FCTUC DEPARTAMENTO DE ENGENHARIA CIVIL
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Free-surface flow interface and air-entrainment modelling using OpenFOAM

Thesis Project in Hydraulic, Water Resources and Environment
Doctoral Program in Civil Engineering

Author

Pedro Miguel Borges Lopes

Supervisors

Jorge Leandro

Rita Fernandes de Carvalho

This project is the sole responsibility of its author, not having suffered corrections after the public defence trials. The Department of Civil Engineering FCTUC accepts no responsibility for the use of information presented

Coimbra, August, 2013

FCT
Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

imar

· U C ·
UNIVERSIDADE DE COIMBRA

RESUMO

A utilização de estruturas hidráulicas para controlo de cheias conhece uma longa história na área de infra-estruturas em engenharia civil. As estruturas hidráulicas submetidas a escoamentos fortemente turbulentos envolvem constantes trocas entre o ar e a água pela superfície livre. Estes fenómenos podem ser observados em diferentes tipos de estruturas hidráulicas como é o caso dos sumidouros, caixas de visita e descarregadores de cheias. Neste programa doutoral serão usados modelos numéricos de Computação Dinâmica de Fluidos para simular os escoamentos que ocorrem nestes dispositivos hidráulicos de controlo de cheias, e os resultados validados em instalações experimentais à escala real.

O desafio em numericamente simular o ar dentro da água é a principal motivação deste estudo. O enfoque recai na interacção do ar com a água, bem como na revisão dos modelos capazes de capturar esta interacção. O "solver" *interFoam* da "Toolbox" OpenFOAM^{TM} foi escolhido como ponto de partida deste estudo por ser "open-source" e amplamente utilizado na modelação destes fenómenos. O *interFoam* será estudado em detalhe e algumas simulações serão aqui reproduzidas.

Palavras-Chave: estruturas para controlo de cheias; escoamentos ar-água; emulsionamento de ar; interFoam; OpenFOAM^{TM}.

ABSTRACT

The use of hydraulic structures to control flooding has a history of long practice within civil engineering infrastructure. Hydraulic structures under turbulent flow conditions frequently involve free surface flow and interactions between air and water. This can be observed in different kinds of structures, e.g. gullies, manholes or stepped spillways. In this doctoral program, Computational Fluid Dynamics numerical models will be used to simulate flood control devices and the results validated using real scale physical models.

The challenge in numerical prediction of air mixed with water is the main motivation for this study. The focus of this work is primarily the air-water interaction and a revision of the numerical models able to capture it. The *interFoam* solver available in the OpenFOAM^{TM} Toolbox is chosen as the starting point of this study because it is open-source and widely used to numerically simulate such phenomena. This solver will be thoroughly investigated and some simulations involving air and water will be presented.

Keywords: flood control structures; air-water flow; air-entrainment; interFoam; OpenFOAM^{TM}.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

$\mathbf{A}$	Outward-pointing face area vector
A_s	Surface area
C_{air}	Air-concentration
Co	Courant number
C_S	Smagorinsky constant
$\mathbf{D}$	Orthogonal part of the face vector
$\mathbf{d}$	Vector between the computational point P and the neighbour N
dS	General surface area vector
F	Face flux; Source term of the momentum surface tension
f	Face; Point in the center of the face
$\mathbf{g}$	Gravitational acceleration
$\mathbf{H(u)}$	Transport and source part of momentum equation
h_C	Critical flow depth
k	Turbulent kinetic energy
L	Characteristic length
N	Point in the center of the neighbour cell
$\mathbf{n}$	Normal vector to the interface
n	Amount of faces of a control volume
P	Point in the center of the computational cell; Pressure
p	Kinematic pressure
p*	Modified pressure or Dynamic pressure
Q	Volume energy source; Flow
q	Heat flux
Re	Reynolds number
S_h	Height of the step
t	Time
U	Mean velocity
$\mathbf{u}$	Velocity field
$\mathbf{u}_r$	Relative velocity
$\bar{\mathbf{u}}$	Mean velocity field
V	Volume
x	Arbitrary point in the flow domain

α	Indicator function; Volume fraction
ε	Rate of viscous dissipation
Δ	Difference operator
Δt	Time step
κ	Interface Curvature
μ	Dynamic viscosity
μ_1	Dynamic viscosity of fluid 1
μ_2	Dynamic viscosity of fluid 2
μ_{SGS}	Dynamic SGS viscosity
η	Smallest length of turbulence scales
ρ	Density
ρ_1	Density of fluid 1
ρ_2	Density of fluid 2
σ	Surface tension coefficient
ω	Turbulence frequency
ϕ	General variable
∂V	Surface area control volume
Γ	Diffusivity

ACRONYMS

2D	Two-dimensional
3D	Three-dimensional
BC	Boundary Conditions
BD	Blended Differencing scheme
BIV	Bubble Image Velocimetry
CD	Central Differencing scheme
CFD	Computational Fluid Dynamics
CSF	Continuum Surface Force model
CVs	Control Volumes
DIC	Diagonal-based Incomplete Cholesky
DILU	Diagonal-based Incomplete Lower-Upper
DNS	Direct Numerical Simulation
FAVOR	Fractional Area-Volume Obstacle Representation
FVM	Finite Volume Method
IPP	Image Processing Procedure
LES	Large Eddie Simulation
LPT	Lagrange Particle Tracking
MAC	Marker-And-Cell
NVD	Normalised Variable Diagram
OpenFOAM	Open source Field Operation And Manipulation
PBE	Population Balanced Equation
PbiCG	Preconditioned Bi-Conjugate Gradient
PDEs	Partial Derivate Equations
PISO	Pressure Implicit with Splitting of Operators
PIV	Particle Image Velocimetry
PLIC	Piecewise Linear Interface Calculation
PCG	Preconditioned Conjugate Gradient
RANS	Reynolds Average Navier-Stokes
RAS	Reynolds Average Simulation
RNG	Re-Normalization Group
RSM	Reynolds Stress Model
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations

SLIC	Simple Line Interface Calculation
TVD	Total Variation Diminishing
UD	Upwind Differencing scheme
VOF	Volume Of Fluid

1. INTRODUCTION

1.1. General

Air and water are constantly interacting in diverse forms. In high-velocity open channel flows, if the turbulence level at the free-surface is large enough to overcome both surface tension and gravity forces, the air begins to entrain in the water. Simultaneously, large amounts of droplets are ejected from the water body. The resulting sharing process produces small air-bubbles, which after submerged, contribute to the increased oxygen content in the water flow.

In hydraulic structures such as spillways, the air in the flow is important, perhaps an indispensable design factor. The presence of air in water (1) increases the bulk of the flow thus influencing the height of the chute side walls (Falvey, 1980), (2) preventing the damage of the chute caused by cavitation (Bung and Schlenkhoff, 2010) (3) increasing the momentum when the air within the boundary layer reduces the shear stress and (4) re-oxygens the water flow which contributes to the downstream river quality and the preservation of aerobic species (Chanson, 1996).

Since the 1950's, several experimental studies have been conducted on the complexity of air-entrainment phenomena. The studies with more historical impact belong to Straub and Anderson (1958), Rajaratnam (1962), Bormann (1968), Volkart (1980a), Volkart and Rutschmann (1984), Wood (1991), Chanson (1996), Chanson and Toombes (2002), Gonzalez *et al.* (2008), Pothof (2011) or Kiger and Duncan (2012). Due to the increase of computational power and the development of Computational Fluid Dynamic (CFD) tools, the numerical studies have been replacing some experimental studies. Numerous works have been developed in the last decade using numerical tools to characterize the air-entrainment: Cheng *et al.* (2006), Tongkratoke *et al.* (2009), Lubin *et al.* (2011), Eghbalzadeh and Javan (2012), Deshpande *et al.* (2012) or Xiangju and Xuewei (2012).

The OpenFOAM^{TM} CFD Toolbox is a free, open-source software used in continuum-mechanics problems and written in C++ (Weller *et al.*, 1998). Several pre-built CFD solvers can be found in OpenFOAM^{TM} which have been used in wide range of problems from complex fluid flow involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetics. One of the strengths of OpenFOAM^{TM} is the ability to study the multiphase flows, achieved mainly through the solver *interFoam* (Ubbink, 1997). However, the *interFoam* solver is not able to accurately reproduce some characteristics of highly-aerated flows, such as air-entrainment or sharply surfaces (Lobosco *et al.*, 2011; Tøge, 2012). This is the primary focus of the proposed Project.

1.2. Objectives

Currently CFD multiphase solvers fail in the prediction of (1) sharp interfaces, (2) highly self-aerated flows and (3) air-entrainment phenomena. The main objective of this Thesis is to collect a review about the two-phase flows with special incidence in air-water flows, numerical techniques to predict the interface, the flow field characteristics, and study an existent CFD multiphase solver to suggest possible improvements. More specifically, the objectives are as follows:

1. Describing the existent numerical and experimental techniques that deal with air-water flows;
2. Conducting a literature review on the different techniques, experimental and numerical, to measure and predict the air-entrainment in flood control devices such as chute spillways, gullies and manholes;
3. Describing the *interFoam* multiphase solver; the mathematical formulation, the equations discretization and the source code in OpenFOAM^{TM};
4. Performing tests using *interFoam* solver coupled with some turbulence models, namely the standard k-ε, RNG k-ε, k-ω SST and LES Smagorinsky. These tests are useful to attest the capacity of *interFoam* in predicting air-water flows;
5. The above stated objectives are vital to achieve the Thesis purpose – Improvement of the *interFoam*-solver' numerical-simulation of air-concentration in hydraulic structures under turbulent conditions.

1.3. Thesis Structure

This Thesis project is divided into seven chapters, including the present introduction. The detailed description of the chapters contents are as following:

Chapter 1 introduces the topic of air-water flow and air-entrainment. In this Chapter the motivation, the objectives and the Thesis structure are presented.

Chapter 2 presents a literature review about air-water flow and the air-entrainment phenomena, the current numerical tools to accurately represent the free surface, topics about the existing numerical models for air-water flows and experimental techniques to measure the air on the water. Additionally in this Chapter, the *interFoam* solver within the OpenFOAM^{TM} toolbox is described together with a perspective of the numerical equations, its discretization and the solution procedure.

Chapter 3 describes the main structure of the *interFoam* code highlighting the most important lines within the solver.

Chapter 4 describes the tests performed with the *interFoam* coupled with different turbulence models and the comparative experimental tests. The results of these tests are also discussed.

Chapter 5 outlines the work plan for the next two years of the doctoral program and the motivation for the entire project behind this.

Chapter 6 presents a brief summary of the state-of-art and some final remarks on this Thesis project.

Chapter 7 presents a list of the work already published during the first year of the Doctoral Program. This work will be part of the final Thesis. Due to lack of space it was decided not to include it on this Thesis project.

2. LITERATURE REVIEW

2.1. Air-Water Flow and Air-entrainment

Air-water flow is characterized by the presence of two-phase fluids, water and air. They interact among each other with high complexity through the interface, often called as ‘free-surface’. In cases where the flow is highly turbulent, this can be sufficient to disrupt the free-surface and allow the air-entrainment into the water body. The entrained air changes the properties of the flow, mainly the density and compressibility and consequently, the turbulent structure of the flow (Carvalho, 2002).

One earliest description of the **air-entrainment phenomena** was made by Straub and Anderson (1958). They showed that the aeration of the flow begins in a region characterized by the appearance of white froth where the boundary layer reaches the water surface. Consequently, the aeration depends considerably on the turbulence intensity near the interface. One form of air-entrainment, described by Volkart (1980a), occurs after some ejected droplets fall into the flow disrupting the free surface and causing an entrainment of air in the form of bubbles. These flows where the phenomena of air-entrainment occur naturally are called **self-aerated flows**. In a turbulent and horizontal flow, Straub and Anderson (1958) divided the vertical structure of the water column in four distinct zones with different concentrations of air (Figure 2.1):

1. An **upper zone** where water is ejected from the main flow. Normally this region is neglected in engineering problems due to its small volume of water;
2. A **mixing zone** where surface waves exist with random amplitudes and frequencies;
3. An **underlying zone** where the air bubbles are mixed with the water flow. In this region the air concentration is measured by the volume of the bubbles;
4. An **air-free zone** where the air concentration is so small that it cannot be detected by less sensitive air concentration measuring equipment.

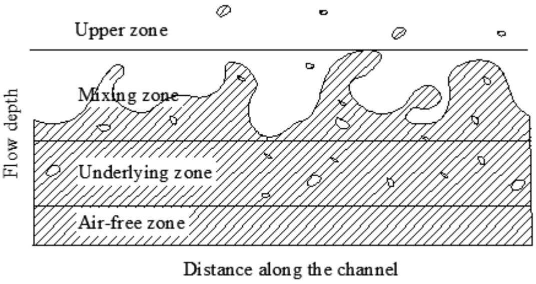


Figure 2.1 – Vertical structure of air-water flows.

Theoretical and experimental studies in self-aerated flows were performed mainly after the 1980's. Aerated flows were studied by Volkart (1980b) and Wood (1991) whereas Cain (1978), Wood *et al.* (1983) and Chanson and Toombes (2002) dedicated some studies to spillways with air-entrainment. Air-entrainment and air concentration measurements inside hydraulic jumps are presented by Rajaratnam (1962), Resch *et al.* (1974), Chanson and Brattberg (2000), Chanson (2007), Rodríguez-Rodríguez *et al.* (2011) or Leandro *et al.* (2012). Air-entrainment in vertical circular plunging jets was studied by McKeogh (1978), Cummings and Chanson (1997), Deswal and Verma (2007), Kendil *et al.* (2010) or Kiger and Duncan (2012). Páscoa *et al.* (2013) analysed qualitatively the air-entrainment in gullies during both drainage and surcharge conditions.

2.2. Experimental Air Measurement Techniques

Air concentration is a vital parameter to characterize the presence of air in the flow. The air concentration or void fraction is defined as the volume of air inside the volume of the mixture of water and air. The air concentration (C_{air}) can be represented as:

$$C_{air}=\frac{V_a}{V_a+V_w} \quad (2.1)$$

where V_a is the volume of air and V_w is the volume of water. Cain (1978) and Chanson (1988) define the free-surface where C_{air}=0.9. This value is linked to the high homogeneity of the air-water mixture for values lower than C_{air}=0.9. Above 90% the velocity of the air is no longer equal to the velocity of the water and the measurement of the air concentration becomes inaccurate.

The first attempt to measure the air concentration on the fluid body was made by Viparelli (1953) using a **modified Pitot tube**. Unfortunately, this method only shows good results in a zone with low void fraction values. Matos (1999) used a modified Pitot tube to characterize stepped spillways, and Carvalho (2002) used it to measure air concentration in the hydraulic jump. However, this method needs to know beforehand the flow direction which, in strong hydraulic jumps, is not always possible.

Another methodology to measure the void fraction is the **hot-film anemometry**. It has the advantage of barely being an intrusive device. Resch and Leutheusser (1972) measured the instantaneous velocity in the hydraulic jump. However, there are some difficulties in the signal interpretation and equipment calibration (Nagash, 1994). Resch et al. (1974) also used hot-film anemometry coupled with conical probes to obtain the void ratio and the bubble size in the hydraulic jump.

The traditional Particle Image Velocimetry (PIV) technique fails due to the reflection on the bubbles of the laser light. To overcome the difficulty, Ryu *et al.* (2005) proposed the **Bubble Image Velocimetry** (BIV) technique that uses the bubble particles as a tracker. The bubble velocity is measured by correlating the texture of the bubble images. This technique was applied in the measurement of mean velocity fields of plunging breaking wave impinging on structure. Leandro *et al.* (2012), followed the work of Mossa and Tolve (1998), proposing an improved **Image Processing Procedure** (IPP) to measure the instantaneous and averaged void fractions on hydraulic jump by analysing pixel intensity on images. This technique can provide measurement in different positions simultaneously, without any interference in the flow conditions. The results were compared with dual-tip conductivity probe measurements. Alike BIV, this approach is not able to measure the component along the axis perpendicularly to the camera image, thus the application in strongly 3D flows is compromised, since the image captured represents only a 2D plane.

The most common way to measure the air concentration on the flows is using **intrusive probes**. Several studies were made by resistive/conductive or optical fibre and single or dual-tip probes. The principle behind the optical fibre probes is the change in optical index between the two phases, while in the resistive/conductive probes is the difference between electrical resistivity in the water and air. The difference between the single tip and dual-tip probes is that the latter besides allowing the measurement of the void fraction, also measures the velocity of the bubbles by correlating the time periods in which the probe is in water or air.

Resistive/conductive probes were used by Rajaratnam (1962) to measure the air concentration in the hydraulic jump. Volkart (1980a) used a resistive probe to measure the air concentration in a transversal section of a partially full pipe with high longitudinal slopes, while Afshar *et al.* (1994) made measurements in the aerated zone of a stepped spillway with different slopes. Chanson (2002) and Chanson (2007) used single and dual-tip probes to study the air structure in the hydraulic jump. A similar study was made by Murzyn and Chanson (2008) using optical fibre dual-tip probes which compared their results with Chanson and Brattberg (2000). The main disadvantage of the standard probes is that the preferential direction of the flow needs to be known *a priori*. To overcome this difficulty, Borges *et al.* (2010) developed a new concept of conductivity probes combining three-holes pressure circuit and back-flushing. The probe can be set in two ways: (1) aligned with the flow so that the pressure in two symmetrically placed pressure holes is equal; or (2) collocated onto the flow and the different pressure values of the three holes, defines the angle between the probe and flow direction.

2.3. Numerical Techniques for Free-surface Flows

In most hydraulic problems the free surface is calculated by approximation due to the computational effort caused by the complete resolution of the tri-dimensional Navier-Stokes equations. In case of open channel flows, the equations are integrated at water depth which results in an indirect prediction of the free surface, e.g. Saint-Venant or Boussinesq equations. The weakness is that each spatial coordinate corresponds to only one depth. Consequently, in case of complex phenomena, the solution of the complete Navier-Stokes is indispensable and approximation techniques to capture or track the interface are necessary.

Several methodologies to predict the free-surface using static meshes or dynamic meshes can be distinguished. In case of static meshes, the grid is static and the interface is followed or captured. In this section two different methodologies to predict the free surface will be followed (Ubbink, 1997): **Surface Methods** and **Volume Methods**. Additionally, the air-entrainment needs to be modelled using **additional techniques**.

2.3.1. Surface Methods

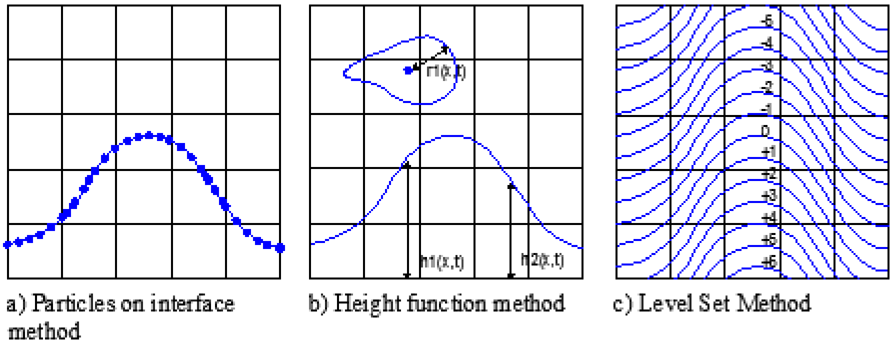
The Surface Methods treat the free surface either by a **sharp interface**, whose position is followed, or tracked by marking it with special points, also known as **marker points** (Ubbink, 1997). Between those marker points, the free surface is described by a polynomial function. The accuracy of the surface tracking methods depends strongly on the stability and precision of the interpolation method (Hyman, 1984). There are many ways to mark the interface (Ubbink, 1997):

a) Particles on interface method: This method was presented by Daly (1969) where the interface is tracked explicitly by a set of connected massless marker particles on a fixed grid (Figure 2.2a). In cases where the particles are distant from each other, the interface may not be well represented.

b) Height function method: In this method the interface is tracked introducing a height function that returns the distance of the point on the interface and the reference plane (Figure 2.2b). For closed interfaces such as bubbles or droplets, one defines a representative point inside the object and the radius at different angular positions is set as the distance function.

The major difficulty of this method is that each coordinate of the reference plane is associated to only one interface value. Consequently, in case of breaking waves, this model fails.

c) Level Set method: This method was originally proposed by Osher and Sethian (1988) introducing a continuous function, known as a level set function, over all computational domain (Figure 2.2c). The function is positive in one fluid phase and negative in the other. The zero level represents the exact position of the free surface.



a) Particles on interface method

b) Height function method

c) Level Set Method

Figure 2.2 – Surface methods to treat the interface. Adapted from Galambos (2012).

2.3.2. Volume Methods

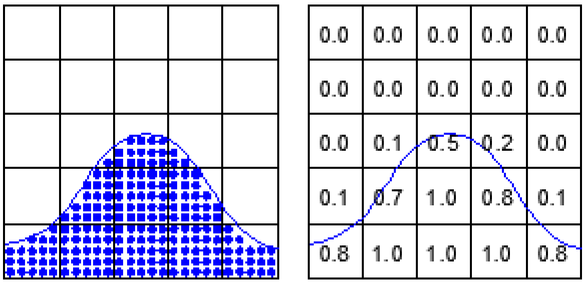
In Volume methods the free surface is defined by a boundary of volume. The entire domain is marked by **massless particles** or by an **indicator function**. Unlike the surface methods, in the volume methods the exact position of the interface is not known and, special techniques part of the solution algorithm, need to be applied to capture the interface (Ubbink, 1997). On the other hand, they can simply and accurately account for the interactions smooth varying interfaces (Hyman, 1984). Two important techniques have been developed:

a) Particles on fluid method: One of the earliest volume methods of particles on fluid for material interfaces is the Marker-And-Cell (MAC) method of Harlow and Welch (1965). The location of fluid within the fixed grid is determined by a set of massless marker particles that move with the fluid. Cells full of marker particles are filled of fluid and cells with no marker particles are consequently empty. Hence, cells with marker particles which are adjacent to at least one empty cell, are interface cells (Figure 2.3a). The success of this method is recognized mainly due to the fact that the markers do not track surfaces directly, but track fluid volumes instead. Consequently, the surfaces are merely the boundaries of the volumes.

b) Volume fraction methods: The volume fraction methods are one of the most common methods to treat the free surface. In a short period between the 70's and 80's, three important volume fraction methods were established: the DeBar's method (DeBar, 1974), the SLIC method (Noh and Woodward, 1976) and the Hirt and Nichols' VOF method (Hirt and

Nichols, 1981). All of them use a scalar indicator function, also known by volume fraction function, that ranges from zero (no material) to one (completely filled with material) to distinguish the presence of phase fluid (Figure 2.3b). Those methods present clear advantages regarding the MAC in matters of computational economy and variable storage, as only one value is stored (the value of the fraction of volume), instead of the coordinates of the marked particles.

The VOF method has the advantage of the volume occupied by one fluid not being able to be occupied by the other and thus the continuity is always verified. The flow properties (i.e. density and viscosity) are a weighted mixture of the properties of both phases. The main downside of the VOF technique is that in a numerical simulation with large grid sizes, the formation of small bubbles or droplets, smaller than the minimum grid size, is ignored, thus limiting the method.



0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.1	0.5	0.2	0.0
0.1	0.7	1.0	0.8	0.1
0.8	1.0	1.0	1.0	0.8

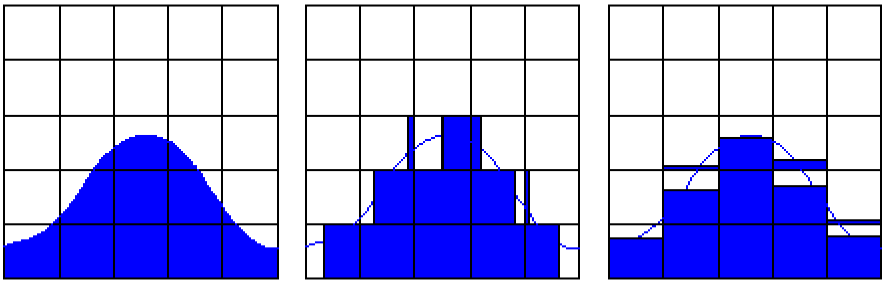
a) Particles on fluid - Marker And Cell (MAC)

b) Volume Fraction method

Figure 2.3 **-** *Volume methods to treat the interface. Adapted from Galambos (2012).*

The precision of the VOF models is mainly dependent on two topics: the **interface reconstruction** and the **advection technic** (Ubbink, 1997).

b.1) Interface Reconstruction: The simplest type of volume fraction method is the Simple Line Interface Calculation (SLIC) of Noh and Woodward (1976). It approximates the interface in each cell as **piecewise constant**, i.e. the interface is a line (or a plane in 3D domain) parallel to one of the coordinate axes. In two-dimensional cases this assumption results in two different situations: (1) the x-sweep, where the interface approximation uses the volume fraction values on the left and the right of the cell (Figure 2.4b) and (2) the y-sweep, which uses the values above and under the cell (Figure 2.4c).

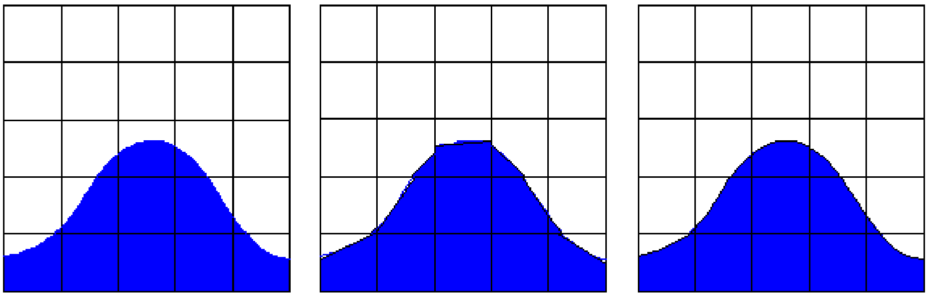


a) Accurate distibution b) SLIC x-sweep c) SLIC y-sweep

Figure 2.4 – Volume fraction method SLIC. Adapted from Ubbink (1997).

The most popular Volume-of-Fluid (VOF) method is the well-established Hirt and Nichols' VOF method (Hirt and Nichols, 1981) which uses a **piecewise constant/stair-stepped** interface reconstruction method. It forces the interface to align with mesh coordinates, but additionally allows to "stair-step", i.e. aligning with other mesh coordinates, depending upon the local distribution of the volume of fluid (Rider and Kothe, 1997).

In modern volume tracking methods, the **piecewise linear** reconstruction or **PLIC** (Piecewise Linear Interface Calculation) methods are, in most cases, preferable. This "family" of piecewise methods was introduced by Youngs (1984) (Figure 2.5b). The method positioned each reconstructed interface line within the volume fraction of the neighbouring cells. An improved PLIC method, named FLAIR was proposed by Ashgriz and Poo (1991) by constructing line-segments on the cell faces (Figure 2.5c). Another improved PLIC method is well implemented in the 2D VOF-FAVOR model of Carvalho (2002).



a) Accurate distibution b) PLIC (Youngs, 1984) c) PLIC (Ashgriz and Poo, 1991)

Figure 2.5 - Volume fraction method PLIC. Adapted from Ubbink (1997).

b.2) Advection technique: The advection volume method defines the quantity of volume transported in time for the adjacent cells. The donor-acceptor formulation from Hirt and Nichols (1981) uses the volume fraction value of the upwind/downwind cell (donor/acceptor cell) to predict the level of volume fraction transported through it during a time step. The donor/acceptor cell depends on the volume fraction and free surface slope.

The introduction of high-resolution schemes allows the discretization of the scalar transport equation with high-order difference schemes. Several techniques can be found in the literature such as the Total Variation Diminishing (TVD) or Normalized Variable Diagram (NVD).

In OpenFOAM^{TM}, the VOF model was implemented by Ubbink (1997) within the *interFoam* solver. There are numerous numerical applications of *interFoam* concerning air/water interactions in the literature. Experimental and numerical studies of horizontal jets below a free surface was particularized by Trujillo *et al.* (2007). Lobosco *et al.* (2011) tested the *interFoam* in self-aeration regions of stepped spillways. The Authors correctly reproduced the entrapped-air but found some issues with the air-entrainment simulations. Deshpande *et al.* (2012) applied the *interFoam* to horizontal jets plunging into a pool and compared it with experimental data. The average vertical velocity profile along the experiment shows satisfactory results. The solver shows to be accurate, regarding to the surface curvature, even in modest grid resolution; excellent mass conservation; and acceptable advection errors. As expected the numerical simulations did not reproduce the smallest elements of droplets or bubbles.

2.3.3. Air-Entrainment Modelling

In case of flows with air-entrainment the surface and volume methods are insufficient. Two different techniques can be used to modelling the air-entrainment, dependent on the percentage of air in cause.

For lower values of air entrainment, the water fluid is treated as a continuous phase in which the Navier-Stokes equations are solved. The air is the discrete phase whose localization is obtained by calculating the movement of the sparse particles in the fluid. Enright *et al.* (2002) proposed a method that uses a surface Level-Set method and a particle marker Lagrangian Scheme that was tested on a circular body subjected to a three dimensional deformation field. Grosshans (2011) developed their own VOF-LPT (VOF-Lagrangian Particle Tracking) model and applied it to dilute spray regimes. Vallier *et al.* (2011) developed and implemented in OpenFOAM^{TM}, the method VOF-LPT to identify the behaviour of an air bubble breaking up under the impact of a water jet. The source code can be found in the literature but it is not a part of the standard package of OpenFOAM^{TM}. These methods are accurate when simulating a very small number of bubbles; therefore, the tests found throughout literature were performed with only one discrete particle. In a large domain those methods are not suitable to be used, since a large number of particles exist, thus increasing the computational time and required memory.

Dealing with large amounts of air entrainment, the models need to be more complex. The multicomponent models solve the two fluids (air and water) separately and the interaction is made using buoyancy effects. The drag and lift forces are taken into account in these models (Silva, 2008). The weakness of these models is the large amount of calculation and the hard convergence. Moraga *et al.* (2005) performed numerical studies about air-entrainment by simulating the wave breaking with a Two-fluid 3D subgrid model. The model predicts the air-entrainment in all the regions observed at sea, namely the breaking blow wave. The most suitable solvers to predict the air-entrainment, presented in the standard package of OpenFOAM^{TM} are the *twoPhaseEulerFoam* (Rusche, 2002) and the *bubbleFoam*. In case of multi-phase flow simulations (i.e. simulation of two or more phases), the OpenFOAM^{TM} has *multiphaseEulerFoam* (Silva and Lage, 2007; Silva *et al.*, 2008). This last model, besides simulating multi-phase flow, also couples with the Population Balanced Equation (PBE), which includes the effects of particle-particle interaction, the breakage and the aggregation. The simulation of dispersed phases uses drag and virtual mass models, whereas the resolved phases use the interface compression and surface-tension models of the VOF method (OpenFOAM, 2011).

The two different scenarios described before were implemented in a unique model in FLOW-3D$^{®}$ by Hirt (2003). Since it is a commercial model, the information regarding the solved equations and the implemented methodology, is restricted. Nevertheless, theoretical description choices presented by the model is given in (Hirt, 2003). For volume fractions of relatively lower entrained air, the model uses a scalar value which records the air volume fraction inside the fluid. In this model the air entrained does not interact with the water and does not change the dynamics of the flow. For larger values of void fraction, the model has a second option that considers a density variable. The addition of air (ΔV) is allowed in the model and the buoyancy effects are taken into account. The additional volume of air is calculated by:

$$\Delta V = C_{air} A_s \sqrt{2\frac{P_t - P_d}{\rho}} \tag{2.2}$$

where C_{air} is the air concentration, A_s the surface area, Pt the turbulent kinetic energy per unit volume (Pt=ρk), k the turbulent kinetic energy, Pd the surface tension energy ($Pd=\rho g_n L_t+\sigma/L$), L the characteristic length of turbulence eddied, σ the surface tension and g_n the component of gravity normal to the free surface.

2.4. Flood Control in Hydraulic Structures

From the large amount of studies for air-entrainment, the flood control structures are of special importance. The hydraulic structures studied in this work are the spillways, gullies and manholes.

The study of the hydraulic behaviour of some components of urban drainage systems is important in case of flooding in order to predict affected areas. During a flood event, while the sewer systems do not reach their full capacity, the gullies and manholes (linking-elements) are working under "normal" conditions. After pipes surcharging, the gullies and the manholes start working in reverse conditions, performing contrarily to the conditions that they are originally conceived for. In reverse conditions the pressurized flow through the linking-elements reaches the urban surface with high velocity and can eventually originate "urban geysers" (Lopes *et al.*, 2012).

The stepped spillways are widely used to release water from reservoirs if the inflows are higher than the storage capacity of the dams. The flow over spillway structures is mostly turbulent and self-aerated having been intensely studied.

2.4.1. Spillways

In the traditional smooth spillway structure it is possible to identify different flow regimes in its longitudinal direction and the evolution of the boundary layer (Figure 2.6). Bormann (1968) identified three distinct zones:

1. Zone without air where the turbulent boundary layer has not reached the water surface;
2. Zone where the air entrainment is developed but the air concentration profiles are not constant in depth;
3. Zone of fully developed air entrainment with constant air concentration profile.

These zones were subject of modification by Keller *et al.* (1974) which divided the Bormann's Zone 2 into two sections. The first, where the air entrainment is developing but the air has not yet reached the bottom of the channel - *Partially Aerated Section* - and a second, where the air already reached the bottom of the channel but the air concentration profile remains variable with depth - *Fully Aerated Section.* The transition point or "critical point" which divides the zones, is studied by Keller and Rastogi (1977). This point is used by engineers in the prediction of the self-aeration zone, thus rendering it of large importance.

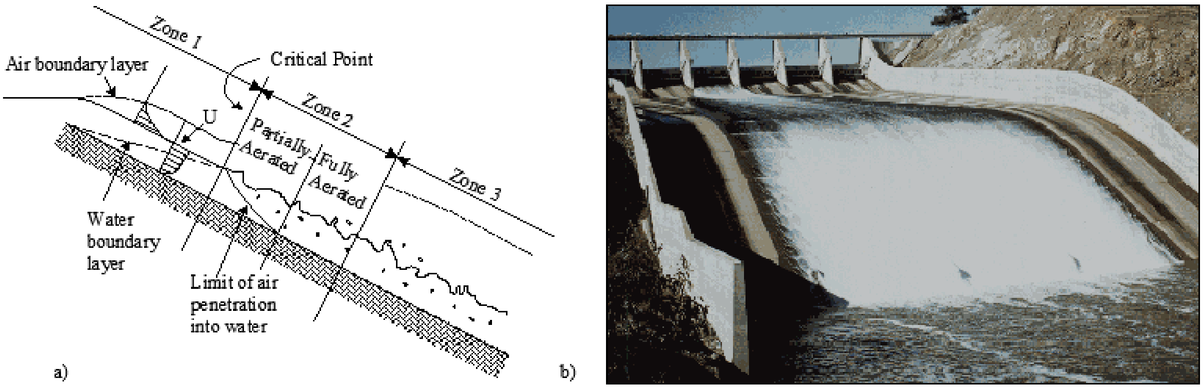


Figure 2.6 – (a) Longitudinal structure of the flow over a spillway (Adapted from Falvey (1980)) (b) Photograph of Burrendong Dam spillway (Australia) showing fully lined chute and full energy dissipator (Retrieved from http://members.optusnet.com.au/~engineeringgeologist/page21.html)

The research interest in stepped spillways hydraulics increased during the last decades. The stepped spillways design, comparatively to the smooth version, increases the rate of energy dissipation and reduces the size of the downstream energy dissipator structure. Hence, the velocity is lower than in the other versions of spillways. The flow regimes, the height of the steps and the desired flow rates are subject of adjustment in the design phase.

Regarding the flow regime, there are two different cases, dependent of the discharge and step geometry: the **napped flow** and the **skimming flow**. The napped flow is characteristic of low discharges with high and large steps. In this case, the water plunges from one step directly to the other resembling a series of cascades. In the skimming flow, the main body of the water, skims over the steps, forming a "pseudo-bottom". Between the steps and below the pseudo-bottom recirculating vortexes are formed. The transition from nape to skimming flow can be expressed through the ratio between the critical flow depth (h_C) and the height of the step (S_h). Rajaratnam (1990) proposes the occurrence of skimming flows for $h_C/S_h>0.8$. Chanson (2006) defines limits for both regimes. For the napped flow, the limit is $h_C/S_h=0.89-0.4(S_h/l_S)$ and for the skimming flow the limit is $h_C/S_h=1.2-0.325(S_h/l_S)$.´

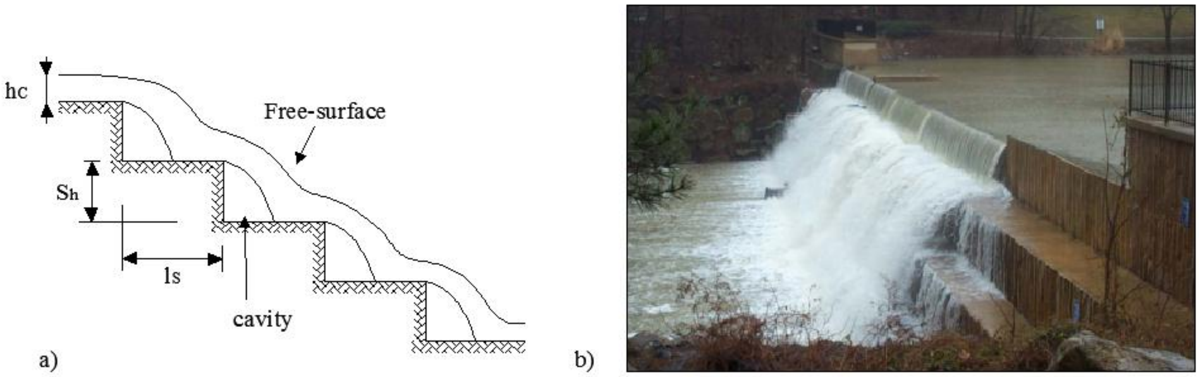


Figure 2.7- (a) Schematic longitudinal profile of napped flow (b) and photograph from Lake Wilde dam spillway (Maryland, USA) (Retrieved from Gonzalez *et al.* (2005)*).*

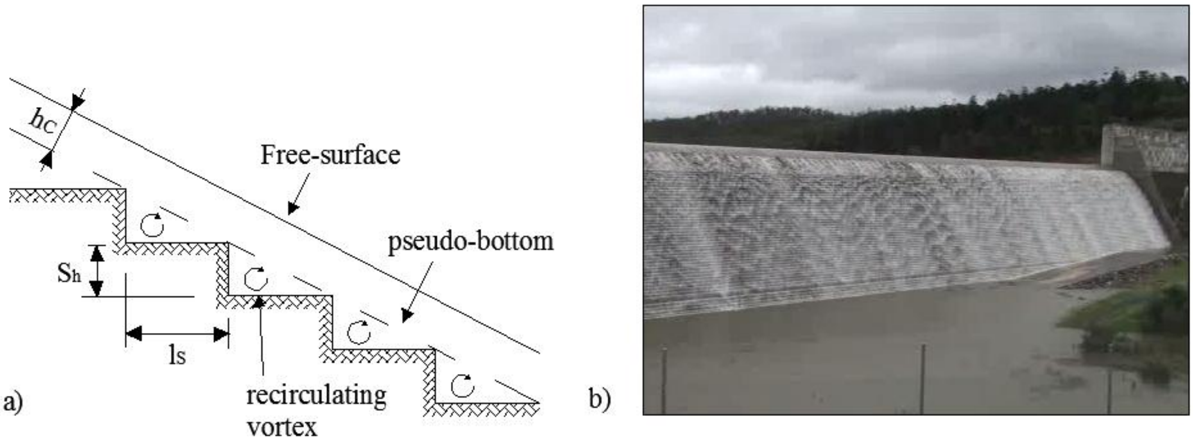


Figure 2.8- (a) Schematic longitudinal profile of skimming flow and (b) photograph from Paradise Dam stepped spillway at 5.30pm 2nd march 2010 (Retrieved from http://rogercurrie.wordpress.com/paradise-dam-flood/).

Some numerical simulations of skimming flows over stepped spillways can be found in the literature. Cheng *et al.* (2006) adopted a mixture model with RNG k-ε turbulence approach from ANSYS FLUENT software to compare the velocities and pressure profiles and the air-entrainment in some steps. The Authors conclude that the mixture model successfully simulates the interactions between entrained air bubbles and cavity recirculation in the skimming flow regime. Tongkratoke *et al.* (2009) tested different multiphase models (VOF, Eulerian and Mixture) and different turbulence approaches (standard k-ε, RNG, realizable k-ε and LES Smagorinsky) to compare the velocities and the air concentration over the step. They found more accurate results to the experience with the k-ε combined with the wall function. The domain contains only one step. This assumption can explain the discrepancies between the results since the influence of the flow at upstream and downstream are unsolved. Lobosco *et al.* (2011) solved a skimming flow using the *interFoam* solver of OpenFOAM^{TM} toolbox employing the standard k-ε turbulence model. The Authors found some difficulties in reproducing a realistic interface and the breaking configuration. Some bubbles and droplets were found, but their size is largely dependent on the mesh refinement. Simões *et al.* (2012) numerically studied the stepped spillway using ANSYS CFX®. The Authors concluded that the two-phase flow characteristics had still not been reproduced by the numerical model.

New alternatives have been developed to increase both the energy dissipation and the oxygen transfer potential. The addition of micro-roughness on the horizontal step surfaces in skimming flows was subject of study by many researchers. Gonzalez *et al.* (2005) observed that step roughness affect the recirculation patterns in the step cavities. In smooth stepped, the recirculation zones are more aerated than in the rough version. Bung and Schlenkhoff (2010) concluded that the micro-roughness cannot increase the aeration of the flow, however, the flow characteristics are largely influenced by the arrangement of the micro-elements.

2.4.2. Gullies and Manholes

To the authors' knowledge, there are only two experimental installations to study gullies and manholes. One is installed in the University of Sheffield and the other located at the University of Coimbra. The second installation differs from the first one by completely simulating the Urban Drainage System.

Gómez and Russo (2009) investigated the efficiency of transversal gullies with grates. Carvalho *et al.* (2011) used a VOF/FAVOR model to study the influence of different outlet locations of a 2D gully under drainage conditions. This study was extended to the case of surcharge conditions and compared with OpenFOAM™ simulations (Carvalho *et al.*, 2012). Djordjević *et al.* (2011) presented numerical and experimental investigations of interactions between surface flow and the flow through an gully. Numerical and experimental studies on 3D gullies under drainage and surcharge were conducted by Martins *et al.* (2012) and Lopes *et al.* (2012), respectively using the OpenFOAM™. Romagnoli *et al.* (2013) experimentally characterized the turbulence in a gully with reverse flow. Páscoa *et al.* (2013) measured the velocity fields inside the gully and characterized the air entrainment on gullies under different operating conditions. Lopes *et al.* (2013) used the experimental results from Páscoa *et al.* (2013) to validate an numerical model and later derived numerical pressure-flow and height-flow relations for a gully under surcharge conditions. These results are useful to calibrate and validate the linking elements found in the Dual Drainage (DD) models.

The existing studies in manholes are in surcharge conditions. Stovin *et al.* (2008) compared the experimental velocity distributions in a 2D plan closer to the ground manhole with numerical results obtained with FLUENT. Zhao *et al.* (2008) simulated a quadratic manhole combined laterally with a 90º inflow junction. The Authors measured vertical velocity profiles inside the manhole and compared it with numerical results obtained using ANSYS CFX. Descriptive analysis about air-entrainment was also conducted.

2.5. InterFoam Solver

The hydraulic structures presented previously are characterized by a high interaction between air and water. The *interFoam* solver of OpenFOAM^{TM} Toolbox is a multiphase solver able to reproduce some of those characteristics. In this Section, the mathematical formulation and equation discretization of *interFoam* solver are explored to better understand the source code. This Section follows closely the works of Jasak (1996), Ubbink (1997) and Rusche (2002).

2.5.1. Mathematical Formulation

2.5.1.1. Continuity and Momentum Equations

The fluid movement is governed by a set of equations expressing the conservation of mass, momentum and energy. In a 3D system, the governing equations of fluid continuum mechanics can be written in the differential form as (Aris, 1989):

Conservation of mass:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \qquad (2.3)$$

Conservation of momentum:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = \rho \mathbf{g} + \nabla \cdot \sigma \qquad (2.4)$$

Conservation of energy:

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) = \rho \mathbf{g}\mathbf{u} + \nabla \cdot (\sigma \mathbf{u}) - \nabla \cdot \mathbf{q} + \rho Q \qquad (2.5)$$

where ρ is the fluid density, **u** is the three-dimensional velocity field, σ is the shear stress tensor, e is the total specific energy, Q is the volume energy source, **q** is the heat flux and **g** is the gravity acceleration vector.

This system of three equations is indeterminate since the number of unknown variables is larger than the number of equations. Consequently, it is necessary to include a set of constitutive relations which can be consulted in Jasak (1996) Thesis. For Newtonian, incompressible (ρ=constant) and isothermal fluid, the system (2.3), (2.4) and (2.5) can be simplified in the form:

$$\nabla \cdot \mathbf{u} = 0 \qquad (2.6)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) = \mathbf{g} - \nabla p + \nabla \cdot (\upsilon \nabla \mathbf{u}) \tag{2.7}$$

where υ is the kinematic viscosity and p the kinematic pressure. Multiplying the momentum equation by the density of the fluid, the final form of the continuity and momentum equations for a single field of fluid becomes:

$$\nabla \cdot \mathbf{u} = 0 \tag{2.8}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{uu}) = -\nabla P + \nabla \cdot \tau + \rho \mathbf{g} + F \tag{2.9}$$

where P is the pressure (P=p×ρ), τ is the viscosity stress tensor and F represents the source of the momentum in regard to the surface tension (Rusche, 2002):

$$F = \int_{S(t)} \sigma \kappa' \mathbf{n}' \delta(\mathbf{x}\text{-}\mathbf{x}') dS \tag{2.10}$$

In the equation above, σ represents the surface tension coefficient, κ denotes the curvature and **n** is the normal vector of the interface. The viscous stress term can be reformulated to obtain more efficiency. The final form of this term is as follows:

$$\nabla \cdot \tau = \nabla \cdot (\mu[\nabla \mathbf{u} + (\nabla \mathbf{u})^T]) = \nabla \cdot (\mu \nabla \mathbf{u}) + (\nabla \mathbf{u}) \cdot \nabla \mu \tag{2.11}$$

The modified pressure p* (p_rgh in OpenFOAM^{TM} code) is adopted in *interFoam* removing the hydrostatic pressure ($\rho \mathbf{g} \cdot \mathbf{x}$) from the pressure P. This is advantageous for the specification of pressure at the boundaries of the space domain (Rusche, 2002). The gradient of the modified pressure is defined as:

$$\begin{aligned} \nabla p^* &= \nabla P - \nabla(\rho \mathbf{g} \cdot \mathbf{x}) \\ &= \nabla P - \rho \mathbf{g} - \mathbf{g} \cdot \mathbf{x} \nabla \rho \end{aligned} \tag{2.12}$$

2.5.1.2. Indicator Function (VOF model)

In the *interFoam* solver, the conventional VOF method presented by Hirt and Nichols (1981) is applied. As mentioned in Section 2.3.2, it uses the volume fraction α as an **indicator function** (alpha in OpenFOAM^{TM} code) to define which portion of the cell is occupied by the fluid, as mentioned in (2.13).

$$\alpha(x,y,z,t) = \begin{cases} 1 & \text{for a place (x,y,z,t) occupied by the fluid 1} \\ 0 < \alpha < 1 & \text{for a place (x,y,z,t) in the interface} \\ 0 & \text{for a place (x,y,z,t) occupied by the fluid 2} \end{cases} \qquad (2.13)$$

The transport of α in time is expressed by an advection function:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) = 0 \qquad (2.14)$$

The previewed equation shows that in an incompressible fluid, the conservation of mass is equivalent to the conservation of volume and, consequently, conservation of the function α (Černe *et al.*, 2001) is observed. The local fluid properties (ρ and μ) are a weight mixture of the physical properties of both fluids. The subscripts 1 and 2 denote different fluids.

$$\rho = \alpha\rho_1 + (1\text{-}\alpha)\rho_2 \qquad (2.15)$$

$$\mu = \alpha\mu_1 + (1\text{-}\alpha)\mu_2 \qquad (2.16)$$

The conservation of the phase fraction is essential, particularly in the case of high density fluids, where small errors on the volume fraction generate significant errors on the physical properties. Function (2.14) opposes the previous statement (Rusche, 2002) and, to respond to this issue, many researchers have been presenting alternative techniques to overcome this problem (Ubbink, 1997; Ubbink and Issa, 1999). The best alternative was formulated by Weller (2002), introducing an extra term in the phase fraction function – the **artificial compression term**.

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \bar{\mathbf{u}}) + \underbrace{\nabla \cdot [\mathbf{u}_r \alpha (1-\alpha)]}_{\text{Artifial compression term}} = 0 \qquad (2.17)$$

where $\mathbf{u}_r = \mathbf{u}_1 \text{-} \mathbf{u}_2$ is the vector of relative velocity between the two fluids, also called as compression velocity (Berberović *et al.*, 2009) and $\bar{\mathbf{u}}$ is the mean velocity, calculated by a weighted average of the velocity between the two fluids:

$$\bar{\mathbf{u}} = \alpha \mathbf{u}_1 + (1\text{-}\alpha)\mathbf{u}_2 \qquad (2.18)$$

2.5.1.3. Surface Tension Force

The surface tension force acts on the interface between the two phases. In the interface-capturing methodology, the interface is not tracked explicitly and consequently its exact form and location are unknown (Rusche, 2002). Then, the source term (F) of the momentum equation (2.9) relative to the surface tension, cannot be solved directly. The Continuum

Surface Force (CSF) model developed by Brackbill *et al.* (1991) overcomes this problem, converting the F term into a volume force function of the surface tension. In this CSF model, the surface curvature (κ) is formulated from local gradients in the surface normal (**n**) at the interface, which is a function of the phase fraction (**n**=∇α) (Tang and Wrobel, 2005):

$$\kappa=\nabla\cdot\hat{\mathbf{n}}=\nabla\cdot\frac{\mathbf{n}}{|\mathbf{n}|}=\nabla\cdot\left(\frac{\nabla\alpha}{|\nabla\alpha|}\right) \tag{2.19}$$

The volumetric surface tension force (**F**) is written in terms of the surface tension, and subsequently, to the jump pressure across the interface.

$$\mathbf{F}=\sigma\kappa\frac{\rho}{0.5(\rho_1+\rho_2)}\nabla\alpha \;\approx\; \sigma\kappa\nabla\alpha \tag{2.20}$$

Taking into account the volumetric form of surface tension (2.20), the viscous stress term (2.11) and the modified pressure (2.12), the **final form of the momentum equation** is:

$$\frac{\partial\rho\mathbf{u}}{\partial t}+\nabla\cdot(\rho\mathbf{u}\mathbf{u})-\nabla\cdot(\mu\nabla\mathbf{u})=-\nabla p*+(\nabla\mathbf{u})\cdot\nabla\mu-\mathbf{g}\cdot\mathbf{x}\nabla\rho+\sigma\kappa\nabla\alpha \tag{2.21}$$

To conclude, the final form of the mathematical model using VOF concept is constituted by the continuity equation (2.6), the modified indicator function (2.17) and the momentum equation (2.21). These equations are solved together with the constitutive relations for density and dynamic viscosity given by (2.15) and (2.16).

2.5.1.4. Turbulence Modelling

Kolmogorov (1941) defined the process of transference of energy from the largest scales (productive scales) to the smallest (viscous or dissipative scales), described through the **turbulence spectrum of energy**. The Kolmogorov scales are given for:

$$\eta=\left(\frac{\nu^3}{\varepsilon}\right)^{1/4} \tag{2.22}$$

$$\varepsilon=\frac{U^3}{L} \tag{2.23}$$

Where η is the smallest length of turbulence scales, U is the mean velocity and L is the characteristic length or the domain length. It is necessary for $N=L/\eta$ cells to capture all the scales in one direction, i.e. $N=Re^{3/4}$. Extrapolating for a 3D domain, the number of cells necessary to capture all scales is $N_{3D}=\left(Re^{3/4}\right)^3=Re^{9/4}$, which for Reynolds numbers around the 10^5, results in $1.8x10^{11}$ cells.

The methodology to solve the complete Navier Stokes Equations and all the spectrum of energy is denominated as Direct Numerical Simulation (DNS) (Moin and Mahesh, 1998; Pope, 2000). This methodology is associated to a fine mesh and large computational effort, it is employable in cases of small domains and low Reynolds numbers.

The RAS methodology ("Reynolds Average Simulation") or RANS ("Reynolds Average Navier-Stokes") solves the time-averaged Navier-Stokes using the Reynolds equations. Attention is focused on the mean flow and the effects of turbulence on mean flow properties. The RANS turbulence model is classified through the number of the additional number of transport equations, which need to be solved with the RANS flow equations: zero-equation model (Mixing-length), one-equation model, two-equations model (k-ε, k-ω, SST k-ω) and seven-equations (RSM - Reynolds Stress Model). The computer requirements to use these models are modest, so RANS are largely used on most of the CFD problems.

The standard k-ε model (Launder and Spalding, 1974) is based on two equations; one for k and another for ε, that represent the turbulent kinetic energy and the rate of viscous dissipation respectively. At high Reynolds number, the standard k-ε needs to integrate wall turbulence functions (Versteeg and Malalasekera, 1995). Yakhot *et al.* (1991) improved the standard k-ε removing the small scales of motion from the governing equations and represent their effects in the large scales. This process is called as Re-Normalization Group (RNG). Other alternative to the k-ε is the k-ω proposed by Wilcox (1988, 2008) introducing the turbulence frequency $\omega = \varepsilon/k$ as second variable. Menter (1993) suggested a model between the k-ε and k-ω. The k-ω model is implemented in zones near the walls, while the k-ε is used in the fully turbulent region of the flow. Those modifications are included on the SST k-ω model.

An intermediate methodology to capture the turbulence is the LES (Large Eddy Simulation). The effects of turbulence on the mean flow and the large eddies are completely resolved, while small eddies are included on the solutions by means of a sub-grid scale model. To separate those two groups of eddies, LES uses a spatial filtering operation. The computational resources on terms of memory are large, however, the recent developments on parallel processing extend their application.

In the Standard Smagorinsky Model (Smagorinsky, 1963) the effects of smallest turbulent eddies (unsolved part) on the large eddies are descripted using the Boussinesq hypothesis. This model employs a constant of proportionality μ_{SGS} (dynamic SGS viscosity), calculated using an ad-hoc constant, the Smagorinsky constant (C_S).

2.5.2. Finite Volume Method

There is no analytical solution for the complete Navier-Stokes equations. Thus, the transport equations derived in the previous section requires a numerical model and a computational mesh on which each term of the partial differential equations is subsequently solved. This process is known as ‘discretization’. This section describes the discretization of the governing equations using the Finite Volume Method (FVM). This process can be extensively found in the bibliography (Versteeg and Malalasekera, 1995; Jasak, 1996; Ubbink, 1997; Rusche, 2002). The FVM of solution is subdivided in two components: space and time domains (Rusche, 2002).

2.5.2.1. Discretization of the General Transport Equation

The FVM discretization of space requires a subdivision of the solution domain into a finite number of small Control Volumes (CVs) (Ferziger and Peric, 2002). The control volumes are cells bounded by a set of flat faces. Each face is shared by two cells – the *control volume cell* (or *owner cell*) and the *neighbouring cell.*

An example of an owner cell is shown in Figure 2.9. The point P is the centroid of the computational cell and N is the centroid of the neighbouring cell. Those cells have the internal face *f* in common, having the normal vector represented by **A**. The vector **A** points always outwards from the computational cell, with magnitude equal to the area of the face *f*. The vector **d** connects the point P to N while the vector **D** is the vector with the same direction of **d** but magnitude able to satisfy the conditions proposed by Jasak (1996):

$$\mathbf{A}=\mathbf{D}+\mathbf{k} \tag{2.24}$$

$$\mathbf{D}=\frac{\mathbf{d}|\mathbf{A}|^2}{\mathbf{d}\cdot\mathbf{A}} \tag{2.25}$$

In orthogonal meshes, the angle between **A** and **d** is zero and the vectors **D** and **k** are omitted (Ubbink, 1997).

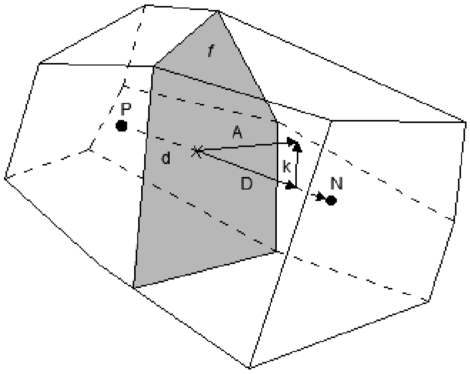


Figure 2.9 – Control Volume and parameters of the discretization of the solution domain. P and N are the centroid of two neighbouring cells, ***d*** *is the vector between P and N and* ***A*** *the vector normal to the face f common to both cells (addapted from Ubbink, 1997).*

The FVM discretization of the governing equations is completed in a few steps. Firstly, the equations are written as volume integrals over each CV and later converted to surface flux terms using Gauss's theorem. Then, the surface integrals are calculated by a sum of fluxes over all CVs faces. Finally, to determine the fluxes, cell face values of variables are estimated by interpolation using cell centred values at neighbouring cells.

The following description summarizes the finite volume discretization for a generic transport equation using a generic scalar ϕ. The genic transport equation represents any conservation law.

$$\underbrace{\frac{\partial \rho\phi}{\partial t}}_{\substack{\text{temporal}\\ \text{derivate}}} + \underbrace{\nabla\cdot(\rho\mathbf{u}\phi)}_{\substack{\text{advection}\\ \text{term}}} - \underbrace{\nabla\cdot\left(\rho\Gamma_\phi\nabla\phi\right)}_{\substack{\text{diffusion}\\ \text{term}}} = \underbrace{S_\phi(\phi)}_{\substack{\text{source}\\ \text{term}}} \qquad (2.26)$$

where ρ is the density, **u** the velocity field and Γ the diffusivity. Discretizing (2.26) over a time interval t, t+Δt and over the volume V_P (cell with the centroid point P), the volume integral form, results:

$$\int_t^{t+\Delta t}\left[\frac{\partial}{\partial t}\int_{Vp}\rho\phi\,dV + \int_{Vp}\nabla\cdot(\rho\mathbf{u}\phi)\,dV - \int_{Vp}\nabla\cdot\left(\rho\Gamma_\phi\nabla\phi\right)dV\right]dt = \int_t^{t+\Delta t}\left[\int_{Vp} S_\phi(\phi)\,dV\right]dt \qquad (2.27)$$

The following Sections show the main steps for the spatial and temporal discretization of the transport equation.

Gradient terms

As mentioned before, the advection and diffusion terms needs to be simplified into surface integrals over the cell faces, using the Gauss theorem. The generalized form of the Gauss's theorem for any tensor field ϕ (i.e. vector or scalar) is represented:

$$\int_{V} \nabla \circledast \phi \; dV = \oint_{\partial V} \phi \circledast dS \tag{2.28}$$

where ∂V is the closed surface bounding the volume V and dS represents an infinitesimal surface element with associated outwards pointing normal on ∂V. The star notation ⍟ is used to represent any tensor product: inner, outer or cross; and the respective derivate: divergence $\nabla \cdot \phi$, gradient $\nabla\phi$ and curl $\nabla \times \phi$.

Since the variable **ϕ** are stored on the cell center, the respective value on the face needs to be obtained by interpolation. Taking into account the linear variation of ϕ in space x, the integral of this variable in the volume V_P is:

$$\int_{V_P} \phi(x)\; dV = \int_{V_P} [\phi_P + (x - x_P) \cdot (\nabla\phi)_P]\; dV = \phi_P V_P \tag{2.29}$$

Assuming that Equation (2.28) can be transformed into a sum of integrals over the faces and a linear variation of ϕ, the discretized form of Gauss theorem results in:

$$\int_{V_P} \nabla\boldsymbol{\phi}\; dV = \oint_{\partial V_P} \boldsymbol{\phi} dS = \sum_f \left(\int_f \boldsymbol{\phi} dS \right) \approx \sum_f \mathbf{A}_f \boldsymbol{\phi}_f \tag{2.30}$$

where **A** is the outward normal surface area vector of the faces in the control cell, and $\boldsymbol{\phi}_f$ is the value of the variable in the face.

Face Interpolation Schemes

There are many schemes to interpolate the field ϕ. Three different schemes are presented next, with differences regarding boundedness and accuracy.

The **Central Differencing** (CD) scheme assumes a linear variation of ϕ between P and N (Figure 2.9) and the face value is calculated according to:

$$\phi_f = f_x \phi_P + (1 - f_x)\phi_N \tag{2.31}$$

where the interpolator factor f_x is defined as the ratio of the distances $\overline{fN}$ and $\overline{PN}$. Ferziger and Peric (2002) showed that this scheme is second order accurate, although it causes unphysical oscillations in the solution and violates the boundedness of the solution (Pantakar, 1980).

The **Upwind Differencing** (UD) scheme assumes the face value of ϕ, determined according to the direction of the flow. If the flux (F) goes from the point P to N, then $\phi_f=\phi_P$, otherwise $\phi_f=\phi_N$ (2.32). In this scheme, the boundedness of solution is guaranteed, however the order of accuracy of the discretization is not guaranteed as second order and the solution can become distorted (Jasak, 1996).

$$\phi_f = \begin{cases} \phi_P & \text{for } F \geq 0 \\ \phi_N & \text{for } F < 0 \end{cases} \tag{2.32}$$

The **Blended Differencing** (BD) scheme combines the CD and UD schemes in an attempt to preserve boundedness with some accuracy on the solution (Rusche, 2002). The BD scheme is also named by **Gamma differencing** scheme, since it takes special attention in the selection of the blending coefficient γ.

$$\phi_f = (1-\gamma)(\phi_f)_{UD} + \gamma(\phi_f)_{CD} \tag{2.33}$$

Jasak (1996) developed and implemented the Gamma differencing scheme presented on the OpenFOAM (Gamma ψ). OpenFOAM offers several others well-known schemes such as Van Leer, MUSCL or SFCD.

Temporal Derivate

The temporal derivate [first term of equation (2.27)] is usually discretized using a fist order or a second order accurate scheme in time. The **Euler implicit** is an example of a first order time differencing scheme:

$$\int_{V_P} \frac{\partial \rho \boldsymbol{\phi}}{\partial t} \, dV \approx V_P \frac{\rho^n \boldsymbol{\phi}^n - \rho^0 \boldsymbol{\phi}^0}{\Delta t} \tag{2.34}$$

where u^0 is the known value of u from the previous time step and u^n is the unknown value of u at the current time step $t_n=n\Delta t$. An example of second order accurate in time is the **Backward differencing** scheme:

$$\int_{V_P} \frac{\partial \rho \boldsymbol{\phi}}{\partial t} \, dV \approx V_P \frac{3(\rho^n \boldsymbol{\phi}^n) - 4(\rho^0 \boldsymbol{\phi}^0) + (\rho^{00} \boldsymbol{\phi}^{00})}{2\Delta t} \tag{2.35}$$

where u^{00} is the value of u from a time step previous to the last.

Convection Term

Appling the Equation (2.30) to the convection term of the momentum equation gives:

$$\int_{V_P} \nabla \cdot (\rho \mathbf{u} \boldsymbol{\phi})\, dV = \sum_f \mathbf{A}_f \cdot (\rho \mathbf{u} \boldsymbol{\phi})_f = \sum_f \mathbf{A}_f \cdot (\rho \mathbf{u})_f \boldsymbol{\phi}_f = \sum_f F_f \boldsymbol{\phi}_f \qquad (2.36)$$

where F_f represents the face mass flux through the face based on a known velocity field and ϕ_f is determined through one interpolation scheme.

Diffusion Term

The diffusion term (also called as Laplacian term) is discretized in a similar way to the convection term.

$$\int_{Vp} \nabla \cdot (\Gamma \nabla \boldsymbol{\phi})\, dV = \sum_f \Gamma_f \mathbf{A}_f \cdot (\nabla \boldsymbol{\phi})_f \qquad (2.37)$$

where the diffusivity Γ at the face Γ_f is calculated using the one interpolation scheme.

If the mesh is **orthogonal**, i.e. the vectors **A** and **D** are parallel, considering the Equation (2.25), it is possible to use the following expression:

$$\mathbf{A}_f \cdot (\nabla \boldsymbol{\phi})_f = |\mathbf{D}_f| \frac{\boldsymbol{\phi}_N - \boldsymbol{\phi}_P}{|\mathbf{d}_f|} \qquad (2.38)$$

If the mesh in **non-orthogonal**, it is necessary to introduce an additional explicit term (k_f) to induce higher accuracy of the Equation (2.38) (see also Figure 2.9):

$$\mathbf{A}_f \cdot (\nabla \boldsymbol{\phi})_f = \mathbf{D}_f \cdot (\nabla \boldsymbol{\phi})_f + \mathbf{k}_f \cdot (\nabla \boldsymbol{\phi})_f \qquad (2.39)$$

There are numerous possible decompositions to correct the orthogonality. In OpenFOAM, the orthogonal correction is made using an Over-relaxed approach (Jasak, 1996; Ubbink, 1997).

Source Term

The terms on the right side of the Equation (2.27) which cannot be written as convection, diffusion, or temporal terms are treated as sources. A simple procedure of linearization follows the work of Pantakar (1980):

$$S_\phi(\phi) = Su + Sp\, \phi \qquad (2.40)$$

where Su and Sp can also depend on ϕ. Using the assumption made in Equation (2.29) the volume integral of the source term is:

$$\int_{V_P} S_\phi(\phi)\, dV = Su\, V_P + Sp\, \phi_P V_P \qquad (2.41)$$

2.5.2.2. Discretization of the Spatial Terms of Momentum Equation

Following the discretization process presented for the general transport equation, the momentum equation (2.21) of the Navier Stokes Equations can be presented over the control volume and the time step Δt as follow:

$$\begin{aligned}&\int_t^{t+\Delta t}\left[\frac{\partial}{\partial t}\int_{Vp}\rho\mathbf{u}\, dV + \int_{Vp}\nabla\cdot(\rho\mathbf{u}\mathbf{u})\, dV - \int_{Vp}\nabla\cdot(\mu\nabla\mathbf{u})\, dV\right]dt\\ &= \int_t^{t+\Delta t}\Bigg[\Bigg[\int_{Vp}(-\nabla p* \ +(\nabla\mathbf{u})\cdot\nabla\mu \ + \ g\cdot x\nabla\rho \ + \ \sigma\kappa\nabla\alpha)\, dV\Bigg]dt\end{aligned} \qquad (2.42)$$

The final form of the momentum equation after the terms discretization is expressed as:

$$\begin{aligned}&\int_t^{\Delta t}\left[V_P\frac{\rho^n\mathbf{u}^n-\rho^0\mathbf{u}^0}{\Delta t}+\sum_f F_f\mathbf{u}_f - \sum_f \mu_f A_f \ \cdot(\nabla\mathbf{u})_f\right]dt\\ &= \int_t^{\Delta t}[Su\, V_P + Sp\, \phi_P V_P]\, dt\end{aligned} \qquad (2.43)$$

2.5.2.3. Discretization of the Phase Fraction Transport Equation

The final form of the Phase Fraction Transport Equation is described previously by the equation (2.17). The finite volume discretization over the volume control and the time step Δt assumes the form:

$$\begin{aligned}&\int_t^{t+\Delta t}\left[\int_V\frac{\partial\alpha}{\partial t}dV\right]dt + \int_t^{t+\Delta t}\left[\int_V\nabla\cdot(\alpha\mathbf{u})dV\right]dt + \int_t^{t+\Delta t}\left[\int_V\nabla\cdot[\mathbf{u}_r\alpha(1-\alpha)]dV\right]dt\\ &= 0\end{aligned} \qquad (2.44)$$

Assuming the linear variation of α, the first term of the equation can reduced to:

$$\int_V\frac{\partial\alpha}{\partial t}dV = \frac{\partial\alpha_P}{\partial t}V_P \qquad (2.45)$$

The second and third terms of the equation (2.44) are discretized applying the Gauss theorem. Exceptionally, in the discretization of the artificial compression term (third term), the velocity at the cells is determined from the gradient of phase fraction:

$$\mathbf{u}_{\mathrm{r},f} = \mathbf{n}_f \min\left[\mathrm{C}_\alpha \frac{|\phi|}{|\mathbf{A}_f|}, \max\left(\frac{|\phi|}{|\mathbf{A}_f|}\right)\right] \tag{2.46}$$

where ϕ is the face volume flux and n_f is the face unit normal flux, calculated in the interface region based on the phase fraction gradient at cell faces:

$$\mathbf{n}_f = \frac{(\nabla\alpha)_f}{\left|(\nabla\alpha)_f + \delta_\mathrm{n}\right|} \cdot \mathbf{A}_f \tag{2.47}$$

where δ_n is the stabilization parameter, which takes into account the non-uniformity of the grid. This parameter usually takes the value of 10^{-5}.

2.5.2.4. Temporal Discretization

Following the description of the OpenFOAM™ manual (OpenFOAM, 2012), the transient PDE presented in (2.27) can be simplified as following, considering the spatial terms $\mathcal{A}\phi$ where $\mathcal{A}$ is a spatial operator:

$$\int_{\mathrm{t}}^{\mathrm{t}+\Delta\mathrm{t}} \left[\frac{\partial}{\partial \mathrm{t}} \int_{\mathrm{Vp}} \rho\phi \, \mathrm{dV} + \int_{\mathrm{Vp}} \mathcal{A}\phi \, \mathrm{dV}\right] \mathrm{dt} = 0 \tag{2.48}$$

Using the Euler implicit method (2.34), the first term of the equation (2.48) returns:

$$\begin{aligned}\int_{\mathrm{t}}^{\mathrm{t}+\Delta\mathrm{t}} \left[\frac{\partial}{\partial \mathrm{t}} \int_{\mathrm{Vp}} \rho\phi \, \mathrm{dV}\right] \mathrm{dt} &= \int_{\mathrm{t}}^{\mathrm{t}+\Delta\mathrm{t}} \left[\frac{(\rho_\mathrm{P}\phi_\mathrm{P}\mathrm{V}_\mathrm{P})^\mathrm{n} - (\rho_\mathrm{P}\phi_\mathrm{P}\mathrm{V}_\mathrm{P})^0}{\Delta\mathrm{t}}\right] \mathrm{dt} \\ &= \rho_\mathrm{P} \frac{(\phi_\mathrm{P})^\mathrm{n} - (\phi_\mathrm{P})^0}{\Delta\mathrm{t}} \mathrm{V}_\mathrm{P}\end{aligned} \tag{2.49}$$

The second term of the equation (2.48) can be discretized in OpenFOAM™ by three ways:

The **Euler implicit** is a first order scheme accurate in time, guarantees boundedness and it is unconditionally stable. This scheme uses the current values of ϕ, thereby the solution needs to be achieved using a matrix.

The **Euler explicit** is a fist order scheme accurate in time and is unstable for Courant numbers greater than unity. This scheme uses only the old values of ϕ. The Courant number is defined as:

$$\mathrm{Co} = \frac{\mathbf{u}_\mathrm{f} \cdot \mathbf{d}}{|\mathbf{d}|^2 \Delta t} \tag{2.50}$$

The **Crank Nicholson** is a second order scheme accurate in time. It is unconditionally stable (Hirsch, 1988) but does not guarantee the boundedness of the solution (Jasak, 1996). It uses the trapezoidal rule to discretize the spatial terms, thereby taking the mean value of the current values and old values of ϕ.

$$\int_t^{t+\Delta t} \boldsymbol{\mathcal{A}}\phi \, dt = \frac{1}{2}\boldsymbol{\mathcal{A}}(\phi^n + \phi^0) \tag{2.51}$$

Assuming now that the density and viscosity do not change in time, and applying the Crank Nicolson scheme (2.51), the momentum equation returns:

$$\begin{aligned} &V_P \frac{\rho \mathbf{u}^n}{\Delta t} + \frac{1}{2}\left[\sum_f F_f \mathbf{u}_f{}^n - \sum_f \mu_f \mathbf{A}_f \cdot (\nabla \mathbf{u})_f{}^n - \mathrm{SpVp}\phi_P{}^n\right] \\ &= V_P \frac{\rho \mathbf{u}^0}{\Delta t} - \frac{1}{2}\left[\sum_f F_f \mathbf{u}_f{}^0 - \sum_f \mu_f \mathbf{A}_f \cdot (\nabla \mathbf{u})_f{}^0 - \mathrm{SpVp}\phi_P{}^0\right] + \mathrm{SuVp} \end{aligned} \tag{2.52}$$

The discretisation and linearization procedure outlined produces a linear algebraic equation for each control volume. Since $\mathbf{u}_f$, $(\nabla\mathbf{u})_f$ and other terms depend on the values on the neighbouring cells, the new values of $\mathbf{u}_P$ can be implicitly achieved from the generic equation:

$$a_P \mathbf{u}_P^n + \sum_N a_N \mathbf{u}_N^n = R_P \tag{2.53}$$

2.5.2.5. Boundary and Initial Conditions

To complete the problem, it is essential to define the boundary and the initial conditions. The Boundary Conditions (BC) are specified on all boundary faces of the domain and can be divided essentially into two types (OpenFOAM, 2012):

- **Dirichlet** – prescribes the value of the dependent variable on the boundary and is therefore termed as "**fixed value**" in OpenFOAM^{TM}.
- **Neumann** - prescribes the gradient of the variable normal to the boundary and is therefore termed as "**fixed gradient**" in OpenFOAM^{TM}.

The treatment of the boundary conditions is not subject of this Thesis project, thus a simple summary of the common characteristics is suitable. Rusche (2002) summarises the boundary conditions within the following Table 1.

Table 1 – Numerical boundary conditions [Retrieved from Rusche (2002)].

Variable	No-Slip Wall	Inlet	Outlet	Symmetry Plane
u	Fixed Value	Fixed Value	Zero Gradient	Mirror Image
α	Zero Gradient	Fixed Value	Zero Gradient	Mirror Image
p	Zero Gradient	Zero Gradient	Fixed Value	Zero Gradient
k	Fixed Value	Fixed Value	Zero Gradient	Zero Gradient
ε	Fixed Value	Fixed Value	Zero Gradient	Zero Gradient

The initial conditions determine the state of fluid at time t=0, or at first step of an iterative scheme.

More detailed description about boundary and initial conditions can be found in Jasak (1996), Ubbink (1997) or Rusche (2002).

2.5.3. Solution Procedure

The form of the equations discretized in the previous sections shows linear dependence of velocity on pressure and vice-versa. This inter-equation coupling of the two-fluid system requires a special treatment by one of two forms:

a) The **Simultaneous Algorithms** (Caretto *et al.*, 1972; Vanka, 1986) solve the complete system of equations simultaneously over the entire domain. However, this procedure is only applicable when the computational domain is small and the number of equations is not too large.

b) Using the **Segregated Approach** (Pantakar, 1980; Issa, 1985), the equations system is solved sequentially. One special treatment need to be considered relatively to the inter-equation coupling in the pressure-velocity system. The SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) (and their variables) and PISO (Pressure Implicit with Splitting of Operators) are the most popular methods used. The solution of the equations system through the PISO algorithm will be explored in section 2.5.3.1.

2.5.3.1. Pressure-Velocity Solution Procedure – PISO algorithm

The derivation of the pressure-velocity equation follows the works of (Rusche, 2002; Berberović *et al.*, 2009). A semi-discretized form of the momentum equation, obtained from the equation (2.53) can be written as:

$$\mathrm{a_P}\mathbf{u}_\mathrm{P} = \mathbf{H}(\mathbf{u}) - \nabla \mathrm{p}^* - \mathbf{g}\cdot\mathbf{x}\nabla\rho + \sigma\kappa\nabla\alpha \tag{2.54}$$

The term **H**(**u**) is divided into two parts: the first is the *transport part*, which includes the matrix of coefficients for the neighbouring cells, multiplied by the correspondent velocity, and the second part or *source part*, composed by the source terms apart from the surface tension and buoyancy terms. The equation (2.54) is used to express the velocity **u** at the cell center:

$$\mathbf{u}_\mathrm{P} = [\mathrm{a_P}]^{-1}[\mathbf{H}(\mathbf{u}) - \nabla \mathrm{p} - \mathbf{g}\cdot\mathbf{x}\nabla\rho + \sigma\kappa\nabla\alpha] \tag{2.55}$$

Interpolating linearly the values at the cell centres using central differencing, the flux predictor and corrector can be achieved using:

$$\mathrm{F} = -[\mathrm{a_P}]^{-1}\mathbf{A}_f\cdot\nabla \mathrm{p}_f + [\mathrm{a_P}]^{-1}\mathbf{A}_f\cdot[\mathbf{H}(\mathbf{u}) - \mathbf{g}\cdot\mathbf{x}\nabla\rho + \sigma\kappa\nabla\alpha]_f \tag{2.56}$$

Replacing the velocity reach in (2.55) in the **continuity equation**, the undiscretized form for the **Pressure Poisson Equation** is:

$$\nabla\cdot\left[[\mathrm{a_P}]^{-1}\nabla \mathrm{p}_f\right] = \nabla\cdot\left[[\mathrm{a_P}]^{-1}[\mathbf{H}(\mathbf{u}) - \mathbf{g}\cdot\mathbf{x}\nabla\rho + \sigma\kappa\nabla\alpha]_f\right] \tag{2.57}$$

Combining the equations (2.56) and the Gauss theorem, the discretized form of the pressure equation is obtained.

The solution procedure in the *interFoam* employs the PISO algorithm, proposed by Issa (1985). The summary of this algorithm is presented by Rusche (2002):

1. **Momentum Predictor** - The momentum equation (2.54) is solved first. Nevertheless, the exact pressure gradient is not known at this step. Thus, the pressure field from the previous time step is used;

2. **Pressure Solution** – Using the previous velocities, the pressure equation can be formulated assembling the **H**(**u**) operator. The solution of the discretized form of the pressure equation (2.57) gives the first estimate of the new pressure fields.

3. **Velocity Correction** – The explicit correction of the velocity is done using the new pressure fields applying the equation (2.55).

In the explicit resolution of the Equation (2.55) it was assumed that a whole velocity error comes from the error in the pressure term and this assumption is not absolutely true. Effectively, part of the error was originated implicitly by the term **H**(**u**). In order to obtain a better approximation of the "correct" pressure field, it would be necessary to solve the pressure equation again. This procedure is repeated until pre-determined tolerance between consecutive values of velocity fields is reached at each time step.

2.5.3.2. Adaptive Time-Step

The stability of the solution procedure is ensured using an adaptive time step, which is adjusted at the beginning of the time iteration loop based on the Courant number [Equation (2.50)]. Using values for $\mathbf{u}_f$ and Δt from previous time step, a maximum local Courant number Co^0 is calculated and the new time step is evaluated from the expression

$$\Delta t^n = \min\left\{\frac{\mathrm{Co_{max}}}{\mathrm{Co}^0}\Delta t^0; \left(1+\lambda_1\frac{\mathrm{Co_{max}}}{\mathrm{Co}^0}\right)\Delta t^0;\ \lambda_2\Delta t^0;\ \Delta t_{max}\right\} \tag{2.58}$$

where Δt_{max} and $\mathrm{Co_{max}}$ are predefined limit values for the time-step and Courant number. According to Berberović *et al.* (2009), the Courant number should not exceed the value of $\mathrm{Co_{max}}=0.2$ and the damping factors are set to $\lambda_1=0.1$ and $\lambda_2=1.2$ (consult the *interFoam* code - section 3.1.2). Nevertheless, in the beginning of the simulation an intermediate time-step value (Δt_{ini}^*) are calculated from a very small initial value of time-step (Δt_{ini}) given by the equation:

$$\Delta t_{ini}^* = \min\left(\frac{\mathrm{Co_{max}}}{\mathrm{Co}^0}\Delta t_{ini}; \Delta t_{max}\right) \tag{2.59}$$

Hence, the value of Δt^0 in the first time-step (2.58) is, in fact, the intermediate time-step.

2.5.3.3. Temporal Subcycling of Alpha Equation

The convergence and stability of the solution in VOF-based methods are very sensitive regarding the phase fraction equation. Together with some procedures such as the Time-step control, bounded discretization of the convective term, and set the Courant number to a value away and below the unity, it is essential to solve the phase fraction equation in several subcycles within a single time-step, dividing the global time (Δt) by a number of subcycles (n_{sc}) (Berberović *et al.*, 2009):

$$\Delta \mathrm{t}_{\mathrm{sc}} = \frac{\Delta \mathrm{t}}{\mathrm{n}_{\mathrm{sc}}} \qquad (2.60)$$

After the phase fraction in each updated subcycle, the corresponding face flux is calculated (F_{sc}). The total face flux ($F=\mathbf{u}_f.\mathbf{A}_f$) in the global time step is a weight temporal sum of the subcycles face fluxes.

2.5.3.4. Sequence of solution

The coupled system of equations is solved sequentially i.e. for the algebraic system resulting from the discretization. The procedure of a transient solution can be summarized as follow (Ubbink, 1997; Rusche, 2002):

1. Initiate all the variables;
2. Calculate Courant number (2.50) and adjust time-step (2.58);
3. Solve the Phase Fraction Equation (2.44) using the old time level's volumetric fluxes;
4. Use the new α values together with the constitutive relations to obtain an estimate for the new viscosity, density and the face densities;
5. Calculate the unit normal vector and the curvature (2.19);
6. Assemble and solve the momentum predictor equation (2.54);
7. Go through the PISO loop until the pre-defined tolerance for pressure-velocity system is reached.

The convergence rate can be improved by applying preconditioning matrix. In case of symmetric matrixes, the Preconditioned Conjugate Gradient (PCG) solver with Diagonal-based Incomplete Cholesky (DIC) preconditioner is used. In case of asymmetric matrixes, the Preconditioned Bi-Conjugate Gradient (PBiCG) solver with Diagonal-based Incomplete Lower-Upper (DILU) preconditioner is used.

3. INTERFOAM CODE DESCRIPTION

The computational implementation of the discretized form of the equations descripted in Section 2.5 for the *interFoam* solver is carried out using OpenFOAM™ libraries written in C++. A simpler description of those libraries is available in the OpenFOAM™ User Manual and OpenFOAM™ Programmer's Guide. Since one topic of this study is study the *interFoam*, this chapter has the intention to makes a brief introduction to the main libraries and subroutines of the *interFoam* solver version 2.1.1.

The libraries interFoam.H, setDeltaT.H, alphaEqSubCycle.H, alphaEqn.H, UEqn.H and pEqn.H are presented firstly in different sections. Afterwards the code is explained.

3.1. Source Code

3.1.1. interFoam.H

```
#include "fvCFD.H"
#include "MULES.H"
#include "subCycle.H"
#include "interfaceProperties.H"
#include "twoPhaseMixture.H"
#include "turbulenceModel.H"
#include "interpolationTable.H"
#include "pimpleControl.H"

//  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//

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

    pimpleControl pimple(mesh);

    #include "initContinuityErrs.H"
    #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;

        twoPhaseProperties.correct();

        #include "alphaEqnSubCycle.H"

        // --- Pressure-velocity PIMPLE corrector loop
        while (pimple.loop())
        {
            #include "UEqn.H"

            // --- Pressure corrector loop
            while (pimple.correct())
            {
                #include "pEqn.H"
            }

            if (pimple.turbCorr())
            {
                turbulence->correct();
            }
        }

        runTime.write();

        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }

    Info<< "End\n" << endl;

    return 0;
}

// *********************************************************************** //
```

3.1.2. setDeltaT.H

```
if (adjustTimeStep)
{
    scalar maxDeltaTFact =
        min(maxCo/(CoNum + SMALL), maxAlphaCo/(alphaCoNum + SMALL));

    scalar deltaTFact = min(min(maxDeltaTFact, 1.0 + 0.1*maxDeltaTFact), 
1.2);

    runTime.setDeltaT
    (
        min
        (
            deltaTFact*runTime.deltaTValue(),
            maxDeltaT
        )
    );

    Info<< "deltaT = " <<  runTime.deltaTValue() << endl;
}

// ************************************************************************* //
```

3.1.3. alphaEqSubCycle.H

```

label nAlphaCorr(readLabel(pimple.dict().lookup("nAlphaCorr")));

label nAlphaSubCycles(readLabel(pimple.dict().lookup("nAlphaSubCycles")));

if (nAlphaSubCycles > 1)
{
    dimensionedScalar totalDeltaT = runTime.deltaT();
    surfaceScalarField rhoPhiSum(0.0*rhoPhi);

    for
    (
        subCycle<volScalarField> alphaSubCycle(alpha1, nAlphaSubCycles);
        !(++alphaSubCycle).end();
    )
    {
        #include "alphaEqn.H"
        rhoPhiSum += (runTime.deltaT()/totalDeltaT)*rhoPhi;
    }

    rhoPhi = rhoPhiSum;
}
else
{
    #include "alphaEqn.H"
}

interface.correct();

rho == alpha1*rho1 + (scalar(1) - alpha1)*rho2;

// ************************************************************************* //
```

3.1.4. alphaEqn.H

```
{
    word alphaScheme("div(phi,alpha)");
    word alpharScheme("div(phirb,alpha)");

    surfaceScalarField phic(mag(phi/mesh.magSf()));
    phic = min(interface.cAlpha()*phic, max(phic));
    surfaceScalarField phir(phic*interface.nHatf());

    for (int aCorr=0; aCorr<nAlphaCorr; aCorr++)
    {
        surfaceScalarField phiAlpha
        (
            fvc::flux
            (
                phi,
                alpha1,
                alphaScheme
            )
          + fvc::flux
            (
                -fvc::flux(-phir, scalar(1) - alpha1, alpharScheme),
                alpha1,
                alpharScheme
            )
        );

        MULES::explicitSolve(alpha1, phi, phiAlpha, 1, 0);

        rhoPhi = phiAlpha*(rho1 - rho2) + phi*rho2;
    }

    Info<< "Phase-1 volume fraction = "
        << alpha1.weightedAverage(mesh.Vsc()).value()
        << "  Min(alpha1) = " << min(alpha1).value()
        << "  Max(alpha1) = " << max(alpha1).value()
        << endl;
}


// ********************************************************************** //
```

3.1.5. UEqn.H

```
    surfaceScalarField muEff
    (
        "muEff",
        twoPhaseProperties.muf()
      + fvc::interpolate(rho*turbulence->nut())
    );

    fvVectorMatrix UEqn
    (
        fvm::ddt(rho, U)
      + fvm::div(rhoPhi, U)
      - fvm::laplacian(muEff, U)
      - (fvc::grad(U) & fvc::grad(muEff))
    //- fvc::div(muEff*(fvc::interpolate(dev(fvc::grad(U))) & mesh.Sf()))
    );

    UEqn.relax();

    if (pimple.momentumPredictor())
    {
        solve
        (
            UEqn
         ==
            fvc::reconstruct
            (
                (

fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)
                  - ghf*fvc::snGrad(rho)
                  - fvc::snGrad(p_rgh)
                ) * mesh.magSf()
            )
        );
    }

// ************************************************************************* //
```

3.1.6. pEqn.H

```
{
    volScalarField rAU(1.0/UEqn.A());
    surfaceScalarField rAUf(fvc::interpolate(rAU));

    U = rAU*UEqn.H();
    surfaceScalarField phiU
    (
        "phiU",
        (fvc::interpolate(U) & mesh.Sf())
      + fvc::ddtPhiCorr(rAU, rho, U, phi)
    );

    adjustPhi(phiU, U, p_rgh);

    phi = phiU +
    (
        fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)
      - ghf*fvc::snGrad(rho)
    )*rAUf*mesh.magSf();

    while (pimple.correctNonOrthogonal())
    {
        fvScalarMatrix p_rghEqn
        (
241                 fvm::laplacian(rAUf, p_rgh) == fvc::div(phi)
242             );
243 
244             p_rghEqn.setReference(pRefCell, getRefCellValue(p_rgh, pRefCell));
245 
246             p_rghEqn.solve(mesh.solver(p_rgh.select(pimple.finalInnerIter())));
247 
248             if (pimple.finalNonOrthogonalIter())
249             {
250                 phi -= p_rghEqn.flux();
251             }
252         }
253 
254         U += rAU*fvc::reconstruct((phi - phiU)/rAUf);
255         U.correctBoundaryConditions();
256 
257         #include "continuityErrs.H"
258 
259         p == p_rgh + rho*gh;
260 
261         if (p_rgh.needReference())
262         {
263             p += dimensionedScalar
264             (
265                 "p",
266                 p.dimensions(),
267                 pRefValue - getRefCellValue(p, pRefCell)
268             );
269             p_rgh = p - rho*gh;
270         }
271     }
272 
273     // ********************************************************************* //
```

3.2. Code Description

The following table highlights the main lines within the *interFoam* solver code.

Table 2 – Description of the main lines within the interFoam code.

Line(s)	*Description*
1-77	**interFoam.H**: this is the main library of *interFoam* solver.
2-6	The present lines include other libraries or functions of the *interFoam*. The most essential is the `fvCFD.H`, included with the aim of have available all the Finite Volume machinery.
33-70	This is the main loop of *interFoam*. Starts reading the Controls, define the Courant Number and the time step (`deltaT`). The time step is defined through the adaptive time-step explored in Section 2.5.3.2.

Continued on next page

Line(s)	*Description*
46	Include the alpha subcycle as mentioned in section 2.5.3.3 and showed in section 3.1.3
51	After solving the alpha equation in subcycles, the continuity and momentum equations are solved integrated with the PISO loop.
56	Include the pressure Equation
65-69	Print on the screen the information relatively to computational and clock time.
78-100	**SetDeltaT.H**: the adjustable time step is defined here.
82-95	This library set the simulation time-step as section 2.5.3.2. Firstly is defined the factor `maxDeltaTFact` $\left(\frac{\mathrm{Co}_{\mathrm{max}}}{\mathrm{Co}^{0}}\right)$ in line 82. The `runTime.setDeltaT` value (lines 88-95) represents the result of equation (2.58).
101-133	**alphaEqSubCycle.H**: in this library the phase fraction equation is solved in several subcycles.
103-105	`nAlphaCorr` and `nAlphaSubCycles` values are read from PISO section of ./system/fvSolution dictionary, previously defined by the user.
107-127	If the number of subcycles is greater than 1, the total time step is divided in subcycles, otherwise the alpha Equation is solved once (line 126). In line 109 is defined the time step calculated in SetDeltaT.H. The total flux (F)(`rhoPhiSum`) is initially set to 0 at line 110.
118-119	It is introduced the `alphaEqn.H` library which solves the equation of alpha (2.17). In line 119 the total Flux (`rhoPhiSum`) is calculated as a sum of the values obtained from alpha Equation.
129-131	Curvature κ (2.19) and density field (2.15) and (2.16) are updated in these lines.
134-175	**alphaEqn.H**: the alpha equation is solved here.
137-138	`words` are used to save the divergence schemes defined by the user to discretize the terms which contains $\bar{\mathbf{u}}$ and $\mathbf{u}_r$ in equation (2.17)
140-142	The term `phir` represents the result of the compression term of Equation (2.46) whereas, `phic` represents the term $\frac{\lvert\phi\rvert}{\lvert\mathbf{A}_f\rvert}$.
144-165	The alpha Equation is solved interactively until reach `nAlphaCorr` defined by the user in /system/fvSolution dictionary. Between lines 148-160 is solved the second and third term of the alpha Equation (2.17) represented by `phiAlpha`. MULES is called in order to solve alpha, giving it the unknown variable (alpha1), the overall flux (phi), the non-temporal terms (`phiAlpha`), and the bounds for the unknown ($0 < \alpha < 1$). Finally, the `rhoPhi` is actualized from the phiAlpha and densities (line 164).
167-171	These lines give a report about alpha field.

Continued on next page

Line(s)	*Description*
176-214	**UEqn.H**: The solution of the momentum equation is presented in this library.
185-192	Part of the momentum equation (2.21) is solved. The code: `UEqn(fvm::ddt(rho, U)+ fvm::div(rhoPhi, U)- fvm::laplacian(muEff, U)- (fvc::grad(U) & fvc::grad(muEff)))` represents respectively the terms: $\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{uu}) - \nabla \cdot (\mu \nabla \mathbf{u}) - (\nabla \mathbf{u}) \cdot \nabla \mu$.
206-208	The remaining terms of momentum equation are solved in these lines.
215-273	**pEqn.H**:
218-221	The central coefficient of the velocity equation matrix are saved and inverted, where UEqn.A() give the central coefficients. After, these coefficients are interpolated into faces values (rUAf). Finally, the velocity field is predicted by U=rAU*H().
222-235	The terms in the right side of the pressure equation (2.57) is assembled in two step fluxes. The first (`phiU`) (lines 222-227) corresponds to the non-diagonal arguments of the matrix. After the second flux, correspond to the gravity and surface tension terms is added to the first, giving the flux `phi` (lines 231-235).
241	The pressure equation is defined as indicated in (2.57).
250	The new flux are assembled using Eq. (2.56). Non orthogonal corrections are made using a loop until reach the nNonOrthCorr value.
254	Velocity field is corrected after Pressure loop using (2.55). After, the velocity is also corrected for the boundaries.
259	The total pressure is calculated.

4. TEST CASE

Previous studies elaborated by the Author in gullies, one presented in the 9th International Conference on Urban Drainage Modelling (Lopes *et al.*, 2012) and other published in the Urban Water Journal (Lopes *et al.*, 2013) demonstrate that in case of free-surface flows with lower values of air-entrainment, the *interFoam* solver can be a powerful tool to predict the behaviour of the flow. In this Thesis project, the intention is to study air-water flows in presence of reasonable quantities of air. The results were achieved numerically, using the *interFoam* solver with various turbulence models and, experimentally, in a real-scale facility built for this purpose, at the Department of Civil Engineering of the University of Coimbra.

4.1. Experimental Facility and Equipment

The experiments were conducted at a real-scale physical facility built at the Hydraulic, Water Resources and Environment Laboratory (LHRHA) at the Civil Engineering Department of the University of Coimbra (DEC-FCTUC). This experimental model is shown in the Figure 4.1. It consists of a 2m long and 0.42m interior diameter PVC Glass tube, positioned vertically where the experiments will occur. The PVC Glass tube is fed by a superior reservoir that ensures the static pressure and the steady state of water flow. The superior reservoir is supplied by a pump from another reservoir (inferior reservoir). Both air and water flows, are measured by commons flow meters with specifications consultable in Table 3.

At the inferior end of the PVC Glass tube a transparent air tube is introduced. The air tube extends through the interior of the PVC tube until it reaches a height of 260 mm below the top (see amplified figure in Figure 4.1a) where it is connected to a pneumatic silencer to produce air bubbles. At the end of the acrylic tube a made in-house resistive dual-tip probe is positioned.

Table 3 - Specifications of the water and air flow meters used in the experimental facility

Flow meters	**Water**	**Air**
Connections	1’’	¾’’
Minimum Flow (l/s)	0.019	0.014
Nominal Flow (l/s)	0.972	0.694
Maximum Flow (l/s)	1.944	1.389
Sensibility (±l/s)	0.0047	0.0038
Maximum Pressure (m.c.a)	163.2	163.2

The installation allows a large range of air flow rates (Q_{air}), from 0.0718 l/s to 0.2752 l/s that correspond to values of air concentrations (C_{air}) between 12.33% and 39.7%. These values were calculated using the volume measurements from the two flow meters and through the formula: $C_{air}=Q_{air}/(Q_{air}+Q_{water})$. In this test case were chosen Q_{air} = 0.115 l/s thus, C_{air} = 17.9%.

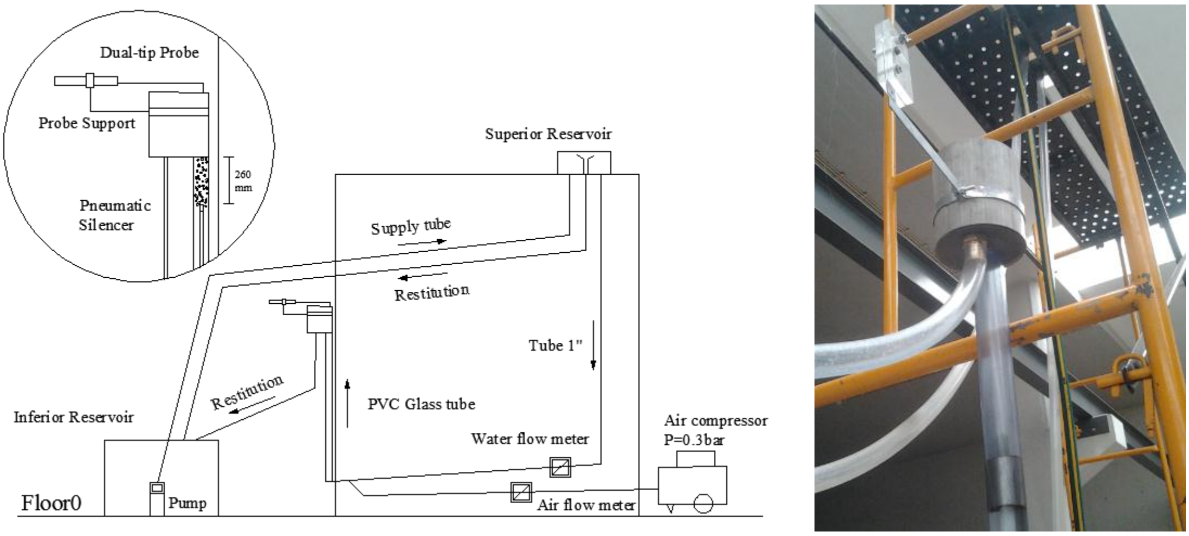
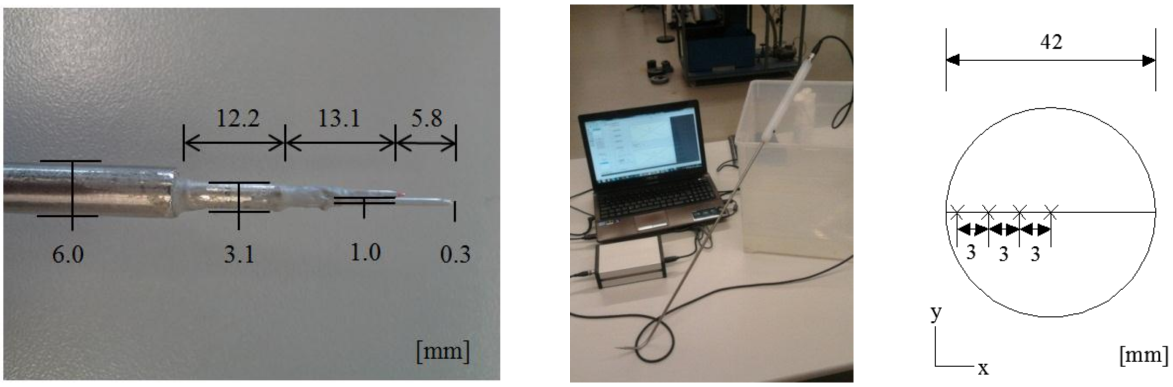


Figure 4.1 – (a) Schematic defining the experimental facility constructed at DEC-FCTUC (University of Coimbra). (b) Photography of the mixing zone.

The dual-tip resistive probe, developed by the Civil Engineering and Electrical and Computers Engineering Departments of the University of Coimbra is used to measure the void fraction on the vertical PVC tube. This probe is also known as conductive probe and it is able to detect the electrical conductivity in water or air using two electrodes which are arranged back-to-back in the flow direction. The dual-tip probe differs from the single type by measuring simultaneously the void fraction and the velocity of the bubbles by correlating the tips signals. The detailed dimensions of probe tips are showed in (Figure 4.2a). The internal electrode is a copper wire with an outer diameter of 0.3mm. The external electrode is a stainless steel tube with outer diameter of 1mm. The two probes tips are in the flow direction distanced by 5.8mm. The transversal distance was minimized as it could possibly be assuming that the two tips were aligned along the same streamline. Eventual corrosion of the copper electrode was removed by cleaning the tips regularly.

The probe works together with an electronic acquisition system. The electronic system is designed and developed in the Department of Electrical Engineering and Computers (Figure 4.2b). The acquisition software was developed in MatLAB GUI and the signals analysed in a personal computer Core i7 with 6 GB of RAM. The output of the system has two different modes: (1) Raw signal: the raw signal is function of the voltage. The voltage ranges from 0V (voltage in air) to 5V (voltage in water). (2) Threshold-signal: The signal obtained in (1) is transformed in a square-wave signal by implementing a single cut-off value. If the raw signal is lower than threshold, it is assumed that probe is in the air, otherwise the probe is in the water. The cut-off value of this experiment was fixed in a single threshold of 65% (Bung, 2012). The output signals can be exported to a *.csv file.

Experimentally, four points in-a-row on top of the tube (see Figure 4.2c) were measured. The dual-tip probe was initially positioned in the center of the transversal section and, between each measurement, the probe was calibrated only in the presence of water. The signal acquisition was performed during 30 seconds and the respective average void fraction calculated.



a) b) c)

Figure 4.2 – Dual-tip resistive probe: (a) detailed measures and (b) electronic acquisition system. (c) Set of four points measured experimentally with resistive probe.

4.2. Numerical Simulations

The mesh used in the numerical solution was created using the *blockMesh* utility of OpenFOAM^{TM}. It was generated a control volume with approximately 90 000 cells limited by four types of boundary faces: *inlet1, inlet2, walls* and *outlet* (Figure 4.3). Those boundary faces are dependent on their functional characteristics. The *inlet1* allows the mixture flow of air and water to enter in the domain; the *inlet2* only allows the entrance of water in the system; the *outlet* represents the atmosphere and allows the fluids to come out of the domain with zero pressure; the *walls* represent ‘no slip’ boundaries (i.e. the velocity is zero near the wall).

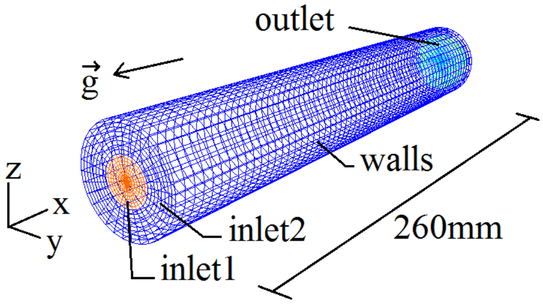


Figure 4.3 – Mesh created and boundary faces.

Overall, five numerical simulations were performed using the *interFoam* solver of OpenFOAM^{TM} v.2.1.1. These simulations differ in the turbulence model chosen. Table 4 summarises the turbulence models and the OpenFOAM^{TM} dictionaries required.

Table 4 – OpenFOAM^{TM} dictionaries required by the different turbulence models.

	laminar	*stand. k-ε*	*RNG k-ε*	*k-ω SST*	*Smagorinsky*
alpha1	✓	✓	✓	✓	✓
U	✓	✓	✓	✓	✓
p_rgh	✓	✓	✓	✓	✓
k		✓	✓	✓	
epsilon		✓	✓		
omega				✓	
nut		✓	✓	✓	
nuTilda		✓	✓	✓	
nuSgs					✓
B					✓

4.3. Results

Figure 4.4 illustrates the comparison between the experimentally measured void fraction and the numerical results. The average air concentration predicted using the flow meters (C_{air}=17.9%) is confirmed by the probe which measured approximately 18% (Figure 4.4). The experimental air concentration profile seems to be similar to the velocity profile along the transversal section of the pipe. The maximum velocity occurs in the centre, whereas zero velocities are found near the lateral walls. This result is reliable, since the areas with highest velocities transport more quantities of bubbles than the ones with lower velocities.

The simulations k-ε and k-ω SST exhibit a peak in the centre of the profile that is in absolute disagreement with the experimental result. The remaining simulations can predict the maximum value of air concentration, however, whereas the experimental profile is convex, the numerical profile is characterized by two concave sides.

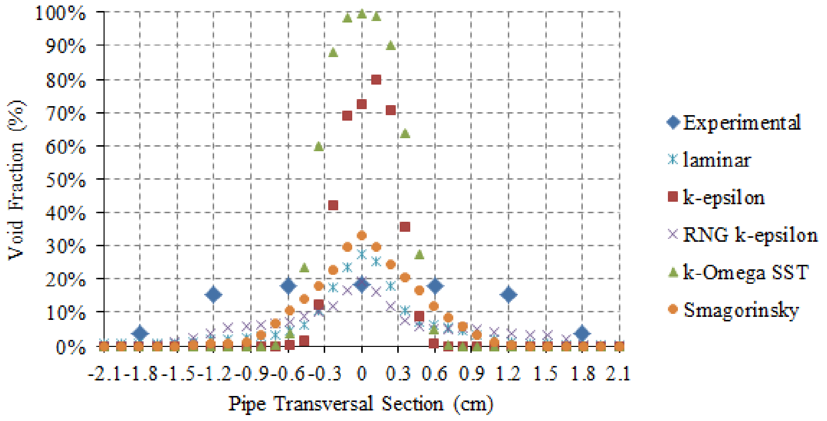


Figure 4.4 – Air concentration profiles on the top of the vertical tube.

Figure 4.5 shows 2D average profiles of air concentration in top of the PVC tube. The hypothetic 2D experimental profile was based on Figure 4.4 considering a circular distribution of the air concentrations. Figure 4.6 presents the vertical profiles of air concentration in the middle section of the tube. The white lines represent the limit of C_{air} = 5%.

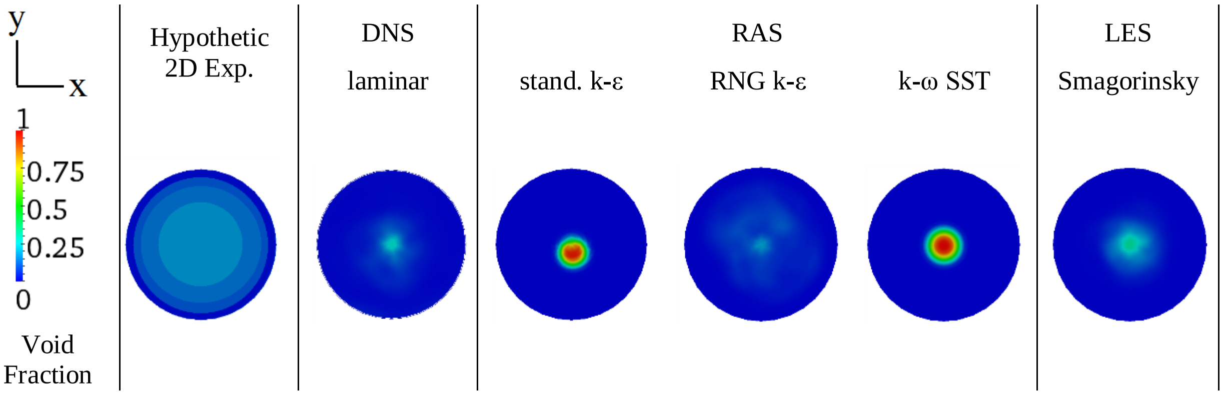


Figure 4.5 – 2D average profiles of air concentration on top of the vertical pipe.

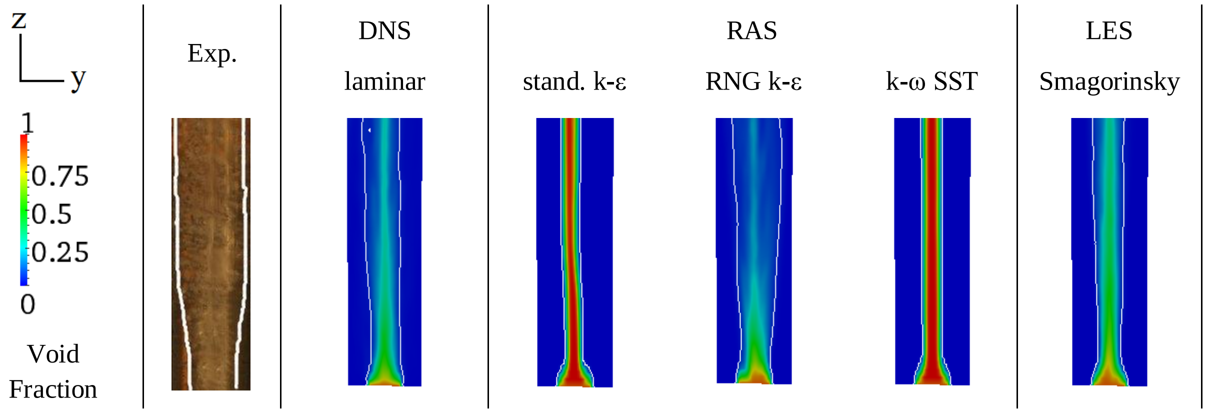


Figure 4.6 – 2D average profiles of air concentration in vertical plane of the pipe. The white lines limit the values of C_{air}=5%.

Figure 4.5 and Figure 4.6 can be analysed together. The preliminary results based on the qualitative profiles show that the simulations more distant of the experimental profile are those that are applied to the turbulence models standard k-ε and k-ω SST. In these cases, the mixture between the air and the water did not occur and a thick spray can be found.

In the author's perspective, the model Smagorinsky is the one that better represents the real behaviour of the flow in terms of air concentration profile and dispersion. Moreover, the Smagorinsky Model can provide superior results calibrating the Smagorinsky constant (C_S) by changing the OpenFOAM^{TM} coefficients C_ε and C_k (Lopes *et al.*, 2013).

5. FUTURE WORK

The applicability limits of the *interFoam* in hydraulic structures with free-surface and highly aerated flows require more studies. The numerical results will be validated with experimental data acquired in real-scale facilities.

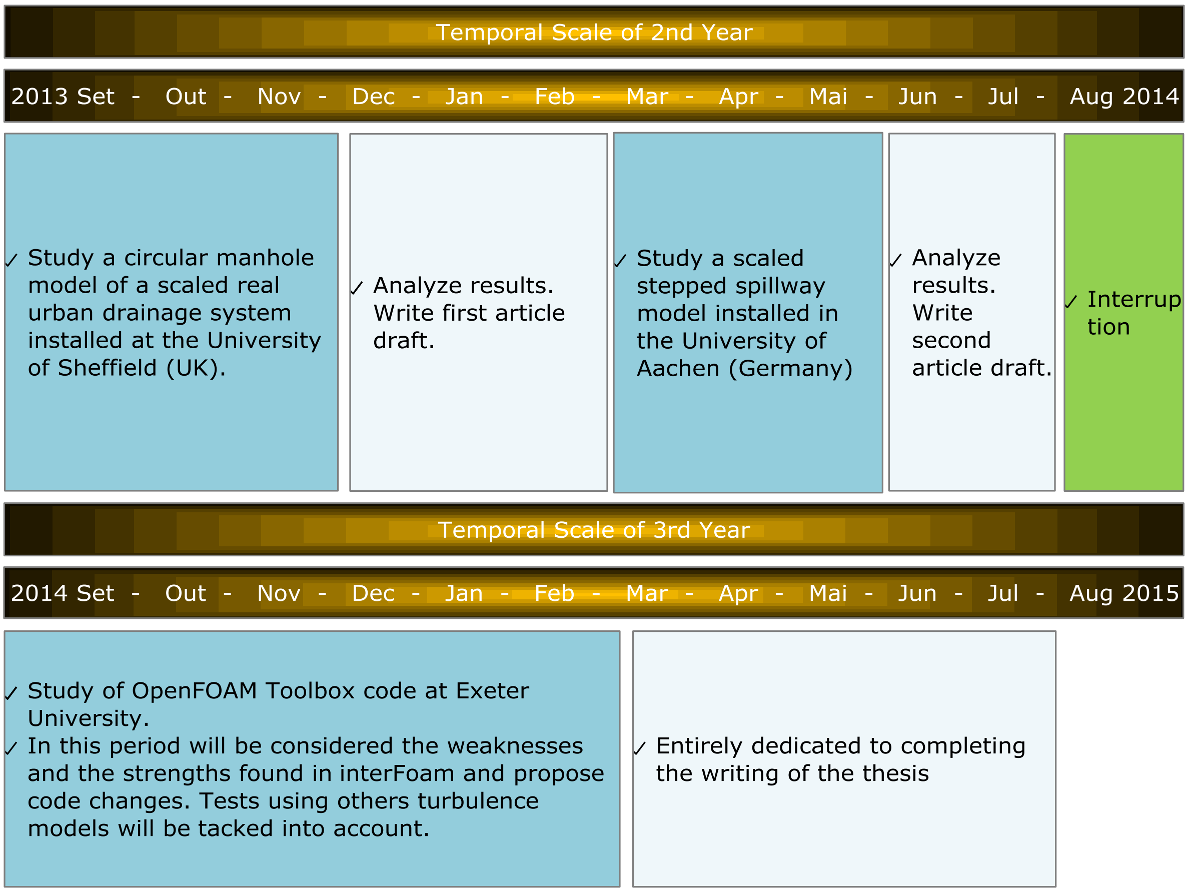
It is foreseeable that, the next two years of the doctoral program should follow three periods. In the second year of this doctoral program and, more specifically, in the first trimester, (September 2013-November 2013) the Author will study a circular manhole model of a scaled real urban drainage system installed at the University of Sheffield under the supervision of Dr. James Shucksmith. This study will measure the ability of the OpenFOAM^{TM} Toolkit to predict free-surface on circular manholes. The model will be validated using a surface PIV (to measure the surface velocity field) and a series of pressure transducers (to measure the surface flow depth around the manhole) will be made.

Also in the second year, in the trimester from March 2014 to May 2014, the Author will study a stepped spillway model under the supervision of Dr. Daniel Bung at the University of Applied Sciences - Aachen (Germany). The study will focus on the aerated and non-aerated part with the influence of macro-roughness.

The third year of this Doctoral Program is dedicated to improve the results obtained with the *interFoam*. During a period of six months, from September 2014 to March 2015, the Author will be under the supervision of Dr. Gavin Tabor, a specialist in the application of CFD to water resources engineering and a contributor to the OpenFOAM^{TM} code.

The proposed work should follow the flowchart presented on Table 5.

Table 5 – Flowchart presenting the future work.



Temporal Scale of 2nd Year

2013 Set - Out - Nov - Dec - Jan - Feb - Mar - Apr - Mai - Jun - Jul - Aug 2014

✓ Study a circular manhole model of a scaled real urban drainage system installed at the University of Sheffield (UK).

✓ Analyze results. Write first article draft.

✓ Study a scaled stepped spillway model installed in the University of Aachen (Germany)

✓ Analyze results. Write second article draft.

✓ Interruption

Temporal Scale of 3rd Year

2014 Set - Out - Nov - Dec - Jan - Feb - Mar - Apr - Mai - Jun - Jul - Aug 2015

✓ Study of OpenFOAM Toolbox code at Exeter University.
✓ In this period will be considered the weaknesses and the strengths found in interFoam and propose code changes. Tests using others turbulence models will be tacked into account.

✓ Entirely dedicated to completing the writing of the thesis

6. SUMMARY AND CONCLUSIONS

This research work started with the study of the air-water flow and air-entrainment phenomena. The turbulence at the free-surface of high-velocity flows can be sufficient to trap large amounts of air that changes the properties and the behaviour of the water flow (e.g. density, direction, etc.). Several theoretical and experimental studies have been conducted along the past decades to characterize the air entrained into the flow and define the pros and cons of this interaction. The increase of the computational technology and the accuracy of the CFD models allowed the study of fluid dynamics to become faster and cheaper in a wide range of hydraulic structures. The spillways, manholes and gullies are hydraulic structures of flood control where the air-entrainment plays an important role.

The OpenFOAM^{TM} is a free open-source CFD toolbox largely used in multiphase flows. The *interFoam* is one of these solvers, which employs the VOF method to predict the free-surface. Despite the accuracy demonstrated by the solver in some work where the Author was involved in (Carvalho *et al.*, 2012; Lopes *et al.*, 2012, 2013), the VOF presents some limitations when the percentage of air is higher than 10-15%. Proving this comment is the test case presented in this Thesis project in a vertical filled pipe. Therefore, the applicability limits of this solver in hydraulic structures with free-surface and highly aerated flows require more studies. The *interFoam* with diverse turbulence models in different grid sizes is the starting point of this study.

The next year will be dedicated to the validation of the *interFoam* solver in hydraulic structures. This solver will be used to simulate flood control devices and the results validated with physical models built in the Universities of Sheffield and Aachen. The study will focus primarily on the air-water interaction and how well the existent numerical model is able to capture it. In the third year the Author will suggest changes to the *interFoam* solver in order to improve the numerical-simulation of air-concentration in hydraulic structures under turbulent conditions, the main purpose of the final Thesis.

7. FIRST YEAR WORK

Starting in September 2012, the Author published the following works:

Lopes, P., Leandro, J., Carvalho, R.F., and Martins, R., 2012. Hydraulic behaviour of a gully under surcharge conditions. *In: 9th International Conference on Urban Drainage Modelling*. Belgrade. Serbia: USB flash drives, ISBN 978-86-7518-156-9.

Carvalho, R., Páscoa, P., Leandro, J., Abreu, J., **Lopes, P.**, Quinteiro, R., and de Lima, J.L.M.P., 2013. Experimental investigation of the linking element gully - drop manhole. *In: Proceedings of 35th IAHR World Congress 2013*. Chengdu, China.

Lopes, P., Leandro, J., Carvalho, R.F., Páscoa, P., and Martins, R., 2013. Numerical and experimental investigation of a gully under surcharge conditions. *Urban Water Journal [paper accepted for publication on 31/07/2013].*

8. REFERENCES

Afshar, N.R., Asawa, G.L., and Raju, K.G.R., 1994. Air concentration distribution in self-aerated flows. *Journal of Hydraulic Research*, 32 (4), pp. 623–631.

Aris, R., 1989. *Vectors, tensors and the basic equations of fluid mechanics*. Dover Publications.

Ashgriz, N. and Poo, J.., 1991. FLAIR: Flux line-segment model for advection and interface reconstruction. *Journal of Computational Physics*, 93 (2) DOI: 10.1016/0021-9991(91)90194-P, pp. 449–468.

Berberović, E., van Hinsberg, N.P., Jakirlić, S., Roisman, I., and Tropea, C., 2009. Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. *Physical Review E*, 79 (3) DOI: 10.1103/PhysRevE.79.036306, pp. 1–15.

Borges, J.E., Pereira, N.H.C., Matos, J., and Frizell, K.H., 2010. Performance of a combined three-hole conductivity probe for void fraction and velocity measurement in air–water flows. *Experiments in Fluids*, 48 (1) DOI: 10.1007/s00348-009-0699-1, pp. 17–31.

Bormann, K., 1968. *Der Abfluss in Schussrinnen unter Berucksichtigung der Luftaufnahme [Chute flow under consideration of air entrainment]*. Beri cht Nr. 13 der Versuchsansta lt fur Wasserbau, TH München.

Brackbill, J.U., Kothe, D.B., and Zemach, C., 1991. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100, pp. 335–354.

Bung, D.B., 2012. Sensitivity of Phase Detection Techniques in Aerated Chute Flows To Hydraulic Design Parameters. *In*: *2nd IAHR Europe Conference*. Munich.

Bung, D.B. and Schlenkhoff, A., 2010. Self-Aerated Skimming Flow On Embankment Stepped Spillways - The Effect Of Additional Micro-roughness On Energy Dissipation And Oxygen Transfer. *In*: *IAHR European Congress*.

Cain, P., 1978. Measurements within Self-Aerated Flow on a Large Spillway. *Research Report No. 78-18*. University of Canterbury, New Zealand.

Caretto, L.S., Curr, R.M., and Spalding, D.B., 1972. Two numerical methods for three-dimensional boundary layers. *Computer Methods in Applied Mechanics and Engineering*, 1 (1) DOI: 10.1016/0045-7825(72)90020-5, pp. 39–57.

Carvalho, R.F., 2002. Acções Hidrodinâmicas em Estruturas Hidráulicas: Modelação Computacional no Ressalto Hidráulico. *PhD Thesis*. Universidade de Coimbra, Portugal.

Carvalho, R.F., Leandro, J., David, L.M., Martins, R., and Melo, N., 2011. Numerical Research of the Inflow Into Different Gullies Outlets. *In*: *Computing and Control for the Water Industry*. Exeter, UK.

Carvalho, R.F., Leandro, J., Martins, R., and Lopes, P., 2012. Numerical study of the flow behaviour in a gully. *In*: *4th IAHR International Symposium on Hydraulic Structures*. Porto. Portugal.

Černe, G., Petelin, S., and Tiselj, I., 2001. Coupling of the Interface Tracking and the Two-Fluid Models for the Simulation of Incompressible Two-Phase Flow. *Journal of Computational Physics*, 171 (2) DOI: 10.1006/jcph.2001.6810, pp. 776–804.

Chanson, H., 1988. Study of Air Entrainment and Aeration Devices on Spillway Model. *PhD Thesis*. Department of Civil Engineering, University of Canterbury, Christchurch, New Zealand.

Chanson, H., 1996. *Air Bubble Entrainment In Free-Surface Turbulent Shear Flows*. Academic Press - Elsevier Ltd.

Chanson, H., 2002. Air-Water Flow Measurements with Intrusive , Phase-Detection Probes : Can We Improve Their Interpretation? *Journal of Hydraulic Engineering*, 128 (3) DOI: 110.1061/(ASCE)0733-9429(2002)128:3(252), pp. 252–255.

Chanson, H., 2006. Hydraulics of skimming fows on stepped chutes: The effects of infow conditions? *Journal of Hydraulic Research*, 44 (1) DOI: 10.1080/00221686.2006.9521660, pp. 51–60.

Chanson, H., 2007. Bubbly flow structure in hydraulic jump. *European Journal of Mechanics - B/Fluids*, 26 (3) DOI: 10.1016/j.euromechflu.2006.08.001, pp. 367–384.

Chanson, H. and Brattberg, T., 2000. Experimental study of the air-water shear flow in a hydraulic jump. *International Journal of Multiphase Flow*, 26, pp. 583–607.

Chanson, H. and Toombes, L., 2002. Air-water flows down stepped chutes: turbulence and flow. *International Journal of Multiphase Flow*, 28 (11) DOI: 0301- 9322, pp. 1737–1761.

Cheng, X., Chen, Y., and Luo, L., 2006. Numerical simulation of air-water two-phase flow over stepped spillways. *Science in China Series E: Technological Sciences*, 49 (6) DOI: 10.1007/s10288-006-2029-2, pp. 674–684.

Cummings, P.D. and Chanson, H., 1997. Air Entrainment in the Developing Flow Regio of Plunging Jets - Part 1: Theoretical Development. *Journal of Fluids Engineering*, 119, pp. 597–602.

Daly, B.J., 1969. A technique for including surface tension effects in hydrodynamic calculations. *Journal of Computational Physics*, 4 (1) DOI: 10.1016/0021-9991(69)90042-4, pp. 97–117.

DeBar, R.B., 1974. Fundamentals of the KRAKEN code. [Eulerian hydrodynamics code for compressible nonviscous flow of several fluids in two-dimensional (axially symmetric) region]. *Technical Report UCID-17366*.

Deshpande, S.S., Trujillo, M.F., Wu, X., and Chahine, G., 2012. Computational and experimental characterization of a liquid jet plunging into a quiescent pool at shallow inclination. *International Journal of Heat and Fluid Flow*, 34 DOI: 10.1016/j.ijheatfluidflow.2012.01.011, pp. 1–14.

Deswal, S. and Verma, D.V.S., 2007. Air-Water Oxygen Transfer with Multiple Plunging Jets. *Water Quality Research Journal*, 42 (4), pp. 295–302.

Djordjević, S., Saul, A.J., Tabor, G.R., Blanksby, J., Galambos, I., Sabtu, I., and Sailor, G., 2011. Experimental and numerical investigation of interactions between above and below ground drainage systems. *In*: *12th International Conference on Urban Drainage*. Porto Alegre/RS, Brazil.

Eghbalzadeh, A. and Javan, M., 2012. Comparison of Mixture and VOF Models for Numerical Simulation of Air–entrainment in Skimming Flow over Stepped Spillways. *Procedia Engineering*, 28 (2011) DOI: 10.1016/j.proeng.2012.01.786, pp. 657–660.

Enright, D., Fedkiw, R., Ferziger, J., and Mitchell, I., 2002. A Hybrid Particle Level Set Method for Improved Interface Capturing. *Journal of Computational Physics*, 183 (1) DOI: 10.1006/jcph.2002.7166, pp. 83–116.

Falvey, H.T., 1980. *Air-water flow in hydraulic structures*. Water Resources Technical Publication.

Ferziger, J.H. and Peric, M., 2002. *Computational Methods For Fluid Dynamics*. 3rd ed. Spinger.

Galambos, I., 2012. Improved Understanding of Performance of Local Controls Linking the above and below Ground Components of Urban Flood Flows. *PhD Thesis*. University of Exeter, UK.

Gómez, M. and Russo, B., 2009. Hydraulic Efficiency of Continuous Transverse Grates for Paved Areas. *Journal of Irrigation and Drainage Engineering*, 135 (2) DOI: 10.1061/(ASCE)0733-9437(2009)135:2(225), pp. 225–230.

Gonzalez, C.A., Takahashi, M., and Chanson, H., 2005. Effects of Step Roughness in Skiming Flows: An Experimental Study. *Reseach Report CE160, University of Queensland*.

Gonzalez, C.A., Takahashi, M., and Chanson, H., 2008. An experimental study of effects of step roughness in skimming flows on stepped chutes. *Journal of Hydraulic Research*, 46 (1), pp. 24–35.

Grosshans, H., 2011. Development of a combined VOF-LFP method to simulate two-phase flows in various regimes. *In*: *7th International Symposium on Turbulence and Shear Flow Phenomena (TSFP-7)*.

Harlow, F.H. and Welch, J.E., 1965. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *The Physics of FLuids*, 8 (12), pp. 2182–2189.

Hirsch, C., 1988. *Numerical Computation of Internal and External Flows Vol.1 - Fundamentals of Numerical Discretization*. JohnWiley & Sons, Inc.

Hirt, C.W., 2003. Modeling Turbulent Entrainment of Air at a Free Surface. *Flow Science, Inc.*

Hirt, C.W. and Nichols, B.D., 1981. Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. *Journal of Computational Physics*, 39 DOI: 10.1016/0021-9991(81)90145-5, pp. 201–225.

Hyman, J.M., 1984. Numerical Methods for Tracking Interfaces. *Physica D: Nonlinear Phenomena*, 12 (1-3) DOI: 10.1016/0167-2789(84)90544-X, pp. 396–407.

Issa, R.I., 1985. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62 (1) DOI: 10.1016/0021-9991(86)90099-9, pp. 40–65.

Jasak, H., 1996. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows. *PhD Thesis*. Imperial College, London, UK.

Keller, R.J., Lai, K.K., and Wood, I.R., 1974. Developing region in self-aerated flows. *Journal of the Hydraulics Division*, 100 (4), pp. 553–568.

Keller, R.J. and Rastogi, A.K., 1977. Design Chart for Predicting Critical Point on Spillways. *Journal of the Hydraulics Division*, 103 (12), pp. 1417–1429.

Kendil, F.Z., Salah, A.B., and Mataoui, A., 2010. Assessment of three turbulence model performances in predicting water jet flow plunging into a liquid pool. *Nuclear Technology and Radiation Protection*, 25 (1) DOI: 10.2298/NTRP1001013Z, pp. 13–22.

Kiger, K.T. and Duncan, J.H., 2012. Air-Entrainment Mechanisms in Plunging Jets and Breaking Waves. *Annual Review of Fluid Mechanics*, 44 (1) DOI: 10.1146/annurev-fluid-122109-160724, pp. 563–596.

Kolmogorov, A.N., 1941. The local structure of isotropic turbulence in an incompressible viscous fluid. *Doklady Akademii Nauk SSSR*, 30, pp. 301–305.

Launder, B.E. and Spalding, D.B., 1974. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3 (2) DOI: 10.1016/0045-7825(74)90029-2, pp. 269–289.

Leandro, J., Carvalho, R.F., Chachereau, Y., and Chanson, H., 2012. Estimating void fraction in a hydraulic jump by measurements of pixel intensity. *Exp Fluids*, 52 DOI: 10.1007/s00348-011-1257-1, pp. 1307–1318.

Lobosco, R.J., Schulz, H.E., and Simões, A.L.A., 2011. Analysis of Two Phase Flows on Stepped Spillways. *In*: *Hydrodynamics - Optimizing Methods and Tools*.

Lopes, P., Leandro, J., Carvalho, R.F., and Martins, R., 2012. Hydraulic behaviour of a gully under surcharge conditions. *In*: *9th International Conference on Urban Drainage Modelling*. Belgrade. Serbia: USB flash drives, ISBN 978-86-7518-156-9.

Lopes, P., Leandro, J., Carvalho, R.F., Páscoa, P., and Martins, R., 2013. Numerical and experimental investigation of a gully under surcharge conditions. *Urban Water Journal [paper accepted for publication on 31/07/2013]*.

Lubin, P., Glockner, S., Kimmoun, O., and Branger, H., 2011. Numerical study of the hydrodynamics of regular waves breaking over a sloping beach. *European Journal of Mechanics - B/Fluids*, 30 (6) DOI: 10.1016/j.euromechflu.2011.01.001, pp. 552–564.

Martins, R., Leandro, J., and Carvalho, R.F., 2012. Hydraulic behaviour of a gully : numerical vs . experimental. *In*: *9th International Conference on Urban Drainage Modelling*. Belgrade. Serbia: USB flash drives, ISBN 978-86-7518-156-9.

Matos, J., 1999. Emulsionamento de ar e dissipação de energia do escoamento em descarregadores em degraus. *PhD Thesis*. Instituto Superior Técnico de Lisboa, Portugal.

McKeogh, E.A., 1978. A Study of Air Entrainment Using Plunging Water Jets. *PhD Thesis*. Queen's University of Belfast.

Menter, F.R., 1993. Zonal two-equation k-ω turbulence model for aerodynamic flows. *AIAA Paper*, pp. 1993–2906.

Moin, P. and Mahesh, K., 1998. Direct Numerical Simulation: A Tool in Turbulence Research. *Annual Review of Fluid Mechanics*, 30 (1) DOI: 10.1146/annurev.fluid.30.1.539, pp. 539–578.

Moraga, F.J., Carrica, P.M., Drew, D.A., and Lahey, R.T., 2005. The Modeling Of Air Entrainment Processes For Surface Ships. *In*: *VIII Congreso Argentino de Mecánica Computacional*.

Mossa, M. and Tolve, U., 1998. Flow visualization in bubbly two-phase hydraulic jump. *Journal of Fluids Engineering*, 120 (1), pp. 160–165.

Murzyn, F. and Chanson, H., 2008. Experimental assessment of scale effects affecting two-phase flow properties in hydraulic jumps. *Experiments in Fluids,* 45 (3) DOI: 10.1007/s00348-008-0494-4, pp. 513–521.

Nagash, B.W., 1994. Void fraction measurement techniques for gas- liquid bubbly flows in closed conduits: a literature review. *Proceedings of hydraulic engineering conference, ASCE, Buffalo*, pp. 278–288.

Noh, W.F. and Woodward, P., 1976. SLIC (Simple Line Interface Calculation). *In*: A.I. Vooren and P.J. Zandbergen, eds. *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 – July 2*. Berlin, Heidelberg: Springer Berlin Heidelberg.

OpenFOAM, 2011. OpenFOAM v2.1.0: Multiphase Modelling [online]. Available from: http://www.openfoam.org/version2.1.0/multiphase.php [Accessed24 Apr 2013].

OpenFOAM, 2012. OpenFOAM - Programmer's Guide v.2.1.1.

Osher, S. and Sethian, J., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79 (1) DOI: 10.1016/0021-9991(88)90002-2, pp. 12–49.

Pantakar, S. V., 1980. *Numerical heat transfer and fluid flow: Computational methods in mechanics and thermal science*. Taylor & Francis.

Páscoa, P., Leandro, J., and Carvalho, R.F., 2013. Characterization of the flow in a gully: average velocity and air entrainment. *In*: *International Workshop on Hydraulic Design of Low-Head Structures (IWLHS 2013)*. Aachen, Germany, 20-22 February 2013.

Pope, S.B., 2000. *Turbulent flows*. Cambridge University Press.

Pothof, I.W.M., 2011. Co-current air-water flow in downward sloping pipes. *PhD Thesis*. Delft University of Technology.

Rajaratnam, N., 1962. An experimental study of air entrainment characteristics of the hydraulic jump. *Journal of The Institution of Engineers (India)*, 42 (7), pp. 247–273.

Rajaratnam, N., 1990. Skimming flow in stepped spillways. *Journal of Hydraulic Engineering, ASCE,* 116 (4), pp. 587–591.

Resch, F.J. and Leutheusser, H.J., 1972. Reynolds stress measurements in hydraulic jumps. *Journal of Hydraulic Research*, 10 (4) DOI: 10.1080/00221687209500033, pp. 409–430.

Resch, F.J., Leutheusser, H.J., and Alemum, S.., 1974. Bubbly two-phase flow in hydraulic jump. *Journal of Hydraulic Division*, 100 (1), pp. 137–149.

Rider, W.J. and Kothe, D.B., 1997. Reconstructing Volume Tracking. *Journal of Computational Physics*, 141 (2) DOI: 10.1006/jcph.1998.5906, pp. 112–152.

Rodríguez-Rodríguez, J., Marugán-Cruz, C., Aliseda, A., and Lasheras, J.C., 2011. Dynamics of large turbulent structures in a steady breaker. *Experimental Thermal and Fluid Science*, 35 (2) DOI: 10.1016/j.expthermflusci.2010.09.012, pp. 301–310.

Romagnoli, M., Carvalho, R.F., and Leandro, J., 2013. Turbulence characterization in a gully with reverse flow. *Journal of Hydraulic Engineering-ASCE*, DOI: 10.1061/(ASCE)HY.1943-7900.0000737.

Rusche, H., 2002. Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions. *PhD Thesis*. University of London.

Ryu, Y., Chang, K.-A., and Lim, H.-J., 2005. Use of bubble image velocimetry for measurement of plunging wave impinging on structure and associated greenwater. *Meas. Sci. Technol.*, 16, pp. 1945–1953.

Silva, L., 2008. Desenvolvimento de metodologias para simulação de escoamentos polidispersos usando código livre. *PhD Thesis*. COPPE, Universidade Federal do Rio de Janeiro, Brasil.

Silva, L., Damian, R., and Lage, P., 2008. Implementation and analysis of numerical solution of the population balance equation in CFD packages. *Computers & Chemical Engineering*, 32 (12) DOI: 10.1016/j.compchemeng.2008.03.007, pp. 2933–2945.

Silva, L. and Lage, P., 2007. Implementation of an Eulerian Multi-phase Model in OpenFOAM and its Application to Polydisperse Two-Phase Flows. *In*: *OpenFOAM International Conference 2007*. pp. 1–16.

Simões, A.L., Schulz, H.E., Lobosco, R.J., and Porto, R.M., 2012. Stepped Spillways: Theoretical , Experimental and Numerical Studies. *In*: *Hydrodynamics - Natural Water Bodies*.

Smagorinsky, J., 1963. General Circulation Experiments with the Primitive Equations. *Weather Review*, 91 (3) DOI: 10.1126/science.27.693.594, pp. 99–164.

Stovin, V.R., Guymer, I., and Lau, S.D., 2008. Approaches to validating a 3D CFD manhole model. *In*: *11th International Conference on Urban Drainage*. Edinburgh, Scotland, UK.

Straub, L.G. and Anderson, A.G., 1958. Experiments on self-aerated flow in open channels. *Journal of the Hydraulics Division*, 85 (11), pp. 119–121.

Tang, H. and Wrobel, L.C., 2005. Modelling the interfacial flow of two immiscible liquids in mixing processes. *International Journal of Engineering Science*, 43 (15-16) DOI: 10.1016/j.ijengsci.2005.03.011, pp. 1234–1256.

Tøge, G.E., 2012. The significance of Froude number in vertical pipes – A CFD study. *MSc Thesis*. University of Stavanger, Norway.

Tongkratoke, A., Chinnarasri, C., Pornprommin, A., Dechaumphai, P., and Juntasaro, V., 2009. Non-linear turbulence models for multiphase recirculating free-surface flow over stepped spillways. *International Journal of Computational Fluid Dynamics*, 23 (5) DOI: 10.1080/10618560902886882, pp. 401–409.

Trujillo, M.F., Hsiao, C., Choi, J., Paterson, E.G., Chahine, G.L., and Peltier, L.J., 2007. Numerical and Experimental Study of a Horizontal Jet Below a Free Surface. *In*: *9th International Conference on Numerical Ship Hydrodynamics*. pp. 15.

Ubbink, O., 1997. Numerical prediction of two fluid systems with sharp interfaces. *PhD Thesis*. Imperial College of Science, UK.

Ubbink, O. and Issa, R.I., 1999. A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes. *Journal of Computational Physics*, 153 (1) DOI: 10.1006/jcph.1999.6276, pp. 26–50.

Vallier, A., Revstedt, J., and Nilsson, H., 2011. Procedure for the break-up of cavitation sheet. *In*: *4th International Meeting on Cavitation and Dynamic Problems in Hydraulic Machinery and Systems*. Belgrade, Serbia.

Vanka, S.., 1986. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journal of Computational Physics*, 65 (1) DOI: 10.1016/0021-9991(86)90008-2, pp. 138–158.

Versteeg, H.K. and Malalasekera, W., 1995. *An Introduction to Computational Fluid Dynamics - The Finite Volume Method*. 2nd ed. Longman Scientific & Technical. Pearson Education.

Viparelli, M., 1953. The flow in the flume with 1:1 slope. *In*: *International Association for Hidraulic Research, Joint Conference, Minneapolis, Minesota*. pp. 415–423.

Volkart, P., 1980a. Self-aerated Flow in Steep, Partially Filled Pipes. *Journal of the Hydraulics Division*, 108 (9), pp. 1029–1046.

Volkart, P., 1980b. The mechanism of air bubble entrainment in self-aerated flow. *International Journal of Multiphase Flow*, 6 (5) DOI: 10.1016/0301-9322(80)90003-8, pp. 411–423.

Volkart, P. and Rutschmann, P., 1984. *Air Entrainment Devices (Air Slots)*. Mitteilungen der Versuchsanstalt fur Wasserbau, Hydrologie und Glaziologie.

Weller, H., 2002. Derivation modelling and solution of the conditionally averaged two-phase flow equations. *Technical Report TR/HGW/02, Nabla Ltd.*

Weller, H.G., Tabor, G.R., Jasak, H., and Fureby, C., 1998. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12 (6), pp. 620–631.

Wilcox, D.C., 1988. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal*, 26 (11) DOI: 10.2514/3.10041, pp. 1299–1310.

Wilcox, D.C., 2008. Formulation of the k-w Turbulence Model Revisited. *AIAA Journal*, 46 (11), pp. 2823–2838.

Wood, I.R., 1991. *Air entrainment in free-surface flows*. Hydraulic structures design manual 4, Balkema.

Wood, I.R., Ackers, P., and Loveless, J., 1983. General method for critical point os spillways. *ASCE*, 2/09.

Xiangju, C. and Xuewei, C., 2012. Progress in numerical simulation of high entrained air-water two-phase flow. *In*: *Third International Conference on Digital Manufacturing & Automation*. Guilin, China, July 31 - Aug. 2, 2012, pp. 626–629.

Yakhot, V., Thangam, S., Gatski, T.B., Orszag, S.A., and Speziale, C.G., 1991. Development of Turbulence Models for Shear Flows by a Double Expansion technique. Engineering. pp. 1–27.

Youngs, D.L., 1984. An Interface Tracking Method for a 3D Eulerian Hydrodynamics Code. *Thecnical Report 44/92/35*.

Zhao, C.-H., Zhu, D.Z., and Rajaratnam, N., 2008. Computational and Experimental Study of Surcharged Flow at a 90° Combining Sewer Junction. *Journal of Hydraulic Engineering*, 134 (6), pp. 688–700.