



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA  
INGENIERÍA CIVIL – HIDRÁULICA

MODELACIÓN NUMÉRICA DE FLUJOS MULTIFÁSICOS: INTERACCIÓN DEL OLEAJE  
CON DESCARGAS SUBMARINAS DE AGUA DULCE

TESIS  
QUE PARA OPTAR POR EL GRADO DE:  
MAESTRA EN INGENIERÍA

PRESENTA:  
PAOLA ELIZABETH RODRÍGUEZ OCAMPO

TUTORES PRINCIPALES  
RODOLFO, SILVA, CASARÍN, INSTITUTO DE INGENIERÍA  
JUAN CARLOS, ALCÉRRECA, HUERTA, INSTITUTO DE INGENIERÍA

CIUDAD UNIVERSITARIA, MÉXICO, D. F. NOVIEMBRE 2016



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**JURADO ASIGNADO:**

Presidente: DR. EDGAR GERARDO MENDOZA BALDWIN

Secretario: DR. JUAN CARLOS ALCÉRRECA HUERTA

Vocal: DR. RODOLFO SILVA CASARÍN

1 er. Suplente: DRA. LILIA REYES CHÁVEZ

2 d o. Suplente: DR. ISMAEL DE JESÚS MARIÑO TAPIA

Lugar donde se realizó la tesis: CIUDAD DE MÉXICO

**TUTOR DE TESIS:**

RODOLFO SILVA CASARÍN

-----  
**FIRMA**

*(Segunda hoja)*

## Agradecimientos

*A mis papás Delia y Víctor, porque con su ejemplo y apoyo me han enseñado a valorar las oportunidades y a superarme en todos los aspectos de mi vida.*

*A mis hermanos Juan Carlos e Isaac, con cariño.*

*A Gabriel y a Juan Carlos A., que sin su incalculable apoyo, esta tesis no sería lo que es hoy.*

*A mis amigos más cercanos, Nikte, Margarita, Julieta, Mónica, Miriam, Enrique, Juan V. y Emiliano porque siempre, y en especial en esta etapa, han estado ahí para ayudar a levantarme de los tropiezos y desaires. A mis entrañables amigos de licenciatura y maestría, Julio, Juan L., Diego, Itzel, Armando y César.*

*A todos aquellos que con una palabra de aliento, una sugerencia o simplemente con su compañía, hicieron más llevaderos los momentos difíciles. Sin dejar de mencionar al resto de mi familia materna, primas, primos, tías y tíos.*

*A la UNAM por su generosa educación y a CONACYT por solventar los gastos de mis estudios. Al programa Exceed por el apoyo brindado durante la estancia en Alemania. Al personal de Laboratorio de Costas y Puertos. A mis amigos y compañeros del grupo de Costas y Puertos.*

*A todos los maestros que me inspiraron y me apoyaron en esta etapa, en especial al Dr. Rodolfo y al Dr. Edgar.*

*Al Big Bang todo poderoso, cuya explosión hace 13 800 millones de años, hizo posible que todos estemos aquí.*

# Index

Index.....	4
1. ABSTRACT.....	6
<i>Resumen</i> .....	6
2. INTRODUCTION.....	7
2.1. MOTIVATION.....	7
2.2. OBJECTIVES .....	8
2.3. DESCRIPTION OF THE WORK.....	8
3. THEORETICAL FRAMEWORK .....	10
1 SUBMARINE GROUND WATER DISCHARGE (SGD) .....	10
3.1.1. Estimation and measurement of submarine groundwater discharge.....	11
3.1.2. Coastal Karst aquifers .....	13
3.1.3. The Karstic aquifer of the Yucatan Peninsula .....	14
3.2. NUMERICAL MODELING IN COMPUTATIONAL FLUID DYNAMICS .....	15
3.2.1. Background and generalities.....	15
3.2.2. Conservation principles.....	17
3.2.3. CFD software.....	19
3.2.4. Multiphase flow modeling .....	20
3.2.5. Turbulence modeling .....	22
3.3. NUMERICAL MODELLING APPLIED FOR SGD DESCRIPTION.....	23
3.4. SUMMARY AND IMPLICATIONS FOR THE STUDY.....	24
4. DEVELOPMENT OF THE CFD MODEL.....	27
4.1. GOVERNING EQUATIONS.....	27
4.2. EXTENSION OF THE SOFTWARE .....	29
4.2.1. Code modification.....	33
5. CALIBRATION AND VALIDATION OF THE MODEL.....	37
5.1. Design and construction of the laboratory model.....	37
5.2. Numerical setup.....	41
5.2.1. Mesh and boundary conditions .....	42
5.2.2. Initial field, physical and fluid properties.....	43
5.2.3. Field boundary and initial conditions.....	45

5.2.4.	Control and data extraction .....	46
5.3.	Calibration and validation of the numerical model .....	46
5.3.1.	Qualitative analysis .....	46
5.3.2.	Quantitative analysis: image processing and error matrixes.....	51
5.3.3.	Quantitative analysis: numerical probes .....	57
5.4.	Qualitative and quantitative description of the cases 26 to 30.....	60
6.	APPLICATION TO A STUDY CASE .....	62
6.1.	Description of the study area.....	62
6.2.	Results .....	65
7.	CONCLUSIONS.....	70
7.1.	Restrictions and limitations of the model.....	70
7.2.	Impact of the model.....	71
8.	Annexes.....	75
8.1.	Annex A. Case setup example with <code>interMixingWaveFoam</code> .....	75
8.1.1.	Preliminary model setup .....	75
8.1.2.	Mesh generation .....	76
8.1.3.	Physical and fluid properties.....	80
8.1.4.	Field boundary and initial conditions.....	82
8.1.5.	Initial field .....	87
8.1.6.	Control .....	88
8.1.7.	Case running and post-processing .....	89
9.	References .....	92

## 1. ABSTRACT

Regions such as the Yucatan Peninsula, in Mexico, are constituted of karst soils which encourage the formation of underground conduits through which fresh water is transported from the aquifer into the sea. As a result of this process, submarine springs are originated and commonly called “ojos de agua”, which carry freshwater, diluted nutrients and pollutants from the land to the sea. Submarine groundwater discharge (SGD) is relevant in coastal engineering because it influences the formation of marine ecosystems such as coral reefs. Coral reefs are important in the dissipation of waves and coastal protection. No suitable tool is yet available to simulate in detail SGD and therefore, the development of a Computational Fluid Dynamic (CFD) model is required. The CFD model must fulfil the most relevant features involved: i) generation, propagation and absorption of waves; ii) three phases with different density (air, fresh and salt water or brackish water); iii) mixing between the two liquid phases, and iv) turbulence with an appropriate model. Thus, the present thesis describes the development, calibration and validation of a numerical model, named “interMixingWaveFoam”, for the simulation of these processes as well as a preliminary implementation for a case study.

### **Resumen**

*En lugares como la península de Yucatán, México, las condiciones kársticas del suelo favorecen la formación de conductos subterráneos a través de los cuales el agua dulce es transportada desde el acuífero hacia el mar. Como consecuencia de dicho proceso se originan salidas de agua dulce en el mar a modo de manantiales submarinos los cuales son conocidos comúnmente como ojos de agua, los cuales transportan agua dulce, nutrientes y contaminantes diluidos desde la tierra hasta el mar. Estos procesos son importantes en la ingeniería de costas debido a que condicionan la presencia de ecosistemas como los arrecifes de coral, los cuales a su vez son disipadores de oleaje, importantes en la protección de la costa. Debido a que no se cuenta con una herramienta adecuada para simular a detalle las descargas submarinas de agua dulce y su comportamiento hidrodinámico, se planteó la necesidad de generar un modelo de mecánica de fluidos computacional que cubriera los aspectos más importantes involucrados: i) generación, propagación y absorción de oleaje; ii) tres fases de fluido con distinta densidad (aire, agua salada y agua dulce o salobre); iii) mezclado entre las dos fases líquidas, y iv) turbulencia con un modelo apropiado. De esta forma, en la presente tesis se muestra el desarrollo, calibración y validación del modelo numérico “interMixingWaveFoam” para simulación de los procesos ya mencionados, así como su aplicación preliminar a un caso de estudio.*

## 2. INTRODUCTION

### 2.1. MOTIVATION

The submarine groundwater discharge (SGD) (Figure 1) is an important coastal process in which hydrodynamic interactions between the inland piezometric level, sea level, incident waves, different water densities and changes in fluid flow directions take place. This phenomenon generates turbulence and modifies the hydrodynamics of both incident waves and freshwater discharge. The SGD can be found in areas where geological conditions allow the formation of conduits in which the freshwater from the rain flows into the sea. Point source SGD (or springs) can be either diffusive or punctual. SGD and saltwater intrusion are two complementary processes, however, several studies have been conducted in the field of saltwater intrusion through the porous medium, and thus mainly focused on the water movement in the direction of the mainland. There are a few detailed studies of the influence of SGD in the marine environment and available results from large scale studies provide a description of the hydrodynamic behavior but based mainly on field measurements. Therefore, it is noticed the absence of an appropriate model to reproduce this phenomenon by considering the different agents involved (e.g. sea levels, density gradients and waves).

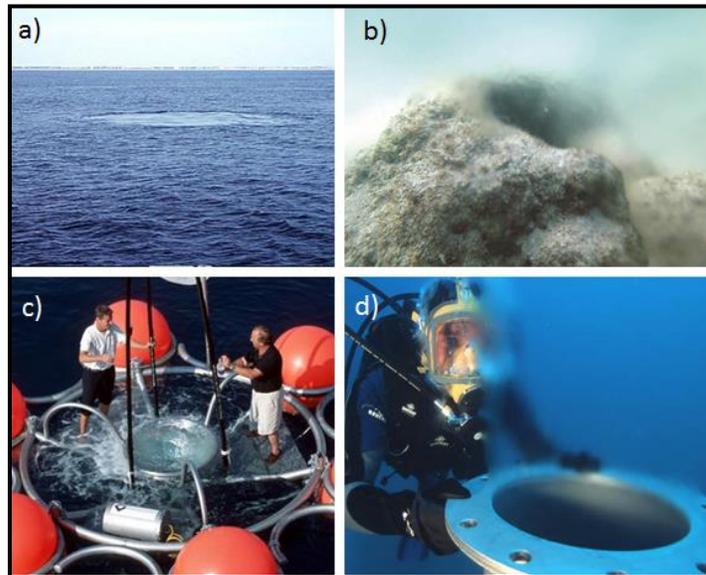


Figure 1. a) Crescent Beach Submarine Spring main view. The spring is situated 59 feet below the surface of the Atlantic Ocean, and the spring boil can be seen at the sea surface on a calm day (“St. Johns River Water Management District,” 2016). b) Submarine point source freshwater spring (Yousuf, 2014) c) and d) Nymphaea Company offers its know-how and patented technologies to locate, analyze and tap the submarine freshwater springs (“Solutions for Water,” 2012).

In the present thesis, focus has been set in the development of a tool able to reproduce point source SGD numerically in order to: i) achieve a better understanding of the SGD phenomenon, and ii) predict possible scenarios with certain conditions of interest. The numerical model could also be useful for the study of

other cases of interest such as: the submarine sewage discharge, hot water discharge in rivers or analysis for assessing marine energy by considering the density gradients between salt and freshwater.

## 2.2. OBJECTIVES

The development of a numerical model for the description of fluid motion, considering density gradients and their mixing, represents a key objective for the better understanding of SGD. From this, specific objectives are found:

- Development of a numerical model based on a CFD toolbox for the simulation of a wave field and a three-phase fluid system (where two of the phases are miscible): air, saltwater and fresh/brackish water.
- Validation and calibration of the model with experimental data from benchmarking tests.
- Application of the model in a real study case of the Yucatan Peninsula, Mexico. The study case involves a particular type of submarine groundwater discharge known as “ojos” which are submarine springs of brackish water.

## 2.3. DESCRIPTION OF THE WORK

This thesis consists in the development, calibration, validation and application of a computational fluid dynamics (CFD) solver, which will be capable to describe numerically the point source SGD. The Reynolds Averaged Navier-Stokes Equations (RANS Equations) will be used to describe the turbulent flow caused by: i) different water densities and ii) different fluid flow directions (groundwater input and incident waves). These equations are implemented within the CFD package OpenFOAM, which was extended to obtain the complete description of the aforementioned processes. As a result, a new solver has been created and, after its validation with laboratory tests, used to solve a problem of a field study case.

The thesis firstly shows a brief introduction to the theoretical framework related to SGD, turbulence and numerical modelling. Afterwards, a detailed description of the process that was followed to create the numerical model is given. Then, validation and calibration of the numerical model is provided and followed by an application of the model to a study case. Finally, results and conclusions are drawn.

# **THEORETICAL FRAMEWORK**

### 3. THEORETICAL FRAMEWORK

The most important concepts that will be used in this work are described in this chapter. An introduction to the submarine groundwater discharge, its importance, occurrence, common locations, measurement and the current knowledge are firstly presented. Then, a summary of numerical modeling in hydraulics is made considering the most important equations and methods suitable for the simulation of fluid motion/interaction. Multiphase numerical modeling methods and a description of the turbulence phenomenon are then described. Finally, a brief overview of current studies related to the modelling of SGD will be made together with a summary for implications to this work.

#### 1 SUBMARINE GROUND WATER DISCHARGE (SGD)

Submarine groundwater discharge is defined by Paulsen *et al.* (2004) as “...any water that discharges across the sediment-water interface between permeable sediments and overlying surface water. SGD is a mixture of water originating from terrestrial aquifers as fresh ground water and overlying surface water if fluid density contrast, tidal loading, and wave action are significant”.

Within the global water cycle, there are two important sources of continental freshwater to the oceans: i) river discharge and ii) submarine groundwater discharge (SGD). The first one is a major pathway to discharge water from great distances inland and it is represented by rivers. These water contributions to the ocean are easily quantifiable by various techniques such as: gauging stations, semi-analytical methods, discharge measuring structure. On the other hand, SGD occurs as springs and seeps on continental margins, at or below the water surface. The length where SGD takes place can be large, so that SGD contributions could be very significant per unit length, whether or not rivers exist in the region (Taniguchi *et al.*, 2002).

SGD hasn't drawn much attention from the oceanographic community until relatively recently. SGD studies have been restricted mainly to the case of water movement from the sea to the land (e.g., saltwater intrusion) even though interactions between surface water and groundwater have been analyzed in the fields of hydrology and limnology. Additionally, SGD has been focused on the water output from a basin-scale hydrological cycle (Taniguchi *et al.*, 2002), which means that SGD has been generally studied from a large-scale point of view.

Saltwater intrusion (SWI) and the submarine groundwater discharge (SGD) are two entirely complementary processes in the interaction between seawater and the groundwater in coastal aquifers, even though they may seem at first glance to be exactly opposite. Any variation of the boundary conditions in this balance may alter both SWI and SGD; e.g., if a change in the sea-level takes place, the SGD rate and the location of the subsurface saltwater-freshwater boundary will be modified (Taniguchi *et al.*, 2002). Within this context, (Valle-Levinson *et al.*, 2011) studied the effects of tidal forcing on an intense buoyant spring discharge located at the transition between a subterranean estuary and the coastal ocean. The variability of the velocity field and the spatially averaged salinity distributions in the saltwater column was attributed to tidal variations of hydrostatic pressure acting on the spring outflow (Figure 2) and the salinity distributions showed to have lowest values at low tides and highest values at

high tides. Thus, the extent of SWI and SGD is an issue of balance between hydraulic and density gradients in groundwater and seawater.

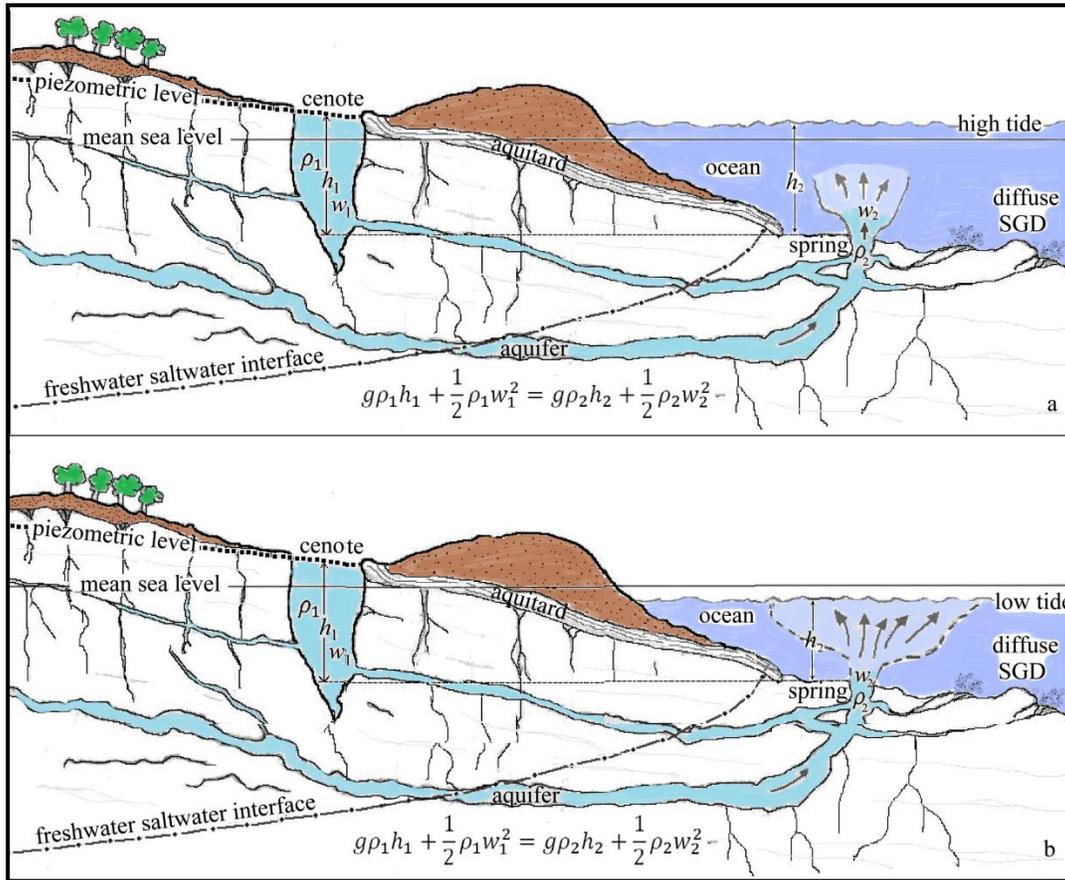


Figure 2. Point source of a SGD in a coastal Karst topography and the effect of low and high tides. Modified from Valle-Levinson et al., (2011).

### 3.1.1. Estimation and measurement of submarine groundwater discharge

Due to the difficulty of finding and measuring SGD, this subject has been neglected scientifically. Information about SGD has been made mainly anecdotally, even though its existence has been realized for centuries (W. Burnett *et al.*, 2006).

The volume of SGD has two components (W Burnett *et al.*, 2006): i) the net fresh groundwater discharge driven by terrestrial hydraulic gradients, and ii) the recirculated seawater as a consequence of sea level variations, wave set-up, tides, density-driven convection and thermal convection. Some authors mean with the term SGD only the freshwater component of the outflow, whilst others refer to all direct discharge of subsurface fluids across the seafloor, no matter where it comes from. This discrepancies has caused the estimation ranges of SGD to differ significantly from each other (W Burnett *et al.*, 2006).

Worldwide estimations of SGD have been conducted by Church (1996), in which SGD range from 0.1 % to 10 % of the surface water runoff. It has been observed that the input of freshwater in bays and estuaries

can be relatively large, and in some places, it ranges from 10 % to 20 % of the water entering the bay (Paulsen *et al.*, 2004). According to W. C. Burnett *et al.* (2006) about 6 % of the world’s river discharge is represented by fresh groundwater discharge to the ocean (2400 km<sup>3</sup>). Regarding discharge rates, significant differences are found between seepage fluxes of groundwater and spring discharges: seepage fluxes can be relatively slow, e.g. less than 1 m/day (Taniguchi *et al.*, 2002), on the other, hand spring discharges can be relatively fast, e.g. ~1 m/s (Valle-Levinson *et al.*, 2011). Slow seepages may have a wider area of discharge, thus resulting in a larger flux of water volume. Nevertheless, point source SGD (spring discharges) represent a faster connection between the sea and the freshwater aquifer (Parra *et al.*, 2014).

The assessment of SGD is performed based on three basic approaches (W. Burnett *et al.*, 2006):

- a) Modelling approaches can go from simple on-shore groundwater balance calculations through to more complex numerical models of sub-surface flow. Depending on the complexity of the model, it could require a computer with great performance and capacity, which could be expensive. Also, it has to be taken in consideration the mathematical simplicity, conformity with physical theory and numerical compatibility with methods. On the other hand this approach has several advantages, such as: it doesn’t require sophisticated meters; the user can “control” external variables; it is useful to predict scenarios or obtain a preliminary result of the behavior expected of the studied phenomenon.
- b) Direct physical measurement. Although several variations in design of meters have been developed (e.g. seepage meters and piezometers), it is limited to seepage flux meters and measure of the direction and magnitude of hydraulic gradients across the sediment-water interface. The two main problems in this method are that two or more approaches are rarely employed in one study. There are so many assumptions made in the calculation, therefore errors are rarely reported because it is extremely difficult to set reasonable uncertainty limits on the final results. Valle-Levinson *et al.*, (2011) and Parra *et al.*, (2014) have conducted direct measurements with a propeller type current meter.
- c) Tracer Techniques. Natural geochemical species (radium isotopes, radon, methane, artificial tracers, etc.) or artificial tracers are used for this purpose. An advantage of groundwater tracer is that they present an integrated signal as they enter the marine water column via various pathways in the aquifer (W. Burnett *et al.*, 2006). One limitation is that this method require that all other tracer sources and sinks except groundwater be evaluated, which is often a difficult task.

Units of measurement commonly applied to SGD are (Taniguchi *et al.*, 2002):

1. Volume per unit time (  $\frac{m^3}{day}, \frac{l}{s}$  ). The flux rate for point source SGD and larger-scale studies are usually measured with these units.
2. Volume per unit time per unit length of shoreline (  $\frac{m^3}{(day)*(mshoreline)}, \frac{m^3}{(year)*(kmshoreline)}$  ). These units are appropriate for smaller-scale investigations (local or regional scale).

3. Volume per unit time per unit area which is Darcy's flux ( $\frac{cm^3}{cm^2*s}, \frac{cm}{s}, \frac{m}{año}$ ). The diffusive SGD is usually measured with these units. This requires knowing the size of the area subject to the SGD flux.

The values are compatible when the discharge area is well-known but in most of the cases the problem arises because the area extent of offshore SGD is unknown.

### 3.1.2. Coastal Karst aquifers

The aquifers are of special interest since they represent an important freshwater reservoir for highly populated areas. They can be found in great heights in the mountains and at the sea level as long as the right geological, weather and morphological conditions exist. Coastal karst aquifers can be developed in areas with soluble rocks. Of greatest hydrogeological interest are the karsts composed by carbonated materials (Fernández Rubio and Baquero Úbeda, 2006) or limestone, which are easily dissolved with water that contains gaseous CO<sub>2</sub>. This causes the formation of conduits and caverns that can store large amounts of freshwater infiltrated from the surface.

Karsts aquifers have a complex geological history, which begins with carbonated formations deposited under the ocean for thousands of years. After a lithification process, those materials may be brought to the surface, being thus exposed to karstification due to the geochemical processes. As a consequence, an increase of secondary permeability takes place because of the dissolution of karst conduits. Example cases of this process occurs at the Peninsula of Yucatan, Mexico, at the Nullabar plain in southern Australia and at the Portugal coast. Karstified materials present primary permeability (which depends on interconnecting pores between the grains of the material) and/or secondary permeability due to clefts, fractures and dissolution conduits.

Dissolution processes can occur from few meters depth to hundreds of meters beneath the piezometric level. The typical water circulation in a coastal karstic aquifer is described in Figure 3 where the underground conduits through which the freshwater flows toward the shore and discharge into the sea can be observed. Depending on factors such as the piezometric level and the wave/tidal pumping, flux direction can be inverted, that is, saltwater inflow into the aquifer. Consequently, there can exist a freshwater zone, a mixing zone (brackish water) and a saltwater zone.

Fluctuations of sea level due to tectonic or glacial processes cause vertical displacements of the transition fringe of fresh and saltwater (Figure 3). This also influences the dissolution processes at different depths (Fernández Rubio and Baquero Úbeda, 2006). The geographical distribution of the karstifiable rocks isn't uniform around the world and the karst aquifers are mostly located between parallels 30°N and 50°N.

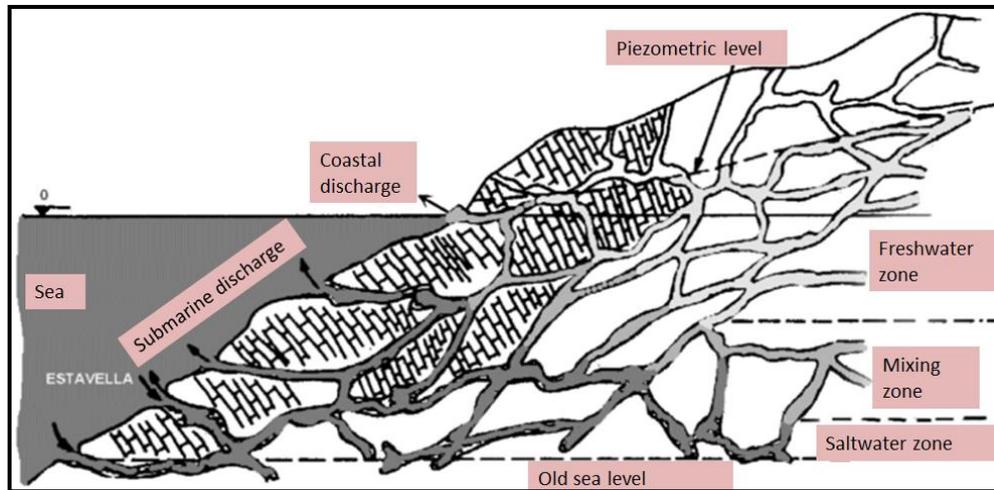


Figure 3. Karstic circulation diagram in the karstic littoral zone (Mijatovic, 1987)

The geographical distribution of the karstifiable rocks isn't uniform around the world. The karst aquifers are mostly located between parallels 30°N and 50°N.

### 3.1.3. The Karstic aquifer of the Yucatan Peninsula

As already mentioned, the Peninsula of Yucatan is one of the places where karstification processes are highly relevant. The Peninsula, with an area of  $\sim 350\,000\text{ km}^2$ , is located in the eastern part of the coastal plain at the Gulf of Mexico. The northern portion of the Peninsula is of karstic nature and its superficial geology consists of sedimentary rocks with carbonate deposits (limestones, dolomites and evaporites) (Figure 4). The summer rain and the presence of hurricanes generates large precipitations, which recharge the aquifer system. Due to the high hydraulic conductivity ( $k = 0.1 - 2000\text{ m/day}$ ) and porosity ( $n = 0.05 - 0.5$ ) of the aquifers, the water travels rapidly into the coastal zones and finally discharges as submarine groundwater discharge (Escolero *et al.*, 2007).

Flow paths in the Yucatan karst aquifer occur over a range of different scales: a) at regional-scale fracture zones of 10 to hundreds of kilometers; b) at large dissolution conduits of 1 to tenths of kilometers and/or c) at small-scale fractures and dissolution cavities of tenths of meters (Bauer-Gottwein *et al.*, 2011). Also, the groundwater flow in the conduit systems is considered turbulent.

Data taken from the Yucatan Peninsula Karst Aquifer review of Bauer-Gottwein *et al.* (2011) reveals An average annual rainfall ranging from 550 to 1500 mm per year in the Peninsula, which contributes to 14-17 % of the recharge of the aquifers (around 150 mm per year). Three groundwater outflows are mainly noticed in the Peninsula: i) coastal outflow, within average of  $0.73\text{ m}^3/\text{s}$  per km of coastline; ii) pumping of the water for human consumption (the amount of groundwater pumping from the Peninsula is not known precisely because of weak monitoring infrastructure) and iii) outflow due to phreatic evapotranspiration, which varies from 350 to 2500 mm per year.

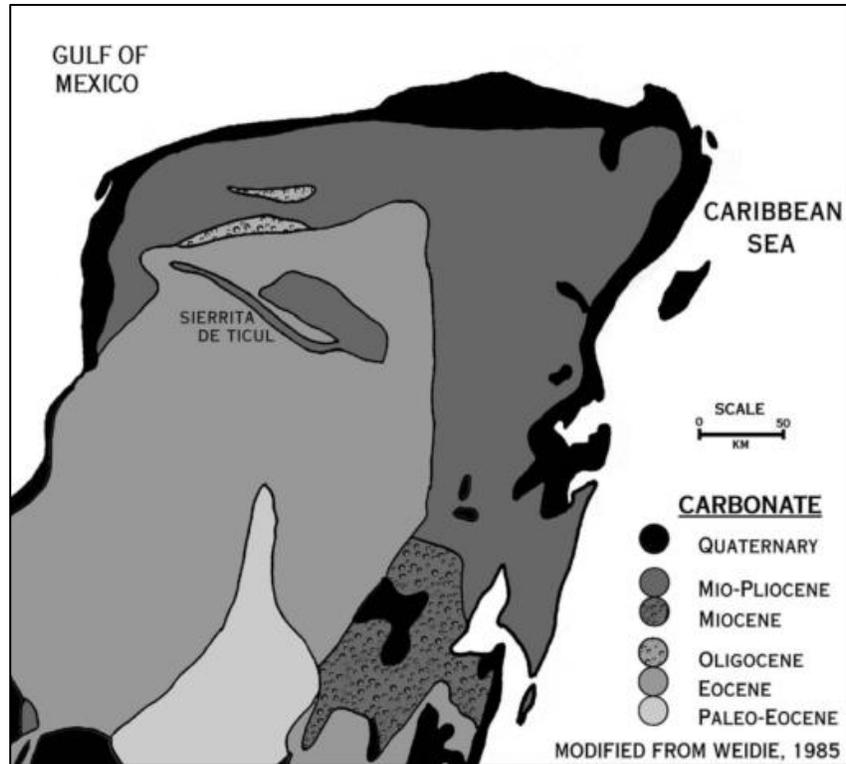


Figure 4. Carbonate geology of the Yucatan Peninsula showing the off-lapping sequences of successively younger carbonate deposits. From Beddows (2004).

The Yucatan Peninsula karst has a great importance as it is one of the most extensive and spectacular karst aquifer systems in the world and hosts large amounts of groundwater resources, which maintain several ecosystems. However, this aquifer system is highly affected by saltwater intrusion as well as by anthropogenic pollution. It is important to understand how SGD behaves because it influences the growth of coral reefs, especially in the Mexican Caribbean. Recently, it has been detected that SGD causes discontinuities in the reef system of Puerto Morelos Lagoon, due to the pollutants that decrease pH (Ruiz, 2012). Coral reefs are important for the preservation of species and also for the coastal protection because of their ability to dissipate wave energy. Thus the study of the preservation factors of coral reefs, such as SGD, concerns both ecologist and engineers.

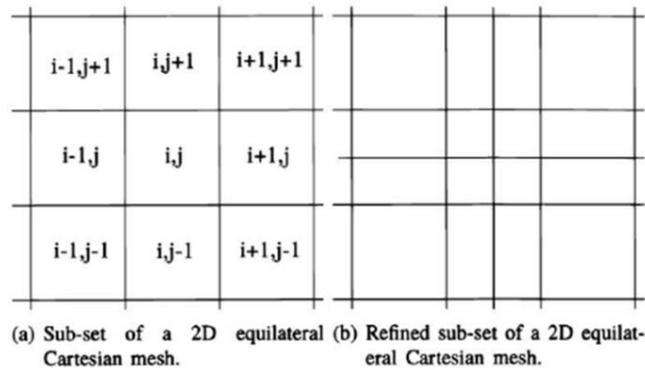
## 3.2. NUMERICAL MODELING IN COMPUTATIONAL FLUID DYNAMICS

### 3.2.1. Background and generalities

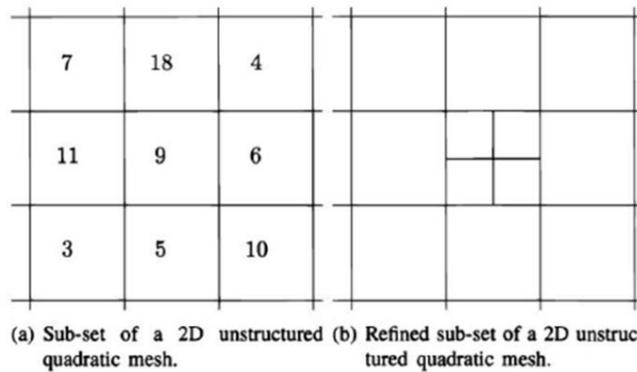
The numerical modeling is a technique used to solve a mathematic model when the “exact” or “analytical” procedures are not able to provide an answer. This method consists in formulating a mathematical model in such a way that it can be solved by arithmetic operations and the programming of algorithms. The computational fluid dynamics (CFD) uses numerical modeling to simulate the performance of fluids under

a stated condition. As a consequence of the development of the digital computer, modern CFD techniques improved its results since their beginnings in the 1950s.

Among other things, the solution of a numerical model relies on the correct implementation of the numerical mesh, which affects the precision, the speed and the stability of the solution. There are 2 types of meshes, the structured and the unstructured (Figure 5); which have different requirements in terms of a) domain discretization, which consist in the process of transferring a continuous domain into discrete counterparts b) equation discretization and c) the style the source code has to be formulated. In a summarized form, in structured meshes the cells are labeled with the indexes increasing in the directions of the coordinate axis; while unstructured meshes may not have apparent direction.



Structured quadratic mesh.



Unstructured quadratic mesh.

Figure 5. Difference between a structured and an unstructured mesh. From: Tomislav *et al.* (2014)

The numerical modeling and particularly CFD are based on three main discretization methods for the solution of partial differential equations (PDE):

- a) Finite difference methods (FDM).

FDM are distinguished because their formulation is relatively easy (compared to other methods). These methods are normally based on the application of the Fourier series, Taylor series expansions and the

Legendre polynomial. Some limitations of FDM methods are found since structured meshes are needed for their implementation. For multidimensional problems, it is necessary to transform curved meshes into structured meshes in order to work within an orthogonal Cartesian coordinate system.

b) Finite element methods (FEM)

FEM methods are based on the separation of the computational domain into a smaller number of regions (cells), thus needing a mathematical rigor for their specification and the solution of the formulations. Like the FDM, the solution of the PDE is approximated in the computational domain and each cell is a mesh element composed by nodes and vertices. FEM methods have the advantage of not needing a transformation of coordinates of an unstructured mesh into a Cartesian coordinate mesh and the Neumann boundary conditions (the value of the derivative of the solution is imposed which means, for example, to determine the flux rate in the boundary) are enforced exactly (Cung T. J, 2002).

c) Finite volume methods (FVM).

For FVM, the formulations could be based either on FDM or FEM. This approach works by integrating the PDE to be solved. The surface integrals of normal fluxes in FVM guarantee the conservation properties through the domain, which allows the usage of complex geometries and unstructured meshes without a coordinate transformation. In that sense, FVM are more powerful and they offer the possibility for working with refined meshes if necessary; i.e., manipulating the cells sizes along the mesh instead of having a very refined/coarse mesh through the whole domain. In this case, the mesh is composed by control volumes or cells with the PDE's solved in each cell.

Since the creation of FDM and FEM for stress analysis, there have been earlier applications in the fields of fluid dynamics and heat transfer (Cung T. J, 2002). However, current CDF mainly uses the FEM and FVM because they are able to handle complicated geometries than FDM.

### 3.2.2. Conservation principles

In fluid mechanics, it is convenient to consider the flow within a certain spatial region called *control volume* (CV) instead of a given substance quantity or *control mass* (CM). The law of mass conservation relates the changing rate of an extensive property in a given control mass. For mass ( $m$ ) conservation, the equation can be written as in Equation 1, which means that mass can be neither created nor destroyed:

$$\frac{dm}{dt} = 0 \quad \text{Equation 1}$$

On the other hand, the Newton's second law of motion leads to the momentum conservation equation as depicted in Equation 2:

$$\frac{d(mU)}{dt} = \sum f \quad \text{Equation 2}$$

where  $t$  is time,  $U$  the velocity and  $f$  the forces acting on the control mass.

The mass and the momentum are both extensive properties. If  $\phi$  is any conserved intensive property per unit of mass, the corresponding extensive property  $\Phi$  can be expressed as:

$$\Phi = \int_{\Omega_{CM}} \rho \phi d\Omega \quad \text{Equation 3}$$

where  $\Omega_{CM}$  is the volume occupied by the CM. Using this definition, an equation called control volume equation or the Reynold's transport theorem is defined. This equation states that the changing rate of the property in the control mass,  $\Phi$  is equal to the rate of change of the property (within the control volume) plus the net flux of it through the CV boundary, which is caused by the fluid motion relative to the CV boundary (Ferziger and Peric, 2002). Thus, for a given fixed CV, the control volume equation can be expressed as:

$$\frac{d}{dt} \int_{\Omega_{CM}} \rho \phi d\Omega = \frac{\partial}{\partial t} \int_{\Omega_{CV}} \rho \phi d\Omega + \int_{S_{CV}} \rho \phi \mathbf{U} \cdot \mathbf{n} dS \quad \text{Equation 4}$$

where  $\Omega_{CV}$  stands for the CV volume,  $S_{CV}$  is the surface enclosing the CV,  $\mathbf{n}$  is the unit vector orthogonal to  $S_{CV}$  and directed outwards (Figure 6). The first and second term on the right side of Equation 4 the temporal and the convective (advective) terms, respectively.

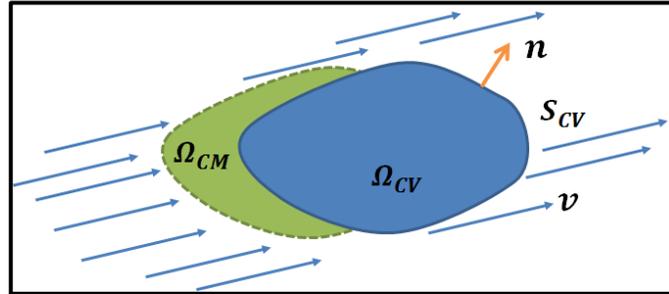


Figure 6. Control volume approach for the analysis of fluid field flow.

Three conservation equations follow directly from the control volume equation: mass, momentum and scalar conservation.

### Mass conservation

Mass conservation equation is obtained by doing  $\phi = 1$  and the Equation 4 becomes the left side of Equation 1Equation :

$$\frac{\partial}{\partial t} \int_{\Omega_{CV}} \rho d\Omega + \int_{S_{CV}} \rho \mathbf{U} \cdot \mathbf{n} dS = 0 \quad \text{Equation 5}$$

The convection term is transformed from a surface integral into a volume integral by applying the Gauss divergence theorem. If the control volume is infinitesimally small, the differential coordinate-free form of the continuity equation (without reference to any particular coordinate system) is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0 \quad \text{Equation 6}$$

### Momentum conservation

The momentum conservation equation is obtained by assuming  $\phi = U$  :

$$\frac{\partial}{\partial t} \int_{\Omega_{CV}} \rho U d\Omega + \int_{S_{CV}} \rho U U \cdot n dS = \sum f \quad \text{Equation 7}$$

Considering surface and body forces, momentum equation (Equation 7) can be expressed as

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) = \nabla \cdot (\mu \nabla U) - \nabla \cdot (P) + \rho b \quad \text{Equation 8}$$

where  $\mu$  is the dynamic viscosity,  $P$  is the total pressure and  $b$  is a source/sink term (body forces per unit mass).

### Conservation of scalar quantities

The third conservation equation is the conservation of scalar quantities. The integral form is:

$$\frac{\partial}{\partial t} \int_{\Omega_{CV}} \rho \phi d\Omega + \int_{S_{CV}} \rho \phi U \cdot n dS = \sum f_{\phi} \quad \text{Equation 9}$$

where  $f_{\phi}$  is the transport of  $\phi$  by mechanisms other than convection and any sources or sinks of the scalar (e.g. diffusive transport described by Fick's law for mass diffusion). Analogous to the previous equations, this equation can be written as:

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \phi U) = \nabla \cdot (D \nabla \phi) + q_{\phi} \quad \text{Equation 10}$$

where  $D$  is the diffusivity for the quantity  $\phi$  and  $q_{\phi}$  is a source/sink term of  $\phi$ . The first term of the right side of this equation is the diffusive term.

A detailed description of the obtainment of the right side of Equation 8 and Equation 10 is given by Ferziger & Peric, (2002).

### 3.2.3. CFD software

A brief description of some examples of the most popular CFD software is presented in this section. Nowadays, there are numerous software toolkits that are used for fluid flow analysis, simulation and modelling in CFD. The software might be composed by numerical simulation codes, data analysis packages, symbolic math packages and/or visualization packages. Within this context, some examples of current CFD software available for fluid dynamics simulations are:

- OpenFOAM®. It is a free, open source software package developed by OpenCFD (<http://www.openfoam.com/>). The key advantage of this CFD toolkit is that the user can freely use and modify the CFD code, which can be used as common platform for collaborative projects between developers. A considerable amount of effort is needed in order to learn how to use and extend the platform, particularly if it is compared to several commercial CFD software, whose performance, governing equations or their implementation remain as a black-box to the user. This tool is gaining popularity because of its potential and the costs reduction for engineers due to the elimination of license fees (Tomislav et al., 2014) as well as for the possibility of extending/changing/reviewing the codes originally implemented. This tool includes several solvers capable to work with porous medium, multiphase flow, waves, free surface and different turbulence models.
- ANSYS CFX®. This software is a high-performance, general purpose CFD program developed and used for over 20 years. ANSYS® has its solvers wrapped in a modern, intuitive, and flexible user environment based on FVM. It is not an open source and proprietary software and license costs can ascend to thousand dollars (ANSYS, 2015). This software is especially useful for engineering, design and analysis (e.g., gas in turbine engines, aircraft aerodynamics, pumps, fans, fluid dynamics of ship hulls, vacuum cleaners and general fluid flows in a virtual environment). This model supports multiphase modeling, porous medium, wave simulation and different turbulence models.
- FLOW-3D®. It is a package for simulation and post-processing free-surface flows with advantages such as the no addition of extra-modules; simplification of the mesh to be used by the models and an interface for collaborators and clients who are not necessarily professionals in the CFD field. Similarly to ANSYS CFX®, FLOW-3D® is a non-open source. However, there are some open codes available, so that the user can customize new turbulence models, new contour conditions, etc. (FLOWScience, 2015). It supports multiphase modeling, porous medium, wave propagation and different turbulence models.
- OpenFlower®: This is an open source CFD software written in C++ and it is mainly useful for the resolution of the turbulent unsteady incompressible Navier-Stokes equations with a Large Eddy Simulation approach (OpenFlower, 2004). The advantage of OpenFlower® is that it uses FVM, so it can deal with arbitrary complex 3D Geometries. It also works for either Linux or Windows.
- Other CFD software may be: Autodesk CFD®, FLASH®, Star-Cd®, CFDRC®, GADGET®, HYDRA®, ANSYS CFD®, etc.

#### 3.2.4. Multiphase flow modeling

Any fluid flow consisting on more than one phase or component is called multiphase flow (Figure 7). According to the state of the different phases or components, multiphase flows can be classified as gas/solid, liquid/solid, gas/particle, bubbly flows, etc. (Brennen, 2005). Information about multiphase flows is focused on specific types of fluid flow (e.g. low Reynolds number suspension flows and dusty gas dynamics) or specific applications such as slurry flows and cavitating flows.

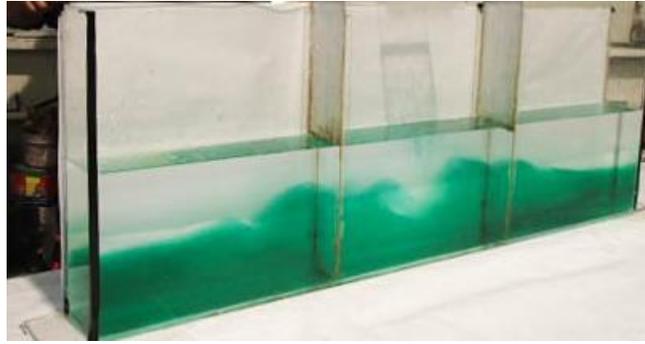


Figure 7. Example of a multiphase flow. The image shows a 3-fluid system: air (free-surface condition), freshwater (transparent) and saltwater (green). Photograph and laboratory model by the author of this thesis.

The ability to predict the fluid flow behavior is critical to assess the efficiency and effectiveness of processes that involve multiphase flows, such as cavitating pumps and turbines. According to Brennen (2005) there are three ways in which the multiphase fluid motion can be explored:

1. Experimentally, through laboratory-sized models equipped with appropriate instrumentation. Although there are many applications in which full-scale laboratory models are possible, in many instances, the laboratory model must have a significant different scale than the prototype. There are also cases in which laboratory models are not feasible for a variety of reasons.
2. Theoretically, by considering analytical and/or mathematical equations to model the fluid motion. This method is limited as it requires many simplification and many complex cases that cannot be solved with enough accuracy or necessary detail.
3. Computationally, using the power and size of modern computers to address the complexity of the flow. The computer power and speed are relevant elements to consider when modeling the flows that are commonly experienced, such as turbulent flows. A reliable theoretical or computational model is thus needed to provide a confident extrapolation to the scale of prototypes.

For a free-surface fluid flow condition, such as waves in the sea, there exist a numerical technique for tracking and locating the free surface or fluid-fluid interface. This technique is used in CFD tools and is called Volume of Fluid (VOF) method, which is detailed by Berberović *et al.*, (2009). VOF method consists on tracking each fluid through every cell in the computational grid (while all fluids share a single set of momentum equations) using a scalar function that takes the value of zero when the cell is empty, one when the cell is full, and values between zero and one when a fluid interface is found (Figure 8).

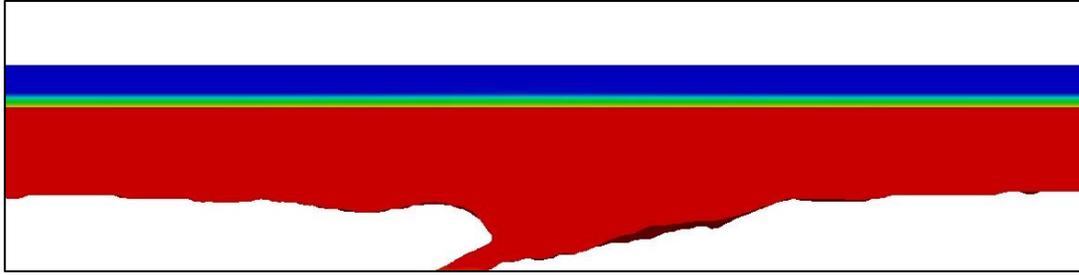


Figure 8. Example of a CFD simulation in which VOF method is coupled with the momentum equation. The red area is filled with water and the blue area is filled with air. The interface between air and water is the thin layer in green and yellow.

The VOF method introduces only one extra equation, thus it is computationally friendly. This method is shown to be more flexible and efficient than other methods for treating complicated free boundary configurations and sharp topological changes in non-linear problems (Hirt and Nichols, 1981).

### 3.2.5. Turbulence modeling

Turbulence is characterized by several observable properties such as: Irregular signal in space and time, rotational flow, high diffusivity, unpredictable character of trajectories, coexistence of eddies of very different scales and dissipation (Figure 9). Ferziger and Peric (2002) describe the turbulent flow with the following features: i) it is highly unsteady; ii) it is a three-dimensional phenomenon; iii) the vorticity is one of its principal mechanisms; iv) it includes a process called turbulent diffusion; v) it brings fluids of different momentum content into contact; vi) it produces a reduction of the kinetic energy of the flow by a dissipative mixing process which is irreversible; vii) the turbulent flow contains coherent structures but also a random component; and viii) it fluctuates on a broad range of length and time scales which make the numerical simulation a very difficult task.

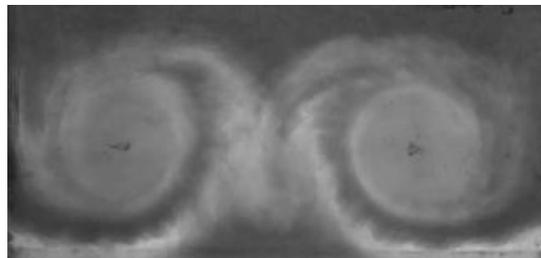


Figure 9. Example of a turbulent flow. The properties of a turbulent flow such as eddies of different scale, high diffusivity and rotational flow, are present.

There are different approaches to model turbulence in fluids simulations, which have been developed over a praxis framework in order to model and simulate the phenomenon. Those approaches have different levels of complexity, for instance: a) statistical models with a turbulence-viscosity hypothesis, or b) advanced models that involve transport equations or turbulent large eddy numerical simulations. However, each approach has its own performance advantages and limitations. They are not considered to compete but are complementary and are chosen depending on the type of problem to be solved as

well as the response to be expected (Schiestel, 2008). Turbulent flows contain variations on a much wider range of length and time scales than laminar flows. The most common approaches are herein briefly described and further references can be found in Bardina *et al.* (1997), Ferziger and Peric (2002), Mcdonough (2007) and Kobayashi and Tsubokura (2011).

- Reynolds-Averaged Simulation (RAS)

This approach is called one-point closure and is based on the equations obtained by averaging the equations of motion over time. The Reynolds-averaged Navier-Stokes equations (RANS) do not form a closed set of equations so the introduction of approximations (models) is required. Those approximations are known as turbulence models (Ferziger and Peric, 2002). The most common turbulence models are, among others, the two equation models. Those models add two extra transport equations to represent the turbulent properties of the flow. One of the transported variables is the turbulent kinetic energy, which is represented with the letter  $k$ . The second variable depends on what type of model it is being used (e.g.,  $k$ - $\epsilon$  where  $\epsilon$  is the turbulent dissipation and determines the rate of dissipation of the turbulent kinetic energy,  $k$ - $\omega$ , where  $\omega$  is the specific dissipation, and  $k$ - $k_L$ , where  $k_L$  is the turbulent length scale). The introduction of this wide range of RANS approaches began around 1972 and are one of the most common type of turbulence models used in the industry and for most of engineering problems.

- Large eddy simulation (LES)

Proposed by Deardorff in 1970, LES was the first computational technique for turbulence modeling. This approach solves the largest scale motions of the flow and approximates (models) the small scale motions. It is one of the most promising turbulent simulations for engineering purpose (Kobayashi and Tsubokura, 2011). Although LES was not feasible for practical engineering problems in the 70s and 80s (McDonough, 2007); the transition from RAS to LES was enhanced thanks to the development High-Performance Computing technology and the progress of parallel-processor computers (Kobayashi and Tsubokura, 2011).

- Direct numerical simulation (DNS)

In this approach the Navier-Stokes equations (NS) are solved for all the scales of the turbulent fluid flow. The first Direct Numerical Simulation (DNS) was introduced by Orszag and Patterson (1972). The equations are solved without averaging or approximation other than the numerical discretization. Therefore, it is the most accurate approach. However, this method has several limitations and thus its major role has been focuses as a research tool (Ferziger and Peric, 2002). The limitations of DNS are that it is computationally expensive for practical simulations.

### 3.3. NUMERICAL MODELLING APPLIED FOR SGD DESCRIPTION

Recent research exists related to the characterization of coastal processes with SGD, particularly focused on: a) its occurrence (Bakalowicz, 2014), b) relationship with hydrological processes (Beddows, 2004), c) quantification and measurement techniques (Loaiciga and Zektser, 2003, and W. Burnett et al., 2006), d) chemical interchange and input of nutrients (Johannes, 1980). Parra *et al.* (2014) observed the influence of sea level variations due to tides and wave setup on turbulent kinetic energy at a point source SGD in a

coral reef lagoon in the Peninsula of Yucatan. Null et al. (2014) demonstrated the importance of considering the beach zone as a significant nutrient source for the coastal zone. The importance of understanding the connectivity of submarine spring discharge with the nearshore environment was identified as very significant to assess the impact of inland anthropogenic activities on coastal human/environmental health.

The SGD has been widely analyzed in terms of numerical modeling and hydrodynamics but mainly oriented on the analysis of the underground and groundwater flow; e.g. Parra *et al.* (2015) and Bokuniewicz *et al.* (2004). Recently, Shishaye (2015) has presented the impact of SGD on the position of saltwater-freshwater interface by modeling the groundwater flow. Also, Qu *et al.* (2014) has conducted numerical simulations of steady-state salinity distribution and SGD in homogeneous anisotropic coastal aquifers.

In the field of CFD numerical modeling, Jacobsen *et al.* (2011) have shown the usage of the CFD library OpenFOAM® for the modelling of waves by extending the method for solving free surface Newtonian flows included in OpenFOAM® as well as for the inclusion of porous media (Jacobsen *et al.*, 2011; Alcérreca-Huerta and Oumeraci, 2016). These models uses the Reynolds-Averaged Navier-Stokes (RANS) equations coupled with a volume of fluid method (VOF). In this way, they are able to represent the turbulence distribution in the domain by considering the interaction between two immiscible and incompressible fluids.

Further studies and research has been made regarding turbulence associated with wave motion. Thais and Magnaudet (1996) conducted large-scale tests in a laboratory wave flume with pure wind waves and wind-ruffled waves (mechanically generated) in order to explore the structure of turbulent motion induced by surface gravity waves. The turbulent structure was found to be affected at both large and small scales by the wave motion. The phase variability was also analyzed and direct interaction between the waves and the underlying turbulence (particularly at the wave crests) was found, with turbulence generation just above the wave troughs. The analysis of the turbulence velocity spectrum showed that the turbulence is dynamically forced by the surface wave. Further references regarding numerical modelling of the turbulence induced by wave action can be found in Guo and Shen (2013) and Yang and Shen (2010).

### **3.4. SUMMARY AND IMPLICATIONS FOR THE STUDY**

After reviewing the state of the art as well as similar studies and available CFD tools, it is clear that there still exists a knowledge gap in the field of SGD and its numerical simulation. In order to get a detailed description of point source SGD (“ojos”), an appropriate numerical tool must be developed.

SGD and point source discharge are processes that have been identified and their importance recognized. As mentioned, a tool is required in order to predict and better understand the performance of aquifer systems and of SGD under different scenarios of waves, sea level and flow rate. High-quality results can be obtained with an appropriate numerical model, which should be able to simulate/consider: i) waves generation/absorption); ii) the interaction between three different density phases (air, saltwater and fresh/brackish water); iii) inclusion of miscible fluid phases; and iv) turbulence assessment through the

implementation of an appropriate turbulence model. After a comprehensive review of available CFD software, up to now no CFD model can handle simultaneously with all the features aforementioned. Thus, an appropriate tool must be developed in order to correctly and accurately describe the performance of a point source of SGD.

OpenFOAM® is a free, open source software package which has gained popularity due to its multiple advantages over free and proprietary CFD codes. One of the main advantages of OpenFOAM® over many other CFD codes is that the user can freely use and modify the programming code, thus allowing the generation of new codes without starting from scratch. Additionally, a native solver in OpenFOAM®, named interMixingFoam, is already available for the simulation of three incompressible fluids, two of which are miscible, using a VOF method to capture the interface. Thus, it can be used as a platform for the addition of a wave field and for the integration of a turbulence model. The later is of great relevance for the present thesis, since it would allow a detailed description of the hydrodynamic phenomenon in point sources of SGDs.

Within this context, the main purpose of this work is thus focused on the development of a numerical model for the simulation of a three-phase fluid system (i.e., air, saltwater and freshwater) in order to be applied for the analysis of a study case with SGD in the Peninsula of Yucatan, Mexico, and to other cases involving fluid phases interaction in terms of density gradients and wave motion.

# DEVELOPMENT OF THE CFD MODEL

## 4. DEVELOPMENT OF THE CFD MODEL

This chapter presents a description of the process that was followed to extend the OpenFOAM® CFD platform in order to simulate the study case of interest: a point-source SGD located in the Peninsula of Yucatan, Mexico. The resultant numerical model is capable to reproduce: i) waves (generation and absorption); ii) three different density phases (air, saltwater and fresh/brackish water); iii) mixable liquid phases; iv) turbulence with an appropriate turbulence model. The governing equations of the model and the process carried out to extend the native codes included in OpenFOAM® are described in detail in this chapter.

### 4.1. GOVERNING EQUATIONS

The momentum and the continuity equations are usually known as the Navier-Stokes Equations (NSE) and are particular cases of the *control volume equation* (Equation 4). NSE are non-linear and coupled equations that are able to accurately describe the flow of a Newtonian fluid (Ferziger and Peric, 2002). The momentum and the continuity equations or NSE are difficult to solve but simplifications are usually justified to reduce computational costs. For incompressible flow, the NSE can be expressed as:

$$\nabla \cdot U = 0 \quad \text{Equation 11}$$

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) = \nabla \cdot (\mu \nabla U) - \nabla(P) + \rho b \quad \text{Equation 12}$$

where  $U$  is the velocity,  $\rho$  the density,  $\mu$  the dynamic viscosity,  $P$  the total pressure and  $b$  is a source/sink term (body forces per unit mass).

In order to solve the NSE, an approximated solution is given by a time averaging process, which consists in the decomposition or separation of the flow variable into the mean (time-averaged component and the fluctuating component). If the flow variable is the velocity  $u$ , it is separated into the mean component  $\bar{u}$  and the fluctuating component  $u'$ , that is  $u(x, t) = \bar{u}(x) + u'(x, t)$ , where  $x = (x, y, z)$  is the position of the vector (Figure 10).

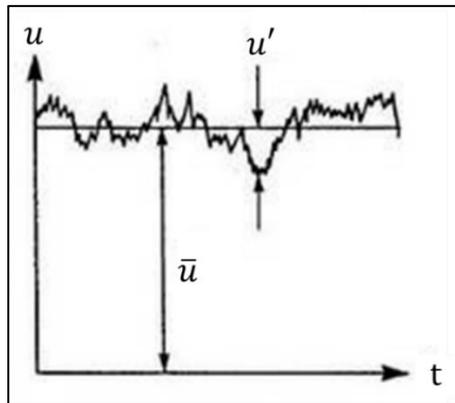


Figure 10. Decomposition of the flow variable of the velocity  $u$  into its mean component and the fluctuating component.

The averaged equations are known as Reynolds Average Navier-Stokes equations (RANS equations). These new set of equations are not a closed set of equations and therefore, turbulence models must be introduced. RANS equations can be written as in Equation 11 and Equation 13 for incompressible flows:

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) = \nabla \cdot (\mu \nabla U + \rho \tau) - \nabla(p^*) - g \cdot x \nabla \rho + \sigma_T \kappa_\gamma \nabla \gamma \quad \text{Equation 13}$$

where  $p^*$  is the dynamic pressure,  $g$  the acceleration due to gravity,  $x = (x, y, z)$  the Cartesian coordinates, and  $\tau$  is the specific Reynolds stress tensor (Equation 14):

$$\tau = \frac{2}{\rho} \mu_t S - \frac{2}{3} k I \quad \text{Equation 14}$$

where  $\mu_t$  is the dynamic eddy viscosity,  $S = \frac{1}{2}(\nabla U + (\nabla U)^T)$  the strain rate tensor,  $k$  the turbulent kinetic energy per unit mass, and  $I$  the Kronecker delta.

The last term in Equation 13 is the superficial tension effect, where  $\sigma_T$  is the superficial tension coefficient,  $\kappa_\gamma$  is the surface curvature and  $\gamma$  represents a scalar field for the identification of the different phases. The value of  $\gamma$  is obtained with the Volume of Fluid (VOF) method (Berberović et al., 2009). For a free surface fluid system,  $\gamma = 0$  for the air phase and  $\gamma = 1$  for the liquid. The intermediate values represent a mixture between the fluids (or interface).

The distribution of  $\gamma$  is modeled with the convective equation:

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\gamma U) + \nabla \cdot (U_r \gamma (1 - \gamma)) = 0 \quad \text{Equation 15}$$

where  $U_r$  is a relative velocity. The last term in the left side of Equation 15 is a compression term that limits the interface size.

The model developed in this study is a three-phase fluid system: two liquid phases (freshwater and saltwater) and one gas phase (air). The model code works by solving separately the fluid phases and their mixture, and then it incorporates them as one single fluid phase in the VOF method. For that, besides the governing equations given by the NSE, the model uses a supporting equation for scalar quantities conservation. This equation is also derived from the control volume equation (Equation 4) but written in terms of a scalar  $\phi$  (Equation 16).

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \phi U) = \nabla \cdot (D \nabla \phi) + q_\phi \quad \text{Equation 16}$$

The first term in the left side of the equation is known as the temporal term and the second one as the convective term. In the right side of the equation, the first term is the diffusive term and the second one any source/sink term. This equation is useful to represent scalar quantities in the model such as the different fluid phases involved (named "alphas" within the code). The diffusive term and its diffusivity term  $D$  are important to define the mixing between liquid phases. For the model developed,

the scalar  $\phi$  represents each of the three phases involved within the simulations: air, saltwater and freshwater.

## 4.2. EXTENSION OF THE SOFTWARE

OpenFOAM® was selected as the most appropriate CFD package to simulate point source SGD. This software contains different solvers in order to simulate a diversity of problems mostly related to fluid mechanics. Some examples of these solvers are:

- `IcoFoam`: this library solves isothermal, incompressible flow, for a single fluid phase.
- `InterFoam`: it is a transient solver that considers two incompressible fluid phases separated by a sharp interface or free surface.
- `InterMixingFoam`: this solver is derived from `InterFoam` but considers three incompressible phases, where two of them are mixable.
- `MultiphaseInterFoam`: this solver is able to simulate cases with three or more different phases, all of them considered as incompressible and not mixable.

Other solvers are related to different phenomena such as combustion, electromagnetics, heat transfer, stress analysis and financial problems. The solvers mentioned above are included in the official OpenFOAM® version that can be downloaded from the OpenFOAM® web page. Additionally, OpenFOAM® has an Open-Source General Public License, which allows the users to freely use and modify the CFD code. Thus, several extensions of the software have been developed by the user's community in order to simulate specific cases that are not considered in the official OpenFOAM® version. Extensions for OpenFOAM® are available in the contributions web page, which contains the libraries and solver's codes for downloading and compilation.

For the current work, OpenFOAM® version 2.2.0 was used and it was necessary to develop a new solver capable to handle wave generation/absorption, as well as the modelling of three fluid phases, where two of them are mixable. Turbulence models, mesh generation schemes and other generic OpenFOAM® utilities are available for the case setting and running. Thus, the new solver was not coded from scratch but from pre-existing libraries/solvers by adding/changing code lines, i.e., through the modification and integration of available codes included within the OpenFOAM® framework. One contribution that is particularly useful for this work is the library `waves2Foam`, developed and validated by Jacobsen *et al.* (2011). This toolbox is based on the `interFoam` solver and it is used to generate/absorb free surface water waves considering two incompressible fluid phases.

A detailed description of the process followed to extent the OpenFOAM® platform is now presented:

1. As a first stage, downloading and installation of OpenFOAM® is required, together with the extension toolbox `waves2Foam`. The `waves2Foam` directory contains subdirectories with the available applications (i.e. `waveFoam`, `porousWaveFoam`), utilities and sources, as well as dictionaries with the respective C++ codes. The solver `waveFoam` was used due to the available functionality of waves generation/absorption.

- For the second stage, a copy of the folder, where `interMixingFoam` solver is found (`opt/openfoam231/applications/solvers/multiphase/interFoam/interMixingFoam`), was made and saved with a new user-defined name within the `waveFoam` solver's directory (`home/.../waves2Foam/applications/solvers/solver220`) (Figure 11.). The chosen name of the new model developed was "`interMixingWaveFoam`" in agreement with the conventional names provided by OpenFOAM® and representative of the function of this new solver (Figure 12). The code file (.C extension) was also renamed as shown in Figure 13 where the content of the original `interMixingWaveFoam` folder can be observed. At this point, the folder's content is exactly the same as the one in `interMixingFoam`.

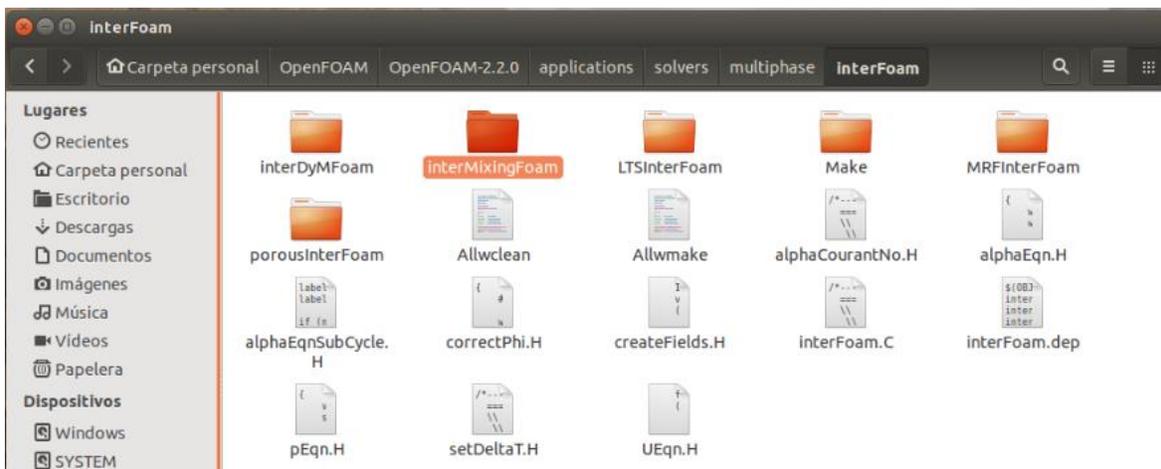


Figure 11. Location of the `interMixingFoam` folder. This folder was copied into the `waves2Foam` directory.



Figure 12. Location of the new solver named `interMixingWaveFoam`. The folder from the previous Figure was copied and pasted in this location on `waves2Foam` toolbox and its name was simply changed to `interMixingWaveFoam`.

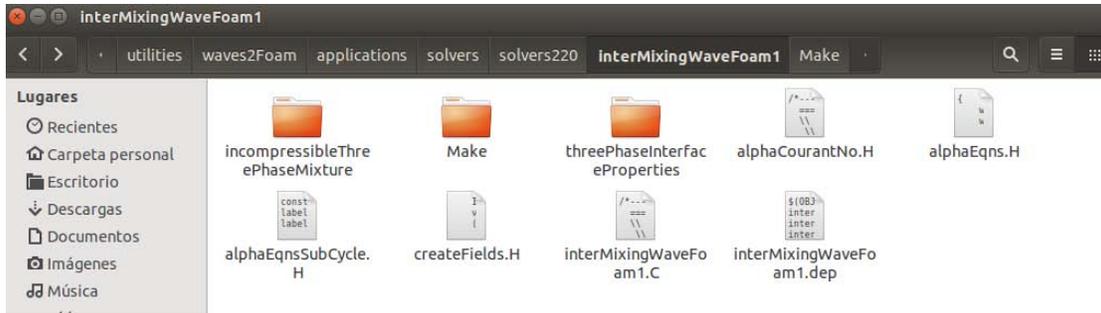


Figure 13. Original content of the new solver, which is exactly the same as the one in the `interMixingFoam` folder. The name given to the new solver is `interMixingWaveFoam`.

As stated previously, `interMixingFoam` was derived from the `interFoam` solver, thus some important files have to be added as they were included in the `interFoam` folder. The files added are `alphaEqn.H`, `alphaEqnSubCycle.H`, `correctPhi.H`, `pEqn.H`, `setDeltaT.H` and `UEqn.H` (Figure 14). Afterwards, the `Make` folder in `interMixingWaveFoam` was modified. This folder contains 2 files and a folder Figure 15, from which the `linux64GccDPOpt` folder is generated once the solver is compiled. The second file is `files`, which calls needed folders/libraries and the solver's C++ code (i.e. creates the files dependency tree). In `files`, the solver's name was modified as showed in Listing 1. The third file is `options` (Figure 15). `Options` contains a list of libraries needed to run the solver. The libraries required by `waveFoam` were added to this list (Listing 2.).

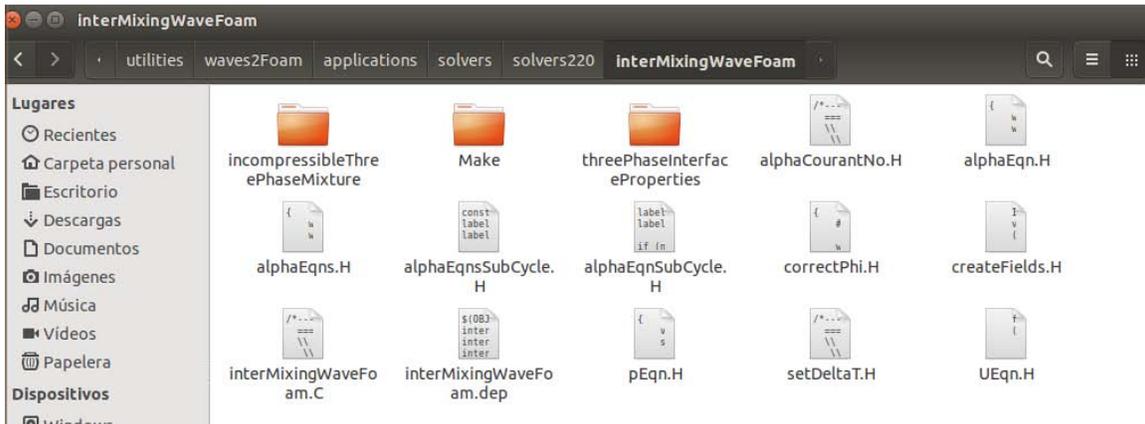


Figure 14. Missing files of `interFoam` were added to `interMixingWaveFoam`.



Figure 15. Files included in the `Make` directory.

Listing 1. The name of the new solver (`interMixingWaveFoam`) was introduced in files, in Make directory.

```
incompressibleThreePhaseMixture/threePhaseMixture.C
threePhaseInterfaceProperties/threePhaseInterfaceProperties.C
interMixingWaveFoam.C
EXE = $(FOAM_USER_APPBIN)/interMixingWaveFoam
```

Listing 2. List of libraries required by the solver. The list includes libraries from `interMixingFoam` and `waveFoam` and is contained within the options file, in Make directory.

```
EXE_INC = \
-I.. \
-IincompressibleThreePhaseMixture \
-IthreePhaseInterfaceProperties \
-I$(LIB_SRC)/transportModels/incompressible/lnInclude \
-I$(LIB_SRC)/transportModels/interfaceProperties/lnInclude \
-I$(LIB_SRC)/transportModels/twoPhaseInterfaceProperties/alphaContactAngle/
alphaContactAngle \
-I$(LIB_SRC)/turbulenceModels/incompressible/turbulenceModel \
-I$(LIB_SRC)/finiteVolume/lnInclude \
-I$(LIB_SRC)/transportModels \
-I$(LIB_SRC)/meshTools/lnInclude \
-I$(LIB_SRC)/fvOptions/lnInclude \
-I$(LIB_SRC)/sampling/lnInclude \
-DOFVERSION=220 \
-DEXTBRANCH=0 \
-DXVERSION=$(WAVES_XVERSION) \
-I$(WAVES_SRC)/waves2Foam/lnInclude \
-I$(WAVES_SRC)/waves2FoamSampling/lnInclude \
-I$(WAVES_GSL_INCLUDE)

EXE_LIBS = \
-ltwoPhaseInterfaceProperties \
-lincompressibleTransportModels \
-lincompressibleTurbulenceModel \
-lincompressibleRASModels \
-lincompressibleLESModels \
-lfiniteVolume \
-lmeshTools \
-lfvOptions \
-lsampling \
-L$(WAVES_LIBBIN) \
-lwaves2Foam \
-lwaves2FoamSampling \
-L$(WAVES_GSL_LIB) \
-lgsl \
-lgslcblas
```

3. As a final stage, `interMixingWaveFoam` files were modified by adding libraries and code lines of the original `waveFoam` directory, in order to implement wave generation and absorption. Some code lines of the preexisting files were modified in order to include instructions and libraries from `waveFoam`. The files that were modified are: `createFields.H` and `interMixingWaveFoam.C`. The original folders `incompressibleThreePhaseMixture`

and `threePhaseInterfaceProperties` were not modified since they are used by the solver in order to declare the three fluid phases and their properties. This step is detailed in the next section.

The Software extension is summarized in Figure 16. The process basically consisted in examining the preexisting OpenFOAM® and `waves2Foam` codes and equations, and recycling code lines/libraries from different solvers, in order to create one that could handle all the specifications required for this study.

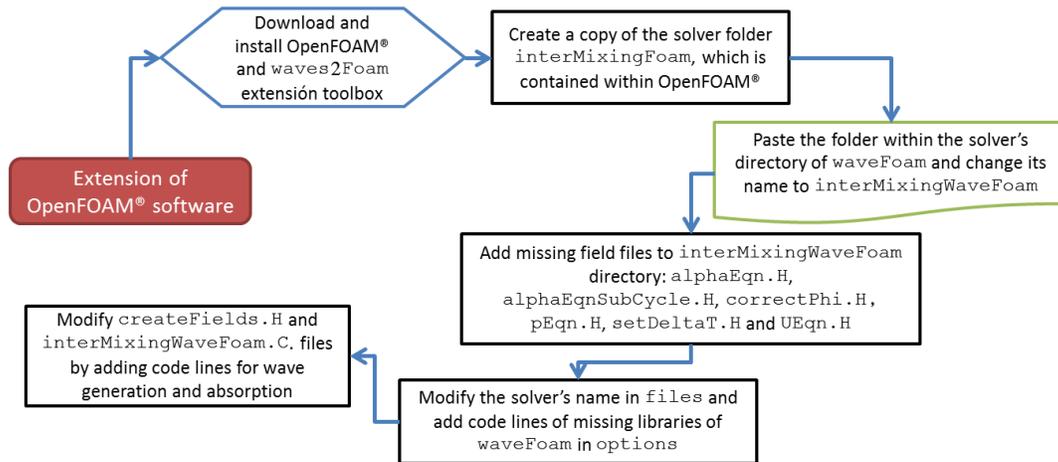


Figure 16. Flow chart of the extension of OpenFOAM® in order to build the new developed `interMixingWaveFoam` solver.

#### 4.2.1. Code modification

In the current section, description of the lines that were added/changed in each file is addressed. As mentioned in the previous section the files that were modified are `createFields.H` and `interMixingWaveFoam.C`, both located within the `interMixingWaveFoam.C` solver directory.

A code line in `createFields.H` was added in the declaration of the density field `rho` (Code 1). This instruction allows the printing of the `rho` field and its visualization after the case has been modelled. Moreover, a code line with the instruction `relaxationZone` was added at the end of the file, which creates a relaxation zone in the mesh that affects the velocity field and the fluid phase that represents saltwater (at the section where the waves are generated). This instruction has been referred to `alpha2` because this phase will be represented by one of the mixable fluids and is also affected by the wave generation/absorption (for the cases to be modelled it would represent the saltwater fluid phase).

Code 1. Code lines added to `createFields.H` in order to print and visualize the density field `rho` and to create relaxation zones.

```

...
dimensionedScalar D23(threePhaseProperties.lookup("D23"));

// Need to store rho for ddt(rho, U)
volScalarField rho
  
```

```

(
    IOobject
    (
        "rho",
        runTime.timeName(),
        mesh,
        IOobject::READ_IF_PRESENT,
        IOobject::AUTO_WRITE //added
    ),
    alpha1*rho1 + alpha2*rho2 + alpha3*rho3,
    alpha1.boundaryField().types()
);
rho.oldTime();

...

relaxationZone relaxing(mesh, U, alpha2); //Added

```

The file `interMixingWaveFoam.C` was also modified by adding missing libraries from `waveFoam.C`, as shown in Code 2. Those libraries allow the wave generation/absorption through relaxation zones and of an external wave forcing (Jacobsen et al., 2011). The relaxation zones and the external wave forcing are then implemented during the time loop of the solver for the calculation of the pressure, velocity and the interface between fluids.

Code 2. Libraries and code lines added to the `interMixingWaveFoam.C` file.

```

...
#include "fvCFD.H"
#include "MULES.H"
#include "subCycle.H"
#include "threePhaseMixture.H"
#include "threePhaseInterfaceProperties.H"
#include "turbulenceModel.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"
#include "relaxationZone.H" //added
#include "externalWaveForcing.H" //added

// * * * * *
int main(int argc, char *argv[])
{
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"

    pimpleControl pimple(mesh);
    #include "initContinuityErrs.H"
    #include "readGravitationalAcceleration.H"
    #include "readWaveProperties.H" //added
    #include "createExternalWaveForcing.H" //added
    #include "createFields.H"
    #include "readTimeControls.H"
    #include "correctPhi.H"
    #include "CourantNo.H"
    #include "setInitialDeltaT.H"

```

```

// * * * * *

Info<< "\nStarting time loop\n" << endl;
while (runTime.run())
{
    #include "readTimeControls.H"
    #include "CourantNo.H"
    #include "alphaCourantNo.H"
    #include "setDeltaT.H"

    runTime++;
    Info<< "Time = " << runTime.timeName() << nl << endl;
    externalWave->step(); //added
    threePhaseProperties.correct();

    #include "alphaEqnsSubCycle.H"
    relaxing.correct(); //added
    interface.correct();

    #define twoPhaseProperties threePhaseProperties
...
}

// Close down the external wave forcing in a nice manner
externalWave->close(); //added
Info<< "\n end \n";
return(0);
}
// ***** //

```

The files that were not modified are: alphaCourantNo.H, alphaEqn.H, alphaEqns.H, alphaEqnsSubCycle.H, alphaEqnSubCycle.H, correctPhi.H, pEqn.H, setDeltaT.H and UEqn.H.

After the aforementioned changes were made, the solver can be compiled by means of the command `wmake` in the terminal window and inside the `interMixingWaveFoam` directory. As a result an executable called `interMixingWaveFoam` is created. This executable runs the developed solver and allows the numerical simulation of a 3-phase fluid system, with 2 mixable phases, interface calculation between the 2 mixable phases and the third phase, wave generation and absorption, as well as all the functionalities of the `waves2Foam` toolbox, such as different relaxation shapes and wave theories. Furthermore, the generic OpenFOAM® platform libraries, allow the implementation of several utilities, such as different turbulence models or different solution schemes.

# **CALIBRATION AND VALIDATION OF THE MODEL**

## 5. CALIBRATION AND VALIDATION OF THE MODEL

Several benchmarking tests were considered for the calibration of the model concerning input variables and boundary conditions for the new solver `interMixingWaveFoam`. After calibration of the model, validation has been conducted. For this purpose, a laboratory model was designed and built in order to observe the intermixing process of fluids as well as the interface between the mixture and the third fluid. The results of the experiments were later compared with the numerical simulations, and have been useful for the selection of the most appropriate turbulence model for the phenomenon as well as for the validation of the numerical model developed.

### 5.1. Design and construction of the laboratory model

The design of the laboratory model was done taking into account the simplicity of the test that should be well reproduced by the numerical model developed. The chosen design consisted on a container of 1.10 m length, 0.40 m height and 0.10 m width, divided into three sections with the same volume. The sections are connected by a 0.05 m height opening which is closed by waterproof gates at the initial-time  $t=0$  s. Each section is partially filled with either saltwater or freshwater and since the upper part of the container is opened to the atmosphere, both fluids are in contact with air. At the time zero, the gates are removed and the liquid fluids start their mixing due to the density gradient and the pressure gradient caused by the different water levels in each compartment. A schematic representation of the experimental model is shown in Figure 17.

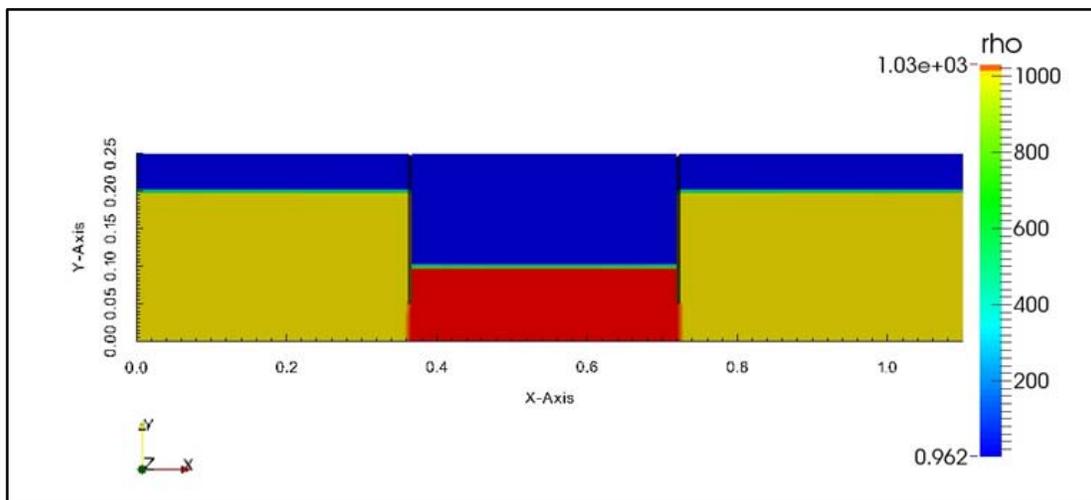


Figure 17. Schematic representation of the laboratory model. Saltwater is represented in red, freshwater in yellow and air in blue color. Axes X and Y are in meters and the density ( $\rho$ ) in  $kg/m^3$

The experimental model was made with acrylic sheets of 0.006 m width, while the gates separating each of the sections consisted on two metal plates, which can be opened simultaneously (Figure 18 and Figure 19).

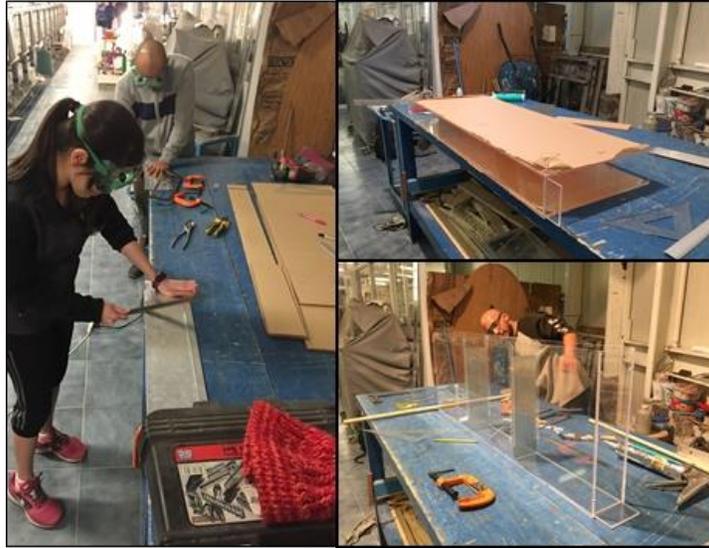


Figure 18. Construction of the acrylic laboratory model.

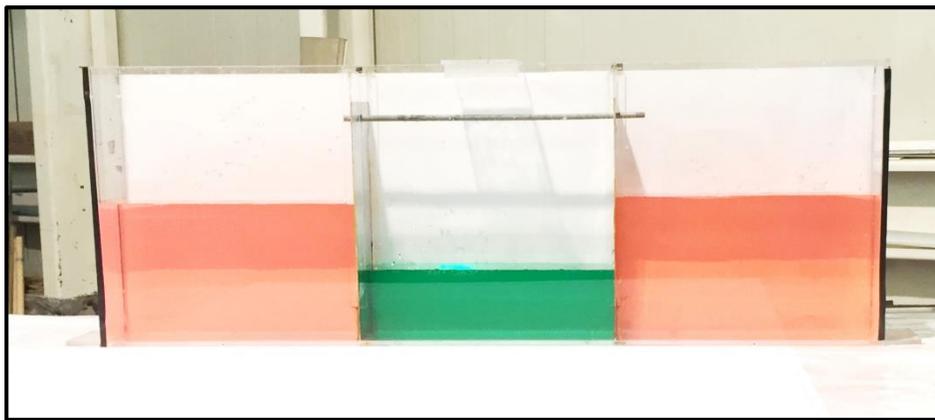


Figure 19. Picture of the acrylic laboratory model that was used to test the solver.

A high speed camera was installed to record the behavior of the fluids motion and of the mixture between phases (Figure 20). The camera used was a HiSpec1 model that records at up 506 fps with 1280 x 1024 px full resolution and up to 112,000 fps at reduced resolution; it has a compact size of 63mm H x 63mm W x 65mm D and 0.28 kg. The software-based ImageBLITZ® Auto Trigger was linked to the camera in order to set its configuration and to pre-visualize the images captured. The configuration of the benchmarking tests was set to 500 frames per second and the recording time to 12.2 seconds. The camera records black and white images, therefore an appropriate illumination was required to visualize the contrast between phases and to avoid shadows that could affect the image processing (Figure 20).



Figure 20. A high speed camera was installed, as well as the necessary illumination and a computer to save all the data recorded by it. The camera is circled in red.

At the beginning of the test, the metal gates were sealed with wax to avoid leakage. Then, each compartment was filled with saltwater and freshwater, respectively. Afterwards, the camera is started and the metal gates are simultaneously opened. Once the experiment is finished, the water is taken out of the container and the process can be repeated (Figure 21).



Figure 21. a) The gates were closed and sealed with wax; b) the center compartment was filled with dyed saltwater and the 2 remaining compartments with freshwater; and c) once the experiment was recorded, the water was taken out of the container.

30 tests were performed and the configurations considered are described in Table 1. The first 26 experiments were done with the following setup: a) the central section of the container was partially filled with saltwater with a density  $\rho_{sw} \approx 1030 \text{ kg/m}^3$  and to a level of 0.10 m; b) the two remaining sections were filled with freshwater with a density of  $\rho_w \approx 1000 \text{ kg/m}^3$  and a level of 0.20 m. The density of the water was

measured with a hydrometer 151H (resolution of  $1.0 \text{ kg/m}^3$ ). Also, the saltwater was colored with organic vegetable dye in order to contrast the two phases without modifying its density (Figure 22).

Table 1. Test program for the experimental tests. Saltwater is represented in red, freshwater in yellow and air in blue color.

Test program					
Case	Number	Name	Description	Observations	Initial setup scheme (saltwater in red, freshwater in yellow and air in blue)
1	1 to 26	Standard validation tests	Center compartment filled with colored saltwater ( $\rho=1030 \text{ kg/m}^3$ ) at a height of 0.10 m; the two remaining compartments filled with freshwater ( $\rho=1000 \text{ kg/m}^3$ ) at a height on 0.20 m.	Only 18 tests were considered because of experimental errors.	
2	27	Low density gradient	Center compartment filled with colored saltwater ( $\rho=1010 \text{ kg/m}^3$ ) at a height of 0.10 m; the two remaining compartments filled with freshwater ( $\rho=1000 \text{ kg/m}^3$ ) at a height on 0.20 m.	Test successfully performed	
3	28	Medium density gradient	Center compartment filled with colored saltwater ( $\rho=1020 \text{ kg/m}^3$ ) at a height of 0.10 m; the two remaining compartments filled with freshwater ( $\rho=1000 \text{ kg/m}^3$ ) at a height on 0.20 m.	Test successfully performed	
4	29	Inverted phases	Center compartment filled with colored freshwater ( $\rho=1000 \text{ kg/m}^3$ ) at a height of 0.10 m; the two remaining compartments filled with saltwater ( $\rho=1030 \text{ kg/m}^3$ ) at a height on 0.20 m.	Test successfully performed	
5	30	Same levels	Center compartment filled with colored saltwater ( $\rho=1030 \text{ kg/m}^3$ ) at a height of 0.20 m; the two remaining compartments filled with freshwater ( $\rho=1000 \text{ kg/m}^3$ ) at a height on 0.20 m.	This test was repeated twice because of experimental errors.	

The Table 1 shows 5 types of experiments. The first type of experiment is repeated 26 times and the remaining five are just performed once (Figure 22). The purpose was to perform the validation of the model with the first configuration and average the fluid flow behavior of several identical experiments and integrate one single averaged experimental result. This way, the random component of each experiment is minimized. The remaining configurations were performed and simulated in order to evaluate the accuracy of the model under certain conditions once the numerical model is validated.

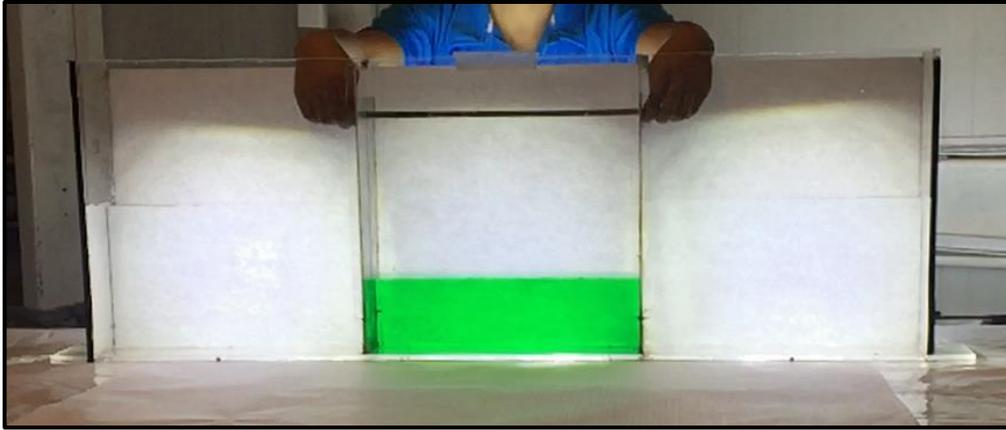


Figure 22. Configuration for the first 26 experiments that were recorded with a high speed camera.

The results of the high speed camera records are exemplarily shown in Figure 23., for a configuration similar to the first 26 experiments. As noticed, black and white images are obtained, but contrast between saltwater and freshwater is well captured as well as the free surface interface.

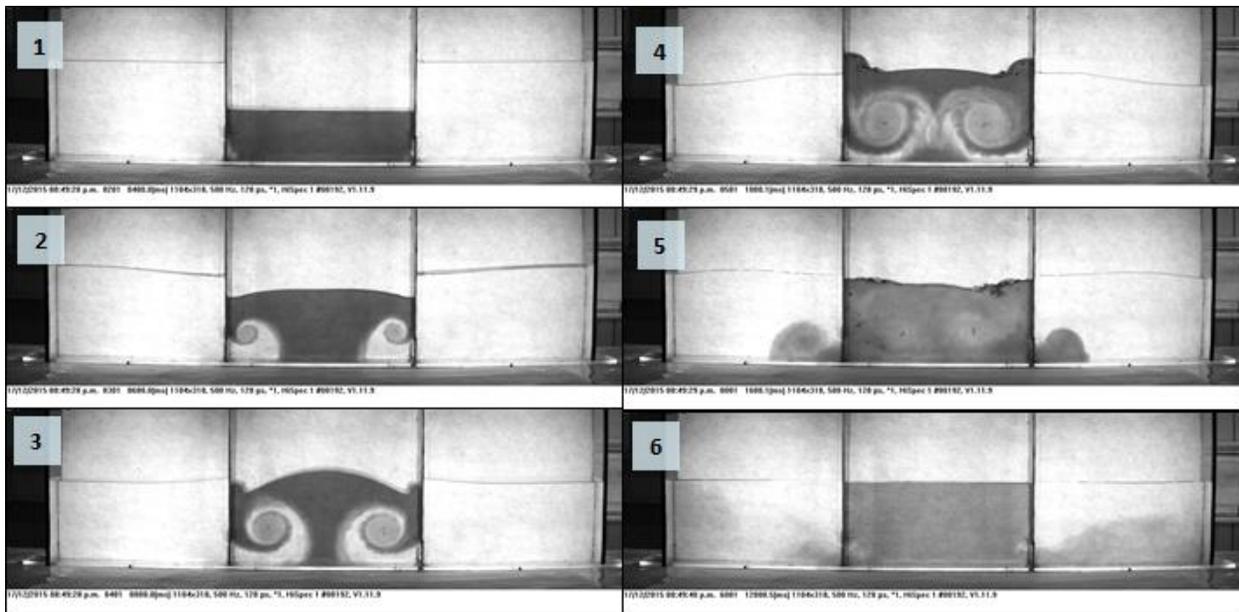


Figure 23. Frames captured by the high speed camera of the 8th test (out of 26) for the times  $t_1 = 0 s$ ,  $t_2 = 0.2 s$ ,  $t_3 = 0.4 s$ ,  $t_4 = 0.6 s$ ,  $t_5 = 1.2 s$  and  $t_6 = 11.6 s$ .

## 5.2. Numerical setup

The experimental cases were reproduced with the developed model `interMixingWaveFoam` in order to evaluate its accuracy against real conditions. The mesh, the boundary conditions and control of the simulation are the same for the 5 tests types described in Table 1. Some physical fluid properties and the initial field, change for each case simulated. It is important to mention that only the first configuration in

Table 1 (standard validation test) was used to validate the numerical model, thus, its simulation was done with three different turbulence approaches, as mentioned in section 4.2.2. The remaining configurations were simulated with one turbulence approach in order to evaluate the accuracy of the model under certain conditions.

It was observed that the most significant fluid motion and mixing occur during the first 10 seconds, and thus, the simulation time for each case was set to 12 seconds. Furthermore, the maximal time that could be recorded with the high speed camera was 12 seconds. Increasing the simulation time beyond this limit would increase the computational costs and the information cannot also be compared with any experimental data.

### 5.2.1. Mesh and boundary conditions

The domain is 1.10 m length (X direction), 0.25 m height (Y direction) and 0.10 m width (Z direction), just like the acrylic container dimensions except for the height (Figure 24). The height in the simulation is smaller than the acrylic container because it was observed that the liquid doesn't ascend beyond this limit. The use of the 0.4 m height would result in an increase in numerical cells and thus, in computational cost.

The mesh blocks and cell divisions per each side were established, for this case: 120 divisions in X-direction, 35 divisions in Y-direction and 8 in Z-direction (Table 2.). The number for divisions on each side defines the cell's shape and size. The cell shape should be approximately a square in order to maintain the aspect ratio close to 1 and for decreasing numerical diffusion problems. The number of cells and the mesh resolution should be enough to represent accurately the phenomenon, maintain a low courant number, achieve numerical stability and minimize the error, without considerably increasing the computational costs.

Table 2. Domain and mesh for the simulations of the 6 experimental cases performed.

	Domain and mesh		Gravitational acceleration
Direction	Dimension (m)	Cell number	a (m/s <sup>2</sup> )
x	1.1	120	0
y	0.25	35	-9.81
z	0.1	0	0

Because it is not a dynamic mesh but a static one, the gates opening is not represented in the simulation. Instead, the mesh includes boundaries to represent the gates already opened at a height of 0.05 m (Figure 24). In order to decrease the error that this produces, the gates were simultaneously opened as fast as possible during the experimental tests.

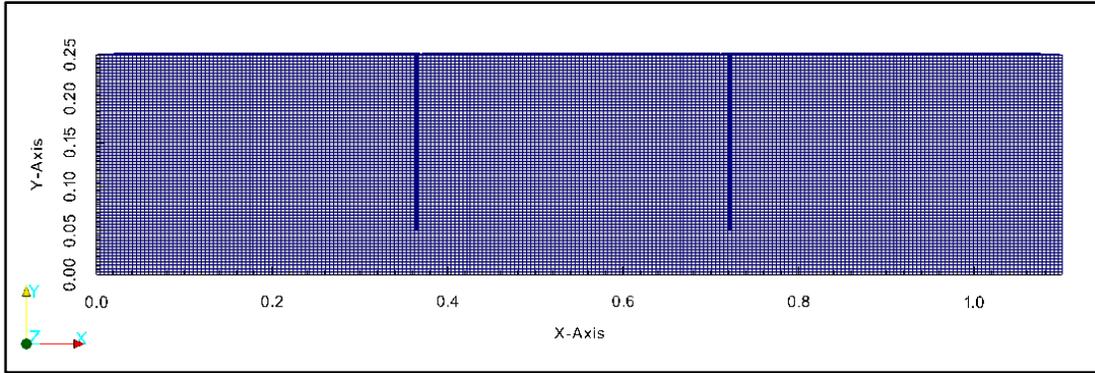


Figure 24. Domain and mesh of the numerical simulation of the experimental tests. The numerical mesh includes two boundaries that represent the gates opened at a height of 0.05 m.

The boundary conditions for this case are defined as follows. The patches that define the bottom, the side walls of the container and the gates have a `wall` boundary condition, all of them characterized by a patch that coincides with a solid wall. The boundary in the top of the container has a generic `patch` boundary condition, which will be defined later in the field's boundary conditions in order to allow inflow and outflow (patch free to the atmosphere).

### 5.2.2. Initial field, physical and fluid properties

The original field is the initial configuration of the phases in the domain. For the standard validation case, the initial field consists on a block of saltwater (alpha 2) that fills the central compartment of the domain at a height of 0.10 m. The two remaining compartments are filled by two blocks of freshwater (alpha 3) at a height of 0.20 m. The rest of the domain is filled by air (alpha 1) and is set as a default field value (Figure 25). Additionally, the velocity field is initially set to 0 m/s.

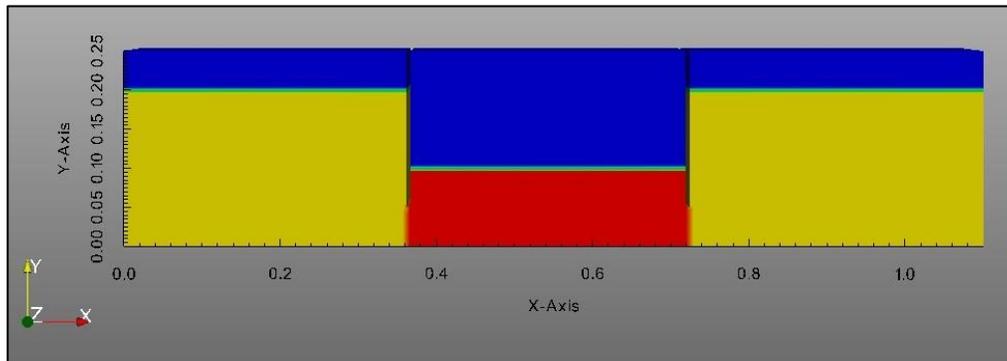


Figure 25. Initial field of the standard validation test case. Saltwater block is represented in red and freshwater blocks are represented in yellow. The rest of the domain is filled by air (blue).

The physical and fluid properties that need to be defined are: data of environmental properties such as the value and direction of the gravitational acceleration, transport properties, turbulence properties and

wave properties. The transport properties are the properties of each fluid phase, that is, kinematic viscosity, density, surface tension (Newtonian fluid essential properties) and diffusivity between miscible phases. Finally, the turbulence properties define whether the solution includes or not a turbulence model and which turbulence model will be used. Each required value is given in units of the International System.

The only environmental property required is the gravitational acceleration. For this case, the gravitational acceleration considered is  $9.81 \text{ m/s}^2$  and its direction is negative with regard to the Y direction.

The transport properties such as kinematic viscosity ( $\nu$ ), density ( $\rho$ ), surface tension ( $\sigma$ ) are defined for each fluid phase (air, saltwater and freshwater). The kinematic viscosity, air density, freshwater density and surface tension are considered constant for every type of experimental configuration described in Table 1. The last property defined is the molecular diffusivity ( $D$ ), which is another temperature-dependent value but will be considered as constant as the model does not include a temperature field. The value considered is taken from Holz et al., (2000) for a temperature of  $15 \text{ }^\circ\text{C}$ . Table 3 lists the values considered for the mentioned properties.

Table 3. Values of kinematic viscosity, air density, freshwater density, surface tension and molecular diffusion coefficient considered for the simulations of the laboratory experiments.

Phase property	Freshwater	Saltwater	Air
Density ( $\text{kg/m}^3$ )	1000	Variable	1
Kinematic viscosity ( $\text{m}^2/\text{s}$ )	$1 \times 10^{-6}$	$1 \times 10^{-6}$	$1.48 \times 10^{-5}$
Surface tension ( $\text{N/m}$ )	0.7	0.7	
Molecular diffusion coefficient ( $\text{m}^2/\text{s}$ )	$1.76 \times 10^{-9}$		

On the other hand, density values for saltwater was different in cases 2 and 3 (see Table 1). The case 2 (low density gradient case) has a density of  $1010 \text{ kg/m}^3$  for the saltwater phase and the case 3 (medium density gradient case) of  $1020 \text{ kg/m}^3$ . The remaining cases consider a density of  $1030 \text{ kg/m}^3$  for the saltwater phase.

The turbulence properties are also changeable for each case. The standard validation test was set with three different turbulence approaches: i) LES model, ii) RAS model and iii) zero-equation model (i.e., dissipation as a function of the velocity field). After running the three simulations with different turbulence approaches for the same case, it was qualitatively and quantitatively compared with the experimental results as described in the next section. The best turbulence model approximation according to the qualitative analysis was chosen to run the remainder cases described in Table 1.

### 5.2.3. Field boundary and initial conditions

This directory contains one file per variable field considered in the case. The fields whose boundary and initial conditions need to be defined are: `alpha1` (air), `alpha2` (saltwater), `alpha3` (freshwater), `p_rgh` (dynamic pressure) and `U` (velocity). Each field needs an individual set of boundary and initial conditions that must be defined separately.

For the scalar fields `alpha1`, `alpha2` and `alpha3` the patches that represent the walls of the container and the gates have a `slip` boundary condition. This boundary condition is a combination of the `zeroGradient` (which means that the gradient of the field is normal at that boundary) and `fixedValue` boundary conditions: `zeroGradient` is applied if  $\phi$  is a scalar, while if  $\phi$  is a vector, the normal component is `fixedValue` zero and tangential components are `zeroGradient`.

In the scalar field `alpha1` the patch `atmosphere` is in contact with the air phase (`alpha1`), so it is given uniform values of 1.0. This patch will have values of 0.0 in `alpha2` and `alpha3` because salt- and freshwater are not introduced in this patch.

The boundary conditions and initialization values must be also defined for the pressure and the velocity field. The `p_rgh` input dictionary contains the information for the dynamic pressure field, which is defined as:

$$p\_rgh = p - (rho * gh)$$

with  $p$  as the total pressure and  $rho * gh$  representing the hydrostatic pressure.

For the pressure field, the container walls patches have a `zeroGradient` boundary condition, the gates a `slip` boundary condition and the patch `atmosphere` has a `totalPressure` boundary condition, which fixes the total pressure  $p$  when the velocity  $U$  changes.

Finally the velocity field  $U$  is initialized: the patches for the container walls are set with a `fixedValue` zero and the gates with `slip` boundary condition. Also, the `atmosphere` patch has a `pressureInletOutletVelocity` boundary condition, which is used for patches adjacent with the atmosphere. Therefore, this patch allows both outflow and inflow according to the internal flow, for which a combination of boundary conditions for pressure and velocity is used to perform this task while maintaining stability. The `pressureInletOutletVelocity` boundary condition applies `zeroGradient` on all components, except where there is inflow, in which case a `fixedValue` condition (specific value of  $\phi$ ) is applied to the tangential component.

For the case where a RAS model was used, a two equation turbulence approach called k-epsilon was implemented. Thus four extra required fields are needed: i) a field  $k$  or turbulent kinetic energy which determines the energy of the turbulence, ii) a field  $\epsilon$  or turbulent dissipation which determines the rate of dissipation of the turbulent kinetic energy, iii) a field  $\nu_t$  or turbulent eddy viscosity, and iv) a field  $\nu_{tilda}$  or viscosity for the “Spalart Allmaras turbulence model” (the reader is referred to the source code of the official OpenFOAM® version for further information).

For the case where a LES model is used to solve turbulence, 3 additional fields are added: one for the turbulent kinetic energy and two for the “Spalart Allmaras turbulence model”.

Both RAS and LES models require specific boundary and initial conditions for each annexed field. Further details on how to set a RAS or a LES model with OpenFOAM® can be found in the OpenFOAM® User Guide and in the tutorials included with the official OpenFOAM® package (both available at <http://www.openfoam.com/>).

#### 5.2.4. Control and data extraction

As mentioned before, the simulation time was set to 12 s in order to obtain enough information for the qualitative and quantitative comparison with the experimental data. The results were written every 0.05 s to visualize a smooth transition from one time step to another. The Courant number was also defined considering the expression:

$$Co = \frac{\delta t |U|}{\delta x}$$

Where  $\delta t$  is the time step, which is 0.01 s for this case; and  $\delta x$  is the cell size in the direction of the velocity  $U$ . The Courant number is a dimensionless parameter that indicates the maximal time step allowed to achieve temporal accuracy and convergence while solving the partial differential equations system numerically. For this case, a Courant number lower than 1.0 is required, thus the maximal Courant number is set to 0.25, and the maximal time step is set to 0.01 s, so convergence within the modeling of the phenomenon can be achieved with a suitable time stepping.

Data of velocity vectors and pressure for the three phases (alpha 1, 2 and 3) was extracted from numerical probes set across the numerical domain. The points were placed every centimeter in the X (horizontal) and Y direction (vertical) and in the middle of the Z direction (width). That means 110 points in the X direction, 40 in the Y direction and 1 in the Z direction.

### 5.3. Calibration and validation of the numerical model

This section shows the analysis of results obtained from the validation of the model through laboratory tests. A qualitative comparison between the experimental results and the simulation results is firstly presented. Afterwards, a quantitative comparison is done after an image processing technic of both experimental and simulation results.

#### 5.3.1. Qualitative analysis

The standard validation test case was modeled with OpenFOAM® and solved with `interMixingWaveFoam`. Three cases were analyzed in terms of the turbulence modelling: i) zero-equation model; ii) Reynolds Average Simulation (RAS) turbulence model with a two equation approach (i.e. k- $\epsilon$  model), and iii) Large Eddy Simulation (LES) model (one-equation model). When solving the same case configuration with the three different turbulence approaches, it is observed that eddies development and the mixing between liquid phases is different. Thus, the validation and selection of the best turbulence model approach is significant for the estimation of accurate results.

It is important to notice that the numerical simulations do not reproduce the movement of the gates. Instead, both gates are completely opened at a time  $t_0 = 0$  s, which allows the flux between compartments instantly. The gates opening was done as fast as possible during the experiments to decrease the error of not considering this condition in the numerical simulations. The aperture time for each experimental test was measured and averaged as shown in Table 4.

Table 4. Measured time of the opening of the gates for each experiment and its average. The start time is when the gates are about to open and the end time when both gates are completely opened. Data extracted from the images recorded by the high speed camera.

Number	Test number	Start (ms)	End (ms)	Time (ms)
1	P3	716	868	152
2	P6	448	504	56
3	P8	354	400	46
4	P9	336	380	44
5	P10	68	102	34
6	P11	338	376	38
7	P13	162	196	34
8	P14	286	332	46
9	P16	538	572	34
10	P17	310	358	48
11	P18	324	374	50
12	P19	454	500	46
13	P20	370	406	36
14	P21	542	594	52
15	P22	222	288	66
16	P23	444	480	36
17	P24	634	672	38
18	P25	482	526	44
			Average (ms)	50.00

The average time of gates opening is 50 ms, short time compared to the total time of the experiment (12 s), representing only 0.42 % of the total time.

To match time zero of the experiment with time zero of the simulation, the frames that correspond to the gates aperture were not considered, thus, time zero of the experiments is assumed when both gates are completely opened.

The first 25 experiments can be analyzed in two parts by considering the flow behavior and the dominant process within the experiment:

- i) The first part is mostly dominated by the pressure gradient between the sections of the tank, each filled with freshwater or saltwater. The first part of the experiment is considered from

the opening of the gates (time zero) to approximately 0.8 s. The development of large eddies and the abrupt fluctuation of the free surface is noticed. Large eddies of around 15 cm diameter are developed due to the entrance of freshwater from the lateral compartments into the central compartment of the container (with saltwater). The fluctuation of the free surface during this process corresponds to a bell-shaped Gauss curve as shown in Figure 26.

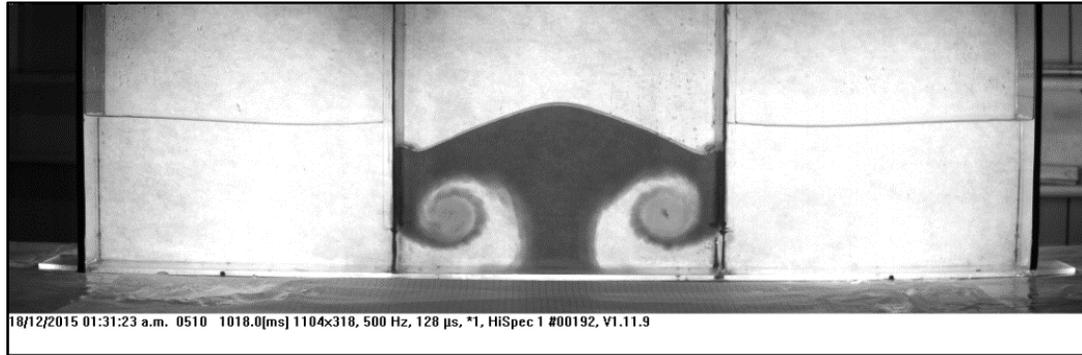


Figure 26. In the first part of the experiment large eddies (around 15 cm diameter) are developed. The free surface in the central compartment corresponds to a bell-shaped Gauss curve.

- ii) In the second part of the experiments, the mixing between phases and the fluid flow stabilization take place from time  $t=0.8$  s until the end of the experiment. The larger eddies observed in the first part of the experiment are dissolved into increasingly smaller eddies in the central compartment of the tank. In the lateral compartments, medium eddies of 10 cm are developed due to the entrance of saltwater but they are quickly dissolved and the mixing between phases is appreciated. The turbulence model chosen is highly relevant in the way the short scale eddies are developed, thus affecting the mixture between the two liquid phases (Figure 27).

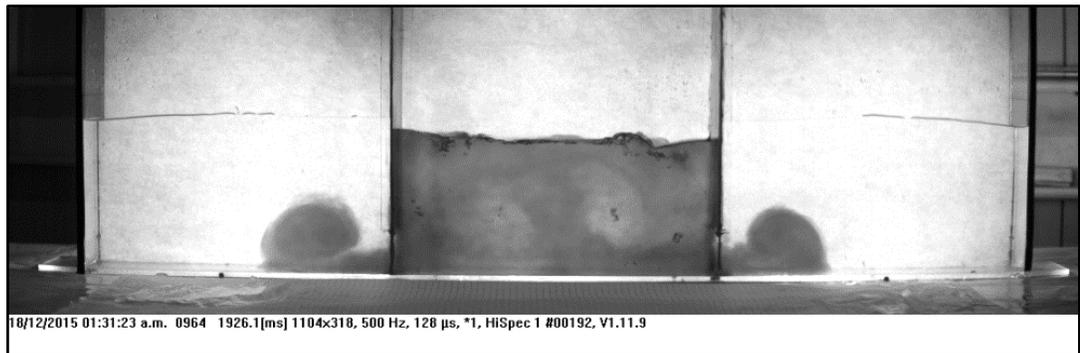


Figure 27. Stabilization of the free surface and the Mixing between the liquid phases take place during the second part of the experiment.

The performance of the different turbulence conditions used for the simulations is also described as follows (a comparative chart of these results is presented in Figure 28).

- **Zero-equation model.** The results of this simulation are closer to the experimental results. In the first part of the experiment, both free surface and eddies in the simulation are very similar to the experimental results. Also, the mixing between liquid phases in the second part of the experiment is similar to that shown by the numerical model, however, the energy dissipation due to turbulence seems to be underestimated, which is appreciated in the time stability of the fluid flow. The mixing between phases is rapid and chaotic in comparison with the use of RAS and LES models. Nevertheless, the rough structure of the interface between saltwater and freshwater is the most satisfactory above the other 2 simulations.
- **RAS model (k-epsilon).** The results given by considering the RAS model were generally less accurate than the other models. In the first part of the experiment, the development of eddies and the free surface behavior is satisfactory but less accurate than the simulation with no turbulence model. In the second part of the experiment, where the mixture between liquids is predominant, this turbulence approach is far away from the reality. The energy lost due to turbulence seems to be overestimated and it can be noticed a clear stratification between both liquids, instead of the chaotic mixture observed in the laboratory model and the other turbulence approaches.
- **LES model.** In the first part of the experiment, this simulation provides very different results from those observed in the laboratory tests. The large scale eddies are not completely developed and quickly vanish within the central section of the container. However, in the second part of the experiment, the phase-mixture is homogeneous and very similar to the experimental results. The mixture is more stable (less chaotic) than the simulation without considering turbulence energy dissipation, but not as *structured* or stratified as in the RAS simulation. The stabilization of the fluid motion seems to be consistent with the observations.

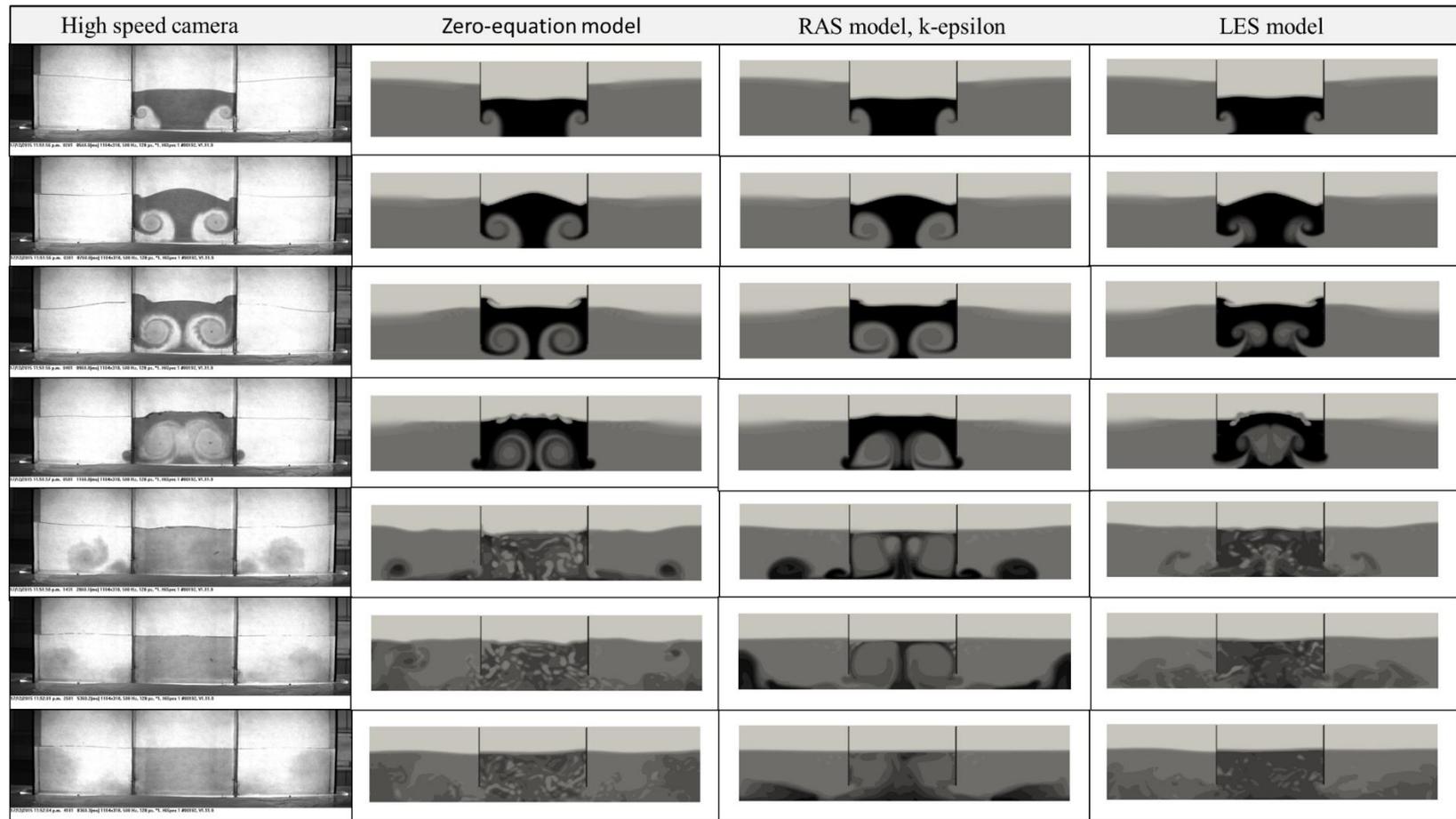


Figure 28. Comparative chart between the 3 different turbulence approaches vs. the experimental results recorded by the high speed camera. Each column contains the results of the experiment, the zero-equation, RAS, and LES turbulence approach respectively. Each row shows the results of the experiment vs. the simulations for the same time step ( $t_1 = 0.2 s, t_2 = 0.4 s, t_3 = 0.6 s, t_4 = 0.8 s, t_5 = 2.5 s, t_6 = 5.0 s$  and  $t_7 = 8 s$ ).

### 5.3.2. Quantitative analysis: image processing and error matrixes

A computational processing was made for a quantitative analysis of the model performance. For this purpose, black and white images were captured by a high speed camera of the 18 standard validation tests. Those results were studied along with the results of the numerical model in order to determine the accuracy of the model.

As stated before, the high speed camera recorded only 12 s of each experiment without considering the gap between the beginnings of the experiments (when the gates were opened) and the start of the recording, which was between 0.3 and 0.8 s. Furthermore, the most important fluid motion processes occurs within the first 4-6 seconds, the following time is mostly related to the stabilization of the system and no significant fluid motion is observed (also the contrast of colors does not allow the clear visualization of the movement and mixing within the flux). Taking this into account, and optimizing the information that needed to be processed only the first 5 seconds of the experiment are considered for the quantitative analysis. The following steps summarize the image processing carried out.

#### i. Selection of experiments.

Around 18 out of 25 experiments were selected. Just the tests that were correctly performed were considered, i.e. when the gates of the container were simultaneously opened, providing a symmetrical behavior of the fluid flow (Figure 29).

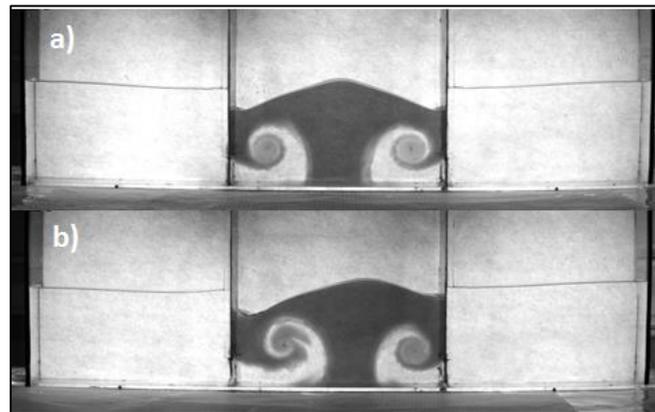


Figure 29. Contrast between an experiment that was correctly performed (a) and one in which the right gate was opened slightly before the left one (b).

#### ii. Image capturing and analysis.

The image captured by the camera contains borders and other elements surrounding the area of interest. With a rate of 500 fps during 5 seconds, each of the images taken was clipped in order to keep the area of interest. The original image has 1104x360 pixels and it was clipped into a 914x206 pixels.

#### iii. Generation of the density matrix.

Each of the 18 experiments was calibrated according to the time-zero frame and its color contrast before the mixing process. A pixel within the central section of the container with freshwater and given a density value of  $1000 \text{ kg/m}^3$ . Additionally, a pixel was selected but for the section with

saltwater and a density value of  $1030 \text{ kg/m}^3$  was given (Figure 30). The color scale was calibrated based on a linear interpolation between these values, thus leading to a density matrix per frame. The purpose of considering a pixel in the central section of each compartment instead of one in the periphery or an average, was to avoid *polluting* of the information with the shadows that were inevitable when illuminating the experiment (the central area of the container was priority than the side ends).

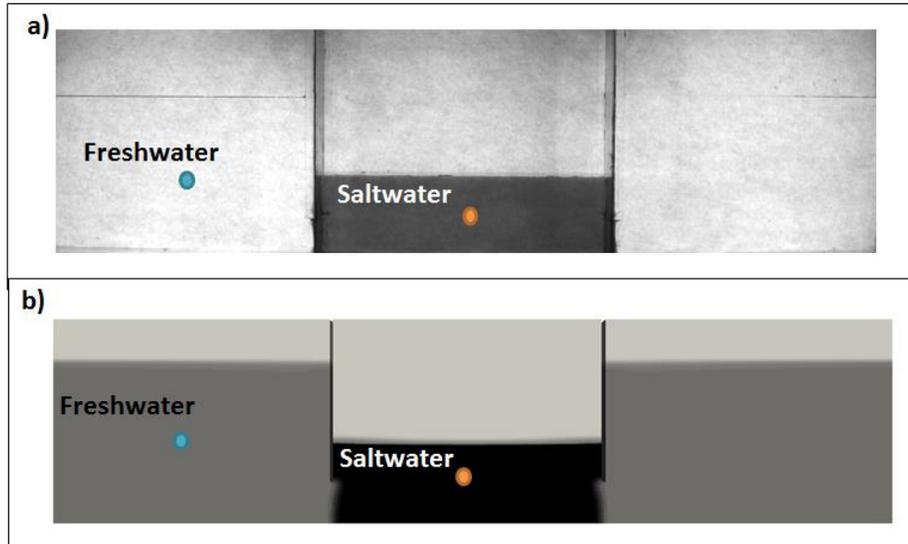


Figure 30. The density matrix for each time step in the experiments and in the simulation was generated from a calibration of the black and white tones at time zero. The darker the color the higher the density value, thus a linear interpolation between gray tones was done. A density value was given to each pixel of each frame image accordingly to this interpolation.

**iv. Frequency sampling match**

To match the time step between the experiments (500 fps) and the model (20 fps), a subsampling was required. This subsampling consisted on considering the density matrix of the experiment every 25 frames, instead of the 500 frames originally recorded. The decision of reducing the resolution of the time steps instead of increasing the resolution of the simulation data was taken by considering the amount of time required to process such data in both the simulation post-processing as well as in the data comparison through the time. The average time to process the data (20 fps) was of about 2 - 3 days.

**v. Interpolation between experiment pixels and snapshots pixels of the simulation:**

Another limitation to compare the experimental data with the simulation results was the density matrix size, which was determined with the number of pixels on each case. In order to obtain the same amount of points, an interpolation in the smallest matrix (extracted from the post processing) was required. The matrix of the numerical simulation is determined by the pixel number of the images extracted from the post-processing (Figure 30, b) and it was of  $832 \times 190$

pixels. Thus, this resolution was interpolated with the 914x206 pixels resolution of the captured images of the experiments.

**vi. Averaging of the density matrices.**

The purpose of repeating the experiment several times was to obtain a final average of the fluid motion, and to minimize the errors due to human factors and the random component of the water motion. The density matrices of the 18 experiments for each time step were averaged. The result of this process is an averaged set of density matrices generated with a frequency of 20 Hz and during 5 s per experiment.

**vii. Estimation of the relative error:**

The averaged density matrix  $\overline{\rho_{exp}}$  for each time step (0.05 s) was compared with the density matrix obtained with the simulation  $\rho_{sim}$  for each time step. The relative error for each time step was obtained by considering the absolute value of the difference between the density of the experiment and the density of the simulation in one pixel, divided by the density rank (30 kg/m<sup>3</sup>), which oscillates between 1000 and 1030 kg/m<sup>3</sup> (for the fluid phases). The final result is multiplied by 100 to be expressed as a percentage. By assuming the aforementioned error, the air phase is not considered within the analysis, since the mixture between phases is more relevant for the current work.

$$e = \left| \frac{\overline{\rho_{exp}} - \rho_{sim}}{30 \text{ kg/m}^3} \right| \times 100 \quad \text{Equation 17}$$

The relative error calculated for each of the 18 density matrices was then averaged to obtain the average error per time step and per turbulence model. To simplify the error visualization, it was graphed giving each pixel a color that represents the error estimated (Figure 31 and Figure 32.).

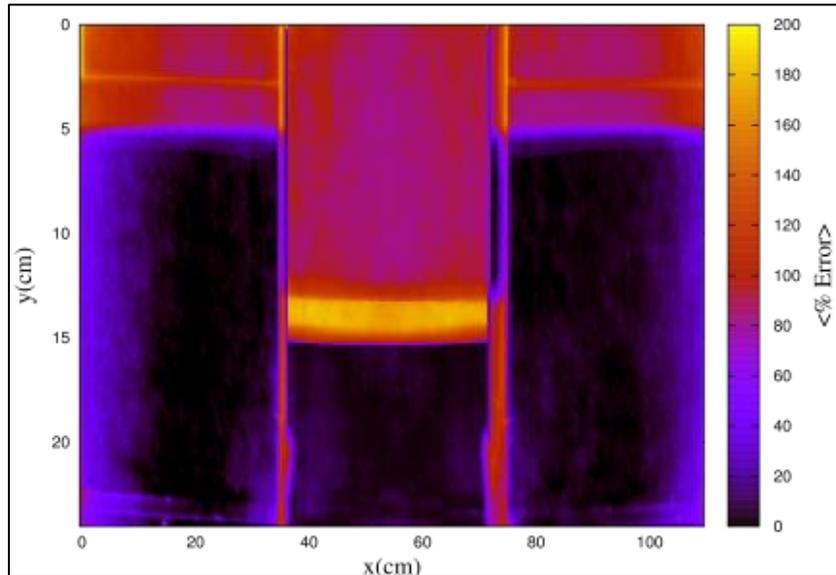


Figure 31. Error matrix for time zero ( $t_0 = 0 \text{ s}$ ). The black and blue areas are the pixels that represent the regions with the liquid phases. The yellow and red areas are outside the area of interest.

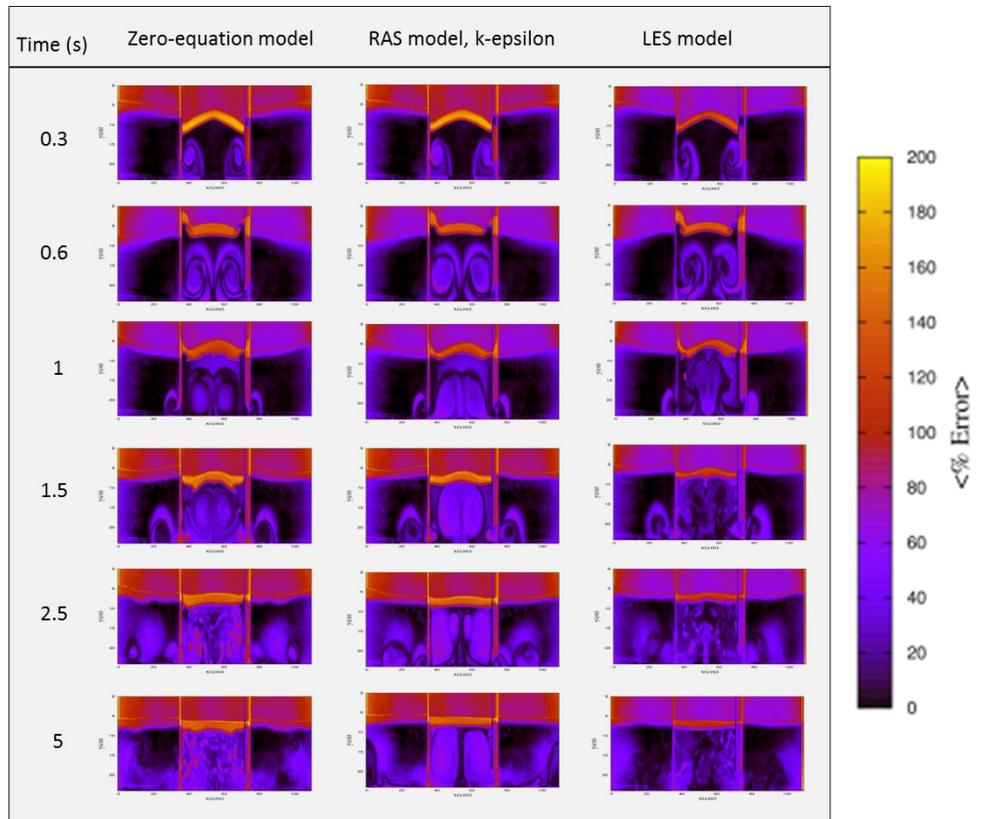


Figure 32. Error matrixes for different time steps and the 3 turbulence models tested.

As observed in Figure 31 the area filled by liquids shows an error close to zero (0 to 10 %), that is, an extensive black region with some stains of higher errors. As expected, this time step ( $t_0 = 0$  s) has a low error percentage but is not completely zero in all the liquid-filled area because of the errors that the shadows in the photographs introduce. Although those shadows were difficult to decrease and impossible to delete, the result is satisfactory in the central region of the container, where the most important fluid motions take place. The regions above the liquid-filled area exhibit a very large error percentage because they are filled by air which was not analyzed. Thus, even though that region will not be studied, the interface between air and liquids can be clearly observed.

As times goes by, the error increases particularly at those zones where eddies are developed and the mixing takes place, as shown in Figure 32. Even though the error increases, it is below 20 % (in average) in most of the region analyzed and there are still some regions where zero error was obtained (darker areas in the graphics are the ones with lower errors). In zero-equation model case, small regions with larger errors coincide with the introduction of bubbles of air shown in the numerical results. Those bubbles were present also in the experiments but they are not as big as in the numerical results, therefore, this could be attributed to the interface tracking in the numerical model as well as the mesh size.

At a first sight, it can be seen that the one of the LES model has darker regions than the other 3 models through time. Thus, it is expected to have lower average error with this model even though the qualitative analysis placed the zero-equation model as the best approach. However, the next step is to obtain an average error through time in order to observe its variation and determine which turbulence model provides results closer to those of the experiment.

**viii. Error variation with time.**

The average error percentage was graphed through time to observe its variation. In order to consider only the region of interest (area filled by the liquid phases), the free surface was tracked through each time step with the help of the density matrixes. The tracking gave a binary matrix, where the value of 1.0 is given to the area with liquid and 0.0 for the area filled with air. Then, the binary matrix was multiplied by the error matrix. Therefore, the area above the free surface was given values of zero and was not considered in the error averaging through time. The process of tracking the free surface and multiplying the given binary matrix was repeated for each time step and each turbulence approach. Also, the domain was divided in 3 parts according to the compartment division of the experimental model. In this way, it was possible to obtain an error average for each compartment. Thus, there is an error average for the left, central and right compartments individually. Graphs in Figure 33 show the average error through time for each turbulence model and each compartment.

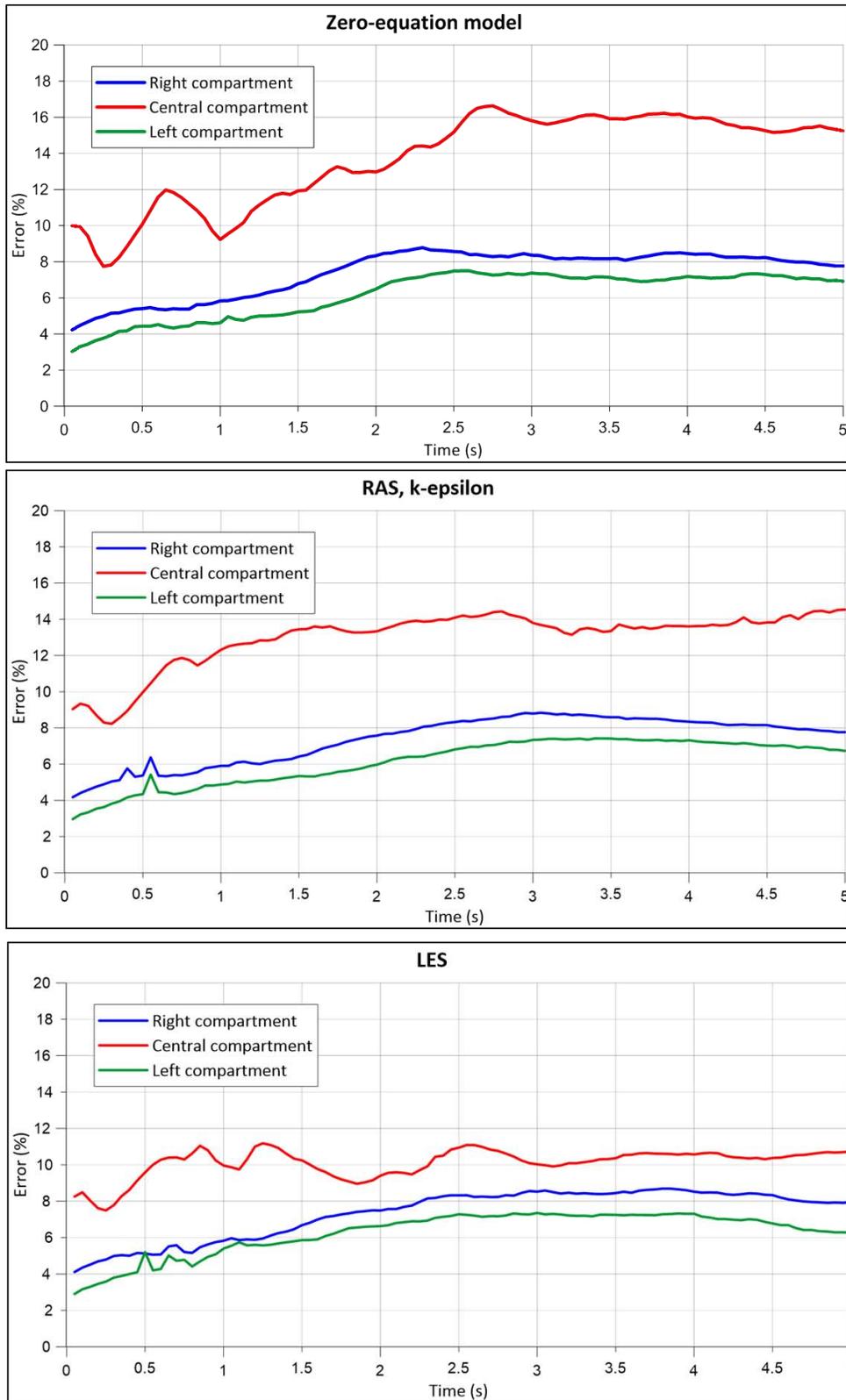


Figure 33. Error percentage vs. time of the 3 turbulence models respect to the experimental results. Graphic results vs. high speed camera photographs.

As shown in Figure 33 the tendency is to increase the error with the time during the first 3 seconds and it tends to stabilize after this. Although similar error estimates were expected for the left and right compartments, as it is a symmetrical experiment, there was a slight difference (1 %) between them that could be attributed to the experiment illumination on each sides, the synchronicity of the gates opening. Even though this difference, both sides follow a very similar tendency with a phase shift of 1 %, approximately.

For the zero-equation model case, it is noticed an error of 3 - 8.8 % for the lateral compartments and between 7.7 - 16.6 % for the central compartment. Averaging the 3 compartments, an average error of 9.1 % is obtained. For the *first part of the experiment* (before  $t=0.8$  s) and the *second part* (after  $t=0.8$  s), the central compartment contributes with an error of 9.93 % and 14.4%, respectively. Even though this turbulence approach was recognized as the best one in the qualitative analysis (at least for the first part of the experiment), in the quantitative results is the one with higher average error. As expected, the error for the second part of the experiment is the higher with respect of the other 2 turbulence approaches.

By considering a RAS model (k-epsilon), it is noticed an error of 3.0 - 8.8 % for the lateral compartments and between 8.2 - 14.5 % for the central compartment. Averaging the 3 compartments, an error of 8.8 % is obtained. For the *first and second parts of the experiment*, the central compartment contributes with an average error of 9.88 % and 13.6 %, respectively.

Finally, when using a LES model, errors of 2.9 - 8.7 % and 7.5 - 11.2 % for the lateral and central compartments are obtained. Averaging the 3 compartments, it is found an error of 7.9 %. For the *first and second parts of the experiment*, the central compartment contributes with an average error of 9.1 % and 10.3 %.

In contrast with the qualitative analysis, the calculated error of the LES model shows a different result than the observations alone. Thus, in terms of average error, the LES approach is the closest for representing the mixing process through time. It is worthy to mention that the higher error of the LES model occurs in the *first part of the experiment* if compared to the second part and also with the other turbulence models. Also, the tendency of the error curve for the central compartment remains almost constant after  $t = 0.8$  s.

### 5.3.3. Quantitative analysis: numerical probes

In order to verify the validity of the image analysis, one more validation routine was performed. The process consisted on comparing specific points locations of the images recorded with the high speed camera, against the same point locations (numerical probes) where data was extracted from the numerical simulation each centimeter (in the 3 directions). As described in the previous section, the recorded images of the experiment were processed in order to obtain a density matrix (one density value per pixel) for each time step at a 100 Hz frequency. Then, the density matrix was compared with the field's matrixes (obtained from the numerical probes) that represent the liquid phases.

Error matrixes, similar to the ones obtained with the previous image analysis, were obtained also for the numerical results. Those matrixes were averaged for the right, central and left compartment and graphed through time, for each turbulence model studied (Figure 34).

The results are consistent with the previously obtained in terms of: i) the central compartment contributes with the higher error (2 to 11 % higher than the lateral compartments); the lateral compartments have a similar tendency but the right compartment contributes with more error than the left compartment (except in a section of the graphic for the first case); the error tendency increases in the first seconds but it the behavior is stabilized mainly after  $t=0.8$  s.

In the first case (zero-equation model), it is noticed an error of 2.1 to 9.9 % for the lateral compartments and of 4.1 to 20.5 % for the central compartment. By averaging the 3 compartments a total error of 9.5 % is obtained. Average errors of 9.98 % and 16.9% are noticed for the first and *second parts* of the experiment.

In the second case with the RAS model, it was noticed an error of 2.1 to 9.9 % for the lateral compartments and between 4 to 13.6 % for the central compartment. Averaging the 3 compartments it gives a total error of 7.9 %. For the *first part of the experiment*, the central compartment contributes with an average of 9.88 % error. In the *second part* it has 9.62 % of error on average for the 3 compartments.

For the case with the LES model an error of 2.1 to 8.9 % for the lateral compartments is obtained. The error in the central compartment was of 4.1 to 21.5 %. Averaging the 3 compartments it gives a total error of 8.2 %. For the *first part of the experiment*, the central compartment contributes with an average of 10.28 % error. In the *second part* it has 12.6 % of error on average.

In general terms, the results of this validation approach are consistent with the ones obtained only by comparing photographs and images of the model post-processing, which was the objective of doing both ways of validation.

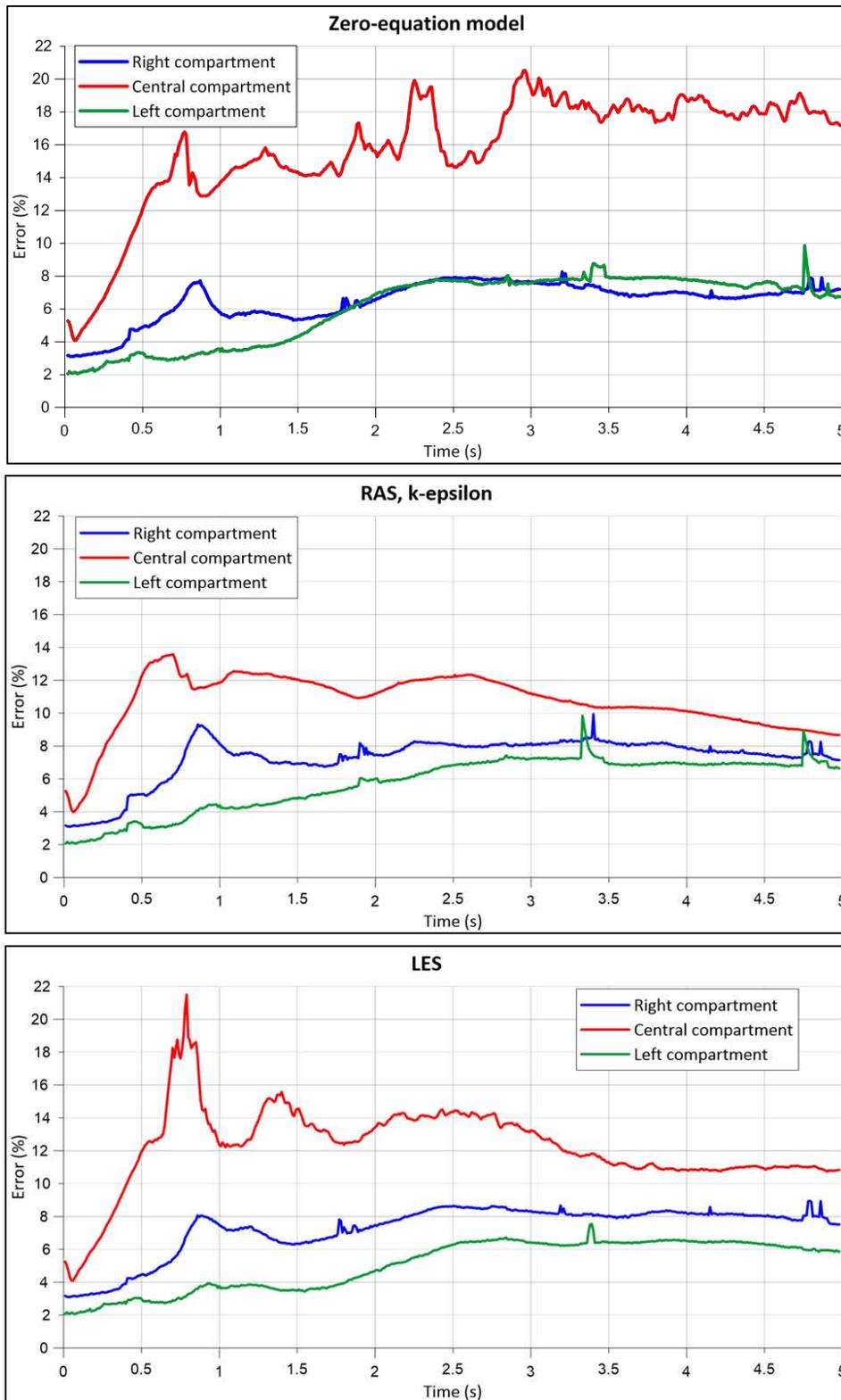


Figure 34. Error percentage vs. time of the 3 turbulence models respect of the experimental results. Numerical results vs. high speed camera photographs.

#### 5.4. Qualitative and quantitative description of the cases 26 to 30

After validating the model, it was applied to the tests 26 to 30 (descriptions in Table 1). Those tests are variations of the same experiment used to validate the model. Because the model is already validated and its implementation gives acceptable results compared with the experiments (average errors below 15 %), the averaging of multiple tests is not executed but only one single test per each case is considered to perform the qualitative and quantitative comparison with the numerical simulation.

In order to illustrate the accuracy of the model with one single experimental test case vs. the numerical results, case number 3 of Table 1 will be shown (Figure 35).

Case number 3 correspond to the configuration where central compartment filled with colored saltwater ( $\rho = 1020 \text{ kg/m}^3$ ) at a height of 0.10 m; the two remaining compartments filled with freshwater ( $\rho = 1000 \text{ kg/m}^3$ ) at a height on 0.20 m. The average error through time for each compartment and  $k - \varepsilon$  turbulence model is shown in

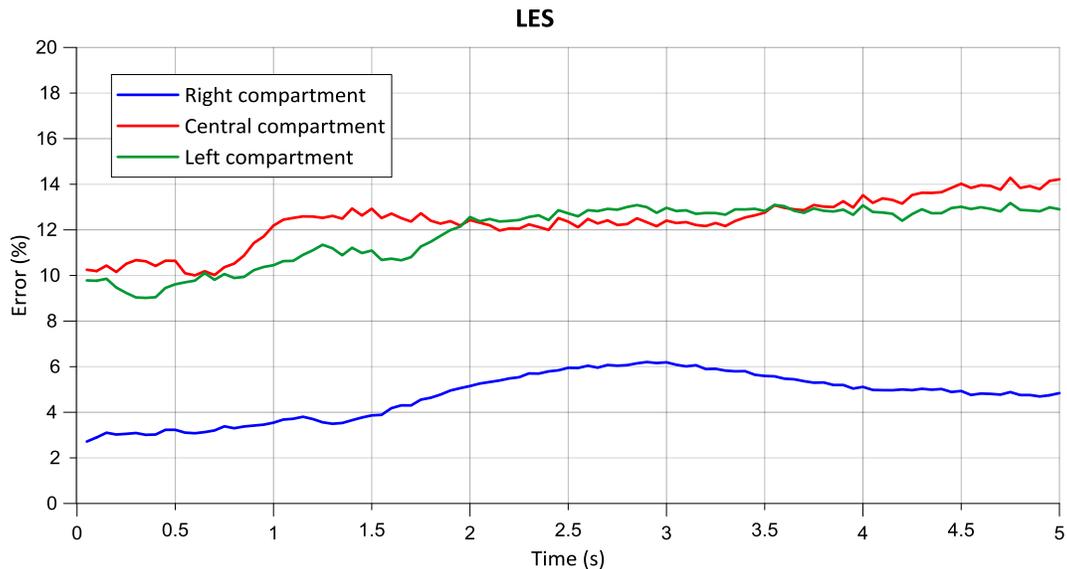


Figure 35. Error percentage vs. time of LES turbulence respect of the experimental results for the configuration of case number 3 of table 1. Graphic results vs. high speed camera photographs.

In this case, the left compartment contribute with significantly higher error than the right compartment (as it is a symmetrical configuration, the right and left compartment errors are expected to be very close) which could be a result of comparing only one single experiment instead of series of experiments averaged (as done in the validation case). Thus, a slight lack of coordination in the gates opening, could have affect this results. In general terms, the average error is less than 15 % for all the cases tested, which corroborates the capacity of the numerical model to recreate the reality.

# **APPLICATION TO A STUDY CASE**

## 6. APPLICATION TO A STUDY CASE

After validation of the numerical model, a study case was analyzed to exemplify the application of *interMixingWaveFoam* for a SGD simulation. Since the application of the model provides preliminary results, further studies must be conducted to provide more detailed information about the hydrodynamic performance of SGD.

### 6.1. Description of the study area

The study area, where the SGD is found and for which the *interMixingWaveFoam* model is implemented, is located at the Puerto Morelos coral reef lagoon (20° 52' 48" N, 86° 51' 34" W), in the western Caribbean Sea in the Yucatan Peninsula, Mexico. Several (~10 to 15) point source SGD can be found in the reef lagoon floor. One of those point source SGD is the *Pargos Spring* (Figure 36), which is named after the large number of snappers that congregate at this location (Parra et al., 2014).

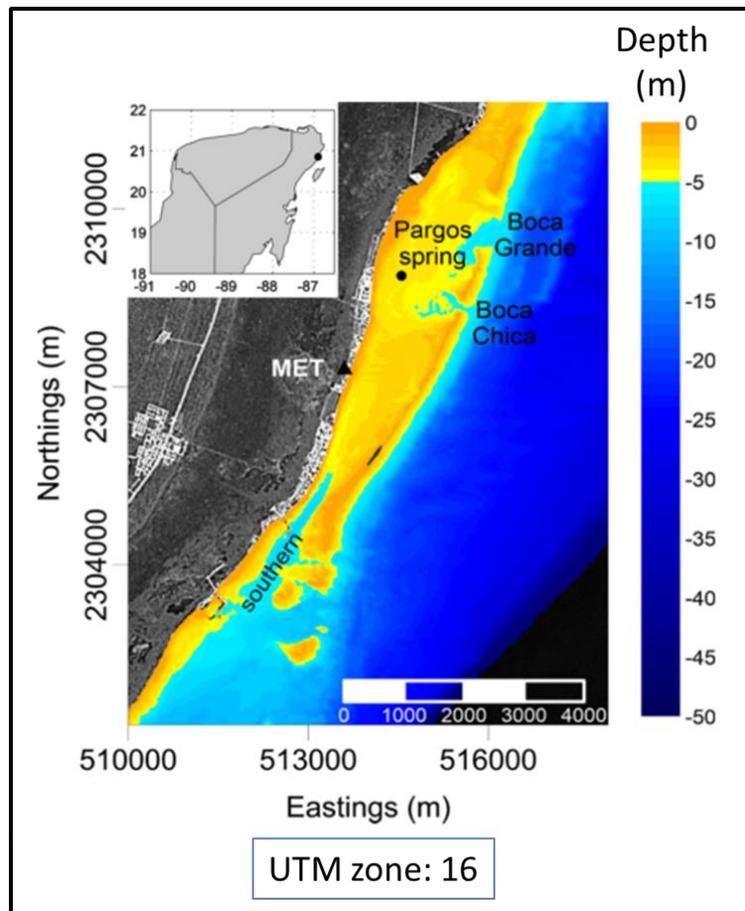


Figure 36. Location of the study area and bathymetric map of the Puerto Morelos coral reef Lagoon. The submarine spring *Pargos* is marked out with a black circle. Depth and scale in meters. Taken from Parra et al., (2014)

According to Parra et al., (2015) the characteristics of the lagoon and the Pargos spring are described as follows, respectively:

#### Lagoon

- Mean water depth of ~3.5 m and a maximum depth of 8 m.
- Affected by onshore wind waves, dominant semidiurnal micro-tides of maximum 0.4 m tidal range and long-term sea level oscillations of ~15 days and ~0.3 m (associated with oceanographic or atmospheric processes).
- Significant wave height within the lagoon of  $H_s=0.2-0.3$  m.

#### Pargos spring (Figure 37)

- Mean discharge of the outflow  $Q= \sim 0.4-0.5$  m<sup>3</sup>/s at a distance of 300 m away from the coastline
- The bathymetry exhibits an opening to the spring vent of ~2.5 m diameter.
- Two meters below the 2.5 m diameter opening there is the spring vent with ~1.2 m diameter.

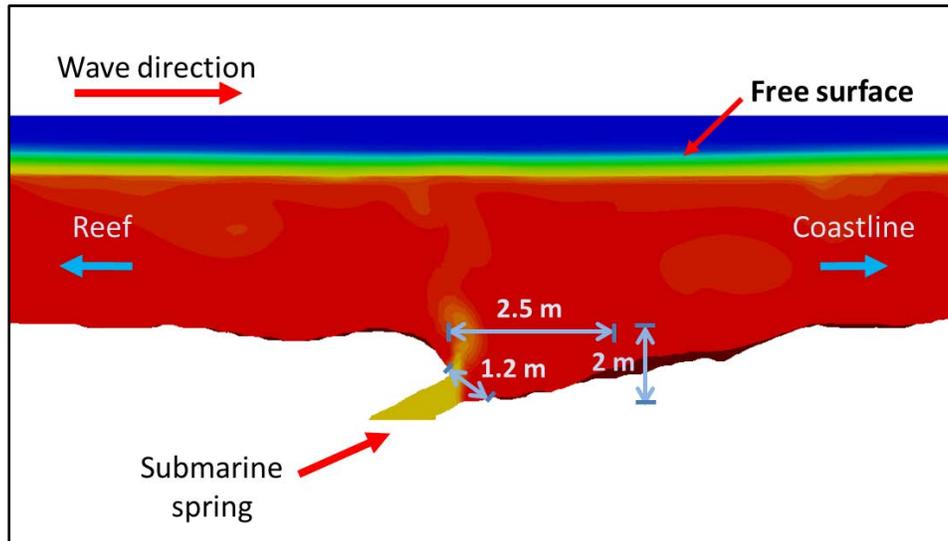


Figure 37. Sidecut of the Pargos spring bathymetry.

The detailed bathymetry of the study area around the *Pargos* spring was provided by the CINVESTAV Research Center in Mérida, Mexico. General data of the *Pargos* case setting and chosen parameters are summarized in Table 5 and Table 6.

Table 5. Characteristics of the domain, mesh, gravity, freshwater flow rate and phases properties for the *Pargos* case

		Domain and mesh		Gravitational acceleration	Submarine spring (1.2 m diameter)
Direction	Dimension (m)	Cell number	a (m/s <sup>2</sup> )	Velocity v (m/s)	
x	180	800	0	0.35	
y	1	5	0	0	
z	6	35	-9.81	0.35	
Phase properties					
Phase property		Freshwater	Saltwater	Air	
Density (kg/m <sup>3</sup> )		1000	1030	1	
Kinematic viscosity (m <sup>2</sup> /s)		1x(10) <sup>(-6)</sup>	1x(10) <sup>(-6)</sup>	1.48x(10) <sup>(-5)</sup>	
Surface tension (N/m)		0.7	0.7		
Molecular diffusion		1.26x(10) <sup>(-9)</sup>			

Table 6. Wave properties and simulation time for the *Pargos* case.

Wave properties	
Type	Regular wave train
Wave theory	Linear
Period T (s)	7
Wave height H (m)	0.3
Depth h (m)	2.1
Simulation time (s)	150

Although the simulation is three-dimensional, the attention is focused on the wave propagation direction; thus the domain is larger in this direction in comparison with its width. Waves are generated on the left side of the domain and absorbed on the right side (Figure 38). For this case, the mesh was generated from the bathymetry surface. An OpenFOAM® tool named *snappyHexMesh* “sculpt” the domain according to the bathymetry surface and creates a mesh refinement around it, as seen in Figure 38.

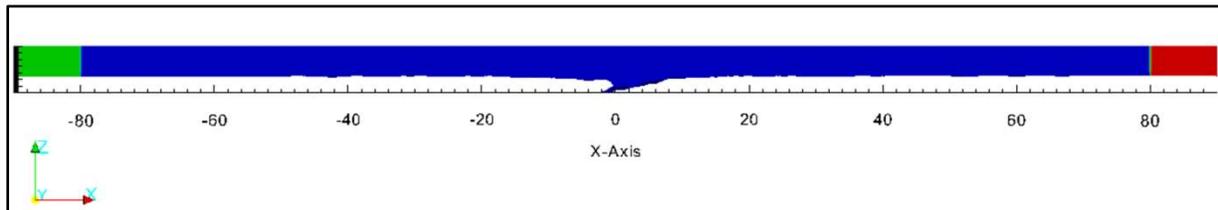


Figure 38. Complete 180 m length domain of the *Pargos* case. The wave generation zone is in green and the wave absorption zone in red. The bottom of the domain that represents the ocean floor and the submarine spring was “sculpted” according to the given bathymetry.

The wave height, wave period, water depth and flux rate were chosen according to the mean conditions in the study area. A regular wave train was tested although it could be implemented an irregular wave train that reproduces the conditions measured in situ.

## 6.2. Results

The zero-equation and LES models were tested and contrasted for the study case (Figure 39 and Figure 40). The results of the *Pargos* case for the two best turbulence models tested (according to the validation) are now described. In order to clearly visualize the area of interest in the simulations, the results showed in the following figures are from the central section of the 180 m domain modeled.

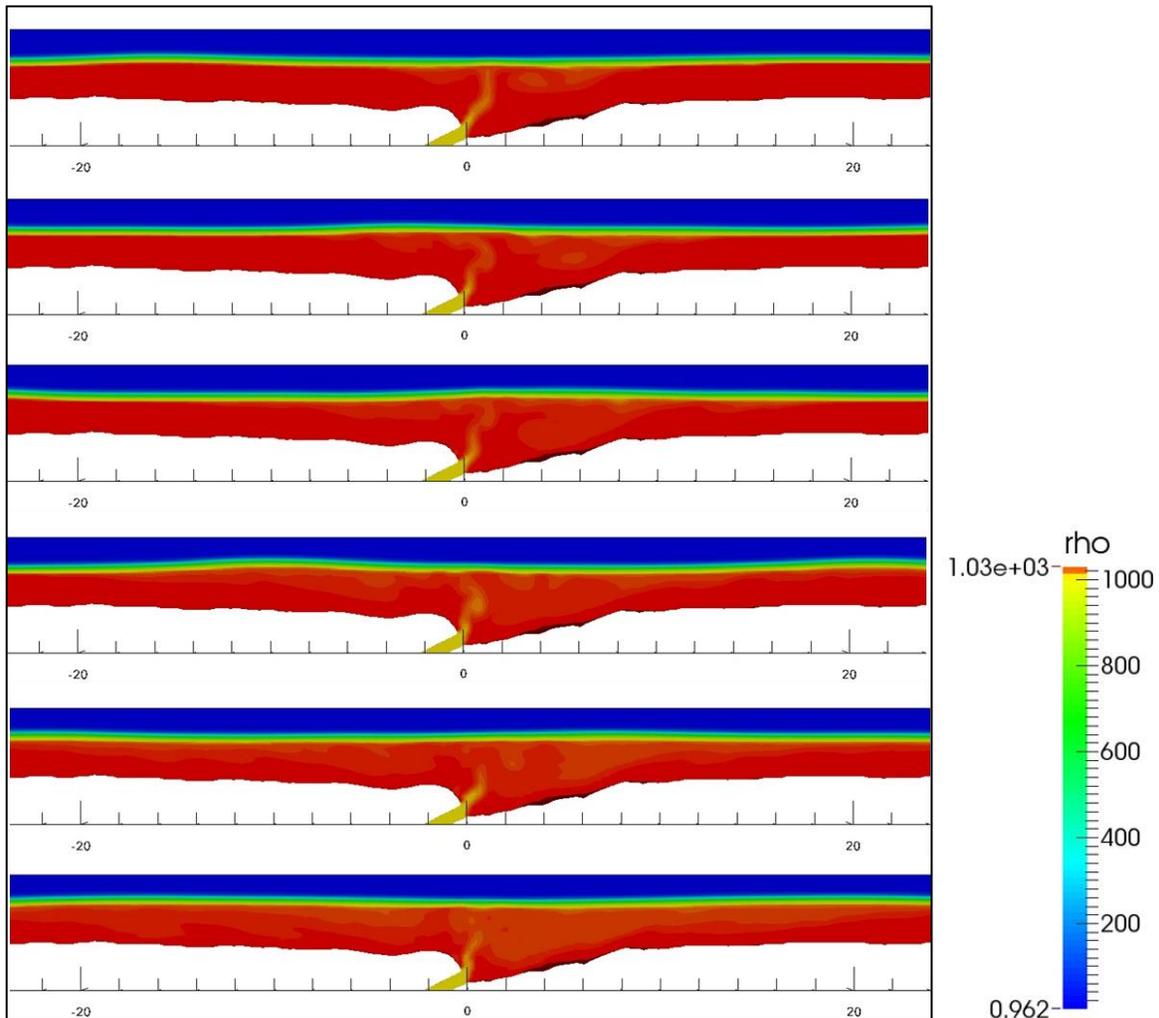


Figure 39. *Pargos* case modeled with *interMixingWaveFoam* and zero-equation model. Wave propagation from left to right. The snapshots correspond to time  $t_1 = 25\text{ s}$ ,  $t_2 = 35\text{ s}$ ,  $t_3 = 50\text{ s}$ ,  $t_4 = 75\text{ s}$ ,  $t_5 = 100\text{ s}$ , and  $t_6 = 150\text{ s}$  respectively.

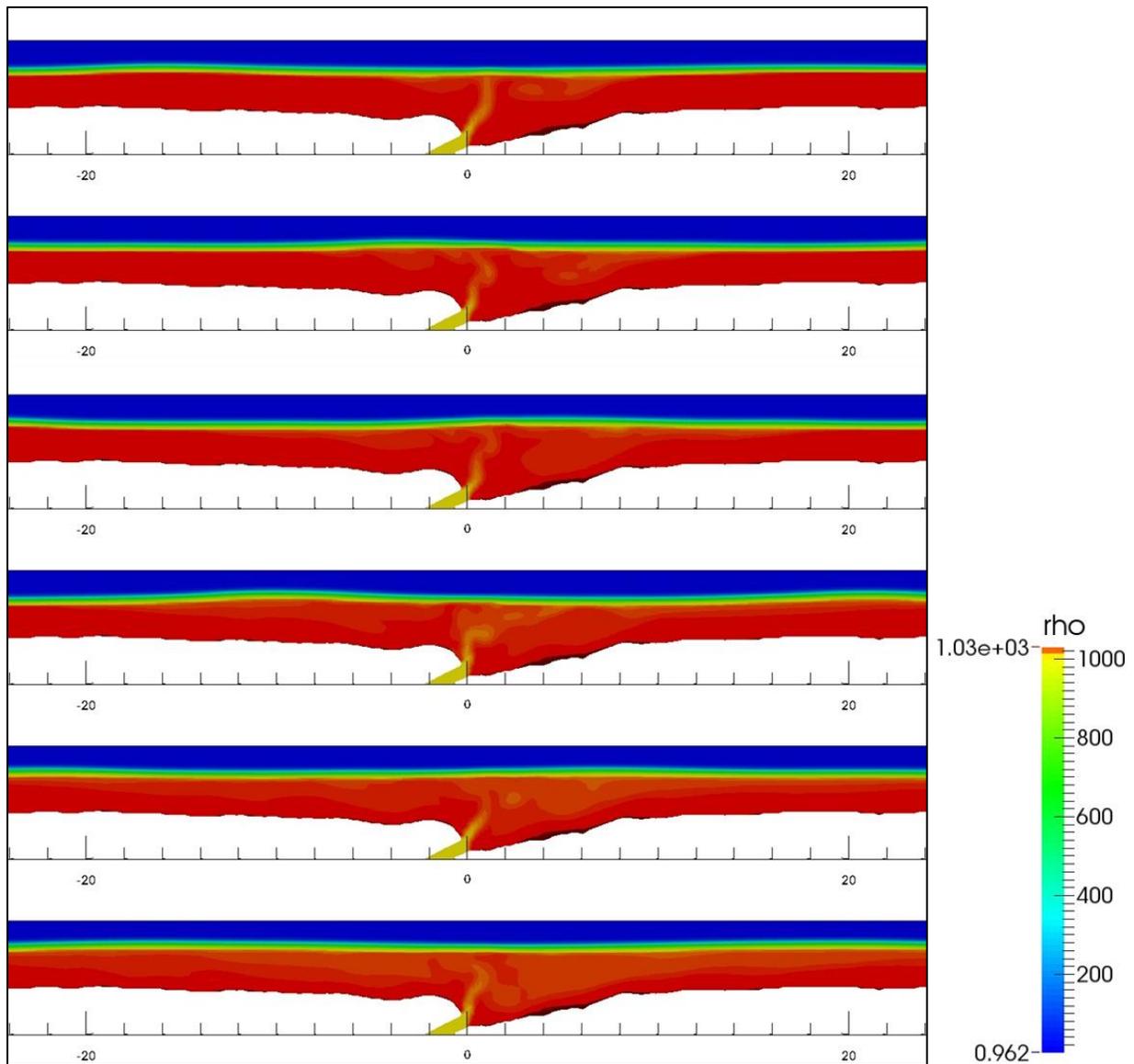


Figure 40. Pargos case modeled with interMixingWaveFoam and LES turbulence approach. Wave propagation from left to right. The snapshots correspond to time  $t_1 = 25 s, t_2 = 35 s, t_3 = 50 s, t_4 = 75 s, t_5 = 100 s,$  and  $t_6 = 150 s$  respectively.

In general, the two turbulence approaches show the capability of the solver to represent a free-surface system with a wave train affecting a saltwater medium and the inflow of freshwater. With a qualitative comparison between the two turbulence approaches, it can be seen some slight differences in the way that eddies are developed and in the mixing between liquid phases. As observed in the validation tests, the zero-equation model shows that the mixture between saltwater and freshwater is more violent compared with the LES approach that shows a better-defined stratification between the phases. Furthermore, the influence of the wave train on the brackish water column that emerges from the submarine spring can be observed, together with the formation of large eddies of about 1.5-2 m of

diameter. These eddies appear near the water surface due to the lower density and are later dissolved in the saltwater phase. The influence zone tends to remain in the upper area of the saltwater column forming a brackish water layer. For the conditions tested, the influence zone has a radius of ~20 m around the submarine spring (for both turbulence approaches).

Streamlines and velocity vectors were plotted for a case without the inflow of freshwater (wave propagation only) and for the case with an “active” submarine spring (Figure 41. and Figure 42).

Figure 41. and Figure 42 show how the freshwater inflow affects the flow, especially in the zone near the free surface and near the submarine spring. The brackish water layer that is formed near the free surface represents an obstacle for the wave train and causes a zone with a chaotic flux and the formation of eddies (4 to 6 m diameter).

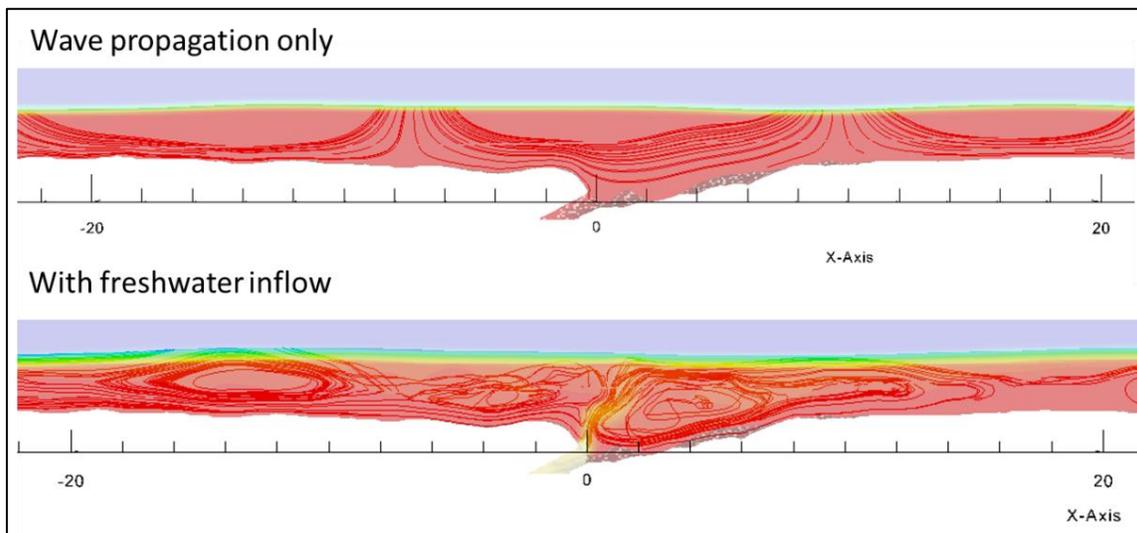


Figure 41. Streamlines for the case with wave propagation only and the case with freshwater inflow.

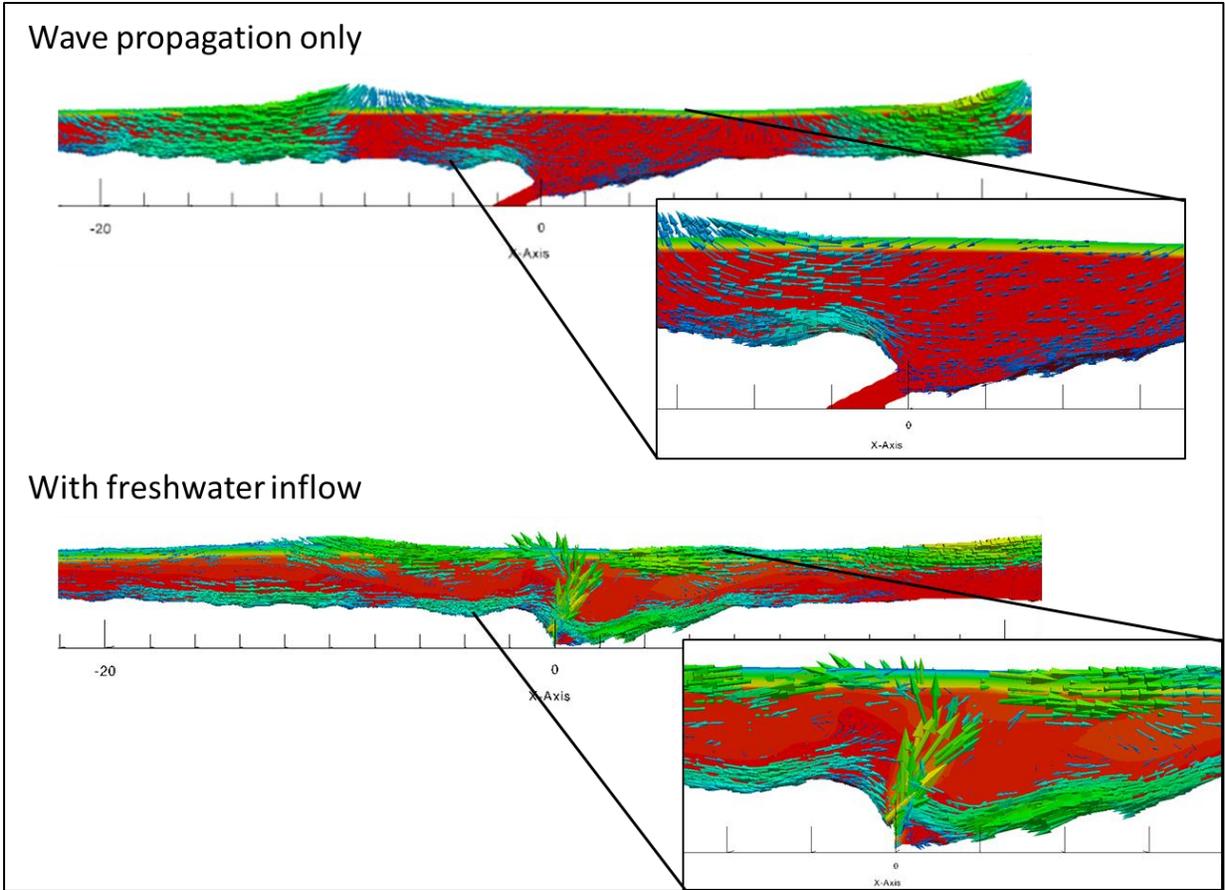


Figure 42. Velocity vectors for the case with wave propagation only and the case with freshwater inflow. The vector's size is proportional to the vector's value. Each case shows a zoom of the area near the submarine spring.

The results presented in this section only demonstrate the capacity of the solver `interMixingWaveFoam` to simulate point source SGD. These results are preliminary, thus further analysis are required in order to determine the real influence of submarine springs under certain conditions of interest. Further analysis could be helpful to determine the influence in the system of factor such as the exit angle of the brackish water, the wave intensity and the flux rate discharged.

# CONCLUSIONS

## 7. CONCLUSIONS

In this investigation it was found the need for developing a tool capable to simulate in detail a coastal process that involves submarine groundwater discharge. Several factors are involved in this process, such as waves, flow rate input from the submarine spring, the density gradient between brackish water and saltwater, sea level, free surface condition and the mixture between fluids. Therefore the numerical tool should be able to reproduce, at least, the aforementioned aspects. Within this context, three main tasks have been fulfilled in the present thesis: i) the development of a numerical model based on a CFD toolbox for the simulation of a wave field environment and a three-phase fluid system (air, saltwater and fresh/brackish water) where two of the phases are mixable; ii) the validation and calibration of the model with experimental data from benchmarking tests; and iii) the application of the model in a study case of a point-source SGD in the Yucatan Peninsula, Mexico.

The CFD platform OpenFOAM® was selected to develop the numerical model required. The solver developed, validated and implemented was shown to provide satisfactory results and accomplishing the objectives for which it was created.

After the development of the numerical tool `interMixingWaveFoam` many restrictions and possibilities of the model were identified. The limitations and the impact of the numerical tool presented are described in this section.

### 7.1. Restrictions and limitations of the model

Every CFD model must be used carefully and `interMixingWaveFoam` is not the exception. Additionally to the typical restrictions with any other OpenFOAM® solver, there has to be taken into account the following restriction of `interMixingWaveFoam`:

- i. The model doesn't consider temperature in an explicit way but it is considered only in the temperature-dependent values such as the viscosity and density. Furthermore, these values are introduced as constants in the case setup.
- ii. The salinity is not explicitly considered in the model, neither the separation processes of already mixed fluids. The model is only able to work with different densities and one molecular diffusivity coefficient between mixable phases.
- iii. The flux is considered incompressible for each fluid phase, including the air phase.
- iv. For large domains and complex meshes it is important to monitor the convergence of the solutions, since the presence of relative small cells can induce the collapse of the model and stopping the simulations.

Another important issue to take into account is the selection of the turbulence model for the equations closure. As it was observed in the simulations, the choice of the turbulence model affects the fluid behavior in a significant way, in terms of eddies development, energy dissipation and mixture between fluid phases. Thus, one turbulence model could reproduce the reality better than the other one or even give results little close to the reality. Furthermore, two or more turbulence models could be useful for one simulation depending on the dominant conditions. This was observed in the validation process where

the simulation was very precise under certain turbulence models for the first part of the experiment, and other turbulence models for the second part of the experiment.

## 7.2. Impact of the model

Previous numerical simulations of point source SGD have been conducted with OpenFOAM® preexisting solvers in order to compare the numerical results with the collected (in situ) data in *Pargos* spring (the studies were conducted by CINVESTAV Investigation Center in collaboration with UNAM). Those simulations were done without considering the buoyant effect of the fluid with lower density. Instead, the simulations only considered the effect caused by the different fluid flow directions between the wave train and the spring discharge in a free surface condition. The preliminary results of those tests have shown coincidences between one oscillation type (~100 s period that does not belong to the infragravity oscillations) found in the measurements as well as in the controlled conditions of the numerical simulation. Thus, the existence of this oscillation is attributed to the presence of the submarine spring and a recirculation cell that is induced by it. The next step is to include the buoyant effect of the brackish water discharge, which can be done by conducting the same analysis with `interMixingWaveFoam` tool.

As mentioned in section 2 of this work, point source SGD has an important ecological implication since it is a fast connection between the sea and the freshwater aquifer, thus submarine springs can bring into the sea nutrients and pollutants from the land. The nutrient loading can be significant to reef environments and to understand this process is important for future management strategies regarding coastal development. Therefore, `interMixingWaveFoam` is a tool that can be used to simulate different study cases with similar characteristics of point source SGD; that means, an incident wave train, the discharge of a fluid with different density to the discharge environment, the mixture between the liquid phases and a free surface condition. One example of a similar case is the sewage discharge into the shore.

As any other CFD tool, `interMixingWaveFoam` can be improved and extended in order to include more issues such as the temperature effects, which could multiply the application possibilities and precision of the results.

### **Conclusiones**

*En esta investigación se encontró la necesidad de desarrollar una herramienta capaz de simular a detalle un proceso costero que involucra las descargas submarinas de agua dulce. Existen múltiples factores que afectan este proceso, por ejemplo, el gasto que descarga el manantial submarino, el gradiente de densidad entre el agua salobre que descarga y el agua salada del medio, el nivel del mar, la condición de superficie libre, el oleaje y la mezcla entre el fluido descargado y el agua de mar. Por lo tanto, la herramienta numérica para representar la descarga submarina de agua dulce, debe ser capaz de representar por lo menos estos aspectos. En este contexto, tres tareas principales se llevaron a cabo en esta tesis: i) el desarrollo de un modelo*

numérico basado en un paquete de CFD (mecánica de fluidos computacional) para la simulación de un tren de oleaje a superficie libre y la descarga submarina de agua dulce o salobre, así como la mezcla entre fases; ii) la validación y calibración del modelo con datos experimentales de pruebas de laboratorio representativas; iii) la aplicación del modelo en un caso de estudio de un ojo de agua en la península de Yucatán, México.

La plataforma de CFD OpenFOAM® fue seleccionada para el desarrollo del modelo numérico. El módulo desarrollado dentro de la plataforma, *interMixingWaveFoam*, fue validado e implementado, demostrándose que provee resultados satisfactorios y cumple con los objetivos para los cuales fue creado.

Después del desarrollo de *interMixingWaveFoam* fueron identificadas tanto restricciones o limitaciones así como potenciales aplicaciones. Las limitaciones y el impacto de esta herramienta se describen en esta sección.

#### Restricciones y limitaciones del modelo

Todo modelo CFD debe utilizarse con el debido cuidado y la herramienta *interMixingWaveFoam* no es la excepción. Adicionalmente a las limitaciones de cualquier módulo en OpenFOAM®, se deben tomar en cuenta los siguientes puntos referentes a *interMixingWaveFoam*:

- i. El modelo no considera el gradiente de temperatura en una forma explícita, sino que solamente toma en cuenta parámetros constantes dependientes de la temperatura, como la viscosidad y la densidad.
- ii. La salinidad no es considerada explícitamente; tampoco se consideran los procesos de separación de los fluidos una vez mezclados. El modelo solo es capaz de trabajar con diferentes densidades y un coeficiente de difusividad molecular.
- iii. El flujo es considerado incompresible en cada una de las fases, incluyendo la fase gaseosa del aire.
- iv. Para dominios muy grandes y mallas complejas es importante monitorear la convergencia de las soluciones, ya que la presencia de celdas muy pequeñas puede ocasionar el colapso de la corrida del modelo.

Otra cuestión que es importante tomar en cuenta es la selección del modelo de turbulencia para las ecuaciones de cierre. Como se observó en las simulaciones, la selección del modelo de turbulencia afecta el comportamiento hidrodinámico en una forma significativa, en términos de la formación de remolinos, la energía disipada y la mezcla entre fases. Entonces, debe considerarse que un modelo de turbulencia puede representar mejor la realidad en comparación con otro, dependiendo de las condiciones dominantes. Se observó en el proceso de validación del modelo numérico, que las simulaciones eran muy precisas con ciertos modelos de turbulencia en la primera parte del experimento, y otros modelos de turbulencia para la segunda parte del experimento.

### Impacto del modelo

Recientemente, simulaciones numéricas de las descargas submarinas de agua dulce han sido desarrolladas con el software OpenFOAM y sus módulos preexistentes, con el fin de comparar los resultados numéricos con los datos recolectados en sitio, en particular el ojo de agua Pargos, en la Península de Yucatán. Estos estudios fueron llevados a cabo en el Centro de Investigación CINVESTAV en colaboración con la UNAM. Sin embargo, las simulaciones se hicieron sin considerar el efecto del gradiente de densidades entre el agua salobre y el agua salada, sino que solo se consideró el efecto de las diferentes direcciones de flujo entre el tren de oleaje en condición de superficie libre y la descarga del ojo. Los resultados preliminares de esos estudios han mostrado reproducir las oscilaciones (de aproximadamente 100 s de periodo que no corresponden a las oscilaciones de infragravedad) que fueron encontradas en campo, pero también dentro de las condiciones controladas del modelo numérico. Así pues, la existencia de estas oscilaciones es atribuida entonces a la presencia del ojo de agua. El siguiente paso es incluir el efecto boyante del agua del ojo, lo cual ahora podrá ser llevado a cabo este análisis con la herramienta *interMixingWaveFoam* validada en este trabajo.

Como se mencionó en la sección 2 de esta tesis, las descargas submarinas de agua dulce en forma de manantial, tienen un importante impacto ecológico, ya que son un medio de comunicación directo entre el agua del acuífero y el mar, por lo tanto los ojos de agua pueden descargar nutrientes y contaminantes al medio marino. La descarga de nutrientes puede afectar ecosistemas como los arrecifes de coral, los cuales tienen importancia ecológica y de preservación de las playas; por lo que entender cómo se lleva a cabo para futuras estrategias de cuidado y planeación costera. Por ello, *interMixingWaveFoam* es una herramienta que puede ser utilizada en casos con características afines a los procesos costeros con ojos de agua. Un ejemplo es la descarga de aguas negras dentro del mar.

Como cualquier otra herramienta de CFD, *interMixingWaveFoam* puede ser mejorado y extendido a fin de incluir más variables como son los efectos del gradiente de temperatura, lo cual podría multiplicar su rango de aplicación, así como la precisión de los resultados.

# Annexes

## 8. Annexes

This section includes an example on how to set a case with the numerical tool developed in this study, *interMixingWaveFoam*, within the platform OpenFOAM®. There are also code lines and listings that could be helpful in order to understand how to set and run any case that includes wave generation/absorption, 2 mixable liquid phases and a free surface condition.

### 8.1. Annex A. Case setup example with *interMixingWaveFoam*

A detailed description of how to set a case in OpenFOAM® platform with *interMixingWaveFoam* is now presented.

#### 8.1.1. Preliminary model setup

The case exemplified in this section is named *waveFlumeSpring*, which consists on a rectangular domain partially filled with saltwater. Waves are generated on the left side of the domain and absorbed on the right side (Figure 43). The rest of the domain is filled with air, so that the saltwater block has a free surface. In the middle of the domain there is a patch with a constant freshwater fluid flow and representing the point source SGD or spring.

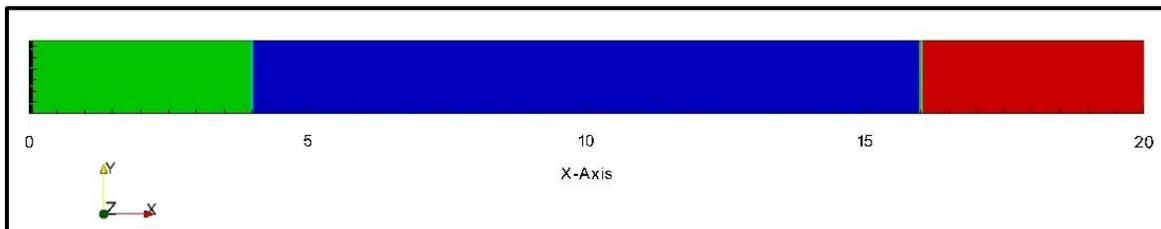


Figure 43. Relaxation zones defined in the domain of the *waveFlumeSpring* case.

The green zone represents the wave generation zone and the red one the absorption zone.

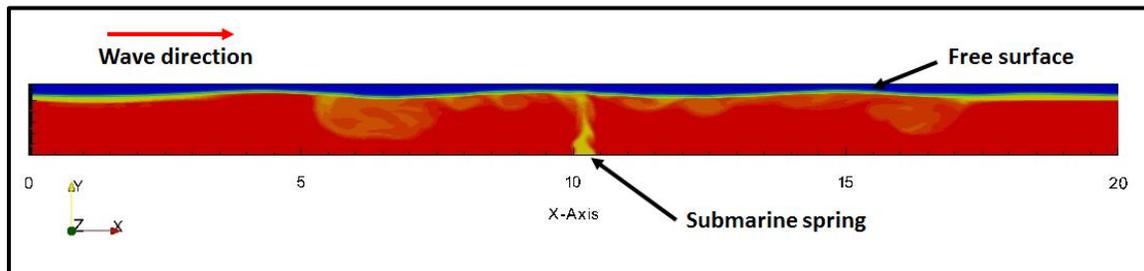


Figure 44. *waveFlumeSpring* case. Saltwater is represented in red, freshwater in yellow and air in blue.

As mentioned above, this case is a schematic representation of the study case; therefore it has a rectangular domain and no refinements or sculpts in the mesh. The case doesn't include a turbulence model. The basic data for the setup (domain size, fluid properties, wave properties, initial conditions, etc.) of this case are summarized in Table 7 and Table 8.

Table 7. Characteristics of the domain, mesh, gravity, freshwater flow rate and phases properties for the *waveFlumeSpring* case

		Domain and mesh		Gravitational acceleration	Submarine spring (0.4 m length)
Direction	Dimension (m)	Cell number	a (m/s <sup>2</sup> )	Velocity v (m/s)	
x	20	500	0	0	
y	1.3	65	-9.81	0.2	
z	0.1	1	0	0	
Phase properties					
Phase property		Freshwater	Saltwater	Air	
Density (kg/m <sup>3</sup> )		1000	1030	1	
Kinematic viscosity (m <sup>2</sup> /s)		1x(10) <sup>(-6)</sup>	1x(10) <sup>(-6)</sup>	1.48x(10) <sup>(-5)</sup>	
Surface tension (N/m)		0.7	0.7		
Molecular diffusion coefficient (m <sup>2</sup> /s)		1.26x(10) <sup>(-9)</sup>			

Table 8. Wave properties and simulation time for the *waveFlumeSpring* case.

Wave properties	
Type	Regular wave train
Wave theory	Linear
Period T (s)	2
Wave height H (m)	0.1
Depth h (m)	1
Simulation time (s)	40

### 8.1.2. Mesh generation

As any other case in OpenFOAM®, a folder must be designated for the case setup. When the case is run, the results will be automatically saved within this directory. The folder's name is *waveFlumeSpring* and is located on any directory of the Linux platform used. Inside the folder there are 3 basic folders named: i) *0*, ii) *constant* and iii) *system* (Figure 45). A folder named *0.org* is created in order to save the original files of the boundary conditions.



Figure 45. Content of the *waveFlumeSpring* case. There are 3 basic main folders required by any case in OpenFOAM®: i) *0*, ii) *constant* and iii) *system*.

The files for the mesh generation are located inside the *constant/polyMesh* directory (Figure 46).

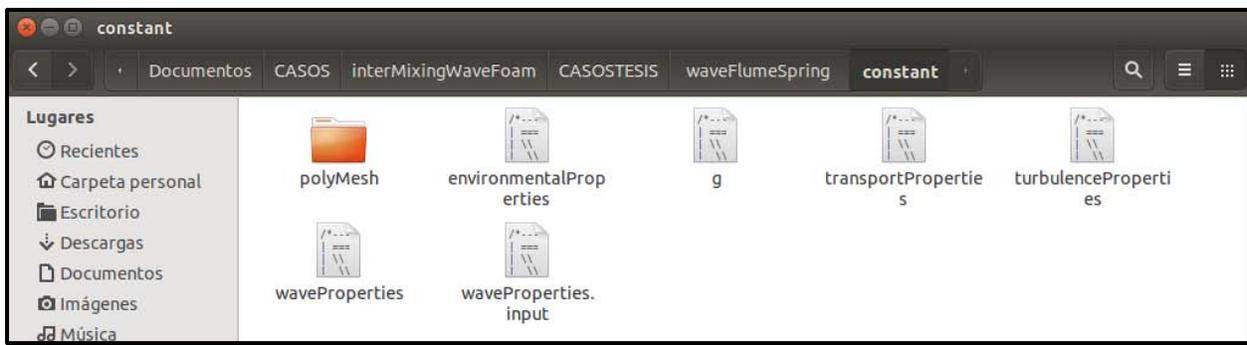


Figure 46. Inside the folder *constant*, there is a folder called *polyMesh*. *polyMesh* contains the files needed for the mesh generation.

The *polyMesh* folder contains a file called *blockMeshDict*, which is an input dictionary for the mesh generation. This file is designed to be run with *blockMesh*, a pre-processing OpenFOAM® tool which generates simple, unstructured meshes. The *blockMeshDict* entries for the case of interest are as shown in Listing 3.

The first input is *vertices*, where the coordinate vertices are written. The domain is 20 m long, 1.30 m high and 0.10 m width (Listing 3, line 2 to 13). Then, the mesh blocks and cells divisions per each side are established. The domain for this case has 500 divisions in the X direction, 65 divisions in the Y direction and 1 in the Z direction (2D condition) (Listing 3, line 14 to 17). The number for divisions on each side defines the cell's shape and size. The cell shape should be approximately a square in order to maintain the aspect ratio and decrease numerical diffusion problems. The number of cells and the mesh resolution should be enough to represent accurately the phenomenon, that is, there is a minimum amount of points to describe correctly the waves vertically and horizontally. Finally, each block patch is defined (Listing 3, line 21 to 46).

Listing 3. Entries for the *blockMeshDict* input Dictionary in the *waveFlumeSpring* case.

```
// * * * * *
...
1  convertToMeters 1;
```

```

2 vertices
3 (
4     ( 0 -1 0 )
5     ( 20 -1 0 )
6     ( 0 0.3 0 )
7     ( 20 0.3 0 )
8
9     ( 0 -1 0.1 )
10    ( 20 -1 0.1 )
11    ( 0 0.3 0.1 )
12    ( 20 0.3 0.1 )
13 );
14 blocks
15 (
16     hex (0 1 3 2 4 5 7 6) ( 500 65 1 ) simpleGrading (1 1 1)
17 );
18 edges
19 (
20 );
21 patches
22 (
23     patch inlet
24     (
25         (0 4 6 2)
26     )
27     wall bottom
28     (
29         (0 1 5 4)
30     )
31     patch outlet
32     (
33         (1 5 7 3)
34     )
35
36     patch atmosphere
37     (
38         (2 3 7 6)
39     )
40
41     empty frontBack
42     (
43         (0 1 3 2)
44         (4 5 7 6)
45     )
46 );
47 mergePatchPairs
48 (
49 );
// ***** //

```

The boundary conditions for this case are defined as follows. The patch in the bottom of the domain has a `wall` boundary condition and it creates an impermeable patch. The patches in the front and in the back of the domain have an empty boundary condition, which is used for the 2D modelling. The patch that represents the opening to the atmosphere will allow the inflow and outflow of fluid, which maintains the free surface condition of the water block. The walls that represent the inlet and outlet of the domain are set with a generic `patch` condition and they will be later configured to generate and absorb waves respectively.

In order to define the mesh in the simplest way, the patch for the submarine spring is not defined in this dictionary, but is later created with other OpenFOAM® applications. This process is now described. The following steps are specific for *waveFlumeSpring* mesh generation.

The patch that represents the submarine spring is called “*ojo*”. This patch is done with two OpenFOAM® tools: *topoSet* and *createPatch*. Both applications require dictionaries located in the *system* directory. The first input dictionary is called *topoSetDict* and defines a new cell set called “*ojo*” and its coordinates (Listing 4).

Listing 4. Entries for the *topoSetDict* dictionary. This dictionary contains the instructions for the creation of a cell set called “*ojo*” and is used by the *topoSet* application.

```

// * * * * *
...
actions
(
  {
    name      ojo;
    type      faceSet;
    action    new;
    source    boxToFace;
    sourceInfo
    {
      box      (10.0012 -1.002 0) (10.40012 -0.999 0.1);
    }
  }
);
// * * * * *

```

The second one is *createPatchDict* and contains the instructions to create the patch “*ojo*” from the cell set generated by *topoSet* tool (Listing 5).

Listing 5. Entries for the *createPatchDict* dictionary. This file contains the instructions to create the patch “*ojo*” from the cell set generated by *topoSet* tool and is used by *createPatch* application.

```

// * * * * *
...
pointSync false;
// Patches to create.
patches
(
  {
    // Name of new patch
    name ojo;

    // Dictionary to construct new patch from
    patchInfo
    {
      type patch;
    }
    // How to construct: either from 'patches' or 'set'
    constructFrom set;
    // If constructFrom = patches : names of patches. Wildcards allowed.
    patches ();
    // If constructFrom = set : name of faceSet
    set ojo;
  }
);

```

```

    }
);
// ***** //

```

The final mesh is generated by the following instructions (Listing 6), which are run in the terminal, within the case directory.

Listing 6. Instructions to generate the *waveFlumeSpring* case mesh.

```

blockMesh
topoSet
createPatch -overwrite

```

Once the mesh is generated, the mesh can be viewed in a compatible visualizer tool, e.g. Paraview®.

### 8.1.3. Physical and fluid properties

The physical properties and the fluid properties required by *interMixingWaveFoam* are set in the *constant* folder directory. The solver needs data of environmental properties such as the value and direction of the gravitational acceleration, transport properties, turbulence properties and wave properties. The transport properties are the properties of each fluid phase, that is, kinematic viscosity, density, surface tension (Newtonian fluid essential properties) and diffusivity between miscible phases. The turbulence properties define whether the solution includes or not a turbulence model and which turbulence model will be used. Finally, the wave properties are the wave train characteristics (period, height, direction and wave theory) and the definition of the relaxation zones and its dimensions. Each required value is given in International System. For each property, an input dictionary is created within the *constant* directory (Figure 47).

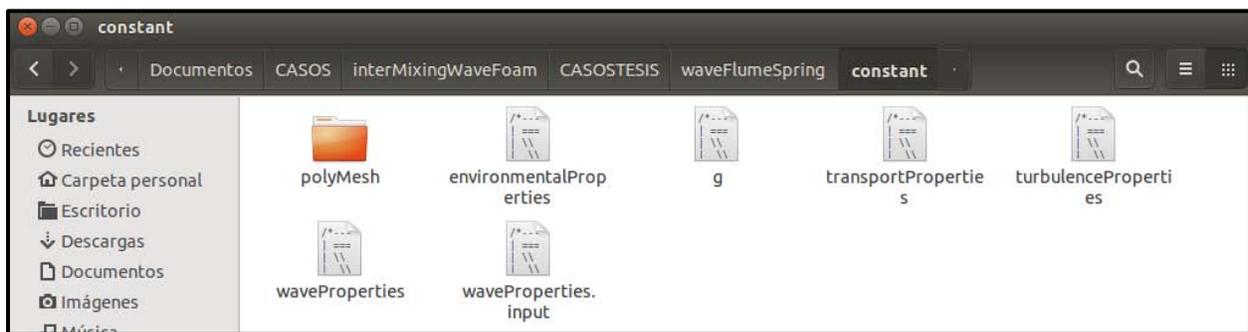


Figure 47. Content of the *constant* folder. This folder contains input dictionaries for each physical and fluid properties required by *interMixingWaveFoam*.

The only environmental property required is the gravitational acceleration and is defined in *environmentalProperties* input dictionary as it is shown in Listing 7. The units for this property are  $m/s^2$ .

Listing 7. Definition of the gravitational acceleration in *environmentalProperties* dictionary.

```

// ***** //
g          g [0 1 -2 0 0 0] (0 -9.81 0);
// ***** //

```

The kinematic viscosity, density, surface tension and diffusivity between miscible phases are defined inside the `transportProperties` input dictionary, as shown in Listing 8. Phase 1, 2 and 3 represent air, saltwater and freshwater respectively. The surface tension is defined between air and saltwater, and between air and freshwater. Units for the surface tension are  $\text{kg/s}^2$ . Finally the diffusivity between miscible phases is defined for saltwater and freshwater. This value is the molecular diffusivity and is measured in  $\text{m}^2/\text{s}$ . The value for diffusivity coefficient is taken from Holz et al., 2000

Listing 8. Definition of fluid phases' transport properties in `transportProperties` dictionary.

```
// * * * * *
//air
phase1
{
    transportModel Newtonian;
    nu nu [ 0 2 -1 0 0 0 0 ] 1.48e-05;
    rho rho [ 1 -3 0 0 0 0 0 ] 1;
}
//saltwater
phase2
{
    transportModel Newtonian;
    nu nu [ 0 2 -1 0 0 0 0 ] 1e-06;
    rho rho [ 1 -3 0 0 0 0 0 ] 1030;
}
//freshwater
phase3
{
    transportModel Newtonian;
    nu nu [ 0 2 -1 0 0 0 0 ] 1e-06;
    rho rho [ 1 -3 0 0 0 0 0 ] 1000;
}
// Surface tension coefficients
sigma12          sigma12 [1 0 -2 0 0 0 0] 0.07;
sigma13          sigma13 [1 0 -2 0 0 0 0] 0.07;

// Diffusivity between miscible phases
D23              D23   [0 2 -1 0 0 0 0] 3e-09;
// * * * * *
```

In the input dictionary `turbulenceProperties` the simulation type is defined. For this case, the simulation type is set as “*laminar*” (Listing 9). This doesn’t mean that the fluid system behaves as laminar, but the simulation will not use a turbulence model. Other examples for the simulation type are RAS model and LES model.

Listing 9. The simulation type in `turbulenceProperties` dictionary is set as “*laminar*”. This means that no turbulence model will be used in the simulation.

```
// * * * * *
simulationType laminar;
// * * * * *
```

Finally, the wave properties are set in the `waveProperties.input` dictionary (Listing 10). In this file, the relaxation zones are specified as two subdictionaries. The first one is *inlet* and it defines the wave

parameters for a regular (or irregular) wave train. The second one is *outlet* and represents the absorption zone. This relaxation zone *outlet* avoids the wave reflection at the end of the domain. Both, *inlet* and *outlet* zones have 4 m length at the beginning and at the end of the domain respectively. For this case, a regular wave train is chosen; the wave period is 2 s and, the wave height is 0.10 m and the depth is 1 m. The wave theory for the wave generation is linear (airy). Thus the wave type is set to *stokesFirst*. Further instructions on how to set this dictionary are given by Jacobsen et al., (2011).

Listing 10. Entries for *waveProperties.input* dictionary.

```

// * * * * *
seaLevel      0.00;
relaxationNames (inlet outlet);
initializationName outlet;
inletCoeffs
{
    waveType      stokesFirst;
    Tsoft         2;
    depth         1.00000;
    period        2.0;
    phi           0.000000;
    direction     (1.0 0.0 0.0);
    height        0.1;
    relaxationZone
    {
        relaxationScheme Spatial;
        relaxationShape Rectangular;
        beachType      Empty;
        relaxType      INLET;
        startX         (0 0.0 -1);
        endX           (4.0025 0.0 1);
        orientation    (1.0 0.0 0.0);
    }
};
outletCoeffs
{
    waveType      potentialCurrent;
    U             (0 0 0);
    Tsoft         2;
    relaxationZone
    {
        relaxationScheme Spatial;
        relaxationShape Rectangular;
        beachType      Empty;
        relaxType      OUTLET;
        startX         (16.0025 0.0 -1);
        endX           (20 0.0 1);
        orientation    (1.0 0.0 0.0);
    }
};
// * * * * *

```

#### 8.1.4. Field boundary and initial conditions

The boundary and initial conditions are defined within the *0* directory. This directory contains one file per variable field considered in the case. The files are *alpha1* (air), *alpha2* (saltwater), *alpha3* (freshwater), *p\_rgh* (dynamic pressure) and *U* (velocity). The input dictionary *alpha1* contains the boundary conditions for the air phase and they are defined for each patch (Listing 11). Since no turbulence





```

    }
    bottom
    {
        type          zeroGradient;
    }
    outlet
    {
        type          zeroGradient;
    }
    atmosphere
    {
        type          inletOutlet;
        inletValue    uniform 0;
        value         uniform 0;
    }
    frontBack
    {
        type          empty;
    }
    ojo
    {
        type          inletOutlet;
        inletValue    uniform 1;
        value         uniform 1;
    }
}
// ***** //

```

The boundary conditions and initialization values must be also defined for the pressure and the velocity field. The `p_rgh` input dictionary contains this information for the dynamic pressure field, which is defined as

$$p\_rgh = p - (\rho * gh)$$

with  $p$  as the total pressure and  $\rho * gh$  representing the hydrostatic pressure.

Most patches have a `zeroGradient` boundary condition, except for the `frontBack` and `atmosphere` patches. Patch `atmosphere` has a `totalPressure` boundary condition, which fixes the total pressure  $p$  when the velocity  $U$  changes (Listing 14).

Listing 14. Entries for the `p_rgh` input dictionary. This directory contains the boundary conditions or initialization values for the dynamic pressure field.

```

// ***** //
dimensions      [1 -1 -2 0 0 0 0];
internalField    uniform 0;
boundaryField
{
    inlet
    {
        type          zeroGradient;
    }
    bottom
    {
        type          zeroGradient;
    }
    outlet

```

```

{
    type          zeroGradient;
}
atmosphere
{
    type          totalPressure;
    U             U;
    phi           phi;
    rho           none;
    psi           none;
    gamma         1;
    p0            uniform 0;
    value         uniform 0;
}
frontBack
{
    type          empty;
}
ojo
{
    type          zeroGradient;
}
}
// ***** //

```

Finally the velocity field  $U$  is initialized in the  $U$  input dictionary (Listing 15). The patch `inlet` has `waveVelocity` boundary condition, which generates the wave train velocity in the `inlet` patch. The `bottom` and `outlet` patches have `fixedValue` boundary condition, which sets the initial velocity value to zero. Atmosphere patch has a `pressureInletOutletVelocity` boundary condition, which is used for patches adjacent with the atmosphere. Therefore, this patch needs to permit both outflow and inflow according to the internal flow, for which a combination of boundary conditions for pressure and velocity is used to do this task while maintaining stability. The `pressureInletOutletVelocity` boundary condition applies `zeroGradient` on all components, except where there is inflow, in which case a `fixedValue` condition (specific value of  $\phi$ ) is applied to the tangential component. The `frontBack` patch is set to `empty` because of the 2D condition. The patch that simulates the freshwater spring, `ojo`, has also a `fixedValue` boundary condition but it is used to set the constant flow in the vertical direction through this patch. The velocity is set to 0.20 m/s in the vertical direction.

Listing 15. Entries for the  $U$  input dictionary. This directory contains the boundary conditions or initialization values for the velocity field.

```

// ***** //
dimensions      [0 1 -1 0 0 0 0];
internalField   uniform ( 0.0 0.0 0.0);
boundaryField
{
    inlet
    {
        type          waveVelocity;
        refValue       uniform ( 0 0 0 );
        refGradient    uniform ( 0 0 0 );
        valueFraction  uniform 1;
        value          uniform ( 0 0 0 );
    }
}

```

```

    }
    bottom
    {
        type          fixedValue;
        value          uniform ( 0 0 0 );
    }
    outlet
    {
        type          fixedValue;
        value          uniform ( 0 0 0 );
    }
    atmosphere
    {
        type          pressureInletOutletVelocity;
        value          uniform ( 0 0 0 );
    }
    frontBack
    {
        type          empty;
    }
    ojo
    {
        type          fixedValue;
        value          uniform (0 0.20 0);
    }
}
// ***** //

```

### 8.1.5. Initial field

The initial fields are set in the input dictionary named `setFieldsDict` (Listing 16). A block of saltwater (phase2) partially fills the computational domain with depth of 1 m. The rest of the domain is filled by air (alpha1) and is set as a default field value. Additionally, the velocity field is initially set to 0 m/s.

Listing 16. Input dictionary `setFieldsDict`. A block of saltwater (alpha2) partially fills the domain and the rest is filled by air (alpha1).

```

// ***** //
defaultFieldValues
(
    volScalarFieldValue alpha1 1
    volScalarFieldValue alpha2 0
    volScalarFieldValue alpha3 0
    volVectorFieldValue U ( 0 0 0 )
);
regions
(
    boxToCell
    {
        box (-1 -5 0) (30 0 1);
        fieldValues
        (
            volScalarFieldValue alpha1 0
            volScalarFieldValue alpha2 1 //SALTWATER
            volScalarFieldValue alpha3 0
        );
    }
);
// ***** //

```



```

timeFormat      general;
timePrecision   6;
runTimeModifiable yes;
adjustTimeStep yes;
maxCo           0.25;
maxAlphaCo     0.25;
maxDeltaT      1;

functions
{
    #includeIfPresent "../waveGaugesNProbes/surfaceElevationAnyName_controlDict";
}
// ***** //

```

The maximal Courant number allowed for the case is also set in this dictionary. The Courant number is a dimensionless parameter that indicates the maximal time step allowed to achieve temporal accuracy and convergence while solving the partial differential equations system numerically. The Courant number is defined for a cell as

$$Co = \frac{\delta t |U|}{\delta x}$$

Where  $\delta t$  is the time step, which is 0.001 s for this case; and  $\delta x$  is the cell size in the direction of the velocity  $U$ . For this case Courant number of less of 1 is required, thus the maximal Courant number is set to 0.25, and the maximal time step is set to 1 s.

**8.1.7. Case running and post-processing**

The case running is done through running the respective commands for the mesh generation, the wave parameters and field’s settings, and finally, the execution of `interMixingWaveFoam`. Instead of running each application command in the terminal, a bash script file was created for this purpose (Code 3). The applications that must be run before `interMixingWaveFoam` are the following:

- 1) *blockMesh*: this application generates a simple, unstructured mesh which represents the case physical domain.
- 2) *topoSet*: it defines the cell set that will be used as a new patch.
- 3) *createPatch*: it creates the patch that represents the freshwater spring at the bottom of the domain.
- 4) *setWaveParameters*: it sets the wave parameters, e.g. wave number and wave length.
- 5) *setFields*: it sets the default values of the fluid phases within the domain.
- 6) *relaxationZoneLayout* (optional): this application is optional and is used to visualize the correct specification of the relaxation zones in the domain.

Code 3. Bash script created for the case running with `interMixingWaveFoam`.

```

#!/bin/bash
# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

exec="prepareCase.sh"
if [ -x "$exec" ]

```

```

then
  . $exec
else
  echo "The file $WAVES_DIR/bin/prepareCase.sh is not executable."
  echo "Make the file executable before continuing."
  echo
  echo "Exiting tutorial case."
  exit 1
fi

# Set application name
application="interMixingFoamPaola"

runApplication blockMesh
runApplication topoSet
runApplication createPatch -overwrite

runApplication setWaveParameters
runApplication setFields
runApplication relaxationZoneLayout

runApplication $application

```

The instructions for the generation of a new patch, `topoSet` and `createPatch`, are specific for the *waveFlumeSpring* case and its mesh. Only `blockMesh`, `setWaveParameters` and `setFields` are generic and necessary instructions for the correct use of `interMixingWaveFoam` in any proposed case.

Figure 49 shows some screenshots of the case that was run. The colors represent the different densities for the 3-phase system (air in blue, saltwater in red and freshwater in yellow). The wave propagation and the mixture between fluid phases is also shown.

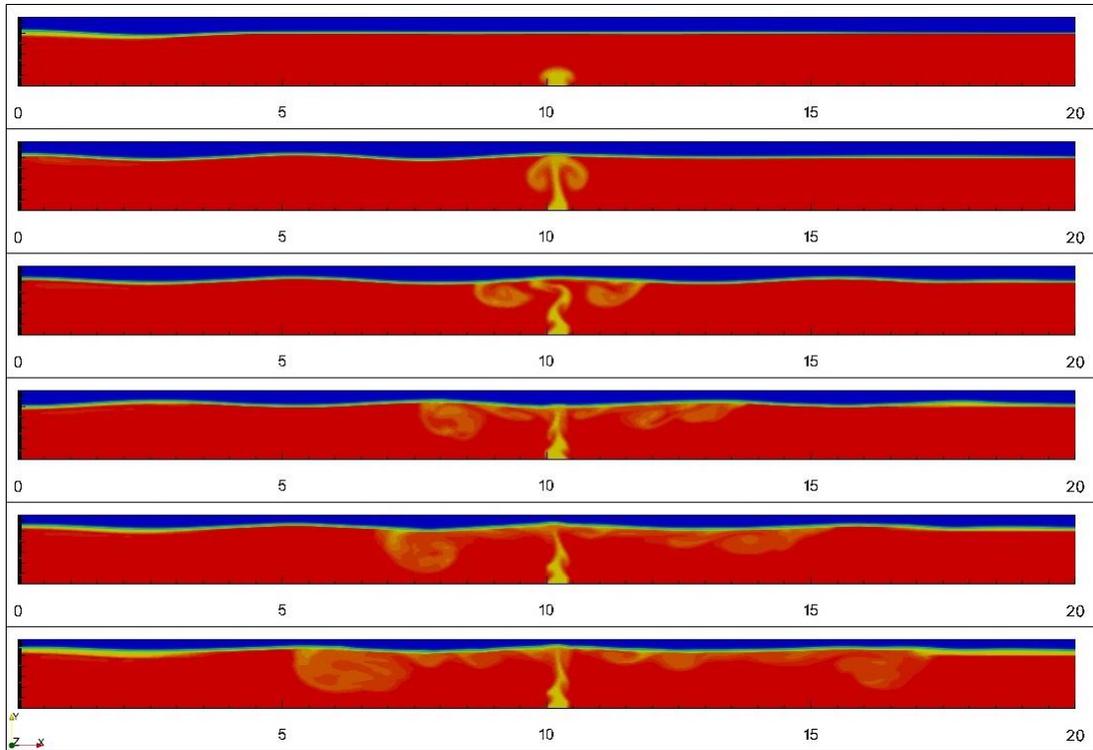


Figure 49. Screenshots of the *waveFlumeSpring* case that was run with *interMixingWaveFoam*. Saltwater is represented in red, freshwater in yellow and air in blue. The scenes correspond to the time 2, 6, 10, 15, 20 and 30 s respectively.

The results in Figure 49 are representative of the ability of the solver to reproduce a wave train in a free surface condition and the inflow of a liquid with different density. The lower density of the input fluid causes that it remains in the upper part of the saltwater block. Additionally, the influence of the wave motion induces the formation of eddies and the dissolution between phases. Thus, the model can predict an approximate radius of the influence zone of the SGD.

## 9. References

- Alcérreca-Huerta, J.C., Oumeraci, H., 2016. Wave-induced pressures in porous bonded revetments. Part II: Pore pressure just beneath the revetment and in the embankment subsoil. *Coast. Eng.* 110, 76–86.
- ANSYS, 2015. ANSYS CFX [WWW Document]. URL <http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics/Fluid+Dynamics+Products/ANSYS+CFX> (accessed 11.6.15).
- Bakalowicz, M., 2014. Karst at depth below the sea level around the Mediterranean due to the Messinian crisis of salinity . *Hydrogeological consequences and issues. Geol. Belgica* 96–101.
- Bardina, J.E., Field, M., Huang, P.G., Coakley, T.J., Field, M., Aeronautics, N., 1997. *Turbulence Modeling Validation , Testing , and Development*.
- Bauer-Gottwein, P., Gondwe, B.R.N., Charvet, G., Marín, L.E., Rebolledo-Vieyra, M., Merediz-Alonso, G., 2011. Review: The Yucatán Peninsula karst aquifer, Mexico. *Hydrogeol. J.* 19, 507–524. doi:10.1007/s10040-010-0699-5
- Beddows, P.A., 2004. Groundwater hydrology of a coastal conduit carbonate aquifer : Caribbean coast of the Yucatan Peninsula, Mexico.
- Berberović, E., Van Hinsberg, N.P., Jakirlić, S., Roisman, I. V., Tropea, C., 2009. Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* 79. doi:10.1103/PhysRevE.79.036306
- Bokuniewicz, H., Pollock, M., Blum, J., Wilson, R., 2004. Submarine ground water discharge and salt penetration across the sea floor. *Ground Water* 42, 983–989. doi:10.1111/j.1745-6584.2004.tb02637.x
- Brennen, C.E., 2005. *Fundamentals of Multiphase Flows*.
- Burnett, W., Aggarwal, P.K., Aureli, A., Bokuniewicz, H., Cable, J.E., Charette, M. a., Kontar, E., Krupa, S., Kulkarni, K.M., Loveless, A., Moore, W.S., Oberdorfer, J. a., Oliveira, J., Ozyurt, N., Povinec, P., Privitera, a. M.G., Rajar, R., Ramessur, R.T., Scholten, J., Stieglitz, T., Taniguchi, M., Turner, J. V., 2006. Quantifying submarine groundwater discharge in the coastal zone via multiple methods. *Sci. Total Environ.* 367, 498–543. doi:10.1016/j.scitotenv.2006.05.009
- Burnett, W., Dulaiova, H., Stringer, C., Peterson, R., 2006. Submarine Groundwater Discharge : Its Measurement and Influence on the Coastal Zone Tracing Groundwater Discharge via Natural Isotopic Tracers. *J. Coast. Res.* 2004, 2004–2007.
- Church, T.M., 1996. An underground route for the water cycle. *Nature*.
- Cung T. J, 2002. *Computational Fluid Dynamics*. Cambridge.

- Escolero, O., Marin, L.E., Domínguez-Mariani, E., Torres-Onofre, S., 2007. Dynamic of the freshwater-saltwater interface in a karstic aquifer under extraordinary recharge action: The Merida Yucatan case study. *Environ. Geol.* 51, 719–723. doi:10.1007/s00254-006-0383-1
- Fernández Rubio, R., Baquero Úbeda, J.C., 2006. *Acuíferos Kársticos Costeros. Introducción a su conocimiento* 60–97.
- Ferziger, J.H., Peric, M., 2002. *Computational Methods for Fluid Dynamics*, 3rd ed, Vasa. Springer. doi:10.1016/S0898-1221(03)90046-0
- FLOWScience, 2015. FLOW-3D [WWW Document]. URL <http://www.flow3d.com/> (accessed 11.6.15).
- Guo, X., Shen, L., 2013. Numerical study of the effect of surface waves on turbulence underneath. Part 1. Mean flow and turbulence vorticity. *J. Fluid Mech.* 733, 558–587. doi:10.1017/jfm.2013.451
- Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* 39, 201–225. doi:10.1016/0021-9991(81)90145-5
- Holz, M., Heil, S.R., Sacco, A., 2000. Temperature-dependent self-diffusion coefficients of water and six selected molecular liquids for calibration in accurate 1H NMR PFG measurements. *Phys. Chem. Chem. Phys.* 2, 4740–4742. doi:10.1039/b005319h
- Jacobsen, N.G., Fuhrman, D.R., Fredsøe, J., 2011. A wave generation toolbox for the open-source CFD library: OpenFoam. *Int. J. Numer. Methods Fluids* 1073–1088. doi:10.1002/flid
- Johannes, R., 1980. The Ecological Significance of the Submarine Discharge of Groundwater. *Mar. Ecol. Prog. Ser.*
- Kobayashi, T., Tsubokura, M., 2011. Current Status on Large-Eddy Simulation for Engineering Applications 10, 149–152.
- Loaiciga, H.A., Zektser, I.S., 2003. Estimation of Submarine Groundwater Discharge. *Water Resour.* 30, 473–479.
- Mcdonough, J.M., 2007. *INTRODUCTORY LECTURES on TURBULENCE Physics , Mathematics and Modeling.*
- Mijatovic, B.F., 1987. Problems de captage en regions karstiques littorales. *Bull. du Cent. d’Hidrogeologie. Unive. Neuchâtel* 65–106.
- OpenFlower, T., 2004. OpenFlower [WWW Document]. URL <http://openflower.sourceforge.net/index2.html> (accessed 11.6.15).
- Orszag, S.A., Patterson, G.S., 1972. Numerical Simulation of Three-Dimensional Homogeneous Isotropic Turbulence. *Phys. Rev. Lett.* 28, 76–79. doi:10.1103/PhysRevLett.28.76
- Parra, S.M., Mariño-Tapia, I., Enriquez, C., Valle-Levinson, A., 2014. Variations in turbulent kinetic energy

- at a point source submarine groundwater discharge in a reef lagoon. *Ocean Dyn.* 64, 1601–1614. doi:10.1007/s10236-014-0765-y
- Parra, S.M., Valle-Levinson, A., Mariño-Tapia, I., Cecilia, E., 2015. Salt intrusion at a submarine spring in a fringing reef lagoon Sabrina. *J. Geophys. Res.* 1–15. doi:10.1002/2014JC010459. Received
- Paulsen, R.J., O'Rourke, D., Smith, C.F., Wong, T.F., 2004. Tidal load and salt water influences on submarine ground water discharge. *Ground Water* 42, 990–999. doi:10.1111/j.1745-6584.2004.tb02638.x
- Qu, W., Li, H., Wan, L., Wang, X., Jiang, X., 2014. Numerical simulations of steady-state salinity distribution and submarine groundwater discharges in homogeneous anisotropic coastal aquifers. *Adv. Water Resour.* 74, 318–328. doi:10.1016/j.advwatres.2014.10.009
- Ruiz, D., 2012. Detectan acidificación de agua en arrecife coralino de Puerto Morelos. *sipse.com Inf. en todo momento.*
- Schiestel, R., 2008. *Modeling and Simulation of Turbulent Flows.* ISTE. doi:10.1002/9780470610848
- Shishaye, H. a., 2015. Technical Note: Groundwater flow modeling in coastal aquifers – the influence of submarine groundwater discharge on the position of the saltwater–freshwater interface. *Hydrol. Earth Syst. Sci. Discuss.* 12, 3753–3785. doi:10.5194/hessd-12-3753-2015
- Solutions for Water [WWW Document], 2012. URL <http://www.solutionsforwater.org/solutions/a-solution-to-the-world%E2%80%99s-water-scarcity-problem-exists-the-submarine-springs-of-fresh-water>
- St. Johns River Water Management District [WWW Document], 2016. URL <http://www.sjrwmd.com/springs/crescentbeach.html> (accessed 7.9.16).
- Taniguchi, M., Burnett, W.C., Cable, J.E., Turner, J. V., 2002. Investigation of submarine groundwater discharge. *Hydrol. Process.* 16, 2115–2129. doi:10.1002/hyp.1145
- Thais, L., Magnaudet, J., 1996. Turbulent structure beneath surface gravity waves sheared by the wind. *J. Fluid Mech.* 328, 313. doi:10.1017/S0022112096008749
- Tomislav, M., Höpken, J., Mooney, K., 2014. *The OpenFOAM Technology Primer, First. ed.* Sourceflux, Germany.
- Valle-Levinson, A., Mariño-Tapia, I., Enriquez, C., Waterhouse, A.F., 2011. Tidal variability of salinity and velocity fields related to intense point-source submarine groundwater discharges into the coastal ocean. *Limnol. Oceanogr.* 56, 1213–1224. doi:10.4319/lo.2011.56.4.1213
- Yang, D., Shen, L., 2010. Direct-simulation-based study of turbulent flow over various waving boundaries, *Journal of Fluid Mechanics.* doi:10.1017/S0022112009993557
- Yousuf, K., 2014. *Oman Daily Observer.*