



Escola de Camins

Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

Comparative evaluation of two different 3D OpenFOAM modules in a dam-break test

Treball realitzat per:

Júlia Boix Oliva

Dirigit per:

Manuel Gómez Valentín

Esteban Sánchez Cordero

Grau en:

Enginyeria Civil

Barcelona, 16 de juny de 2017

Departament d'Enginyeria Civil i Ambiental

TREBALL FINAL DE GRAU

Summary

The main objective of this work is to analyse the two different turbulence methodologies, LES and RANS, using three numerical OpenFOAM simulations based on the Finite Volume Method. Also, the non-hydrostatic effects produced by the highly curved surface are analysed. In order to assess the 3D model, two dam-break experiments are used to calibrate (Kleefsman et al., 2005) and validate (Fraccarollo and Toro, 1995) the model, respectively.

LES methodology is simulated using Smagorinsky submodel while in RANS the RNG k-epsilon is used. To simulate the water surface, the Volume of Fluid and the *interFoam* solver are used.

A computer with 32 GB of RAM memory and Intel Core i7-6700k processor is used to simulate the experiments. The computational time to simulate the second experiment with 1 cm mesh is around 2 days.

In conclusion both models simulate quite well the water height, whereas the pressure results are less precise. Even though, Smagorinsky model gets better results in the first experiment. In the second experiment, Smagorinsky model shows that it is able to follow the variability in time of the variables under study; however, there are parts in which the model underestimates the values while in others it overestimates the values. Regarding to the comparison, done in this experiment, between the pressure head and the water height, they are not equal when there is a highly curved water surface.

INDEX

1. INTRODUCTION AND OBJECTIVES.....	4
2. OPENFOAM	6
2.1. CASE STRUCTURE	6
2.1.1. <i>System</i>	6
2.1.2. <i>Constant</i>	7
2.1.3. <i>Time directories</i>	7
2.2. <i>INTERFOAM</i> SOLVER	8
2.2.1. <i>Phase-parameter</i>	8
2.3. FINITE VOLUME METHOD (FVM)	8
2.4. MESH	8
2.4.1. <i>General description</i>	9
2.4.2. <i>blockMesh</i> utility	10
3. GOVERNING EQUATIONS.....	11
3.1. TURBULENCE MODELS IN OPENFOAM	11
3.1.1. <i>LES models (Smagorinsky)</i>	13
3.1.2. <i>RANS models (RNG k-epsilon)</i>	13
3.1.3. <i>LES vs. RANS</i>	14
4. DAM-BREAK TESTS	16
4.1. DAM-BREAK TEST 1	16
4.1.1. <i>Experimental data</i>	16
4.1.2. <i>Numerical set-up in OpenFOAM</i>	17
4.1.2.1. <i>System</i>	17
4.1.2.2. <i>Geometry definition</i>	18
4.1.2.3. <i>Transport properties</i>	20
4.1.2.4. <i>Turbulence properties</i>	20
4.1.2.5. <i>Initial conditions</i>	20
4.1.2.6. <i>Data extraction</i>	21
4.2. DAM-BREAK TEST 2	22
4.2.1. <i>Experimental data</i>	22
4.2.2. <i>Numerical set-up in OpenFOAM</i>	24
4.2.2.1. <i>System</i>	24
4.2.2.2. <i>Geometry definition</i>	25
4.2.2.3. <i>Transport properties</i>	27
4.2.2.4. <i>Turbulence properties</i>	27
4.2.2.5. <i>Initial conditions</i>	27
4.2.2.6. <i>Data extraction</i>	28
4.3. MESH CONVERGENCE	29

5. RESULTS	31
5.1. DAM-BREAK TEST 1.....	31
5.1.1. <i>Smagorinsky</i>	32
5.1.1.1. Mesh discretization	32
5.1.1.2. Water height	33
5.1.1.3. Pressure	35
5.1.2. <i>RNG k-epsilon</i>	39
5.1.2.1. Mesh discretization	39
5.1.2.2. Water height	41
5.1.2.3. Pressure	42
5.1.3. <i>Comparison</i>	46
5.2. DAM-BREAK TEST 2.....	49
5.2.1. <i>Water height</i>	49
5.2.2. <i>Bottom pressure</i>	51
5.2.3. <i>Water height vs. pressure head</i>	52
5.2.4. <i>Velocity</i>	53
6. CONCLUSIONS AND RECOMMENDATIONS	55
REFERENCES.....	56

1. Introduction and Objectives

A dam is built to mitigate the effects of floods and control the river flow, especially in semiarid areas where during the dry seasons the natural flow is not enough to satisfy the industrial, agricultural and domestic demand. One of the problems associated with a dam is its breaking, producing a large dangerous situation for the downstream inhabitants and the environment. This is one of the reasons why the behavior of the water flow when the dam breaks is so important and many studies have been carried out during the last years.

The local regulations impose the compulsory study of the dam break to estimate the possible damages produced downstream, and with this data the dam is classified in different categories. Which are associated with the possible danger produced in case of its break and consequently with the safe factors that have to be used in its construction.

In the past, the numerical methods used to simulate this break were 1D or 2D. Nowadays the technological improvements towards incrementing computational power and storage capacity, allow us to perform 3D cases using the Navier-Stokes equations. These models need more storage and computational power to carry out the same study case. On the other hand, they are more precise than the old versions because do not assume hydrostatic pressure.

When the dam breaks, all the water in the reservoir is suddenly released and succumbs to the vertical acceleration of gravity, resulting in the hydrostatic pressure distribution model being invalid. Because there is not just the static pressure but also the dynamic pressure. According to that, the 3D numerical models are thought to be an extension to the 2D equations known as shallow water equations (SWE).

The present work studies and evaluates the numerical modeling of two 3D dam-break laboratory tests using the Finite Volume Method (FVM) of OpenFOAM. In the numerical method, the turbulence is simulated with the Large Eddy Simulation (LES) and the Reynolds Averaged Simulation (RANS), ending up with a comparison of both. The Volume of Fluid (VOF) is used to simulate the water's free surface. The experimental data for the calibration model is obtained from the Kleefsman et al. (2005) test, and for the validation model the Fraccarollo and Toro (1995) experimental data is used.

Therefore, the main objective of the present study is to evaluate the ability of the 3D *interFoam* solver to simulate the free surface and analyze the non-hydrostatic effects produced by the highly curved surface. This objective is carried out in the second experiment.

Moreover, an additional objective is to figure out if the Smagorinsky model (LES) and the RNG k-epsilon (RANS) simulate properly the different known variables of the first experiment. This is related to the fact that the mesh is an important factor to be

considered, which means that an optimal mesh has to be found, accounting for both accuracy and computational cost. Once the two models are simulated, a comparison between both is done in order to determine which one is better. After that, another experiment is simulated to validate the chosen mesh and the best model of both; checking if this model simulates properly the water height, the bottom pressure and the velocity field.

2. OpenFOAM

OpenFOAM (Open Field Operation And Manipulation) is a C++ toolbox used for Computational Fluid Dynamics (CFD). The code is a free, open-source software created by Henry Weller in 1989. It has been used across most areas of engineering and science, because it has an extensive range of features to solve complex fluid flows involving from chemical reactions, heat transfer, solid dynamics and electromagnetics, to turbulence, which is the important feature for this work (Openfoam.com, 2017). In this work the version 3.0.1 is used.

2.1. Case structure

The basic structure of an OpenFOAM case, before any simulation, is summed up at Figure 1.



Figure1 Case directory

2.1.1. System

System directory includes at least these three files: *controlDict*, *fvSchemes* and *fvSolution*. Figure 2 shows a snapshot of the files.



Figure 2 System directory

The *controlDict* file includes the control parameters; basically, the start and the end time of the simulation, the delta time of the simulation and the write interval (how often the program records the results). When choosing the write interval, it is important to consider that the Courant number must be smaller than 1 to obtain the full development of the simulation, else wrong results may be obtained. In other words, the delta time value is computed by the program and at the end of the simulation it is independent of the initial defined value. The important thing is that the write interval defined cannot be smaller than the stabilized delta time; if it were, the problems described above would happen.

The Courant number is defined in Equation 1.

$$Co = \frac{\delta t |U|}{\delta x} \quad (1)$$

Where:

δt is the time step.

$|U|$ is the speed through the cell.

δx is the cell size in the direction of the velocity.

The *fvSchemes* file defines the discretization schemes used, and the *fvSolution* file includes the equations that will be solved, tolerances and other algorithm controls.

2.1.2. Constant

The *Constant* directory includes the files that can be seen in Figure 3.



Figure 3 *Constant* directory

The *polyMesh* subfolder contains the *blockMeshDict* file (see Figure 4), which describes the mesh used at the simulation (its content will be explained at Section 2.4.).



Figure 4 *polyMesh* folder contents

The *g* file includes the gravity acceleration vector (9.81 m/s²).

The *transportProperties* file contains the viscosity and the densities of the fluids involved at the simulation.

The *turbulenceProperties* file defines the turbulent model used and all the constants and coefficients associated to it.

2.1.3. Time directories

The time directories consist of different folders, as many as writing times has the simulation. Each one includes data associated to its time step. To define the experiment, folder *0* includes the values and the boundary conditions at the initial time, before the simulation starts running. Once the simulation has started the other folders are automatically created by the program.

2.2. *InterFoam* solver

In this case of study, the *interFoam* solver is used. In the simulation both water and air are considered to be incompressible¹. This is the reason why this is the solver chosen, because it is a solver for two incompressible, isothermal immiscible fluids (multiphase flow). A phase-parameter is used to define where each fluid is.

2.2.1. Phase-parameter

The *interFoam* solver, which uses the Volume of Fluid (VOF), has a phase parameter that can take values from 0 to 1. Each extreme value is set to a fluid, and the intermediate values refer to a combination of both fluids, known as the interface zone.

2.3. Finite volume Method (FVM)

The FVM is used to discretize the problem. This means approximating the problem to discrete quantities. It is an alternative to the finite difference and finite element methods. There are three discretization ambits: spatial, equation and temporal.

Spatial discretization: Considering a discretization mesh of the fluid area, a control volume is built around each point of this mesh; resulting in a total volume that is the sum of all the considered control volumes.

Equation discretization: The differential equation that has to be solved is integrated at each control volume, giving a discretization version of this equation.

Temporal discretization: Defining the variation of the dependent variable between the nodes of the mesh.

The main property of the resultant discretization equation system is that the obtained solution satisfies exactly the conservative equations considered, independently of the mesh size (OpenFOAM, Programmer's Guide, 2015 and Niño, 2002).

2.4. Mesh

This section describes how the mesh is introduced to the OpenFOAM program, including a general description and a particular explanation of the utility used by the program (OpenFOAM, Programmer's Guide, 2015 and OpenFOAM, User's Guide, 2015).

¹ Air can be considered as incompressible because all the air in the atmosphere is taken into account, and the compression at the atmospheric scale is negligible.

2.4.1. General description

The mesh must satisfy certain criteria to ensure a valid and accurate solution, because it is part of the numerical solution. During the run, OpenFOAM checks if the mesh satisfies a set of constraints and will cease running if any constraint is not satisfied. By default, OpenFOAM generates arbitrary polyhedral cells in 3D, bounded by arbitrary polygonal faces; known as *polyMesh*. However, a different mesh can be created.

A set of faces defines the mesh, which are defined by points. A point is a location in a 3D space, defined by a vector in units in meters. All the points are compiled in a list, and each point has a numerical label according to its position, starting from zero. A face is an ordered list of points, referred to by their labels. The ordering is such that each two neighboring points are connected by an edge. The direction of the face normal vector is defined by the right-hand rule (see Figure 5). The faces are compiled in a list, like the points; each face has a numerical label according to its position in the list, starting from zero.

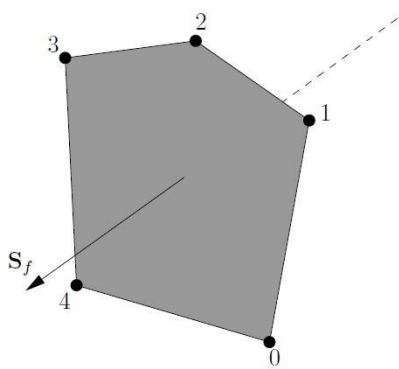


Figure 5 Face normal vector (OpenFOAM, User's guide, 2015)

There are two kinds of faces: internal faces and external. The difference between both is the amount of cells that they connect. The internal faces connect two cells; the numbering is such that the face normal vector points into the cell with the larger label, whereas the external faces belong only to one cell, because they coincide with the boundary of the domain. In that case the face normal vector must point outside of the computational domain. A list of external face labels creates a patch; while a list of patches creates a boundary and it has to be associated to a boundary condition. The Table 1 shows the different possible boundary conditions.

Selection Key	Description
<i>patch</i>	Generic patch
<i>symmetryPlane</i>	Plane of symmetry
<i>empty</i>	Front and back planes of a 2D geometry
<i>wedge</i>	Wedge front and back for an axially-symmetric geometry
<i>cyclic</i>	Cyclic plane
<i>wall</i>	Wall; used for wall functions in turbulent flows
<i>processor</i>	Inter-processor boundary

Table 1 Basic patch types (adapted from OpenFOAM, User's guide, 2015)

Cells are a list of faces in arbitrary order; the easiest shape to fulfill the properties required is a cube, the only property that has to be checked is that the cells must cover the entire domain and they cannot overlap each other (convexity, closure and orthogonality are the other properties).

2.4.2. *blockMesh* utility

The *blockMesh* utility creates parametric meshes with grading and curved edges defined in the *blockMeshDict* dictionary file that was introduced previously. By reading this dictionary, *blockMesh* generates the mesh and writes out the mesh data to *points*, *faces*, *cells* and *boundary* as can be seen at Figure 6. These are the components that are described before.



Figure 6 *polyMesh* folder contents after the use of the *blockMesh* utility

At *blockMeshDict* the different points are defined with its coordinates, known as vertices. If some edge is not straight a list of these non-straight edges must be defined from the different kinds of lines. Once the vertices are defined, a list named *block* is created, each block is defined by a list of vertex labels such that the first vertex defines the origin of the local coordinate system, the second with the first define the direction x_1 and the third with the second define the direction x_2 . These three vertexes with the fourth define the plane $x_3=0$. The fifth vertex is found by moving from the first vertex in the x_3 direction. The other three vertexes are found by moving in the same directions as in the plane $x_3=0$. As the vertexes are defined, so are the edges, indirectly, so the order of the vertex must be logic. After the vertexes, a vector giving the number of cells and the cell expansion ratios required in each direction are defined.

The expansion ratio enables the mesh to be graded, or refined in specified directions. The expansion can be specified at the axis directions by *simpleGrading* or at each edge by *edgeGrading*. If no expansion wants to be done, a vector with ones must to be written.

3. Governing Equations

The water and air flows can be determined by the Navier-Stokes equations. Assuming that those flows are incompressible, the equations become simpler because the density is a known function of position. Equations 2 and 3 define the continuity equation for incompressible flows and the momentum equation, respectively (Reddy, 2008).

$$\nabla \cdot \mathbf{v} = 0 \quad (2)$$

$$\mu \nabla^2 \mathbf{v} - \nabla p + \rho \mathbf{f} = \rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) \quad (3)$$

Where:

\mathbf{v} is the velocity vector.

μ is the viscosity.

p is the pressure.

ρ is the density.

\mathbf{f} is the body force vector.

t is the time.

In most of the cases, it is an impossible task to find an exact solution of the Navier-Stokes equations. The main reason is the non-validity of the principle of superposition, because of the nonlinearity of the equations. To resolve that inconvenience the turbulent models appear, their function being to introduce more information in order to solve the equations. This information is obtained modelling the flows, meaning that it is a non-exact solution.

3.1. Turbulence Models in OpenFOAM

For most engineering applications it is not necessary to obtain the full detailed turbulent flow, but in some other turbulence effects can be relevant in the hydraulic description, so this is why the turbulence models can be used. They allow obtaining the mean flow without having to calculate the full-dependent flow field (which should be solved beforehand). In other words, the turbulence models are used to close the momentum and the continuity equation presented above.

The turbulent flow consists of different scales or eddy scales. Each scale has a characteristic velocity and length, known as velocity and length scale, respectively. The region covered by a large eddy might enclose smaller eddies. The smallest eddies are dissipated by viscous forces into thermal energy resulting in a temperature increase, whereas the largest eddies are of the order of the flow geometry. It is also known that a turbulent flow results from a high Reynolds number; in a boundary layer the transition occurs at a Reynolds number of 500 000 (Davidson, 2016).

As mentioned before, the turbulent flow is three-dimensional and unsteady, that is the reason why the 3D models are used in this work to simulate the turbulent flow of the dam-break. In this point, it is where the different turbulence models appear. Depending on the precision and on the amount of equations that have to be solved different models are distinguished.

The ones that are used in this work are the Smagorinsky of the LES model and the Re-Normalisation Group (RNG k-epsilon) of the RANS model. The main different between both is that LES models resolve more equations and only model the small eddies, while the RANS equations solve simply the large-scale-eddies (Bakker, 2002).

The large eddies are very much affected by the boundary layer. Therefore, it is important the near-wall treatment. The value that decides if a wall function is needed is the wall distance units(y^+). When its value is smaller than 30 no wall function is needed. Figure 7 shows a representation of the different eddies depending on the distance from the wall. The value of y^+ is obtained by Equations 4 and 5 (Guerrero, 2015). The wall functions impose a velocity profile at the buffer layer and the viscous sublayer. At OpenFOAM to obtain that value the function *yPlus* is used for RANS models. LES models do not need wall functions.

$$y^+ = \frac{\rho \times U_\tau \times y}{\mu} \quad (4)$$

$$U_\tau = \sqrt{\frac{\tau_\omega}{\rho}} \quad (5)$$

Where:

ρ is the density.

U_τ is the friction velocity.

y is the distance to the first cell center normal to the wall.

τ_ω is the wall shear stresses.

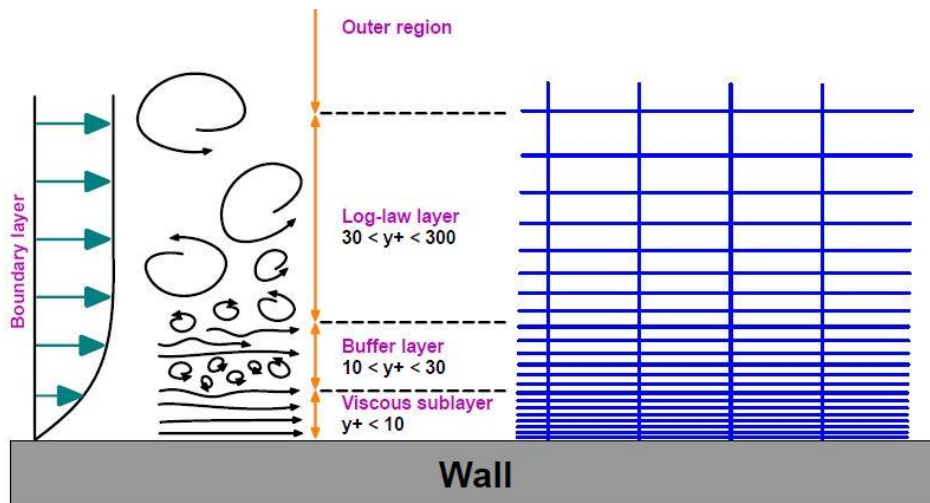


Figure 7 No wall-function (Guerrero, 2015)

3.1.1. LES models (Smagorinsky)

LES models are based on space-filtered equations where time dependent calculations are done. These models calculate explicitly the large eddies whereas the effect of the small eddies on the flow pattern is taken into account with a subgrid model. In this case the Smagorinsky SGS model is used and the filter-width is taken as the local grid size.

The turbulent or eddy viscosity (ν_{sgs}) is obtained through dimensional analysis with Equations 6, 7 and 8 (Davidson, 2016 and Efron, 2006).

$$\nu_{sgs} = (C_s \Delta)^2 |\bar{\bar{S}}| \quad (6)$$

$$|\bar{\bar{S}}| = (2\bar{S}_{ij}\bar{S}_{ij})^{1/2} \quad (7)$$

$$\Delta = (\Delta V_{IJK})^{1/3} \quad (8)$$

Where:

$|\bar{\bar{S}}|$ is the three-dimensional gradient.

C_s is a flow-dependent constant and can vary in the range from 0.065 to 0.25.

Δ is the filter-width taken as the local grid size.

3.1.2. RANS models (RNG k-epsilon)

In RANS models, as it is previously briefly introduced, part of the turbulence is resolved and part of it is modelled. The total turbulence is the sum of both (Davidson, 2016).

The RNG k-epsilon is a turbulence model for compressible and incompressible flows with the following assumptions: The turbulence statistics are homogeneous when the mean strain S_{ij} is not too large. The fluctuating velocity field u is governed by the Navier-Stokes equations driven by a random force, chosen in an order such that the global properties of the resulting field are the same as those in the flow driven by the mean strain S_{ij} (Yakhot, Orszag, Thangam, Gatski and Speziale, 1992).

Equations 9, 10 and 11 summarize the main results of the procedure.

$$\frac{\partial U_i}{\partial t} + \mathbf{U} \cdot \nabla U_i = -\nabla_i p + \frac{\partial}{\partial x_j} \left[\nu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right] \quad (9)$$

$$\frac{\partial \bar{K}}{\partial t} + \mathbf{U} \cdot \nabla \bar{K} = -\bar{\epsilon} - \bar{\epsilon}_{ij} S_{ij} + \frac{\partial}{\partial x_i} \left(\alpha_k \nu \frac{\partial \bar{K}}{\partial x_i} \right) \quad (10)$$

$$\frac{\partial \bar{\varepsilon}}{\partial t} + \mathbf{U} \cdot \nabla \bar{\varepsilon} = -C_{\varepsilon 1} \frac{\bar{\varepsilon}}{\bar{K}} \bar{\tau}_{ij} S_{ij} - C_{\varepsilon 2} \frac{\bar{\varepsilon}^2}{\bar{K}} - 2\nu_o S_{ij} \frac{\partial U_i}{\partial x_i} \frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial}{\partial x_i} \left(\alpha_\varepsilon \nu \frac{\partial \bar{\varepsilon}}{\partial x_i} \right) \quad (11)$$

Where:

U is the mean velocity.

p is the pressure.

ν is the total viscosity.

\bar{K} is the mean turbulent kinetic energy.

$\bar{\epsilon}$ is the mean turbulent dissipation rate.

$\bar{\tau}_{ij}$ is the deviatoric part of the Reynolds stress tensor.

S_{ij} is the mean strain.

α_k is a proportionality constant.

C_{ε_1} is a constant (1.42).

$C_{\varepsilon 2}$ is a constant (1.68).

ν_o is the molecular viscosity.

α_ε is a proportionality constant.

3.1.3. LES vs. RANS

A short summary of the difference between the two models is described below. As it is said previously, RANS models all turbulent scales whereas LES only models small, isotropic turbulent scales. Consequently, LES is much more expensive in computational time than RANS. Figure 8 shows a summary of the difference related with what is simulated and what is not in each turbulence model.

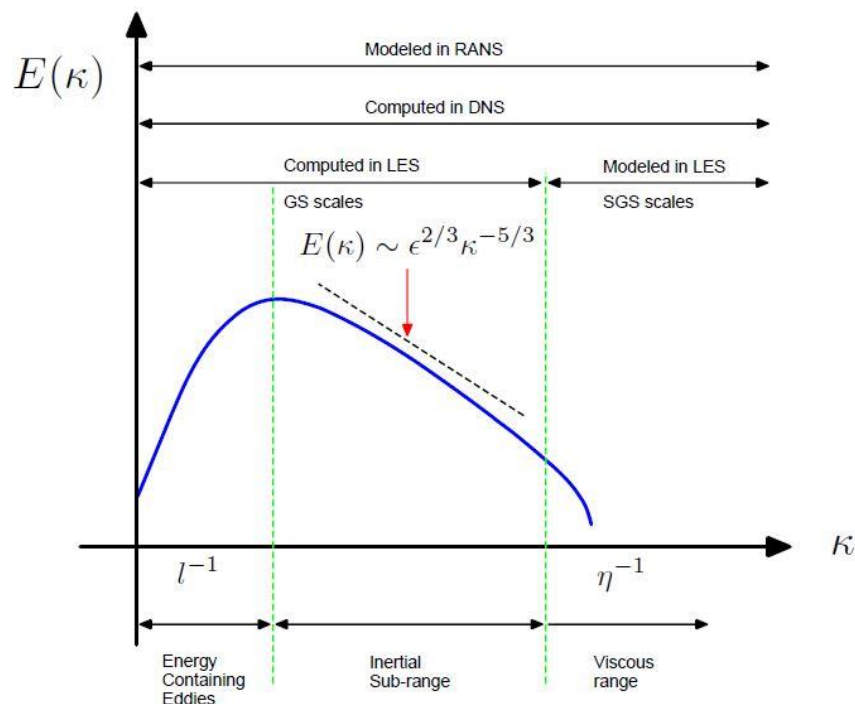


Figure 8 Energy spectrum for a turbulent flow. Log-log scales. (Guerrero, 2015)

LES also has some restrictions. For instance, it is only used for 3D domains and unsteady time domains while RANS can handle 2D or 3D domains and steady or unsteady time domains.

Concerning to space discretization LES is a central scheme whereas RANS is a 2nd order upwind. Another difference is that LES is a 2nd order time discretization and RANS 1st.

LES turbulence model has zero or one equation while RANS has more than two equations. Theoretically, RANS results should be less accurate, because it models in the same way all the eddy scales; while the small scales tend to depend only on viscosity, the large ones are affected by the boundary conditions. This will be contrasted at the end of this work.

4. Dam-break Tests

Two different tests are performed in this work. The first is used to figure out which turbulent model is better to simulate the experimental data. It is also used to define the grid mesh that offers the best balance between computational cost and accuracy.

The second test is used to check that the mesh and the turbulent model selected, from the previous test, are also valid, and the model can simulate properly the experimental data. The simulated data provided it is used to analyze the non-hydrostatic effects produced by the highly curved surface.

4.1. Dam-break test 1

4.1.1. Experimental data

The first experiment (Kleefsman, Fekken, Veldman, Iwanowski, Buchner, 2005) is obtained from the Maritime Research Institute Netherlands (MARIN) and it consists of a large open roof tank of $3.22 \times 1 \times 1$ m. The right part of the tank, that will contain 0.55 m water height at the beginning, is closed by a door. This door is supposed to be pulled up instantaneously by releasing a weight. In addition to all this, a box is included downstream; in this case it is supposed to represent a scale model of a container on the deck of a ship. This dam-break test has an easy set-up; no inflow or outflow conditions are needed. There is only a certain amount of water resting on a side of the tank at the start of the experiment. When the experiment starts the water is released and the water flows into the entire tank.

The measurements, which are recorded every 0.001 s from 0 s to 7.394 s, are water height at four different locations along the x-axis of the tank and the pressure on eight points of the box. Figure 9 shows the domain and the location of the four vertical probes and the eight pressure sensors, four on the front of the box and four on the top.

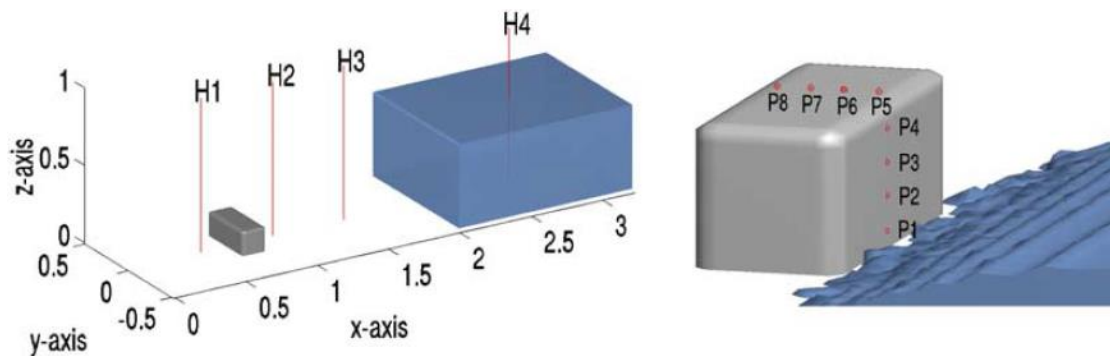


Figure 9 Measurement positions for water heights and pressures (Kleefsman et al., 2005)

Figure 10 shows a snapshot of the video from the experiment at time 0.4 seconds. At the top-right, the smaller picture shows the water in the reservoir.



Figure10 Snapshot of the experiment at time 0.4 s (Kleefsman et al., 2005)

4.1.2. Numerical set-up in OpenFOAM

The following sections describe the set-up introduced in OpenFOAM. This conditions are valid for both turbulence models, Smagorinsky and RNG k-epsilon, unless stated otherwise.

4.1.2.1. System

The *controlDict* file of the simulation contains the information that was introduced at *section 2.1.1.* The first thing that has to be defined is the application that will be used; as it has already been mentioned, the application used in this simulation is the *interFoam*.

The simulation starts at time zero and ends at time 8 s. There are few more seconds than the experimental data, but it is just to end at an integer number. Delta t value is the initial one; after some calculations the program itself will fix another delta t in order to fulfil the Courant number condition by a given mesh; the maximum Courant number is set as 1. In this case the initial value is chosen to be equal to 0.0001 s. The write interval is set to ensure that the equations are fully developed and the results obtained are good; therefore, the chosen value is 0.05 s.

At *fvSchemes* the *GaussinterfaceCompression* is chosen to compress the interface between water and air.

In addition to the basic documents of the *System* directory, another one is added: *setFieldsDict*. It is used to define the initial water box. Water value is set as 1 and air as 0. Figure 11 shows the water box in the domain.

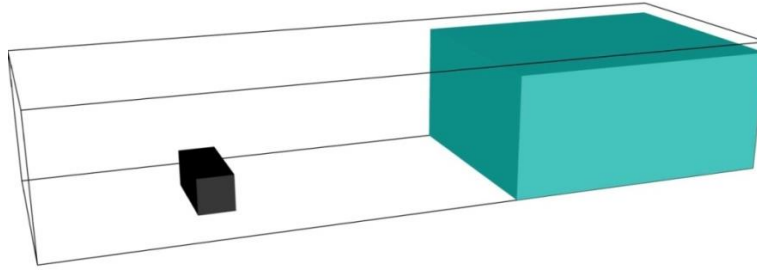


Figure 11 Water box in the domain

4.1.2.2. Geometry definition

The geometry is defined by the OpenFOAM program itself, as it has been explained at the mesh *section 2.4.*

To define the geometry, the computational time is considered. Meaning that, the vertical height of the domain is reduced as much as possible to reduce the computational cost. The chosen height is 0.60m, a few more centimetres than the initial water height at the reservoir (0.55 m) and a few less than the experimental domain (1 m). Figure 12 shows the domain dimensions implemented at OpenFOAM.

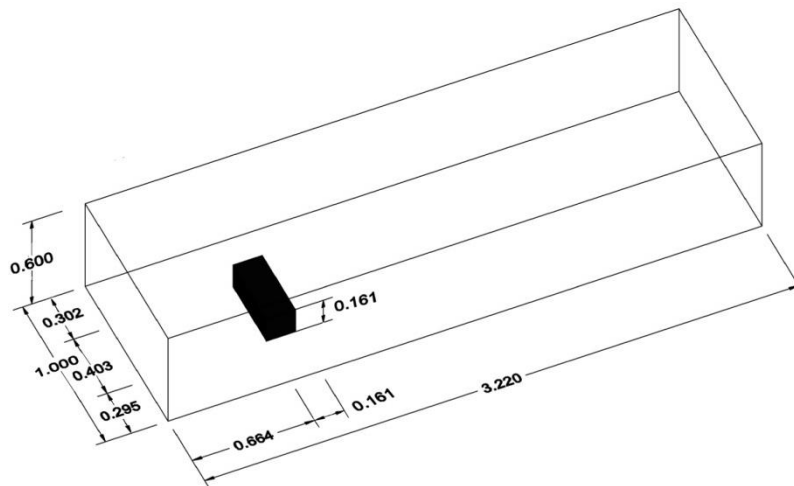


Figure 12 Dimensions of the domain

The *blockMeshDict* file includes the points that will define the different vertexes of each block. Afterwards, the different blocks that define the domain can be specified, reminding the criteria explained previously. Figure 13 shows the different blocks that form the domain.

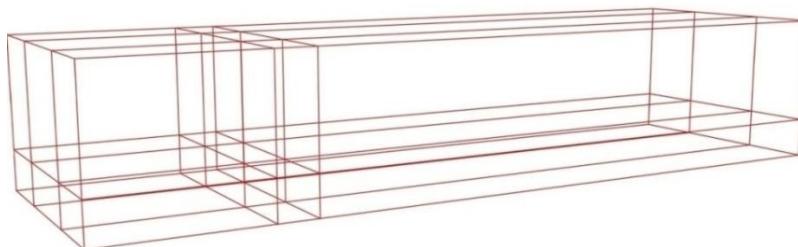


Figure 13 Domain divided by blocks

Each block, in addition to its vertices, has a certain number of divisions in each local axis direction. Said number of divisions will depend on the required mesh size and on the length of the block in the axis direction that is considered. To summarize, the number of divisions will be different for the 1 cm mesh, the 1.5 cm mesh or the 2 cm mesh. For example; if an 18 cm length block is considered, the number of divisions for a 1 cm mesh will be 18; for the 1.5 cm mesh will be 12 and 9 divisions to get a 2 cm mesh.

Regarding the expansion ratio, it is defined as *simpleGrading (1 1 1)* in all cases, because a uniform mesh all along the block length is desired.

Following the sections of the mesh document, the external faces are defined. These are the ones that border the domain. They have to be defined such that the normal vector, following the right-hand rule, points out of the domain.

Figure 14 shows the different faces that are defined and the associated conditions. The faces that have the *wall* condition, do not allow fluid to go through them. Moreover, this condition allows to implement the wall functions, which are required at the RNG k-epsilon simulation because y^+ is bigger than 30 in the entire domain. The top face is defined with the general condition of *patch*, because specific boundary conditions (atmospheric conditions) will be applied on it. Table 2 has a short description of the different types of faces.

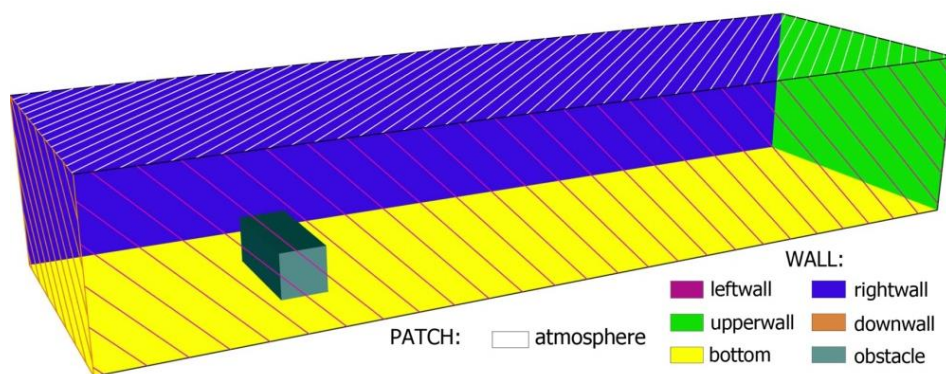


Figure 14 Face legend

Face name	Description
<i>leftwall</i>	Left wall
<i>rightwall</i>	Right wall
<i>upperwall</i>	Upstream wall
<i>downwall</i>	Downstream wall
<i>bottom</i>	Bottom surface
<i>obstacle</i>	Obstacle walls
<i>atmosphere</i>	Atmospheric conditions

Table 2 Face types

Figure 15 shows the 2 cm mesh domain with the initial state of the water in blue.

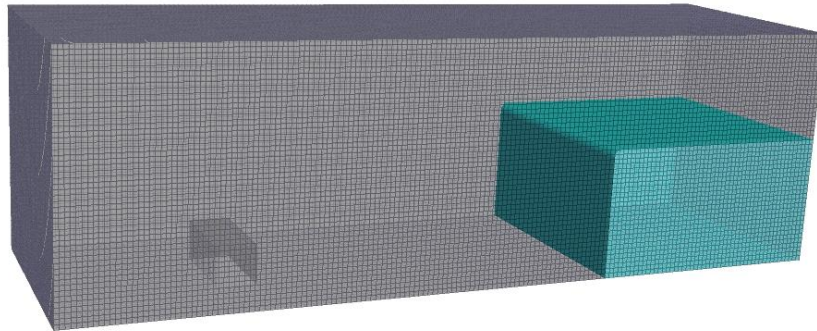


Figure 15 Geometry of the OpenFOAM simulation (2 cm mesh)

4.1.2.3. Transport properties

At the *transportProperties* file the viscosity and the density of water and air are set. Table 3 sums up the values.

Fluids	ν (m ² /s)	ρ (kg/m ³)
Water	$1 \cdot 10^{-6}$	1000
Air	$1.48 \cdot 10^{-5}$	1

Table 3 Transport properties

4.1.2.4. Turbulence properties

At the *turbulenceProperties* file the desired turbulence model is defined, be it Smagorinsky or RNG k-epsilon.

4.1.2.5. Initial conditions

At folder *0* the different initial conditions are set. These are *nut* (turbulent viscosity field), *p_rgh* (pressure), *k* and *epsilon*. The main difference between Smagorinsky and RNG k-epsilon approaches is that RNG k-epsilon uses wall functions and Smagorinsky does not.

The files differentiate between the *internalField* and the *boundaryField*. The second refers to the boundary faces defined at the *blockMeshDict* and the other to the rest of the domain. All the files have a uniform value at the internal field (0 at *nut* and *p_rgh* files, $1.8 \cdot 10^{-5}$ at *k* file and $3.18 \cdot 10^{-4}$ at *epsilon* file).

At the Smagorinsky *nut* file, the *ZeroGradient* is set at all the faces of the boundary field except the *atmosphere* face that it is calculated and its initial value is a uniform value of zero. *ZeroGradient* means that the gradient of this variable is zero, meaning that the actual value is constant.

At the RNG k-epsilon *nut* file, the *nutkWallFunction* is set at all the faces of the boundary field with a uniform value of zero.

At *p_rgh* file, the *FixedFluxPressure* option is set at all the *wall* faces with a uniform value of zero, because there is no flux of pressure at the walls. At the *atmosphere* face the option *totalPressure* is chosen with a uniform value of 0.

At the Smagorinsky *k* file, the *ZeroGradient* option is used at the *wall* faces with a uniform value of $1.8 \cdot 10^{-5}$. At the *atmosphere* face the *inletOutlet* option is chosen, with a uniform *inletValue* of $1.8 \cdot 10^{-5}$. The *inletOutlet* option is used because this face allows entrance and exit of fluid.

At the RNG k-epsilon *k* file, the *kqRWallFunction* option is used at the *wall* faces with a uniform value of $8.5 \cdot 10^{-4}$. At the *atmosphere* face the *inletOutlet* option is chosen, with a uniform *inletValue* of $1.8 \cdot 10^{-5}$. The *inletOutlet* option is used because this face allows entrance and exit of fluid.

At Smagorinsky *epsilon* file, the *ZeroGradient* option is chosen at the *wall* faces with a uniform value of $3.18 \cdot 10^{-4}$. At the *atmosphere* face the *inletOutlet* option is used, with a uniform value of $3.18 \cdot 10^{-4}$.

At RNG k-epsilon *epsilon* file, the *epsilonWallFunction* option is chosen at the *wall* faces with a uniform value of $3.18 \cdot 10^{-4}$. At the *atmosphere* face the *inletOutlet* option is used, with a uniform value of $3.18 \cdot 10^{-4}$.

4.1.2.6. Data extraction

Water heights are obtained at locations H1, H2, H3 and H4 specified at Figure 16. To extract these data, the tool *ParaView*² is used. By differentiating the alpha values smaller or bigger than 0.5 the interface between air and water is obtained and therefore the water height.

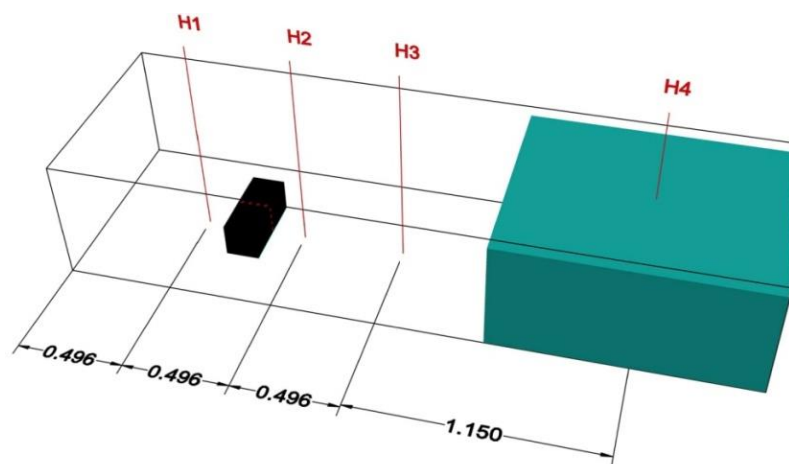


Figure 16 Height locations

² ParaView is an open-source, multi-platform data analysis and visualization (Bu.edu, 2017).

Pressure data is obtained at the same locations as the experiment, P1, P2, P3, P4, P5, P6, P7 and P8, specified at Figure 17. The points 1 to 4 are located at the upstream face of the obstacle and the rest of them at the top of the obstacle. To extract this data, the tool *probeLocations*³ is used.

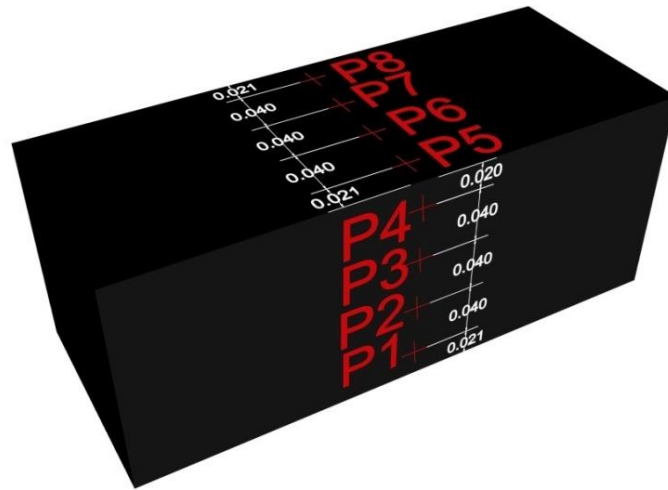


Figure 17 Pressure locations

Both data are extracted every 0.05 s.

4.2. Dam-break test 2

4.2.1. Experimental data

The second experiment (Fraccarollo and Toro, 1995) has a more complex structure in order to simulate different slopes downstream of the reservoir; in this work only the results obtained with slope equal to zero will be used. Another difference from the first one is that in this experiment the water is not released along the total width of the reservoir, and instead through a centred 0.40 m width gate activated by a pneumatic cylinder. In addition, the downstream plane has open boundaries on all three sides.

Initially 0.6 m in height of water is located in a 1 x 2 x 0.8 m reservoir upstream. The dimensions of the total flume (reservoir included) are 3 m long and 2 m width. Figure 18 shows a top view and a side view of the experimental set-up respectively.

³*probeLocations* is an OpenFOAM tool.

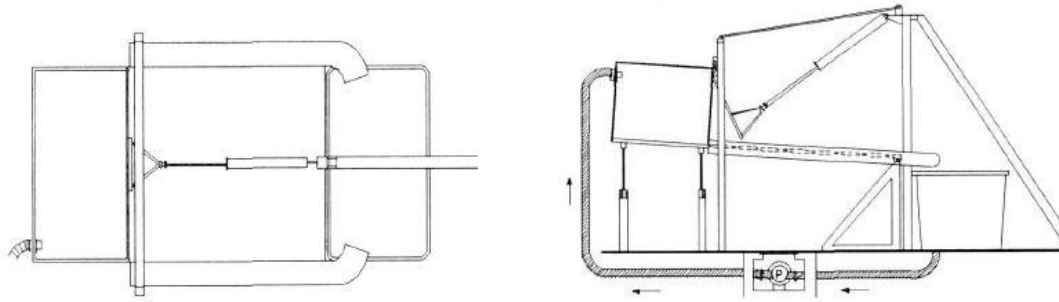


Figure 18 Schematic diagrams of the experimental set-up (Fraccarollo and Toro, 1995)

The measurements obtained from the experiment are water depths, pressure and velocities at different locations shown at Figure 19. It includes a plan view and a table with the coordinates of the different points.

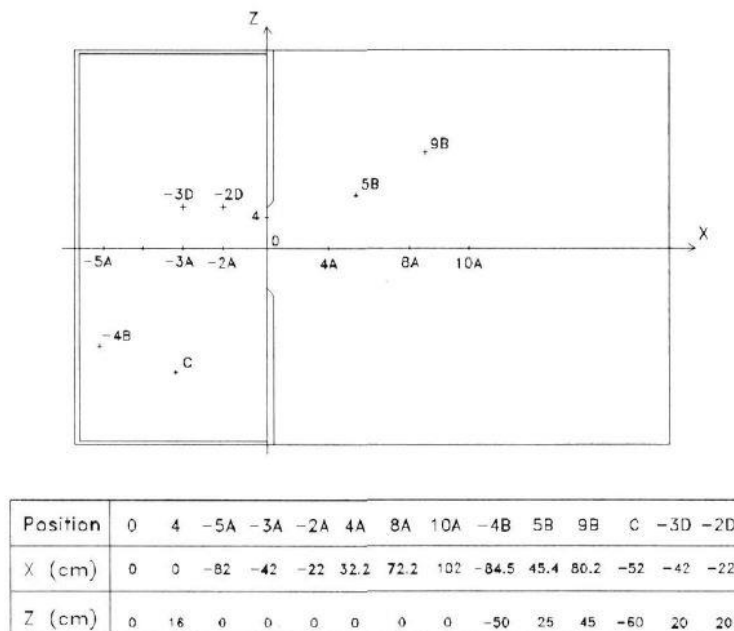


Figure 19 Probe positioning with reference to the Cartesian plan $\{x, z\}$ (Fraccarollo and Toro, 1995)

The water depths are measured by using wave height meters, the pressures at the bottom of the channel by using pressure gauges, and the velocity by means of an electromagnetic velocity meter.

The first instrument consists of a pair of stretched 0.3 mm diameter nickel/chrome wires, placed parallel to each other at a distance of about 1.0 cm, linked to an electrical circuit; an analogical signal is obtained measuring the electric conductivity of the water depth between the wires. Due to some calibration problems, their use is not allowed outside the tank, except at the position "8A". Therefore, in the places where the wave height meters are not installed, through the pressure gauges an indirect measure of the water depth is obtained by the hydrostatic assumption.

The pressures are obtained by the BHL 4310 low range pressure transducer, produced by IMO Transinstruments.

Finally, the x and z velocity components are measured by using E.M.S. gauge. It is a four quadrant electromagnetic velocity meter produced by Delft Hydraulics, which works in a range from 0.0 to 2.5 m/s.

The velocity data is obtained every 5 cm depth at each point. Afterwards the average over the depth is done. This data is the one obtained to compare the results with the simulation.

The accuracy of the three instruments described above is $\pm 1\%$ f.s..

The data of this experiment is obtained from the different graphs of Fraccarollo and Toro (1995) by a web that digitalizes a given plot (Arohatgi.info, 2017). The duration of the experiment is from 0 to 10 seconds.

4.2.2. Numerical set-up in OpenFOAM

In this simulation the same configuration as before is used. To sum up, the main differences are the geometry, the simulation time and the probe locations. Moreover, in this case only one turbulence model and one mesh are used. Only the differences will be written down.

4.2.2.1. System

The *controlDict* file of the simulation contains the information that is introduced at the previous *section 2.1.1.*. The simulation starts at time zero and ends at time 10 s. Delta t value is the initial one; after some calculations the program itself will fix another delta t in order to fulfil the Courant number condition by a given mesh; the maximum Courant number is set as 1. In this case the initial value is chosen to be equal to 0.005 seconds. The write interval is set to ensure that the equations are full developed and the results obtained are good, therefore the chosen value is 0.05 seconds.

At the *setFieldsDict* the initial water box is defined. Following the same criteria as before, water value is set as 1 and air as 0. Figure 20 shows the water box in the domain.

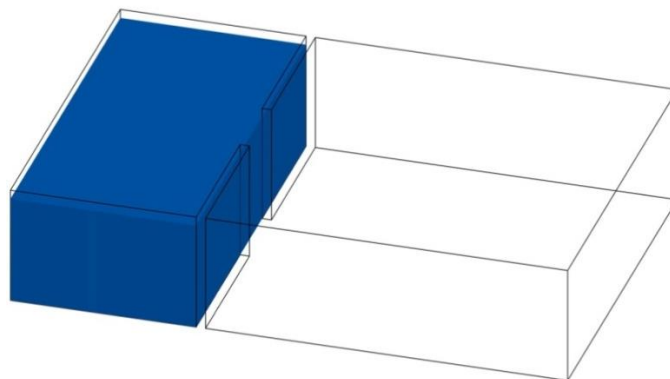


Figure 20 Water box in the domain

4.2.2.2. Geometry definition

The geometry is defined by the OpenFOAM program itself, as it has been explained at the mesh *section 2.4.*

To define the geometry, the computational time is considered, meaning that the vertical height of the domain is reduced as much as possible to reduce the computational cost. The chosen height is 0.65 m, a few more centimetres than the initial water height at the reservoir (0.6 m) and a few less than the experimental domain (0.8 m). Figure 21 shows the domain dimensions implemented at OpenFOAM.

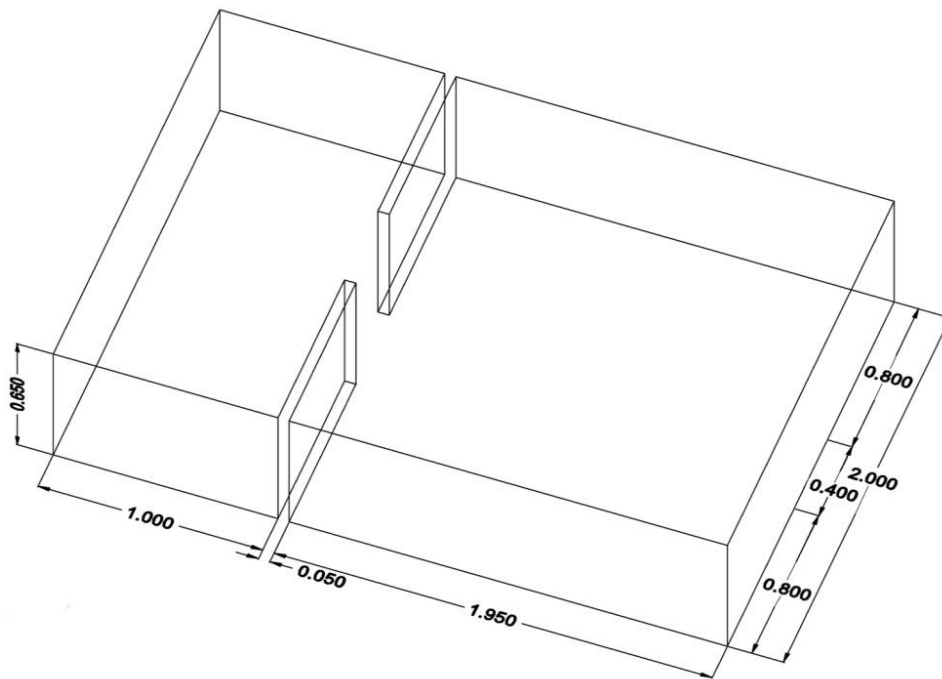


Figure 21 Dimensions of the domain

The *blockMeshDict* file includes the points that will define the different vertexes of each block. Afterwards, the different blocks that define the domain can be specified, following the criteria explained above. Figure 22 shows the different blocks that form the domain.

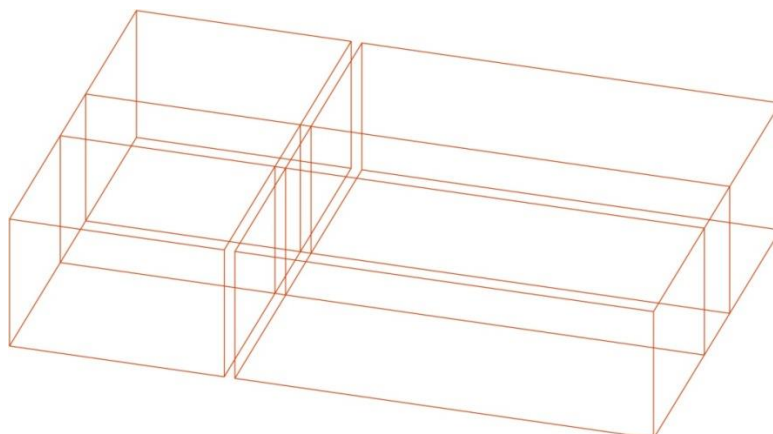


Figure 22 Domain divided by blocks

Each block, in addition to its vertices, has the number of divisions needed in each local axis direction to get a grid size of 1 cm. The number of division will depend on the length of the block in the axis direction that is considered.

Following the sections of the mesh document, defined previously; the external faces are defined. These are the ones that are the border of the domain. They have to be defined such that the normal vector, following the right-hand rule, points out of the domain.

Figure 23 shows the different faces that are defined and the associated conditions. The faces that have the *wall* condition do not allow fluid to go through them. The wall faces are the green and yellow ones. The top face is defined with the general condition of *patch*, because specific boundary conditions (atmospheric conditions) will be applied on it. Moreover, the blue faces are defined as a patch as well, because they allow fluid to go through them. Table 4 has a short description of the different types of faces.

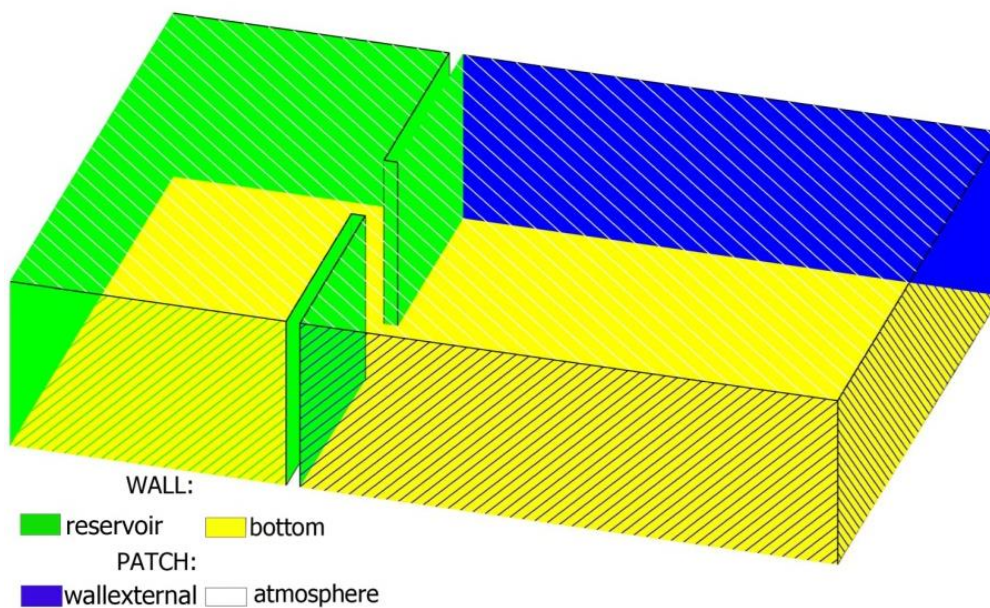


Figure 23 Face legend

Face name	Description
reservoir	reservoir walls
bottom	bottom surface
wallexternal	outlet, no-walls
atmosphere	atmospheric conditions

Table 4 Face types

Figure 24 shows the 1 cm mesh domain.

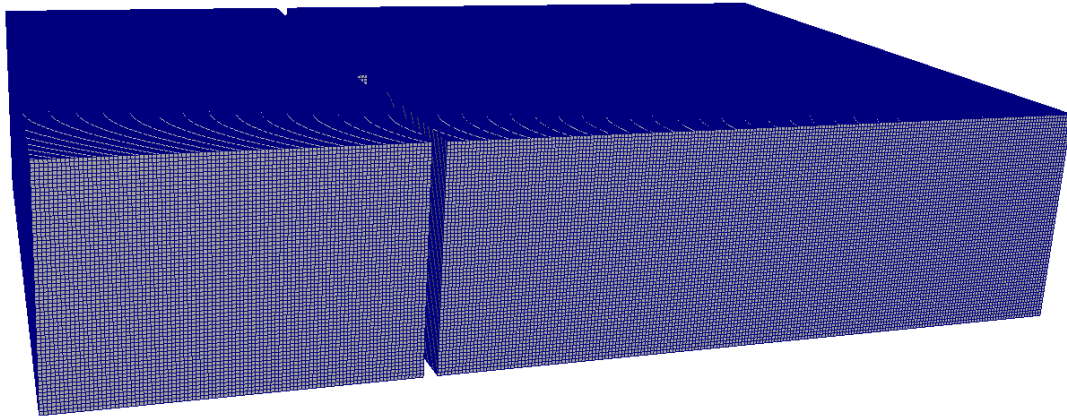


Figure 24 Geometry of the OpenFOAM simulation (1 cm mesh)

4.2.2.3. Transport properties

The same values of the other simulation are used in the transport properties file, which are summed up at Table 5.

Fluids	ν (m ² /s)	ρ (kg/m ³)
Water	$1 \cdot 10^{-6}$	1000
Air	$1.48 \cdot 10^{-5}$	1

Table 5 Transport Properties

4.2.2.4. Turbulence properties

In this simulation only the Smagorinsky turbulence model is used.

4.2.2.5. Initial conditions

At folder 0 the different initial conditions are set. These are *nut* (turbulent viscosity), *p_rgh* (pressure), *U*, *k* and *epsilon*. The same values used for the Smagorinsky at the previous simulation are used. Despite that, they are defined here.

The files differentiate between the *internalField* and the *boundaryField*. The second refers to the boundary faces defined at the *blockMeshDict* and the other to the rest of the domain. All the files have a uniform value at the internal field (0 at *nut* and *p_rgh* files; $1.8 \cdot 10^{-5}$ at *epsilon* file; $3.18 \cdot 10^{-4}$ at *k* file and a zero vector at *U* file).

At *nut* file, the *zeroGradient* is set at all the faces of the boundary field with a uniform value of zero except the *atmosphere* and *wallexternal* faces, where the value is calculated from a uniform initial value of zero.

At *p_rgh* file, the *FixedFluxPressure* option is set at all the *wall* faces with a uniform value of zero. At the *atmosphere* face the option *totalPressure* is chosen with a uniform value of 0.

At *U* file, the *fixed value* option is set at all the *wall* faces with a uniform value of a zero vector, because there is no movement of the water along the walls. At the *atmosphere* face the option *pressureInletOutletVelocity* is chosen with a uniform value of a zero vector. At the *wallexternal* faces the *inletOutlet* option is chosen with a uniform inlet value of zero.

At *k* file, the *zeroGradient* option is used at the *wall* faces with a uniform value of $8.5 \cdot 10^{-4}$. At the *atmosphere* and *wallexternal* faces the *inletOutlet* option is chosen, with a uniform *inletValue* of $3.18 \cdot 10^{-4}$. The *inletOutlet* option is used because this face allows entrance and exit of fluid.

At *epsilon* file, the *zeroGradient* option is chosen at all the *wall* faces with a uniform value of $1.8 \cdot 10^{-5}$. At the *patch* faces the *inletOutlet* option is used, with a uniform value of $1.8 \cdot 10^{-5}$.

4.2.2.6. Data extraction

Water heights are obtained at locations 0, 4, -5A, 8A and C specified at Figure 25. To extract these data, the *ParaView* tool is used. By differentiating the alpha values smaller or bigger than 0.5 the interface between air and water is obtained and therefore the water height.

Bottom pressure is obtained at locations 0, 4, -5A and 8A, specified at Figure 25. To extract the data, the tool *probeLocations* is used.

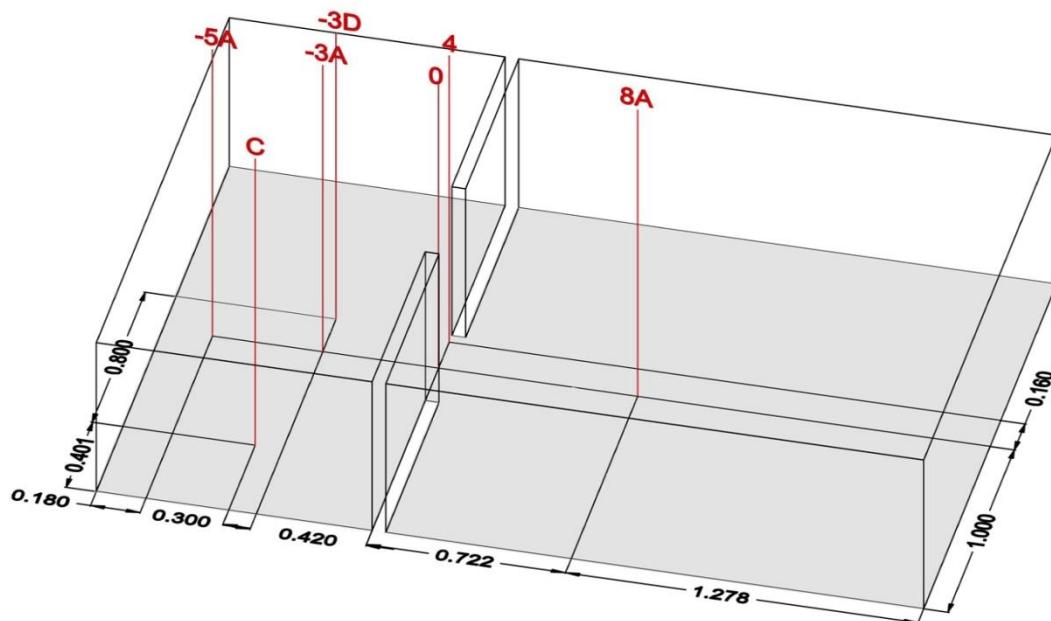


Figure 25 Probe locations

Velocity data is obtained at the same locations as the experiment; the longitudinal mean velocity component at -3D, 0 and -3A locations; the transversal mean velocity component at the location -3D and the longitudinal velocity at 0 location (only at the heights of 0.05 m and 0.4 m) specified at Figure 25. To extract this data, the tool *ProbeLocations* is used.

All data is extracted every 0.05 s.

4.3. Mesh convergence

As previously explained, the mesh is determining when numerical methods are used, because depending on the mesh size, it can result in a poor simulation or in a huge computational cost. The first problem is caused by too large of a mesh size and the second one due to the fact that there is a too fine mesh.

In this work a cubic mesh is used. Three different mesh sizes are studied: 2 cm, 1.5 cm and 1 cm. In order to choose the one that gives the best balance between accuracy and computational cost the Convergence Grid Index (CGI) is obtained for the different meshes and models.

The idea of this index is to find out when the result is independent of the mesh size, because with the Navier-Stokes equations, that are non-linear problems strongly influenced by boundary conditions, it is difficult to prove its convergence analytically. Nevertheless, the experimental data can be used at the analytical result.

The meshes do not have the same refinement factor defined in Equation 12. This refinement factor (r) needs to be greater than 1.3 (Schwer, 2008 and Celik, Ghia, Roache, Freitas, Coleman, Raad, 2008).

$$r = \frac{h_{coarse}}{h_{fine}} \quad (12)$$

Where:

h is the mesh size.

Because the refinement factor is not constant, Equations 13 to 18 may be used to define the order of convergence (p) and the GCI.

$$p = \frac{|\ln|f_{32}/f_{21}| + q(p)|}{\ln r_{21}} \quad (13)$$

$$f_{ij} = f_i - f_j \quad (14)$$

$$q(p) = \ln \left(\frac{r_{21}^p - s}{r_{32}^p - s} \right) \quad (15)$$

$$s = \text{sign}(f_{32}/f_{21}) \quad (16)$$

$$GCI_{21} = F_s \frac{e_{21}}{r_{21}^p - 1} \quad (17)$$

$$e_{21} = \left| \frac{f_1 - f_2}{f_1} \right| \quad (18)$$

Where:

f is the grid solution. ($i > j$)

F_s is a safety factor, whose value is 1.25 based on experience.

5. Results

The results of the two experiments described above are explained down here.

5.1. Dam-break test 1

The first experiment, as it is detailed above, consists of a tank with some water on one extreme that is released. The following explanation is divided on the two turbulence models and a last part with a comparison between them. The mesh analysis is done for each turbulence model.

Although the data is obtained until 8 seconds; it is observed that from around 4 s until the end of the simulation there is a phase difference. This may be produced by the second wave. Figure 26 shows the water height at H2, the experimental value and both simulated values, Smagorinsky and RNG k-epsilon. It can be clearly seen that from 4.5 seconds the simulations have a phase difference compared with the experimental value, even though the general shape is well simulated for both models.

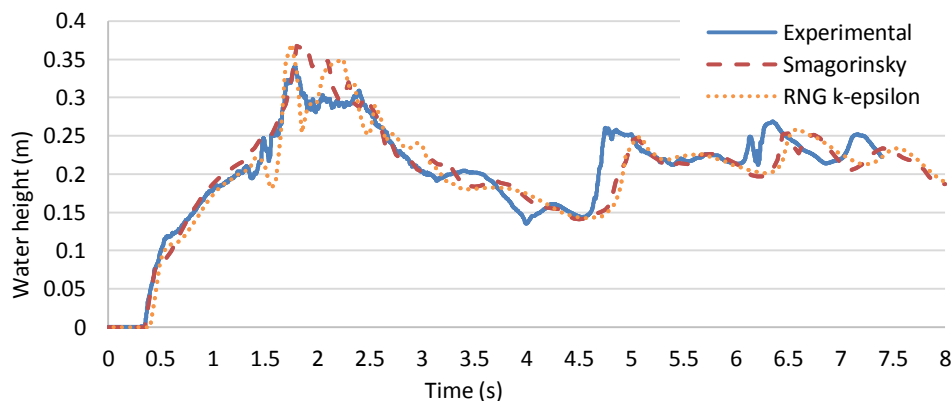


Figure 26 H2 water height values (1 cm mesh)

This phase difference is reduced as the grid size is smaller. Figure 27 shows the interval from 4 s to 6 s of the previous probe for two different meshes (1 cm and 2 cm) and both models.

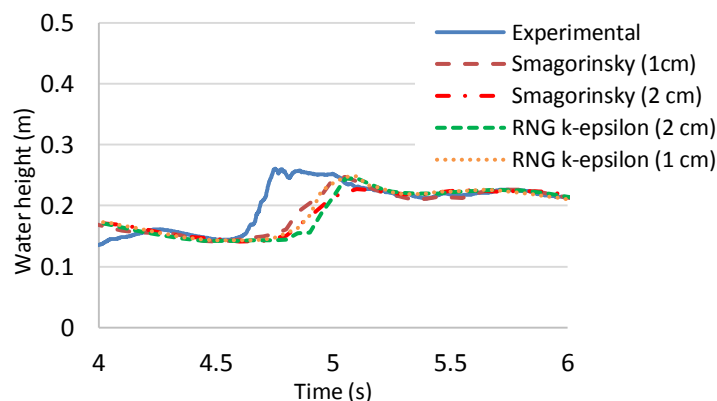


Figure 27 H2 values from 4 to 6 s

According to the information provided above, the following analysis will only take into account values from 0 to 4 s.

5.1.1. Smagorinsky

5.1.1.1. Mesh discretization

This modelling is computed for three different mesh sizes (2 cm, 1.5 cm and 1 cm). In order to choose the optimal mesh, a balance between computational cost and precision must be achieved; the water height for every simulated mesh in meters obtained at each known point is compared to the experimental value at the same point. To compare the results of the different meshes the following plots are done (see Figures 28 to 31), having the experimental data at the abscissa axis and the simulated data at the ordinate axis; both in meters. In addition, a lineal tendency line is plotted; its coefficient of determination (R^2) is shown at Figure 32. The closer to 45 degrees the tendency line is, the more exact are the results; because it means that the experimental data and the simulated data have the same value. In other words, if the slope is 1 both data have the same value. Moreover, the closer to one the R^2 value is the less variance in the data.

The legend of the symbols that appear on the plots is at Table 6.

Symbol	Mesh size
◆	1 cm
■	1.5 cm
▲	2 cm

Table 6 Legend

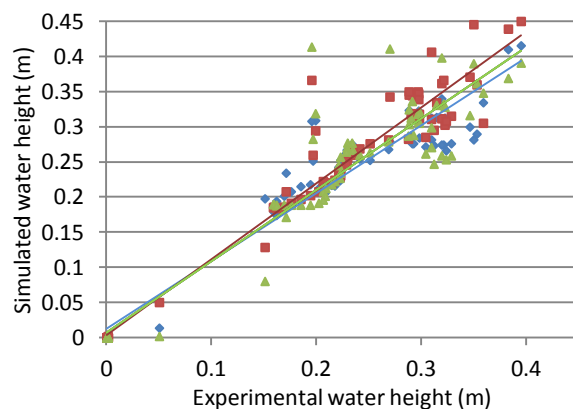


Figure 28 H1 water height values

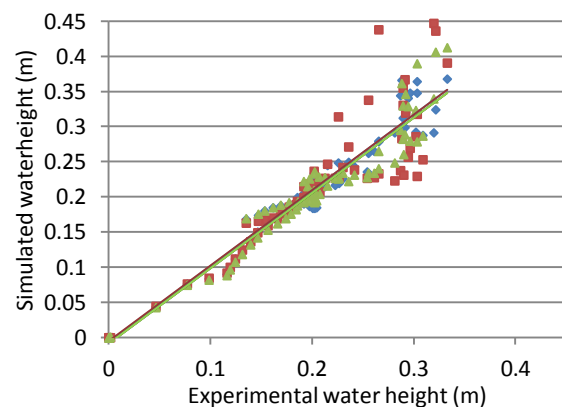


Figure 29 H2 water height values

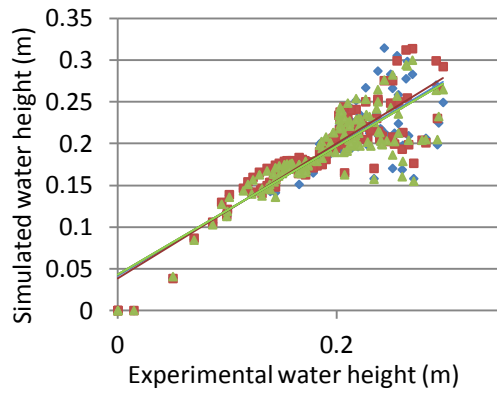


Figure 30 H3 water height values

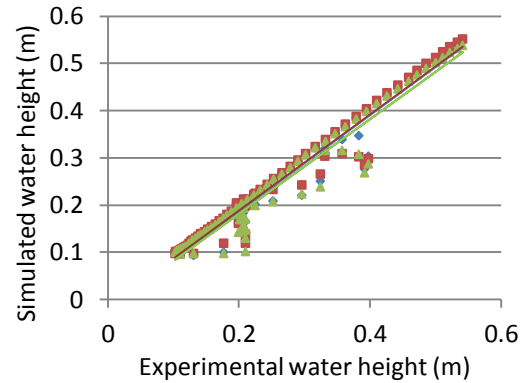


Figure 31 H4 water height values

At most of the Figures the three mesh sizes have almost the same linear tendency line slope, only Figure 28 has different slopes for each mesh size. In this case the finest mesh has the slope closer to 45 degrees.

Having a look on their R^2 a small oscillation can be noticed, see Figure 32, the R^2 does not have a clear tendency and it is increasing and decreasing depending on the probe location and the mesh size (Δx); that means that the values have an oscillatory convergence (60 % to 85 % values have a negative solution at the Equation 16 meaning that they have an oscillatory convergence). Moreover, the results of GCI are not small, meaning that the grid does not converge.

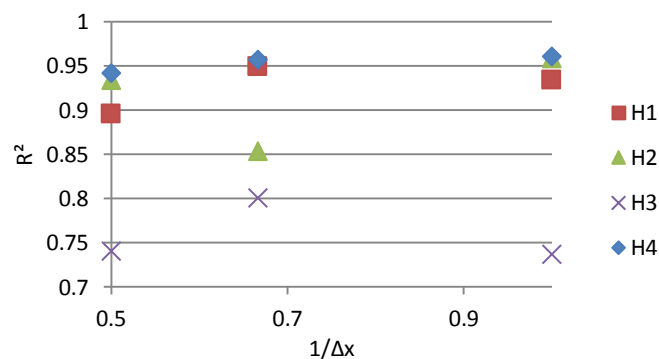


Figure 32 R^2 of water height values

Except the H3 probe location which has a very small R^2 , among the other probe locations the mesh that has a bigger R^2 is the finest mesh. A possibly reason why the values at H3 are less precise is because it is where there are highest water slopes after the water release.

The chosen mesh is 1 cm because it is the one that in most of the data points gives a better approximation and a bigger R^2 . With this mesh the results are analyzed.

5.1.1.2. Water height

Figures 32 to 36 show different water heights of the data points in meters. At the X axis there is the time in seconds, it takes values from zero to 4 s. At the Y axis there are

the water heights of the corresponding points in meters, including the experimental data (continuous line) and the simulated data (dotted line).

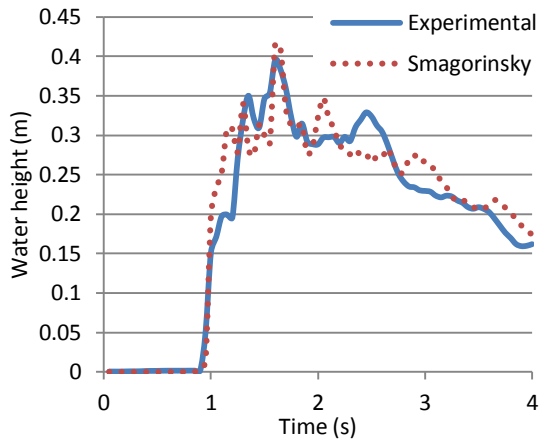


Figure 33 H1 water height values

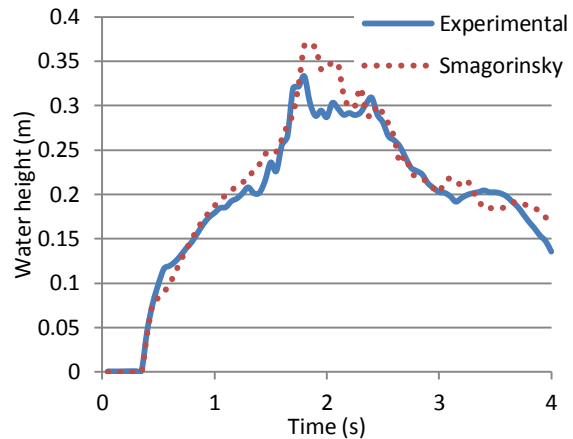


Figure 34 H2 water height values

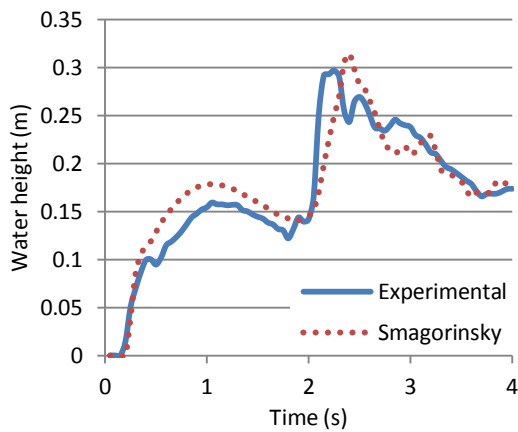


Figure 35 H3 water height values

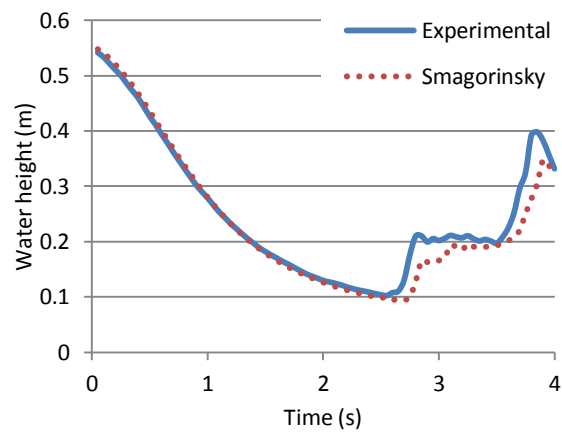


Figure 36 H4 water height values

Generally, the Smagorinsky model simulates quite well the general shape of the water level. Specifically, at each probe location there is some strength and some weaknesses.

At H1 location, the farthest one from the reservoir, there is a little of overestimating values when the first wave rebounds at the wall and interacts with the slower water that goes through the obstacle. This imprecision might be produced by the difficulty of recording precisely the water heights when there is this interaction. This can also be the reason why between 2 and 3 s the simulated values are more different than the experimental, because the water is still stabilizing after the crash at the end wall.

At H2 the simulated data is almost the same as the experimental data. There are only a few differences around the 2 seconds, when the simulation tends to overestimate the results. This might be a problem related with the way that the water level is extracted, because the program might take into account the highest water level, whereas the experimental data takes into account the water surface and neglects the possible overhang of the waves.

At H3 there is a general overestimation at the begging, even though the shape is captured. At the last two seconds of the simulation there is less agreement and the peak is a little bit out-of-step.

At H4 the simulation is almost perfect while the level is decreasing, once it starts to grow up again there is a small phase difference and the data is underestimated. Despite that, the shape is perfectly simulated.

Looking at Figures 28 to 31, the conclusion is that the mean water height values are well simulated whereas at peak values there is more variance.

5.1.1.3. Pressure

The pressure results at the upstream face of the obstacle are shown at Figures 37 to 40. At the X axis there is the time in seconds, it takes values from zero to 4 s. At the Y axis there are the pressures in pascals of the corresponding points, including the experimental data (continuous line) and the simulated data (dotted line).

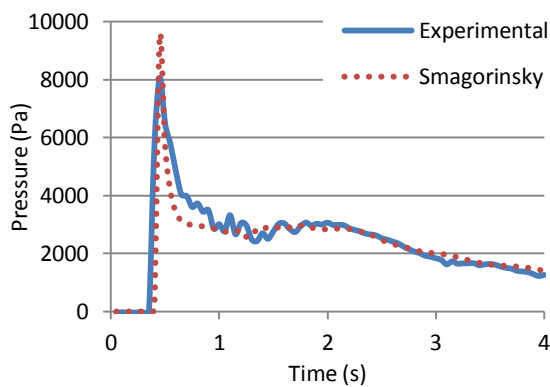


Figure 37 P1 pressure values

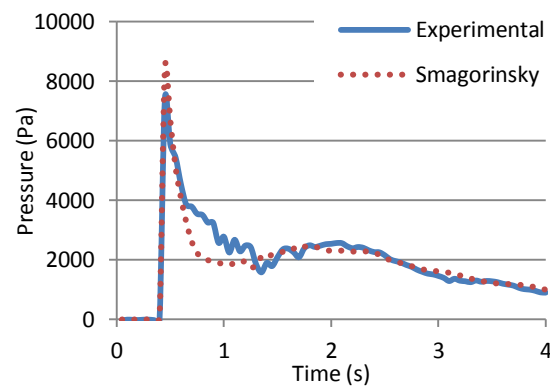


Figure 38 P2 pressure values

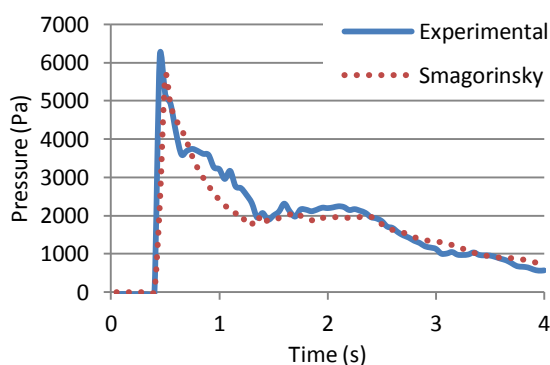


Figure 39 P3 pressure values

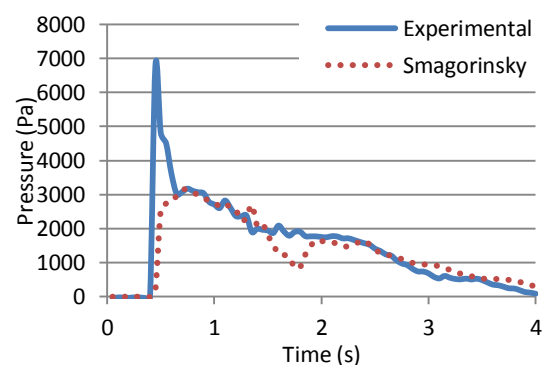


Figure 40 P4 pressure values

At P1 the peak is simulated later and overestimated. After the peak the data is underestimated and from 1 s the simulated values seem to represent the mean value of the experimental data without if the small up and down values are ignored. The last two seconds are perfectly simulated by the Smagorinsky model.

At P2 something happens similar to P1, but the peak has no phase difference, it is less overestimated and the decrease of the pressure after the peak is better simulated until around 0.5 s, where it is more underestimated than in P1. Then the last two seconds and a half are quite well simulated, having almost the same values in most of the remaining simulation.

At P3 the peak is a little bit underestimated and with a small phase difference; after that, follows the same criteria as the other two probe locations. But the last seconds are less exact than the two previous (P1 and P2).

At P4 the Smagorinsky simulation does not reach the peak value, even though the values after the peak are similar to the experimental data, except one negative peak produced before the two seconds. Maybe this reduction of the pressure at the simulation is induced by the suction produced by the rebounded wave going over the obstacle.

Figures 41 to 44 relate the experimental data and the simulated data. At the abscissa axis there are the experimental values and the simulated values are in the ordinate axis. Both data are in pascals.

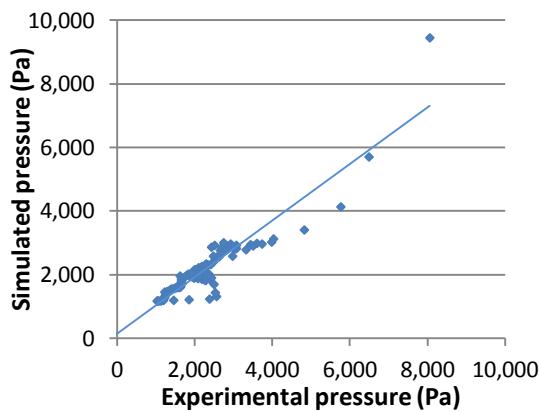


Figure 41 P1 pressure values

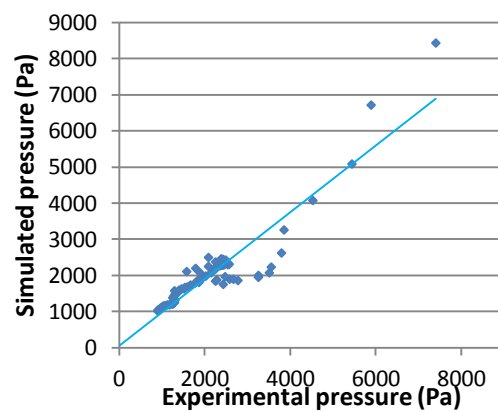


Figure 42 P2 pressure values

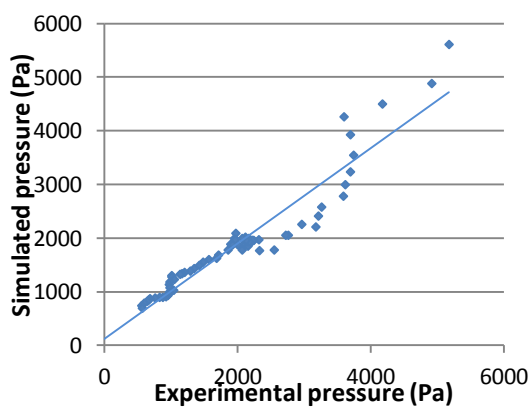


Figure 43 P3 pressure values

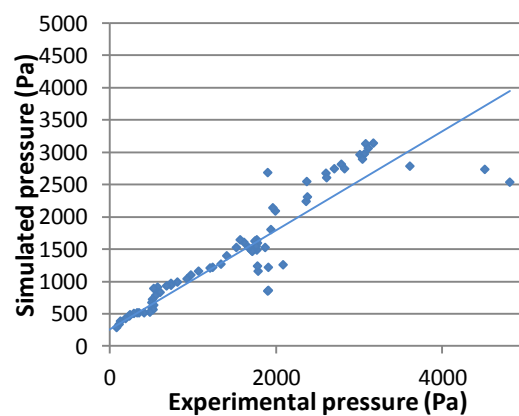


Figure 44 P4 pressure values

Approximately at all the probe locations the values smaller than 2000 Pa are well simulated, following the linear tendency line, which almost have 45 degrees (except

P4, which its linear tendency line denotes that the values are underestimated). Only few values, the biggest than 2000 Pa are outside the tendency line. Most of them are underestimated except a few of them (one in P1, three in P2 and five in P3). Note that at P4 the values between 2000 and 3000 Pa approximately are in the imagine 45 degrees' line.

To sum up the information of the upstream face of the obstacle, it can be said that generally speaking the Smagorinsky model simulates well enough the general shape and tends to overestimate the peak pressures close to the bottom and underestimate the peaks close to the surface as can be seen at Figures 37 to 40.

In the same way that the other pressure values, the pressure results at the top face of the obstacle are shown at Figures 45 to 48. At the X axis there is the time in seconds, it takes values from zero to 4 s. At the Y axis there are the pressures in pascals of the corresponding points, including the experimental data (continuous line) and the simulated data (dotted line).

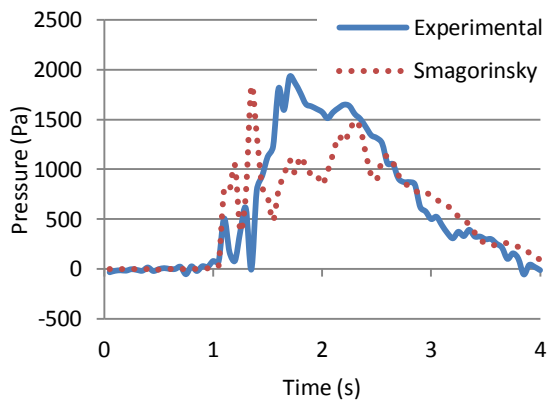


Figure 45 P5 pressure values

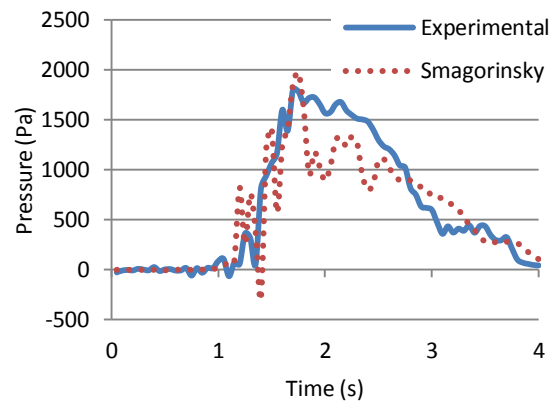


Figure 46 P6 pressure values

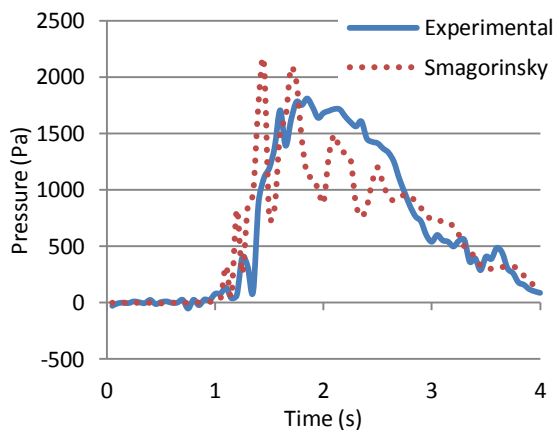


Figure 47 P7 pressure values

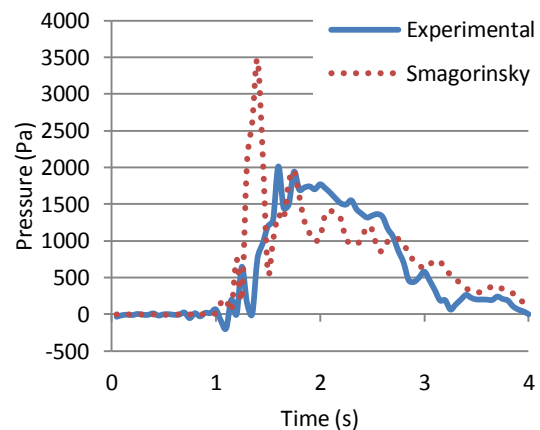


Figure 48 P8 pressure values

At first glance, it can be easily seen that this pressure results are less precise than the others. Even though, the general shape is still obtained.

When the wave reaches the P5 probe location the values are overestimated until approximately 1.5 seconds when they become underestimated during 1 second and after that the last second and a half are in broad terms well simulated.

Something similar happens at P6 probe location, but in this case the first seconds after the wave reaches that point the simulated values are less overestimated, and they are more similar to the experimental values. Although once the highest pressure value is reached the simulated values are smaller than the experimental, as happens at P5.

P7 results are similar to P5, the first results are overestimated and then the results follow the same shape (with the peaks more pronounced) but underestimating the results. At the last second and a half the results are quite similar to the experimental data.

P8 experimental data have a similar behaviour than the others, but when the wave reaches its location the experimental data records a huge peak, that the experimental data does not have.

Figures 49 to 52 relate the experimental data and the simulated data. At the abscissa axis there are the experimental values and the simulated values are in the ordinate axis. Both data are in pascals.

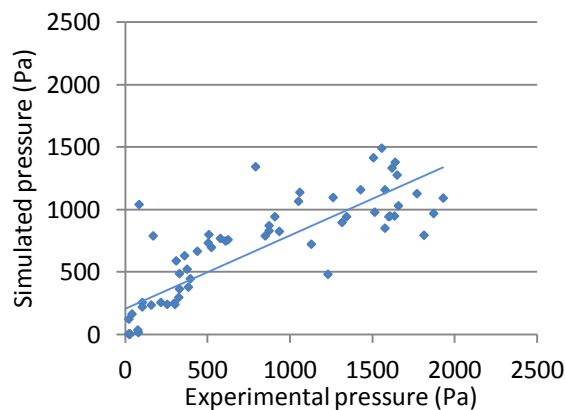


Figure 49 P5 pressure values

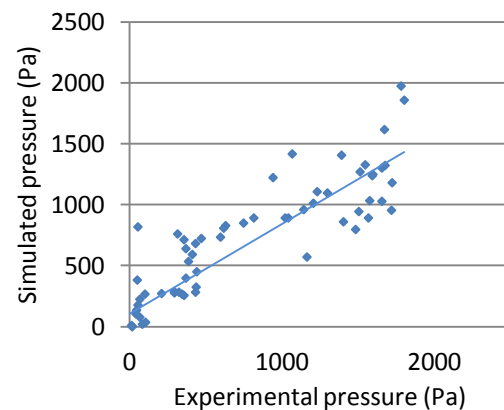


Figure 50 P6 pressure values

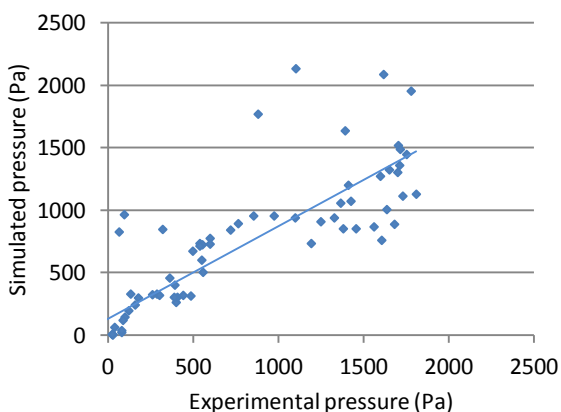


Figure 51 P7 pressure values

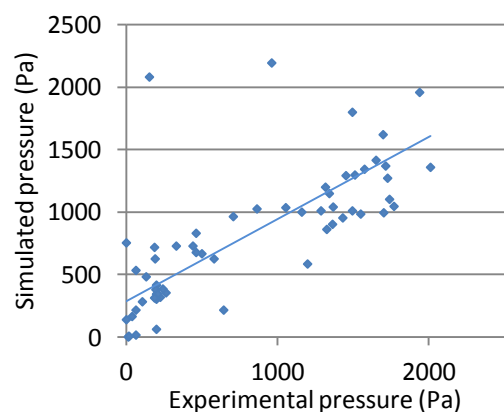


Figure 52 P8 pressure values

Approximately at all the probe locations the smallest values, are best simulated that the others. Between 500 Pa and 1000 Pa the simulated values are overestimated, whereas in broad terms the biggest results are underestimated.

To sum up the information of the top face of the obstacle, it can be said that generally speaking the Smagorinsky model simulates quite well the general shape and tends to overestimate the first peak produced by the first wave and underestimated the values after it, as it can be seen at Figures 45, 46, 47 and 48.

5.1.2. RNG k-epsilon

5.1.2.1. Mesh discretization

This model is computed for three different mesh sizes as before (2 cm, 1.5 cm and 1 cm). In order to choose the optimal mesh, a balance between computational cost and precision must be achieved; the water height in meters obtained at the different known points is compared to the respectively experimental data for the different meshes.

To compare the results of the different meshes, the following plots are done (see Figures 53 to 56), having the experimental data at the abscissa axis and the simulated data at the ordinate axis. Both data are in meters. In addition, a lineal tendency line is plotted; its coefficient of determination (R^2) is shown at Figure 57. The closer to 45 degrees the tendency line is, the more exact are the results; because it means that the experimental data and the simulated data have the same value. In other words, if the slope is 1 both data have the same value. Moreover, the closer to one the R^2 value is, the less variance there is in the data.

Table 7 has the legend of the symbols that appear on the plots.




Symbol	Mesh size
	1 cm
	1.5 cm
	2 cm

Table 7 Legend

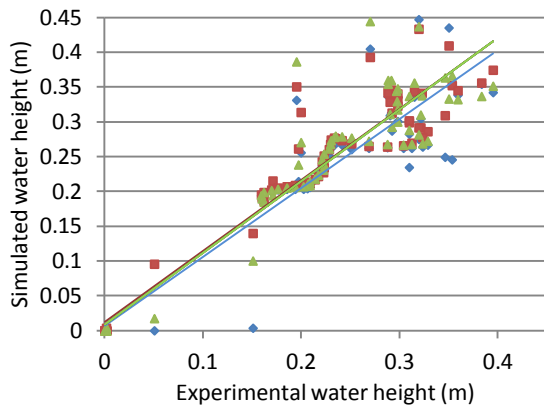


Figure 53 H1 water height values

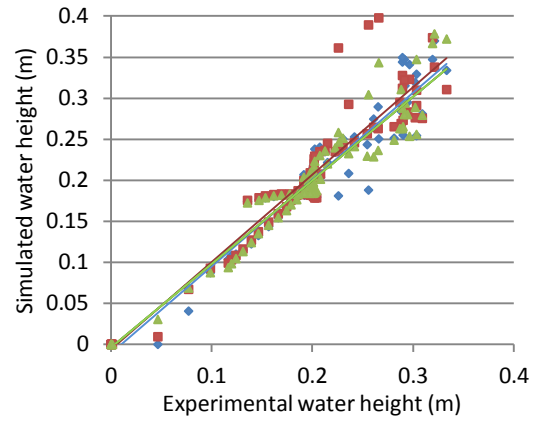


Figure 54 H2 water height values

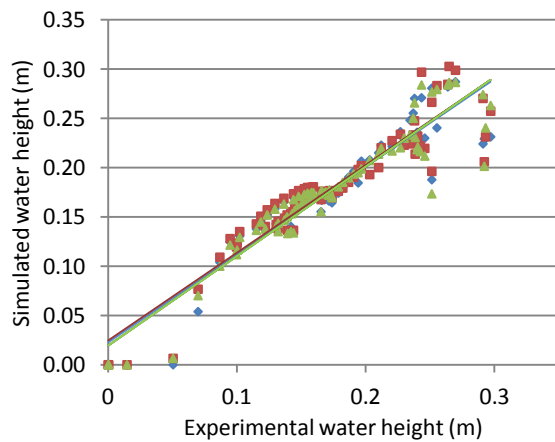


Figure 55 H3 water height values

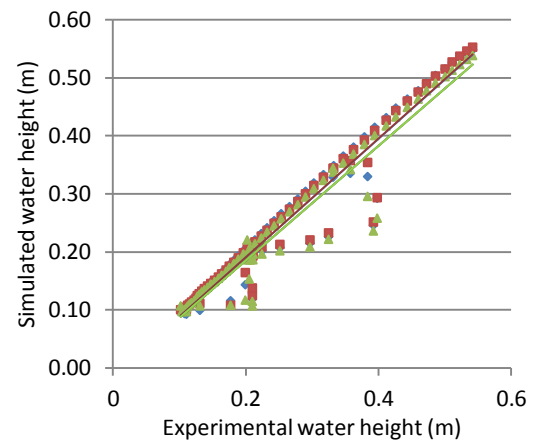


Figure 56 H4 water height values

At most of the Figures the three mesh sizes have almost the same linear tendency line slope. The 1 cm mesh slope at Figure 53 is closer to 45 degrees than the other coarser meshes, which their tendency lines are steeper, meaning that the simulated values are overestimated.

At the probe location H2 the finest mesh has the best tendency line in terms of slope. Then the other two meshes, one overestimate the values (1.5 cm mesh) and the other underestimate the results (2 cm mesh). This might be an indicator of the existence of oscillatory convergence; with the results of the Equation 16 this will be checked.

At H3 location all the meshes provide similar results. Therefore, the linear tendency lines are almost the same at the three cases.

The results inside of the reservoir (H4) are almost along the 45 degrees' line, independently of the mesh. Resulting in a nearly three equal tendency lines, the difference is that the slope of the 2 cm mesh tendency line is less than 45, but very close to it.

Having a look on their R^2 a small oscillation can be noticed, see Figure 57, the R^2 does not have a clear tendency and it is increasing and decreasing depending on the probe location and the mesh size (Δx); that means that the values have an oscillatory

convergence (35 % to 65% values have a negative solution at the Equation 16 meaning that they have an oscillatory convergence). Moreover, the results of GCI are not small, meaning that the grid does not converge.

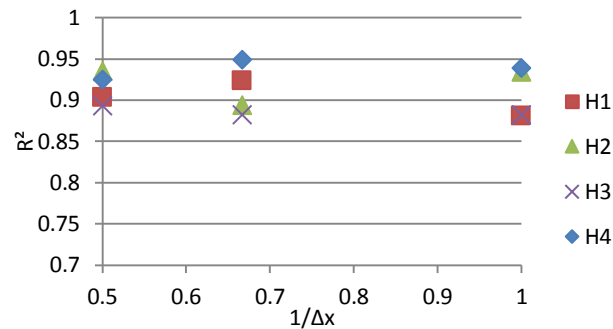


Figure 57 R² of water height values

The chosen mesh is 1 cm because is the one that in most of the data points gives a better approximation (slope closer to 45 degrees) and a bigger R^2 (except the location H1). With this mesh the results are analyzed.

5.1.2.2. Water height

Figures 58 to 61 show the water heights of the four data points. At the X axis there is the time in seconds, it takes values from zero to 4 s. At the Y axis there are the water heights of the corresponding points in meters. Plots include the experimental data (continuous line) and the simulated data (dotted line).

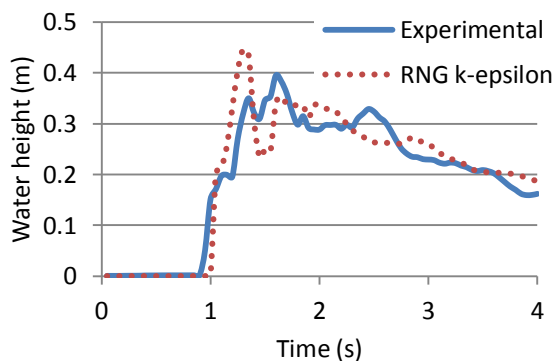


Figure 58 H1 water height values

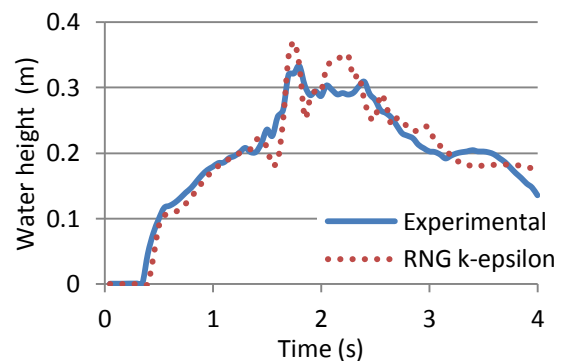


Figure 59 H2 water height values

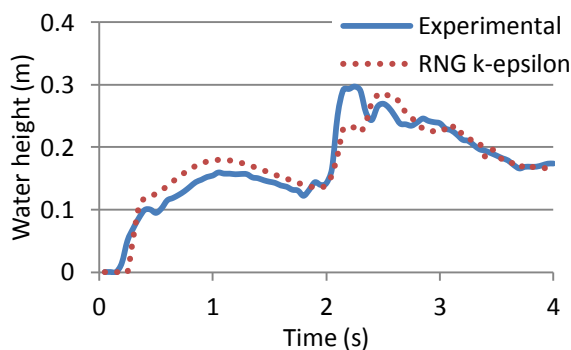


Figure 60 H3 water height values

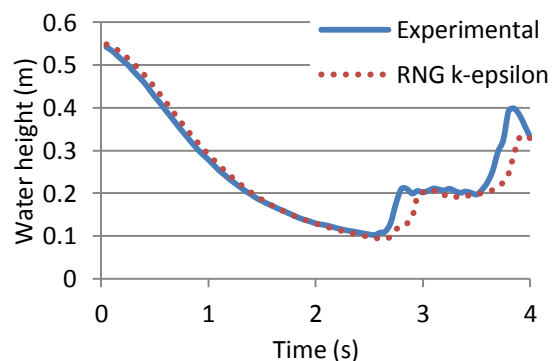


Figure 61 H4 water height values

Generally, the RNG k-epsilon model simulates in an acceptable way the general shape of the water level. Specifically, at each probe location there is some strength and some weaknesses.

At H1 location, the farthest one from the reservoir, there is a small delay on the wave arrival and some overestimating values when the first wave rebound at the wall and interacts with the slower water that goes through the obstacle. In the same way than in the Smagorinsky model, this imprecision might be produced by the difficulty of recording precisely the water heights when there is this interaction. This can also be the reason why between 1.5 and 3 s the simulated values are more different than the experimental, because the water is still stabilizing after the crash at the end of the wall.

At H2 the simulated data is almost the same as the experimental data at the first one and a half seconds. After that it is more inaccurate. The peak values (negative or positive) are overestimated and the simulated values do not have the same shape as the experimental data.

At H3 the general shape is achieved, even though when the water level rises there is a small delay and after that the water level is overestimated. At the last two seconds the major peak is underestimated but the second one is overestimated. These times correspond when the rebounded wave reaches this location. Dismissing this, the general shape of the simulated data matches the experimental data's shape.

At H4 the general shape is almost perfect simulated. There is a small delay from around the second and a half seconds until the end of the simulation time. The last peak is a little bit underestimated.

Looking at Figures 53 to 56, the conclusion is similar to the Smagorinsky's conclusion. The mean water height values are quite well simulated, with some exceptions, whereas at peak values there is more variance.

5.1.2.3. Pressure

The pressure results at the upstream face of the obstacle are shown at Figures 62 to 65. At the X axis there is the time in seconds, it takes values from zero to four seconds. At the Y axis there are the pressures in pascals of the corresponding points, including the experimental data (continuous line) and the simulated data (dotted line)

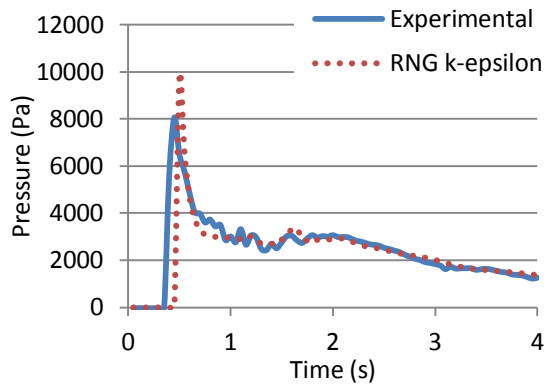


Figure 62 P1 pressure values

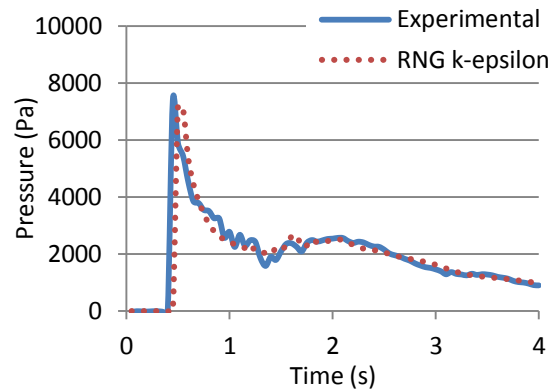


Figure 63 P2 pressure values

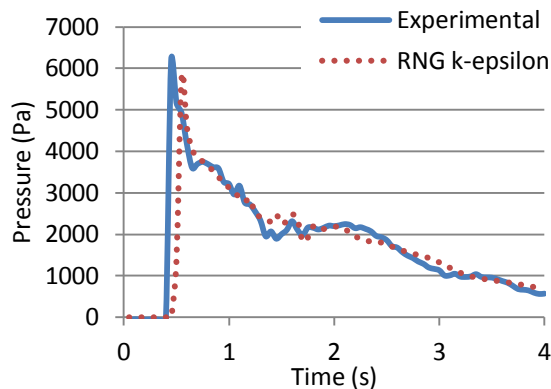


Figure 64 P3 pressure values

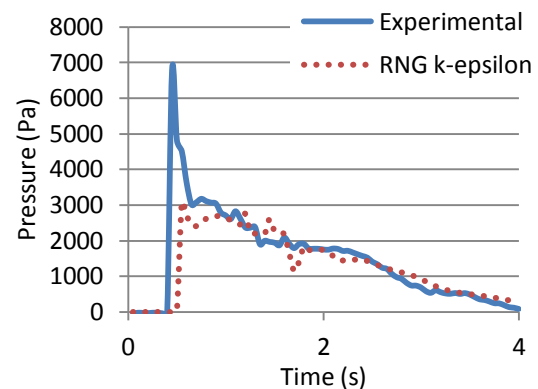


Figure 65 P4 pressure values

At P1 the peak is simulated with some delay and overestimated. The decrease of the peak is quite well simulated but after that the values are a few underestimated during less than a half second. The three remaining seconds are almost perfect simulated, except the first second which the simulated data have kind of the mean water height of the experimental values.

At P2 happens something similar to P1, but the peak is not overestimated otherwise is a little bit underestimated. The rest of the simulation has the same strength and weaknesses than P1 data.

At P3 there is more phase difference at the peak, but the rest of the values are much better simulated and sometimes overestimated. Like the other probe locations, the general shape is well simulated.

At P4 the simulation does not reach the peak value and there is a phase difference. Until the simulation time does not reach one second the values are underestimated. After one second the values are more or less well simulated, at least the shape is archived. There is some exaggeration of the negative and positive peaks. From 2 to 4 seconds the simulation follows more or less the experimental values.

Figures 66 to 69 relate the experimental data and the simulated data of the upstream pressure results in pascals. At the abscissa axis there are the experimental values and the simulated values are in the ordinate axis. Both data are in pascals.

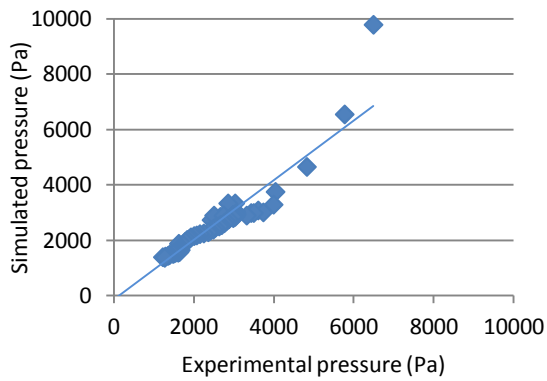


Figure 66 P1 pressure values

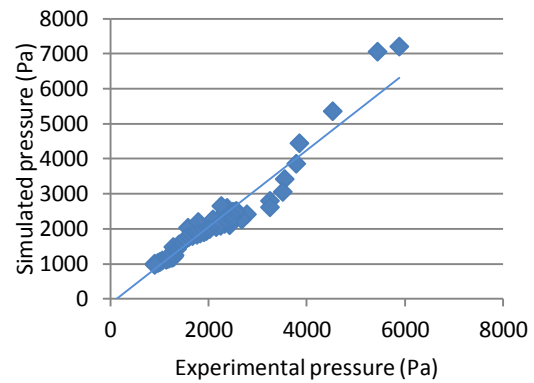


Figure 67 P2 pressure values

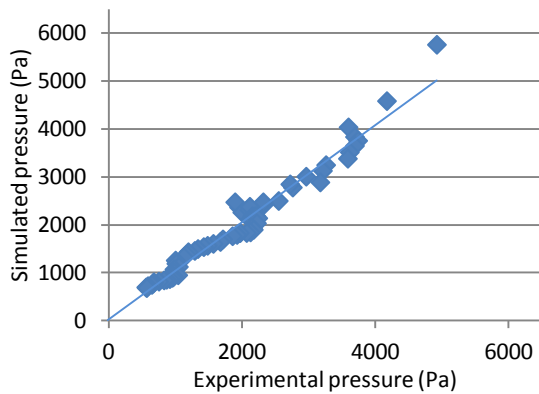


Figure 68 P3 pressure values

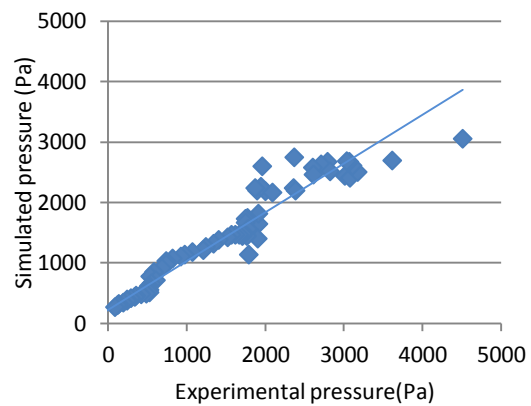


Figure 69 P4 pressure values

Approximately at all the probe locations the values smaller than 2000 Pa are well simulated (at P1 probe location until 3000 Pa), following the linear tendency line, which is almost 45 degrees (except at P4, which its linear tendency line denotes that the values are underestimated). At P1, P2 and P3 few values are outside the tendency line. At P1 and P2 most of those are overestimated, except two or four respectively. At P3 considering the 45 degree line the values are overestimated. At P4 the values outside its tendency line create a point cloud, being overestimated and underestimated.

To summarize the information of the upstream face of the obstacle, it can be said that generally speaking the RNG k-epsilon model simulates well enough the general shape and has a small delay at the initial peak.

In the same way that the other pressure values, the pressure results at the top face of the obstacle are shown at Figures 70 to 73. At the X axis there is the time in second, it takes values from zero to four seconds. At the Y axis there are the pressure in pascals of the corresponding points, including the experimental data (continuous line) and the simulated data (dotted line).

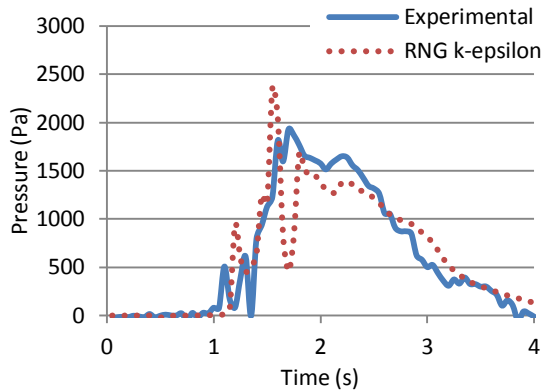


Figure 70 P5 pressure values

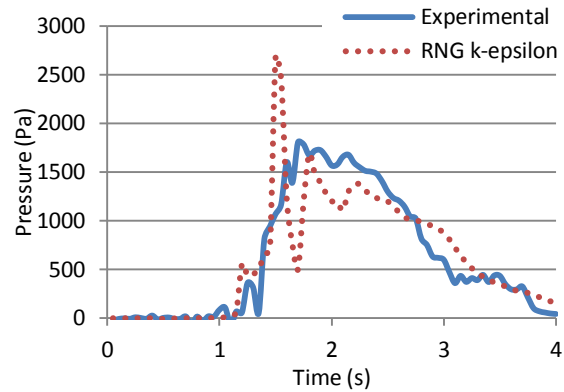


Figure 71 P6 pressure values

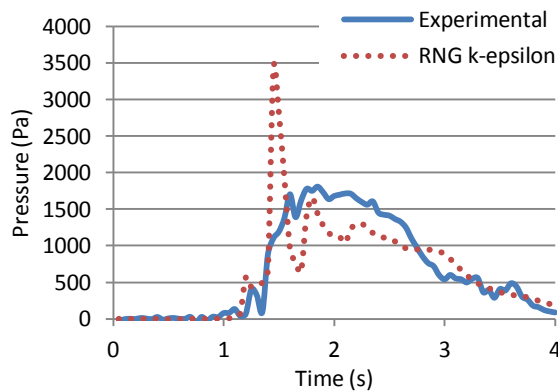


Figure 72 P7 pressure values

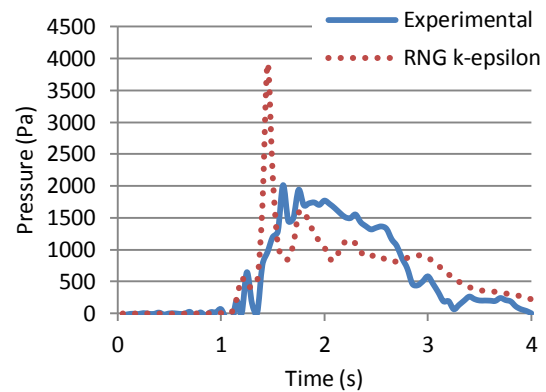


Figure 73 P8 pressure values

At first glance, it can be easily seen that the results are less precise than the others. Even though, the general shape is still obtained.

When the wave reaches P5 probe location there are some peaks (positive and negative) at the simulation data that do not exist at the experimental data. Except for these initial peaks, the rest of the simulation obtains a quite good approximation of the experimental values with some overestimating and underestimating values, but obtaining the general shape.

Something similar happens to P6 probe location, but in this case there are only overestimated or underestimated peaks.

P7 results are similar to P5, the first one and a half seconds is well simulated with a few overestimated values and then appears the peak that does not appear at the experimental data that corresponds approximately when the rebounded wave crashes to the obstacle. After that peak the next second approximately is underestimated and after that there are a few overestimated values. The last second is simulated quite well.

P8 experimental data have a similar behavior than P7; the main difference is that the last second is overestimated.

Figures 74 to 77 relate the experimental data and the simulated data. At the abscissa axis there are the experimental values and the simulated values are in the ordinate axis. Both data are in pascals.

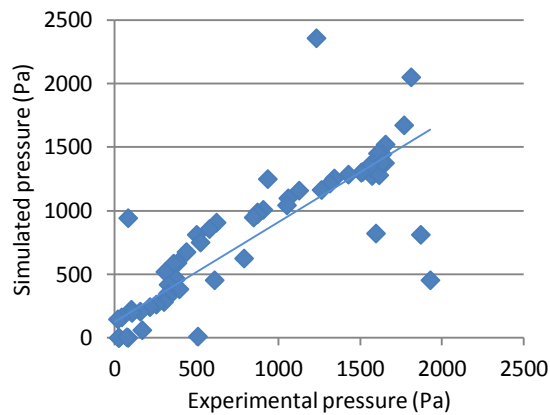


Figure 74 P5 pressure values

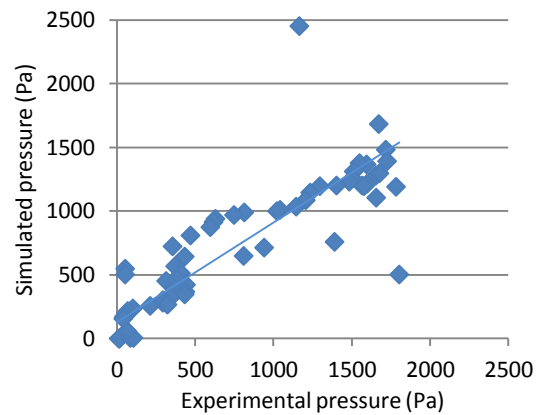


Figure 75 P6 pressure values

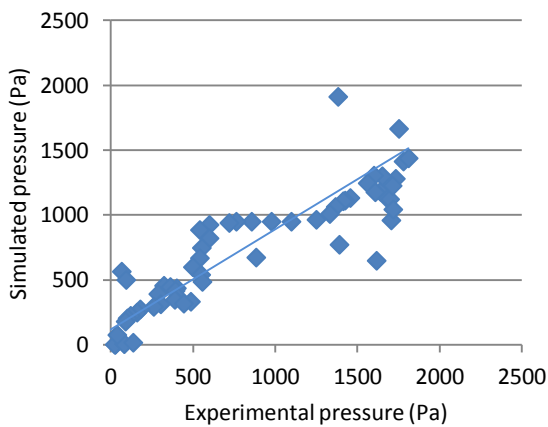


Figure 76 P7 pressure values

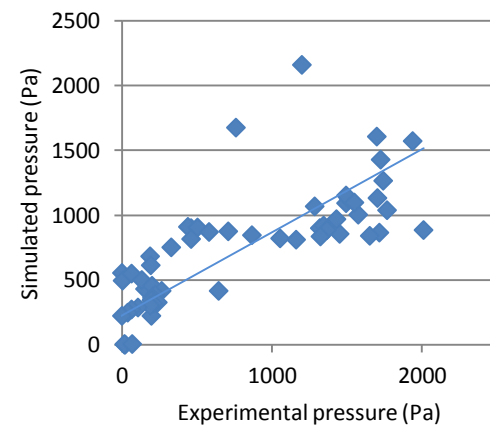


Figure 77 P8 pressure values

Approximately at all the probe locations the smallest values (less than 500 Pa), are best simulated that the others. Between 500 Pa and 1000 Pa the simulated values are overestimated, whereas in broad terms the biggest results are underestimated. The tendency line in all the probe locations indicates that there is a general tendency to underestimate the values.

To sum up the information of the top face of the obstacle, it can be said that generally speaking the RNG k-epsilon model simulates quite well the general shape and tends to overestimate the first peak produced by the first wave and underestimated the values the next second, and after it the values tend to be overestimated, as it can be seen at Figures 70 to 73.

5.1.3. Comparison

In order to compare the two models, the best and the worst simulation of each model will be used, considering water heights and pressures independently.

Both models simulate better the water level at the probe location inside the reservoir (H4). Figure 78 shows the three data: experimental, Smagorinsky and RNG k-epsilon.

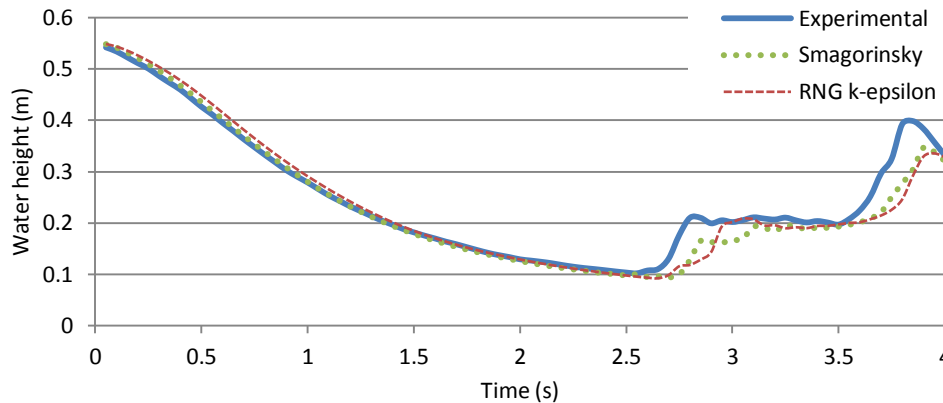


Figure 78 H4 water height values

The conclusion of Figure 78 is that the Smagorinsky model simulates much better the shape and has less phase difference. The disadvantage is that underestimates some values, at the end of the simulation time.

The worst water height result of the Smagorinsky simulation is the probe location H3. Figure 79 includes the following data: experimental, Smagorinsky and RNG k-epsilon.

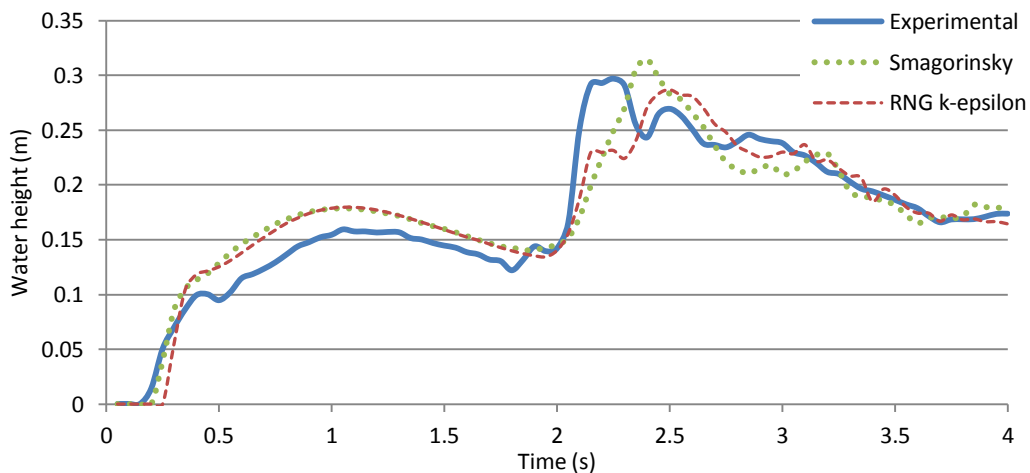


Figure 79 H3 water height values

The simulation in this case is quite bad for both simulations. At the beginning the Smagorinsky has less, almost zero, phase difference. The following seconds until 2 seconds the behaviour of both simulations is almost the same, overestimating the experimental data. At the last two seconds the RNG k-epsilon seems to tend to simulate better the shape, but it underestimates or overestimates the values following no rule. The Smagorinsky model also overestimates or underestimates the values.

The worst water height result of the RNG k-epsilon simulation is the probe location H1. Figure 80 includes the following data: experimental, Smagorinsky and RNG k-epsilon.

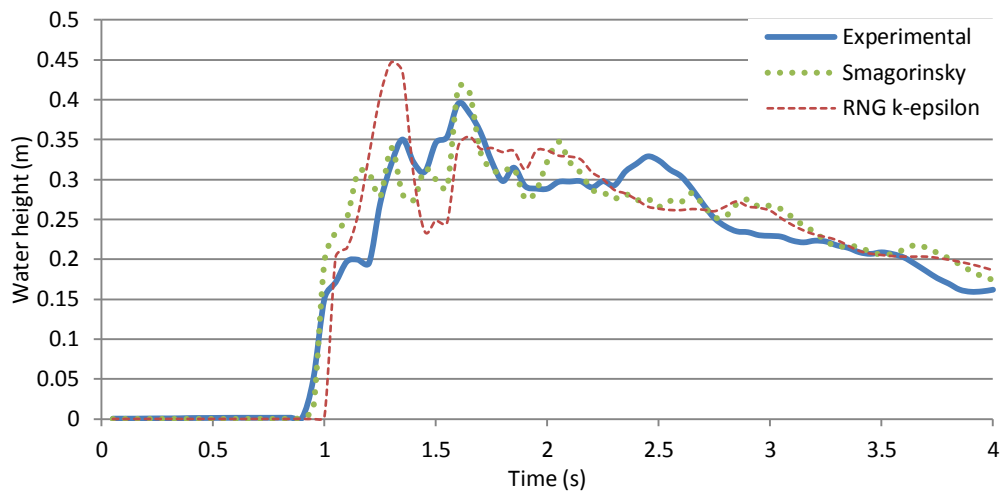


Figure 80 H1 water height values

The Smagorinsky simulation has less phase difference at the beginning than the RNG k-epsilon simulation. The rest of the simulation the Smagorinsky simulates much better the shape of the experimental data and has no huge overestimated positive peaks or underestimated negative peaks as Smagorinsky.

Summarizing, Smagorinsky seems to simulate better the water height shape than RNG k-epsilon. Even though sometimes overestimate or underestimate the values.

Both models simulate better the pressure at the probe location P3, at the upstream face of the obstacle. Figure 81 shows the three data: experimental, Smagorinsky and RNG k-epsilon.

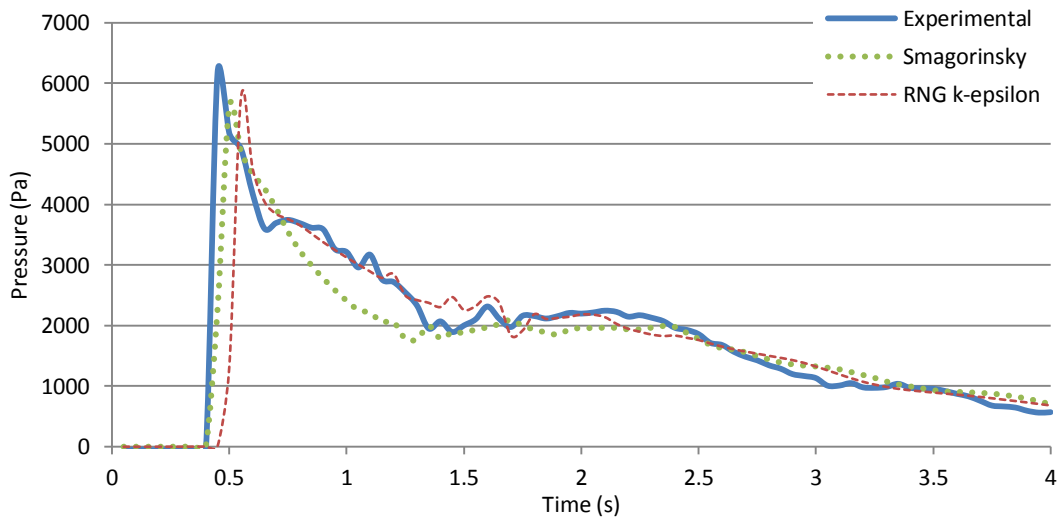


Figure 81 P3 pressure values

Smagorinsky model simulates better the peak, with less phase difference, whereas RNG k-epsilon simulates better the other values.

Both models also coincide with the worst simulated pressure probe location. It is P8.

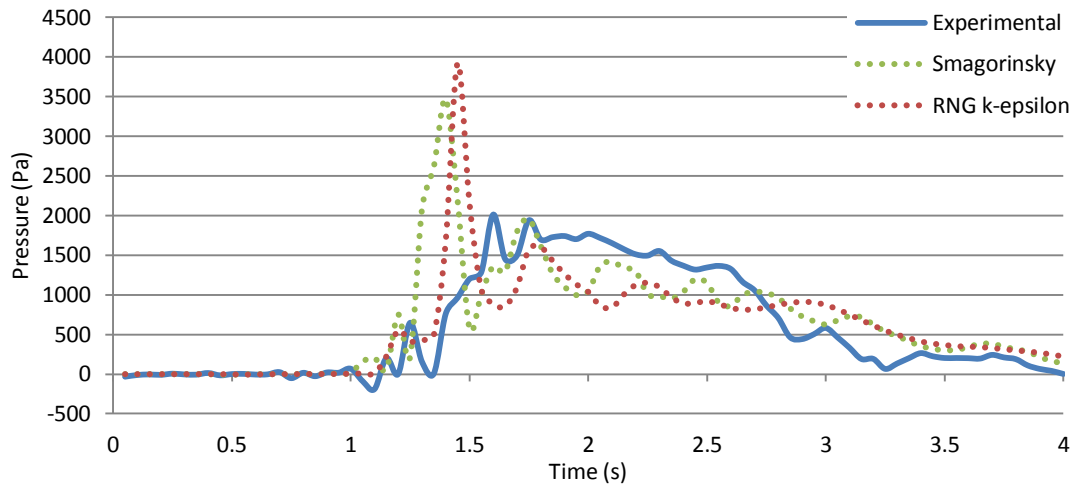


Figure 82 P8 pressure values

Both models simulate a peak around 1.5 s which the experimental data does not have. Considering the rest of the data, both are quite similar; but Smagorinsky model captures better the general shape and has values closer to the experimental data.

Summarizing, Smagorinsky model seems to simulate better the small pressure values (less than 2000 Pa) whereas RNG k-epsilon seems to simulate better the biggest. Moreover, Smagorinsky model has less phase difference.

5.2. Dam-break test 2

The chosen model according to the test 1 results is Smagorinsky turbulence model. To check if the Smagorinsky model simulates well other domains the test 2 is carried out. To simulate this experiment, the 1 cm mesh is used.

5.2.1. Water height

Figures 83 to 87 simulate the water level at different locations. Abscissa axis has the water level in meters and the ordinate axis represents the simulation time in seconds.

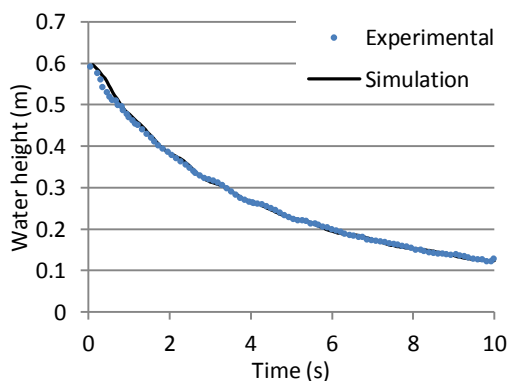


Figure 83 5A water height values

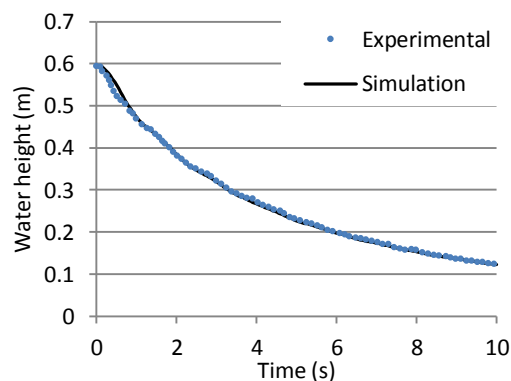


Figure 84 C water height values

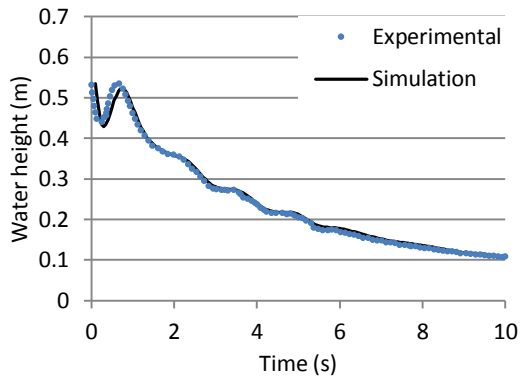


Figure 85 0 water height values

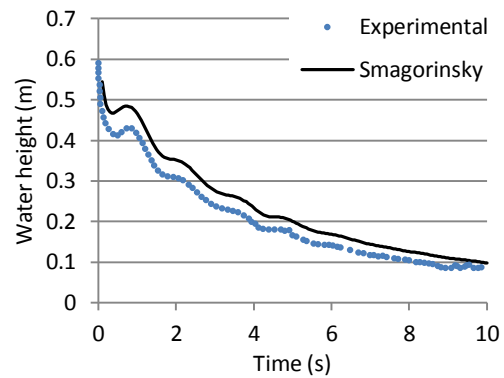


Figure 86 4 water height values

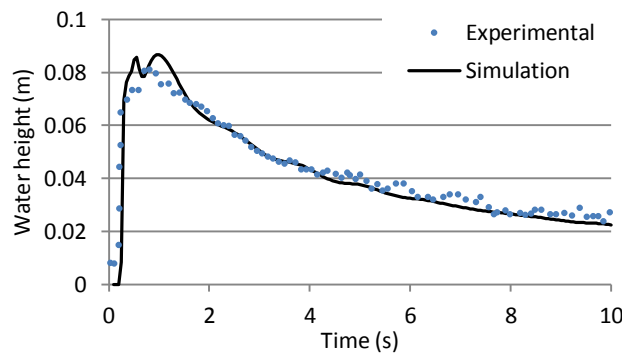


Figure 87 8A water height values

The probe locations 5A and C are inside the reservoir. The probe points 0 and 4 are located on the gate, the first in the middle and the second one in the left hand side of it. Finally, the probe location 8A is outside the reservoir.

The Smagorinsky model simulates almost perfect the water surface inside the reservoir when there is no high curved surface (Figures 83 and 84).

At location 0, in the middle of the gate gap, the simulation is almost perfect as well, except for the initial second when there is a small phase difference; the Smagorinsky model is slower than the experimental data.

At location 4, at the left hand side of the gate gap, the Smagorinsky model overestimates the water level. This may be produced by the effect of the wall when the water tries to go through a small gap; it is similar to the effect produced upstream of a bridge pile. This effect is not taken into account in the Smagorinsky model and it is not part of this work. Even though, the shape of the water surface is well simulated.

At location 8A, outside the reservoir, the simulation is worse than inside, but it is good enough. There are only a few discrepancies around the first second, where the Smagorinsky model simulates a double peak, whereas the experimental value only has one. Even though, the order of magnitude of the simulated data is similar to the experimental one.

Therefore, the Smagorinsky model simulates the water surface of the experiment 2 with good results.

5.2.2. Bottom pressure

Figures 88 to 91 show the probe locations 0, 4, 5A and 8A, but in this case the magnitude compared is the total pressure. The pressure value is converted to water height by dividing its value for the water density and the gravity acceleration. Abscissa axis has the water level in meters, obtained from the bottom pressure, and the ordinate axis represents the simulation time in seconds.

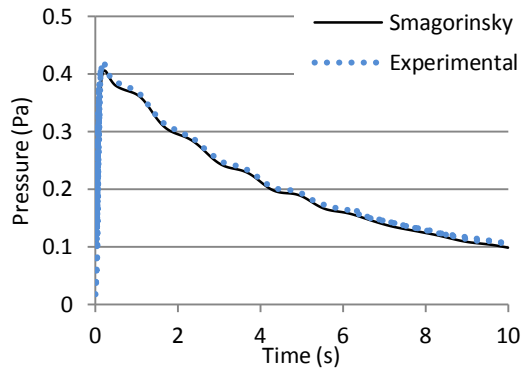


Figure 88 0 pressure values

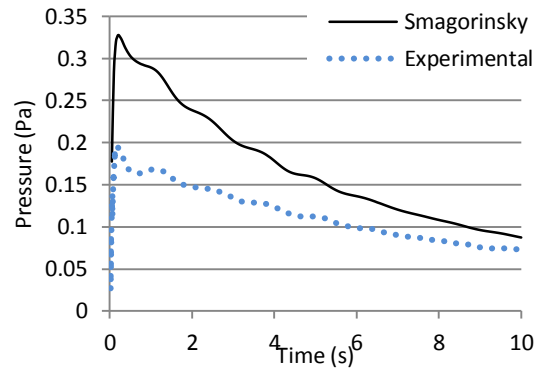


Figure 89 4 pressure values

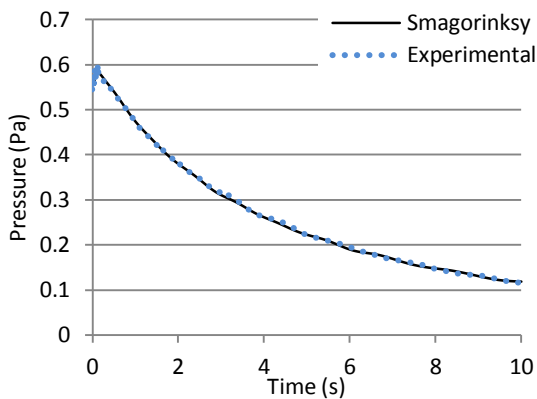


Figure 90 5A pressure values

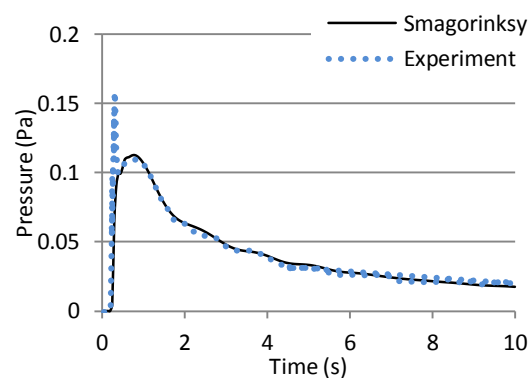


Figure 91 8A pressure values

The Smagorinsky model simulates almost perfectly the experimental values recorded in the middle of the gate and inside of the reservoir, Figures 88 and 90, respectively.

At the probe location 4 the results are overestimated: the same that has happened with the water height. Even though, the shape is perfectly simulated without phase difference.

Outside of the reservoir at the location 8A the simulation data is almost identically at the experimental, except for a peak at the beginning of the experimental data, when the water reaches this point. It is a local peak, maybe is an error produced in the pressure gauge of the experiment.

Summarizing the Smagorinsky model simulates quite well the pressure values, except near to the reservoir wall which produces a narrow gap.

5.2.3. Water height vs. pressure head

Figures 92 to 95 compare the water height simulated and the pressure head⁴ obtained from the total simulated pressure. Abscissa axis has the water level in meters and the ordinate axis represents the simulation time in seconds.

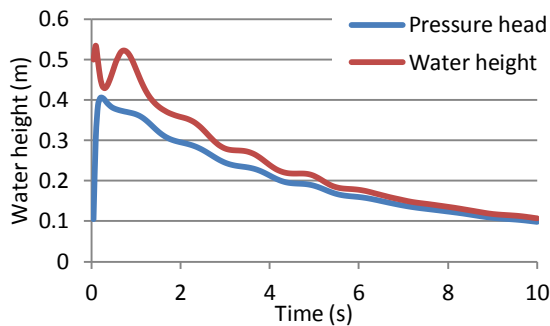


Figure 92 0 pressure head and water height

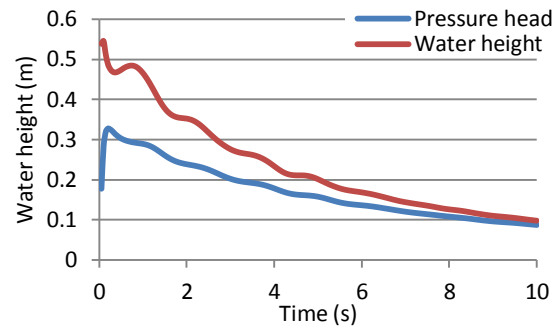


Figure 93 4 pressure head and water height

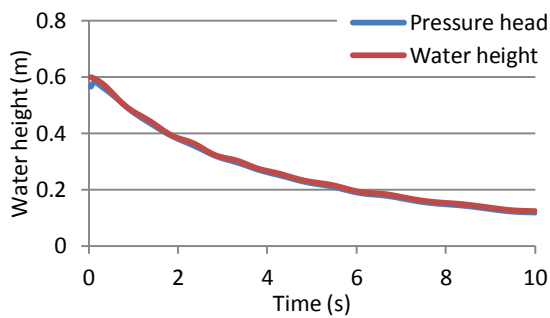


Figure 94 5A pressure head and water height

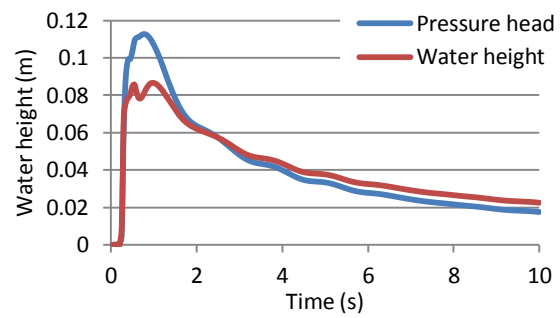


Figure 95 8A pressure head and water height

The pressure head and the water height are completely different at the gate. They start to become similar when the water level is smaller, at the end of the simulation. This is because the total pressure is the sum of the static pressure and the dynamic pressure (Grc.nasa.gov, 2017). When the water height is smaller means that the reservoir is almost empty and therefore the water velocity is smaller.

Inside the reservoir at the probe location 5A, far away of the highly curve water surface, the pressure head and the water height have almost the same values. There is only a small difference at the beginning.

Outside the reservoir at probe location 8A there is difference between both data. When the wave reaches that point the pressure head has a higher peak than the real water height. After that peak, or double peak in the water height case, the water height becomes bigger than the pressure head, in the same way that happens in the gate.

Summarizing, it is proved that the pressure head is not equal to the water height when there is a highly curved water surface.

⁴Pressure head is obtained dividing the total pressure (see Table 3) by the water density and the gravity acceleration (9.81 m/s^2).

5.2.4. Velocity

There are different velocity data, five in the longitudinal direction and one in the transversal direction. Figures 96 to 99 show the mean velocity values of the water; whereas Figure 100 and 101 shown the longitudinal velocity at the location 0 at two different heights. Figure 100 represents the longitudinal velocity at 0.05 m from the bottom and the Figure 101 represents the longitudinal velocity at 0.40 m from the bottom.

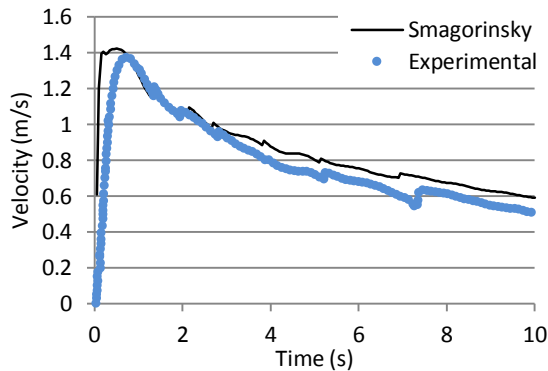


Figure 96 Longitudinal mean velocity at 0

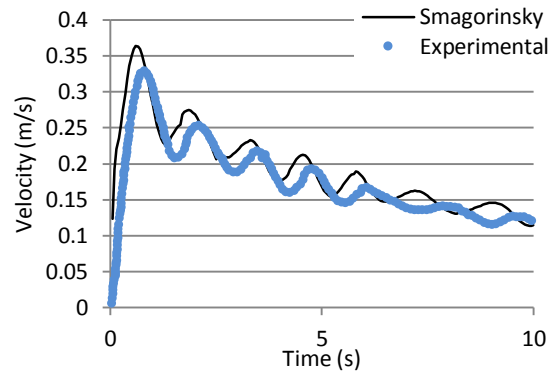


Figure 97 Longitudinal mean velocity at -3A

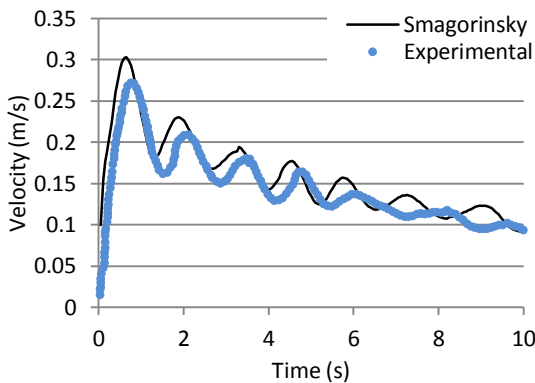


Figure 98 Longitudinal mean velocity at -3D

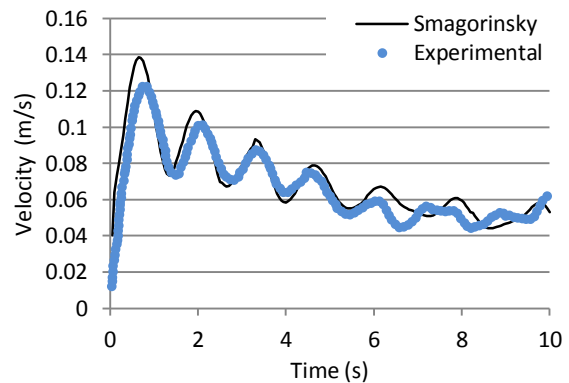


Figure 99 Transversal mean velocity at -3D

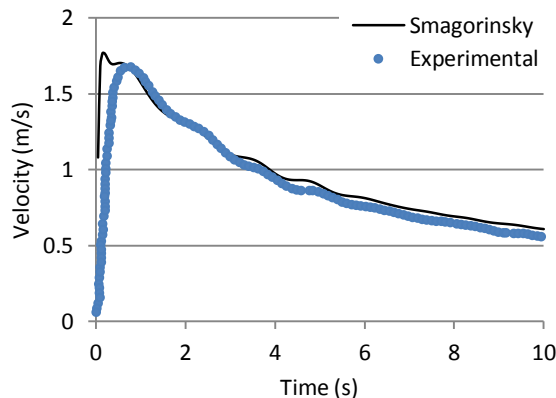


Figure 100 Longitudinal velocity at 0 ($z=0.05$ m)

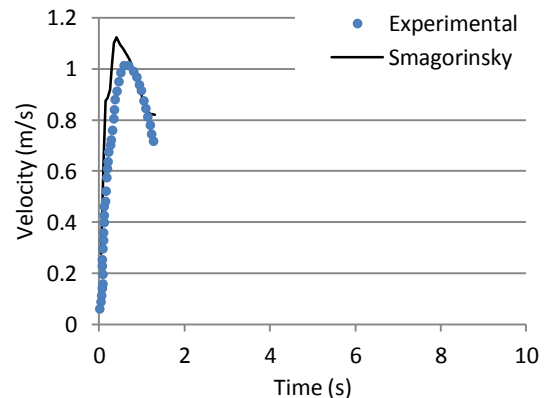


Figure 101 Longitudinal velocity at 0 ($z=0.4$ m)

The simulated mean longitudinal velocity at 0 is higher at the beginning and it is a little bit ahead of time. After the peak during approximately two seconds the velocity is well simulated. After that the Smagorinsky model overestimates the longitudinal velocity.

The general shape is almost simulated, except around the seventh second of the simulation that the experimental data has a negative peak that the simulation does not have.

The simulated mean longitudinal velocity at the location -3A is a little bit overestimated and head of time all along the simulation.

The simulated mean longitudinal velocity at -3D is less ahead of time than the previous one, but it is still a little bit overestimated.

At the same location, -3D, but in the transversal direction; the simulation is less ahead of time but is still overestimated. Moreover, at the last 3 seconds the shape is not very well simulated.

Generally speaking, all the mean velocities are a little bit ahead of time and overestimated.

Comparing the velocities in a certain distance from the bottom the results are better. At 0.05 m from the bottom, the longitudinal velocity at 0 is well simulated during the last 9 seconds. At the first one there is a peak earlier than in the experimental data.

At Figure 101 there is data of the longitudinal velocity at 0.4 m from the bottom at the location O. There is only data until 1.3 s. After that time there is no water at that height, this is the reason why there is no velocity data. The Smagorinsky model simulates the general shape, but it overestimates the peak and goes a little bit ahead of time.

6. Conclusions and Recommendations

The OpenFOAM allows the 3D simulation, which might be better than the 2D or 1D, the inconvenient is that it has to implement turbulence models to simulate properly the flow. The 1D or the 2D programs do not need it because they assume hydrostatic pressure. At this work Smagorinsky and RNG k-epsilon turbulence model are used with the *interFoam* solver. Which according to the results obtained is able to simulate the free surface in these 3D experiments. At the first test the free surface is less precise than in the second test, where it is almost identical, although the general shape is simulated; there are only few overestimated or underestimated values.

Checking the non-hydrostatic effects produced by the highly curved surface, it is proved that where there is this abrupt change of the water surface the pressure head does not represent the water height. Whereas, when there is no highly curved surface the pressure head is equal or almost equal to the water height, meaning that there is hydrostatic pressure.

The chosen mesh in both turbulence models is 1 cm; the results show that it has not reach convergence, meaning that a more refined mesh will be needed to improve the results. Because the R^2 values, the Equation 16 results and the high values of the GCI seem to indicate that the simulation results have an oscillatory convergence. No finest mesh is computed in this work because of computational and storage limitations. To simulate the experiments of this work with 1 cm mesh in a computer with 32 GB of RAM memory and an intelCore i7-6700k processor takes 2 days.

Comparing both turbulence models, Smagorinsky seams to simulate better the experimental data. In one hand it simulates better the water height shape than RNG k-epsilon; although it sometimes overestimates or underestimates the values. On the other hand, it simulates better the small pressure values (less than 2000 Pa) and it has less phase difference.

The Smagorinsky water height and pressure simulations of the second experiment prove that this model is able to simulate other experiments giving good results of the water surface and bottom pressure. In fact, the results in this experiment are better than the results of the first experiment. That means that the results are dependent of the situation that they are simulating and therefore, some turbulence models simulate better one experiment than another one.

Analyzing the second experiment is detected that the Smagorinsky model is not able to simulate well enough the water height and the bottom pressure when there is a reduction of the section width.

The second experiment proves that the Smagorinsky model is able to simulate well enough the velocity field, but the simulated values are a little bit overestimated and ahead of time.

References

- Arohatgi.info (2017). *WebPlotDigitizer – Copyright 2010-2017 Ankit Rohatgi*. [online] Available at: <http://arohatgi.info/WebPlotDigitizer/app/> [Accessed 7 Apr. 2017]
- Bakker, A. (2002). *Lecture 10- Turbulence Models. Applied Computational Fluid Dynamics*
- Bu.edu. (2017). *Using ParaView to Visualize Scientific Data (online tutorial) : TechWeb : Boston University*. [online] Available at: <http://www.bu.edu/tech/support/research/training-consulting/online-tutorials/paraview/> [Accessed 9 Jun. 2017]
- Celik, I.B., Ghia, U., Roache, P.J., Freitas, C.J, Coleman, H., Raad, P.E. (2008). *Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications*, Journal of Fluids Engineering.
- Davidson, L. (2016). *Fluid mechanics, turbulent flow and turbulence modeling*, Chalmers University.
- Efros, V. (2006). *Large Eddy Simulation of Channel Flow using Wall Functions*, Chalmers University.
- Fraccarollo, L., and Toro, E. F. (1995). *Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems*. Journal of Hydraulic Research, Taylor & Francis Group , 33(6), 843–864.
- Grc.nasa.gov. (2017). *Dynamic Pressure*. [online] Available at: <https://www.grc.nasa.gov/www/k-12/airplane/dynpress.html> [Accessed at 10 Jun. 2017].
- Guerrero, J. (2015). *Introductory OpenFOAM Course*, University of Genoa.
- Kleefsman, K.M.T., Fekken, G., Veldman, A., Iwanowski, B., Buchner, B. (2005). *A Volume-of-Fluid based simulation method for wave impact problems*. Journal of computational physics, 206(1), 363-393.
- Niño, Y. (2002). *Método de los volúmenes finitos*, Universidad de Chile.
- OpenFOAM, Programmer's Guide (2015)
- OpenFOAM, User's Guide (2015)
- Openfoam.com (2017). *OpenFOAM Documentation*. [online] Available at: <http://www.openfoam.com/documentation/> [Accessed 30 Jan. 2017]

Reddy, J.N. (2008). *An introduction to Continuum Mechanics*, Cambridge University Press.

Schwer, L. E. (2008). *Is your mesh refined enough? Estimating Discretization Error using GCI*, Schwer Engineering & Consulting Services.

Yakhot,V. , Orszag, S. A., Thangam, S. , Gatski, T. B.and Speziale,C. G. (1992). *Development of turbulence models for shear flows by a double expansion technique*, American Institute of Physics, 4, 1510.