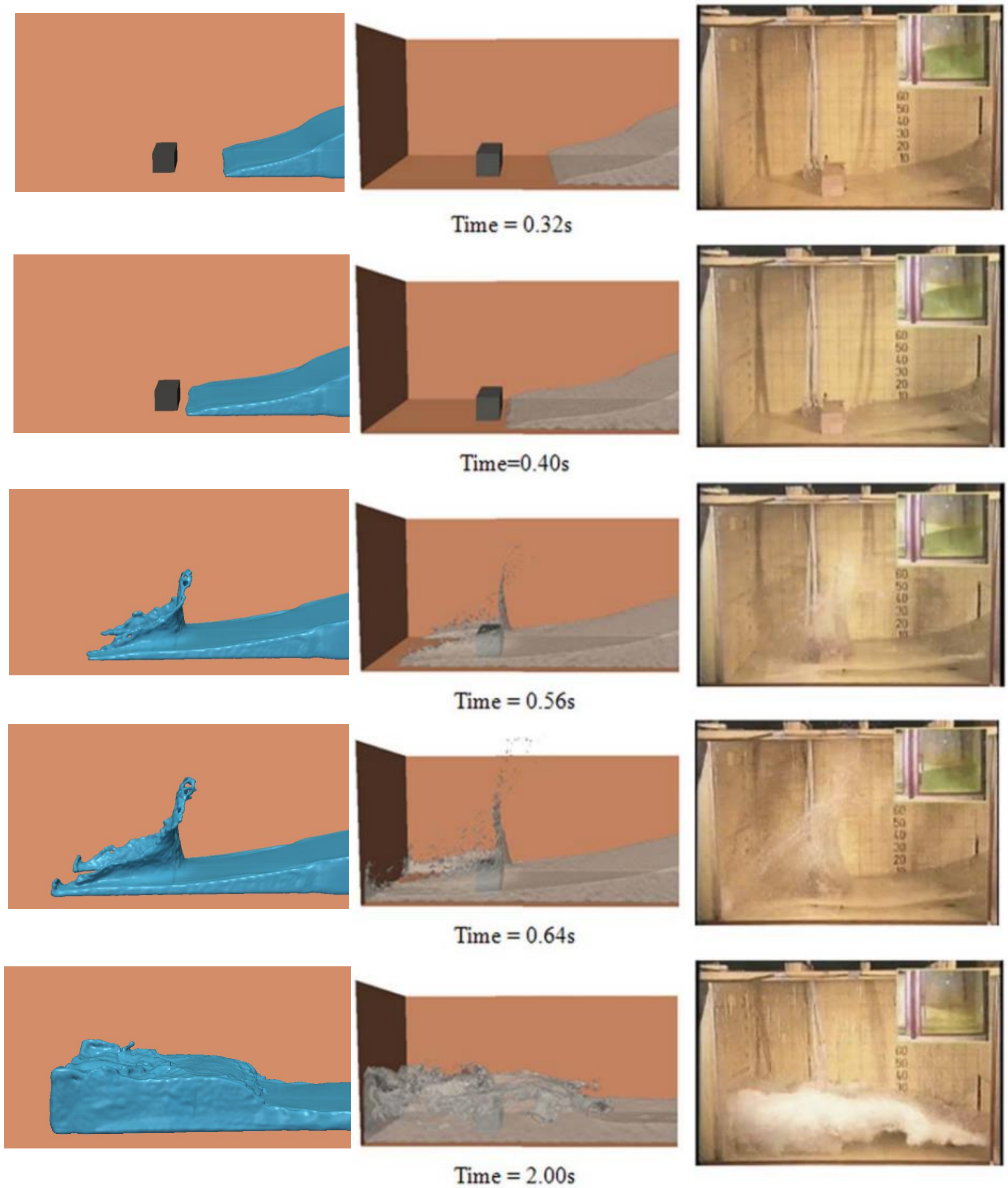
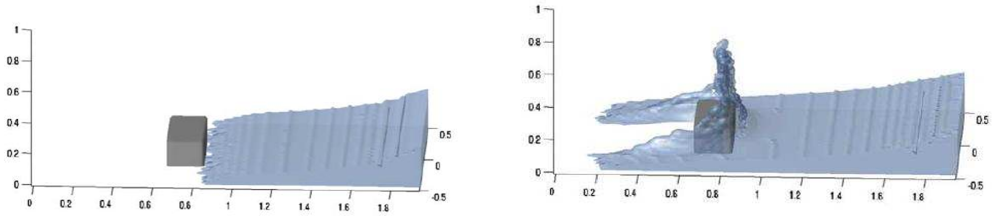


Figure 3.3: Different instants of the SPH simulations and the experiment



VOF at $t = 0.4$s VOF at $t = 0.56$s

Table 3.1: Volume of Fluid simulation of the test case

3.3 Quantitative validation

Surface elevation

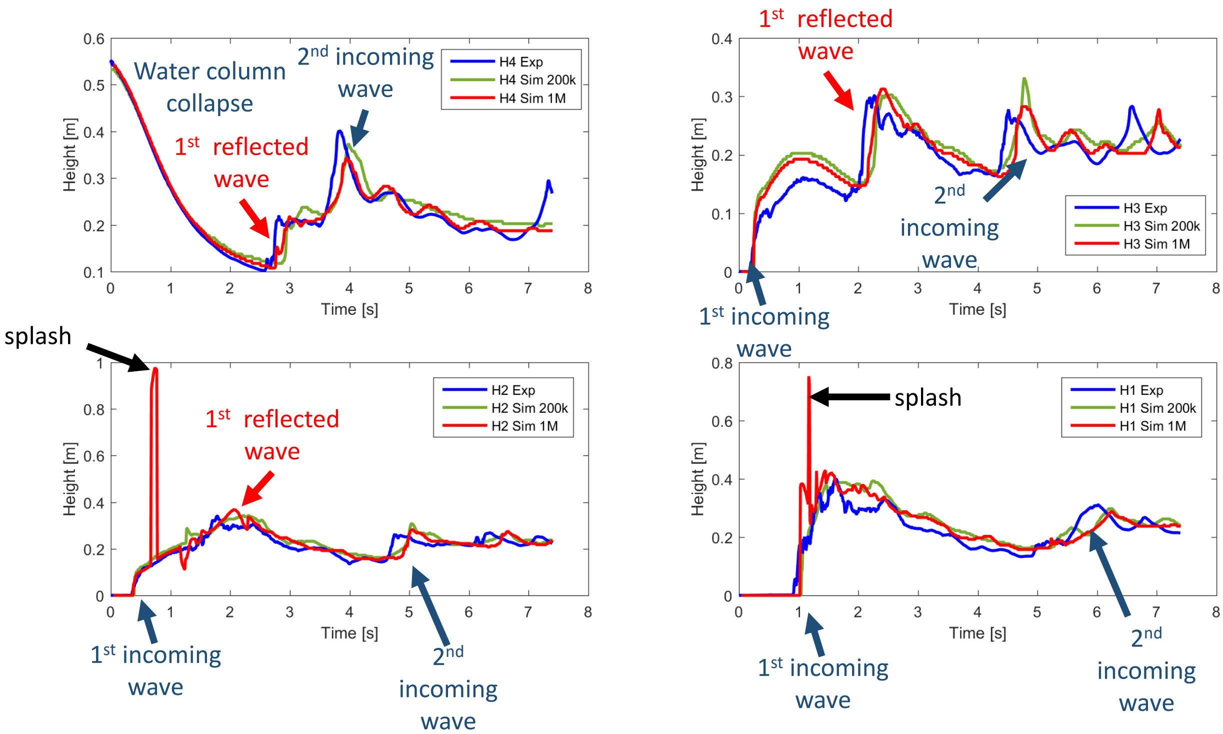


Figure 3.4: Experimental and numerical water heights

Fig. 3.4 shows the experimental and numerical wave height time series at the four selected locations. We see that the agreement is really good. The main difference is that the numerical simulation tends to be delayed compared to the experiment, this might be due to inaccurate physical coefficient modeling the wave propagation and reflection. A second observation is that the overall surface elevation is higher than the experimental one. However, the height of the reflected waves are slightly underestimated.

Finally, the two large peaks in H_2 and H_1 are due to the splashes. On H_2 it is the splash generated by the wave impact on the obstacle. It is oriented slightly backward and therefore eventually intersects H_2. The peak in H_1 is due to the flow impacting the back wall of the domain. They are not explicitly present in the experimental data and I assume that this is due to the measurement method. In the experiment they are performed visually and hence do not consider splashes as representative of the surface elevation.

It should be pointed out that going from 200'000 particles to 1 million permitted to capture the splash and the small waves present after the second incoming wave in H_4. Yet, it does not reduce the delay in the dynamics of the wave propagation or the underestimation of the wave heights. This delay is hence most likely due to a small inaccuracy in the numerical parameters of the simulation and induces an error in the wave propagation and reflection. The delay seems to increase at each reflection.

Pressure

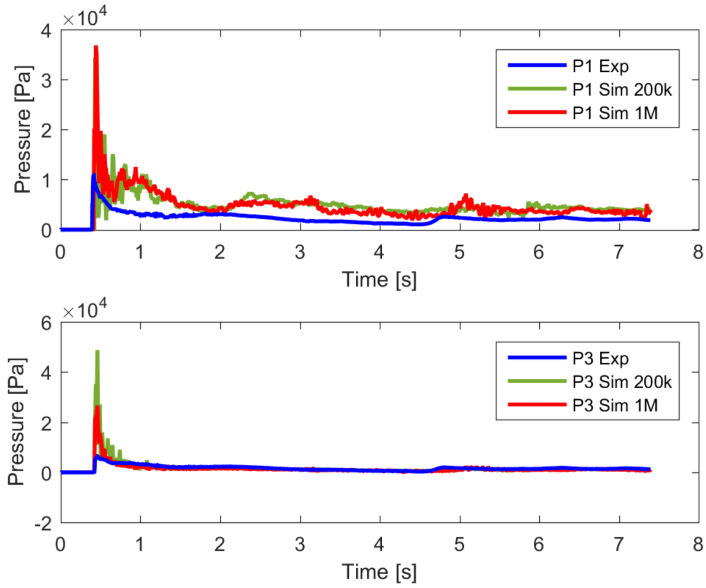


Figure 3.5: Experimental and numerical pressure values

From Fig. 3.5 we can see that the numerical pressures fluctuate a lot, moreover there is a large over-prediction of the impact pressure. The pressure at P_3 is only over-predicted at the impact whereas the over-prediction lasts during the entire simulation at P_1.

This type of behavior is typical of a simulation without Sherpard density filter [4]. The use of more particles reduces only the impact pressure peak at P_3 but does not really affect the rest of the simulation. Here again the improvement should be on the numerical parameters and in the coefficients of the Tait's equation. A Sherpard filter should also be used.

3.4 Quantitative verification

With VOF

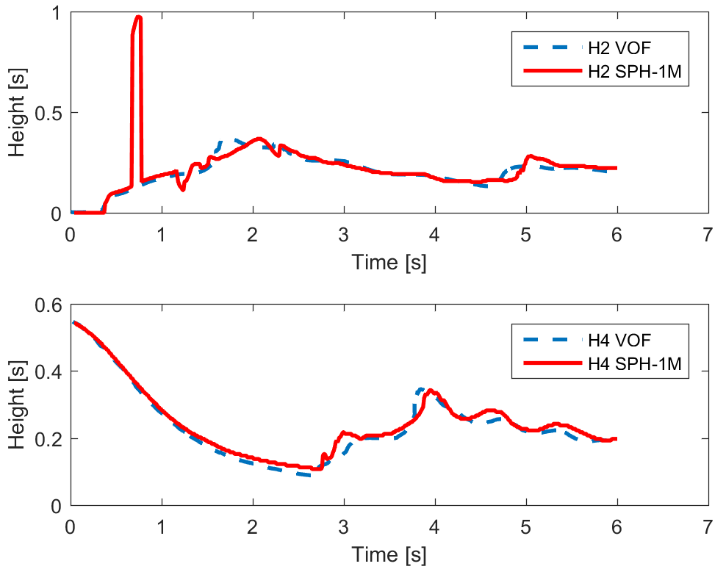


Figure 3.6: Numerical water heights for VOF and SPH simulations

The two simulation are in good agreement, the splash is not present in the VOF but this most likely due to a different definition of the surface elevation. SPH water height is again a little delayed, for example in the prediction of the second incoming wave at H_2 around $t = 5\text{s}$.

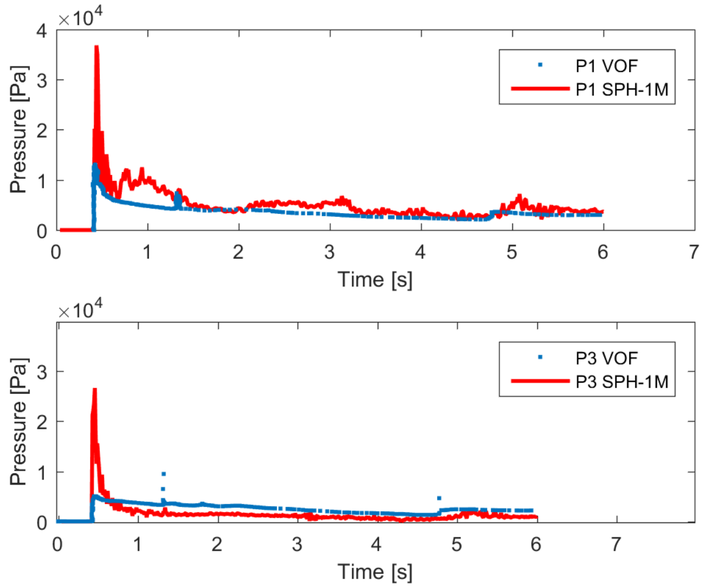


Figure 3.7: Pressure values for VOF and SPH simulations

Regarding the pressure, we find the same deviations than in the validation section. Moreover, the SPH simulation does not capture the pressure variations due to the first reflected wave (around $t = 1.5$s).

4 Discussion and possible improvements

The overall verification and validation is satisfactory considering the complexity of the flow. The present simulation routine could be used to predict the water height from a tsunami impact on different geometries. However it should be improved in order to reduce the delay in the wave propagation if one is interested in the time dependency of the surface elevation. Similarly, the actual simulation could not be used to predict the force on the obstacle accurately. However, it could be helpful to compare the pressure magnitude on two obstacles of different shapes and give an indication on which design reduces the experienced force.

To give more predictive power to this tool it should be improved. We saw that the number of particles has a great influence on the visual results (see Tab. 3.1) but not much on the quantitative results (see Fig. 3.5). Moreover, [3] archived a much more accurate simulation with also 1 million particles. Therefore, one should try to improve the tuning of the SPH model parameters. Trying these new sets of parameters on simulations with only 200'000 particles is enough to asses the improvements. The main parameters to change are the one intervening in the Tait's state equation, in the artificial viscosity treatment and in the coefficient of sound. Indeed, they are directly related to the wave propagation and reflection and a good tuning should remove the time delay. In addition, [3] suggests that the use of a Sherpard filter reduces the pressure overshoot and fluctuations. Indeed, it has been assessed that the weakly compressible SPH formulation WCSPH (used by `DualSPHysics`) tends to underestimate the water splash and spray but to over estimate the pressure fluctuations [4]. This can be sort out by using the incompressible SPH (ISPH) formulation or by complementing the WCSPH with a density filter (the Sherpard filter for example) and a velocity correction (XSPH variant for example)[3]. We are already using the XSPH correction but the density filter may greatly improve our simulation regarding the pressure.

On the visualization part we might want to develop a thinner surface visualization to detect sprays for example and to permits the use of smaller time steps to track the spays particles.

Finally, to come back to the general scope of the study, the topology should be modified in order to integrate for example a beach behind the obstacle. We could also add another structure on this beach. This way we could see the effect of different obstacles on the resulting force of the wave impacting the on-shore structure.

5 Conclusion

In the present report we produced in a relatively short time (approximately 36 hour) a simulation that already has a predictive power. Indeed, it has been satisfyingly verified and validated on multiple sources. The produced knowledge can therefore be used to quantitatively study the wave height on different topologies. Of course, room remains for improvement, particularly on the pressure distribution. Regarding this last quantity the current simulation only provides indicative information about the qualitative behavior. We provided leads that should be investigated to improve the simulation design toward this direction. This work illustrates the efficiency of the SPH method in the modelization of complex free-surface flows compared to other method such as the Eulerian VoF method. As mentioned previously, the simulation could, after minors improvement, be used to investigated the optimal design of obstacle in order to prevent tsunami induced damages. However, the obtained tool is not restricted to this application and could be used for other investigation. One of them could be for example the analysis of wave unfurling on a ship's deck and their interaction with containers. Predicting if the containers should be fastened to the deck or if their own weight is enough to prevent them to be washed away.

A Appendix

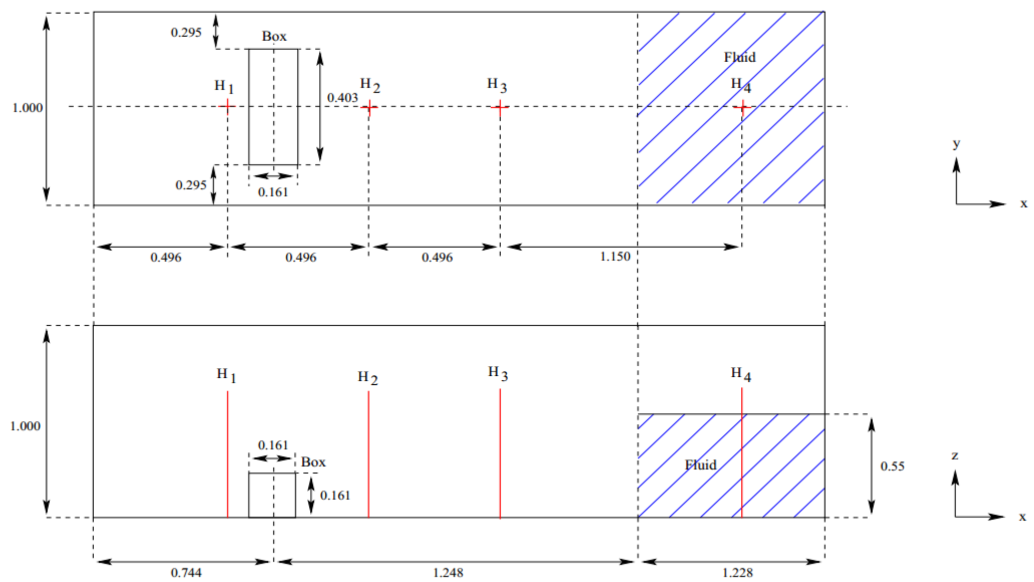


Figure A.1: General description of the system: top (top picture) and side (bottom picture) views

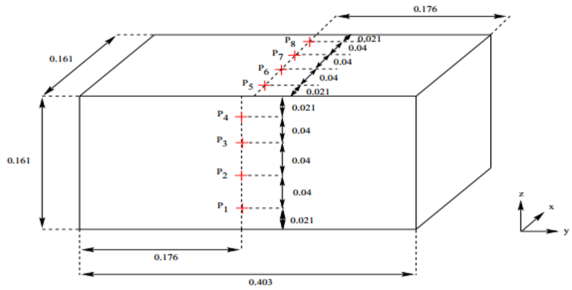


Figure A.2: Description of the obstacle

B Appendix

Case_Projet_Def.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<case application="DualSPHysics Pre-processing Interface"
 date="11/04/2016 20:15:03">
    <casedef>
        <constantsdef>
            <lattice bound="2" fluid="1"/>
            <gravity x="0.0" y="0.0" z="-9.81"/>
            <cflnumber value="0.2"/>
            <hswl auto="true" value="0"/>
            <coefsound value="20.0"/>
            <coefficient value="1.2"/>
            <gamma value="7"/>
            <rhop0 value="1000.0"/>
            <eps value="0.5"/>
        </constantsdef>
        <mkconfig boundcount="2" fluidcount="1"/>
        <geometry>
            <definition dp="0.0183">
                <pointmin x="-0.5" y="-0.5" z="-0.5"/>
                <pointmax x="5" y="3" z="3"/>
            </definition>
            <commands>
                <mainlist>
                    <setmkbound B="0" G="0" R="0" mk="0"
                     name="DomainBox"/>
                    <drawbox>
                        <boxfill>all</boxfill>
                        <point x="0.0" y="0.0" z="0.0"/>
                        <size x="3.22" y="1.0" z="1.0"/>
                    </drawbox>
                    <shapeout file="DomainBox"/>
                  <setshapemode> dp | bound</setshapemode>
```

```xml
            <setdrawmode mode="full"/>
            <setmkfluid B="0" G="0" R="0" mk="0"
             name="Fluid"/>
            <fillbox x="0.1" y="0.1" z="0.1">
              <modefill>void</modefill>
              <point x="-0.1" y="-0.1" z="-0.1"/>
              <size x="1.328" y="1.2" z="0.65"/>
            </fillbox>
            <shapeout file="Fluid"/>
              <setmkbound B="0" G="0" R="0" mk="1"
               name="ObstacleFull"/>
              <drawbox>
                  <boxfill>solid</boxfill>
                  <point x="2.3955" y="0.295" z="0.0"/>
                  <size x="0.161" y="0.403" z="0.161"/>
              </drawbox>
              <shapeout file="ObstacleFull"/>
          </mainlist>
      </commands>
  </geometry>
</casedef>
<execution>
    <parameters>
        <parameter comment="Time-stepping algorithm. 1:
         Velocity-Verlet; 2:Symplectic" key="StepAlgorithm"
          value="1"/>
        <parameter comment="Frequency that Eulerian equations
         are applied at when Velocity-Verlet in use"
          key="VerletSteps" value="40"/>
        <parameter comment="Interaction Kernel. 1: Cubic
         Spline; 2: Wendland" key="Kernel" value="1"/>
        <parameter comment="Viscosity Formulation Method. 1:
         Artificial; 2: Laminar+SPS" key="ViscoTreatment"
          value="1"/>
        <parameter comment="Viscosity Value" key="Visco"
         value="0.01"/>
        <parameter comment="Frequency that the Shepard density
         filter is applied. 0: Not used" key="ShepardSteps"
          value="0"/>
        <parameter comment="delta-SPH coefficient, 0: Not
         used" key="DeltaSPH" value="0.1"/>
        <parameter comment="The size of the initial time-step"
         key="DtIni" value="1.0e-4.0"/>
        <parameter comment="The minimum allowed size of a
         time-step" key="DtMin" value="1.0e-5.0"/>
        <parameter comment="The length of time to simulate"
```

```xml
          key="TimeMax" value="7.4"/>
         <parameter comment="The time that elapses between data
          being output" key="TimeOut" value="0.01"/>
         <parameter comment="The allowed extra space in the Z$
          dimension" key="IncZ" value="2.0"/>
         <parameter comment="Allowed percentage of fluid
          particles out the domain, 1: 100%" key="PartsOutMax"
           value="1"/>
         <parameter comment="Maximum density value allowed
          before particles are excluded" key="RhopOutMax"
           value="0"/>
         <parameter comment="Minimum density value allowed
          before particles are excluded" key="RhopOutMin"
           value="0"/>
      </parameters>
   </execution>
</case>
```

CaseProjet_win64_GPU.bat:

```
@echo off

rem "name" and "dirout" are named according to the testcase

set name=CaseProjet
set dirout=%name%_out


rem "executables" are renamed and called from their directory

set gencase="../../EXECS/GenCase_win64.exe"
set dualsphysics="../../EXECS/DualSPHysics_win64.exe"
set boundaryvtk="../../EXECS/BoundaryVTK_win64.exe"
set partvtk="../../EXECS/PartVTK_win64.exe"
set measuretool="../../EXECS/MeasureTool_win64.exe"
set isosurface="../../EXECS/IsoSurface_win64.exe"

rem "dirout" is created to store results or it is removed if it
 already exists

if exist %dirout% del /Q %dirout%\*.*
if not exist %dirout% mkdir %dirout%


rem CODES are executed according the selected parameters of
execution in this tescase

%gencase% %name%_Def %dirout%/%name%
if not "%ERRORLEVEL%" == "0" goto fail

%dualsphysics% %dirout%/%name% %dirout% -svres -gpu
if not "%ERRORLEVEL%" == "0" goto fail

%partvtk% -dirin %dirout% -savevtk %dirout%/PartFluid -onlytype:-all,+fluid
if not "%ERRORLEVEL%" == "0" goto fail


%measuretool% -dirin %dirout% -points PointsPressure.txt
-onlytype:-all,+fluid,+bound -vars:-all,+press
-savevtk %dirout%/PointsPressure -savecsv %dirout%/PointsPressue
if not "%ERRORLEVEL%" == "0" goto fail

%measuretool% -dirin %dirout% -points PointsHeight.txt
-onlytype:-all,+fluid -height -savevtk %dirout%/PointsHeight
-savecsv %dirout%/PointsHeight
```

```
if not "%ERRORLEVEL%" == "0" goto fail

%measuretool% -dirin %dirout% -points PointsHeightH4.txt
-onlytype:-all,+fluid -height -savevtk %dirout%/PointsHeightH4
-savecsv %dirout%/PointsHeightH4
if not "%ERRORLEVEL%" == "0" goto fail

%isosurface% -dirin %dirout% -saveiso %dirout%/Surface
 -onlytype:-all,+fluid
if not "%ERRORLEVEL%" == "0" goto fail

:success
echo All done
goto end

:fail
echo Execution aborted.

:end
pause
```

References

[1] Reza ISSA and Damien VIOLEAU. Test-case 2: 3d dambreaking. *ERCOFTAC - SPH European Research Interest Community - SIG*, March 2006.

[2] A.J.C. Crespo, J.M. Domínguez, B.D. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, and O. García-Feal. Dualsphysics: Open-source parallel {CFD} solver based on smoothed particle hydrodynamics (sph). *Computer Physics Communications*, 187:204 – 216, 2015. ISSN 0010-4655. doi: http://dx.doi.org/10.1016/j.cpc.2014.10.004. URL http://www.sciencedirect.com/science/article/pii/S0010465514003397.

[3] Crespo AC, Dominguez JM, Barreiro A, Gomez-Gesteira M, and Rogers BD. Gpus, a new tool of acceleration in cfd: Efficiency and reliability on smoothed particle hydrodynamics methods. 2011. URL http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0020685.

[4] Philippe St-Germain, Ioan Nistor, and Ronald Townsend. Numerical modeling of tsunami-induced hydrodynamic forces on onshore structures using sph. *Coastal Engineering Proceedings*, 1(33):81, 2012. ISSN 2156-1028. URL https://icce-ojs-tamu.tdl.org/icce/index.php/icce/article/view/6788.

[5] URL http://www.dual.sphysics.org/index.php/forums1/.

[6] Ayachit and Utkarsh. The paraview guide: A parallel visualization application. Kitware,, 2015. URL http://www.paraview.org/. ISBN 978-1930934306.