

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/280577306>

An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces

Thesis · March 2013

DOI: 10.13140/RG.2.1.3182.8320

CITATIONS
108

READS
2,270

1 author:



Santiago Márquez Damián
National Scientific and Technical Research Council

50 PUBLICATIONS 284 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:

- Project Development of a robust computational solver for Fluidized Bed applications [View project](#)

- Project VOF-PLIC solvers - Granular Flow - Mesh dynamics [View project](#)

UNIVERSIDAD NACIONAL DEL LITORAL



An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces

Santiago Márquez Damián

FICH

FACULTAD DE INGENIERIA
Y CIENCIAS HIDRICAS

INTEC

INSTITUTO DE DESARROLLO TECNOLOGICO
PARA LA INDUSTRIA QUIMICA

Tesis de Doctorado **2013**



UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas
Instituto de Desarrollo Tecnológico para la Industria Química

**AN EXTENDED MIXTURE MODEL
FOR THE
SIMULTANEOUS TREATMENT OF
SHORT AND LONG SCALE INTERFACES**

Santiago Márquez Damián

Tesis remitida al Comité Académico del Doctorado
como parte de los requisitos para la obtención
del grado de
DOCTOR EN INGENIERIA
Mención Mecánica Computacional
de la
UNIVERSIDAD NACIONAL DEL LITORAL

2013

Comisión de Posgrado, Facultad de Ingeniería y Ciencias Hídricas, Ciudad Universitaria,
Paraje "El Pozo", S3000, Santa Fe, Argentina.

**AN EXTENDED MIXTURE MODEL
FOR THE
SIMULTANEOUS TREATMENT OF
SHORT AND LONG SCALE INTERFACES**

by

Santiago Márquez Damián

Dissertation submitted to the Postgraduate Department of the

FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS

of the

UNIVERSIDAD NACIONAL DEL LITORAL

in partial fulfillment of the requirements for the degree of

Doctor en Ingeniería - Mención Mecánica Computacional

2013

Author Legal Declaration

This dissertation have been submitted to the Postgraduate Department of the *Facultad de Ingeniería y Ciencias Hídricas* in partial fulfillment of the requirements for the degree of *Doctor en Ingeniería - Mención Mecánica Computacional* of the *Universidad Nacional del Litoral*. A copy of this document will be available at the University Library and it will be subjected to the Library's legal normative.

All this work has been developed using free-software, including calculations, figures and typesetting except where is expressly indicated.

All figures are property of the author except where is expressly indicated.

Some parts of the work presented in this thesis were and could be published in referred journals and annals of congresses.

Santiago Márquez Damián

© Copyright by
Santiago Márquez Damián
2013

A mi esposa e hijos,
por su compañía y paciencia.

A mis padres, quienes me mostraron que
la fe y la razón son parte de un
gran conjunto.

A mi hermano José Ignacio,
la persona que la vida me dio
para compartir la pasión por la ciencia.

A mi hermana Ana Paula,
por ayudarme a entender las cosas
desde otro punto de vista.

AN EXTENDED MIXTURE MODEL FOR THE SIMULTANEOUS TREATMENT OF SHORT AND LONG SCALE INTERFACES

Thesis advisors:

Norberto M. Nigro INTEC (CONICET-UNL)
Facultad de Ingeniería y Ciencias Hídricas - UNL
Mario A. Storti INTEC (CONICET-UNL)
Facultad de Ingeniería y Ciencias Hídricas - UNL

Evaluation committee members:

Dr. Patricio Bohórquez Rodríguez de Medina
Dr. Mariano I. Cantero
Dr. Jorge D'Elia
Dr. Axel E. Larreteguy

FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS

UNIVERSIDAD NACIONAL DEL LITORAL

2013

Acknowledgements

This thesis has received financial support from *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET), *Universidad Nacional del Litoral* (UNL), *Agencia Nacional de Promoción Científica y Tecnológica* (ANPCyT) and *Fundação de Amparo à Pesquisa do Estado de São Paulo* (FAPESP). The research work was carried out at the *Centro Internacional de Métodos Computacionales en Ingeniería* (CIMEC) of the *Instituto de Desarrollo Tecnológico para la Industria Química* (INTEC), dependent of UNL and CONICET and at the *Instituto de Ciências Matemáticas e de Computação* (ICMC) dependent of Universidade de São Paulo, Brasil.

An special acknowledgement is given to all the Free Software community and particularly to the Debian, OpenFOAM®, octave, KDE, Gnome and LaTEX projects. I'm also in great debt with the cfd-online.com OpenFOAM®'s community forum for sharing all your collective knowledge freely, in special with Dr. Alberto Passalacqua.

I would like to thank my advisors Norberto M. Nigro and Mario A. Storti for their guidance and confidence. This work hadn't been possible without the support and companionship from all the people in the CIMEC's informatics laboratory.

Abstract

The study of multi-phase models is a field of great interest in industry and academia. Multi-phase flows are present in hydraulics, petrochemical industry, oceanography, siderurgy, atomic energy and many other human activities. This field is far from being completely understood and the available tools are still in a developing stage. Nowadays the only general models for this kind of problems are the Direct Numerical Simulation or other models based in the physics of fluids. In this scenario, the aim of this thesis is to develop a new model based on the Volume of Fluid method and the Mixture Model in order to solve multi-phase flows with different interface scales and the transition among them. The interface scale is characterized by a measure of the grid, which acts as a geometrical filter and is related with the accuracy in the solution. This coupled model allows to reduce the grid requirements for a given accuracy. Having this objective in mind, a generalization of the Algebraic Slip Mixture Model is proposed to solve problems involving short and long scale interfaces in an unified framework. This model is implemented using the OpenFOAM® libraries to generate a state-of-the-art solver capable to solve large problems in High Performance Computing facilities. In addition several other contributions were made regarding to the conceptualization of the Mixture Model, the development of a new Riemann-free solver used to solve mixture problems and a set of tools to help in the implementation process.

Contents

Table of Contents	xv
List of Abbreviations	xv
List of Symbols	xvii
List of Figures	1
1 Introduction	1
1.1 Background and motivation	1
1.2 Objectives	5
1.3 Thesis outline	6
1.4 Contributions of the work	8
2 Cell-centered Finite Volume Method	11
2.1 Introduction	11
2.2 Disc. of a gral. advection-diffusion-reaction eqn.	12
2.3 Discrete version of differential operators	15
2.3.1 Convective operator	18
2.3.2 Diffusive operator	19
2.3.3 Temporal discretization	21
2.4 Boundary conditions	23
2.4.1 Basic boundary conditions	24
2.4.2 Derived boundary conditions	27
2.5 High Resolution Schemes Implementation	28
2.6 Flux Corrected Transport	32

2.6.1	Theoretical foundation	32
2.6.2	Zalesak's weighting factors	33
2.6.3	OpenFOAM®'s weighting factors	36
2.7	Discretization of the Navier-Stokes equations	38
2.7.1	Derivation of an equation for the pressure	39
2.7.2	The PISO algorithm for pressure-velocity coupling	41
3	ASMM characterization	45
3.1	Introduction	45
3.2	Theory fundamentals	46
3.2.1	Velocity of center of mass based formulation	46
3.2.2	The role of α_p equation	52
3.2.3	Velocity of center of volume or volumetric flux formulation	53
3.3	1D simplified formulation	56
3.3.1	One dimensional semi-analytical solution for sedimentation	57
3.3.2	One dimensional analytic solutions for the equation for α_p	63
4	Num. solutions for the phase fraction eqn.	73
4.1	Introduction	73
4.2	A Riemann-free solver with centered flux	74
4.2.1	First order Kurganov and Tadmor scheme	74
4.2.2	KT as a MUSCL based scheme	79
4.2.3	A Riemann-free solver with given centered flux	80
4.2.4	Multidimensional extension	83
4.3	TVD-FCT method	85
5	Volume of Fluid Method	87
5.1	Introduction	87
5.2	Derivation of VOF from ASMM	88
5.3	Solver implementation	90
5.4	Some classical examples	94
5.4.1	Rayleigh-Taylor instability	96
5.4.2	Dam Break	98

6 Numerical solvers for the mixture model	101
6.1 Introduction	101
6.2 Formulations for the ASMM	101
6.3 Solver implementation	105
6.4 Results	107
6.4.1 1D semi-analytic tests	107
6.4.2 Bubble column reactor	112
7 An Extended Mixture Model	117
7.1 Introduction	117
7.2 Motivation	119
7.3 Theoretical foundation	122
7.3.1 Černe criterion	123
7.3.2 Face gradient criterion	124
7.4 Solver implementation	126
7.5 Examples	126
7.5.1 Interaction of a bubble plume with the water surface . .	128
7.5.2 Bubble reactor with free surface capturing	133
7.5.3 Dam Break test with degassing	140
7.5.4 Rayleigh-Taylor instability	146
8 Conclusion	151
8.1 Conclusions of the work	151
8.2 Future work	154
A A note in OpenFOAM® programming	157
A.1 Introduction	157
A.2 Basic debugging	160
A.3 Advanced Debugging	161
A.3.1 System matrix	161
A.3.2 Mesh Search	163
A.3.3 Graphical debugging	165
A.4 Tests	166
A.4.1 Scalar Transport Test	166

A.4.2 Laplacian Test	170
A.4.3 Multiphase Test	172
B Resumen extendido en castellano	179
B.1 Modelo de mezcla extendido	179
B.1.1 Antecedentes	179
B.1.2 Desarrollo	185
Bibliography	189

List of Abbreviations

- 1D:** One dimension/dimensional
2D: Two dimensions/dimensional
3D: Three dimensions/dimensional
ASMM: Algebraic Slip Mixture Model
CARPT: Computer Automated Radiactive Particle Tracking
CD: Central Difference
CFD: Computational Fluid Dynamics.
CPU: Central Processing Unit.
DIC: Diagonal Incomplete-Cholesky
DNS: Direct Numerical Simulations
FCT: Flux Corrected Transport
FDM: Finite Difference Method
FDIC: Faster Diagonal Incomplete-Cholesky
FEM: Finite Element Method.
FVM: Finite Volume Method
GAMG: Geometric-Algebraic Multi-Grid
HRS: High Resolution Scheme
IC: Initial Condition/s
KT: Kurganov & Tadmor
LES: Large Eddy Simulation
l.h.s: Left hand side
LSI: Large-lenght-scale-interface
MULES: Multidimensional Universal Limiter with Explicit Solution
MUSCL: Monotone Upstream-centered Scheme for Conservation Laws
NVD: Normalized Variable Diagram

- PBiCG:** Preconditioned Bi-Conjugate Gradient
PCG: Preconditioned Conjugate Gradient
PDE: Partial Differential Equation/s
PISO: Pressure Implicit Split of Operators
RANS: Reynolds Averaged Navier-Stokes
SIMPLE: Semi-Implicit Method for Pressure-Linked Equations
SSI: Small-lengthscale-interface
r.h.s: Right hand side
TVD: Total Variation Diminishing
UD: Upwind Difference
VOF: Volume of Fluid

List of Symbols

- a : exponent for relative velocity law
 \vec{a} : generic vector magnitude/local velocity
 a' : a front speed
 a_N : off-diagonal coefficient of the discretization matrix
 a_P : diagonal coefficient of the discretization matrix
 \vec{a} : secondary phase acceleration
 b : boundary face centroid
 b' : b front velocity
 \vec{b} : relationship between velocity of center-of-mass and velocity, \vec{v}_m , of center-of-volume \vec{u}
 A : anti-diffusive flux
 $\overline{\overline{A}}_{ij}$: discretization matrix
 \vec{b} : generic vector magnitude/r.h.s vector for discretization system
 C_D : drag coefficient
 c : generic reactive constant, mass phase fraction
 c_k : mass phase fraction for k phase
Co: Courant number
 d_p : diameter of the particulated phase
 \vec{d}_{PN} : cell centers' difference vector
 \vec{d} : cell-centre to face-centre vector
 \vec{d}_n : boundary face normal vector starting in P
 f : face index
 f_{drag} : drag force
 f_x : generic weighting factor for interpolation
 F : face flux

- \tilde{F} : face flux approximation
 F^0 : face flux at the previous time-step
 F^C : corrected total flux
 F^H : high order total flux
 F^L : low order total flux
 F_f^L : face flux for the linear term
 F_f^{NL} : face flux for the non-linear term
 F_m : face flux for \vec{v}_m
 F_{qp} : face flux for \vec{v}_{qp}
 $F_{\rho m}$: mass face flux
 \vec{F} : flux
 F : total face flux, flux in scalar cases
 g, \vec{g} : gravitational acceleration
 g_b : fixed gradient at boundary b
 i : cell index
 j : face index
 k : generic phase index
 \vec{k} : correction vector for non-orthogonal decomposition
 l : left
 m : number of finite volumes/mixture sub-index
 \vec{M} : interface momentum interchange term
 N : sub-index for neighbor cell
 n : temporal step index
 n_i : number of faces of the i-th element in a FVM mesh
 n_f : face unit normal flux
 \vec{n}_f : face unit normal vector
 p : pressure, primary phase
 p_0 : total pressure
 p_{rgh} : modified pressure
 P : subindex for the present cell
 P^\pm : summation of inflows/outflows
 q : secondary phase
 Q : generic source

- Q^\pm : flux difference to extrema
- Re**: Reynolds number
- r : ratio of consecutive gradients/right
- s : shock velocity
- S : generic scalar source
- u : generic scalar
- u_L : Riemann problem left state
- u_R : Riemann problem right state
- \vec{S} : generic vector source
- u_s : stagnation point value
- u^\pm : stagnation point value
- u^* : values of u at each side of a cell face
- v_{rc} : constant for relative velocity law
- v_0 : constant for relative velocity law
- v_m^+ : velocity of center-of-mass at top layer
- \vec{v} : generic velocity
- \vec{v}^0 : generic velocity at the previous time-step
- \vec{v}^0 : generic velocity at previous time-step
- $\vec{v}_{dr,k}$: drift velocity of phase k respect to the velocity of center-of-mass
- \vec{v}_m : velocity of center-of-mass
- \vec{v}_{pq} : relative velocity between phases p and q
- V : cell volume
- V_p : volume of the particulized phase
- V_P : volume of present cell
- \vec{S}_f : face area vector
- w : average values in staggered mesh
- x : spatial coordinate
- \vec{x} : generic position vector
- y : spatial coordinate
- z : spatial coordinate
- α : generic scalar magnitude, void fraction
- α_p^0 : initial dispersed phase distribution
- $\vec{\beta}$: generic vector magnitude

- δ : free surface disturbance
- Δt : time interval
- Δx : space interval
- $\vec{\Delta}$: non-orthogonal decomposition vector parallel to centers' difference vector
- ϵ : maximum deviation from pure phases
- η : face tangent coordinate
- γ : convection schemes' blending factor
- γ_0 : threshold for small gradients
- Γ : fixed-in-time boundary
- Γ_i : boundary of the i-th element in a FVM mesh
- Γ_k : source term for phase k
- τ : time in a interval
- $\bar{\bar{\tau}}_D$: drift stress tensor
- τ_p : particle relaxation time
- $\bar{\bar{\tau}}$: viscous stress tensor
- θ : model indicator function, angle between face normal and cell-center to face-center vector
- ϕ : generic scalar magnitude
- $\vec{\phi}$: vector of unknowns for ϕ scalar magnitude
- $\tilde{\phi}_f$: face NVD value
- $\tilde{\phi}_C$: cell-center NVD value
- κ : mean curvature
- λ : limited weighting factor for High Resolution Schemes/FCT weighting factor, $\Delta t/\Delta x$
- ν : generic diffusivity/iteration counter
- Ω : fixed-in-time domain
- Ω_i : volume of the i-th element in a FVM mesh
- ξ : spatial coordinate within an interval, cell center to face center vector coordinate
- ϕ : generic scalar magnitude transported using a low order scheme
- ρ : density/spectral radius
- ρ_m mixture density
- ρ_m^0 initial mixture density

ψ : Swebys function

σ : surface tension

List of Figures

1.1	Relational scheme of different models used in multiphase flow	2
1.2	Representation of short and long geometrical scales in a bubbly flow	4
2.1	Domain discretization	13
2.2	Linear variation of ϕ between points P and N	19
2.3	Scheme of a non-orthogonal mesh showing the directions of \vec{d}_{PN} and \vec{S}_f vectors.	20
2.4	Face area vector, \vec{S}_f , decomposition in over-relaxed approach for non-orthogonality.	21
2.5	Non-orthogonality treatment in a boundary cell.	24
2.6	Cell array in a 1D dimensional mesh showing the nomenclature of neighbor cells to face f	30
2.7	One dimensional geometry and magnitudes for Zalesak's limiter	35
2.8	Location of variables p , \vec{v} and F in a collocated mesh to avoid pressure checkerboarding.	41
3.1	Example of the velocity of center of mass, relative velocity and drift velocities for a two phase system	49
3.2	Example of velocity of center of volume, relative velocity, drift velocities, velocity of center of mass, relation between velocity of center of volume and center of mass and other auxiliary quantities, for a bi-phasic system	53
3.3	Simple sedimentation experiment	58
3.4	Shape of fluxes for α_p equation	64

3.5	Riemann problem solutions for a convex flux	66
3.6	Riemann problem solutions for a convex flux for given states . .	68
3.7	Riemann problem solutions for a non convex flux	70
3.8	Riemann problem solutions for a non convex flux with initial condition $\alpha_p = 0.3$	70
3.9	Mean velocity, v_m , profiles corresponding α_p distributions in previous figure	71
3.10	Pressure, p , profiles corresponding α_p and v_m distributions in previous figures	71
4.1	Central differencing scheme using given centered flux at faces .	75
4.2	Scheme of face values reconstruction in a multidimensional framework	84
5.1	Evolution of the Rayleigh-Taylor instability.	97
5.2	Evolution of the Dam Break test.	99
6.1	Solutions of α_p with $\alpha_p = 0.6 - 0.7$ initialization and $\alpha_p = 0.7 - 0.5$	108
6.2	Solutions of α_p for $\alpha_p = 0.3$ initialization. Corresponding pressure and center-of-mass velocity	109
6.3	Riemann problem solutions for a convex flux	111
6.4	Riemann problem solutions for a convex flux with initial condition $\alpha_p = 0.5$	111
6.5	Geometry and solutions for a cylindrical bubble reactor	113
6.6	Velocity of center-of-mass and gas hold-up solutions as a function of the radius for the bubble column reactor ($y = 0.475$)	116
7.1	Results for the Rayleigh-Taylor in fine and coarse meshes	120
7.2	Results for the Dam Break test in fine and coarse meshes	121
7.3	Interface reconstruction using the Černe criterion	123
7.4	Face gradient criterion.	125
7.5	Geometry for the a4 case of bubble plume	129
7.6	Mean shape of the fountain	129
7.7	Solution for the bubble plume with VOF for three meshes	132

7.8	Solution for the bubble plume with the extended model for three meshes	132
7.9	Mean surfaces for the bubble plume with VOF and Extended Models	134
7.10	One dimensional reactor solutions	138
7.11	Geometry and solutions for a cylindrical bubble reactor	139
7.12	Velocity of center-of-mass and gas hold-up solutions as a function of the radius for the bubble column reactor with free surface at top ($y = 0.475$)	141
7.13	VOF solution for the Dam Break test at $t = 4$ for different meshes	144
7.14	Extended model solution for the Dam Break test at $t = 4$ for different meshes	145
7.15	Evolution of the trapped phase volume, V_p , along the time for different meshes and models	145
7.16	Extended model and VOF solutions for the Rayleigh-Taylor problem	148
7.17	Quadratic mean error for VOF and extended model solutions respect to VOF reference for the Rayleigh-Taylor problem	149
7.18	Accumulation of the more dense phase at the bottom of the domain for the Rayleigh-Taylor problem	150
A.1	Geometry and patches in scalar transport test (numbers identify cells, figure credits to Juan M. Giménez).	167
A.2	Geometry and patches in Laplacian test (numbers identifies cells, figure credits to Juan M. Giménez).	171
A.3	Geometry in <code>interFoam</code> test.	173
A.4	α field representation in Paraview [®] using <code>pexportfoamformat</code> (with <code>VTK</code> option).	177
B.1	Relación entre los diferentes tipos de modelos utilizados en flujo multifásico.	180

B.2 Representación de interfas es de pequeña y gran escala en un flujo con burbujas. a) Interfas es de gran escala, b) interfas es de pequeña escala, c) presencia de interfas es de ambas escalas en simultáneo.	182
---	-----

Chapter 1

Introduction

1.1 Background and motivation

The study of multi-fluid and multi-phase systems has a great interest in academia and industry. The correct representation and solution of this kind of systems is a key knowledge in the car engines, atomic energy, petrochemical, naval and hydraulics, chemistry, and other industries.

Several models have been devised to simulate these phenomena; within the most often used may be: the Direct Numerical Simulation (DNS), the Volume of Fluid Method (VOF) (Hirt and Nichols, 1981), the Multi-fluid Method (Drew, 1983; Ishii and Hibiki, 2010) and the Mixture Model (Manninen et al., 1996). In DNS applied to multi-phase flow the model is able to represent all the geometrical and turbulent scales for each phase. To do so, the complete set of momentum and mass conservation equations is solved for each phase, setting the corresponding interfacial and boundary conditions. It is clear that this technique is only applicable to simple real cases or laboratory tests due to the great computational resources that are required nowadays. This approach is then more suitable for cases that belong to the physics of fluids.

In the Volume of Fluid Method the geometrical analysis is similar to DNS, being the mesh size the parameter which determines the details in

the representation of the different scales of the problem. The turbulence is generally solved using either temporal or spatial filtering. The temporal filtering is implemented by the Reynolds Averaged Navier Stokes Methods (RANS), meanwhile spatial filtering is done using Large Scale Simulation (LES).

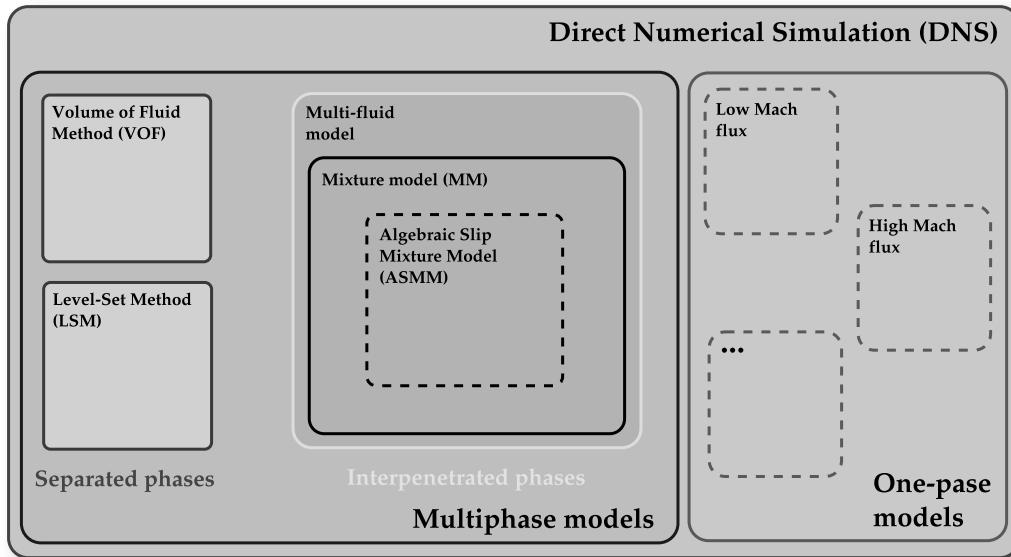


Figure 1.1: Relational scheme of different models used in multiphase flow.

The Multi-fluid model represents the next level of simplification. In this model the contact surface between phases is not explicitly tracked, considering all phases as interpenetrating continua. This approach is generally used when the geometrical structures of the flux cannot be captured by the available mesh and/or this isn't an important part of the solution process. To this end, the mass and momentum conservation equations are solved for each solid, liquid and gas phase. The representation of phases interaction is included by means of interchange terms in the mass and momentum conservation equations.

Finally, the Mixture Model has an additional simplification, in which all the interpenetrating phases are considered as a mixture, solving only one momentum conservation equation, a mass conservation equation for the

mixture and mass phase fraction equation for all but one phase. The physical properties used in the mixture equations are given by a combination of each phase's properties, using the densities and/or volume fractions in the averaging process. It is worthy to note that, even though the Mixture Model represents a simplification respect to the Multi-fluid model, the applicability of each model and the results' quality strongly relies in the nature of the problem, giving similar results in many problems. In addition the Multi-fluid model has a great weakness due to its ill-posedness (Zanotti et al., 2007) and the lack of closure laws for the momentum transfer terms between phases (Manninen et al., 1996).

Therefore, the VOF model is used in problems where surface capturing is crucial, with an important incidence of surface tension and adhesion phenomena (drop formation, capillarity, jet break-up) or where the free surface position prediction is essential (nozzles, free-surface problems in hydraulics, naval industry, reservoirs, liquids separation). In all of these cases the interfaces are considered of long scale, taking as a reference some measure of the mesh size (see Figure 1.2.a).

In the case of Multi-fluid and Mixture models the interest is found in the capacity of predicting the behaviour of flows with small-scale interfaces –“dispersed interfaces”–[see Figure 1.2.b)] when is not possible or desirable a complete modeling. This kind of interfaces is often found in sedimentation tanks, cyclone separators, annular flow in refineries and fine bubbles flow in heat exchangers.

As was presented, the DNS represents the only model which is capable to afford general multi-phase/fluid problems (Scardovelli and Zaleski, 1999; Tryggvason et al., 2006), nevertheless, the actual computational resources limitations turn impossible its direct application. On the other hand, due to their lack of generality, the rest of the models work only in particular cases accordingly to each model's hypothesis.

This situation leaves an open discussion respect to the development of

new models capable to manage several interface scales and/or transitions between them (see Figure 1.2.c). Thus, a new group of applications could be included such as the *annular mist flow* or *droplet annular flow*, the transition from *churn flow* to *bubbly flow* (Ishii and Hibiki, 2010) citing cases from the nuclear and chemical industry or the interaction of bubble plume with a free surface as is frequent in oceanography (Friedl and Fan neløp, 2000; Cloete et al., 2009) or siderurgy (Zanotti et al., 2007; Zanotti, 2007).

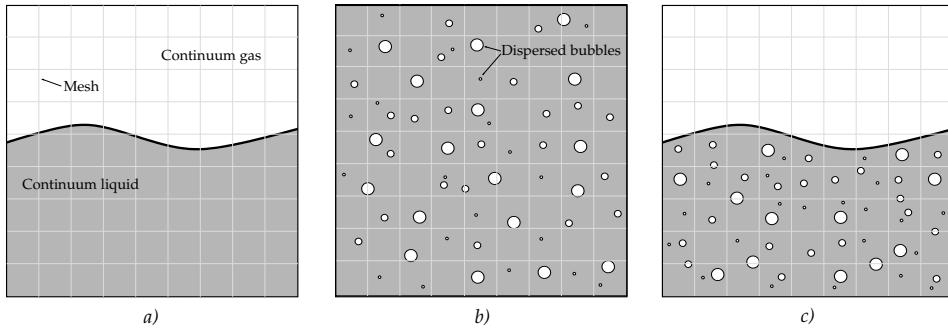


Figure 1.2: Representation of short and long geometrical scales in a bubbly flow. a) Long scale interfaces, b) short scale interfaces, c) presence of both scales simultaneously.

Respect to these cases the state of the art gives few answers. One of the first works in this field corresponds to Cerne *et. al.* (Černe et al., 2001). In this work, a coupling between the VOF and Multi-fluid models is set to the end of solving the Rayleigh-Taylor instability. Both models are treated separately, using a switching criterion for the detection of long and short interface scales. Later Strubelj and Tiselj (Štrubelj and Tiselj, 2011) follow the same work line but using the Level Set method as the surface capturing technique but giving a more unified framework without model switching. Another authors like Yan & Che (Yan and Che, 2010) and Masuda and Nagaoka (Masuda and Nagaoka, 2006) write the Multi-fluid and VOF methods equations in coupled way, closing the model by means of momentum and mass transfer terms. This methodology is used to solve flows with short and long scale bubbles and jet break-ups.

An interesting contribution in the field of coupled models is that proposed by Bohorquez (Bohorquez R. de M., 2008). In this model the Mixture Model is coupled with VOF in order to solve free-surface flows with dispersed sediments into the liquid phase. There were then long scale interfaces (water-air) and short scale interfaces (sediments-liquid). The system is solved using a general mixture formulation where the key point is the definition of the relative velocity between phases.

1.2 Objectives

As is presented in its title, the main objective of this thesis is to explore and develop new techniques for the treatment of long and short scale interfaces simultaneously. The short scale interface is solved by the Algebraic Slip Mixture Model (ASMM) meanwhile the long scale interface is managed by the VOF method. In order to devise a method for the simultaneous treatment of both interface lengths the cited methods are then written in an unified framework such as is proposed by (Bohorquez R. de M., 2008) for gas-solid-liquid systems.

The construction of such a model requires the presentation of all its building blocks either taken from the literature or specially devised for this thesis, which gives the secondary objectives. Therefore the Finite Volume Method is presented giving a summary and introducing the notation; in addition two interesting topics for the OpenFOAM®'s community are then discussed, the High Resolution Schemes framework and the implementation of the Flux Corrected Transport Method.

In addition, the ASMM is recalled and the details and main issues for solving it are addressed. This implies the necessity of the correct integration method for the mass conservation equation of the secondary phase, which is achieved by Riemann-Free solvers and by the FCT method. The ASMM solver

need to manage the short scale interfaces is based on these techniques. On the other hand the VOF method is also recalled and written as a derivation of the ASMM in order to treat the long scale interface. The derivation of the VOF method and the development of the ASMM solver fulfills another secondary objective.

Finally, based on the obtained ASMM solver and working in an unified framework for ASMM and VOF, problems with both interface scales are solved. To this end a switching criterion is proposed and compared to which are given in the literature. This solver is applied in several cases involving gas-liquid flow problems.

1.3 Thesis outline

For a better reading of the thesis it follows a thesis outline. The first chapter represents this introduction.

In Chapter 2 a brief introduction to the Cell-centered Finite Volume method is presented giving a basis to the notation and particular details of the method. The discretization of a general scalar conservation law in integral form is addressed leading to a system of algebraic equations. This requires the explanation of discrete differential operators, boundary condition setting and errors. In addition, two techniques often used in Finite Volume discretization are explained, namely the High Resolution Schemes (HRS) and the Flux Corrected Transport (FCT). At the end of the chapter, the discretization of Navier-Stokes equations is presented, the solution of which is managed by the Pressure Implicit Split of Operators (PISO) method. This chapter, the first appendix and the related references give a solid basis for the use of OpenFOAM® the tool for Finite Volume Method discretization selected for solver implementation.

Chapter 3 presents the Algebraic Slip Mixture Model as a derivation of the Multi-fluid method by means of a mass-weighted averaging. The importance of the dispersed phase transport equation is discussed and the

equation is presented in a general framework. An analysis is carried out looking for the phenomenology linked to different definitions of the relative velocity. The one-dimensional Mixture Model equations are solved obtaining an original semi-analytical solution. In order to obtain this solution it is necessary to recall the theory of hyperbolic systems with non-convex flux which the model pertains to. The model presents additional particularities such as the incompressibility which leads to lack of evolutive equation for the pressure. This issue is investigated and a solution method is presented later.

Chapter 4 is devoted to the methods for the solution of the dispersed phase conservation equation in the Mixture-Model, then a novel centered method for hyperbolic systems is presented. It is a Riemann-free solver which deals with a particular flux function where part of the flux is previously calculated at faces. The derivation of the scheme is then presented, starting from the same geometrical concepts of the Kurganov & Tadmor (Kurganov and Tadmor, 2000) scheme in which the present scheme is inspired. The extension for the use of slope limiters in face values reconstruction is also addressed presenting it in a fully multidimensional framework. In addition the use of the Flux Corrected Transport method based in global extrema is presented.

In Chapter 5 the Volume of Fluid method is revisited looking for the similarities with the Mixture Model giving and special emphasis to the implementation in OpenFOAM®. It relies on the so-called MULES (Multidimensional Universal Limiter with Explicit Solution) which is a variation of the FCT technique.

Chapter 6 presents a novel Mixture Model solver using eigenvalues information; it is based on a solver for the dispersed-phase equation using the technique presented in Chapter 5 and a pressure-velocity coupling by means of the PISO method. The results for this solver are compared to the semi-analytical solutions for the sedimentation case given in Chapter 3 and to the solution of a bubble column reactor.

In chapter 7 a Mixture Model/VOF coupled solver is described and the

corresponding results are presented. The motivation for this solver is based on the results of the previous chapters and the necessity of capturing the dynamics of the under-mesh level and the free-surface. This goal is achieved using an indicator function which switches the models along the geometry. Several cases including which are presented in the previous chapters are solved and compared showing the benefits of the interface-capturing technique given by VOF.

Finally, the last chapter presents the conclusions, the future work and the articles which share it with the scientific community.

1.4 Contributions of the work

A general contribution of this thesis is the development of novel techniques in order to solve problems with mixed geometrical scales for the free-surface. The techniques were implemented in a native parallelized framework as OpenFOAM® with the aim of solving problems from academic to industrial scale using High Performance Computing (HPC).

In addition, the following original contributions were made:

- The conceptualization of the ASMM as a hyperbolic system with restrictions where the definition of the flux function for the void fraction conservation equation plays a principal role.
- An 1D semi-analytical solution for sedimentation and the study of different test cases.
- The development of a new Riemann-Free solver based in the Kurganov & Tadmor solver including a given centered flux at faces.
- The development of a new solver for the ASMM based in eigenvalues information which is capable to manage problems for all the void fraction range and not only for dilute mixtures.

- An ASMM-VOF parallelized coupled solver to manage problems with short and long scale interfaces. This solver is based in a novel indicator function which can be used in non-structured meshes.
- The application of ASMM-VOF coupled solver to academic and industrial cases.
- A toolbox for OpenFOAM® debugging (gdbOF) to help in solvers implementation.

Chapter 2

Cell-centered Finite Volume Method

2.1 Introduction

The solution of the partial differential equations from the models in the continuum requires the process of discretization in order to be solved by a finite capacity machine, in terms of CPU power and memory. This objective can be achieved by several methods, being the Finite Difference Method (FDM), the Finite Element Method (FEM) and the Finite Volume Method (FVM) among the most popular. Beyond some particular advantages and disadvantages of each method the decision of which method will be used is generally a cultural matter around the workgroup legacy, mathematical background, available code capabilities, etc. Therefore, the Cell-centered Finite Volume Method is selected and the practical implementation is given by the OpenFOAM® code. The selection of this particular implementation of the FVM is based on its capabilities for easy solver writing, transparent parallelization, a huge database of solver examples and the free access to the code via the GNU-GPL license. This chapter is devoted to the general details of the discretization procedure of scalar and vector conservation equations in arbitrary meshes. In addition a brief description of the High Resolution Schemes and Flux Corrected Transport frameworks and the pressure-velocity coupling is given. The reader

is referred to the work of Jasak (Jasak, 1996) which is the fundamental source about OpenFOAM®'s FVM implementation and to the books of Ferziger and Peric (Ferziger and Peric, 2002) and Versteeg and Malalasekera (Versteeg and Malalasekera, 2007) for further and in-deep reading about the FVM.

2.2 Discretization of a general advection-diffusion-reaction equation

Let be $\phi(x, y, z, t)$ a generic scalar magnitude, Ω a fixed-in-time domain in \mathbb{R}^3 and Γ its boundary as is shown in Figure 2.1.a); then, it is possible to write a conservation equation for this quantity in integral form as in Eqn. (2.1)

$$\int_{\Omega} \frac{\partial \phi}{\partial t} d\Omega + \int_{\Gamma} \vec{v} \phi \cdot d\vec{\Gamma} + \int_{\Omega} c \phi d\Omega = \int_{\Gamma} \nu \vec{\nabla} \phi \cdot d\vec{\Gamma} + \int_{\Omega} Q d\Omega \quad (2.1)$$

where \vec{v} is an advective field, c a generic reactive constant, ν the diffusivity and Q a source term. The first term represents the variation of ϕ along the time, the second one the flux through Γ due to the velocity (or advective) field, the third one the production or destruction of ϕ due to reaction, the fourth one the flux due to the diffusion and finally the last one is the term associated to the source¹. These terms can be reordered in volumetric and surface terms as in Eqn. (2.2)

$$\int_{\Omega} \left(\frac{\partial \phi}{\partial t} + c \phi - Q \right) d\Omega = \int_{\Gamma} \left(\nu \vec{\nabla} \phi - \vec{v} \phi \right) \cdot d\vec{\Gamma} \quad (2.2)$$

From this equation it follows that the imbalance in ϕ in the interior of Γ is simply which crosses the boundary Γ .

Dividing the domain Ω in m finite volumes of any shape with the only condition of non overlapping such as in Figure 2.1.b) and re-writing Eqn. (2.1) applied to every volume in Ω and summing up, it reads as in Eqn. (2.3)

¹All magnitudes are expressed in the International System of Units in all extension of this thesis as is usual in OpenFOAM®

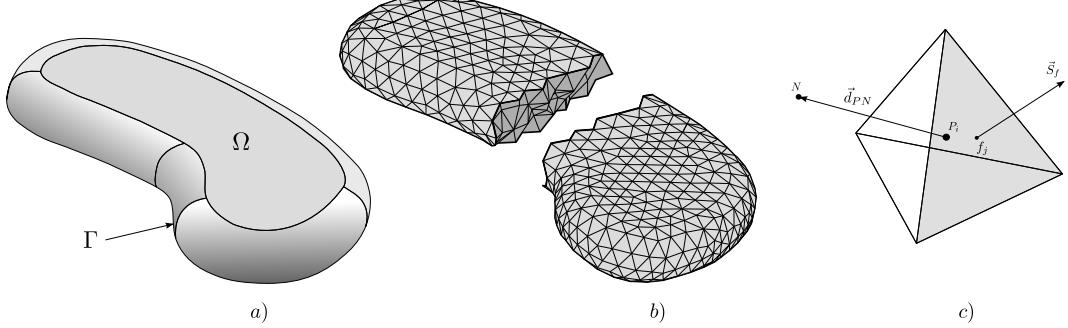


Figure 2.1: Domain discretization. a) Original domain Ω and its boundary Γ ; b) finite volume discretization of Ω , darker zones show volume mesh; c) isolated finite volume².

$$\sum_{i=1}^m \int_{\Omega_i} \left[\left(\frac{\partial \phi}{\partial t} + c \phi - Q \right) d\Omega - \int_{\Gamma_i} (\nu \vec{\nabla} \phi - \vec{v} \phi) \right] \cdot d\vec{\Gamma} = 0 \quad (2.3)$$

In the case of the volumetric integrals their meaning is clear: each one represents the integral of the corresponding magnitude within the i -th finite volume. In the case of the surface integrals it is necessary to take into account that the faces which do not lie on the boundary are included in the integral for both sides of the face. Therefore, the fluxes in these faces cancel each other and then Eqn. (2.3) reduces to Eqn. (2.2). It is not only an interesting property of the scheme but a necessary condition for the discretization to work. Due to this property the discretization manifests the conservative nature of the original law.

It is possible to have a deeper analysis of the scheme considering the finite volumes as polyhedra (faceted volumes) with a finite number of faces n_i . Then Eqn. (2.3) becomes Eqn. (2.4)

²3D modeling and meshing of base sub-figure b) by Santiago F. Corzo using SolidWorks® and ANSA®. Vectorized version and coloring by Santiago Márquez Damián.

$$\sum_{i=1}^m \left\{ \int_{\Omega_i} \left[\frac{\partial \phi}{\partial t} + c \phi - Q \right] d\Omega - \sum_{j=1}^{n_i} \int_{\Gamma_j} \left[\nu \vec{\nabla} \phi - \vec{v} \phi \right] \cdot d\vec{\Gamma} \right\}_i = 0 \quad (2.4)$$

Considering the ϕ values varying linearly by cell (finite volume) and by face, see Eqn. (2.9), it is possible to approximate the integrals as in Eqns. (2.5-2.6), see the proof in Eqns. (2.14)-(2.15)

$$\int_{\Omega_i} \alpha d\Omega_i = \bar{\alpha} V_i \quad (2.5)$$

$$\int_{\Gamma_i} \vec{\beta} \cdot d\vec{\Gamma}_i = \bar{\beta} \cdot \vec{S}_{f_i} \quad (2.6)$$

being α and $\vec{\beta}$ two generic scalar and vector magnitudes respectively. Then, dropping the overbar, Eqn. (2.4) becomes Eqn. (2.7)

$$\sum_{i=1}^m \left[\left(\frac{\partial \phi}{\partial t} + c \phi - Q \right) V_i - \sum_{j=1}^{n_i} \left(\vec{\nabla} \phi - \vec{v} \phi \right) \cdot \vec{S}_{f_j} \right] = 0 \quad (2.7)$$

being this the spatially discretized expression of the conservation equation for an scalar magnitude. Now it's necessary to define the discrete version of the space and time derivatives and the values of ϕ at faces calculated from the cell-center values, which are the real unknowns. The discretization leads to a linear system of m equations with n unknowns as in Eqn. (2.8).

$$\bar{\vec{A}} \cdot \vec{\phi} = \vec{b} \quad (2.8)$$

The discretization of the general advection-diffusion-reaction equation usually leads to non-symmetric linear systems, except for the cases with no advection or with a centered scheme for the advective term. On the contrary, the presence of symmetric systems is usual in the case of the discretization of the pressure equation, thermal problems in solids and solids deformation.

The solution of this system is achieved by the use of iterative Linear Equations Solvers. OpenFOAM® has a set of preconditioned iterative

linear solvers for symmetric and non-symmetric matrices. This set is composed by the Preconditioned Conjugate Gradient (PCG) and the Diagonal Incomplete-Cholesky (DIC) and Diagonal Incomplete-Cholesky with Gauss-Seidel (DICGaussSeidel) smoothers for the symmetric cases. In addition, the Preconditioned Bi-Conjugate Gradient (PBiCG), the Gauss-Seidel smoother and the Geometric-Algebraic Multi-Grid (GAMG) can be used in the case of non-symmetric matrices problems (Saad, 2003). The performance in the solution of the linear systems is usually improved by the use of preconditioners; the list of available methods for preconditioning includes: DIC (symmetric), Faster Diagonal Incomplete-Cholesky (DIC with caching, FDIC), Diagonal Incomplete-LU (asymmetric, DILU), diagonal and GAMG (OpenCFD, 2012).

In the case of the problems presented in this work the non-symmetric cases are solved using a PBiCG solver with DILU preconditioning. For the symmetric problems the PCG method with DIC preconditioner is used for problems up to 100,000 unknowns, beyond this limit the GAMG method results to have better performance. This trend is true in general but the limit of unknowns depends on the type of problem, hardware capabilities, etc.

2.3 Discrete version of differential operators

As a starting point for the definition of the differential operators in discrete form it is necessary describe the variation of the solution ϕ along the domain. Since the solution is given in certain points of the space it is necessary to find the values for other arbitrary points within the domain but not pertaining to the mesh. Having a linear variation (which leads to a second-order accurate method) in space and time the value of an arbitrary point can be found as is shown in Eqns. (2.9-2.10)

$$\phi(\vec{x}) = \phi_P + (\vec{x} - \vec{x}_P) \cdot (\vec{\nabla}\phi)_P \quad (2.9)$$

$$\phi(t + \Delta t) = \phi^t + \Delta t \left(\frac{\partial \phi}{\partial t} \right)^t \quad (2.10)$$

where \vec{x} represents the position of a generic point within the domain Ω , \vec{x}_P is the position of the cell's centroid [see Figure 2.1.c)], $\phi_P = \phi(\vec{x}_P)$ and $\phi^t = \phi(t)$. On the other hand, a three dimensional general form of the Taylor expansion (Gray and Gubbins, 1984; Jasak, 1996) can be recalled as in Eqn. (2.11)

$$\begin{aligned} \phi(\vec{x}) &= \phi_P + (\vec{x} - \vec{x}_P) \cdot (\vec{\nabla}\phi)_P + \frac{1}{2}(\vec{x} - \vec{x}_P)^2 : (\vec{\nabla}\vec{\nabla}\phi)_P + \\ &\quad \frac{1}{3!}(\vec{x} - \vec{x}_P)^3 :: (\vec{\nabla}\vec{\nabla}\vec{\nabla}\phi)_P + \\ &\quad \dots + \frac{1}{n!}(\vec{x} - \vec{x}_P)^n \underbrace{\dots}_{n-1} \underbrace{(\vec{\nabla}\vec{\nabla}\dots\vec{\nabla}\phi)_P}_{n} + \dots \end{aligned} \quad (2.11)$$

where $(\vec{x} - \vec{x}_P)^n$ relies on a n -th tensor product and the operator $\underbrace{\dots}_{n}$ is the inner product of two tensor of rank n -th. Comparing Eqn. (2.9) and Eqn. (2.11) it can be stated that the first term of truncation is proportional to $|\vec{x} - \vec{x}_P|^2$. Then, the linear variation proposed for ϕ leads to a second-order truncation term. The same analysis can be done for the temporal variation.

Another topic is the selection of the location of the point P . From the definition of the centroid [Eqn. (2.12)]

$$V_P \vec{x}_P = \int_{\Omega_i} \vec{x} d\Omega \quad (2.12)$$

it is possible to arrive to a relationship given in Eqn. (2.13)

$$\int_{\Omega_i} \vec{x} d\Omega - \vec{x}_P \int_{\Omega_i} d\Omega = \int_{\Omega_i} (\vec{x} - \vec{x}_P) d\Omega = 0 \quad (2.13)$$

where V_P is the volume of the domain Ω_i . The same expression can be derived for surface integral where the centroid of the cell is replaced by the face's centroid.

Therefore, an integral of $\phi(\vec{x})$ on Ω can be simplified to which is shown in Eqn. (2.14)

$$\begin{aligned}
\int_{\Omega_i} \phi(\vec{x}) d\Omega &\cong \int_{\Omega_i} \left[\phi_P + (\vec{x} - \vec{x}_P) \cdot (\vec{\nabla} \phi)_P \right] d\Omega \\
&= \phi_P \int_{\Omega_i} d\Omega + \underbrace{\int_{\Omega_i} [(\vec{x} - \vec{x}_P) d\Omega] \cdot (\vec{\nabla} \phi)_P}_{=0} \\
&= \phi_P V_P
\end{aligned} \tag{2.14}$$

where the property presented in Eqn. (2.13) was used. Respect to the surface integral, it can be simplified in a similar way as in Eqn. (2.15)

$$\begin{aligned}
\int_{\Gamma_j} \vec{a} d\vec{\Gamma} &= \vec{a}_f \cdot \int_{\Gamma_j} d\vec{\Gamma} + \underbrace{\left[\int_{\Gamma_j} (\vec{x} - \vec{x}_f) d\vec{\Gamma} \right]}_{=0} : (\vec{\nabla} \vec{a})_f \\
&= \vec{a}_f \cdot \vec{S}_f
\end{aligned} \tag{2.15}$$

being \vec{a} a generic vector value at the j -th face, \vec{a}_f the value of \vec{a} at the face centroid and \vec{S}_f the face area vector for the same face. Now, it is possible to obtain the expression for the some differential operators. In the case of the divergence operator and using the Gauss' theorem and Eqns. (2.14-2.15), it reads as in Eqn. (2.16)

$$\begin{aligned}
\int_{\Omega_i} (\vec{\nabla} \cdot \vec{a}) d\Omega &= \int_{\Gamma_i} \vec{a} \cdot d\vec{\Gamma} \\
(\vec{\nabla} \cdot \vec{a}) V_P &= \sum_j \int_{\Gamma_j} \vec{a} \cdot d\vec{\Gamma} = \sum_j (\vec{a}_f \cdot \vec{S}_f)_j \\
\vec{\nabla} \cdot \vec{a} &= \frac{1}{V_P} \sum_j (\vec{a}_f \cdot \vec{S}_f)_j
\end{aligned} \tag{2.16}$$

In the case of the gradient operator it is presented in Eqn. (2.17)

$$\begin{aligned} \int_{\Omega_i} \vec{\nabla} \phi \, d\Omega &= \int_{\Gamma_i} \phi \, d\vec{\Gamma} \\ \vec{\nabla} \phi &= \frac{1}{V_P} \sum_j (\phi_f \cdot \vec{S}_f)_j \end{aligned} \tag{2.17}$$

2.3.1 Convective operator

Recalling the general advection-diffusion-reaction equation in Eqn. (2.1) and taking the convective operator, it is possible to apply the rules given in Eqn. (2.16) as follows in Eqn. (2.18)

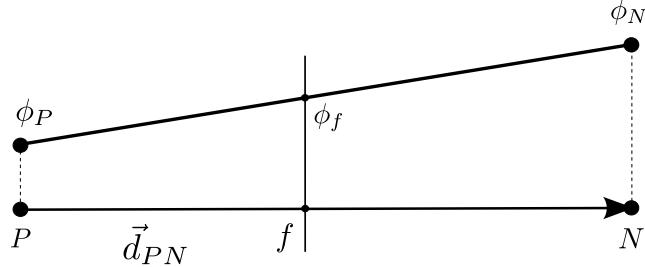
$$\int_{\Gamma} (\vec{v} \phi) \cdot d\vec{\Gamma} = \sum_f (\phi \vec{v})_f \cdot \vec{S}_f = \sum_f \phi_f (\vec{v}_f \cdot \vec{S}_f) = \sum_f F_f \phi_f = \sum_f F_f \tag{2.18}$$

where the notation of the summation was simplified to an index f representing each of the cell faces. In addition, the face flux through a face can be defined as in Eqn. (2.19)

$$F_f = (\vec{v}_f \cdot \vec{S}_f) \tag{2.19}$$

Finally, two magnitudes often used can be defined, the total face flux, $F_f = F_f \phi_f$, and $\vec{F}_f = \vec{v}_f \phi_f$ which is just named as the flux. This flux corresponds to the argument of the spatial derivative of transport problems. In this context, the calculation of ϕ_f leads to the appropriate selection of a *convection scheme*. The basic convection schemes are the Central Differencing (CD) and the Upwind Differencing (UD). Recalling Figure 2.1.c) and focusing on the variation of ϕ between the points P and N it is possible to find the value in the face between these points as is shown in Figure 2.2.

Assuming a linear variation of ϕ between points P and N the face value can be calculated as in Eqn. (2.20)

Figure 2.2: Linear variation of ϕ between points P and N .

$$\phi_f = f_x \phi_P + (1 - f_x) \phi_N \quad (2.20)$$

where $f_x = \frac{\vec{fN}}{\vec{PN}}$ is a weighting factor and \vec{d}_{PN} is the cell centers' difference vector. The presented calculation of f_x corresponds to the CD scheme. On the other hand, the UD scheme can be written in the framework given by Eqn. (2.20) defining $f_x = 1$ if $F > 0$ and 0 otherwise. This implies that ϕ_f takes the value from the *upwind* cell respect to the advective velocity direction. The CD scheme has the property to be second order accurate but unbounded, meanwhile UD is bounded but only first order accurate (Hirsch, 2007). Blendings between both schemes can be used in order to achieve both accuracy and boundedness, this topic is addressed in Section 2.5.

2.3.2 Diffusive operator

In the case of the diffusive operator the discretization is carried out as is shown in Eqn. (2.21)

$$\int_{\Omega} \vec{\nabla} \cdot (\nu \vec{\nabla} \phi) d\Omega = \sum_f (\nu \vec{\nabla} \phi)_f \cdot \vec{S}_f = \sum_f (\nu)_f (\vec{\nabla} \phi)_f \cdot \vec{S}_f \quad (2.21)$$

If the vectors \vec{d}_{PN} and \vec{S}_f are parallel (See Figure 2.3) there is a simple way to calculate the factor $(\vec{\nabla} \phi)_f \cdot \vec{S}_f$ needed in Eqn. (2.21) as shown in Eqn. (2.22)

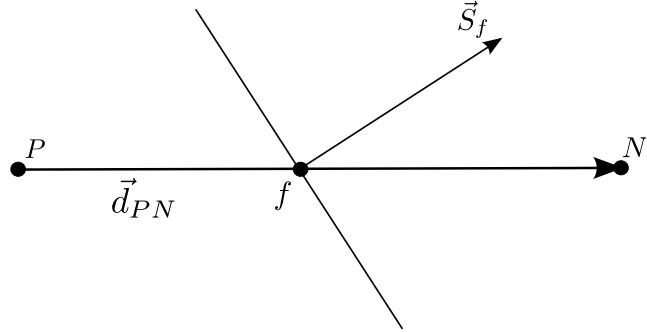


Figure 2.3: Scheme of a non-orthogonal mesh showing the directions of \vec{d}_{PN} and \vec{S}_f vectors.

$$(\vec{\nabla}\phi)_f \cdot \vec{S}_f = |\vec{S}_f| \frac{\phi_N - \phi_P}{|\vec{d}_{PN}|} \quad (2.22)$$

Another way to obtain the value of this factor is evaluating the cell-centered gradient via Eqn. (2.17) both in cell P and cell N and then interpolate it to the face. Even when it is possible, it requires information from second neighbors, which implies a bigger computational molecule and also more entries in the diffusive matrix. The studied case corresponds to an orthogonal mesh (a mesh with all right angles between faces), this case is far from being general since most industrial meshes need some grade of non-orthogonality to fit the geometry. To this end the factor $(\vec{\nabla}\phi)_f \cdot \vec{S}_f$ is calculated having into account the decomposition of the vector \vec{S}_f as in Eqn. (2.23)

$$(\vec{\nabla}\phi)_f \cdot \vec{S}_f = \underbrace{\vec{\Delta} \cdot (\vec{\nabla}\phi)_f}_{\text{orthogonal contribution}} + \underbrace{\vec{k} \cdot (\vec{\nabla}\phi)_f}_{\text{non-orthogonal correction}} \quad (2.23)$$

where the proposed decomposition is $\vec{S}_f = \vec{\Delta} + \vec{k}$. The vector $\vec{\Delta}$ is parallel to \vec{d}_{PN} , then, this contribution is calculated using Eqn. (2.22), while the non-orthogonal correction calculation depends on the decomposition selected. There are several decompositions, the one which is used in OpenFOAM® is the *over-relaxed approach* (Jasak, 1996) [See Appendix A or (Márquez Damián et al., 2012)]. The vector decomposition is depicted in Figure (2.4), where

$\vec{\Delta}$ is defined as $\vec{\Delta} = \frac{\vec{d}_{PN}}{\vec{d}_{PN} \cdot \vec{S}_f} |\vec{S}_f|^2$. As was indicated, the orthogonal part is discretized in the standard way therefore Eqn. (2.23) is now written as in Eqn. (2.24)(Mathur and Murthy, 1997; Versteeg and Malalasekera, 2007)

$$(\vec{\nabla}\phi)_f \cdot \vec{S}_f = |\vec{\Delta}| \frac{\phi_N - \phi_P}{|\vec{d}_{PN}|} + \vec{k} \cdot (\vec{\nabla}\phi^{\nu-1})_f \quad (2.24)$$

Here $\vec{\nabla}\phi^{\nu-1}$ is calculated using Eqn. (2.17) in an iterative way being ν the iterator index. This is a kind of deferred correction as is explained in Sections 5.6 and 8.6 of (Ferziger and Peric, 2002) and implies the solution of the whole system several times for the same time-step, which is more time consuming but allows to keep a compact computational molecule.

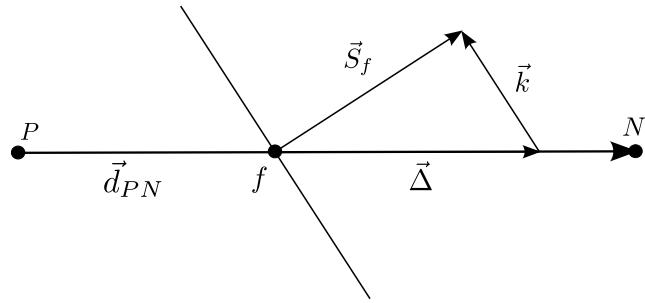


Figure 2.4: Face area vector, \vec{S}_f , decomposition in over-relaxed approach for non-orthogonality.

2.3.3 Temporal discretization

In order to analyze the discretization of the temporal term is worthy to recall the final expression of the spatially discretized general advection-diffusion-reaction equation [Eqn. (2.7)]

$$\sum_{i=1}^m \left(\frac{\partial \phi}{\partial t} + c \phi - Q \right) V_i - \left[\sum_{j=1}^n (\vec{\nabla} \phi - \vec{v} \phi) \cdot \vec{S}_{f_j} \right]_i = 0 \quad (2.25)$$

Integrating this expression in a Δt interval it gives which is shown in Eqn.

(2.26)

$$\int_t^{t+\Delta t} \left\{ \sum_{i=1}^m \left(\frac{\partial \phi}{\partial t} + c \phi - Q \right) V_i - \left[\sum_{j=1}^n (\vec{\nabla} \phi - \vec{v} \phi) \cdot \vec{S}_{f_j} \right]_i \right\} dt = 0 \quad (2.26)$$

Recalling the assumed linear variation of ϕ along the time step given by Eqn. (2.10)

$$\phi(t + \Delta t) = \phi^t + \Delta t \left(\frac{\partial \phi}{\partial t} \right)^t \quad (2.27)$$

it is possible to find the expression for the time derivatives and integral as is shown in Eqns. (2.28)-(2.29)

$$\left(\frac{\partial \phi}{\partial t} \right)_P = \frac{\phi_P^n - \phi_P^0}{\Delta t} \quad (2.28)$$

$$\int_t^{t+\Delta t} \phi(t) dt = \frac{1}{2} (\phi^0 + \phi^n) \Delta t \quad (2.29)$$

where ϕ^n represents the value of ϕ at the present time-step and ϕ^0 its value at the previous time-step. In the case of no time variation in the physical properties the discretization for cell P reads as in Eqn. (2.30)

$$\begin{aligned} & \frac{\phi_P^n - \phi_P^0}{\Delta t} V_P + \frac{1}{2} \sum_f F_f \phi_f^n - \frac{1}{2} \sum_f (\nu)_f (\vec{\nabla} \phi)_f^n \cdot \vec{S}_f \\ & + \frac{1}{2} \sum_f F_f \phi_f^0 - \frac{1}{2} \sum_f (\nu)_f (\vec{\nabla} \phi)_f^0 \cdot \vec{S}_f \\ & = Q V_P - \frac{1}{2} c \phi_P^n V_P - \frac{1}{2} c \phi_P^0 V_P \end{aligned} \quad (2.30)$$

which is an example of Crank-Nicholson discretization and is second order accurate, but can produce unboundedness. This expression can be generalized as in Eqn. (2.31)

$$\begin{aligned}
& \frac{\phi_P^n - \phi_P^0}{\Delta t} V_P + \theta \sum_f F_f \phi_f^n - \theta \sum_f (\nu)_f (\vec{\nabla} \phi)_f^n \cdot \vec{S}_f \\
& + (1 - \theta) \sum_f F_f \phi_f^0 - (1 - \theta) \sum_f (\nu)_f (\vec{\nabla} \phi)_f^0 \cdot \vec{S}_f \quad (2.31) \\
& = Q V_P - \theta c \phi_P^n V_P - (1 - \theta) c \phi_P^0 V_P
\end{aligned}$$

this expression leads to the Generalized Trapezoidal Method (Hirsch, 2007). Choosing $\theta = 0$ it gives the Forward-Euler Method which is explicit and restricted to $\mathbf{Co} \leq 1$ and is not directly available in OpenFOAM®. Taking $0 < \theta \leq 1$ leads to implicit methods, particularly for the case of $\theta = 1/2$ the Crank-Nicholson scheme presented in Eqn. (2.30) is recalled. The case of $\theta = 1$ is known as the Backward-Euler method which is first order accurate and always bounded. These methods are available in OpenFOAM® under the name of `crankNicholson`³ and `Euler`. There is also available a `backward` method which is a back-difference method using two backward time-steps. This scheme is second order accurate, however, the boundedness of the solution is not guaranteed. Here it is important to note that even when implicit methods can be used, some terms are treated always explicitly, this is the case of the non-orthogonal corrections used in the diffusive term studied in section 2.3.2. The High Resolution Methods presented later in this chapter also use this technique in order to evaluate the gradients needed for the limiting of ϕ at faces.

2.4 Boundary conditions

The solution of a problem within the domain Ω depends on the governing equations representing a particular phenomenon and on the boundary conditions given at the Γ boundary [see Figure 2.1.). These boundary conditions represent the interaction of the phenomenon inside the domain with the universe outside it. In this analysis the boundary conditions will be divided in basic and derived

³The courier font will hereinafter be used to denote OpenFOAM® keywords or commands

groups.

2.4.1 Basic boundary conditions

The basic group of boundary conditions is given by the Dirichlet and Neumann boundary conditions which fix the value or the gradient normal to the boundary, respectively. To start the analysis it is important to make clear the treatment of the non-orthogonality at the boundaries. To do so a boundary cell is considered as in Figure 2.5 (De Villiers, 2006). In the figure some new entities are presented, the boundary face centroid b , the vector joining the cell center and face center \vec{d} and the vector normal to the face and starting in P , which is called \vec{d}_n . This vector is calculated as is shown in Eqn. (2.32)

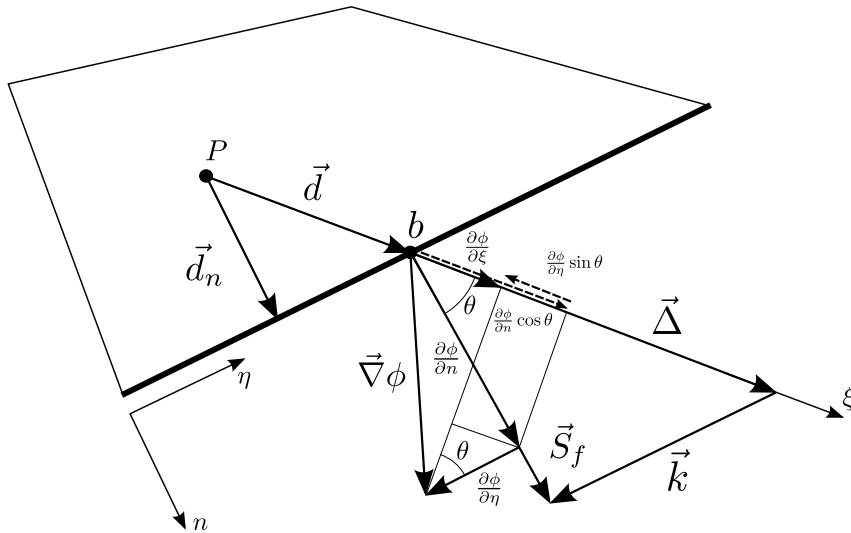


Figure 2.5: Non-orthogonality treatment in a boundary cell.

$$\vec{d}_n = \frac{(\vec{S}_f \cdot \vec{d}) \vec{S}_f}{|\vec{S}_f|^2} \quad (2.32)$$

- Fixed value boundary condition

This boundary condition fixes the value of $\phi = \phi_b$ at the face boundary b . The way that this boundary condition is imposed depends on the kind of

term which is being assembled:

- Convective term. From Eqn. (2.18) the convection term is discretized as follows

$$\int_{\Gamma} \vec{v} \phi \cdot d\vec{\Gamma} = \sum_f \phi_f (\vec{v}_f \cdot \vec{S}_f) = \sum_f F_f \phi_f$$

therefore, fixing the boundary condition is straightforward and relies in taking $\int_{\Gamma} \vec{v} \phi \cdot d\vec{\Gamma} = \sum_f F_b \phi_b$. This approach is ever true except for, perhaps, the pure UD scheme. In this case if the flux F_b is positive, i.e. it gives a transport outside the domain and the UD scheme takes the value of the face from the upwind cell-centroid and not from the given value ϕ_b . The use of the given ϕ_b can result in unbounded solutions for Peclet number beyond unity.

- Diffusive term. According to Eqn. 2.21 the diffusive term is assembled by the following expression

$$\int_{\Gamma} \vec{\nabla} \cdot (\nu \vec{\nabla} \phi) d\Omega = \sum_f (\nu)_f (\vec{\nabla} \phi)_f \cdot \vec{S}_f$$

this expression requires the calculation of the gradient at the boundary face which is achieved by means of the given values and the cell center, ϕ_P , and the face center, ϕ_b , by a first order approximation. In order to derive the value of this face gradient the following additional entities are defined in Figure 2.5: n represents the face normal direction, η is the face tangent direction and ξ the direction of \vec{d} as is proposed by Mathur and Murthy (Mathur and Murthy, 1997). Since the known distance for gradient calculation is $|\vec{d}|$ the calculation of $\frac{\partial \phi}{\partial \xi}$ is straightforward, on the other hand the needed gradient is $\vec{\nabla} \phi \cdot \vec{n} = \frac{\partial \phi}{\partial n}$, now, it is necessary to find a relationship between both expressions. From Figure 2.5 it is clear that this relationship reads as follows

$$\frac{\partial \phi}{\partial \xi} = \frac{\partial \phi}{\partial n} \cos \theta - \frac{\partial \phi}{\partial \eta} \sin \theta$$

where θ is the angle between \vec{d} and \vec{n} . Thus, isolating the face normal gradient it becomes

$$\vec{\nabla} \phi \cdot \vec{n} = \frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial \xi} \frac{1}{\cos \theta} + \frac{\partial \phi}{\partial \eta} \tan \theta$$

now, since $\frac{\partial \phi}{\partial \xi} = \frac{\phi_b - \phi_P}{|\vec{d}|}$ and $\cos \theta = \frac{\vec{d} \cdot \vec{S}_f}{|\vec{d}| |\vec{S}_f|}$ it is possible to write

$$(\vec{\nabla} \phi)_b \cdot \vec{S}_f = \frac{|\vec{d}|}{\vec{d} \cdot \vec{S}_f} \left| \vec{S}_f \right| \frac{\phi_b - \phi_P}{|\vec{d}|} + \frac{\partial \phi}{\partial \eta} \tan \theta$$

finally, due to the transversal gradient $\frac{\partial \phi}{\partial \eta}$ is null because the constant value of ϕ on the face, and doing some algebraic simplifications the desired expression for the face normal gradient is obtained

$$(\vec{\nabla} \phi)_b \cdot \vec{S}_f = |\vec{S}_f| \frac{\phi_b - \phi_P}{|\vec{d}_n|}$$

The effect of this first order approximation is supposed to be marginal and does not affect the order of approximation of the general problem.

This boundary condition is named **fixedValue** and requires the given value at the face.

- Fixed gradient boundary condition

In the case of the fixed gradient boundary condition the fixed magnitude is the gradient normal to the boundary face, it is calculate as follows

$$\left(\frac{\vec{S}_f}{|\vec{S}_f|} \cdot \vec{\nabla} \phi \right)_b = g_b$$

- Convective term. Since the value ϕ_b is not given this has to be calculated from the cell-center value ϕ_P and the given gradient g_b which is obtained by

$$\begin{aligned}\phi_b &= \phi_P + \vec{d}_n \cdot (\vec{\nabla} \phi)_b \\ &= \phi_P + |\vec{d}_n| g_b\end{aligned}$$

this value is used to calculate the expression $\int_{\Gamma} \vec{v} \phi \cdot d\vec{\Gamma} = \sum_f F_b \phi_b$

- Diffusive term. In the case the fixing of the boundary condition is direct since

$$(\vec{\nabla} \phi)_b \cdot \vec{S}_f = g_b$$

then, the diffusive operator is assembled as $\int_{\Gamma} \vec{\nabla} \cdot (\nu \vec{\nabla} \phi) d\Omega = \sum_f (\nu)_f g_b$

The fixed gradient boundary condition is named **fixedGradient** and the value of the normal gradient is required. The case of zero flux (gradient) or Homogeneous Neumann boundary condition is called **zeroGradient**.

2.4.2 Derived boundary conditions

From the basic boundary conditions it is possible to derive a series of new boundary conditions which are useful in the real practice. They are presented in the following list with the OpenFOAM® nomenclature as a future reference:

- **slip**. Equivalent to **zeroGradient** in case of a scalar magnitude; for a vector magnitude the normal component is **fixedValue** zero, and tangential components are set to **zeroGradient**. This boundary condition is often used with the velocity in order to set slip walls;
- **empty**. Used in faces perpendicular to the directions which will be not calculated. Used to set 2D and 1D problems using full 3D meshes;

- **symmetryPlane**. Sets a symmetry plane;
- **wedge**. Allows to set an axi-symmetric problem. This boundary condition has to be set in the plane normal to the tangential direction;
- **inletOutlet**. Sets either `zeroGradient` or `fixedValue` for positive or negative face flux. It is intended to be used for outflow boundaries;
- **totalPressure**. The static pressure of the boundary is set from a given total pressure p_0 as $p = p_0 - \frac{1}{2}\rho |\vec{v}|^2$;
- **buoyantPressure**. This boundary conditions sets the $\vec{\nabla}p \cdot \vec{S}_f = -\vec{\nabla}(\rho |\vec{g}|) \cdot \vec{S}_f$ and is used where the value of the pressure is not known;
- **pressureInletOutletVelocity**. This boundary condition sets a velocity inlet/outlet boundary condition in the patches for where the pressure is specified.

2.5 High Resolution Schemes Implementation

In the general scalar transport equation an important issue is the necessity of calculating face values of the transported magnitude. The improper calculation of these values can bring unboundedness or numeric diffusion problems when the advective terms are assembled and solved. In simple advection discretization schemes, such as Central Difference (CD) and Upwind Differencing (UD) a compromise between accuracy and boundedness must be accepted. While UD is completely bounded it is only first order accurate, on the other hand CD is second order accurate but generates unbounded values when it is used with Courant numbers beyond unity and with high Peclet numbers. In this context the High Resolution Schemes (HRS) allow to assemble advective terms with better accuracy than first order without losing boundedness. The derivation of the basic HRS framework starts with the basic schemes. Therefore, from CD discretization the face value ϕ_f can be calculate as in Eqn. (2.33) (Jasak, 1996).

$$\phi_f = f_{xCD} \phi_P + (1 - f_{xCD}) \phi_N \quad (2.33)$$

where $f_{xCD} = \frac{\bar{f}_N}{\bar{P}_N}$ is the weighting factor for linear interpolation, being P and N the centroids of the cells sharing the face. This expression can be rearranged in Eqn. (2.34).

$$\phi_{fCD} = f_{xCD} (\phi_P - \phi_N) + \phi_N \quad (2.34)$$

On the other hand, for UD the same equation can be used, taking into account that the weighting factor must be redefined as f_{xUD} . This factor is 1 if $F = \vec{v} \cdot \vec{S}_f > 0$ and 0 otherwise.

A linear combination of the aforementioned methods can be used giving place to the Blended Differencing (BD) methods defined by Eqn. (2.35).

$$\phi_f = (1 - \gamma) \phi_{fUD} + \gamma \phi_{fCD}, \quad (2.35)$$

where γ is a blending factor. Replacing Eqn. (2.34) in (2.35), Eqn. (2.36) is obtained, and can be rewritten in Eqn. 2.37 (Berberovic et al., 2009).

$$\phi_f = [(1 - \gamma) f_{xUD} + \gamma f_{xCD}] (\phi_P - \phi_N) + \phi_N \quad (2.36)$$

$$\phi_f = \lambda (\phi_P - \phi_N) + \phi_N \quad (2.37)$$

Equation (2.37) allows for a compact form of blending, being $\lambda = (1 - \gamma) f_{xUD} + \gamma f_{xCD}$, a limited weighting factor. The blending factor γ , $0 \leq \gamma \leq 1$, can be selected as a constant (Ferziger and Peric, 2002) or evaluated in a local basis using a limiter function as is done in the High Resolution Schemes based on Total Variation Diminishing (TVD) (Harten, 1983) or Normalized Variable Diagram (NVD) (Leonard, 1991).

In the particular case of OpenFOAM® the whole HRS framework is implemented around TVD schemes, therefore, those schemes which are based in

the NVD diagram have to be rewritten in TVD form. Following Darwish & Moukalled (Darwish and Moukalled, 2003) the face value can be obtained by means of Eqn. (2.38)

$$\phi_f = \phi_N + \frac{1}{2}\psi(r_f)(\phi_P - \phi_N) \quad (2.38)$$

where $\psi(r_f)$ is the Sweby's function (Sweby, 1984; Hirsch, 2007) for a particular TVD limiter depending on r_f the ratio of consecutive gradients of the solution defined as is Eqn. (2.39) (see Figure 2.6)

$$r_f = \frac{\phi_P - \phi_U}{\phi_N - \phi_P} \quad (2.39)$$

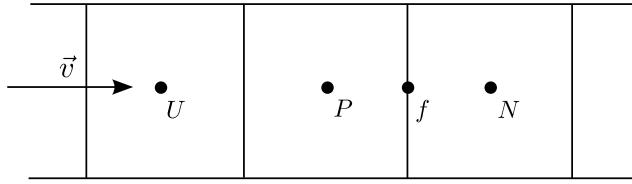


Figure 2.6: Cell array in a 1D dimensional mesh showing the nomenclature of neighbor cells to face f .

Comparing Eqn. (2.39) with Eqn. (2.37) it follows that the blending factor λ is related to Sweby's function as in Eqn. (2.40)

$$\lambda = \frac{1}{2}\psi(r_f) \quad (2.40)$$

The above expressions are strictly valid for structured meshes, where cell U can be univocally determined. In the case of an unstructured mesh a better expression was derived (Darwish and Moukalled, 2003) which is shown in Eqn. (2.41)

$$r_f = \frac{2\vec{\nabla}\phi_P \cdot \vec{d}_{PN}}{\phi_N - \phi_P} - 1 \quad (2.41)$$

where $\vec{\nabla}\phi_P$ is the ϕ gradient in a given cell.

In the case of NVD schemes they are written in terms of $\tilde{\phi}_f$ and $\tilde{\phi}_C$ (Leonard, 1991; Jasak et al., 1999), which are defined as in Eqn. (2.42)

$$\tilde{\phi}_C = \frac{\phi_P - \phi_U}{\phi_N - \phi_U} \quad \tilde{\phi}_f = \frac{\phi_f - \phi_U}{\phi_N - \phi_U} \quad (2.42)$$

These variables are related by the NVD functions as $\tilde{\phi}_f = f(\tilde{\phi}_C)$, where $f(\tilde{\phi}_C)$ is function inscribed in the NVD diagram. In order to use NVD schemes in OpenFOAM® it is necessary to translate the limiter expression in terms of r and $\psi(r)$. Hirsch (Hirsch, 2007) gives an equivalence formula which is shown in Eqn. (2.43)

$$\tilde{\phi}_f = \left[1 + \frac{1}{2}\psi'(r') \right] \tilde{\phi}_C \quad (2.43)$$

where r' is defined as $r' = (\phi_N - \phi_P)/(\phi_P - \phi_U)$ and $\psi'(r')$ is the expression of the Sweby's function for a given limiter in terms of r' . Darwish and Moukalled (Darwish and Moukalled, 2003) gave a relationship between $\psi'(r')$ and $\psi(r)$ such that $1/r'\psi'(r') = \psi(r)$, thus isolating $\psi'(r')$ the result is given by in Eqn. (2.44)

$$\psi'(r') = r' \psi(r) \quad (2.44)$$

replacing $\psi'(r')$ in Eqn. (2.43) a new equivalence formula is found which is shown in Eqn. (2.45)

$$\psi(r) = \frac{\frac{\tilde{\phi}_f}{\tilde{\phi}_C} - 1}{r'} \quad (2.45)$$

now, since $r' = \frac{1}{\tilde{\phi}_C} - 1$ (Hirsch, 2007) then the above expression can be presented as in Eqn. (2.46)

$$\psi(r) = 2 \frac{\frac{\tilde{\phi}_f}{\tilde{\phi}_C} - 1}{\frac{1}{\tilde{\phi}_C} - 1} \quad (2.46)$$

Finally, recalling the OpenFOAM®'s version of Sweby's function given by Eqn. (2.40), $\lambda = \frac{1}{2}\psi(r_f)$ the desired relationship between the NVD formulation and the OpenFOAM®'s TVD framework is given by Eqn. (2.47)

$$\lambda = \frac{\frac{\tilde{\phi}_f}{\phi_C} - 1}{\frac{1}{\phi_C} - 1} \quad (2.47)$$

2.6 Flux Corrected Transport

The Flux Corrected Transport (FCT) is a technique introduced by Boris and Book (Boris and Book, 1973) and improved by Zalesak (Zalesak, 1979) as a way to guarantee boundedness in the solution of hyperbolic problems. Following Kuzmin *et.al* (Kuzmin et al., 2003) Zalesak's limiter remains the only genuinely multidimensional high-resolution scheme available to date.

2.6.1 Theoretical foundation

Consider the solution of Eqn. (2.48)

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \vec{F} = 0 \quad (2.48)$$

by means of the Finite Volume Method (FVM), where ϕ is the transported quantity and \vec{F} a flux of this quantity, being this flux a linear or nonlinear function of ϕ . The solution of the problem by an explicit temporal scheme reads (Equation 2.49):

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} V + \sum_f (\vec{F}^n \cdot \vec{S})_f = 0 \quad (2.49)$$

where i is the index that identifies the analyzed cell, n is the index for temporal steps, V the cell volume, f the i cell face index and \vec{S} the face area vector for the f -th face of cell i . Finally, isolating ϕ_i^{n+1} (Equation 2.50) for a 1-D equispaced grid:

$$\phi_i^{n+1} = \phi_i^n - \frac{\Delta t}{V} (F_{i+1/2}^n - F_{i-1/2}^n) \quad (2.50)$$

where $F^n = (\vec{F}^n \cdot \vec{S})_f$ is the total flux due to the transportive effect of a velocity. The values of the flux depend on many variables but particularly on

the values of ϕ at faces. Boundedness of the temporal solution can be achieved via face value limiting, such as in TVD/NVD schemes, or by limiting the face fluxes. The values of F are obtained by a lower order and bounded method and a limited portion of the values obtained by a high order and possible unbounded method. The sequence described by Zalesak and written as is present in OpenFOAM®'s context can be summarized as follows:

1. Compute F^L , the transportive flux given by some low order scheme which guarantees to give monotonic results.
2. Compute F^H , the transportive flux given by some high order scheme.
3. Define the anti-diffusive flux $A = F^H - F^L$.
4. Compute de corrected flux $F^C = F^L + \lambda A$, with $0 \leq \lambda \leq 1$.
5. Solve the equation by the given temporal scheme (Equation 2.50) using corrected fluxes:

$$\phi_i^{n+1} = \phi_i^n - \frac{\Delta t}{V} (F_{i+1/2}^C - F_{i-1/2}^C) \quad (2.51)$$

The procedure is applied in such a way that no new extrema are found in ϕ_i^{n+1} with respect to the previous time-step. The critical step is clearly the fourth, where it is necessary to find the λ weighting factors. By means of these λ 's it is possible to measure the effect of high order schemes in flux calculation. If $\lambda = 0$ the flux used for time integration has low order, on the contrary with $\lambda = 1$ the flux used has high order.

2.6.2 Zalesak's weighting factors

To start the weighting factor determination given by Zalesak (Zalesak, 1979; Gopala and van Wachem, 2008) it is worthy to recall that given a cell i , its value at time $n + 1$ depends on the net flux in the cell as was shown in Eqn. (2.50). The net flux is the summation of all fluxes into cell i and away from cell i . The value of cell i at time $n + 1$ must be neither

greater than a local maximum nor lesser than a local minimum (due to transportive properties of equation which is being solved). Thus, the net flux must be corrected in order not to create new maximum or minimum [See Eqn. (2.51)].

Eqn. (2.51) can be rewritten as in Eqn. (2.52),

$$\begin{aligned}\phi_i^{n+1} &= (\phi_i^L)^n - \frac{\Delta t}{V} (\lambda_{i+1/2} A_{i+1/2} - \lambda_{i-1/2} A_{i-1/2}) \\ &= \phi_i^n - \frac{\Delta t}{V} (F_{i+1/2}^L - F_{i-1/2}^L) - \frac{\Delta t}{V} (\lambda_{i+1/2} A_{i+1/2} - \lambda_{i-1/2} A_{i-1/2})\end{aligned}\quad (2.52)$$

In this way, the values at time $n + 1$ can be calculated in two steps, firstly by a low order flux, F^L , and then adding a high-order or anti-diffusive flux, A , limited by the λ factors.

Once time advancement is obtained via a low order flux, giving a bounded solution, the possible unboundedness will be due to anti-diffusive fluxes. Then, a conservative criterion is used by which the maximum is created only due to the inflow and the minimum only by the outflow. These flows are calculated as a summation of all face inflows and all face outflows.

Supposing a cell is neighbored by three other cells and the flux in two faces is entering into the cell, then, there are three possibilities for the third flux:

1. This flux is away from the cell, then, considering the flux as the sum of all inflows the result is an overestimated net flow.
2. This flux is null, then summation of all inflows gives the correct net flow.
3. This flux is into the cell, then this flux had to be taken in account in the prior summation.

Therefore, limiting this inflow summation such that value at time $n + 1$ do not reach the local maximum guarantees the boundedness. The same concept is applied for outflow summation. As can be seen, this criterion is excessive in

cases when outflow summation is similar to inflow summation.

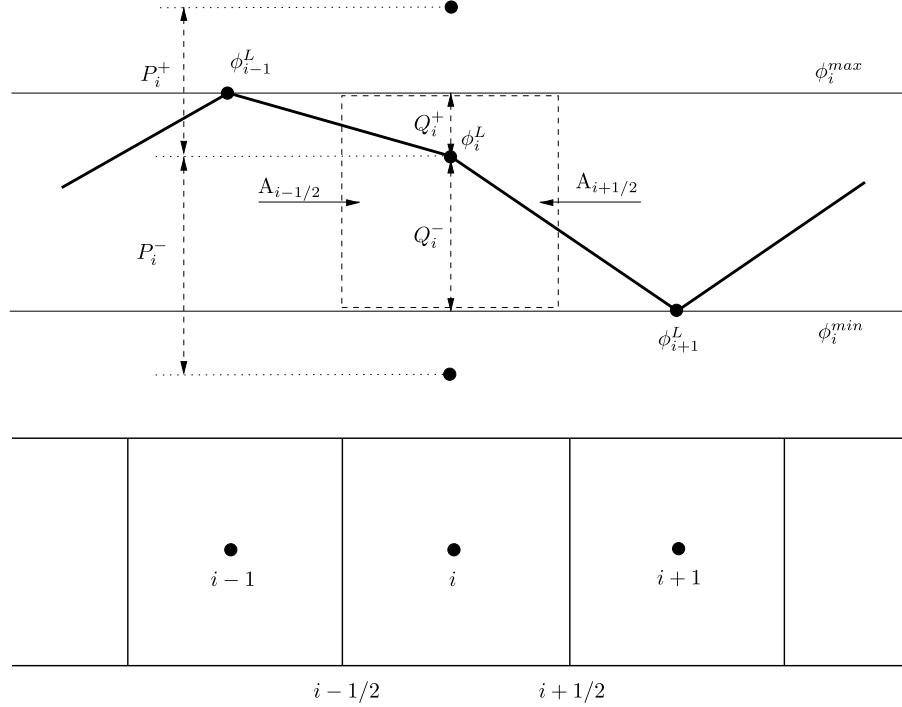


Figure 2.7: One dimensional geometry and magnitudes for Zalesak's limiter. Based on (Kuzmin et al., 2003).

Based on these concepts the following quantities can be defined (see Figure 2.7), P_i^\pm are the summations of inflows and outflows for A , $Q_i^+ = (\phi_i^{max} - \phi_i^L) \frac{\Delta t}{V}$, $Q_i^- = (\phi_i^L - \phi_i^{min}) \frac{\Delta t}{V}$ are the net fluxes in order to reach the local maximum and minimum. Now, it is possible to calculate the maximum fraction of A that can be used in order not to generate new maximum nor minimum, as in Eqn. (2.53)

$$\lambda_i^\pm = \begin{cases} \min \left\{ 1, Q_i^\pm / P_i^\pm \right\}, & \text{if } P_i^\pm > 0, \\ 0, & \text{if } P_i^\pm = 0 \end{cases} \quad (2.53)$$

Finally each cell has its own λ 's; now it is necessary to obtain the face λ 's. The calculus of λ 's was done by means of fluxes that take in account the face area. Thus, the proportional part of anti-diffusive fluxes given by the λ 's can

be directly applied to fluxes on each face regardless of its area. On the other hand each face is shared by two cells, then the flux in this face is into one cell and away from the other. In this sense each anti-diffusive flux can be limited by two λ 's; the logical decision, in order to guarantee boundedness is to choose the smaller one as in Eqn. (2.54),

$$\lambda_{i+1/2} = \begin{cases} \min \{\lambda_{i+1}^+, \lambda_i^-\}, & \text{if } A_{i+1/2} \geq 0, \\ \min \{\lambda_{i+1}^-, \lambda_i^+\}, & \text{if } A_{i+1/2} < 0 \end{cases} \quad (2.54)$$

Finally, the method requires the determination of the local extrema; a simple but useful option is that given by Eqn. (2.55)

$$\begin{aligned} \phi_i^a &= \max(\phi_i^n, \phi_i^L) \\ \phi_i^{max} &= \max(\phi_{i-1}^a, \phi_i^a, \phi_{i+1}^a) \\ \phi_i^b &= \min(\phi_i^n, \phi_i^L) \\ \phi_i^{min} &= \min(\phi_{i-1}^b, \phi_i^b, \phi_{i+1}^b) \end{aligned} \quad (2.55)$$

This selection guarantees to preserve the extrema present in the last solution and not to generate new maximum nor minimum. Zalesak's FCT method has other improvements and implementation details which can be found in the original paper (Zalesak, 1979). The actual presentation gives the required background for OpenFOAM®'s implementation.

2.6.3 OpenFOAM®'s weighting factors

OpenFOAM® implementation of FCT theory is called MULES (Multidimensional Universal Limiter for Explicit Solution); it relies on similar concepts respect to Zalesak's limiter but the determination of λ 's is iterative. Another extension is the possibility of set global extrema for the problem, which is a key feature in multiphase flows. The whole procedure is presented in Algorithm 1.

This formulation takes into account that each cell has a net flux. Then the maxima and minima are not only calculated by the inflow and the outflow respectively, which is a quite conservative assumption. To do so, for example, the outflow is also used in the calculation of the limiters for the maxima. In

Algorithm 1 Steps for MULES limiter

1. Calculate local extrema as

$$\phi_i^a = \max(\phi_i^n, \phi_{i,N}^n)$$

$$\phi_i^b = \min(\phi_i^n, \phi_{i,N}^n)$$

where $\phi_{i,N}^n$ are all the neighbors by face for the i -th cell. In addition the inflows and outflows for each cell have to be calculated as $P^+ = -\sum_f A_f^-$ and $P^- = \sum_f A_f^+$, where A_f^- are the inflows and A_f^+ the outflows;

2. Correct the local extrema by the limits imposed by user's defined global extrema ϕ^{maxG} and ϕ^{minG}

$$\phi_i^a = \min(\phi^{maxG}, \phi_i^a)$$

$$\phi_i^b = \max(\phi^{minG}, \phi_i^b)$$

3. Find Q^\pm_i as

$$Q^+_i = \frac{V}{\Delta t} (\phi_i^a - \phi_i^n) + \sum_f F_f^L$$

$$Q^-_i = \frac{V}{\Delta t} (\phi_i^n - \phi_i^b) - \sum_f F_f^L$$

4. Set $\lambda_f^{\nu=1} = 1$ for all faces. Do the following loop **nLimiterIter** (hard-coded to three) times to find the final λ_f 's

$$\lambda^{\mp,\nu+1}_i = \max \left[\min \left(\frac{\pm \sum_f \lambda_f^\nu A_f^\pm + Q^\pm_i}{P^\pm_i}, 1 \right), 0 \right]$$

$$\lambda_f^{\nu+1} = \begin{cases} \min \{ \lambda_P^{+, \nu+1}, \lambda_N^{-, \nu+1} \}, & \text{if } A_{i+1/2} \geq 0, \\ \min \{ \lambda_P^{-, \nu+1}, \lambda_N^{+, \nu+1} \}, & \text{if } A_{i+1/2} < 0 \end{cases}$$

where λ_P and λ_N represent the λ 's for the owner and neighbor cell of a given face f .

case of small inflows or outflows the values of λ 's rapidly converge to values close to Zalesak's ones.

2.7 Discretization of the Navier-Stokes equations

The presented discretization methods for a general advection-diffusion-reaction equation in terms of the cell-centered finite volume method set the basis for the discretization of the Navier-Stokes system, which is presented in Eqn. (2.56)

$$\begin{cases} \frac{\partial \vec{v}}{\partial t} + \vec{\nabla} \cdot (\vec{v} \otimes \vec{v}) = -\vec{\nabla} p + \vec{\nabla} \cdot (\nu \vec{\nabla} \vec{v}) \\ \vec{\nabla} \cdot \vec{v} = 0 \end{cases} \quad (2.56)$$

Here the momentum equation is presented with the continuity equation which corresponds to the case of constant density in space and time, thus, it is a case of incompressible flow. This system deals with three principal issues, first, the incompressibility which leads to a lack of pressure evolution equation and requires a special treatment for pressure-velocity coupling, the left two other issues are related to the advective term which has to be stabilized and particularly by its non-linearity. The stabilization is treated by the methods early presented in this chapter. Regarding the non-linearity, it can be solved by using a non-linear system or by the linearization which is the chosen option. Therefore, the advective term is linearized using the assumption of small Courant numbers ($\mathbf{Co} < 1$) and then $\vec{v}^0 \cong \vec{v}$, as is presented in Eqn. (2.57)

$$\begin{aligned} \int_{\Omega} \vec{\nabla} \cdot (\vec{v} \otimes \vec{v}^0) d\Omega &\cong \sum_f \vec{v}_f \vec{v}_f^0 \cdot \vec{S}_f \\ &= \sum_f F^0 \vec{v}_f \\ &= a_P \vec{v} + \sum_f a_N \vec{v}_N \end{aligned} \quad (2.57)$$

where \vec{v}^0 is the velocity at the previous time-step, a_P are the diagonal coefficients of the discretization matrix, a_N the off-diagonal ones and F^0 the

face flux at the previous time-step. All these quantities are a function of \vec{v}^0 . Here it is important to note that, for purposes which will be clear soon, the pressure is discretized at cell centres and the velocities are calculated at faces in the form of fluxes, therefore the flux F^0 is only calculated from \vec{v}^0 eventually at the first time-step; in the rest of the calculation this flux is taken from the previous time-step pressure-velocity loop, which assures the enforcement of the continuity equation (conservative flux). Respect to the incompressibility restriction it will be treated in this work by the PISO (Pressure Implicit Split of Operators) procedure (Issa, 1986), as is implemented in OpenFOAM® (Jasak, 1996; Peng Karrholm, 2008).

2.7.1 Derivation of an equation for the pressure

Since no evolution equation is given for the pressure it is necessary to devise a method to obtain at least a discretized equation which allows to solve for the pressure. Writing the momentum equation in a semi-discretized form as in Eqn. (2.58) it is possible to start its derivation

$$a_P \vec{v}_P = \vec{H}(\vec{v}) - \vec{\nabla} p \quad (2.58)$$

This equation is obtained by the integral form of the momentum equation using the discretization method described previously. The pressure gradient is not discretized at this time, which follows the spirit of the Rhie and Chow interpolation procedure (Rhie and Chow, 1983). It is important to note that in order to allow future face interpolations of matrix's coefficients the discretization has been divided by the volume of each cell.

The $\vec{H}(\vec{v})$ operator accounts for the advective and diffusive terms as well as all the source terms including the source part of the transient term, therefore it results to be which is shown in Eqn. (2.59)

$$\vec{H}(\vec{v}) = - \sum_f a_N \vec{v}_N + \frac{\vec{v}^0}{\Delta t} \quad (2.59)$$

Now, discretizing the continuity equation as in Eqn. (2.60)

$$\vec{\nabla} \cdot \vec{v} = \sum_f \vec{v}_f \cdot \vec{S}_f = 0 \quad (2.60)$$

and isolating the velocity at cell-centers from the discretized version of momentum equation [Eqn. (2.58)] as in Eqn. (2.61)

$$\vec{v}_P = \frac{\vec{H}(\vec{v})}{a_P} - \frac{1}{a_P} \vec{\nabla} p \quad (2.61)$$

These velocities can be interpolated at faces as in Eqn. (2.62):

$$\vec{v}_f = \left(\frac{\vec{H}(\vec{v})}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f \left(\vec{\nabla} p \right)_f \quad (2.62)$$

which gives a base for face flux calculation. Now, using the obtained expression for the velocity at faces it can be substituted in Eqn. (2.60) to find an equation for the pressure [Eqn. (2.63)]

$$\vec{\nabla} \cdot \left(\frac{1}{a_P} \vec{\nabla} p \right) = \vec{\nabla} \cdot \left(\frac{\vec{H}(\vec{v})}{a_P} \right) \quad (2.63)$$

Finally the set of discrete equations for the Navier-Stokes system results to be which is shown in Eqn. (2.64)

$$\begin{cases} a_P \vec{v}_P = \vec{H}(\vec{v}) - \sum_f \vec{S}(p)_f \\ \sum_f \left[\left(\frac{1}{a_P} \right)_f \left(\vec{\nabla} p \right)_f \right] \cdot \vec{S}_f = \sum_f \left(\frac{\vec{H}(\vec{v})}{a_P} \right)_f \cdot \vec{S}_f \end{cases} \quad (2.64)$$

where $\vec{S}(p)$ is a source term give by the gradient of the pressure. In addition it is necessary a way to assemble the face flux F , see Eqn. (2.65)

$$F = \vec{v}_f \cdot \vec{S}_f = \left[\left(\frac{\vec{H}(\vec{v})}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f \left(\vec{\nabla} p \right)_f \right] \cdot \vec{S}_f \quad (2.65)$$

2.7.2 The PISO algorithm for pressure-velocity coupling

Having a system to solve is now possible to derive an algorithm for pressure-velocity coupling. As was stated previously this algorithm is based on the PISO method, which uses a segregated approach. In this approach the equations of the system are solved one by one in a loop looking for the convergence to a solution in pressure and velocity. The complete algorithm is shown in Algorithm 2 and shows the sequence necessary to obtain a set of pressure and velocity that satisfies the Navier-Stokes system. To this end and to avoid the pressure checkerboarding present in collocated meshes (Ferziger and Peric, 2002; Versteeg and Malalasekera, 2007), the pressure, p , is solved at the cell-centres and the velocity, \vec{v} , is calculated at the faces in the form of a flux, F , as is shown in Figure 2.8.

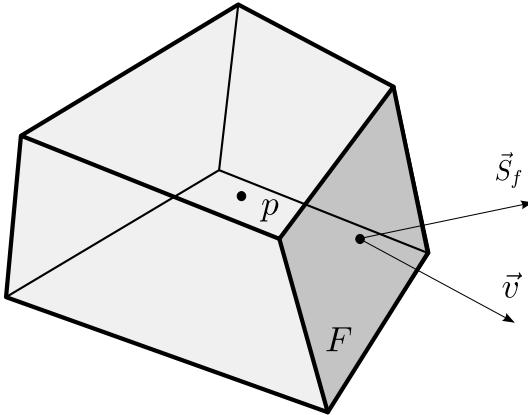


Figure 2.8: Location of variables p , \vec{v} and F in a collocated mesh to avoid pressure checkerboarding.

This method allows to obtain a staggered scheme from the collocated mesh. The iterative solution has a feedback through the operator $\vec{H}(\vec{v})$ which uses the \vec{v} calculated in the previous iteration and takes into account the influence of the solution of the pressure equation. This technique does not take into account that $\vec{H}(\vec{v})$ operator's coefficients also change with \vec{v} due to the nonlinear

coupling of the advective term previously discussed. The hypothesis used for this solution is that the non-linear coupling of the advective term is less important than the pressure-velocity coupling. Thus, the coefficients of $\vec{H}(\vec{v})$ are only re-calculated at the next time-step or doing outer corrections.

Algorithm 2 PISO method for pressure-velocity coupling

1. Obtain an approximation for the velocity field solving the momentum equation [first equation in Eqn. (2.64)], using the previous time-step pressure for the pressure gradient and the previous time-step flux for advective term linearization. This step is called the *momentum predictor*

$$a_P \vec{v}_P = \vec{H}(\vec{v}) - \sum_f \vec{S}(p^0)_f$$

2. Assemble an approximation of the face flux which is also needed in the r.h.s. of the pressure equation

$$\tilde{F}_f = \left(\frac{\vec{H}(\vec{v})}{a_P} \right)_f \cdot \vec{S}_f$$

3. Using the approximated velocity obtained, assemble and solve the pressure equation [second equation in Eqn. (2.64)], this step is called the *pressure solution*

$$\sum_f \left[\left(\frac{1}{a_P} \right)_f (\vec{\nabla} p)_f \right] \cdot \vec{S}_f = \sum_f \tilde{F}_f$$

4. Find the final flux correcting the approximated flux by the pressure effect using Eqn. (2.65)

$$F = \tilde{F} - \left[\left(\frac{1}{a_P} \right)_f (\vec{\nabla} p)_f \right] \cdot \vec{S}_f$$

5. Correct the cell-centered velocity due to the new pressure distribution. This stage is called the *explicit velocity correction* and is achieved by Eqn. (2.61)

$$\vec{v}_P = \frac{\vec{H}(\vec{v})}{a_P} - \frac{1}{a_P} \vec{\nabla} p$$

6. Go to step 2 nCorrectors times or end.
-

Chapter 3

Algebraic Slip Mixture Model characterization

3.1 Introduction

The Algebraic Slip Mixture Model (ASMM) is a multi-phase model for n interpenetrated phases based on the Multi-fluid model (Ishii, 1975; Ishii and Hibiki, 2010). In this model all the phases are treated as a mixture which exhibits mean properties for density and viscosity. In the Multi-fluid model a mass and momentum equation are solved for each phase; on the other hand the ASMM reduces the system to a mass and momentum equation for the whole mixture and one mass conservation equation for each of the $n - 1$ phases. Since the momentum equations for these $n - 1$ phases are not solved, additional *algebraic* relations for each phase velocities respect to the mixture velocity are given. These algebraic relationships for the slip velocities give the name to this mixture model. Finally a closure law for all phases volume fractions is also included. Even when ASMM is physically more limited than the Multi-fluid model its results are in some particular cases comparable to that model due to the lack of closure laws available for the latter. In addition ASMM results to be considerably less computationally expensive, and robust. The objective of this chapter is to revise the fundamentals of the ASMM method as a derivation from the Multi-fluid model, discussing the principal issues in solving the resulting

system of equations. These issues are related to pressure-velocity coupling and the hyperbolic nature of the volume fraction conservation equation. In order to set a basis for further analysis a one dimensional ASMM system is solved finding semi-analytical expressions for the void fraction, pressure and velocity.

3.2 Theory fundamentals

The starting point for the formulation of the ASMM is the Multi-fluid method which solves mass and momentum equations for each phase. Following the theoretical foundation given by Manninen *et.al.* (Manninen et al., 1996) and using Fluent's nomenclature (Fluent Inc., 2006) the basic equations of the Two-Fluid method are presented, i.e. the mass conservation for each phase, Eqn. (3.1) and the corresponding momentum equation, Eqn. (3.2)

$$\frac{\partial(\alpha_k \rho_k)}{\partial t} + \vec{\nabla} \cdot (\alpha_k \rho_k \vec{v}_k) = \Gamma_k \quad (3.1)$$

$$\frac{\partial}{\partial t}(\alpha_k \rho_k \vec{v}_k) + \vec{\nabla} \cdot (\alpha_k \rho_k \vec{v}_k \otimes \vec{v}_k) = -\alpha_k \vec{\nabla} p_k + \vec{\nabla} \cdot (\alpha_k \bar{\tau}_k) + \alpha_k \rho_k \vec{g} + \vec{M}_k \quad (3.2)$$

where α_k is the volume fraction of phase k , which obeys the geometrical closure law $\sum_k \alpha_k = 1$. The term Γ_k represents the rate of mass generation of phase k at the interface and \vec{M}_k is the average interfacial momentum source for phase k . It is important to note that most of the uncertainties in the results of Multi-fluids method come from the determination of this last term. Finally, $\bar{\tau}_k$ represents the viscous stress tensor. The presented set of equations has been given in the so-called mass-weighted average for the velocity.

3.2.1 Velocity of center of mass based formulation

The Mixture Model can be formulated using either the so called *velocity of center-of-mass*, *mixture velocity* or *mass averaged velocity* or in terms of the *velocity of center-of-volume*. Starting from the Multi-fluid method, the derivation of the velocity of center of mass based formulation is as follows.

From the continuity equation for phase k [Eqn. (3.1)] and summing over all phases the result is given in Eqn. (3.3)

$$\frac{\partial}{\partial t} \sum_{k=1}^n (\alpha_k \rho_k) + \vec{\nabla} \cdot \sum_{k=1}^n (\alpha_k \rho_k \vec{v}_k) = \sum_{k=1}^n \Gamma_k \quad (3.3)$$

where n is the number of phases. Since the overall mass is conserved, the right hand side of Eqn. (3.3) vanishes, and then the continuity equation for the mixture is written as in Eqn. (3.4)

$$\frac{\partial}{\partial t} (\rho_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m) = 0 \quad (3.4)$$

where \vec{v}_m is the velocity of center of mass defined by Eqn. (3.5)

$$\vec{v}_m = \frac{\sum_{k=1}^n \alpha_k \rho_k \vec{v}_k}{\rho_m} \quad (3.5)$$

and ρ_m the mixture density which is calculated as in Eqn. (3.6)

$$\rho_m = \sum_{k=1}^n \alpha_k \rho_k \quad (3.6)$$

The mixture momentum equation is obtained by the summation of each phase momentum equation as is presented in Eqn. (3.7)

$$\begin{aligned} & \frac{\partial}{\partial t} \sum_{k=1}^n \alpha_k \rho_k \vec{v}_k + \vec{\nabla} \cdot \sum_{k=1}^n \alpha_k \rho_k \vec{v}_k \otimes \vec{v}_k \\ &= - \sum_{k=1}^n \alpha_k \vec{\nabla} p_k + \vec{\nabla} \cdot \sum_{k=1}^n (\alpha_k \vec{\tau}_k) + \sum_{k=1}^n \alpha_k \rho_k \vec{g} + \sum_{k=1}^n \vec{M}_k \end{aligned} \quad (3.7)$$

Using the definitions of Eqn. (3.6) and Eqn. (3.5) it is possible to rewrite the summation of the advective terms of Eqn. (3.7), as is shown in Eqn. (3.8)

$$\vec{\nabla} \cdot \sum_{k=1}^n \alpha_k \rho_k \vec{v}_k \otimes \vec{v}_k = \vec{\nabla} \cdot (\rho_m \vec{v}_m \otimes \vec{v}_m) + \vec{\nabla} \cdot \left(\sum_{k=1}^n \alpha_k \rho_k \vec{v}_{\text{dr},k} \otimes \vec{v}_{\text{dr},k} \right) \quad (3.8)$$

Replacing in Eqn. (3.7) and using the viscous tensor definition for the stress

tensor (Manninen et al., 1996), the final form of the momentum equation for the mixture is obtained in Eqn. (3.9).

$$\begin{aligned} \frac{\partial}{\partial t}(\rho_m \vec{v}_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m \otimes \vec{v}_m) &= -\vec{\nabla} p + \vec{\nabla} \cdot \left[\mu_m \left(\vec{\nabla} \vec{v}_m + \vec{\nabla} \vec{v}_m^T \right) \right] + \rho_m \vec{g} + \\ \vec{F} - \vec{\nabla} \cdot (\sum_{k=1}^n \alpha_k \rho_k \vec{v}_{dr,k} \otimes \vec{v}_{dr,k}) \end{aligned} \quad (3.9)$$

where \vec{F} is a body force, p is the pressure for the mixture which is often taken to be equal to the pressure of each one of the phases (Drew, 1983) and μ_m is the dynamic viscosity of the mixture given by Eqn. (3.10)

$$\mu_m = \sum_{k=1}^n \alpha_k \mu_k \quad (3.10)$$

This simple expression is valid when the effects of the relative motion and the interfacial deformation are not important (Manninen et al., 1996) and can be used for bubbly flows with high viscosity ratios between the fluid and bubbles (Ishii and Hibiki, 2010). Finally $\vec{v}_{dr,k}$ is the drift velocity (relative velocity between k phase velocity and velocity of center of mass, see Figure 3.1) for the secondary phase k defined by Eqn. (3.11).

$$\vec{v}_{dr,k} = \vec{v}_k - \vec{v}_m \quad (3.11)$$

The drift velocity has no physical meaning but can be related to the relative velocity \vec{v}_{pq} (See Figure 3.1) for a given phase, p , respect to other phase, q . The relative velocity, \vec{v}_{pq} is then defined by Eqn. (3.12)

$$\vec{v}_{pq} = \vec{v}_p - \vec{v}_q \quad (3.12)$$

Now, defining the mass fraction for any phase, k , as in Eqn. (3.13)

$$c_k = \frac{\alpha_k \rho_k}{\rho_m} \quad (3.13)$$

it is possible to relate the drift velocity and the relative velocity, \vec{v}_{pq} , by Eqn. (3.14):

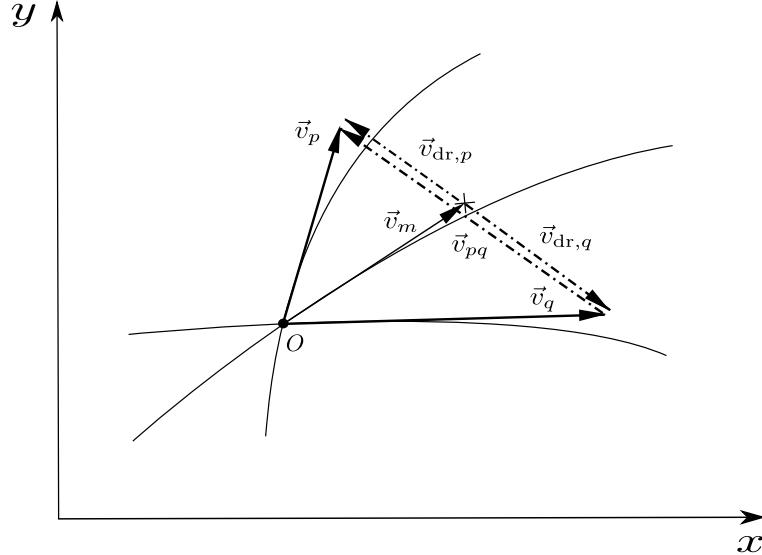


Figure 3.1: Example of the velocity of center of mass \vec{v}_m , relative velocity \vec{v}_{pq} and drift velocities $\vec{v}_{\text{dr},p}$ and $\vec{v}_{\text{dr},q}$ for a two phase system [adapted from (Ishii and Hibiki, 2010)].

$$\vec{v}_{\text{dr},p} = \vec{v}_{qp} - \sum_{k=1}^n c_k \vec{v}_{qk} \quad (3.14)$$

If only one dispersed phase is present Eqn. (3.14) can be written in a simpler form [See (Manninen et al., 1996) Eqn. (27)] as in Eqn. (3.15)

$$\vec{v}_{\text{dr},p} = \vec{v}_{pq} (1 - c_p) \quad (3.15)$$

where c_p is the mass fraction for the dispersed phase.

Using the assumption that a local equilibrium is reached in a short spatial length scale it is possible to use an algebraic formulation for relative velocities. Following Manninen (Manninen et al., 1996) and taking into account only the drag force, the slip velocity between two phases, p and q , can be obtained by Eqn. (3.16)

$$|\vec{v}_{pq}| \vec{v}_{pq} = \frac{4d_p}{3C_D} \frac{(\rho_p - \rho_m)}{\rho_q} \vec{a} \quad (3.16)$$

where d_p is the diameter of the droplets or bubbles of the secondary phase p , C_D defined as in Eqn. (3.17), is the drag coefficient and \vec{a} (Eqn. 3.18) is the secondary-phase acceleration.

$$C_D = \frac{24}{\text{Re}_p} (1 + 0.15 \text{Re}_p^{0.687}) \quad \text{Re}_p \leq 1000 \quad (3.17)$$

where $\text{Re} = \frac{d_p \rho_q |\vec{v}_{pq}|}{\mu_q}$.

$$\vec{a} = \vec{g} - (\vec{v}_m \cdot \vec{\nabla}) \vec{v}_m - \frac{\partial \vec{v}_m}{\partial t} \quad (3.18)$$

Finally, the expression for the slip velocity given in Eqn. (3.19) is obtained replacing Eqns. (3.17) and (3.18) in (3.16)

$$\vec{v}_{pq} = \frac{\tau_p}{f_{\text{drag}}} \frac{(\rho_p - \rho_m)}{\rho_p} \vec{a} \quad (3.19)$$

where $\tau_p = \frac{\rho_p d_p^2}{18\mu_q}$ is the particle relaxation time and f_{drag} is given by Eqn. (3.20), which follows the drag laws given by Schiller and Naumann (Schiller and Naumann, 1935).

$$f_{\text{drag}} = \begin{cases} 1 + 0.15 \text{Re}_p^{0.687} & \text{Re}_p \leq 1000 \\ 0.0183 \text{Re}_p & \text{Re}_p > 1000 \end{cases} \quad (3.20)$$

Finally, it is necessary to have an equation for the evolution of the secondary phases' volume fraction. Starting from the continuity equation of phase p , the volume fraction equation for that phase can be obtained, Eqn. (3.21):

$$\frac{\partial}{\partial t} (\alpha_p \rho_p) + \vec{\nabla} \cdot (\alpha_p \rho_p \vec{v}_m) = -\vec{\nabla} \cdot (\alpha_p \rho_p \vec{v}_{\text{dr},p}) \quad (3.21)$$

In summary, the system of equations for mixture mass and momentum balance and secondary phase mass conservation can be written for a two component mixture [see (Manninen et al., 1996), section 3.5.1] as in Eqn. (3.22)

$$\left\{ \begin{array}{l} \frac{\partial}{\partial t}(\rho_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m) = 0 \\ \frac{\partial}{\partial t}(\rho_m \vec{v}_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m \otimes \vec{v}_m) = -\vec{\nabla} p + \vec{\nabla} \cdot \left[\mu_m \left(\vec{\nabla} \vec{v}_m + \vec{\nabla} \vec{v}_m^T \right) \right] + \\ \quad \rho_m \vec{g} - \vec{\nabla} \cdot [\rho_m c_p (1 - c_p) \vec{v}_{pq} \otimes \vec{v}_{pq}] \\ \frac{\partial}{\partial t}(\alpha_p) + \vec{\nabla} \cdot (\alpha_p \vec{v}_m) = -\vec{\nabla} \cdot [\alpha_p (1 - c_p) \vec{v}_{pq}] \end{array} \right. \quad (3.22)$$

where the momentum equation has a term accounting for the momentum exchanging between the phases, which is calculated by the drift tensor, $\overline{\overline{\tau_D}}$ in Eqn. (3.23)

$$\overline{\overline{\tau_D}} = \rho_m c_p (1 - c_p) \vec{v}_{pq} \otimes \vec{v}_{pq} \quad (3.23)$$

This system of three equations has three unknowns, they are: \vec{v}_m , p and α_p . Respect to ρ_m it is linked to α_p via its constitutive equation, Eqn. (3.6). As is usual in incompressible problems the pressure has no evolution equation, then it becomes a Lagrange Multiplier for the restriction given by the mixture density transport equation. This characteristic leads to a pressure-velocity coupling that may be treated in several ways. Among the most popular methods there are the Fractional-Step or PISO/SIMPLE like methods (Gastaldo et al., 2008). This issue appears also in reacting flows (Babik et al., 2005; Najm et al., 1998; Knio et al., 1999), the Low-Mach solvers applied in that problems are also an inspiration for the solution of ASMM problems. In addition α_p has to be bounded in the $[0, 1]$ interval to have physical meaning. Since v_m is not divergence free and since the momentum and mixture conservation equations depend on α_p the boundedness is not a direct consequence of the correct discretization of the third equation in Eqn. (3.22), but of the whole system (Gastaldo et al., 2011).

3.2.2 The role of α_p equation

As it was stated previously the α_p equation represents the mass conservation of the secondary, dispersed phase. Its relevance is particularly important when the dynamics of the problem strongly depends on the evolution of the phases. The α_p equation can be written using the usual notation in hyperbolic equations analysis such as given by Eqn. (3.24)

$$\frac{\partial \alpha_p}{\partial t} + \vec{\nabla} \cdot \vec{F}(\alpha_p) = \frac{\partial \alpha_p}{\partial t} + \vec{\nabla} \cdot \left\{ \alpha_p \left[\vec{v}_m + \left(1 - \frac{\alpha_p \rho_p}{\rho_m} \right) \vec{v}_{pq} \right] \right\} = 0 \quad (3.24)$$

where $\vec{F}(\alpha_p)$ is the flux for this hyperbolic equation. It is, in general, a non-convex flux, therefore, single and compound waves can be present as a part of the solution (LeVeque, 2002). The kind of waves depends on the relative velocity selected law (\vec{v}_{pq}) and the particular initial conditions for each problem. This information is crucial to apply the necessary stabilization for advective terms.

For the sake of simplicity in the analysis it is valuable to set a general constitutive law for the relative velocity, \vec{v}_{pq} , as it is shown in Eqn. (3.25)

$$\vec{v}_{pq} = \vec{v}_{rc} (1 - \alpha_p)^a \quad (3.25)$$

where \vec{v}_{rc} and a are constants for the model. The \vec{v}_{rc} constant can be interpreted as the velocity of a single bubble or droplet moving in the continuum phase. This expression is flexible and allows to match several other models, for example, the previously presented Schiller and Naumann drag law can be fitted selecting an appropriate value for v_{rc} and with $0 \leq a \leq 1$. Other constitutive law of this kind is that was given by Barceló *et al.* (L.F. Barceló *et al.*, 2010) for their study of water-petroleum segregation. In addition, the book of Ishii & Hibiki (Ishii and Hibiki, 2010) provides a complete reference for drag laws in several industrial cases.

3.2.3 Velocity of center of volume or volumetric flux formulation

As was stated previously another formulation can be devised for ASMM in terms of the velocity of the center of volume or *volumetric flux*. This velocity is defined by Eqn. (3.26)

$$\vec{u} = \sum_{k=1}^n \alpha_k \vec{v}_k \quad (3.26)$$

which in the case of only two phases p and q simplifies to Eqn. (3.27)

$$\vec{u} = \alpha_p \vec{v}_p + \alpha_q \vec{v}_q \quad (3.27)$$

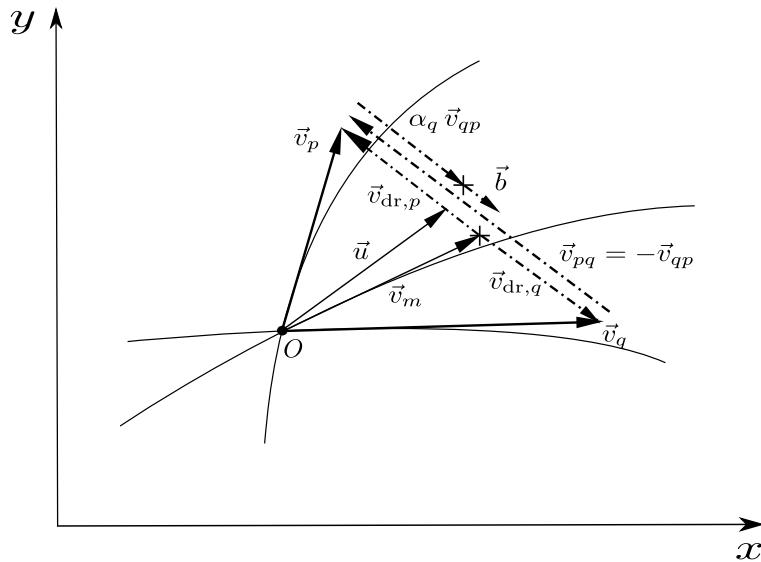


Figure 3.2: Example of velocity of center of volume \vec{u} , relative velocity \vec{v}_{pq} , drift velocities $\vec{v}_{dr,p}$ and $\vec{v}_{dr,q}$, velocity of center of mass \vec{v}_m , relation between velocity of center of volume and center of mass \vec{b} and other auxiliary quantities, for a bi-phasic system.

With the aid of Figure 3.2 it is possible to give a relationship between \vec{u} and the previously studied velocity of center of mass \vec{v}_m . This relationship is

given by vector the \vec{b} [see Eqn. (3.28)] for a bi-phasic system.

$$\vec{v}_m = \vec{u} + \vec{b} \quad (3.28)$$

From the figure it is clear that the value of \vec{b} is given by Eqn. (3.29)

$$\vec{b} = -\vec{v}_{dr,p} - \alpha_q \vec{v}_{qp} = \vec{v}_{qp} \left[1 - \frac{\alpha_p \rho_p}{\rho_m} - \alpha_q \right] \quad (3.29)$$

After some algebraic manipulations the final form is given in Eqn. (3.30)

$$\vec{b} = \vec{v}_{qp} \alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \quad (3.30)$$

Replacing Eqn. (3.30) in Eqn. (3.28), the desired relationship is obtained as is shown in Eqn. (3.31)

$$\vec{v}_m = \vec{u} + \alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \vec{v}_{qp} \quad (3.31)$$

Starting again from the Multi-fluid method it is possible to find a mass conservation equation for the mixture written in terms of the velocity of center of mass. Then, for a given phase k of the system its mass conservation equation (without sources) reads as in Eqn. (3.32)

$$\frac{\partial \alpha_k \rho_k}{\partial t} + \vec{\nabla} \cdot (\alpha_k \rho_k \vec{v}_k) = 0 \quad (3.32)$$

assuming constant densities for all phases and dividing each mass conservation by its corresponding density, the mass conservation equation becomes which is shown in Eqn. (3.33)

$$\frac{\partial \alpha_k}{\partial t} + \vec{\nabla} \cdot (\alpha_k \vec{v}_k) = 0 \quad (3.33)$$

then summing over all phases and recalling that $\sum_{k=1}^n \alpha_k = 1$ the final expression is given by Eqn. (3.34)

$$\vec{\nabla} \cdot \vec{u} = 0 \quad (3.34)$$

where \vec{u} is the previously defined velocity of center of volume [Eqn. (3.26)].

Now, taking a bi-phasic system with phases p and q the mass conservation equation for the primary (continuum) phase can be obtained, following Figure 3.2 the relationship given in Eqn. (3.35) is obtained.

$$\vec{v}_q = \vec{u} + (1 - \alpha_q) \vec{v}_{qp} \quad (3.35)$$

Replacing this expression in Eqn. (3.33) for the q phase allows to write the mass conservation equation for the q phase [Eqn. (3.36)]

$$\frac{\partial \alpha_q}{\partial t} + \vec{\nabla} \cdot (\alpha_q \vec{u}) + \vec{\nabla} \cdot [\alpha_q (1 - \alpha_q) \vec{v}_{qp}] = 0 \quad (3.36)$$

Finally the mixture momentum equation can be rewritten in terms of the primary phase void fraction α_q and the relative velocity of the primary phase respect to the secondary one \vec{v}_{qp} . Starting from the expression of the drift tensor $\overline{\overline{\tau_D}}$ in Eqn. (3.23) and doing some algebraic simplifications, it becomes which is shown in Eqn. (3.37)

$$\overline{\overline{\tau_D}} = \rho_m c_p (1 - c_p) \vec{v}_{pq} \otimes \vec{v}_{pq} = \alpha_p \rho_p \left(1 - \frac{\alpha_p \rho_p}{\rho_m} \right) \vec{v}_{pq} \otimes \vec{v}_{pq} \quad (3.37)$$

Now taking into account that $\vec{v}_{qp} = -\vec{v}_{pq}$ and $\alpha_q = 1 - \alpha_p$ the drift tensor can be expressed as in Eqn. (3.38)

$$\overline{\overline{\tau_D}} = \alpha_q (1 - \alpha_q) \frac{\rho_q \rho_p}{\rho_m} \vec{v}_{qp} \otimes \vec{v}_{qp} \quad (3.38)$$

In summary, the continuity equation and momentum balance for the mixture and the mass conservation equation for the primary phase in center of volume based formulation may be written as in Eqn. (3.39)

$$\left\{ \begin{array}{l} \vec{\nabla} \cdot \vec{u} = 0 \\ \frac{\partial}{\partial t}(\rho_m \vec{v}_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m \otimes \vec{v}_m) = -\vec{\nabla} p + \vec{\nabla} \cdot \left[\mu_m \left(\vec{\nabla} \vec{v}_m + \vec{\nabla} \vec{v}_m^T \right) \right] + \\ \quad \rho_m \vec{g} - \vec{\nabla} \cdot \left[\alpha_q (1 - \alpha_q) \frac{\rho_q \rho_p}{\rho_m} \vec{v}_{qp} \otimes \vec{v}_{qp} \right] \\ \frac{\partial \alpha_q}{\partial t} + \vec{\nabla} \cdot (\alpha_q \vec{u}) + \vec{\nabla} \cdot [\alpha_q (1 - \alpha_q) \vec{v}_{qp}] = 0 \end{array} \right. \quad (3.39)$$

In this formulation the momentum equation can also be written in terms of \vec{u} . The resulting system is often called the Drift-Flux Model (Manninen et al., 1996) because its relies in the calculation of fluxes (center of volume velocities) instead of velocities (center of mass velocities).

3.3 1D simplified formulation

The equation system in the center-of-mass based formulation obtained for a two-phase mixture (3.22) can be written in 1D. In addition, if the inviscid case is taken into account, the results are then expressed like in the system given in Eqn. (3.40).

$$\left\{ \begin{array}{l} \frac{\partial}{\partial t}(\rho_m) + \frac{\partial}{\partial z}(\rho_m v_m) = 0 \\ \frac{\partial}{\partial t}(\rho_m v_m) + \frac{\partial}{\partial z}(\rho_m v_m v_m) = -\frac{\partial}{\partial z} p - \frac{\partial}{\partial z} [\rho_m c_p (1 - c_p) v_{pq}^2] + \rho_m g \\ \frac{\partial}{\partial t}(\alpha_p) + \frac{\partial}{\partial z} \{ \alpha_p [v_m + (1 - c_p) v_{pq}] \} = 0 \end{array} \right. \quad (3.40)$$

where α_p , v_m and p are functions of the time, t , and the axial coordinate, z .

The hypothesis of inviscid flux is based on the fact that all viscous effects related to the dispersed phase are taken into account in the relative velocity definition. In addition the effects of the mixture viscosity are zero in the

transversal direction since no profile is developed in a one dimensional problem. The remaining term in the axial direction has marginal importance as will become clear in the chapter devoted to the implementation of a full 3D solver.

As it was stated in the previous section the system given in Eqn. (3.40) deals with the incompressibility and the determination of p . In Fractional-Step or SIMPLE¹/PISO like methods a prediction of v_m , α_p and p is done based in p data from previous time-step. Next v_m , p and eventually α_p are corrected assembling and solving a Poisson or Quasi-Armonic equation for p based on ρ_m conservation equation. In addition, as will be demonstrated later, the effects of the drift term in the momentum equation are negligible.

3.3.1 One dimensional semi-analytical solution for sedimentation

Starting from the formulation of ASMM in 1D and under particular boundary conditions it is possible to arrive to a semi-analytical solution for the system given in Eqn. (3.40). The considered case consists in the sedimentation of a mixture of two fluids with different densities, starting from a domain completely filled with the mixture, as it is shown in Figure 3.3.

The mixture is composed by two fluids, the more dense phase with density $\rho_q = 1000$ and the dispersed, less dense phase, with density $\rho_p = 1$, the gravitational acceleration is $g = -10$, while α_p represents the volume fraction of the dispersed phase. Starting from the initial condition with $\alpha_p = \alpha_p^0$ and due to buoyancy effects the less dense phase moves upwards and the more dense phase settles down at the bottom of the domain. The dynamics of the problem is governed by the relative velocity law, $v_{pq}(\alpha_p)$.

¹Semi-Implicit Method for Pressure-Linked Equations

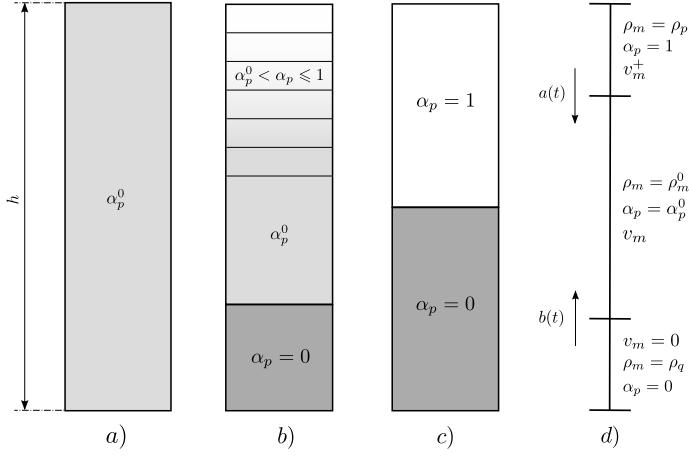


Figure 3.3: Simple sedimentation experiment. *a)* Initial conditions; *b)* settling transitory; *c)* steady state and *d)* scheme of problem variables.

Determination of the center-of-mass velocity

Recalling the mixture density conservation equation in Eqn. (3.40) (first equation) and the constitutive equation for the mixture density in Eqn. (3.6), they read as follows

$$\frac{\partial}{\partial t}(\rho_m) + \frac{\partial}{\partial z}(\rho_m v_m) = 0 \quad \rho_m = \sum_{k=1}^n \alpha_k \rho_k \quad (3.41)$$

Next, if only two phases are taken into account these expressions can be simplified as in Eqn. (3.42)

$$\frac{\partial}{\partial t}(\rho_m) + \frac{\partial}{\partial z}(\rho_m v_m) = 0 \quad \rho_m = \alpha_p \rho_p + (1 - \alpha_p) \rho_q \quad (3.42)$$

Now, taking the temporal derivative of ρ_m Eqn. (3.43) is obtained

$$\frac{\partial}{\partial t}(\rho_m) = (\rho_p - \rho_q) \frac{\partial}{\partial t}(\alpha_p) \quad (3.43)$$

this expression requires the evaluation of the temporal derivative of α_p which can be extracted from the α_p conservation equation in Eqn. (3.40) (third equation) as in Eqn. (3.44)

$$\frac{\partial}{\partial t}(\alpha_p) + \frac{\partial}{\partial z}\{\alpha_p[v_m + (1 - c_p)v_{pq}]\} = \frac{\partial}{\partial t}(\alpha_p) + \frac{\partial}{\partial z}(\alpha_p v_p) = 0 \quad (3.44)$$

then, isolating the temporal derivative Eqn. (3.45) is obtained

$$\frac{\partial}{\partial t}(\alpha_p) = -\frac{\partial}{\partial z}(\alpha_p v_p) \quad (3.45)$$

therefore, replacing the obtained expression in Eqn. (3.43) a new form of the temporal derivative of the mixture density is obtained, which is shown in Eqn. (3.46)

$$\frac{\partial}{\partial t}(\rho_m) = (\rho_q - \rho_p)\frac{\partial}{\partial z}(\alpha_p v_p) \quad (3.46)$$

which can be used in the conservation equation for the mixture density, Eqn. (3.42) as follows

$$\frac{\partial}{\partial t}(\rho_m) + \frac{\partial}{\partial z}(\rho_m v_m) = (\rho_q - \rho_p)\frac{\partial}{\partial z}(\alpha_p v_p) + \frac{\partial}{\partial z}(\rho_m v_m) = 0$$

or

$$\frac{\partial}{\partial z}[(\rho_q - \rho_p)(\alpha_p v_p) + (\rho_m v_m)] = 0 \quad (3.47)$$

In addition, being $v_p = v_m + (1 - c_p)v_{pq}$, then the last expression becomes which is shown in Eqn. (3.48)

$$\frac{\partial}{\partial z}\{[(\rho_q - \rho_p)(\alpha_p v_p) + (\rho_m v_m)] + [\alpha_p \rho_p + (1 - \alpha_p)\rho_q]v_m\} = 0 \quad (3.48)$$

This expression indicates that the argument of the derivative is constant in space. If $v_m = 0$ and $\alpha_p = 0$ are assumed for $z = 0$, (no slip wall and perfect settling at the bottom) this constant is zero, then it becomes

$$[(\rho_q - \rho_p)(\alpha_p v_p) + (\rho_m v_m)] + [\alpha_p \rho_p + (1 - \alpha_p)\rho_q]v_m = 0$$

next, with some extra algebra, the final expression is given by Eqn. (3.49)

$$v_m = \alpha_p \left(\frac{\rho_p}{\rho_m} - 1 \right) v_{pq} \quad (3.49)$$

Now the value of v_m is linked algebraically to α_p therefore, the solution of α_p implies the solution of v_m . To this end, it is necessary to select a law for v_{pq} .

Determination of the secondary phase void fraction

Recalling the third equation in Eqn. (3.40) it reads [Eqn. (3.50)]:

$$\frac{\partial}{\partial t}(\alpha_p) + \frac{\partial}{\partial z} \{ \alpha_p [v_m + (1 - c_p) v_{pq}] \} = 0 \quad (3.50)$$

expanding c_p and v_{pq} using Eqns. (3.13) and (3.15) it results in Eqn. (3.51)

$$\frac{\partial}{\partial t}(\alpha_p) + \frac{\partial}{\partial z} \left\{ \alpha_p \left[v_m + \left(1 - \frac{\alpha_p \rho_p}{\rho_m} \right) v_{rc} (1 - \alpha_p)^a \right] \right\} = 0 \quad (3.51)$$

Here, using the expression obtained for v_m [Eqn. (3.49)] it is possible to arrive to the final equation for α_p [Eqn. (3.52)]

$$\frac{\partial}{\partial t}(\alpha_p) + \frac{\partial}{\partial z} [v_{rc} (\alpha_p - \alpha_p^2) (1 - \alpha_p)^a] = 0 \quad (3.52)$$

Once again it required the selection of a law for v_{pq} based on the physics of the problem; in this case the model law for the relative velocity was used, $v_{pq} = v_{rc} (1 - \alpha_p)^a$. The final expression of the α_p equation is a non-linear hyperbolic equation. The solution of this equation strongly depends on the definition of the flux; in this case, it reads [Eqn. (3.53)]

$$F(\alpha_p) = v_{rc} (\alpha_p - \alpha_p^2) (1 - \alpha_p)^a \quad (3.53)$$

The existence of different kinds of waves in the solution leads to deal with one or more *Riemann problems* and requires appropriate methods for solving them (LeVeque, 2002; Toro, 2009). As is shown, the flux does not depends

explicitly on the physical parameters. The physics is included in the definitions of v_{rc} and the exponent a .

Determination of the pressure

Finally, the pressure of the problem is obtained integrating the momentum equation [second equation in Eqn. (3.40)], that after reordering reads as in Eqn. (3.54)

$$\frac{\partial}{\partial z} p = -\frac{\partial}{\partial t}(\rho_m v_m) - \frac{\partial}{\partial z}(\rho_m v_m v_m) - \frac{\partial}{\partial z} [\rho_m c_p (1 - c_p) v_{pq}^2] + \rho_m g \quad (3.54)$$

Determination of the front velocities

In addition to the semi-analytical solution obtained, two other valuable results can be obtained by the application of the Rankine-Hugoniot conditions (jump conditions) (LeVeque, 2002) in the system given by Eqn. (3.40). Therefore, applying the jump condition at each front (see Figure 3.3) it gives which is shown in Eqns. (3.55)-(3.56)

$$\begin{cases} (\rho_m^0 - \rho_p) a' = \rho_m^0 v_m - \rho_p v_m^+ \\ (-\rho_p v_m^+ + \rho_m v_m) a' = \rho_m^0 v_m^2 + [p]_a - \rho_m c_p (1 - c_p) v_{pq}^2 - \rho_p (v_m^+)^2 \\ (\alpha_p^0 - 1) a' = \alpha_p [v_m + (1 - c_p) v_{pq}^0] - v_m^+ \end{cases} \quad (3.55)$$

where a' is the velocity of the top front, and

$$\begin{cases} (\rho_q - \rho_m^0) b' = -\rho_m^0 v_m \\ -\rho_m^0 v_m b' = -\rho_m^0 v_m^2 + [p]_b + \rho_m c_p (1 - c_p) v_{pq}^2 \\ b' = v_m + (1 - c_p) v_{pq}^0 \end{cases} \quad (3.56)$$

where b' is the velocity of the bottom front. Next, working with the first and

third equations of (3.56) it is possible to isolate the velocity of the bottom front, which is shown in Eqn. (3.57)

$$b' = (1 - \alpha_p^0) v_{pq}^0 \quad (3.57)$$

In the same way, working with the first and third equation of Eqn. (3.55) it is possible to isolate the value of a' , which is given in Eqn. (3.58).

$$a' = -\alpha_p^0 v_{pq}^0 \quad (3.58)$$

Finally, the complete solution procedure for the 1D sedimentation case is sketched in Algorithm 3. This solution is complemented by the front velocities given by Eqns. (3.57)-(3.58).

Algorithm 3 Steps for semi-analytical solution

1. Solve the hyperbolic partial differential equation for α_p :

$$\frac{\partial}{\partial t}(\alpha_p) + \frac{\partial}{\partial z} [v_{rc} (\alpha_p - \alpha_p^2) (1 - \alpha_p)^a] = 0$$

2. Find the mixture velocity, v_m using α_p and the physical parameters

$$v_m = \alpha_p \left(\frac{\rho_p}{\rho_m} - 1 \right) v_{pq}$$

3. Finally, obtain the pressure integrating the momentum balance

$$\frac{\partial}{\partial z} p = -\frac{\partial}{\partial t}(\rho_m v_m) - \frac{\partial}{\partial z}(\rho_m v_m v_m) - \frac{\partial}{\partial z} [\rho_m c_p (1 - c_p) v_{pq}^2] + \rho_m \vec{g}$$

Velocity solution in terms of center-of-volume velocity

The expression found for the center of mass velocity can be translated into center of volume terms. Starting from Eqn. (3.49) and writing it in terms of the primary phase void fraction, α_q , and the primary to secondary phase relative velocity, \vec{v}_{qp} it reads as in Eqn. (3.59)

$$v_m = \alpha_p \left(\frac{\rho_p}{\rho_m} - 1 \right) v_{pq} = - (1 - \alpha_q) \left(\frac{\rho_p}{\rho_m} - 1 \right) v_{qp} \quad (3.59)$$

Now, using the relationship between v_m and u given by Eqn. (3.31), Eqn. (3.60) is obtained

$$v_m = - (1 - \alpha_q) \left(\frac{\rho_p}{\rho_m} - 1 \right) v_{qp} = u + \alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} v_{qp} \quad (3.60)$$

Finally, isolating u it results to have the value $u = 0$. Therefore, the primary phase conservation equation [Eqn. (3.39)] now reads as in Eqn. (3.61)

$$\frac{\partial \alpha_q}{\partial t} + \vec{\nabla} \cdot [\alpha_q (1 - \alpha_q) \vec{v}_{qp}] = 0 \quad (3.61)$$

This formulation resembles the theory of sedimentation of Kynch (Kynch, 1952; Bürger and Wendland, 2001) which is based on the solution of the Drift-Flux Model in a quiescent pool with either bubbles or sediments and using proper flux functions (Wallis, 1969; Nigam, 2003; Pilon and Viskanta, 2004).

3.3.2 One dimensional analytic solutions for the equation for α_p

As was shown in Algorithm 3, the first step to find semi-analytic solutions for 1D sedimentation is the solution of the mass conservation equation for the secondary phase [Eqn. (3.52)]. Then, recalling this equation

$$\frac{\partial}{\partial t} (\alpha_p) + \frac{\partial}{\partial z} [v_{rc} (\alpha_p - \alpha_p^2) (1 - \alpha_p)^a] = 0$$

it is necessary to revise some concepts on hyperbolic systems. As was stated in Section 3.2.2, the kind of solution of these equations relies on the type of flux, convex or non-convex, and the initial and boundary conditions. In the case of $a = 0$, i.e. a constant relative velocity is used in the flux function, the flux becomes which is shown in Eqn. (3.62)

$$F = v_{rc} (\alpha_p - \alpha_p^2) = v_{rc} (1 - \alpha_p) \alpha_p \quad (3.62)$$

which is a convex flux function (see Figure 3.4, continue line). If $a = 1$ the flux is non-convex with a flux function given by Eqn. (3.63)

$$F = v_{rc} (\alpha_p - \alpha_p^2) (1 - \alpha_p) \quad (3.63)$$

which is shown in Figure 3.4 with a dashed line.

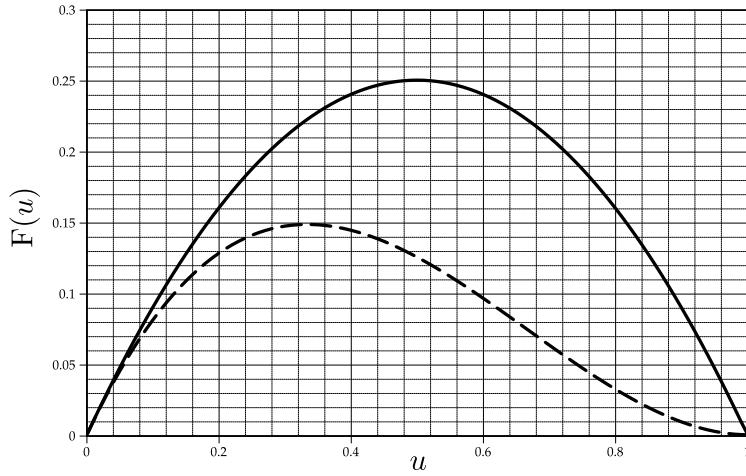


Figure 3.4: Shape of fluxes for α_p equation. — convex flux, - - - non convex flux.

The Riemann problem for α_p equation

Many of the numerical methods for the solution of hyperbolic equations are based on the Riemann problem given at the interface of two cells. In general the Riemann problem for a model equation with convex flux reads as in Eqn. (3.64)

$$\left. \begin{array}{l} \text{PDE : } \frac{\partial}{\partial t}(u) + \frac{\partial}{\partial z}v_0 (1-u) u = 0 \\ \text{IC : } u(z,0) = \begin{cases} u_L, & z < 0 \\ u_R, & z > 0 \end{cases} \end{array} \right\} \quad (3.64)$$

where PDE indicates the partial differential equation which is being solved and IC the particular initial conditions set for the case. In the solution of this

problem, since its has only one curvature, the discontinuities present are *single waves* in the form of shock waves, rarefaction waves and transonic rarefaction waves. In the case of non-convex fluxes *compound waves* can be present like the shock-rarefaction waves. The admissibility condition for shock waves in the general case of a non-convex flux is given by an extension of the Lax Entropy Condition given by Oleinik (LeVeque, 2002) and recalled in the definition 3.3.1.

Entropy condition (Oleinik) 3.3.1. A weak solution $u(z, t)$, is the vanishing-viscosity solution to a general scalar conservation law if all discontinuities have the property that

$$\frac{F(u) - F(u_L)}{u - u_L} \geq s \geq \frac{F(u) - F(u_R)}{u - u_R} \quad (3.65)$$

for all u between u_L and u_R .

Therefore, the solution of the problem depends on the values of the initial states u_L and u_R presenting the following cases:

1. If $u_L < u_R$, Figure 3.5.a) left, the entropy condition is fulfilled and the the solution is a shock wave traveling to the right, see Figure 3.5.1. This proposition can be checked graphically since the slope of the flux for the left point is always bigger than s and the slope of the flux for the right point. Then, the solution of the problem is: $u(0, t) = u_L$.
2. If $u_L > u_R$, Figure 3.5.a) right, the solution is also a shock, but traveling to the left, see Figure 3.5.2. Then, the solution of the problem is: $u(0, t) = u_R$.
3. If $u_R < u_L < u_s$, Figure 3.5.b) left, the entropy condition is not fulfilled so a rarefaction wave is present. Since the speed of the right characteristic is greater than the left characteristic, the rarefaction wave travels to the right as in Figure 3.5.3. The solution of the problem is: $u(0, t) = u_L$.
4. If $u_s < u_R < u_L$, Figure 3.5.b) right, the entropy condition is not fulfilled so a rarefaction wave is present. Since the speed of the left characteristic is greater than the right characteristic the rarefaction wave travels to the left as in Figure 3.5.4. The solution of the problem is: $u(0, t) = u_R$.

5. If $u_R < u_s < u_L$, Figure 3.5.c), the entropy condition is not fulfilled so a rarefaction wave is present in the form of a fan. Since the fan crosses the stagnation point u_s the solution of the problem is: $u(0, t) = u_s$, as is shown in 3.5.5.

The analysis for the opposite curvature is similar. In the case of non-convex fluxes the waves present are a combination of the single waves shown. Now it is possible to obtain the analytical solutions for different Riemann problems involving the α_p equation.

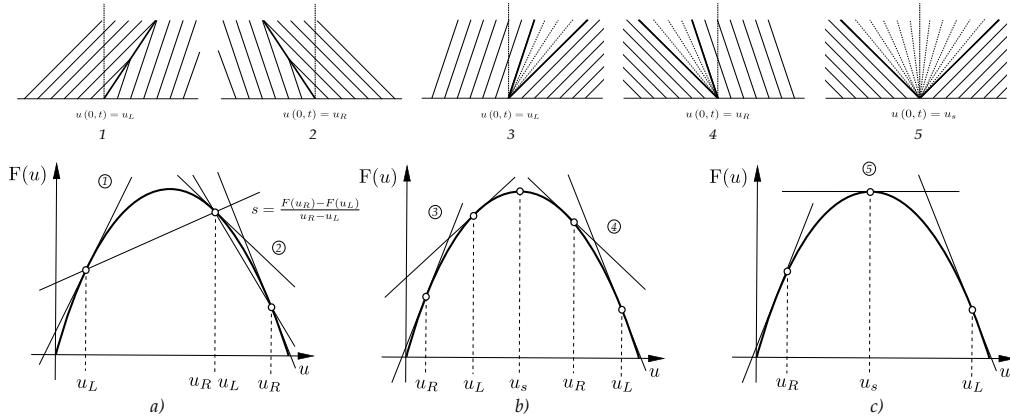


Figure 3.5: Riemann problem solutions for a convex flux. 1, right going shock; 2, left going shock; 3, right going rarefaction; 4, left going rarefaction; 5, transonic rarefaction.

Convex flux

In the case of convex flux the flux function is the same of the model problem (compare Figure 3.4 with 3.5), so the solutions follow directly the rules previously presented. The equation to be solved reads [Eqn. (3.66)]:

$$\frac{\partial}{\partial t}(\alpha_p) + \frac{\partial}{\partial z}\{\alpha_p [v_{rc} (1 - \alpha_p)]\} = 0 \quad (3.66)$$

where $F(\alpha_p) = \alpha_p [v_{rc} (1 - \alpha_p)]$.

In order to explore some of the presented cases, two problems will be solved:

- a) IC : $\alpha_p(z, 0) = \begin{cases} \alpha_{pL} = 0.6, & z < 0.5 \\ \alpha_{pR} = 0.7, & z > 0.5 \end{cases}$ with zero flux at boundaries. Since the walls are impermeable the total quantity of α_p must remain constant. Due to this and that the advective velocity $v_{rc}(1 - \alpha_p)$ is always positive, α_p is moved from left to right. Then α_p reaches zero at the left and one at the right forming two new Riemann problems being three in total. From left to right they are, $\alpha_{pL} = 0$ $\alpha_{pR} = 0.6$, $\alpha_{pL} = 0.6$ $\alpha_{pR} = 0.7$ and $\alpha_{pL} = 0.7$ $\alpha_{pR} = 1$. Using the information given in the beginning of the section, the expected waves are a right going shock, a left going rarefaction and a left going shock, which can be observed in Figure 3.6.a).
- b) IC : $\alpha_p(z, 0) = \begin{cases} \alpha_{pL} = 0.7, & z < 0.5 \\ \alpha_{pR} = 0.5, & z > 0.5 \end{cases}$ with zero flux at boundaries. In this case there are four Riemann problems in total. From left to right they are, $\alpha_{pL} = 0$ $\alpha_{pR} = 0.7$, $\alpha_{pL} = 0.7$ $\alpha_{pR} = 0.5$, $\alpha_{pL} = 0.5$ $\alpha_{pR} = 0.5$ and $\alpha_{pL} = 0.5$ $\alpha_{pR} = 1$. Using the information given in the beginning of the section the expected waves are a right going shock, a left going rarefaction and a stationary zone due to $\alpha_p = 0.5$ matches the sonic point and a left going shock, which can be observed in Figure 3.6.b).

Non convex flux

As was stated, when a non convex flux is present compound waves can be formed. To this end it is necessary to analyze the flux more deeply taking information from its first derivative. As is shown in Figure 3.7 the first derivative of the flux has a minimum, therefore the convexity changes from a concave flux to a convex flux. In this case taking the Riemann problem with initial conditions given by:

$$\text{IC} : \alpha_p(z, 0) = 0.3$$

and the flux function resulting from a linear law for the relative velocity [see Eqn. (3.63)] which recalled as follows

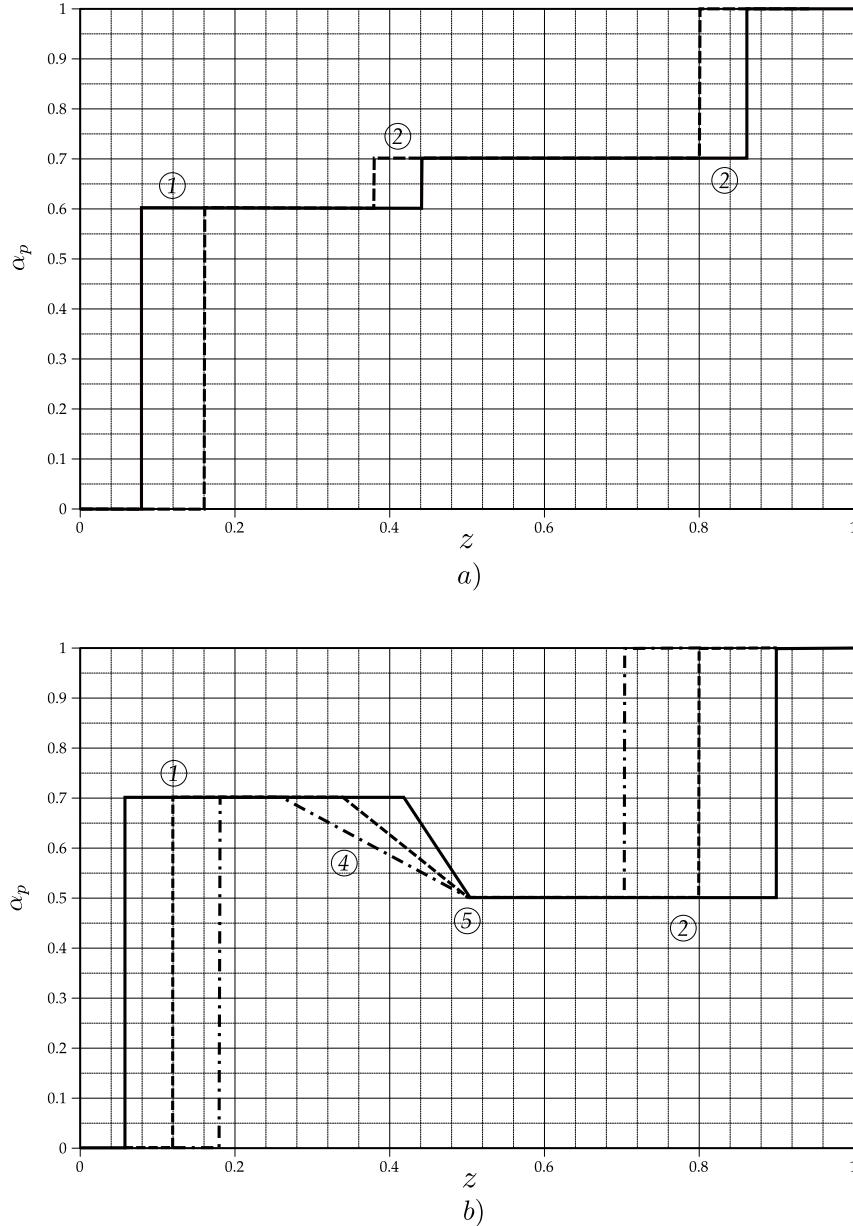


Figure 3.6: Riemann problem solutions for a convex flux for given states, a) $u_L = 0.6, u_R = 0.7$, b) $u_L = 0.7, u_R = 0.5$ along the time, — 0.2, - - - 0.4, — · - 0.6, presenting different kind of waves, 1) right going shock, 2) left going shock, 4) left going rarefaction and 5) stationary point due to sonic point crossing. Right going rarefaction is not presented.

$$F = v_{rc} (\alpha_p - \alpha_p^2) (1 - \alpha_p)^a$$

which is similar to the test proposed by Nigam (Nigam, 2003). Again, due to the convective velocity is always positive, α_p is transported from left to right giving place to two Riemann problems $\alpha_p(z, t) = \begin{cases} \alpha_{pL} = 0, & z < s_1 t \\ \alpha_{pR} = 0.3, & z > s_1 t \end{cases}$ and $\alpha_p(z, t) = \begin{cases} \alpha_{pL} = 0.3, & z < 1 - s_2 t \\ \alpha_{pR} = \alpha_{p\text{FAN}}, & z > 1 - s_2 t \end{cases}$. In Figure 3.7, the evolution goes through $\alpha_p = 0$, $\alpha_p = 0.3$ and $\alpha_p = 1$ crossing the tangency point $\alpha_p \cong 0.87$. The first change (1) from $\alpha_p = 0$ to $\alpha_p = 0.3$ evolves as a right going shock, the second change (6) has to be divided in two jumps, the first from $\alpha_p = 0.3$ to $\alpha_p \cong 0.87$ and the second from $\alpha_p \cong 0.87$ to $\alpha_p = 1$. The paths drawn by straight lines corresponds to shocks and the paths that follow the flux graph are rarefactions. Therefore, the second change is a shock and then a rarefaction, or naming it in the direction of wave's velocity a rarefaction-shock. The shaded zone is the so-called *upper convex hull*. The analytical solution of this problem is shown Figure 3.8 for two different times. There it is clearly seen the structure of the waves, note that the fan at right evolves from $\alpha_p \cong 0.87$ to $\alpha_p = 1$.

Then, having the solution for α_p , and applying the remain steps of Algorithm 3 it is possible to find v_m and p as well. For example, in the case of the non-convex flux they are presented in Figures 3.9-3.10.

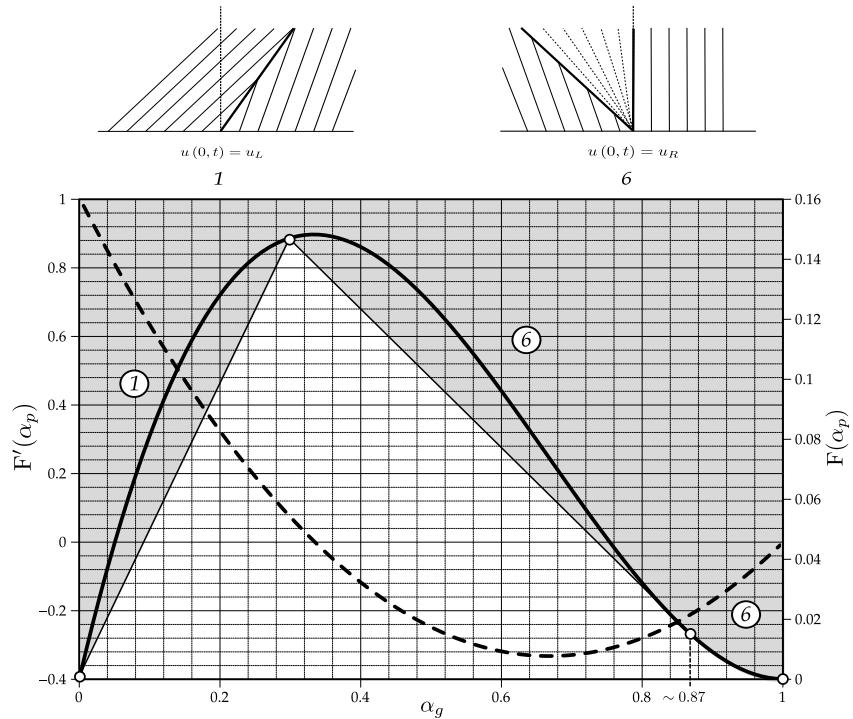


Figure 3.7: Riemann problem solutions for a non convex flux. 1, Right going shock; 6, left going rarefaction-shock. — flux, - - - flux derivative.

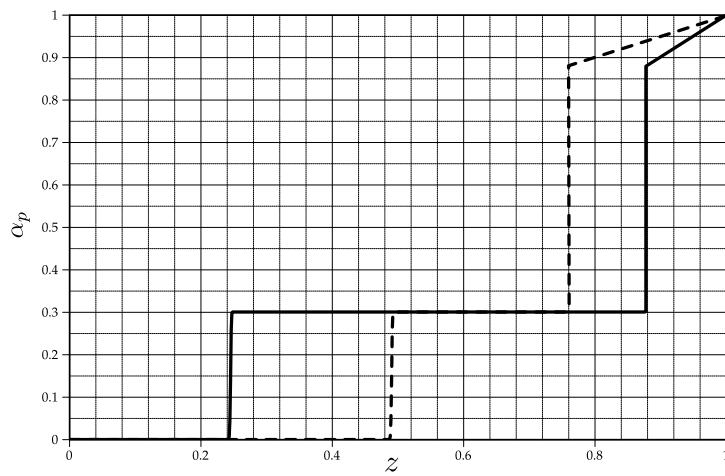


Figure 3.8: Riemann problem solutions for a non convex flux with initial condition $\alpha_p = 0.3$ at time $t = 0.5$ and $t = 1$.

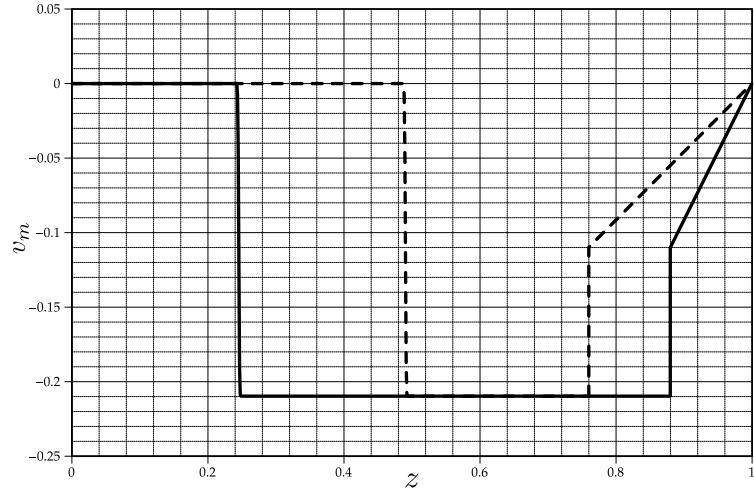


Figure 3.9: Mean velocity, v_m , profiles corresponding α_p distributions in previous figure at time — 0.5 and - - - 1.

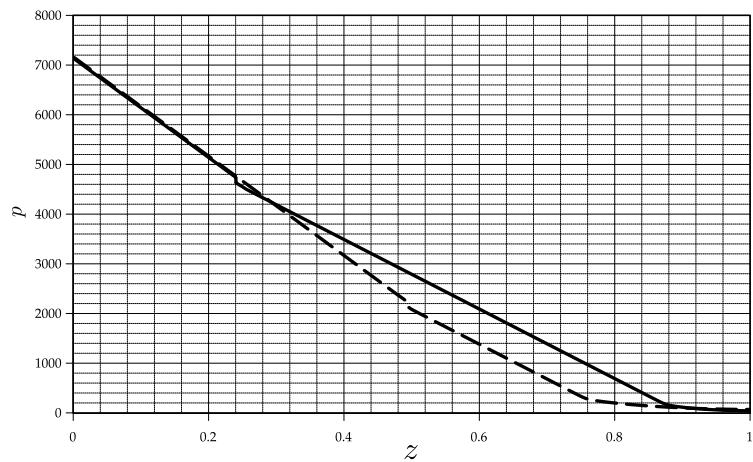


Figure 3.10: Pressure, p , profiles corresponding α_p and v_m distributions in previous figures at time — 0.5 and - - - 1.

Chapter 4

Numerical solutions for the phase fraction equation

4.1 Introduction

As was stated in Section 3.2.2, the solution of the dynamics of the dispersed phase has a principal role in the solution of the multiphase systems. Due to the hyperbolicity of the secondary phase fraction equation, the numerical method selected for its solution has to be capable to capture shocks, rarefactions and compound waves. To this end two techniques have been selected. First, a Riemann-free solver and second a TVD scheme, both with Flux Corrected Transport limiting to avoid spurious unboundedness.

Thus, the purpose of this chapter is to develop a new Riemann-free solver derived from the Kurganov and Tadmor (KT) (Kurganov and Tadmor, 2000) scheme capable to use the face flux given at faces and obtained in the pressure-velocity coupling loop (PISO). This face flux is a part of the total flux of the phase fraction transport equation. In this way the KT scheme is adapted to a cell-centered FVM discretization framework based on flux conservation. The integration of the transport equation is done by means of a modified FCT technique where no local extrema is taken into account. The same technique is used for the limiting of the fluxes obtained by the TVD schemes. Both methods

have multidimensional extension which is explained as well.

4.2 A Riemann-free solver with centered flux

4.2.1 First order Kurganov and Tadmor scheme

As an introduction for the presentation of the new central scheme, the derivation of the fully discrete first order Kurganov and Tadmor (KT) scheme will be recalled. This scheme relies on a stabilization for the advective terms based on the *local* speeds of wave propagation, which for the most practical applications can be defined as in Eqn. (4.1).

$$a_{j+1/2}^n := \max \left\{ \rho \left(\frac{\partial F}{\partial u} (u_{j+1/2,l}) \right), \rho \left(\frac{\partial F}{\partial u} (u_{j+1/2,r}) \right) \right\} \quad (4.1)$$

where ρ is the spectral radius of the $\frac{\partial F}{\partial u}$ matrix at each side of a face. The first step is to integrate over intervals placed around the cell interfaces; these integration intervals are delimited by long-dashed lines in Figure 4.1. The obtained value depends on the side faces' values and the weighting factors are given by the local speeds. If the local speeds indicate that the information comes, for example, from the upwind cell then the integrating interval will be biased in such direction; if the information comes from upwind and downwind direction the interval will be extend along both sides of the interface. The integration procedure gives the values of the w^{n+1} function from the original solution values u^n . Here it is important to note that the w^{n+1} values are placed at the interfaces and not in the original cell centers leading to a staggered mesh; this idea is exploited by the Nessyahu and Tadmor (Nessyahu and Tadmor, 1990) scheme. This staggered solution is not convenient since requires working with two meshes and leads to a more dissipative solution (Kurganov and Tadmor, 2000), therefore, a new integration is done through a \tilde{w}^{n+1} function, defined over the j -th cell and its neighbors. The boundaries of the cells of this intermediante mesh are defined by the influence zone of the local speeds, as is shown with dash-dotted line in Figure 4.1. As a result of this last integration the time updated values of u^n are found as u^{n+1} . The

figure also includes a centered volumetric flux, v_c , given at the interfaces; the meaning of this flux will be explained later.

Thus, the first step for the construction of the KT scheme is to integrate over the intervals $[x_{j+1/2,l}^n, x_{j+1/2,r}^n] \times [t^n, t^{n+1}]$ (See Figure 4.1), where the extrema of the spatial interval are defined as: $x_{j+1/2,l}^n := x_{j+1/2} - a_{j+1/2}^n \Delta t$ and $x_{j+1/2,r}^n := x_{j+1/2} + a_{j+1/2}^n \Delta t$. Then, if $\Delta x_{j+1/2} := x_{j+1/2,r}^n - x_{j+1/2,l}^n$ denotes the width of the Riemann fan originated at $x_{j+1/2}$; the average of u within this interval is given by Eqn. (4.2):

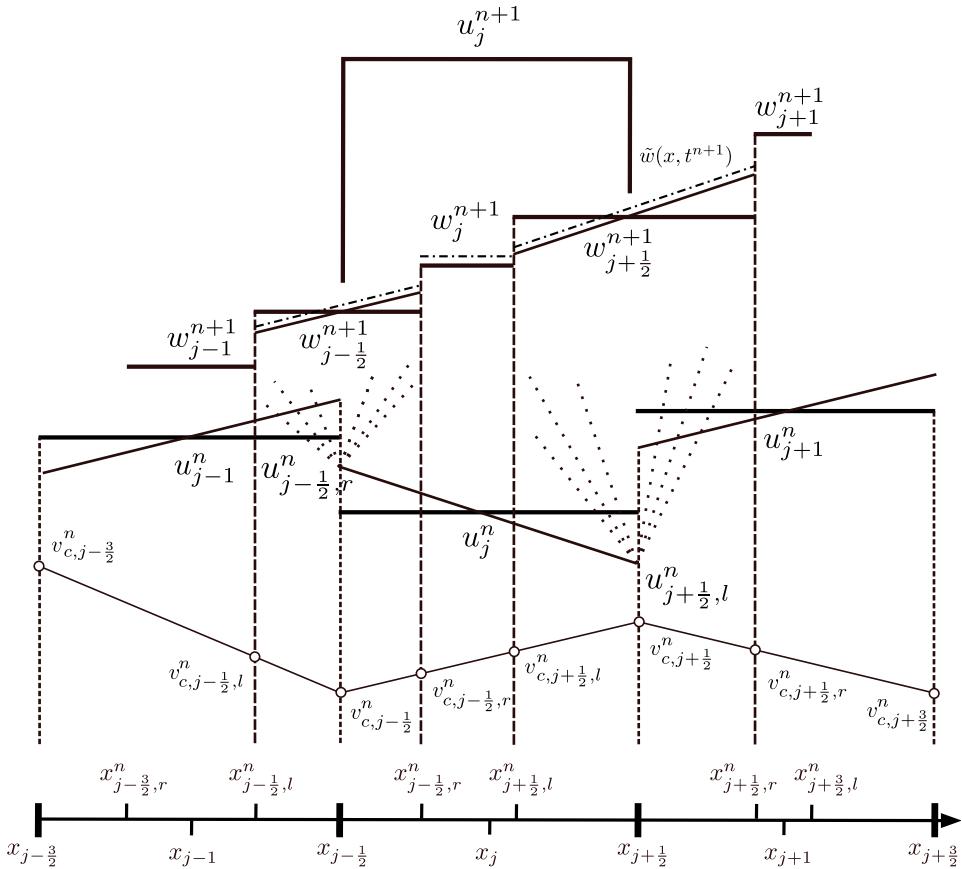


Figure 4.1: Central differencing scheme using given centered flux at faces
[Adapted from (Kurganov and Tadmor, 2000)]

$$\begin{aligned}
w_{j+1/2}^{n+1} &= \frac{1}{\Delta x_{j+1/2}} \int_{x_{j+1/2,l}^n}^{x_{j+1/2,r}^n} u(\xi, t^{n+1}) d\xi \\
&= \frac{u_j^n + u_{j+1}^n}{2} + \frac{\Delta x - a_{j+1/2}^n \Delta t}{4} [(u_x)_j^n - (u_x)_{j+1}^n] \\
&\quad - \frac{1}{2a_{j+1/2}^n \Delta t} \int_{t^n}^{t^{n+1}} \{F[u(x_{j+1/2,r}^n, \tau)] - F[u(x_{j+1/2,l}^n, \tau)]\} d\tau
\end{aligned} \tag{4.2}$$

where ξ is the spatial coordinate within the spatial integration interval, τ is the temporal variable within the temporal integration interval and F is the flux through the boundaries of the integration interval. Similarly taking $\Delta x_j := x_{j+1/2,l}^n - x_{j-1/2,r}^n = \Delta x - \Delta t(a_{j-1/2}^n + a_{j+1/2}^n)$ which is the width of the strip around x_j where there is no influence of the Riemann fans from neighbouring points, the average is given by Eqn. (4.3)

$$\begin{aligned}
w_j^{n+1} &= \frac{1}{\Delta x_j} \int_{x_{j-1/2,r}^n}^{x_{j+1/2,l}^n} u(\xi, t^{n+1}) d\xi \\
&= u_j^n + \frac{\Delta t}{4} (a_{j-1/2}^n - a_{j+1/2}^n) (u_x)_j^n \\
&\quad - \frac{1}{2a_{\Delta x_j}} \int_{t^n}^{t^{n+1}} [F(u(x_{j+1/2,l}^n, \tau)) - F(u(x_{j-1/2,r}^n, \tau))] d\tau
\end{aligned} \tag{4.3}$$

Finally, using the midpoint rule it is possible to approximate the integrals over the flux in Eqns. (4.2-4.3) giving the desired cell averages at $t = t^{n+1}$ in Eqn. (4.4)

$$\begin{aligned}
w_{j+1/2}^{n+1} &= \frac{u_j^n + u_{j+1}^n}{2} + \frac{\Delta x - a_{j+1/2}^n \Delta t}{4} [(u_x)_j^n - (u_x)_{j+1}^n] \\
&\quad - \frac{1}{2a_{j+1/2}^n} [\mathbf{F}(u_{j+1/2,r}^{n+1/2}) - \mathbf{F}(u_{j+1/2,l}^{n+1/2})] \\
w_j^{n+1} &= u_j^n + \frac{\Delta t}{2} \left(a_{j-1/2}^n - a_{j+1/2}^n \right) (u_x)_j^n \\
&\quad - \frac{\lambda}{1 - \lambda(a_{j-1/2}^n + a_{j+1/2}^n)} [\mathbf{F}(u_{j+1/2,l}^{n+1/2}) - \mathbf{F}(u_{j-1/2,r}^{n+1/2})]
\end{aligned} \tag{4.4}$$

where $\lambda = \Delta t / \Delta x$. Now, it is necessary to evaluate $u_{j+1/2,l}^{n+1/2}$ and $u_{j+1/2,r}^{n+1/2}$. This goal is achieved by a Taylor expansion giving the results presented in Eqn. (4.5)

$$\begin{aligned}
u_{j+1/2,l}^{n+1/2} &:= u_{j+1/2,l}^n - \frac{\Delta t}{2} \mathbf{F}(u_{j+1/2,l}^n)_x \\
u_{j+1/2,r}^{n+1/2} &:= u_{j+1/2,r}^n - \frac{\Delta t}{2} \mathbf{F}(u_{j+1/2,r}^n)_x
\end{aligned} \tag{4.5}$$

where $u_{j+1/2,l}^n = u_j^n + \Delta x (u_x)_j^n \left(1/2 - \lambda a_{j+1/2}^n \right)$ and $u_{j+1/2,r}^n = u_{j+1}^n - \Delta x (u_x)_{j+1}^n \left(1/2 - \lambda a_{j+1/2}^n \right)$. At this point, the function u has evolved in time to the approximate cell averages $w_{j+1/2}^{n+1}$, w_j^{n+1} which are located in a new staggered mesh. In order to obtain a solution over the original mesh the solution is newly averaged using a piecewise-linear reconstruction $\tilde{w}(x, t^{n+1})$ based on the w 's (See Figure 4.1, dash-dotted line). This reconstruction is defined by Eqn. (4.6)

$$\begin{aligned}
\tilde{w}(x, t^{n+1}) &:= \left[w_{j-1/2}^{n+1} + (u_x)_{j-1/2}^{n+1} (x - x_{j-1/2}) \right] \mathbf{1}_{[x_{j-1/2,l}^n, x_{j-1/2,r}^n]} + \\
&\quad w_j^{n+1} \mathbf{1}_{[x_{j-1/2,r}^n, x_{j+1/2,l}^n]} + \\
&\quad \left[w_{j+1/2}^{n+1} + (u_x)_{j+1/2}^{n+1} (x - x_{j+1/2}) \right] \mathbf{1}_{[x_{j+1/2,l}^n, x_{j+1/2,r}^n]}
\end{aligned} \tag{4.6}$$

where $\mathbf{1}_{[x_1, x_2]}$ is the indicator function which has a value of 1 within the interval given in the sub-index and zero otherwise and the required $(u_x)_{j-1/2}^{n+1}$, $(u_x)_{j+1/2}^{n+1}$

slopes are given by a TVD limiter. The original KT scheme uses MinMod as the limiter. Therefore, doing the average as shown in Eqn. (4.7)

$$\begin{aligned}
 u_j^{n+1} = & \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \tilde{w}(\xi, t^{n+1}) d\xi = \frac{1}{\Delta x} \left\{ \Delta x_j [w_j^{n+1}] + \Delta x_{j+1/2} \left[w_{j+1/2}^{n+1} + \right. \right. \\
 & \left. \left. (u_x)_{j+1/2}^{n+1} (x - x_{j+1/2}) \right] + \Delta x_{j-1/2} \left[w_{j-1/2}^{n+1} + (u_x)_{j-1/2}^{n+1} (x - x_{j-1/2}) \right] \right\}
 \end{aligned} \tag{4.7}$$

and re-arranging it finally gives Eqn. (4.8)

$$\begin{aligned}
 u_j^{n+1} = & \lambda a_{j-1/2}^n w_{j-1/2}^{n+1} + \left[1 - \lambda (a_{j-1/2}^n + a_{j+1/2}^n) \right] w_j^{n+1} + \lambda a_{j+1/2}^n w_{j+1/2}^{n+1} + \\
 & \frac{\Delta x}{2} \left[(\lambda a_{j-1/2}^n)^2 (u_x)_{j-1/2}^{n+1} - (\lambda a_{j+1/2}^n)^2 (u_x)_{j+1/2}^{n+1} \right]
 \end{aligned} \tag{4.8}$$

This final expression is known as the *Kurganov and Tadmor fully discrete second order* scheme. The piecewise reconstruction used ensures the obtained second-order accuracy. If constant values are used by cell it reduces the order of the scheme to first-order accuracy. This kind of scheme is obtained setting $(u_x)_j^n$, $(u_x)_{j-1/2}^{n+1}$ and $(u_x)_{j+1/2}^{n+1}$ as zero in Eqn. (4.7) and its form is given in Eqn. (4.9)

$$\begin{aligned}
 u_j^{n+1} = & u_j^n - \frac{\lambda}{2} [\mathbf{F}(u_{j+1}^n) - \mathbf{F}(u_{j-1}^n)] + \\
 & \frac{1}{2} \left[\lambda a_{j+1/2}^n (u_{j+1}^n - u_j^n) - a_{j-1/2}^n (u_j^n - u_{j-1}^n) \right]
 \end{aligned} \tag{4.9}$$

This scheme was originally attributed to Rusanov. Even though it is a first order accurate it has a simple form and is useful for implementation checks in high order schemes.

4.2.2 KT as a MUSCL based scheme

In order to give a multidimensional extension of the KT derived schemes is worthy to write these kind of schemes as a Monotone Upstream-centered Scheme for Conservation Laws (MUSCL) based scheme (van Leer, 1979)¹. The MUSCL schemes are based in the idea of flux balances, where the fluxes at faces are calculated on the base of reconstructed values. Therefore, it is then necessary to find the expression of these fluxes. Starting with a general hyperbolic equation as in Eqn. (4.10)

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0 \quad (4.10)$$

the semi-discrete form can be written as in (4.11)

$$\frac{\partial u_i}{\partial t} = -\frac{1}{\Delta x} [F(u_{i+1/2}^*) - F(u_{i-1/2}^*)] \quad (4.11)$$

where $F(u_{i\pm 1/2}^*)$ are the numerical inter-cell fluxes. These fluxes are calculated using u^* face values reconstructed from cell-centered values. Such a reconstruction is a non-linear combination of first and second reconstruction schemes given by a TVD function. Recalling the semi-discrete form of the KT scheme it reads (see Eqn. 4.12)

$$\begin{aligned} \frac{\partial u_j}{\partial t} &= -\frac{1}{2\Delta x} \left\{ F \left[u_{j+1/2}^+(t) \right] + F \left[u_{j+1/2}^-(t) \right] \right\} - \left\{ F \left[u_{j-1/2}^+(t) \right] + F \left[u_{j-1/2}^-(t) \right] \right\} \\ &+ \frac{1}{2\Delta x} \left\{ a_{j+1/2}(t) [u_{j+1/2}^+(t) - u_{j+1/2}^-(t)] - a_{j-1/2}(t) [u_{j-1/2}^+(t) - u_{j-1/2}^-(t)] \right\} \end{aligned} \quad (4.12)$$

where u^\pm represent the values of u at each side of a cell face. This expression reduces to Eqn. (4.9) if constant values by cell are used. Comparing Eqn. (4.11) and Eqn. (4.12) it follows that the numerical fluxes $F(u_{i\pm 1/2}^*)$ are given by Eqn. (4.13)

¹Some parts of the presentation follow the notation and concepts given in http://en.wikipedia.org/wiki/MUSCL_scheme (Visited June 12th, 2012)

$$F(u_{i\pm 1/2}^*) = \frac{F(u_{j\pm 1/2}^+(t)) + F(u_{j\pm 1/2}^-(t))}{2} - \frac{a_{j\pm 1/2}(t)}{2} [u_{j\pm 1/2}^+(t) - u_{j\pm 1/2}^-(t)] \quad (4.13)$$

where the reconstructed values at faces $u_{j+1/2}^\pm$ are given by Eqn. (4.14)

$$u_{j+1/2}^+ := u_{j+1}(t) - \frac{\Delta x}{2}(u_x)_{j+1}(t), \quad u_{j+1/2}^- := u_j(t) + \frac{\Delta x}{2}(u_x)_j(t) \quad (4.14)$$

Recalling the theory given in Section 2.5 and setting the owner cell $P = j$, the neighbour cell $N = j + 1$ and second neighbour cell as $j+2$, Eqn. (4.14) can be written in TVD form as in Eqn. (4.15)

$$\begin{aligned} u_{j+1/2}^+ &:= u_{j+1} - \frac{1}{2}\psi(r_{j+1}, t)(u_{j+2}(t) - u_{j+1}(t)), \\ u_{j+1/2}^- &:= u_j + \frac{1}{2}\psi(r_j, t)(u_{j+1}(t) - u_j(t)) \end{aligned} \quad (4.15)$$

where $\psi(r_j, t)$ is the Sweby function for a given limiter. Again, the order of this MUSCL based scheme is given by the kind of reconstruction used to find the face values.

4.2.3 A Riemann-free solver with given centered flux at faces

Having in mind the basic concepts about the KT scheme and its derivation it is possible to propose a scheme for a slightly different flux. This new kind of flux is presented in Eqn. (4.16)

$$F(u(x, t)) = F(u, t)^R + v_c(u(x, t)) u(x, t) \quad (4.16)$$

The part of the flux given by $F(u, t)^R$ is calculated as usual, it is, using the flux function given in the hyperbolic equation. The other one is based on a volumetric flux precomputed at faces, v_c , and added as $v_c(u(x, t)) u(x, t)$. The necessity of this kind of fluxes is a key concept in the construction of conservative incompressible solvers. In these solvers part of the flux is usually

given at faces by the pressure-velocity coupling loop and not calculated explicitly by the hyperbolic flux function. The flux at faces could be also reconstructed from the cell-centered velocity, but this method is not conservative. Then, in order to derive a scheme with this kind of flux included it is necessary to find the averages over the staggered mesh as is done in Eqn. (4.17) in the same way than the original method.

$$\begin{aligned}
w_{j+1/2}^{n+1} &= \frac{u_j^n + u_{j+1}^n}{2} + \frac{\Delta x - a_{j+1/2}^n \Delta t}{4} ((u_x)_j^n - (u_x)_{j+1}^n) \\
&\quad - \frac{1}{2a_{j+1/2}^n} \left\{ [F^R(u_{j+1/2,r}^{n+1/2}) - F^R(u_{j+1/2,l}^{n+1/2})] + \right. \\
&\quad \left. [v_{c,j+1/2,r}^{n+1/2} u_{j+1/2,r}^{n+1/2} - v_{c,j+1/2,l}^{n+1/2} u_{j+1/2,l}^{n+1/2}] \right\} \\
w_j^{n+1} &= u_j^n + \frac{\Delta t}{2} \left(a_{j-1/2}^n - a_{j+1/2}^n \right) (u_x)_j^n \\
&\quad - \frac{\lambda}{1 - \lambda(a_{j-1/2}^n + a_{j+1/2}^n)} \left\{ [F^R(u_{j+1/2,l}^{n+1/2}) - F^R(u_{j-1/2,r}^{n+1/2})] + \right. \\
&\quad \left. [v_{c,j+1/2,l}^{n+1/2} u_{j+1/2,l}^{n+1/2} - v_{c,j-1/2,r}^{n+1/2} u_{j-1/2,r}^{n+1/2}] \right\}
\end{aligned} \tag{4.17}$$

Now, considering a linear variation of the flux along the cells (see Figure 4.1) it is possible to find the values of the centered flux at the end of local speed influence zones as in Eqn. (4.18)

$$\begin{aligned}
v_{c,j+1/2,r}^{n+1/2} &:= v_{c,j+1/2}^n + \frac{v_{c,j+3/2}^n - v_{c,j+1/2}^n}{\Delta x} a_{j+1/2}^n \Delta t \\
v_{c,j+1/2,l}^{n+1/2} &:= v_{c,j+1/2}^n - \frac{v_{c,j+1/2}^n - v_{c,j-1/2}^n}{\Delta x} a_{j+1/2}^n \Delta t \\
v_{c,j-1/2,r}^{n+1/2} &:= v_{c,j-1/2}^n + \frac{v_{c,j+1/2}^n - v_{c,j-1/2}^n}{\Delta x} a_{j-1/2}^n \Delta t \\
v_{c,j-1/2,l}^{n+1/2} &:= v_{c,j-1/2}^n - \frac{v_{c,j-1/2}^n - v_{c,j-3/2}^n}{\Delta x} a_{j-1/2}^n \Delta t
\end{aligned} \tag{4.18}$$

Using the expression in Eqn. (4.18) in Eqn. (4.17) and doing the recon-

struction of face values with zero slope (first order scheme reduction) the final average values at the staggered mesh are found [see Eqn. (4.19)]

$$\begin{aligned}
w_{j+1/2}^{n+1} &= \frac{u_j^n + u_{j+1}^n}{2} - \frac{1}{2a_{j+1/2}n} [\mathbf{F}^R(u_{j+1}^n) - \mathbf{F}^R(u_j^n)] - \frac{u_{j+1}^n - u_j^n}{2a_{j+1/2}n} v_{c,j+1/2}^n \\
&\quad (1 - \lambda a_{j+1/2}) - \frac{\lambda}{2} [v_{c,j+3/2}^n u_{j+1}^n - v_{c,j-1/2}^n u_j^n] \\
w_j^{n+1} &= u_j^n - u_j^n \lambda [v_{c,j+1/2}^n - v_{c,j-1/2}^n] \tag{4.19} \\
w_{j-1/2}^{n+1} &= \frac{u_j^n + u_{j-1}^n}{2} - \frac{1}{2a_{j-1/2}n} [\mathbf{F}^R(u_j^n) - \mathbf{F}^R(u_{j-1}^n)] - \frac{u_j^n - u_{j-1}^n}{2a_{j-1/2}n} v_{c,j-1/2}^n \\
&\quad (1 - \lambda a_{j-1/2}) - \frac{\lambda}{2} [v_{c,j+1/2}^n u_j^n - v_{c,j-3/2}^n u_{j-1}^n]
\end{aligned}$$

where the value of $w_{j-1/2}^{n+1}$ has been also calculated. Finally using these expressions in Eqn. (4.8) and doing several algebraic steps, the form of the new scheme is obtained in Eqn. (4.20)

$$\begin{aligned}
u_j^{n+1} &= u_j^n - \frac{\lambda}{2} [\mathbf{F}^R(u_{j+1}^n) - \mathbf{F}^R(u_{j-1}^n)] + \\
&\quad \frac{1}{2} \left[\lambda a_{j+1/2}^n (u_{j+1}^n - u_{j+1}^n) - a_{j-1/2}^n (u_j^n - u_{j-1}^n) \right] - \\
&\quad \frac{\lambda}{2} v_{c,j+1/2}^n (u_{j+1}^n + u_j^n) + \frac{\lambda}{2} v_{c,j-1/2}^n (u_j^n + u_{j-1}^n) + \tag{4.20} \\
&\quad \frac{\lambda^2}{2} \left\{ a_{j+1/2}^n [(v_{c,j+1/2}^n - v_{c,j-1/2}^n) u_j^n - (v_{c,j+3/2}^n - v_{c,j+1/2}^n) u_{j+1}^n] + \right. \\
&\quad \left. a_{j-1/2}^n [(v_{c,j+1/2}^n - v_{c,j-1/2}^n) u_j^n - (v_{c,j-1/2}^n - v_{c,j-3/2}^n) u_{j-1}^n] \right\}
\end{aligned}$$

Checking the obtained expression it is clear that the first three terms correspond to the Rusanov scheme [Eqn. (4.9)], the fourth one is the centered flux balance in the j -th cell. The final term deserves a discussion. It is proportional to $\lambda^2 = \Delta t^2 / \Delta x^2$ and to $a v_c$. Both a and v_c have units of velocity,

then, writing all together it reads $\Delta t^2 v^2 / \Delta x^2 \equiv \mathbf{Co}^2$, where v is of the same order that the advective velocity. Then, if due to stability reasons the Courant number is such that $\mathbf{Co} \ll 1$ this term is proportional to $\mathbf{Co}^2 \ll 1$, so it is neglected.

4.2.4 Multidimensional extension

The multidimensional extension of the proposed Riemann-free solver with centered flux is based on the work of Greenshields *et al.* (Greenshields et al., 2010). Recalling the discretization of the convective term in the generalized transport equation [Eqn. (2.18)] it reads as in Eqn. (4.21)

$$\int_{\Gamma} \vec{v} \phi \cdot d\vec{\Gamma} = \sum_f \phi_f (\vec{v}_f \cdot \vec{S}_f) = \sum_f F_f \phi_f = \sum_f F_f \quad (4.21)$$

where ϕ_f is the face interpolated value obtained, in general, by some TVD method as was presented in Section 2.5 according to the direction given by the advective field. The total flux at faces F_f has the same meaning that in the MUSCL schemes; it is a total face flux calculated from reconstructed values of u at faces. This flux has to be calculated in any faces of the polyhedral cell. As was presented, the KT methods are based in face reconstructed values at both sides of a face and the information given by the eigenvalues or *local speeds*. In a polyhedral mesh this reconstruction is then done face-by-face in a fully multi-dimensional framework. The interpolation procedure is split in two directions according to the direction given by the face area vector \vec{S}_f ; the $f+$ direction coincides with \vec{S}_f and the opposite case with the $f-$ direction as is shown in Figure 4.2. Then, the KT methods can be written as in Eqn. (4.22)

$$\sum_f F_f = \sum_f \left[\frac{1}{2} F_{f+} + \frac{1}{2} F_{f-} + \frac{1}{2} a_f (\phi_{f+} - \phi_{f-}) \right] \quad (4.22)$$

where a_f are the local speeds at faces in the sense of KT [see Eqn. (4.1)]. These local speeds are calculated as face fluxes in a multidimensional framework as in Eqn. (4.23)

$$a_f = \max \left\{ \left| \rho \left(\frac{\partial F}{\partial \phi} \right)_{f+} \cdot \vec{S}_f \right|, \left| \rho \left(\frac{\partial F}{\partial \phi} \right)_{f-} \cdot \vec{S}_f \right| \right\} \quad (4.23)$$

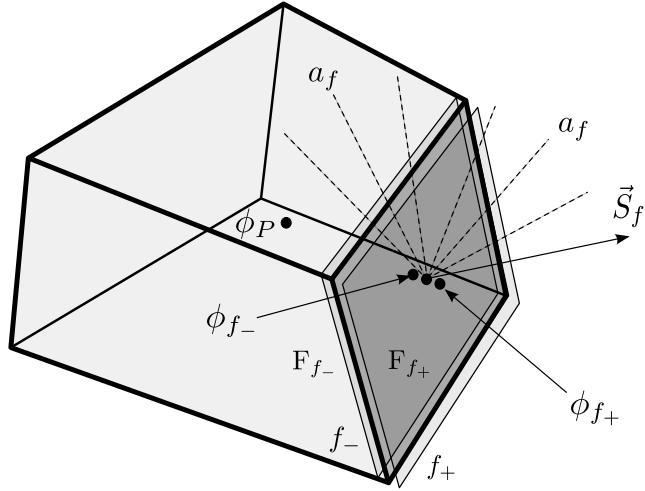


Figure 4.2: Scheme of face values reconstruction in a multidimensional framework

Finally, it is necessary to define the flux F_f in terms of the reconstructed flux F^R and the centered flux given at faces $F_c = v_c \cdot \vec{S}_f$ as in Eqn. (4.16), which leads to Eqn. (4.24)

$$\begin{aligned} \sum_f F_f &= \sum_f \left[\frac{1}{2} (F_{f+}^R + F_c \phi_{f+}) + \frac{1}{2} (F_{f-}^R + F_c \phi_{f-}) + \frac{1}{2} a_f (\phi_{f+} - \phi_{f-}) \right] \\ &= \sum_f \left[\frac{1}{2} (F_{f+}^R + F_{f-}^R) + \frac{1}{2} F_c (\phi_{f+} + \phi_{f-}) + \frac{1}{2} a_f (\phi_{f+} - \phi_{f-}) \right] \end{aligned} \quad (4.24)$$

This expression can be compared to the 1D dimensional zero slope form given in Eqn. (4.20). Therefore, the explicit integration of the model hyperbolic equation results to be which is shown in Eqn. (4.25)

$$\phi^{n+1} = \phi^n - \frac{1}{2} \frac{\Delta t}{V} \sum_f \left[\left(F_{f+}^{R,n} + F_{f-}^{R,n} \right) + F_c^n (\phi_{f+}^n + \phi_{f-}^n) + a_f (\phi_{f+}^n - \phi_{f-}^n) \right] \quad (4.25)$$

4.3 TVD-FCT method

The integration of the hyperbolic model equation by means of the TVD-FCT method relies on the concepts given in Section 2.6. Thus, starting from the mass conservation for the q phase, Eqn. (4.26), it reads

$$\frac{\partial \alpha_q}{\partial t} + \vec{\nabla} \cdot (\alpha_q \vec{u}) + \vec{\nabla} \cdot [\alpha_q (1 - \alpha_q) \vec{v}_{qp}] = 0$$

which reads in semi-discretized form as in Eqn. (4.26)

$$\frac{\partial \alpha_q}{\partial t} + \sum_f (\alpha_q \vec{u})_f \cdot \vec{S}_f + \sum_f [\alpha_q (1 - \alpha_q) \vec{v}_{qp}]_f \cdot \vec{S}_f = 0 \quad (4.26)$$

This equation can be re-written in terms of two face fluxes as in Eqn. (4.27)

$$\frac{\partial \alpha_q}{\partial t} + \sum_f (\alpha_q)_f^L F_f + \sum_f (\alpha_q)_f^{NL} F_f^{NL} = 0 \quad (4.27)$$

where $F_f = \vec{u}_f \cdot \vec{S}_f$ is the face flux of the linear term, and F_f^{NL} the face flux of the non-linear term with their respective values of the unknown at faces, $(\alpha_q)_f^L$ and $(\alpha_q)_f^{NL}$. The TVD-FCT method requires the construction of the F^H , the high order flux; in the construction the user can choose the interpolation method for $(\alpha_q)_f^L$ and $(\alpha_q)_f^{NL}$ meanwhile the faces fluxes are calculated by means for linearly interpolated values. Once the flux is limited the unknown is advanced in time by the usual explicit method.

Another important topic to address in the solution of this kind of equation via FCT is the selection of the maxima. As was previously presented a solution of the α_q which starts having a constant value can evolve having new minima and maxima (see Figure 3.8 for an example); this behavior violates the hypothesis of classical FCT limiters since they inspect each cell neighbours looking for local extrema (see Algorithm 1). In this case the obtained solutions are wrong and have a wave steepening effect (Nigam, 2003). The solution is simple and relies on taking into account only the global maxima as in Eqn. (4.28)

$$\begin{cases} \alpha_i^a = \alpha^{maxG} \\ \alpha_i^b = \alpha^{minG} \end{cases} \quad (4.28)$$

Thus, the FCT has been reduced to a way to avoid going beyond the global extrema; the resolution necessary for shock waves is then achieved by the TVD reconstruction used for the flux.

Chapter 5

Volume of Fluid Method

5.1 Introduction

As was stated in Chapter 1, the Volume of Fluid method is the selected technique for the treatment of the long scale interface. The VOF method can be classified as a Surface Capturing technique which implies that the free-surface is not exactly tracked by the mesh like in Surface Tracking methods, but its position is approximated by a phase fraction function (Carrica et al., 2006). In this sense the phase fraction function plays the same role as in ASMM. This similarity can be exploited in view of the unified framework needed for and extended mixture model.

Therefore, this chapter is devoted to show the derivation of the VOF method from the ASMM which sets a difference respect to the original approach given by Hirt & Nichols (Hirt and Nichols, 1981) in the presentation of the method. This derivation allows to understand the similarities between both methods and the potential for an unified framework and solver. Then, the algorithm for the VOF solver included in OpenFOAM® is explained and two academic cases such as the Rayleigh-Taylor instability and the Dam Break problem are solved. The solution of these problems allows to see how the numerical problem is set in a multi-phase case, by means of the initial and boundary conditions, the numerical settings and mesh details.

5.2 Derivation of VOF from ASMM

The derivation of the VOF method starts recalling the center-of-volume formulation of the ASMM as in Eqn. (3.39) and rewritten in Eqn. (5.1)

$$\left\{ \begin{array}{l} \vec{\nabla} \cdot \vec{u} = 0 \\ \frac{\partial}{\partial t}(\rho_m \vec{v}_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m \otimes \vec{v}_m) = -\vec{\nabla} p + \vec{\nabla} \cdot \left[\mu_m \left(\vec{\nabla} \vec{v}_m + \vec{\nabla} \vec{v}_m^T \right) \right] + \\ \quad \rho_m \vec{g} - \vec{\nabla} \cdot \left[\alpha_q (1 - \alpha_q) \frac{\rho_q \rho_p}{\rho_m} \vec{v}_{qp} \otimes \vec{v}_{qp} \right] \\ \frac{\partial \alpha_q}{\partial t} + \vec{\nabla} \cdot (\alpha_q \vec{u}) + \vec{\nabla} \cdot [\alpha_q (1 - \alpha_q) \vec{v}_{qp}] = 0 \end{array} \right. \quad (5.1)$$

A main difference between ASMM and the VOF method is that the VOF method considers a continuous velocity field along all the interfaces which is consistent with the interface boundary conditions given by the physics of fluids. This is possible since all the interfaces are supposed to be resolved at DNS scale. This hypothesis implies that the relative velocity between phases is null, $\vec{v}_{qp} = 0$. Thus, recalling the relationship between the center-of-mass velocity, \vec{v}_m and the center-of-volume velocity, \vec{u} given by Eqn. (3.31) the result is given by Eqn. (5.2)

$$\vec{v}_m = \vec{u} + \alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \vec{v}_{qp} = \vec{u} + \alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} 0 = \vec{u} \quad (5.2)$$

By inspection of the momentum equation given in the system of Eqn. (5.1) it is clear that the drift tensor $\overline{\tau_D} = \alpha_q (1 - \alpha_q) \frac{\rho_q \rho_p}{\rho_m} \vec{v}_{qp} \otimes \vec{v}_{qp}$ is also null. In addition the possibility of capturing long-scale interfaces allows to model the effects of the surface tension, which is achieved by the Continuum Surface Model (CSF) (Brackbill et al., 1992) which adds the term given in Eqn. (5.3)

$$\vec{F}_\sigma = \sigma \kappa \vec{\nabla} \alpha_q \quad (5.3)$$

where κ is the mean curvature of the free surface which is given by Eqn. (5.4)

$$\kappa = \vec{\nabla} \cdot \left(\frac{\vec{\nabla} \alpha_q}{|\vec{\nabla} \alpha_q|} \right) \quad (5.4)$$

In order to give stability to the solution and to simplify the definition of boundary conditions (Berberovic et al., 2009) the treatment of the pressure is done using the modified pressure p_{rgh} defined in Eqn. (5.5)

$$p_{rgh} = p - \rho_m \vec{g} \cdot \vec{x} \quad (5.5)$$

where \vec{x} is the position vector. Thus, the pressure gradient is then expressed as in Eqn. (5.6)

$$-\vec{\nabla} p = -\vec{\nabla} p_{rgh} - \vec{g} \cdot \vec{x} \vec{\nabla} \rho_m - \rho_m \vec{g} \quad (5.6)$$

regrouping terms the Eqn. (5.7) is obtained

$$-\vec{\nabla} p + \rho_m \vec{g} = -\vec{\nabla} p_{rgh} - \vec{g} \cdot \vec{x} \vec{\nabla} \rho_m \quad (5.7)$$

which allows to replace the pressure gradient and gravity terms in the second equation of Eqn. (5.1) by a function of the modified pressure. Finally it is important to note that the third equation of Eqn. (5.1) can be also simplified since the nonlinear term is zero due to the null relative velocity \vec{v}_{qp} . This term is deliberately included in the formulation with the aim of compressing the interface. It is worthy to note that since in VOF method α_q is expected to be always valued 0 or 1 except for the interfaces this term acts only on this place (Berberovic et al., 2009; Rusche, 2002; OpenCFD, 2005; Weller, 2008) and vanishes otherwise, then the original formulation for VOF is recalled in general and the non-linear term is used only at interface zones. Thus, the solved system reads as in Eqn. (5.8)

$$\left\{ \begin{array}{l} \vec{\nabla} \cdot \vec{u} = 0 \\ \frac{\partial}{\partial t} (\rho_m \vec{u}) + \vec{\nabla} \cdot (\rho_m \vec{u} \otimes \vec{u}) = -\vec{\nabla} p_{rgh} + \vec{\nabla} \cdot [\mu_m (\vec{\nabla} \vec{u} + \vec{\nabla} \vec{u}^T)] \\ -\vec{g} \cdot \vec{x} \vec{\nabla} \rho_m + \sigma \kappa \vec{\nabla} \alpha_q \\ \frac{\partial \alpha_q}{\partial t} + \vec{\nabla} \cdot (\alpha_q \vec{u}) + \vec{\nabla} \cdot [\alpha_q (1 - \alpha_q) \vec{v}_{qp}] = 0 \end{array} \right. \quad (5.8)$$

This formulation will hereinafter referred to as the Weller-VOF method.

5.3 Solver implementation

The solution of the system of equations given by Eqn. (5.8) relies on a pressure-velocity coupling loop based on PISO, basically a derivation of the method presented in section 2.7. In addition it is necessary to solve the α_q equation which is achieved by means of the MULES explicit solver based on the FCT technique (Rudman, 1997). Here the original definition of the local maxima and minima was retained since the α_q doesn't generate new maxima and minima.

The general method for the solution of α_q equation was presented in section 4.3 and is based in its discretized form given in Eqn. (4.26) and recalled in Eqn. (5.9)

$$\frac{\partial \alpha_q}{\partial t} + \sum_f (\alpha_q \vec{u})_f \cdot \vec{S}_f + \sum_f [\alpha_q (1 - \alpha_q) \vec{v}_{qp}]_f \cdot \vec{S}_f = 0 \quad (5.9)$$

This equation can be rewritten in terms of face fluxes and using an explicit integration scheme as in Eqn. (5.10)

$$\frac{\alpha_q^{n+\nu+1} - \alpha_q^n}{\Delta t} V + \sum_f \left\{ (\alpha_q^{n+\nu})_f F^{L,n+\nu} + [\alpha_q^{n+\nu} (1 - \alpha_q^{n+\nu})]_f F^{NL,n+\nu} \right\} = 0 \quad (5.10)$$

where ν is the number of the actual fixed-point iteration in order to circumvent the issue of the non-linearity of the fluxes. This loop is solved `nAlphaCorrectors` times. $F^{L,n}$ is the flux due to the center-of-volume velocity and $F^{NL,n}$ is the flux due to the artificial compressive velocity added at the interfaces. This velocity is not calculated explicitly but through a flux as in Eqn. (5.11)

$$F^{NL,n} = n_f \min \left[C_\alpha \frac{|\Phi^{L,n}|}{|\vec{S}_f|}, \max \left(\frac{|\Phi^{L,n}|}{|\vec{S}_f|} \right) \right] \quad (5.11)$$

where C_α is an adjustment constant, $n_f = \frac{(\nabla \alpha_q)_f}{|(\nabla \alpha_q)_f + \delta_n|} \cdot \vec{S}_f$ is the face unit normal flux with $\delta_n = \frac{\varepsilon}{(\sum_N V_i)^{1/3}}$ as a stabilization factor to avoid division by zero, with $\varepsilon = 1 \times 10^{-8}$. The direction of the compressive velocity is given by the gradient of α_q and ensures to apply the compression in the right direction, that is, perpendicular to the interface. The values of $(\alpha_q^n)_f$ are calculated by a selectable High Resolution Scheme, particularly in the non-linear term they can be discretized by the *interfaceCompression* scheme, which has been devised specially for its use in equation (5.10). The blending factor λ for UD and CD in Eqn. (2.37) is given by Eqn. (5.12)

$$\lambda = \min \left(\max \left[1 - \max \left\{ (1 - 4\alpha_P(1 - \alpha_P))^2, (1 - 4\alpha_N(1 - \alpha_N))^2 \right\}, 0 \right], 1 \right) \quad (5.12)$$

where α_P is the value of α_q at the current cell and α_N in the neighbour cell. This scheme does not obey to an TVD/NVD analysis but was selected in order to guarantee the use of UD when α_q is near the extrema and CD otherwise and is a good combination of boundedness and convergence (Peng Karrholm, 2008; Weller, 2008). See (Gastaldo et al., 2011) for an example of this kind of method in ASMM solving under different solution methodology.

In addition, the solution method of the system given in Eqn. (5.8) includes an adaptive time-step control and the sub-cycling in the solution of α_q equation. The adaptive time-step control is done calculating the time-step by means of

Eqn. (5.13)

$$\Delta t^n = \min \left\{ \frac{\mathbf{Co}_{max}}{\mathbf{Co}} \Delta t^0, \left(1 + \lambda_1 \frac{\mathbf{Co}_{max}}{\mathbf{Co}} \right) \Delta t^0, \lambda_2 \Delta t^0, \Delta t_{max} \right\} \quad (5.13)$$

where $\mathbf{Co} = \frac{|\vec{u}_f \cdot \vec{S}_f|}{d_{PN} \cdot \vec{S}_f} \Delta t$ is the face-computed Courant number, \mathbf{Co}_{max} , Δt_{max} are user-defined parameters and $\lambda_1 = 0.1$ and $\lambda_2 = 1.2$ are two hard-coded factors in order to reduce immediately the time-step but to increase it gradually to avoid unstable oscillations. Regarding to \mathbf{Co}_{max} Gopala and van Wachem (Gopala and van Wachem, 2008) recommend a value of lesser than 0.3, Berberovic *et al.* (Berberovic et al., 2009) a value of approximately 0.2. By the author's experience the last value results successful, but some runnings have been done using $\mathbf{Co}_{max} = 0.1$ to reach the expected results. It is important to note another observed behavior of the adaptive time-step method. Since it is Courant number based, the time-step can go to excessive high values at the beginning of the simulation if no high velocity values are present in the simulated problem. It is mathematically correct but can play against the physics of the phenomenon giving inaccurate values. This issue can be circumvented using the Δt_{max} parameter or using a fixed time-step for a while. Similar conclusions were informed by Berberovic *et al.*

Respect to the sub-cycling it is performed in order to give stability to the solution of the α_q equation, then choosing a number of sub-cycles n_{sc} the sub-step is defined as in Eqn. (5.14)

$$\Delta t_{sc} = \frac{\Delta t}{n_{sc}} \quad (5.14)$$

Once the α_q is solved is then necessary to assemble and solve the discretized version of the momentum equation which is shown in Eqn. (5.15)

$$\begin{aligned}
& \frac{\rho_m^{n+1} \tilde{\vec{u}} - \rho_m^n \vec{u}_n}{\Delta t} V + \sum_f F_{\rho_m}^{n+1} \tilde{\vec{u}} \cdot \vec{S}_f = \\
& \sum_f (\mu_m^{n+1})_f \left(\vec{\nabla} \tilde{\vec{u}} \right)_f \cdot \vec{S}_f + \left(\vec{\nabla} \vec{u}^n \cdot \vec{\nabla} \mu_m^{n+1} \right) V + \\
& \mathcal{R} \left\{ \left[(\sigma \kappa)_f \left(\vec{\nabla} \alpha_q^{n+1} \right)_f - (\vec{g} \cdot \vec{x})_f \left(\vec{\nabla} \rho_m^{n+1} \right)_f - \left(\vec{\nabla} p_{rgh}^n \right)_f \right] \left| \vec{S}_f \right| \right\}
\end{aligned} \tag{5.15}$$

where F_{ρ_m} is the mass face flux given by $F_{\rho_m} = (\rho_m \vec{v}_m)_f \cdot \vec{S}_f$ and $\vec{a} = \mathcal{R}(\vec{a} \cdot \vec{S}_f)$ is an operator to reconstruct cell-centered fields from fields given as fluxes at faces. This mass flux has to be assembled carefully within each sub-cycle. The basic assembling of this flux is given by the relationship between the center-of-mass velocity and the center-of-volume velocity and is necessary since the α_q equation modifies the mass distribution, therefore recalling Eqn. (3.31) and assembling a mass face flux it becomes which is shown in Eqn. (5.16)

$$F_{\rho_m} = (\rho_m \vec{v}_m)_f \cdot \vec{S}_f = (\rho_m \vec{u})_f \cdot \vec{S}_f + [\alpha_q (1 - \alpha_q) (\rho_q - \rho_p) \vec{v}_{qp}]_f \cdot \vec{S}_f \tag{5.16}$$

which can be re-written as in Eqn. (5.17)

$$F_{\rho_m} = \rho_m F + \alpha_q (1 - \alpha_q) (\rho_q - \rho_p) F_{qp} \tag{5.17}$$

where F_{qp} is the face flux for \vec{v}_{qp} . On the other hand the discretization of α_q requires the assembling of the flux given in Eqn. (5.18) [see Eqn. (5.10)]

$$F_{\alpha_q} = (\alpha_q^{n+\nu})_f F^{n+\nu} + [\alpha_q^{n+\nu} (1 - \alpha_q^{n+\nu})]_f F^{NL,n+\nu} \tag{5.18}$$

Now, doing some simple algebraic manipulations is easy to show what is shown in Eqn. (5.19)

$$F_{\alpha_q} (\rho_q - \rho_p) + F \rho_p = \rho_m F + \alpha_q (1 - \alpha_q) (\rho_q - \rho_p) F_{qp} = F_{\rho_m} \tag{5.19}$$

which is the desired relationship, shown more clearly in Eqn. (5.20)

$$F_{\rho m} = F_{\alpha_q} (\rho_q - \rho_p) + F \rho_p \quad (5.20)$$

Therefore, in each subcycle a partial mass flux is assembled as $F_{\rho m, sc, i}$. The mass flux for the complete time-step is obtained by the discrete integral form of the mean value theorem as in Eqn. (5.21)

$$F_{\rho m} = \sum_{i=1}^{n_{sc}} \frac{\Delta t_{sc}}{\Delta t} F_{\rho m, sc, i} \quad (5.21)$$

The whole process of the numerical solution of the Weller-VOF method as is implemented in the OpenFOAM®'s solver `interFoam` is presented in Algorithm 4. The algorithm includes two methods in the PISO loop, `ddtPhiCorr(1/a_P, rho_m, u^v, F^v)` is a flux adjustment due to the time-step needed by the Rhie-Chow interpolation (Choi, 1999), meanwhile `adjustPhi(F_u^v+1, H(u^v)/a_P, p_gh^v)` adjusts the flux in free boundaries to obey the continuity equation for \vec{u} .

5.4 Some classical examples

In order to set a background for more complex models, some classical examples will be solved. The first one corresponds to the Rayleigh-Taylor instability (Štrubelj and Tiselj, 2011), which consists on the evolution of two layers of fluids. The top layer is more dense than the one is placed at the bottom. Due to a little disturbance in the contact surface the more dense fluid goes down and the less dense fluid does the opposite. In the intermediate state a mixture is created, which is lately segregated. The final state reaches an stable equilibrium with the more dense fluid at the bottom layer and the less dense fluid at the top layer.

The second example is the Dam Break problem, which is widely used as test problem for multi-phase solvers (Martin and Moyce, 1952; Cruchaga et al., 2007; Battaglia et al., 2010). In this case a cavity is filled by the less dense fluid and a column of the more dense fluid is formed in a corner. This column

Algorithm 4 Weller-VOF solver with FCT (MULES) and pressure-velocity coupling via PISO (`interFoam`)

1. Solve the mass conservation equation for the primary phase, α_q^{n+1} , assemble the mass face flux $F_{\rho m}^{n+1}$ and get the new mixture density ρ_m^{n+1} by a loop of `nAlphaSubCycles` cycles where:

- a) The compressive velocity at interfaces is calculated as a flux by Eqn. (5.11)

$$F^{NL,n} = n_f \min \left[C_\alpha \frac{|\Phi^{L,n}|}{|\vec{S}_f|}, \max \left(\frac{|\Phi^{L,n}|}{|\vec{S}_f|} \right) \right]$$

- b) The α_q equation [Eqn. (5.10)] is solved $0 < \nu < \text{nAlphaCorrectors}$ times by the MULES integrator which also returns the limited flux F_{α_q}

$$\frac{\alpha_q^{n,\nu+1} - \alpha_q^n}{\Delta t} V + \sum_f \left\{ (\alpha_q^{n,\nu})_f F^n + [\alpha_q^{n,\nu} (1 - \alpha_q^{n,\nu})]_f F^{NL,n,\nu} \right\} = 0$$

- c) The new mass face flux for the present sub-cycle is calculated by Eqn. (5.20). At the end of the loop the final mass face flux is calculated by Eqn. (5.21) and the density is updated

$$F_{\rho m, sc, i} = F_{\alpha_q} (\rho_q - \rho_p) + F \rho_p \quad F_{\rho m} = \sum_{i=1}^{n_{sc}} \frac{\Delta t_{sc}}{\Delta t} F_{\rho m, sc, i} \quad \rho_m^{n+1} = \alpha_q \rho_q + (1 - \alpha_q) \rho_p$$

2. Solve the momentum predictor [discretized version of the second equation in Eqn. (5.8)] for $\tilde{\vec{u}}$ if the `momentumPredictor` flag is set to `yes`

$$\begin{aligned} \frac{\rho_m^{n+1} \tilde{\vec{u}} - \rho_m^n \vec{u}^n}{\Delta t} V + \sum_f F_{\rho m}^{n+1} \tilde{\vec{u}} \cdot \vec{S}_f &= \sum_f (\mu_m^{n+1})_f (\vec{\nabla} \tilde{\vec{u}})_f \cdot \vec{S}_f + (\vec{\nabla} \vec{u}^n \cdot \vec{\nabla} \mu_m^{n+1}) V + \\ \mathcal{R} \left\{ \left[(\sigma \kappa)_f (\vec{\nabla} \alpha_q^{n+1})_f - (\vec{g} \cdot \vec{x})_f (\vec{\nabla} \rho_m^{n+1})_f - (\vec{\nabla} p_{rgh}^n)_f \right] |\vec{S}_f| \right\} \end{aligned}$$

3. Do the PISO loop $0 < \nu < \text{nCorrectors}$ times, where:

- a) A face flux is calculated using the $H(\vec{u})$ operator with $\tilde{\vec{u}}$ obtained in the momentum predictor. This face flux does not take into account the effects of the gravity and the surface tension

$$\Phi_u^{\nu+1}{}_f = \left(\frac{H(\tilde{\vec{u}}^\nu)}{a_P} \right)_f \cdot \vec{S}_f + \text{ddtPhiCorr}(1/a_P, \rho_m, \vec{u}^\nu, \Phi^\nu)$$

The flux is then adjusted to obey continuity, via `adjustPhi($\Phi_u^{\nu+1}, H(\vec{u}^\nu)/a_P, p_{rgh}^\nu$)` method

- b) The final proposed flux is found adding the effects of the gravity and the surface tension

$$\Phi^{\nu+1}{}_f = \Phi_u^{\nu+1} + \left[(\sigma \kappa)_f (\vec{\nabla} \alpha_q^{n+1})_f - (\vec{g} \cdot \vec{x})_f (\vec{\nabla} \rho_m^{n+1})_f \right] \frac{|\vec{S}_f|}{(a_P)_f}$$

- c) The pressure equation is assembled and solved `nNonOrthogonalCorrectors` times for $p_{rgh}^{\nu+1}$

$$\sum_f \left[\left(\frac{1}{a_P} \right)_f (\vec{\nabla} p_{rgh}^{\nu+1})_f \right] \cdot \vec{S}_f = \sum_f \Phi^{\nu+1}{}_f$$

- d) The proposed flux is then adjusted by the effect of the pressure and then the center-of-volume velocity at the cell centers is adjusted as well

$$\Phi^{\nu+1}{}_f = \Phi^{\nu+1}{}_f - \left[\left(\frac{1}{a_P} \right)_f (\vec{\nabla} p_{rgh})_f \right] \cdot \vec{S}_f \quad \vec{u}^{\nu+1} = \vec{u}^{\nu+1} \frac{1}{a_P} \mathcal{R} \left[(\Phi^{\nu+1}{}_f - \Phi_u^{\nu+1}{}_f) (a_P)_f \right]$$

- e) Finally, the static pressure is reconstructed p from the modified p_{rgh} as $p = p_{rgh} + \rho_m \vec{g} \cdot \vec{x}$
-

suddenly collapses evolving within the cavity with waves and splashing which causes mixing between the fluids. In this case an obstacle has been added in order to assure stronger agitation and mixing. The final state is logically a quiescent pool with the more dense fluid at the bottom and the less dense fluid at the top.

5.4.1 Rayleigh-Taylor instability

The Rayleigh-Taylor instability is numerically simulated in the domain $[0, 0] \times [1, 5]$ with an hexahedral mesh of 128×640 (81920) elements (see Figure 5.1), which is created using the `blockMesh` utility. The physical parameters for the fluids are $\rho_q = 3$, $\nu_q = 0.01$ and $\rho_p = 1$, $\nu_p = 0.01$, without surface tension. The gravity is set as $\vec{g} = (0, -10, 0)$ and the expression for initial disturbance in the free surface is given by the expression in Eqn. (5.22), with amplitude $\delta_0 = 0.001$

$$\delta = -\delta_0 \left[\cos \left(\frac{2\pi x}{L} - \pi \right) + 1 \right] + 4.5 \quad (5.22)$$

The shape of the initial disturbance is followed by the mesh on the interface zone since the size of the deformation is smaller to the mesh step. This small size impedes to set the initial disturbance only by cell initialization. Respect to the boundary conditions for α_q , `zeroGradient` was set at bottom boundaries of the domain and symmetry conditions at both sides. The same boundary conditions were set for the modified pressure and a pressure reference point was set at $(0.4999, 0.00078125, 0)$ with $p = 0$ in order not to disturb the initial steps of the running. Finally, the boundary conditions for the velocity were set as no-slip for top and bottom boundaries and symmetry conditions at both sides. Since the mesh has a dummy third dimension, z , front and back boundaries were set as `empty`.

The solver was set with the following parameters: `momentumPredictor=yes`, `nCorrectors=3`, `nNonOrthogonalCorrectors=1`, `nAlphaCorr=1`, `nAlphaSubCycles=2`, `cAlpha=0.25`. The value of `cAlpha` was selected in order not to form spurious ripple in the free surface when the big structures of the flow

are being stretched. The time-step was set as $\Delta t = 0.0001$ which assured a Courant number below 0.1 in all the run and was proper for the development of the mushroom-like structure at the beginning of the simulation.

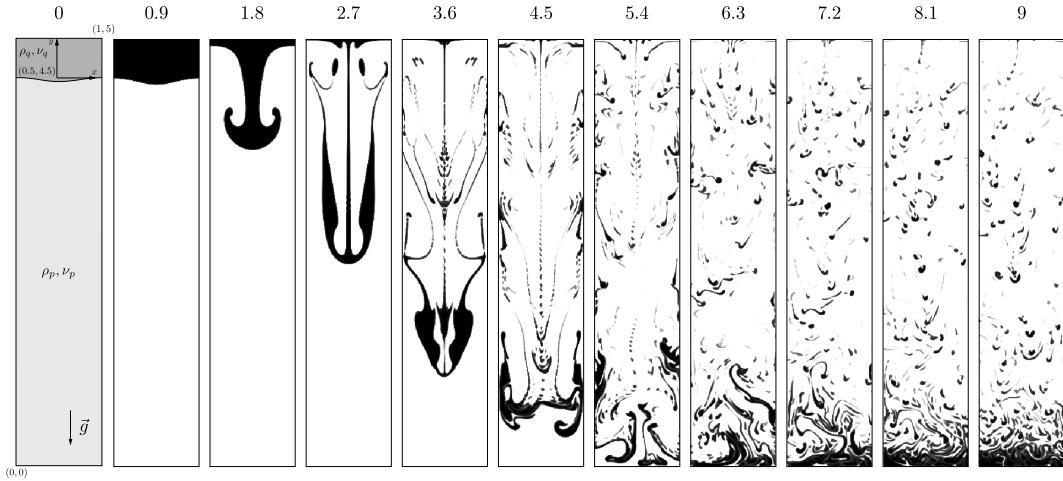


Figure 5.1: Evolution of the Rayleigh-Taylor instability.

The results are presented in Figure 5.1 for several times. The evolution starts with the development of a mushroom-like structure, until approximately $t = 1.8$. After this time the structure is stretched and filaments start to detach forming isolated chunks (see figure at $t = 2.7, t = 3.6$). The detaching continues keeping only chunks and filaments, some of these chunks are fragmentated in a size not trackable by the VOF method. At the end a sedimentation process starts where the more dense chunks fall as “droplets” and the less dense fluid trapped in the bottom layer scape by buoyancy as “bubbles”. The final state (not shown) is clearly a layered solution with the more dense fluid forming the bottom layer. The obtained evolution of the problem is in general accordance with the references setting a basis for comparisons with other methods.

5.4.2 Dam Break

The Dam Break test is numerically simulated in the domain $[0, 0] \times [0.584, 0.584]$ with an hexahedral mesh of approximately 720×688 (495360) elements, meshed with `blockMesh` (the real geometry is not a rectangle but has an obstacle, see Figure 5.2). The physical parameters for the fluids are $\rho_q = 1000$, $\nu_q = 1 \times 10^{-6}$ and $\rho_p = 1$, $\nu_p = 1.48 \times 10^{-5}$ with a surface tension $\sigma = 0.07$. The gravity is set as $\vec{g} = (0, -9.81, 0)$. The solution domain is filled with the less dense fluid except for the area given by $[0, 0] \times [0.1461, 0.438]$ where the more dense fluid is located. This rectangle gives the initial condition of the water column. This column is obviously not in equilibrium, then, when the simulation starts it collapses like in the breaking of a dam.

Respect to the boundary conditions for α_q , `zeroGradient` was set in all walls except for the top where the `inletOutlet` boundary condition was used. Respect to the boundary conditions for the modified pressure it were set as `buoyantPressure` for all wall except for the top, where the `totalPressure` boundary condition was set. Finally, for the center-of-volume velocity the non-slip boundary condition was set for all walls except for the top wall where the `pressureInletOutletVelocity` was used. Due to the mesh has a dummy third dimension, z , front and back boundaries were set as `empty`.

The solver was set with the following parameters: `momentumPredictor=no`, `nCorrectors=3`, `nNonOrthogonalCorrectors=0`, `nAlphaCorr=1`, `nAlphaSubCycles=2`, `cAlpha=1`. The maximum Courant number for this running was set as 0.5 which is greater that the recommended, however it resulted to be stable.

The evolution of the Dam Break problem is shown in Figure 5.2. In the beginning, the more dense fluid column collapses and passes over the obstacle until reaching the right wall in a very ordered flow (up to $t = 0.7$, approximately). Once the fluid reaches the right wall it splashes forming chunks and droplets. The flow oscillates to the left wall ($t = 2.8$) and the interface breaks in several other small interfaces trapping the less dense fluid.

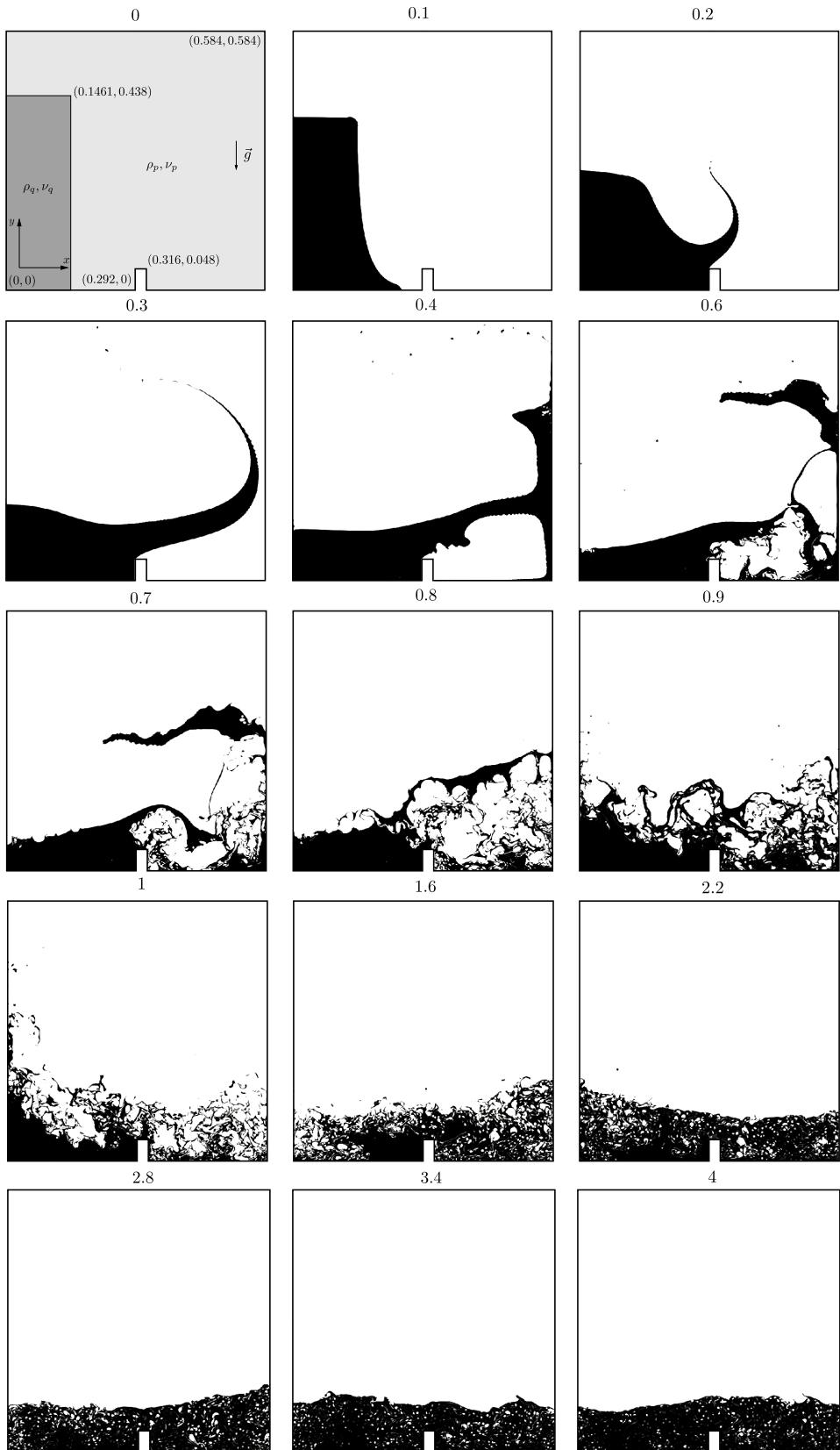


Figure 5.2: Evolution of the Dam Break test.

The oscillation continues as a liquid pendulum which is damped by the wall and internal friction due to the viscosity ($t \geq 3.4$). Finally, the system starts a quiescent stage forming a pool with the more dense fluid at the bottom layer and the less dense fluid trapped in “bubbles” which are removed by buoyancy reaching a complete segregated and hydrostatic state (not shown). Again, the evolution is in accordance with the references and sets a basis for further comparisons.

Chapter 6

Numerical solvers for the mixture model

6.1 Introduction

Having in mind the formulation and particular numerical issues of the ASMM studied in Chapter 3 it is necessary to devise a solver capable to manage them. In addition, the proposed solver has to have enough flexibility to include the VOF model within it in the way that was done in Chapter 5. Therefore this chapter presents an ASMM solver based on a PISO pressure-velocity coupling and the solution of the phase fraction equation using the KT-MULES and TVD-MULES techniques already studied. The use of these techniques assures the assembling of a conservative flux and the bounded solution of the phase fraction equation. The solver is validated by the 1D examples given for sedimentation and calculated by the presented semi-analytic solution and another 1D example taken from literature. Finally a bubble reactor is solved and the results are compared to experiments.

6.2 Formulations for the ASMM

In the derivation of the ASMM explained in chapter 3 two main formulations were obtained for the system of equations of the model. In the first one, the

so called velocity of center-of-mass formulation, all the equations are written in terms of the mass averaged velocity, which is presented in Eqn. (3.22) and recalled in Eqn. (6.1)

$$\left\{ \begin{array}{l} \frac{\partial}{\partial t}(\rho_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m) = 0 \\ \frac{\partial}{\partial t}(\rho_m \vec{v}_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m \otimes \vec{v}_m) = -\vec{\nabla} p + \vec{\nabla} \cdot \left[\mu_m \left(\vec{\nabla} \vec{v}_m + \vec{\nabla} \vec{v}_m^T \right) \right] + \rho_m \vec{g} - \vec{\nabla} \cdot [\rho_m c_p (1 - c_p) \vec{v}_{pq} \otimes \vec{v}_{pq}] \\ \frac{\partial}{\partial t}(\alpha_p) + \vec{\nabla} \cdot (\alpha_p \vec{v}_m) = -\vec{\nabla} \cdot [\alpha_p (1 - c_p) \vec{v}_{pq}] \end{array} \right. \quad (6.1)$$

The solution of this system brings its own problems such as were mentioned previously: the lack of an evolution equation for the pressure and the necessity of the bounded solutions for α_p . The discretization of the system by the FVM leads to another particular issues. One of the most important details is the assembling of a conservative flux, i.e. a flux which satisfies the mass conservation equation for the mixture [first equation in the system of Eqn. (6.1)] and the use of this flux in all other equations. Thus, the flux corresponds to the mass face flux or the face flux of $\rho_m v_m$. Note that this quantity is used in the discretization of the mass conservation equation and the momentum equation but not in the secondary phase conservation equation. This issue can be circumvented using the method proposed by Brennan (Brennan, 2001) which relies in re-write this equation as is shown in Eqn. (6.2)

$$\frac{\partial}{\partial t} A \rho_m + \frac{\partial}{\partial z} \{ A [\rho_m v_m + \rho_m (1 - c_p) v_{pq}] \} = 0 \quad (6.2)$$

where $A = \frac{\alpha_p \rho_p}{\rho_m}$. However, the discretization of this new equation requires the value of ρ_m at the new time-step. This is often obtained by a prediction based in the mass conservation equation of the mixture as is proposed by Brennan. This technique is also used by Gastaldo *et al.* (Gastaldo et al., 2011) in conjunction with a time-shifting. By the author's experience (Márquez Damián and Nigro, 2011; Márquez Damián et al., 2012) and enforced by the conclusions

of the cited literature the solution obtained with such a strategy lacks of mass conservation.

The formulation and solver for the ASMM based on the center-of-mass velocity is revisited by Bohorquez (Bohorquez R. de M., 2008) as a [“Compressible pressure-based solver” since it relies in the determination of a non divergence free velocity by means of a pressure-based method as PISO. As a counterpart for this approach a “Segregated pressure-based solver” is derived (Bohorquez R. de M., 2008; Bohorquez, 2012) based on the velocity of center-of-volume formulation. The principal advantage of this method is the possibility to use a divergence free velocity in the mass conservation equation for the primary phase. In addition, it is then possible to avoid the prediction of ρ_m which is no more an unknown but a derived quantity from ρ_q , ρ_p and α_q .

Thus, starting from the ASMM formulation written in velocity of center-of-volume formulation [Eqn. (3.39)] and using the concept of modified pressure the system reads as in Eqn. (6.3).

$$\left\{ \begin{array}{l} \vec{\nabla} \cdot \vec{u} = 0 \\ \frac{\partial}{\partial t} (\rho_m \vec{v}_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m \otimes \vec{v}_m) = -\vec{\nabla} p_{rgh} + \vec{\nabla} \cdot \left[\mu_m \left(\vec{\nabla} \vec{v}_m + \vec{\nabla} \vec{v}_m^T \right) \right] \\ \quad - \vec{g} \cdot \vec{x} \vec{\nabla} \rho_m - \vec{\nabla} \cdot \left[\alpha_q (1 - \alpha_q) \frac{\rho_q \rho_p}{\rho_m} \vec{v}_{qp} \otimes \vec{v}_{qp} \right] \\ \frac{\partial \alpha_q}{\partial t} + \vec{\nabla} \cdot (\alpha_q \vec{u}) + \vec{\nabla} \cdot [\alpha_q (1 - \alpha_q) \vec{v}_{qp}] = 0 \end{array} \right. \quad (6.3)$$

Comparing this system with the system of equations given for the VOF method [Eqn. (5.8)] is clear that since the latter is derived from the former they have much similarities. The differences are in the velocity formulation in the momentum equation and the particular terms of each method (surface tension for VOF and drift tensor for ASMM) and in the meaning of the relative velocity definition in the primary phase mass conservation equation. These

similarities can be exploited to devise an ASMM solver on the basis of the Weller-VOF solver previously presented in Algorithm 4 (Section 5.3). To obtain such a solver it is necessary to have a flux relationship between the velocity of center-of-mass and the velocity of center-of-volume and a new pressure equation. Therefore, starting from Eqn. (3.31), isolating \vec{u} and multiplying by the face area vector the desired flux relationship is obtained in Eqn. (6.4)

$$\vec{u}_f \cdot \vec{S}_f = (\vec{v}_m)_f \cdot \vec{S}_f - \left[\alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \right]_f (\vec{v}_{qp})_f \cdot \vec{S}_f \quad (6.4)$$

This relationship allows to solve the phase fraction transport equation [third equation in Eqn. (6.3)] using a conservative flux for \vec{u} assembled from the flux of \vec{v}_m given by the last PISO loop. In addition, from the derivation of the pressure equation in section 2.7.1 [Eqn. (2.62)] the velocity of center-of-mass can be expressed at faces (in terms of the modified pressure here) as in Eqn. (6.5)

$$(\vec{v}_m)_f = \left(\frac{\vec{H}(\vec{v}_m)}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f (\vec{\nabla} p_{rgh})_f \quad (6.5)$$

Then, the flux relationship can be re-written as in Eqn. (6.6)

$$\begin{aligned} \vec{u}_f \cdot \vec{S}_f = & \left[\left(\frac{\vec{H}(\vec{v}_m)}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f (\vec{\nabla} p_{rgh})_f \right] \cdot \vec{S}_f \\ & - \left[\alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \right]_f (\vec{v}_{qp})_f \cdot \vec{S}_f \end{aligned} \quad (6.6)$$

Using this expression in the discretized version of the incompressibility restriction for \vec{u} , $\sum_f \vec{u}_f \cdot \vec{S}_f = 0$ and re-arranging, Eqn. (6.7) is obtained

$$\begin{aligned} \sum_f \left[\left(\frac{1}{a_P} \right)_f (\vec{\nabla} p_{rgh}^{\nu+1})_f \right] \cdot \vec{S}_f = & \sum_f \left(\frac{\vec{H}(\vec{v}_m)}{a_P} \right)_f \cdot \vec{S}_f \\ & - \sum_f \left[\alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \right]_f (\vec{v}_{qp})_f \cdot \vec{S}_f \end{aligned} \quad (6.7)$$

Finally, adding the effect of the gravitational acceleration, the pressure

equation is obtained in Eqn. (6.8)

$$\begin{aligned} \sum_f \left[\left(\frac{1}{a_P} \right)_f \left(\vec{\nabla} p_{rgh}^{\nu+1} \right)_f \right] \cdot \vec{S}_f = & \sum_f \left[\left(\frac{\vec{H}(v_m)}{a_P} \right)_f - (\vec{g} \cdot \vec{x})_f \left(\vec{\nabla} \rho_m^{n+1} \right)_f \right] \cdot \vec{S}_f \\ & - \sum_f \left[\alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \right]_f (\vec{v}_{qp})_f \cdot \vec{S}_f \end{aligned} \quad (6.8)$$

6.3 Solver implementation

Given the formulation of the ASMM and the auxiliary equations needed for the pressure-velocity coupling is now possible to describe the solver algorithm. It is based in Algorithm 4 which is the implementation of the Weller-VOF method and is presented in Algorithm 5 and follows its general structure. In the first step the α_q is solved. In this stage the MULES integrator is used in order to guarantee the boundedness, in this case the flux F_{α_q} needed for the integrator can be assembled either using standard TVD methods or the flux for a Riemann-free solver as was presented in chapter 5. The momentum predictor is solved and the PISO loop is performed. Note the extra term in the r.h.s. of the pressure equation which was studied in the previous section in Eqn. (6.8). Finally the flux for the velocity of center-of-volume is recovered in order to be used for the integration of α_q in the next time-step. Thus the relationship between the velocity of center-of-mass, \vec{v}_m , and the velocity of center-of-volume, \vec{u} , needed in the system given in Eqn. (6.3) is expressed in terms of *face fluxes*. The correct calculation of the flux and its treatment along the solving algorithm play a central role in the successful implementation of a solver.

Since the presented algorithm is not available in the OpenFOAM® suite it was programmed starting from the `interFoam` (Weller-VOF) solver and using the `gdbOF` debugging tools [see Appendix A or (Márquez Damián et al., 2012)].

Algorithm 5 Pressure based ASMM solver with velocity of center-of-volume formulation

- Solve the mass conservation equation for the primary phase for α_q^{n+1} , assemble the mass face flux $F_{\rho m}^{n+1}$ and get the new mixture density ρ_m^{n+1} by a loop of `nAlphaSubCycles` cycles where:

a) The relative velocity at faces is calculated as a flux from Eqn. (3.25)

$$F^{NL,n} = \vec{v}_{qp}^n \cdot \vec{S}_f = \vec{v}_{rc} (\alpha_q^n)^a \cdot \vec{S}_f$$

b) The α_q equation [Eqn. (5.10)] is solved $0 < \nu < \text{nAlphaCorrectors}$ times by the MULES integrator which also returns the limited flux F_{α_q} . This flux can be computed using either TVD reconstruction or a Riemann-free solver flux

$$\frac{\alpha_q^{n,\nu+1} - \alpha_q^n}{\Delta t} V + \sum_f \left\{ (\alpha_q^{n,\nu})_f F^{L,n} + [\alpha_q^{n,\nu} (1 - \alpha_q^{n,\nu})]_f F^{NL,n,\nu} \right\} = 0$$

c) The new mass face flux for the present sub-cycle is calculated by Eqn. (5.20). At the end of the loop the final mass face flux is calculated by Eqn. (5.21) and the density is updated

$$F_{\rho m,sc,i} = F_{\alpha_q} (\rho_q - \rho_p) + F \rho_p \quad F_{\rho m} = \sum_{i=1}^{n_{sc}} \frac{\Delta t_{sc}}{\Delta t} F_{\rho m,sc,i} \quad \rho_m^{n+1} = \alpha_q \rho_q + (1 - \alpha_q) \rho_p$$

- Solve the momentum predictor [discretized version of the second equation in Eqn. (6.3)] for \tilde{v}_m if the `momentumPredictor` flag is set to yes

$$\begin{aligned} \frac{\rho_m^{n+1} \tilde{v}_m - \rho_m^n \vec{v}_m^n}{\Delta t} V + \sum_f F_{\rho m}^{n+1} \tilde{v}_m \cdot \vec{S}_f &= \sum_f (\mu_m^{n+1})_f (\vec{\nabla} \tilde{v}_m)_f \cdot \vec{S}_f \\ &+ (\vec{\nabla} \vec{v}_m^n \cdot \vec{\nabla} \mu_m^{n+1}) V - \sum_f \left[\alpha_q^{n+1} (1 - \alpha_q^{n+1}) \frac{\rho_q \rho_p}{\rho_m^{n+1}} \vec{v}_{qp}^n \otimes \vec{v}_{qp}^n \right]_f \cdot \vec{S}_f \\ &- \mathcal{R} \left\{ \left[(\vec{g} \cdot \vec{x})_f (\vec{\nabla} \rho_m^{n+1})_f + (\vec{\nabla} p_{rgh}^n)_f \right] \left| \vec{S}_f \right| \right\} \end{aligned}$$

- Do the PISO loop $0 < \nu < \text{nCorrectors}$ times, where:

a) A face flux is calculated using the $H(\vec{v}_m)$ operator with \tilde{v}_m obtained in the momentum predictor. This face flux doesn't take into account the effects of the gravity

$$F_{mu}^{\nu+1}{}_f = \left(\frac{H(\tilde{v}_m)}{a_P} \right)_f \cdot \vec{S}_f + \text{ddtPhiCorr}(1/a_P, \rho_m, \vec{v}_m^\nu, F_m^\nu)$$

The flux is then adjusted to obey continuity, via `adjustPhi(F_mu^{\nu+1}, H(v_m^\nu)/a_P, p_rgh^\nu)` method

b) The final proposed flux is found adding the effects of the gravity

$$F_m^{\nu+1}{}_f = F_{mu}^{\nu+1}{}_f - (\vec{g} \cdot \vec{x})_f (\vec{\nabla} \rho_m^{n+1})_f \frac{|\vec{S}_f|}{(a_P)_f}$$

c) The pressure equation is assembled and solved `nNonOrthogonalCorrectors` times for $p_{rgh}^{\nu+1}$

$$\sum_f \left[\left(\frac{1}{a_P} \right)_f (\vec{\nabla} p_{rgh}^{\nu+1})_f \right] \cdot \vec{S}_f = \sum_f F_m^{\nu+1}{}_f - \sum_f \left[\alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \right]_f (\vec{v}_{qp})_f \cdot \vec{S}_f$$

d) The proposed flux is then adjusted by the effect of the pressure and then the center-of-volume velocity at the cell centers is adjusted as well

$$F_m^{\nu+1}{}_f = F_m^{\nu+1}{}_f - \left[\left(\frac{1}{a_P} \right)_f (\vec{\nabla} p_{rgh})_f \right] \cdot \vec{S}_f$$

$$\vec{v}_m^{\nu+1} = \vec{v}_m^{\nu+1} + \frac{1}{a_P} \mathcal{R} [(F_m^{\nu+1}{}_f - F_{mu}^{\nu+1}{}_f) (a_P)_f]$$

e) Finally the static pressure is reconstructed p from the modified p_{rgh} as $p = p_{rgh} + \rho_m \vec{g} \cdot \vec{x}$

- The face flux for the velocity of center-of-volume is recovered from the flux using Eqn. (6.4)

$$F^{L,n+1} = F_m^{n+1}{}_f - \left[\alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \right]_f (\vec{v}_{qp})_f \cdot \vec{S}_f$$

6.4 Results

6.4.1 1D semi-analytic tests

The first examples to be solved are those presented in section 3.3.2, corresponding to the semi-analytic solutions for sedimentation. The domain is meshed in 3D as is usual in OpenFOAM® giving an hexahedron with 400 elements in z and one in the other directions. The boundary conditions are similar in all examples. In the case of α_q , `zeroGradient` was set in top and bottom boundaries. The modified pressure p_{rgh} was set as `bouyantPressure` in top and bottom boundaries, in addition the pressure reference was set as 0 in the fourth cell from the bottom of the domain. Respect to the velocity it was set as `fixedValue` with $\vec{v}_m = (0, 0, 0)$ (no-slip) in top and bottom boundaries. The boundary conditions for the sides were `empty` for all the magnitudes.

The solver was set with the following parameters: `momentumPredictor=yes`, `nCorrectors=3`, `nNonOrthogonalCorrectors=1`, `nAlphaCorr=1`, `nAlphaSubCycles=1`. The time-step was set as $\Delta t = 0.001$ and then the problem was run using KT-MULES and TVD-MULES solvers. The results are shown in Figures 6.1-6.2 (compare to Figure 3.6 and Figures 3.8-3.10) and show excellent agreement between between KT-MULES and TVD-MULES solvers respect to the analytical solution in the case of α_p and p . The solution for the problem with initialization in $\alpha_p = 0.3$ respect to v_m (see Figure 6.2.b) give undershoots for both methods. This is caused due to the solution for the jump in α_p is not given within two cells which guarantees to have a bounded solution for the equation for v_m . Note that this problem is particularly important in the most important jump in densities.

The last 1D example corresponds to which was proposed by Gastaldo *et al.* (Gastaldo et al., 2011) based on the work of Coquel *et al.* (Coquel et al., 1997) as a test for FVM/FEM solver for the ASMM. It consists in a sedimentation problem like the previous ones using a constant relative velocity, v_{pq} . This assumption is clearly non-physical but leads to a simple solution that

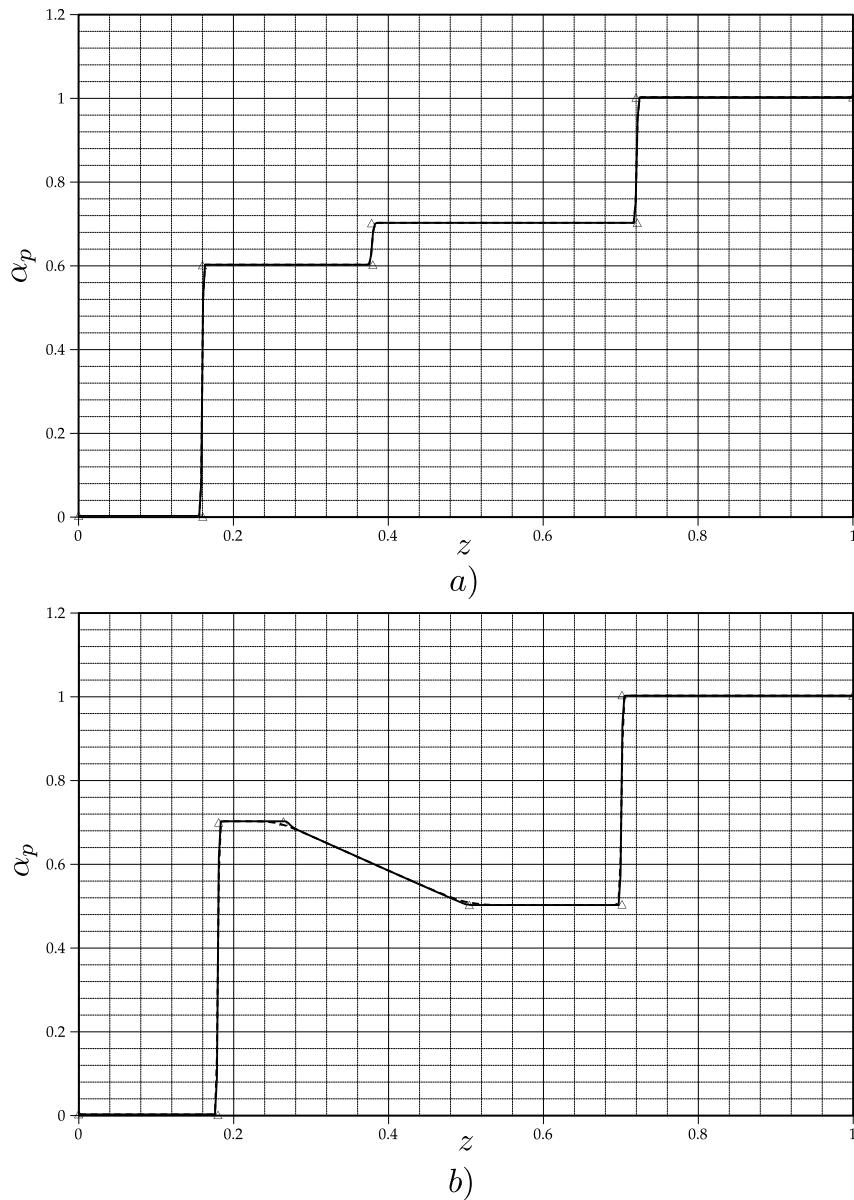


Figure 6.1: Solutions of α_p a) 0.6-0.7 initialization at $t = 0.4$, b) 0.7-0.5 initialization at $t = 0.6$. — KT-MULES solver, - - - TVD-MULES solver. \triangle notable points of analytical solution.

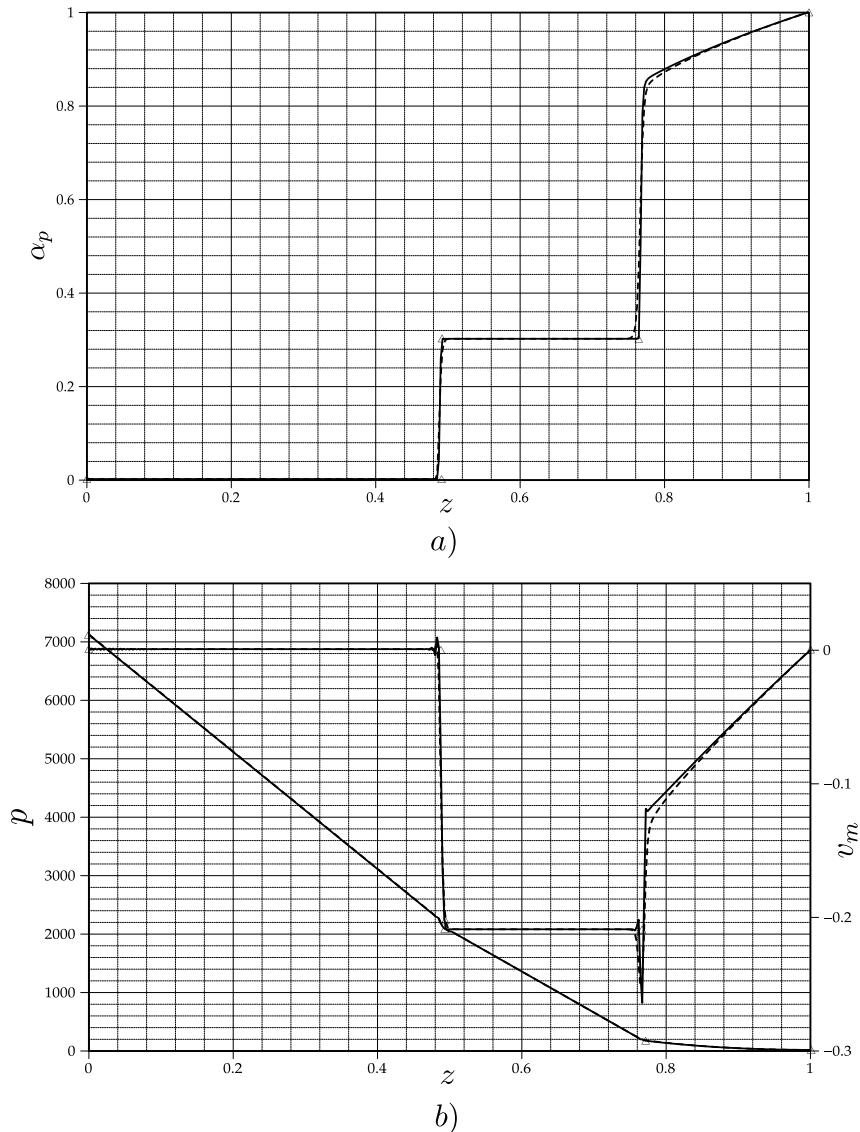


Figure 6.2: Solutions of α_p a) 0.3 initialization at $t = 1$. b) Pressure and center-of-mass velocity for a). — KT-MULES solver, - - - TVD-MULES solver. \triangle notable points of analytical solution.

qualitatively represents the original phenomenon. In this case the flow evolves in stiff conditions setting a good test for the stability of the proposed schemes (Evje and Flåtten, 2003).

The problem is set with $h = 7.5$, $\alpha_p = 0.5$, $v_m = 0$ and $p = 0$. The physical constants are $g = 9.81$, $\rho_l = 1000$ and $\rho_g = 1.2$, the viscosities are set to zero. The relative velocity has the value $v_{pq} = 1$. This selection for the relative velocity implies that in Eqn. (3.12) the constants have the values $v_{rc} = 1$ and $a = 0$. Finally the flux for the α_p results to be which as shown in Eqn. (6.9)

$$F(\alpha_p) = v_{rc} (\alpha_p - \alpha_p^2) \quad (6.9)$$

being the flux derivative given by Eqn. (6.10)

$$F'(\alpha_p) = v_{rc} (1 - 2\alpha_p) \quad (6.10)$$

allowing to determine the convexity or non-convexity of the flux and the eigenvalues. The graphs for both functions are shown in Figure 6.3. The shaded zone is the so-called *convex hull* which is used for wave analysis. From the graph it is possible to show that the flux function is convex. This leads to a solution having two shocks, one going from bottom to the top and the other one in the opposite direction. The convex hull is formed by two straight lines representing the two shocks. Using Eqns. (3.57)-(3.58) the front velocities are $a' = -0.5$ and $b' = 0.5$. The solutions for two different times are shown in Figure 6.4, where the triangles boxes correspond to the solution at time $t = 1$ and the triangles to the solution for $t = 5$. In this solutions only the characteristic points of the curves are given.

The numerical solutions for both KT-MULES and TVD-MULES were obtained with the same settings of the previous 1D tests changing only the domain extension and physical properties. The results are also shown in Figure 6.4. The thick lines correspond to the reference results and the thin lines to the proposed methods. As is shown in the figure the agreement of the proposed methods with the theoretical results is excellent improving the solution respect

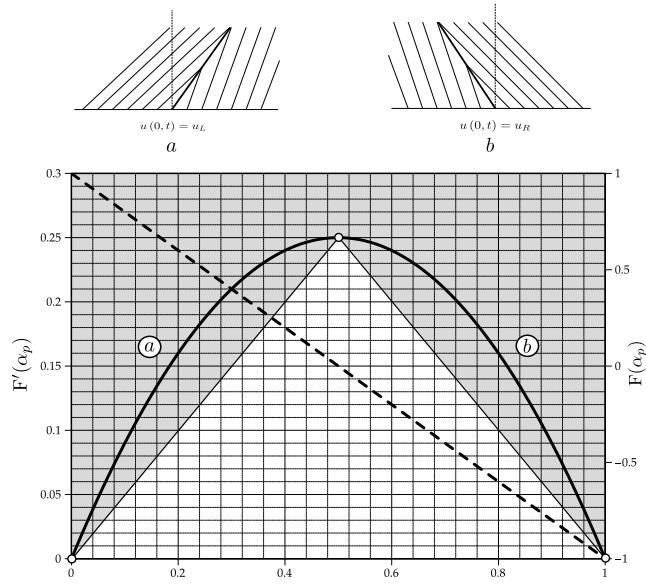


Figure 6.3: Riemann problem solutions for a convex flux. a. Right going shock, b. left going shock. — flux, - - - flux derivative.

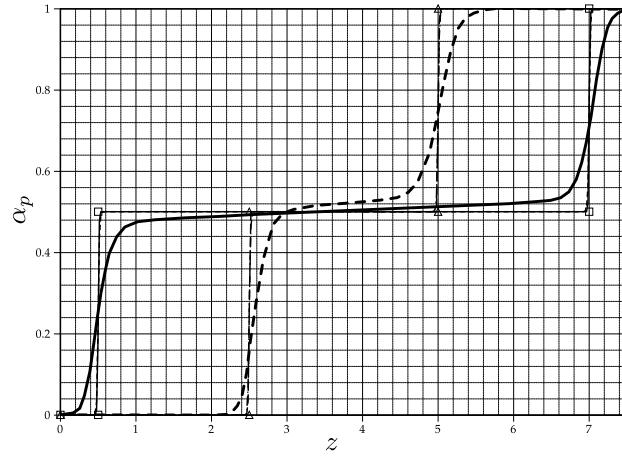


Figure 6.4: Riemann problem solutions for a convex flux with initial condition $\alpha_p = 0.5$. \square theoretical solution at $t = 1$, \triangle theoretical solution at $t = 5$, — reference solution at $t = 1$ and - - - reference solution at $t = 5$. Solutions for KT-MULES and TVD-MULES methods are given in thin lines.

to the reference. Here is important to note that one of the propositions for future work in the work of Gastaldo *et al.* is to present a HRS version of their ASMM solver. This objective is completely fulfilled here.

As a final remark from the 1D examples it is possible to assure from the figures that the hypothesis of neglecting the viscosity in the 1D formulation for the ASMM resulted to be correct. This is clear since the full 3D solvers resolve the shocks for the velocity in few cells not showing a diffusive pattern.

6.4.2 Bubble column reactor

This case corresponds to a chemical reactor where the fluid dynamics is controlled by the interaction of a bubble plume with the liquid within a cylindrical tank. The test is based on the original measurements by Computer Automated Radioactive Particle Tracking (CARPT) of Degaleesan (Dagaleesan, 1997) and later reported by Sanyal *et al.* (Sanyal et al., 1999) and Cartland Glover and Generalis (Cartland Glover and Generalis, 2004). The working principle of the reactor is based in the recirculation flow generated by the bubble plume [See Figure 6.5.a)]. The gas inlet is located at the bottom of the reactor which works as a pool; once the gas crosses the pool it is released by the top. During the gas is crossing the reactor it leads to a volume expansion of the original liquid volume changing the position of the free surface. The original dimensions of the reactor are: diameter $D = 0.19$, height $h \cong 1.2$. The gas inlet is fixed to $v_p = 0.02$ and the bubble diameter is $d = 5 \times 10^{-3}$. With respect to the fluids properties, they are $\nu_q = 1.005 \times 10^{-6}$, $\rho_q = 998.2$, $\nu_p = 1.460 \times 10^{-5}$, $\rho_q = 1.225$. The gravitational acceleration is $\vec{g} = (0, -9.81, 0)$. Under these conditions the Schiller-Naumann correlation can be fitted with a linear law for the relative velocity, \vec{v}_{pq} , with $a = 1$ and $\vec{v}_{rc} = (0, 0.4422, 0)$.

Following the references the selected domain is a shorter ($h = 1.045$) reactor in order to work without taking into account the free surface [see Figure 6.5.b)]. To do so, the top of the domain is considered as a degassing

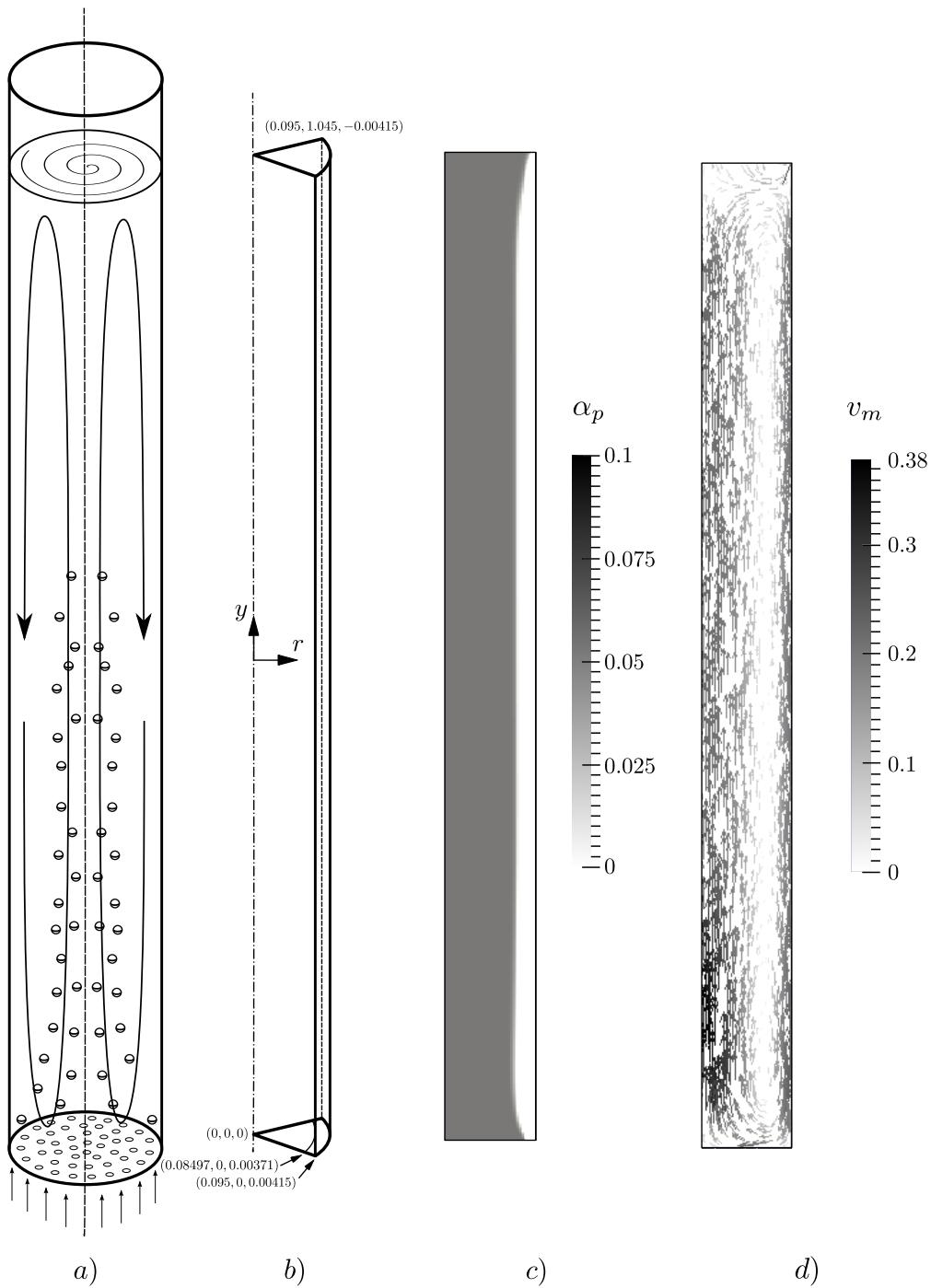


Figure 6.5: Geometry and solutions for a cylindrical bubble reactor. a) sketch of working principle, b) computational domain, c) gas hold-up, d) center of mass velocity vector field.

outlet. In addition, the considered entry area for the gas is selected as the 80% of the diameter giving a gas inlet velocity of $v_p = 0.03125$. The domain was considered axi-symmetric, therefore the selected mesh is basically 2D. The representation of such mesh in OpenFOAM® is achieved by a wedge mesh with only one element in the tangential direction. The final discretization had 44×600 elements (radius and height).

The problem was run using the ASMM with the standard $k - \epsilon$ model. This model allows a simple and effective turbulence closure and can be used without additional terms in case of bubbly flows (Buscaglia et al., 2002). Since the present solver was coded in the general viscous model of OpenFOAM® the use of turbulence models is transparent to the user giving the proper boundary and initial conditions. The flow was initialized with a laminar run, the boundary conditions were set as `zeroGradient` for α_q in the walls and the outlet and `fixedValue` with $\alpha_q = 0$ for the inlet. The velocity was set as `fixedValue` with $\vec{v}_m = (0, 0, 0)$ at the walls, `fixedValue` with $\vec{v}_m = (0, 0.03125, 0)$ at the inlet and `inletOutlet` at the outlet. This boundary condition implies `zeroGradient` if the flux is going outwards and `fixedValue` otherwise, this last value was set as $\vec{v}_m = (0, 0, 0)$. This boundary condition guarantees the possibility of volume expansion due to the gas inlet. The modified pressure was set as `zeroGradient` in the bottom wall and as `buoyantPressure` in the side wall and in the inlet. Finally, this pressure was set to zero at the outlet via a `fixedValue` boundary condition. The front and back walls were set as `wedge` in order to indicate the axi-symmetry.

The solver was set with the following parameters: `momentumPredictor=yes`, `nCorrectors=3`, `nNonOrthogonalCorrectors=1`, `nAlphaCorr=1`, `nAlphaSubCycles=2`. The time-step was adjusted automatically with a condition of $\mathbf{Co} = 0.5$ showing good stability. Once the laminar flow was developed the run continued using the $k - \epsilon$ model. The boundary conditions for k were `fixedValue` of $k = 9.3750 \times 10^{-6}$ at inlet and `zeroGradient` at the outlet, in the case of ϵ the same boundary conditions were used with $\epsilon = 1.9424 \times 10^{-7}$ at the inlet. The initial value for ν_t was estimated in $\nu_t = 4.0723 \times 10^{-5}$. This

turbulent parameters were calculated on the base of a intensity of turbulence of 10% and internal velocity $|\vec{v}_m| = 0.1$. The standard wall functions were used for the wall treatment.

The general results for gas-hold up and velocity of center-of-mass vector field are shown in Figures 6.5.b)-6.5.c). A detailed comparison is usually done based on profiles, particularly for velocity of center-of-mass and gas hold-up. The comparison is shown in Figure 6.6. From the figures it is clear that the recirculation within the reactor is properly predicted with good agreement with similar solvers respect to the velocity profile. The profile for α_p (gas hold up) has some differences with the reference solvers, resulting less diffusive; it can be attributed at the lack of turbulent diffusion term in α_q equation. It is worthy to note that these results are only indicative of the magnitude of gas hold-up since all the models have differences with the experimental results.

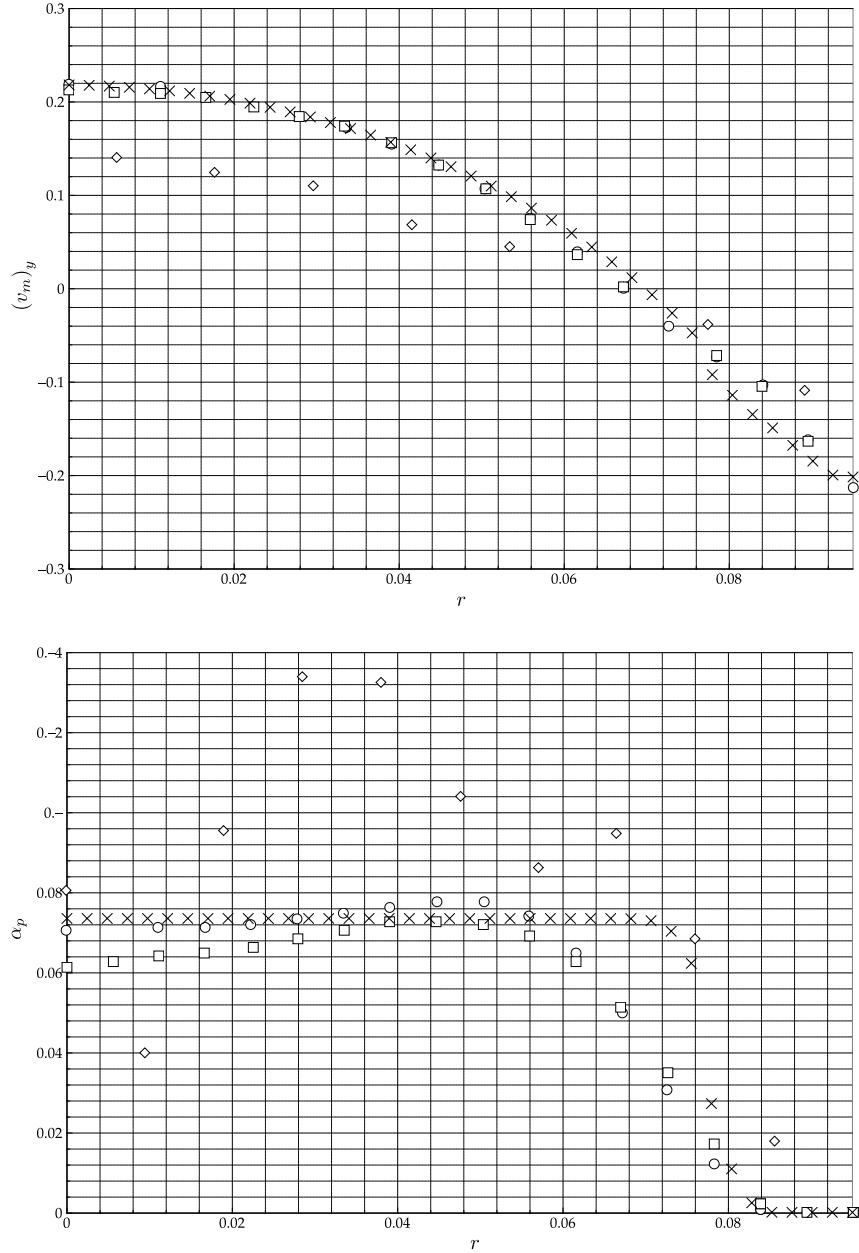


Figure 6.6: Velocity of center-of-mass and gas hold-up solutions as a function of the radius for the bubble column reactor ($y = 0.475$). \diamond CARPT, \circ ASMM (reference), \square Two fluids (reference), \times Present ASMM solver.

Chapter 7

An Extended Mixture Model

7.1 Introduction

The study of coupled models for the treatment of short and long scale interfaces has been in discussion in the last years motivated by the lack of solutions using the known models and the limited solutions given by the DNS. In order to give a basis for the present work is worthy to know the state of the art in the matter. Doing a bibliographic research the first work in the topic seems to be that was presented by Černe *et al.* (Černe et al., 2001). In this work the authors gave a coupled method between VOF and Two-Fluids. The switching parameter is given by an indicator function related with the free-surface reconstruction method; thus, there is a threshold value over which the interface is treated as having long scale and captured by VOF and the opposite case with the Multi-fluid (Two Fluids) model. Then, some test cases are solved comparing the convergence of a pure VOF solution against the coupled solver. Since each base model is written in its original formulation there is not an unified solution framework and it is necessary to switch between models with different number of equations. This issue has particular importance in the treatment of the velocities, since the VOF model has only one velocity field meanwhile the Two-Fluids model has one velocity per phase. The transition from two velocities to one is managed by the definition of the velocity of center-of-volume, in the oppo-

site case the same velocity is assigned to each phase, losing the interface friction.

The solution of multiple scale interface problems with an unified framework was eventually presented some years later. Thus, Masuda and Nagaoka (Masuda and Nagaoka, 2006) devised a coupled VOF/Multi-fluid method for the application in nozzle flows. This method recognizes four fluids, the original two fluids and two mixtures given by the first fluid as the dispersed phase and viceversa. The transition between the models is governed by the dispersion function proposed by Černe *et al.*. In the same line, Štrubelj and Tiselj (Štrubelj and Tiselj, 2011), gave an unified framework for the Level-Set and the Multi-fluid method. In this approach all the scales are solved by the Multi-fluid method and an additional interface tracking term is implemented within it. The detection of the scale interfaces is achieved by the cited dispersion function.

Another example of a model based on the unified VOF/Multi-fluid approach is that was given by Yan and Che (Yan and Che, 2010). It relies in a division of the phases by their physical state and the interface length scale giving three new phases, the liquid (phase 1), the large-length-scale-interface (LSI) with gas phase (phase 2) and the small-lengthscale-interface (SSI) with gas phase (phase 3). Therefore, a shared momentum equation is solved for the mixture of phase phase 1 and phase 2 and a second momentum equation is solved for phase 3. This second momentum equation gives the dynamics of the particulate phase. The geometry of the interfaces is captured by a VOF method for phase 1 and 2, phase 3 is also governed by a mass conservation equation but without interface capturing since it is considered a dispersed phase.

The concept of phase division by their physical state and length scale was also developed by Bohorquez (Bohorquez R. de M., 2008) for the treatment of air-water-sediments in hydraulics problems. Here, there are two principal phases, the air (phase 1) and the water-sediments mixture (phase 2), the third phase are the sediments which are dispersed within the water. Thus, a VOF/ASMM model is devised in an unified framework were the LSI between phase 1 and phase 2 is solved by the VOF model. The geometry of the

dispersed phase (phase 3) is solved by an additional mass conservation equation without interface capturing. Since the whole model is given in the ASMM framework only one momentum equation is solved for the air-water-sediments mixture.

From the study of the presented references is also noticeable that the field of coupled models for different interface length scales is still in development. The advances on the description of this kind of problems require the development and improvement of the models and the possibility of validation. The validation plays a crucial role requiring more experiments and getting analytical or semi-analytical solutions. In this context the objective of this chapter is to present a VOF/ASMM coupled model for two phase problems and its application for academic and industrial problems. The derivation of the model is motivated from the analysis of VOF solutions looking for a better treatment of the unresolved interface scales. On the other hand, some problems usually solved by the ASMM method can be also managed by the proposed extended model improving the predictive capabilities.

7.2 Motivation

The numerical motivation for the use of coupled models rises from the careful observation of the solutions obtained with the basic models. For the present work, where a VOF/ASMM coupling is proposed, the analysis starts by the VOF method which is able to capture of all the scales of the interface with the appropriate mesh. The concept of different length-scale in the interfaces was introduced in chapter 1 by the Figure 1.2. There, the mesh worked as a “filter” for the interface and the long and short scales can be determined. Now, the results given in chapter 5 for the VOF method could be re-examined to detect such interface scales and the behavior of this method. Thus, the result for $t = 5.4$ in Figure 5.1 for the Rayleigh-Taylor problem is recalled in 7.1.a), in addition the result for a mesh four times coarser is given in b) and then the solution is zoomed in c). Figure 7.2 shows a similar comparison starting from the solution for $t = 1.6$ in Figure 5.2 for the Dam Break problem, then a

solution in a four times coarser mesh and a zoomed area for the last solution are given.

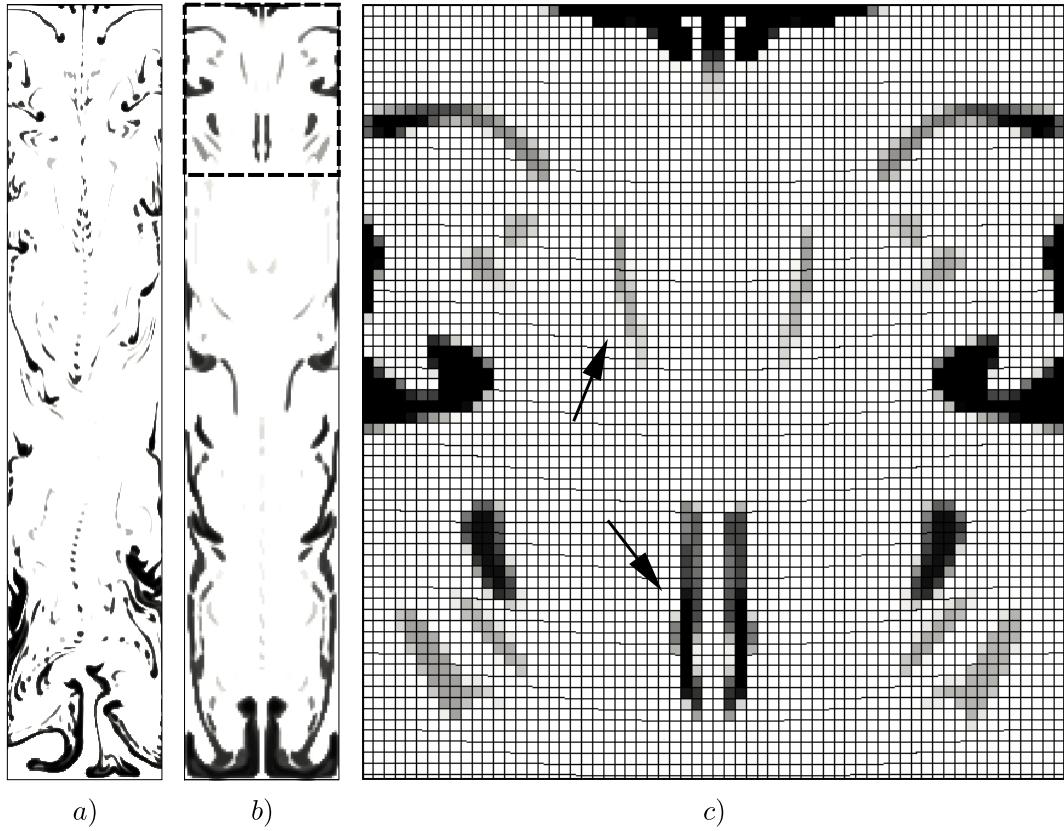


Figure 7.1: Results for the Rayleigh-Taylor problem for $t = 5.4$. a) results in fine mesh, b) results in coarse mesh, c) detail for the dashed line box en b) with superimposed mesh (grayscale saturated to black at $\alpha_q = 0.3$). Selected non-resolved chunks are indicated by the arrows.

The physics of these two problems for the selected times have particular differences. In the Rayleigh-Taylor case the problem is dominated by the falling of the more dense phase from top to bottom, once the initial structures have been lost the flow continues as falling droplets which accumulates at the bottom of the domain . In the case of the Dam Break problem after the initial column has collapsed some droplets are ejected from the splashing waves but

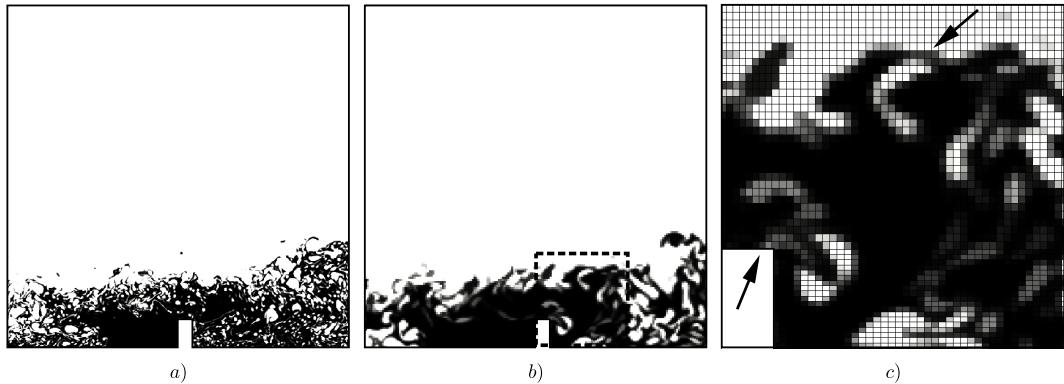


Figure 7.2: Results for the Dam Break problem for $t = 1.6$. a) results in fine mesh, b) results in coarse mesh, c) detail for the dashed line box en b) with superimposed mesh (grayscale saturated to black at $\alpha_q = 0.3$). Selected non-resolved chunks are indicated by the arrows.

in general the more important particulate physics is given by the less dense phase trapped in the mixing process. These particles behave as bubbles which rise by buoyancy forces.

Since the interface is resolved in about two or three cells (Trontin et al., 2008) by the VOF method, a chunk (small fluid structure, see the structures marked by the arrows in the figures) should have around four to six cells in width to be correctly resolved. In the case of bubbles and droplets they should have about ten cells in diameter for a correct curvature calculation. The practical effect is the wrong re-agrupation of these structures due to the surface tension terms which act with a numerical surface tension (Rider and Kothe, 1998). The development of unresolved interface scale chunks has another important drawback. When the values of α_q of these chunks lay in intermediate values between 0 and 1 (grey zones in the figures) these structures loose their physical meaning. The fluid present within them is then treated as a new fluid with density and viscosity given by the value of the mixture properties. These properties do not match the values of none of the original two fluids and then the buoyancy forces and the drop falling will be incorrectly calculated.

The idea behind the model which will be presented is to treat the long scale structures of the flux purely with the VOF method. When some structures fall below the unresolved scale this zones have to be calculated using the ASMM using appropriate closure laws.

7.3 Theoretical foundation

From previous chapters about VOF and ASMM it becomes clear that both models can be written in a very close formulation, and even more, the VOF model can be directly derived from the ASMM. The basic differences between these approaches in the context of the mixture models are the terms related to the different scales. Therefore, since the interfaces are supposed to be completely captured, the VOF model has a term in the momentum equation including the effects of the surface tension. On the other hand the ASMM does not have this term but includes the effect of the drift stresses, or the effect of the small scale interfaces. In addition, the relative velocity between phases has physical meaning in ASMM and in VOF it is only a numerical tool in order to compress the interface. Thus, using θ as a flag to activate or deactivate certain terms according to the interface scale which is being resolved, VOF (Weller-VOF) and ASMM can be coupled as in Eqn. (7.1)

$$\left\{ \begin{array}{l} \vec{\nabla} \cdot \vec{u} = 0 \\ \\ \frac{\partial}{\partial t} (\rho_m \vec{v}_m) + \vec{\nabla} \cdot (\rho_m \vec{v}_m \otimes \vec{v}_m) = -\vec{\nabla} p_{rgh} + \vec{\nabla} \cdot \left[\mu_m \left(\vec{\nabla} \vec{v}_m + \vec{\nabla} \vec{v}_m^T \right) \right] \\ \\ -\vec{g} \cdot \vec{x} \vec{\nabla} \rho_m + \theta \sigma \kappa \vec{\nabla} \alpha_q - (1 - \theta) \vec{\nabla} \cdot \left[\alpha_q (1 - \alpha_q) \frac{\rho_q \rho_p}{\rho_m} \vec{v}_{qp} \otimes \vec{v}_{qp} \right] \\ \\ \frac{\partial \alpha_q}{\partial t} + \vec{\nabla} \cdot (\alpha_q \vec{u}) + \vec{\nabla} \cdot \{ \alpha_q (1 - \alpha_q) [\theta \vec{v}_{qp, \text{VOF}} + (1 - \theta) \vec{v}_{qp, \text{ASMM}}] \} = 0 \end{array} \right. \quad (7.1)$$

where $\vec{v}_{qp, \text{VOF}}$ and $\vec{v}_{qp, \text{ASMM}}$ are the relative velocities calculated either numerically or physically based. The value of θ coefficient is $\theta = 1$ for VOF and

$\theta = 0$ for ASMM. Now it is necessary to devise a method to calculate this coefficient; to do so two methodologies will be studied in order to select the most appropriate for the needs.

7.3.1 Černe criterion

One of the available criteria for long and short scale models coupling, used by several authors, was given by Černe (Černe et al., 2001). It is based on the analysis of the frame obtained having into account a given cell and all its neighbours by faces and edges as is shown in Figure 7.3. The switching function γ is obtained by finding the minimum of the function G as is shown in Eqns. (7.2-7.3)

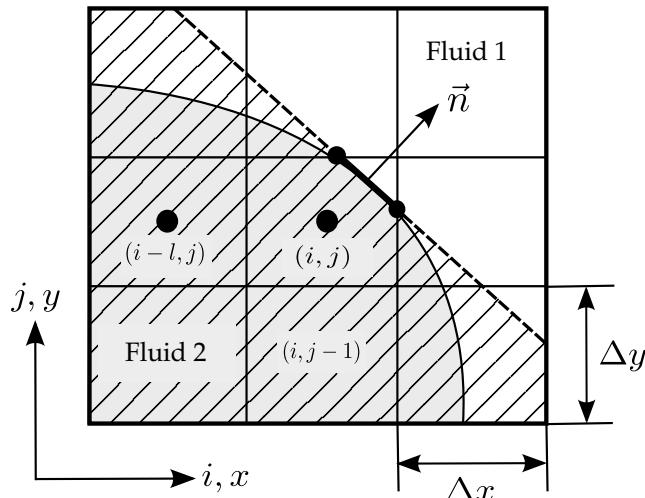


Figure 7.3: Interface reconstruction using the Černe criterion [adapted from (Černe et al., 2001)].

$$G_{i,j}(\vec{n}) = \sum_{l=-1}^1 \sum_{k=-1}^1 (\alpha_{q,i+k,j+l} - \alpha'_{q,i+k,j+l}(\vec{n}))^2 \quad (7.2)$$

$$\gamma_{i,j} = \min(G_{i,j}(\vec{n})) \quad (7.3)$$

where α'_q represents the volume fraction of the hatched area. The minimum value of γ is zero and corresponds to the exact matching of the interface with boundary of the hatched zone. This value increases when the fluid is located in the wrong side of the proposed interface, therefore this function is often called the "dispersion function". Finally, it is necessary to set a threshold value for $\gamma = \gamma_0$ such that the θ criterion could be calculated as in Eqn. (7.4)

$$\theta = \begin{cases} 1, & \text{if } \gamma_{i,j} < \gamma_0 \quad [\text{VOF in cell } (i,j)] \\ 0, & \text{if } \gamma_{i,j} > \gamma_0 \quad [\text{Multi - fluid in cell } (i,j)] \end{cases} \quad (7.4)$$

The threshold value is obtained by several cases study, the recommended values is $\gamma_0 \cong 0.6$. This methodology is attractive since it is based in the reconstruction of the interface (Puckett et al., 1997), but requires the time consuming solution of a minimization problem at each cell and time-step.

7.3.2 Face gradient criterion

Another criterion can be devised based in the gradient of the phase fraction function α_q . Recalling the gradient calculation from Eqn. (2.17) this operation can be done as is shown in Eqn. (7.5)

$$\vec{\nabla}\alpha = \frac{1}{V} \sum_f (\alpha_q)_f \cdot \vec{S}_f \quad (7.5)$$

as a next step this gradient is interpolated at faces obtaining $\vec{\nabla}\alpha_f$. The gradient at faces gives a general idea of the variation of the phase fraction along the domain. Large gradients are associated to big changes in α and then a large scale interface is inferred. This value is weighted with a measure of the mesh in order to normalize the switching function. A clear local jump between two phases requires not only a big gradient but also to be extended in few cells. Therefore, the gradient is multiplied by the face's neighbouring cells center-to-center vector, \vec{d}_{PN} as is shown in Eqns. (7.6)-(7.7) (see Figure 7.4)

$$\vec{d}_{PN} = \vec{x}_P - \vec{x}_N; \quad (7.6)$$

$$\gamma_f = |\vec{\nabla} \alpha_f \cdot \vec{d}_{PN}| \quad (7.7)$$

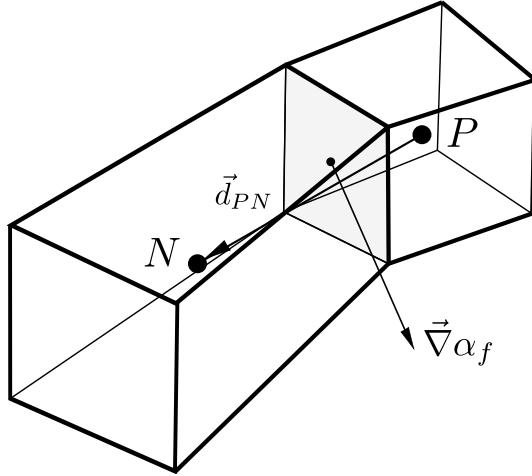


Figure 7.4: Face gradient criterion.

Finally the criterion to switch between VOF and ASMM is given by the rules expressed in Eqn. (7.8). It states that VOF will be used in high relative gradient zones (large scale interfaces) and in zones with α near to $\alpha = 1$ or $\alpha = 0$, indicating pure phases; in all other cases ASMM will be used. This criterion requires the selection of γ_0 , the threshold for small gradients and ϵ which is a magnitude that controls the maximum deviation from $\alpha = 1$ or $\alpha = 0$ to be even considered as a pure phase.

$$\theta_f = \begin{cases} 1, & \text{if } \alpha_f < 0 + \epsilon \text{ o } \alpha_f > 1 - \epsilon \quad [\text{VOF in face } f] \\ 1, & \text{if } \gamma_f > \gamma_0 \quad [\text{VOF in face } f] \\ 0, & \text{if } \gamma_f < \gamma_0 \quad [\text{ASMM in face } f] \end{cases} \quad (7.8)$$

The values for γ_0 and ϵ have to be adjusted according to the problem. Since in VOF the interface is resolved in about three cells a reference value is $\gamma_0 = 0.33$. Respect to ϵ , a typical value is $\epsilon = 5 \times 10^{-3}$. The principal advantages of this indicator function are its intrinsic 3D formulation and its simplicity and low demanding calculation. This method will be then used for model switching.

7.4 Solver implementation

The implementation of the solver for the coupled model or extended mixture model is derived from the solution algorithm of the ASMM solver (Algorithm 5). To this end, it is necessary to add the calculation of the θ indicator function and the extra terms given in Eqn. (7.1). The solver is given in Algorithm 6. Since the presented algorithm is not available in the OpenFOAM® suite it was programmed starting from ASMM solver and using the gdbOF debugging tools [see Appendix A or (Márquez Damián et al., 2012)].

7.5 Examples

In order to validate the proposed method a series of examples will be solved. The first example corresponds to a bubble plume in laboratory conditions and the objective is to correctly predict the deformation of the free-surface due to the plume and the dynamics of the dispersed phase. This kind of problems is often solved using the Mixture Model but the prediction of the free-surface dynamics is not possible since it would be completely smeared. The shape of the free-surface is compared to laboratory experiments and the dynamics of the dispersed phase is qualitatively analyzed. The second example corresponds to the bubble column reactor presented in the ASMM chapter; here the gas hold-up and axial velocity are compared again with the CARP experiments but the dynamics of the free surface is also calculated. This allows to free the top boundary condition needed to allow the volume expansion, since the free surface is included within the domain. To help in the analysis a one dimensional model is proposed able to predict the gas hold-up and the free surface position. These are original examples which give a contribution to the general discussion of the coupled models.

The last two examples have been presented in the literature in the past, nevertheless a new insight of them is given proposing new ways to evaluate the convergence to the DNS solution. Therefore, the Dam Break test is revisited, comparing the overall behaviour of the solutions for both VOF and extended

Algorithm 6 Pressure based extended mixture solver

1. Solve the mass conservation equation for the primary phase for α_q^{n+1} , assemble the mass face flux $F_{\rho m}^{n+1}$ and get the new mixture density ρ_m^{n+1} by a loop of `nAlphaSubCycles` cycles where:

a) The relative velocity at faces is calculated as a flux from a combination of Eqn. (3.25) and Eqn. (5.11) controlled by the indicator function θ_f

$$F^{NL,n} = (1 - \theta_f) \vec{v}_{rc} (\alpha_q^n)^a \cdot \vec{S}_f + \theta_f n_f \min \left[C_\alpha \frac{|\Phi^{L,n}|}{|\vec{S}_f|}, \max \left(\frac{|\Phi^{L,n}|}{|\vec{S}_f|} \right) \right]$$

b) The α_q equation [Eqn. (5.10)] is solved $0 < \nu < \text{nAlphaCorrectors}$ times by the MULES integrator which also returns the limited flux F_{α_q} . This flux is computed using TVD reconstruction.

$$\frac{\alpha_q^{n,\nu+1} - \alpha_q^n}{\Delta t} V + \sum_f \left\{ (\alpha_q^{n,\nu})_f F^{L,n} + [\alpha_q^{n,\nu} (1 - \alpha_q^{n,\nu})]_f F^{NL,n,\nu} \right\} = 0$$

c) The new mass face flux for the present sub-cycle is calculated by Eqn. (5.20). At the end of the loop the final mass face flux is calculated by Eqn. (5.21) and the density is updated

$$F_{\rho m, sc, i} = F_{\alpha_q} (\rho_q - \rho_p) + F_{\rho p} \quad F_{\rho m} = \sum_{i=1}^{n_{sc}} \frac{\Delta t_{sc}}{\Delta t} F_{\rho m, sc, i} \quad \rho_m^{n+1} = \alpha_q \rho_q + (1 - \alpha_q) \rho_p$$

2. Solve the momentum predictor [discretized version of the second equation in Eqn. (7.1)] for $\tilde{\vec{v}}_m$ if the `momentumPredictor` flag is set to `yes`

$$\begin{aligned} \frac{\rho_m^{n+1} \tilde{\vec{v}}_m - \rho_m^n \vec{v}_m^n}{\Delta t} V + \sum_f F_{\rho m}^{n+1} \tilde{\vec{v}}_m \cdot \vec{S}_f &= \sum_f (\mu_m^{n+1})_f (\vec{\nabla} \tilde{\vec{v}}_m)_f \cdot \vec{S}_f \\ &+ (\vec{\nabla} \vec{v}_m^n \cdot \vec{\nabla} \mu_m^{n+1}) V - \sum_f (1 - \theta_f) \left[\alpha_q^{n+1} (1 - \alpha_q^{n+1}) \frac{\rho_q \rho_p}{\rho_m^{n+1}} \vec{v}_{qp}^n \otimes \vec{v}_{qp}^n \right]_f \cdot \vec{S}_f \\ &+ \mathcal{R} \left\{ \theta_f (\sigma \kappa)_f (\vec{\nabla} \alpha_q^{n+1})_f - \left[(\vec{g} \cdot \vec{x})_f (\vec{\nabla} \rho_m^{n+1})_f - (\vec{\nabla} p_{rgh}^n)_f \right] |\vec{S}_f| \right\} \end{aligned}$$

3. Do the PISO loop $0 < \nu < \text{nCorrectors}$ times, where:

a) A face flux is calculated using the $H(\vec{v}_m)$ operator with $\tilde{\vec{v}}_m$ obtained in the momentum predictor. This face flux does not take into account the effects of the gravity

$$F_{mu}^{\nu+1} = \left(\frac{H(\tilde{\vec{v}})}{a_P} \right)_f \cdot \vec{S}_f + \text{ddtPhiCorr}(1/a_P, \rho_m, \vec{v}_m^\nu, F_m^\nu)$$

The flux is then adjusted to obey continuity, via `adjustPhi($F_{mu}^{\nu+1}, H(\vec{v}_m^\nu)/a_P, p_{rgh}^\nu$)` method

b) The final proposed flux is found adding the effects of the gravity and the surface tension

$$F_m^{\nu+1} = F_{mu}^{\nu+1} - (\vec{g} \cdot \vec{x})_f (\vec{\nabla} \rho_m^{n+1})_f \frac{|\vec{S}_f|}{(a_P)_f}$$

c) The pressure equation is assembled and solved `nNonOrthogonalCorrectors` times for $p_{rgh}^{\nu+1}$

$$\sum_f \left[\left(\frac{1}{a_P} \right)_f (\vec{\nabla} p_{rgh}^{\nu+1})_f \right] \cdot \vec{S}_f = \sum_f F_m^{\nu+1} - \sum_f \left[\alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \right]_f (\vec{v}_{qp})_f \cdot \vec{S}_f$$

d) The proposed flux is then adjusted by the effect of the pressure and then the center-of-volume velocity at the cell centers is adjusted as well

$$F_m^{\nu+1} = F_m^{\nu+1} - \left[\left(\frac{1}{a_P} \right)_f (\vec{\nabla} p_{rgh})_f \right] \cdot \vec{S}_f$$

$$\vec{v}_m^{\nu+1} = \vec{v}_m^{\nu+1} + \frac{1}{a_P} \mathcal{R} \left[(F_m^{\nu+1} - F_{mu}^{\nu+1}) (a_P)_f \right]$$

e) Finally the static pressure is reconstructed p from the modified p_{rgh} as $p = p_{rgh} + \rho_m \vec{g} \cdot \vec{x}$

4. The face flux for the velocity of center-of-volume is recovered from the flux using Eqn. (6.4)

$$F^{L,n+1} = F_m^{n+1} - \left[\alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \right]_f (\vec{v}_{qp})_f \cdot \vec{S}_f$$

methods. In addition, a measure of the correct dynamics of the dispersed phase is given which allows to estimate an improvement factor of the extended model respect to the VOF model. Finally, the Rayleigh-Taylor instability, which was presented in the VOF chapter together with the Dam Break test, is newly calculated using the extended model. A qualitative comparison of the solutions is made and the dynamics of the dispersed phase in each case is compared using an integrated measure.

7.5.1 Interaction of a bubble plume with the water surface

The first example gives a semi-quantitative validation from the phenomenon of interaction of a bubble plume and the water surface. This phenomenon appears in blowouts in offshore drilling, broken gas pipelines and natural undersea gas releases forming big bubble plumes. In addition bubble plumes of small extension are used for mixing process in reservoirs or waste water treatment, chemical reactors and metallurgical processes [see (Friedl and Fannelp, 2000) and references, (Zanotti, 2007)].

The example is taken from the cited work of Friedl and Fannelp and consists on the generation of a bubble plume in laboratory conditions released from the bottom of a water pool as is shown in Figure 7.5. The pool has a square cross-section of 1 of side and 0.95 in height. The free surface is set at $H_v = 0.66$. The air is released from the bottom of the tank through a square duct of area $A_i = 0.0005067$ and length of $h_i = 0.04502$ with a release velocity of $v_i = 2.6$ which corresponds to the case a4 of the reference. The physical properties of the fluids are $\rho_q = 1000$, $\nu_q = 1 \times 10^{-6}$, $\rho_p = 1$, $\nu_q = 1.48e - 05 \times 10^{-5}$ and $\sigma = 0.07$. The gravitational acceleration is $\vec{g} = (0, 0, -9.81)$.

As is expected, when the bubble plume reaches the free surface it is disturbed forming a fountain with different shapes but having in common a greater disturbance near the center of the fountain and then decaying to the sides of

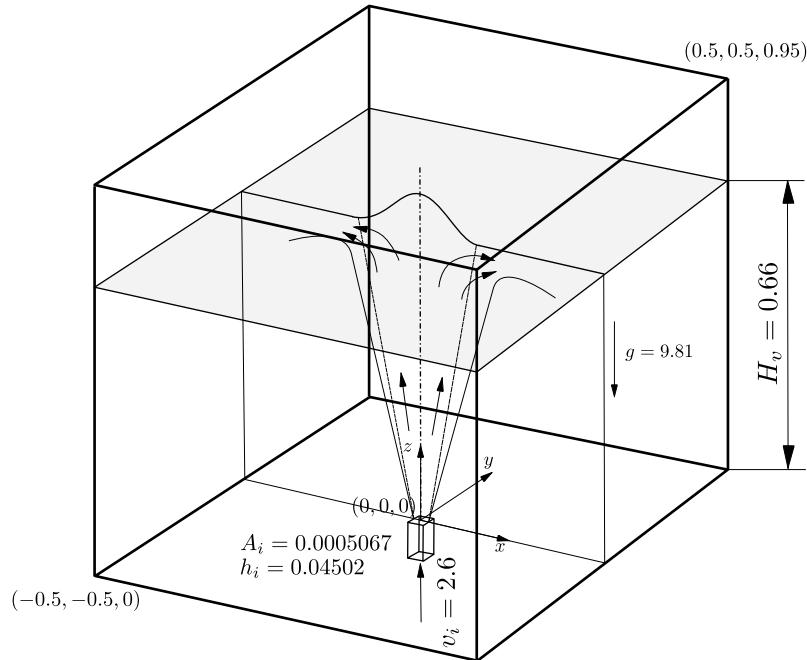


Figure 7.5: Geometry for the a4 case of bubble plume in (Friedl and Fannelløp, 2000). Where A_i and h_i are the transversal area and height of the inlet duct. The shaded zone indicates the original free surface position and the bell-shaped curve the mean free surface for the $x - z$ plane.

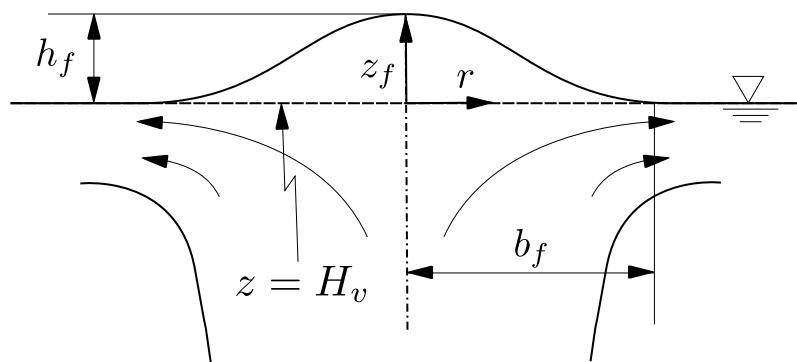


Figure 7.6: Mean shape of the fountain presented in Figure 7.5 [adapted from (Friedl and Fannelløp, 2000)].

the pool, as is shown Figure 7.5. The results reported in the reference are a mean of the experimental results which can be fitted by a bell-shaped function as in Figure 7.6. The expression of the mean free surface shape on a given vertical plane ($x - z$ in the figure) is presented in Eqn. (7.9)

$$h(r) = h_f e^{-r^2/b_f^2} \quad (7.9)$$

where h is the height of the mean free surface for a given radius r (the fountain is considered to be circular), h_f is the maximum height of the fountain and b_f is the semi-diameter of the fountain.

Thus, two series of simulations were done, the first one with the standard VOF solver (`interFoam`) and the second one with the extended model. Each series had three cases with similar settings but with different size hexahedral meshes: a) meshed with `blockMesh` with 364,000 elements; b) meshed in Gambit® with 574,975 elements but using more advanced local refining techniques for a better capturing of the plume and the free surface; c) resulting of the subdivision by two in the three directions of the previous mesh using the tool `refineMesh`, giving a mesh with 4,599,800 elements and the same refinement properties of the original mesh.

Respect to the boundary conditions for α_q , `zeroGradient` was set in all walls except for the top and the inlet. On the top the `inletOutlet` boundary condition was set using a fixed value of zero if the flux is ingoing. At the inlet the value of α_q was fixed as $\alpha_q = 0$ (`fixedValue`). Respect to the boundary conditions for the modified pressure it was set as `buoyantPressure` for all the walls and the inlet. At the top the `totalPressure` boundary condition was set with a value of zero. Finally, for the center-of-volume velocity the non-slip boundary condition was set for all walls except for the top and the inlet. At the top the `pressureInletOutletVelocity` was set. The inlet was set with a `fixedValue` boundary condition with $\vec{u} = (0, 0, 2.6)$. The relative velocity law needed by the zones solved with ASMM in the coupled model was set with $a = 0$ giving a constant velocity. The value reported in the reference is

$v_{qp} = 0.35$. The parameters for model coupling were set as $\gamma_0 = 0.025$ and $\epsilon = 5 \times 10^{-3}$.

The solver was set with the following parameters: `momentumPredictor=no`, `nCorrectors=3`, `nNonOrthogonalCorrectors=0`, `nAlphaCorr=1`, `nAlphaSubCycles=2`, `cAlpha=1`. The maximum Courant number for this running was set as 0.5 which kept the stability even though it is greater than the recommended.

The first case (VOF with coarse mesh) was run until $t = 10$ in order to reach the full development of the bubble plume, then it was run to $t = 20$. The left runnings were done mapping the coarse mesh solution for $t = 10$ into the finer meshes and then running to $t = 20$. The calculation of the mean values reported for the experiments were done always with $t = 10 - 20$ runnings.

The general results are reported in Figure 7.7 for VOF and in Figure 7.8 for the extended model. From the first figure it is clear that the coarse mesh captures few details of the surface mesh, in addition the pool has non-physical chunks spread at the sides of the plume (the grayscale has been saturated to white at $\alpha_q = 0.8$ in order to easily see the gas zones). The refinement of the mesh in b) and c) improves the surface capturing and at the same time the VOF method increases its ability to capture the break-up of the big bubbles. The break-up gives small chunks and bubbles which are not correctly removed by buoyancy and then stay in the pool advected by the lateral flow from the plume to the sides. It is expected that successive mesh refinements could finally capture the fine bubbles dynamics as will be presented later for the Dam Break case. The observation of the pictures from the experimental work confirm that the bubbles concentrate in the plume and there is not recirculation [see Figure 3.3 in (Friedl, 1998)]. On the other hand the second figure represents the solution for the three meshes with the extended model; the effect of the mesh refinement is clear again. At the same time the fragmented chunks are properly removed by the activation of the ASMM giving a clear plume and keeping the pool free of zones without physical meaning.

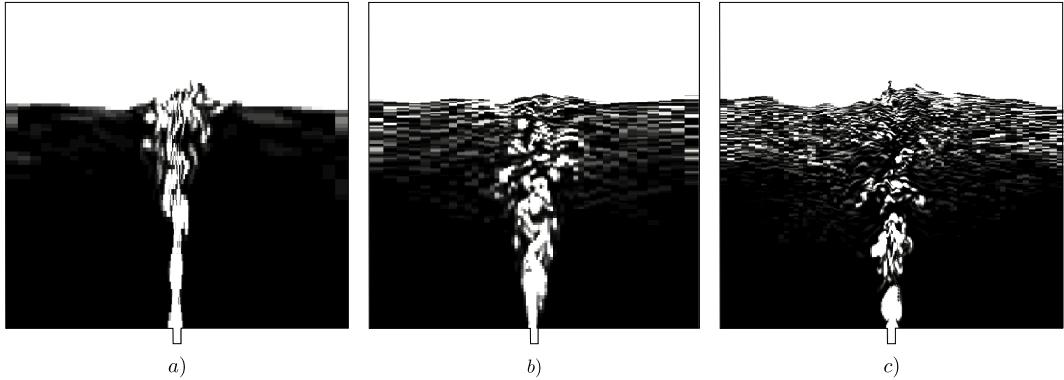


Figure 7.7: Solution for the bubble plume with VOF for three meshes, a) coarse, b) fine, c) finest. The grayscale is saturated to white at $\alpha_q = 0.8$.

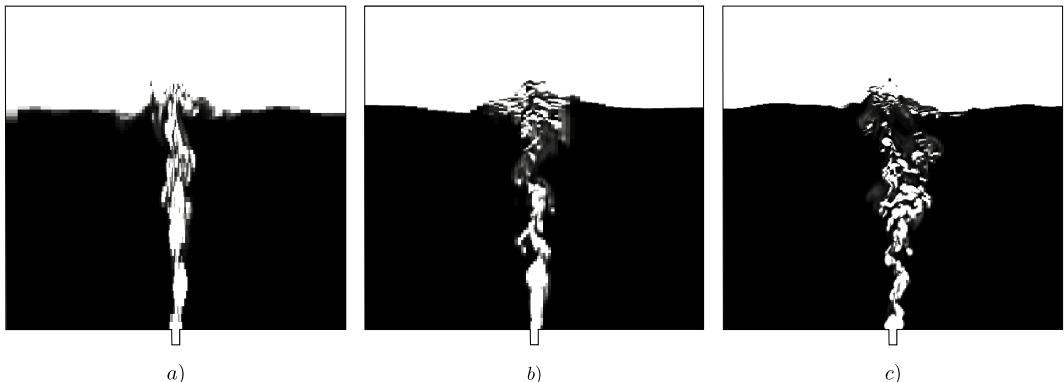


Figure 7.8: Solution for the bubble plume with the extended model for three meshes, a) coarse, b) fine, c) finest. The grayscale is saturated to white at $\alpha_q = 0.8$.

In addition to this qualitative analysis the shape of the free surface can be compared to the expression given in Eqn. (7.9). To do so, the void fraction α_q is sampled in 200 points in $0.6 \leq z \leq 0.8$ on 99 equi-spaced vertical lines in $-0.49 \leq x \leq 0.49$ on the $x - z$ plane. Then, the transition from zero to one (gas to liquid) is detected giving the position of the free surface (strictly speaking it could capture some droplets from wave splashing, this effect is supposed to be

non-determinant). This sampling is done with $\Delta t = 0.05$ and then the mean for z values is obtained for each point in x ; finally, the values are referred to the experiments as $\tilde{r} = x_f/b_f$ and $\tilde{z}_f = (\tilde{z}_f - h_{\text{offset}})/h_f$, where h_{offset} allows the adjustment of the offset of profiles with respect to the quiescent water level. The same is done for the bell-shaped function as $\tilde{r} = r/b_f$ and $\tilde{z}_f = h(r)/h_f$. The values for h_f and b_f are, $h_f = 0.038$ and $b_f = 0.101$. The results are presented in Figure 7.9 and show the positive effect of the refinement. The reconstruction of the free surface is similar in both of the models showing that the extended method retains the surface capturing capabilities of the VOF model. In addition if the solutions are compared to the experimental fitted curve it is clear that the fountain width is underestimated in both of the models. This effect is attributable to the lack of turbulent dispersion modeling. Here is important to note that the VOF model includes the effects of the turbulence only in the momentum equation in order to model the non-resolved scales of eddies. The scales of the interface are supposed to be captured by the mesh, therefore no diffusion term is added in the conservation equation of the void fraction. The extended model offers the possibility to add the turbulence term in the momentum equation and also to include the effect of turbulent dispersion in the non-resolved scales of the interface. It is clear that the diffusive terms for the void fraction have to be activated only in regions where the scales cannot be resolved by VOF (using the indicator function θ).

7.5.2 Bubble reactor with free surface capturing

The dynamic effects of a bubble plume can be used to obtain a better interaction between reagents; this effect is used in the bubble reactors. This kind of systems was studied in section 6.4 by means of the ASMM in a fixed domain shorter than the real geometry. This simplification was done since the real top boundary is movable due to the added volume given by the inlet of gas. Thus, the domain is set with the unexpanded volume and the fluid is freed to expand at the top by a `inletOutlet` boundary for the velocity. In view of the results obtained for the bubble plume it would be valuable to apply the extended model to the reactor case and

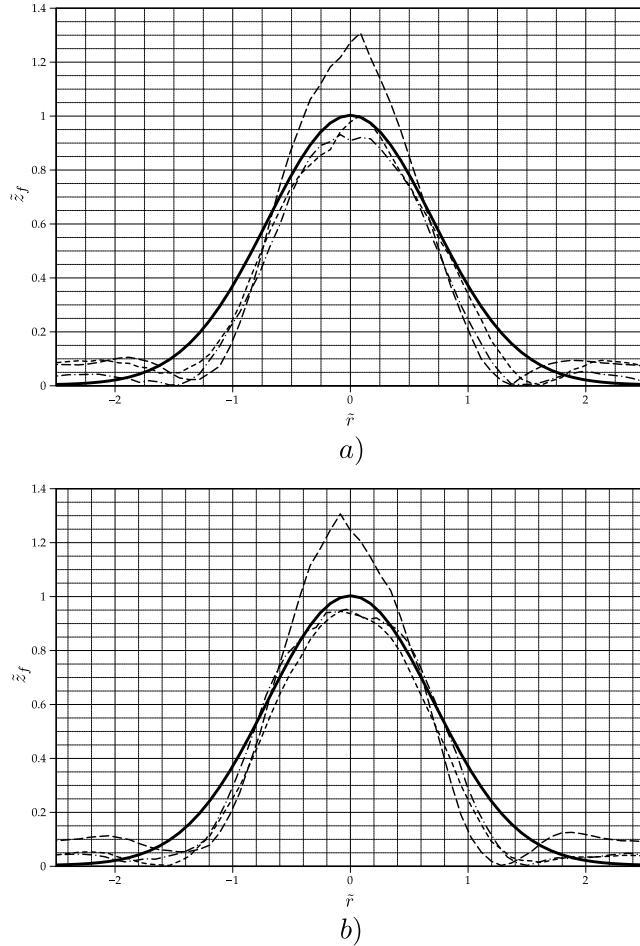


Figure 7.9: Mean surfaces for the bubble plume with a) VOF, b) Extended Model. — free surface theoretical model; - - - coarse mesh; - - - - fine mesh; - - - - - finest mesh.

check the solution, looking for a better treatment of the top boundary condition.

To start the analysis, a simplified model will be solved, given by a one dimensional problem of a column reactor with bottom inlet. As follows from Figure 7.10.a) the z coordinate corresponds to the axis of the reactor, h_0 is the original unexpanded height of the reactor filled with $\alpha_q = 1$ and h_T the total height of the domain. In addition $v_{p,i}$ and $\alpha_{p,i} = 1$ are the velocity and the secondary phase fraction of the inlet and $v_{p,a}$, $\alpha_{p,a}$ $v_{q,a}$ and $\alpha_{q,a}$ the same magnitudes for the secondary and primary phase in an arbitrary point a . The

entered volume of phase p crosses the volume of phase q due to the relative velocity v_{pq} until reaching the free surface. This transition is not instantaneous (for a finite v_{pq}), therefore a new volume is hold within the reactor added to the original volume of phase q . In this scenario it is possible to find the final free-surface position for a given inlet velocity and phase fraction. Then, the analysis starts with the incompressibility restriction [first equation in Eqn. (7.1)] in one dimension as is shown in Eqn. (7.10)

$$\frac{\partial}{\partial z} u = 0 \quad (7.10)$$

where u is the velocity of center-of-volume in one dimension. This restriction implies that u is constant in all the domain. From the definition of this velocity [Eqn. (3.26)] and using it at the inlet and at the point a the relation shown in Eqn. (7.11) is found.

$$v_{p,i} \alpha_{p,i} = v_{p,a} \alpha_{p,a} + v_{q,a} \alpha_{q,a} = v_{q,a} + \alpha_{p,a} v_{pq,a} \quad (7.11)$$

On the other hand, if the mass conservation equation from the primary phase [third equation in Eqn. (7.1)] is taken in the steady form as is shown in Eqn. (7.12)

$$\frac{\partial}{\partial z} u \alpha_q + \frac{\partial}{\partial z} \alpha_q (1 - \alpha_q) v_{qp} = 0 \quad (7.12)$$

this implies that $u \alpha_q + \alpha_q (1 - \alpha_q) v_{qp}$ is a constant in all the domain. Since $\alpha_{p,i} = 1$ and $\alpha_{q,i} = 0$ this constant is zero. After some algebraic manipulations, this expression reduces to $v_p \alpha_p + v_q (1 - \alpha_p) + \alpha_p v_{qp} = 0$, which in expanded form reads as in Eqn. (7.13) for the point a .

$$v_{p,a} \alpha_{p,a} - v_{q,a} \alpha_{q,a} = v_{q,a} + \alpha_{p,a} v_{pq,a} \quad (7.13)$$

Since the r.h.s. of Eqn. (7.11) and Eqn. (7.13) are equal the relationship given in Eqn. (7.14) is obtained

$$\alpha_{p,a} v_{pq,a} = \alpha_{p,a} v_{rc} (1 - \alpha_{p,a})^a = v_{p,i} \alpha_{p,i} \quad (7.14)$$

Taking $a = 1$, as is usual for bubble relative velocity laws the obtained expression is given by Eqn. (7.15)

$$\alpha_{p,a}^2 v_{rc} - \alpha_{p,a} v_{rc} + v_{p,i} \alpha_{p,i} = 0 \quad (7.15)$$

solving this quadratic equation for $\alpha_{p,a}$ it gives which is shown in Eqn. (7.16)

$$\alpha_{p,a} = \frac{1}{2} \left[1 \pm \sqrt{1 - \frac{4 v_{p,i} \alpha_{p,i}}{v_{rc}}} \right] \quad (7.16)$$

with the condition of $v_{rc} \geq 4 v_{p,i} \alpha_{p,i}$. Now, it is possible to find the value α_p within the reactor for a pure p phase at the inlet with the inlet velocity $v_{p,i}$ and v_{rc} as the constant relative velocity. Here it is important to note that the valid results of this equation are in the $[0,1]$ interval. Then, recalling the complete primary phase conservation equation for this case, as it is shown in Eqn. (7.17)

$$\frac{\partial \alpha_q}{\partial t} + \frac{\partial}{\partial z} u_i \alpha_q - \frac{\partial}{\partial z} \alpha_q (1 - \alpha_q) v_{rc} \alpha_q = 0 \quad (7.17)$$

it is clear that this is an hyperbolic equation with flux $F = u_i \alpha_q - \alpha_q (1 - \alpha_q) v_{rc} \alpha_q$ where u_i is the velocity of center-of-volume at the inlet and is equal to $v_{p,i}$ since $\alpha_{p,i} = 1$. The derivative of this flux is given by Eqn. (7.18)

$$\frac{\partial F}{\partial \alpha_q} = F' = u_i - v_{rc} \alpha_q (2 - 3\alpha_q) \quad (7.18)$$

The obtained expression can be used to predict the transitory solution of the hyperbolic problem and the final state. Taking $v_{p,i} = 0.03125$ and $v_{rc} = 0.4422$ it implies by Eqn. (7.16) that the immediate value of the phase fraction will be $\alpha_{p,a} = 0.07853$ or, in terms of the primary phase, $\alpha_{q,a} = 0.92147$. This value contrasts with the phase fraction which fills the tank, it is $\alpha_{q,a} = 1$ and a Riemann problem is given with $\alpha_{qL} = 0.07853$ and $\alpha_{qR} = 1$. The local velocities given by the flux function derivative [(Eqn. (7.18)] are: $F'_L = 0.343$ and $F'_R = 0.47345$ [see Figure 7.10.b)], since the local velocities are diverging and both are positive the predicted wave is a right-going rarefaction. On the other hand the free surface presents the state $\alpha_{qL} = 1$ and $\alpha_{qR} = 0$, with local velocities $F'_L = 0.47345$ and $F'_R = 0.03125$, since the left velocity is greater

than right a right-going shock is formed. When the rarefaction wave reaches the free surface the final state is developed and volume reaches its maximum value. The solution for $t = 1$ and the steady state are shown in Figure 7.10.c).

The volume phase p which is hold constantly within the pool leads to an expansion respect to the original volume of the reactor, changing the position of the free surface. Taking an unitary cross section the final volume can be calculated as in Eqn. (7.19)

$$h_F = \frac{h_0}{\alpha_{q,a}} \quad (7.19)$$

In the present problem, where $h_0 = 1.045$, it gives $h_F = 1.045/0.92147 \cong 1.134$. This value can be compared with the position of the free surface in Figure 7.10.c) given by a mixture solver and good agreement is found. The boundary and solver settings are similar to the 1D cases presented in section 6.4.1 except for the velocity which was set as `fixedValue` $v_m = (0, 0, 0.03125)$, since the inlet has a pure phase $\vec{u} = \vec{v}_m$, at the bottom inlet and `zeroGradient` at the outlet and for α_q which was set as $\alpha_q = 0$ at the inlet.

From this example is possible to see clearly how the reactor changes its volume due to the injection of volumetric flux by the inlet. In addition the model used for the numerical solution was able to capture eventual rarefactions and shocks, the rarefaction is associated to the plume front and the shock represents the free-surface. At first glance using only the mixture model seems to resolve the jumps properly, compressing the interface. This is true only for interfaces which are perpendicular to the relative velocity, but not in general cases. So, in these cases, the numerical compression proposed in section 5.3 is preferred, since it is perpendicular to the interface whatever its direction.

Having these insights about how the reactor works when the free-surface is modeled, the original reactor's problem is run again using the complete geometry height of $h = 1.2$ and initialized with the dense fluid (liquid) until $h_{L0} = 1.045$, as is shown in Figures 7.11.a-c). The mesh has 44×690

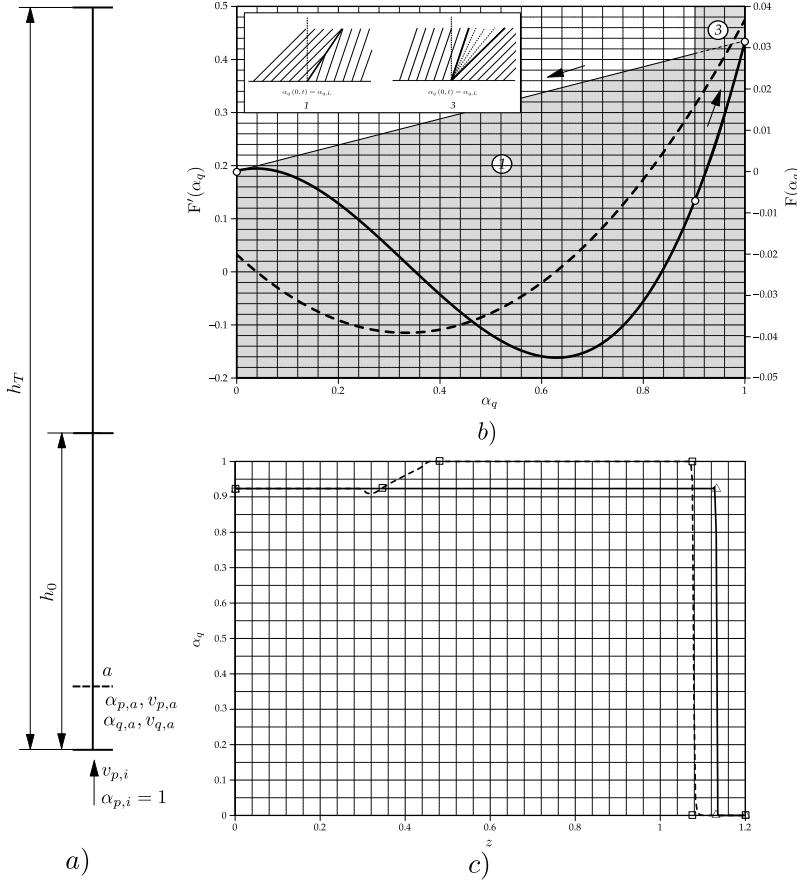


Figure 7.10: One dimensional reactor. a) sketch of geometry ; b) — flux, and - - - flux derivative; c) — initial condition, theoretical solution at \square $t = 1$, \triangle steady state, numerical solution at \cdots $t = 1$, — steady state.

elements and is also axi-symmetrical. The numerical parameters and boundary conditions are similar except for the top where α_q is set as `inletOutlet` with $\alpha_q = 0$ for inflow. For the pressure the total pressure is fixed and the velocity is set as `pressureInletOutletVelocity`. In order to allow for the formation of the bubble-plume a *diffusive zone* was set with a extension of 0.25 from the bottom. This diffusive zone corresponds to a region where only the ASMM is used, or in other words the indicator function is set as $\theta = 0$. The remaining parameters of the extended model are set as $\epsilon = 5 \times 10^{-3}$ and $\gamma_0 = 0.1$.

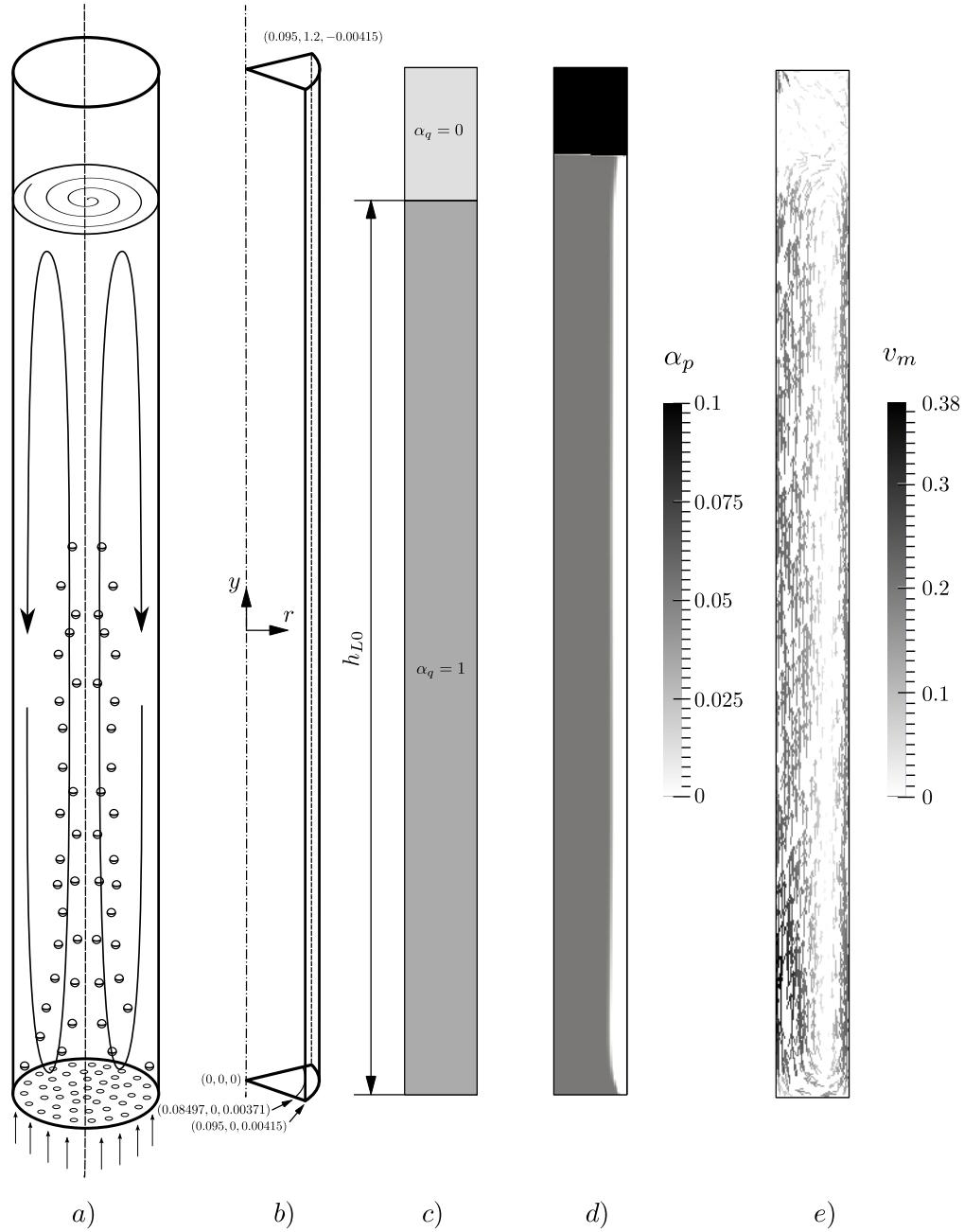


Figure 7.11: Geometry and solutions for a cylindrical bubble reactor. a) sketch of working principle b) computational domain, c) gas hold-up, d) center of mass velocity vector field.

Once the model is run and after a transitory the free surface reaches a stationary solution located $h = 1.0986$. The free surface has a slight slope product of the bubble plume as is shown in Figure 7.11.d), the slope is a function of the intensity of the bubble inlet. The velocity field is also obtained and shown in Figure 7.11.e); as was expected, the main recirculation is formed within the liquid zone below the free-surface. Over the free-surface the velocity field corresponds to the gas phase which ends to escape by the top boundary. In addition the center-of-mass velocity and gas hold-up can be examined on a transversal profile (at $y = 0.475$); which is shown in Figure 7.12, the results are similar to the ASMM case. Here it is important to note that the CARPT measures where done in real reactor (including the free surface) therefore they are more applicable in the present case. Respect to the 1D model proposed, two comparisons can be made; first, the predicted value of the secondary phase fraction, $\alpha_p \cong 0.07853$, which is surprisingly similar to the value reported in the profile $\alpha_p \cong 0.073418$ (see Figure 7.12, right). The second comparison is the value of the free-surface raising. The 1D model predicts a new position in $h_F = 1.134$. From Figure 7.10.d) it is clear that this value is not reached; this is attributable to the fact that not all the liquid in the reactor has trapped the predicted gas phase fraction. Then, adjusting the effective bubble zone to 0.076 in the radius (see Figure 7.12), the transversal area relationship is $0.076^2/0.095^2 = 0.64$, then the new free-surface position results to be $h_F = 1.045/(1 - 0.078530 \times 0.64) = 1.1003$ which is very close to the calculated position for the free surface. These results clearly confirm the validity of the 1D approximation in order to give a first assessment of the full model results.

7.5.3 Dam Break test with degassing

The Dam Break test was presented in section 5.4.2 as a part of the examples for the Volume of Fluid method. There, it became clear that after a first stage of mixing the more dense fluid traps bubbles of the less dense fluid which are lately removed by buoyancy. At the same time, droplets are formed from the splashing of the waves which fall by the gravity. The capacity of the

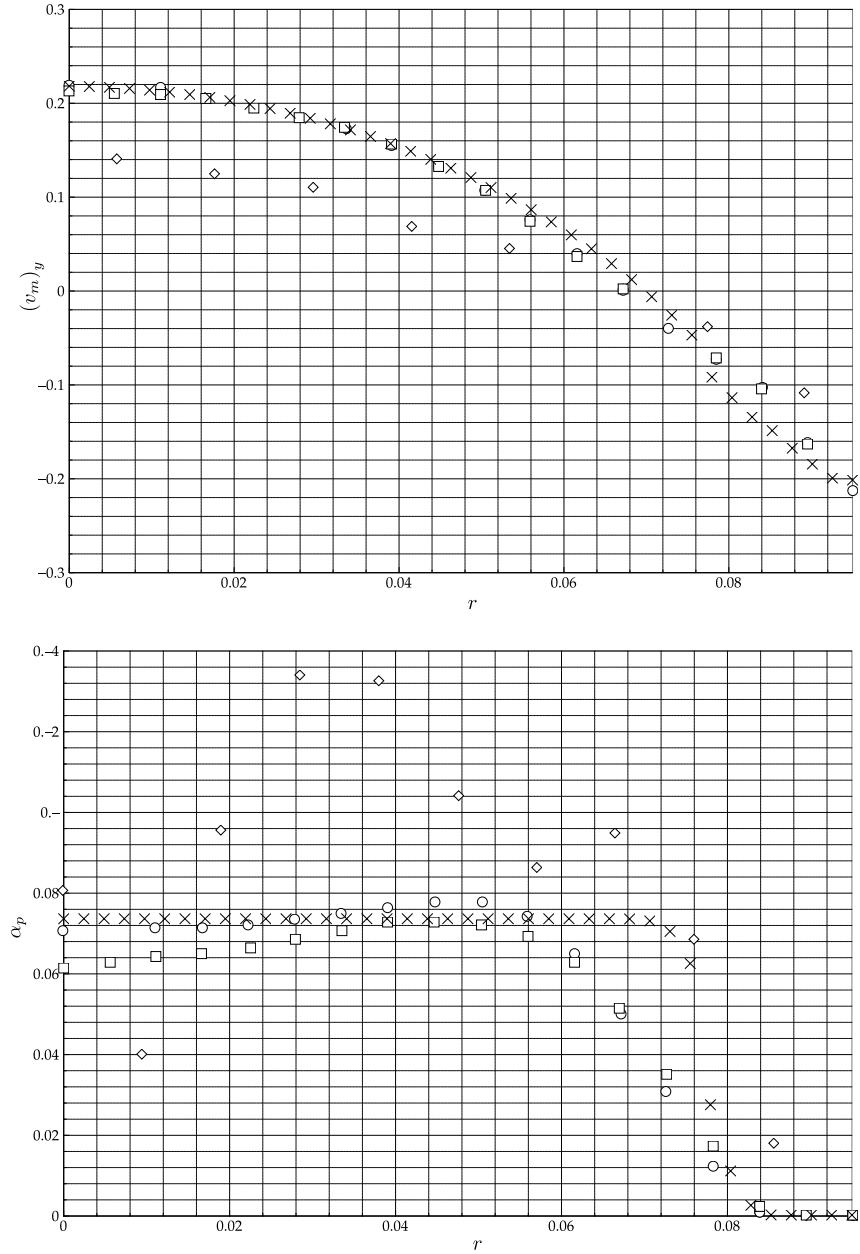


Figure 7.12: Velocity of center-of-mass and gas hold-up solutions as a function of the radius for the bubble column reactor with free surface at top ($y = 0.475$). \diamond CARPT, \circ ASMM (reference), \square Two fluids (reference), \times Present extended solver.

VOF model to capture the physics of the these particulate phases is directly related with the mesh refinement. This effect becomes clear in Figure 7.13. In this figure the solution for the Dam Break problem at $t = 4$ is shown for six different meshes. The first picture ($1\times$) shows the base mesh and then it is successively refined dividing the mesh by two in both dimensions (the example presented in the VOF chapter corresponds to the $1/16\times$ mesh). The expected effect is observed, as the mesh is refined the reconstruction of the trapped bubbles is improved and the buoyancy is consequently better modeled, therefore, the final state as a clean pool is reached more quickly. The improvement in the surface capturing accuracy is also represented. The idea behind the mesh refinement is also to obtain a reference mesh (whenever is possible) which could be considered as a DNS solution of the problem. Since the characteristic size of the bubble population is related to the surface tension this allows to estimate a mean bubble size in order to give a relative velocity law to the ASMM. In addition this size allows to determine from which mesh size it is possible to capture the bubbles individually. If the diameter of the bubbles for the Dam Break test is estimated in 1×10^{-3} , the first mesh to start capturing them would be between the $1/32\times$ and $1/64\times$ meshes.

The same analysis can be done for the extended model. It requires the selection of a relative velocity law for the particulate phase. As was stated the Dam Break test presents two particulate phases, droplets of the more dense phase and bubbles of the less dense phase. The selection of the dispersed phase model depends on the detection of which phase is continue and which phase is dispersed. This is not a trivial problem and is not treated in the available methods. The dispersed phase model is then selected based on a prescribed behaviour either as bubbles or droplets. Another option is to use a symmetric law for the dispersed phase model (Černe et al., 2001; Štrubelj and Tiselj, 2011), this approach has validity for $\alpha_q \sim 0.5$ since the drag laws have similar values, although is not completely correct reaching pure phases. From the figures it is clear that is necessary to give the proper physics to the trapped phase, then, a bubble model is selected for the relative velocity law with $a = 1$ and $\vec{v}_{re} = (0, 0.4422, 0)$. Then, the extended model is run with $\gamma_0 = 0.1$ and

$\epsilon = 5 \times 10^{-3}$ as the parameters for model coupling. The results are shown in Figure 7.14 where the effect of the ASMM applied to the dispersed phase is clear, the bubbles are removed giving a clear pool. The mesh refinement effect is also noted in the improvement of the free surface capturing.

The solution for the Dam Break test is also presented as a validation for the coupled model of Masuda and Nagaoka (Masuda and Nagaoka, 2006) (classic Dam Break, without obstacle). The authors also note the lack of capacity of VOF model to capture the bubbles and droplets and use the Two-Fluid method to give the dynamics for the particulate phase, although no quantitative validation is given respect to this effect. A possible measure of the bubble removal is to track the inventory of the trapped phase along time. To this end the solution for each time-step is filtered by p_{rgh} selecting only the cells with $p_{rgh} \geq 500$. This threshold was selected in order to capture big extensions of the more dense fluid containing either mesh captured or non captured short scale interfaces with the less dense fluid. This subset of the whole domain is denoted \mathcal{C} . Therefore, the inventory of the less dense fluid is given by $V_p = \sum_{\mathcal{C}} (1 - \alpha_1)$. The results are shown in Figure 7.15 in linear and semilogarithmic scale. The semilogarithmic scale in subfigure b) is given in order to have a better insight of the degassing period from $t \sim 1$.

From the figure is possible to assure that the extended model has better convergence than VOF model reaching excellent degassing without increasing the refinement. It becomes clear comparing the $1/4\times$ solution for extended model against the $1/32\times$ solution for VOF model. They show similar evolution and close level of degassing at the end of the run, giving an improvement factor of 8. Here it is important to note that the VOF mesh is $8^2 = 64$ times larger. Following the logic the $1/8\times$ solution presented for the extended model could only be compared to a $1/256\times$ VOF solution. It implies to go from a problem of 2,008,352 cells to another one with 128,534,528 cells, which is only affordable today by large HPC facilities.

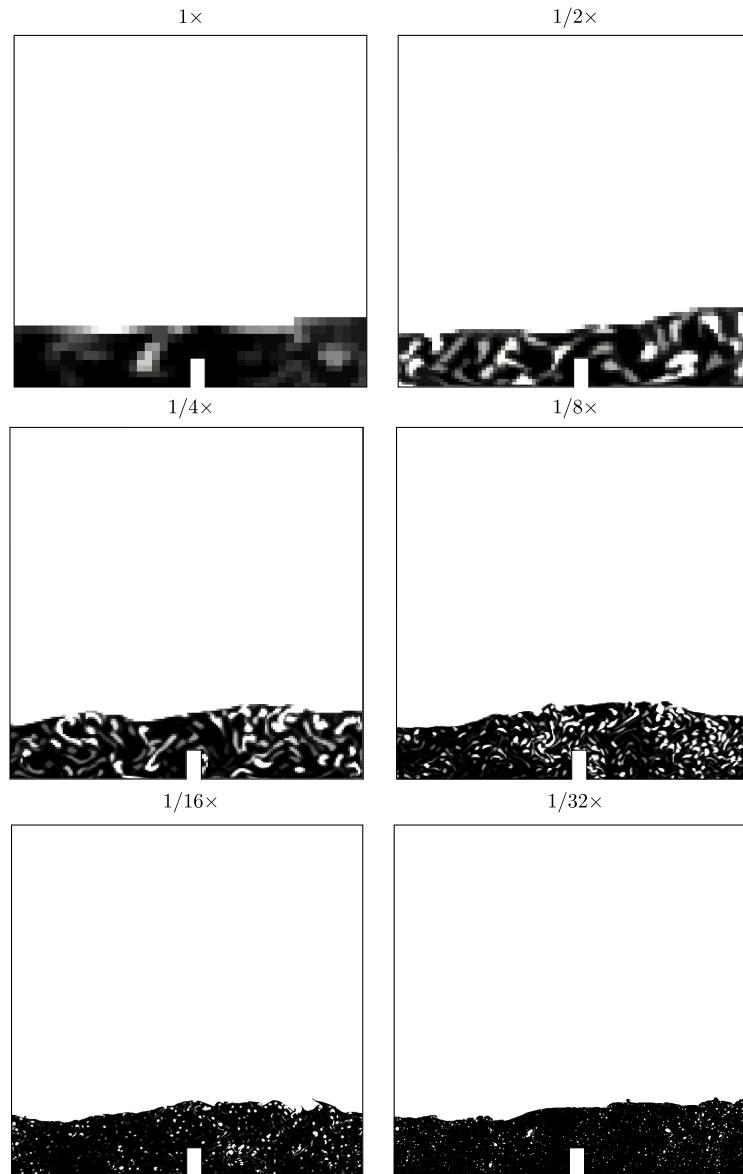


Figure 7.13: VOF solution for the Dam Break test at $t = 4$ for different meshes. The grayscale is saturated to white at $\alpha_q = 0.8$.

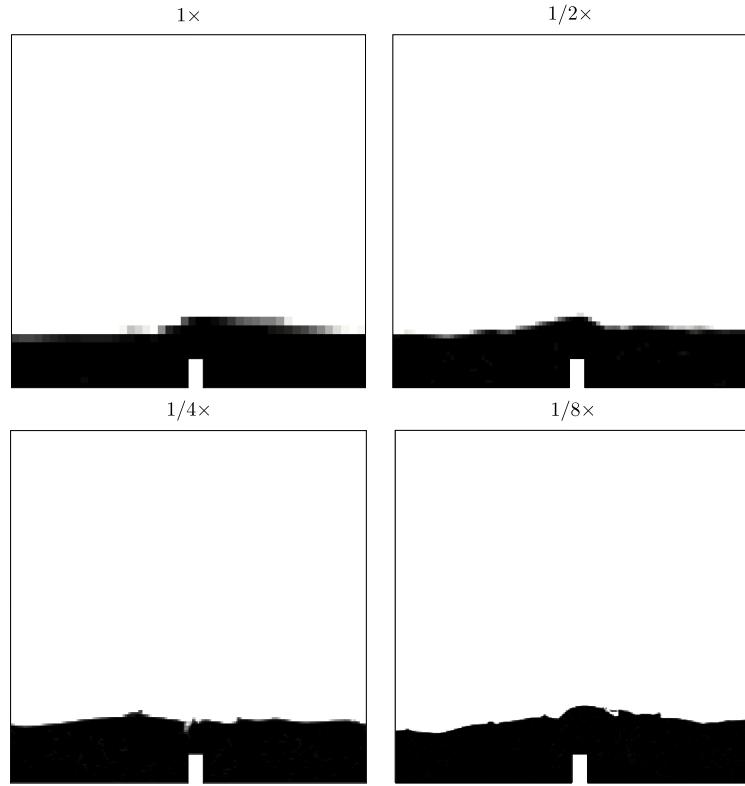


Figure 7.14: Extended model solution for the Dam Break test at $t = 4$ for different meshes. The grayscale is saturated to white at $\alpha_q = 0.8$.

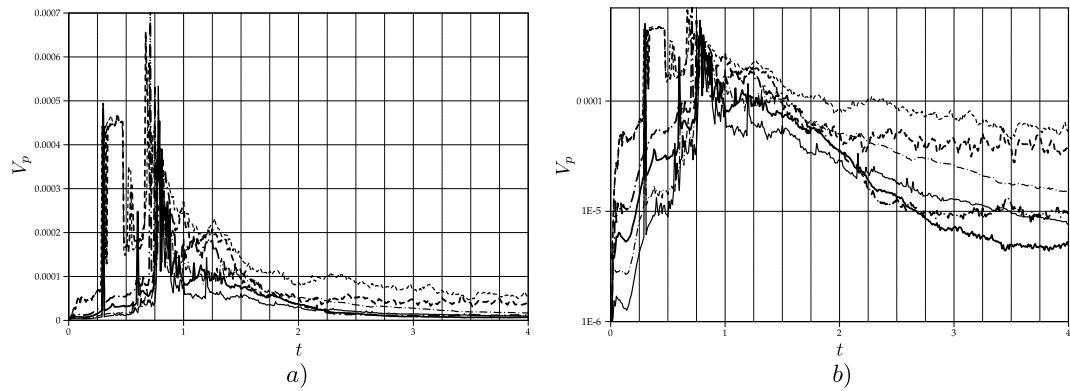


Figure 7.15: Evolution of the trapped phase volume, V_p , along the time for different meshes and models. a) linear scale, b) semilogarithmic scale.
 $1 \times$ VOF; $1/16 \times$ VOF; $1/32 \times$ VOF; $1 \times$ extended; $1/4 \times$ extended; $1/8 \times$ extended.

7.5.4 Rayleigh-Taylor instability

The final example corresponds to the Rayleigh-Taylor instability which was also presented for the VOF method (see section 5.4.1). As was analyzed earlier, after a first stage when the solution evolves with a mushroom-like structure the problem turns into the stretching of these structures and eventually the formation of droplets. The dynamics of this particulate phase can be correctly modeled by the VOF method with greater or lesser detail depending on the mesh size. Then, the main idea behind the use of an extended model is to give the proper physics to the particulate phase in order to have a better prediction, using the same meshes.

In order to evaluate the behavior of the extended model with respect to VOF the Rayleigh-Taylor model was run with VOF in a reference mesh of 256×1280 (327,680) elements. This mesh is a refined version of the base mesh presented as the VOF example, dividing each cell by two in both in x and y directions. Therefore, the base is called $1\times$, and the reference is called $1/2\times$. The series is completed with coarser meshes respect to the base mesh, which are called $2\times$ and $4\times$.

The problem is then run for VOF and extended models for all the meshes using the boundary conditions selected for the original VOF example. In the extended model the parameters for the indicator function are $\gamma_0 = 0.1$ and $\epsilon = 5 \times 10^{-3}$. The relative velocity law needed for the ASMM module in the extended model is adjusted to $\vec{v}_{rc} = (0, -0.1, 0)$ and $a = 1$ from the results in the reference mesh. The results are shown in Figure 7.16 for three different times $t = 1.8$, $t = 3.6$ and $t = 9$. In each time row the results for all meshes are shown comparing each VOF solution with this extended model pair. The first row corresponds to the end of the linear period, the second shows the stretching of the original structures and the presence of fluid chunks, finally the third row shows the falling droplets stage and the formation of the bottom pool. From the figure is clear that the linear period is well represented by both methods in all of the meshes since the mushroom-like structure is formed

by long-scale interfaces. The stretching of the original structures shows the deficiencies of the coarser meshes where non-physical fluid chunks starts to appear. Finally in droplets/pool stage it is clear that the VOF method try to agglomerate the chunks meanwhile the extended model, which is working in ASMM regime treats the chunks without taking into account the mesh resolution.

In order to compare the solution quantitatively the quadratic mean error is calculated for the $4\times$, $2\times$ and $1\times$ respect to the $1/2\times$ reference mesh for both methods. In order to do so, all the results are mapped to the coarsest mesh. The error is calculated as is shown in Eqn. (7.20) (Černe et al., 2001)

$$\delta_{\text{cell}}(t) = \frac{1}{V_1} \sum_{i=1}^n [\alpha_q(t) - \alpha_{q,\text{REF}}(t)]_i^2 \quad (7.20)$$

where i is the index for the cells, n is the total number of cells in the mesh and $V_1 = \sum_{i=1}^n \alpha_{q,\text{REF}}(t)$. The results are presented in Figure 7.17 and show that the effects of the mesh refinement are clear until $t \sim 4.5$. In this period as the meshes are finer the solution is more accurate, this behavior is similar for both VOF and extended models. In addition, each VOF solution is followed by its extended model pair showing the capacity of extended model to capture the long-scale interfaces while the mesh remains fine respect to the interface scale. Once the big structures are stretched and the droplets are formed the convergence in not completely clear.

Another quantitative validation is given by the accumulation of the more dense phase in the bottom of the domain. It gives a measure of the correct capturing of the falling droplets physics. Therefore, the integral of α_q is calculated in a box with its upper boundary located at $y = 0.5$ for all of the times. The results shown in 7.18 allow to conclude again that the VOF and extended model behave similarly, nevertheless the extended model seems to converge better in a factor of 2, since the results for $2\times$ in the extended model

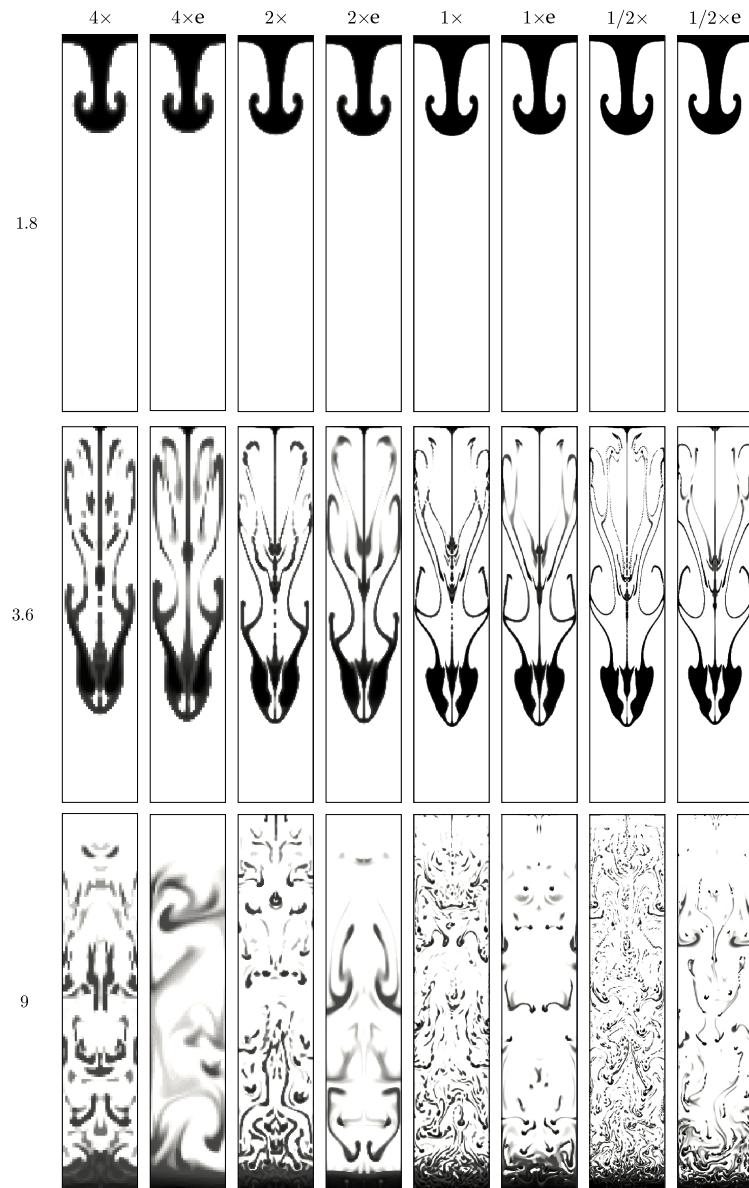


Figure 7.16: Extended model and VOF solutions for the Rayleigh-Taylor problem at $t = 1.8$, $t = 3.6$ and $t = 9$ for different meshes. The grayscale is saturated to black at $\alpha_q = 0.16$.

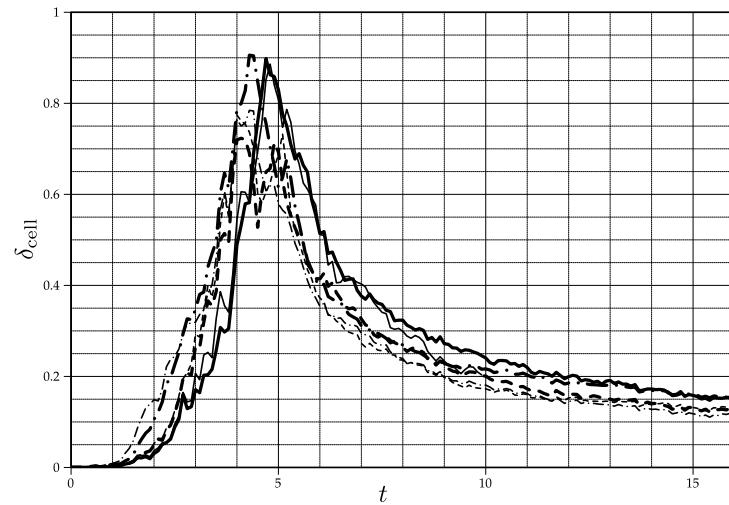


Figure 7.17: Quadratic mean error for VOF and extended model solutions respect to VOF reference for the Rayleigh-Taylor problem. — 1× VOF, - - - 2× VOF, - · - 4× VOF, — 1× extended, - - - 2× extended, - - - - 4× extended.

are comparable to the results for 1× in VOF. The same conclusion is obtained comparing the results in 1× in the extended model and the results for 1/2× in VOF.

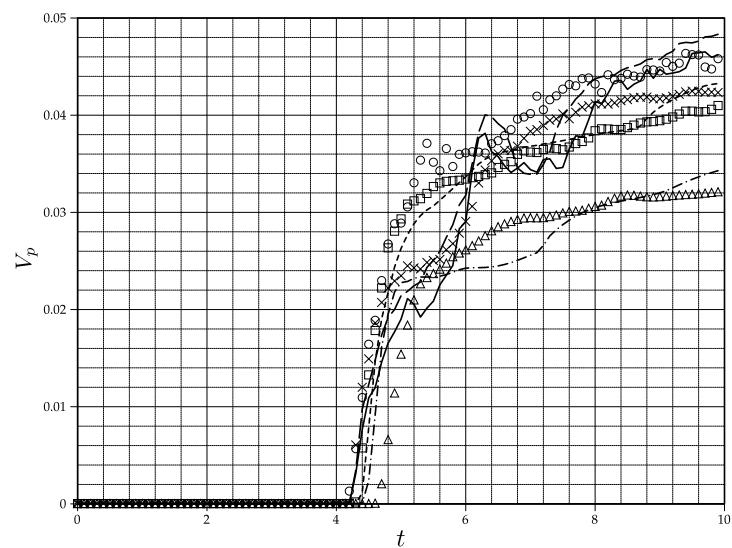


Figure 7.18: Accumulation of the more dense phase at the bottom of the domain for the Rayleigh-Taylor problem. \circ $1/2 \times$ VOF, \times $1 \times$ VOF, \square $2 \times$ VOF, \triangle $4 \times$ VOF, — $1/2 \times$ extended, - - - $1 \times$ extended, - - - - $2 \times$ extended, - - - - - $4 \times$ extended.

Chapter 8

Conclusion

8.1 Conclusions of the work

At the end of this work it is possible to summarize the conclusions given in each chapter completing an unified view of the results.

- A brief introduction to Finite Volume Method discretization was given including a description of the High Resolution Schemes framework based on Total Variation Diminishing schemes and the Flux Corrected Transport (MULES) method as are implemented in OpenFOAM®. The importance of the description of this two methods is superlative since the correct solution of the wave phenomena lately presented strongly relies in these concepts. In addition, this description is a documentary contribution to the user's community.
- The Algebraic Slip Mixture Model was revisited giving emphasis in its strengths respect to the Multi-fluid method and the main numerical issues related to its implementation. These issues are related to the lack of evolution equation for the pressure, due to a low-mach regime, and the necessity of keeping the fraction bounded as a result of the whole system of equation and not only due to the correct solution of the phase fraction equation. The presence of different kind of waves was clearly demonstrated by the study of the flux function of the phase fraction

equation and by examples. To this end, a semi-analytic solution was proposed based on the velocity of center-of-mass formulation. The results of this solution served as a validation for the numerical solvers proposed in later chapters. The relationship of these methods and the theory of sedimentation of Kynch is also addressed.

- The solution of hyperbolic equations being able to capture the different kind of waves led to explore the numerical methods capable to manage this phenomena, therefore the KT method was recalled in order to present a new Riemann-free solver based on a centered phase flux at faces. The necessity of this kind of schemes was based on the structure of FVM solvers. In these solvers a key concept is the conservation of the transported quantities which requires the assembling and use of a *conservative flux* which is the face version of the velocity. The theory of Riemann-free solvers is generally associated to compressible problems where a complete set of hyperbolic equations is available, including an evolution equation for the pressure. Then, the whole problem is given at cell centers. When the problem is related to low-mach cases which leads to an incompressible formulation no evolution equation is given for the pressure and the numerical solutions rely on Fractional Step/SIMPLE/PISO methods. These methods require a spatial separation of pressure and velocity variables which is usually achieved by staggered grids or pseudo-staggered grids where the pressure is given at cell-centers and the velocity is treated as a flux at cell faces. Thus, in the case of the proposed method the velocity is given at cell faces by the pressure-velocity coupling loop (PISO), this face flux is introduced to assemble the final flux which is stabilized by using the information given by the local speeds of the problem. In addition a modification to FCT methods was proposed to allow the use of global maxima instead of the local ones.
- In order to set a base for the extended model the formulation of the VOF model was recalled showing it as a derivation of the ASMM. In this way the role of the relative velocity was presented giving place to the Weller-VOF method. The algorithm of the `interFoam` solver was described

and then used to solve two well known problems as the Rayleigh-Taylor instability and the Dam Break test. These solutions give a basis for further comparison with the methods which are the main objective of this work.

- After giving all the basic construction blocks, an ASMM solver based on Riemann-free and TVD methods was presented assuring the boundedness via FCT methods. The solver was implemented and tested using the examples devised in the third chapter and a sedimentation case from the literature. In addition a laboratory case was solved related to bubble column reactors.
- As the principal contribution of this thesis an extended mixture model was presented and a high performance solver was implemented and tested based on the ASMM and VOF solvers previously presented. This model required the development of an indicator function to allow the correct coupling of both ASMM and VOF methods. Then, a discussion of the state of the art was given in order to have a better understanding of the scope and limitations of the known coupled solvers, this discussion includes the development of new theoretical and experimental solutions which allow the validation of the models. Therefore the model was applied in four examples. The first one was an experimental bubble plume taken from the literature giving place to a new test case. The solution of this case allowed to a semi-quantitative validation based on the inspection of the solution and the comparison of the shape of the free-surface. This last comparison was done against fitting curves based on experimental results. The second case was given by the solution of the bubble column reactor presented in chapter six. Here the presented technique allowed to remove the restriction given by the top boundary condition. The original problem included there an expansion boundary condition which was no more necessary since the whole domain was solved allowing the volume expansion within the reactor. This example includes a simple and powerful 1D model which allows to predict the gas hold-up and free surface position. Both examples are presented as contributions to the

set of available validation problems. The third example relied on the Dam Break problem, solved with the extended method and a comparison was done against the VOF model which is the typical method used to solve it. Therefore an improvement of eight times was found related to mesh requirements based on a novel quantitative comparison. The final example was the Rayleigh-Taylor instability where the extended model was compared to VOF again showing the capacity of correctly capture the big structures giving the same performance that VOF. In a second stage where the problem evolves as particulate phases the extended model gave an improvement factor of two. As a measure of the improvement the accumulation of the primary phase at the bottom of the domain was tracked.

- The implementation of the solvers required a process of coding, debugging, and case testing. To this end, a debugging tool was presented in the Appendix A, called gdbOF. This tool allowed to extend the capabilities of the GNU Debugger to the OpenFOAM® data structures showing them by the text console and graphically by means of Paraview®.

8.2 Future work

The development of the models and tools related to the present thesis give some answers, however, mainly leave a lot of open questions and much work to be done. Part of the future work includes:

- To exploit more deeply the potentialities of the proposed ASMM solver based on Riemann-free methods;
- Adding turbulent diffusion in the ASMM scale of the extended model in order to improve the prediction of turbulent dispersion. The importance of this matter was inspected in the bubble plume problem presented in Chapter 7;
- It is necessary to explore the transitions from SSI to LSI and check the correct capturing of the last. This case is presented in sedimentation cases,

boiling, etc. In addition it is necessary to run tests checking whether is necessary to have a definite LSI at the beginning of the time;

- The influence of the degassing velocity at the free surface needs to be explored, this value is related to the compressive velocity used in VOF zones and is now calculated only based on numeric considerations;
- An open question which deserves much attention and which has not been discussed by the community is related with the interaction of the SSI and the LSI. From the given examples it is clear that the dynamics of the SSI is improved, now is necessary to know how the LSI simulation is improved due to the better calculation of the SSI. This question is not so trivial to answer since requires very accurate examples either theoretical or experimental. A first assessment of this kind of dynamics could be given, for example, solving the bubble plume case using turbulent dispersion for the particulate phase. Therefore, the under-predicted width of the plume given by VOF could be adjusted by the dynamics of the unresolved scale given by the turbulent ASMM;
- Much of this work is inspired in the solution of sedimentation problems and bubble plumes applied to the oil and gas and siderurgy industries. It would be valuable to apply the developed methods to skimmer tanks and bubble plumes in furnaces and other industrial real problems.

Appendix A

A note in OpenFOAM® programming

A.1 Introduction

OpenFOAM® is a CFD library that allows users to program solvers and tools (for pre-processing or post-processing) in a high-level specific language. This high-level language refers to the fact of writing in a notation closer to the mathematical description of the problem, releasing the user from the internal affairs of the code.

This programming approach contrasts with procedural languages approach, such as Fortran, that are widely used in academic and scientific environments but oriented to the low-level problem resolution, i.e., the manipulation of individual floating-points values. Thus, in order to achieve the abstraction from the low-level coding it is necessary to follow another way, therefore the Object-Oriented Programming (OOP) paradigm is selected. This methodology produces code which is easier to write, to validate and to maintain compared with purely procedural techniques. Respect to OpenFOAM® it is completely written in C++. This language is less rigorously object-oriented than the others languages (such as Java, SmallTalk or Eiffel), due to the inclusion of some characteristics that are not strictly object-based. The main add-on is

operator overloading, which is essential to working with tensor, vector and scalar fields objects concepts as in the mathematical notation. On the other hand, it is a multiplatform language and, due to that it is based on C, is as fast as any other procedural languages (Cary et al., 1997).

There are five fundamental concepts in OOP, whereby OpenFOAM® achieves its objectives: *modularization*, *abstraction*, *encapsulation*, *inheritance* and *polimorphism*(Weller et al., 1998). All of them are widely used in the code. Polymorphism is a key concept in OpenFOAM®, which is clearly demonstrated by the proliferation of virtual methods (methods that must be implemented in child classes). Examples of this include the implementation of boundary conditions, which inherit from a base class `patchField`, so they have the same interface but different implementations. Another example is the representation of tensor fields: in this case `geometricField` is the parent class and various tensor fields inherit from it: `scalarField` (rank 0), `vectorField` (rank 1) and `tensorField` (rank 2), each one implementing the interface provided by the parent class in different ways.

In addition to these OOP features, there are other tools of the C++ language which are not strictly object-based and those are used in OpenFOAM®. They are the aforementioned operator overloading and the use of preprocessor macros. Macros allow to insert code directly in the program, avoiding the overhead of invoking a function (passing parameters to the stack, do a jump, take parameters), without losing the code readability (Eckel, 2000).

As it was mentioned, using these techniques a library oriented to high-level development is generated, ensuring that the user only has to take care about the model to solve and not other details of coding (Mangani et al., 2007). On the other hand, some problems could arise in the application creation stage yielding to undesired results. There begins the code debugging work, and this includes monitoring values corresponding to variables involved in the resolution, such as, tensors, vectors and/or scalar fields defined at cell or face centers, coefficients in the system matrix, and many other examples.

In addition, debugging is not ever motivated by problems, but simply for exploratory or control purposes (Ewer et al., 1995).

From the side of debugging tools in GNU-Linux platforms, *gdb* (GNU-debugger) is the *defacto* standard. It includes a variety of tools for code analysis and data inspection at run-time (Matloff and Salzman, 2008) which gives a successful environment for OpenFOAM® debugging. *gdb* offers a powerful print command likely to inspect arrays in memory, nevertheless it can be used directly only in simple data structures like `lists` or `Fields`. Data examination gets hard when viewing the desired data involves polymorphism and inheritance connected with the virtual methods used by the library. This work requires to walk through the general class tree looking for the attributes which are wanted to be inspected. Moreover, once desired attributes are found, these maybe do not directly represent the information required by the developer. In the case of the matrices generated by `fvm` methods, they store the coefficients using the LDU Addressing technique (See *gdbOF* User's Manual, Appendix A¹), therefore it is necessary to apply a decoding algorithm to transform it into the traditional format (full or sparse), and to control and operate with their values.

The main objective of the *gdbOF* tool is to solve problems like those explained in the previous paragraph. This tool is implemented by *gdb* macros and it is based on an implementation of *gdb* macros for STL (Standard Library for C++) debugging (Marinescu, 2008). These macros simplify the task of debugging the OpenFOAM® libraries, performing the work actions transparently to the user: the simple call of a *gdb* macro from console triggers a sequence of actions that include: navigate the OpenFOAM® class tree, collect information and reorder it for representation in an user readable format. Moreover, *gdbOF* includes the option of writing the output into a file on disk and to view it graphically. This output is formatted appropriately to be imported in numerical computation software such as Octave or Matlab®, thus allowing the developer to expand the possibilities of data inspection at

¹http://openfoamwiki.net/index.php/Contrib_gdbOF

debugging time.

In this appendix the design concept of the tools will be presented and several cases will be solved as examples of use. These problems not only emerge in an academic context but also occur in real application environments: the first consists in a scalar advective-diffusive problem in which the emphasis will be placed on the assembling and storage of matrices; the second consists in a non-orthogonal correction method in purely diffusive tests; and the third is an analysis of multiphase solvers based on Volume of Fluid Method. The last examples are focused in volumetric and surface data inspection both in array and graphical format.

A.2 Basic debugging

One of the most common tasks in the debugging process is to look at the values stored in an array, that is possible in gdb with the command of Example 1, where `v` is the array to analyze.

Example 1 View array.

```
$ (gdb) p *v@v_size
```

Nevertheless, as it was pointed out in the previous section, data inspection in OpenFOAM® requires often more complex sentences. A typical example is to verify at debugging time that a certain boundary condition is being satisfied (typically when the boundary condition is coded directly in the solver and the next field information is obtained after solving the first time-step). Boundary conditions in OpenFOAM® are given for each patch in a `GeometricField`, then, assuming that the inspected patch is indexed as 0 (the attribute `BoundaryField` has information of all the patches), sentence presented in Example 2 is needed to observe the values on this patch, where `vSF` is a `volScalarField`.

Note that the statement in Example 2 doesn't include any call to inline functions, which could generate some problems in gdb, giving even more

Example 2 View Boundary Field values.

```
$(gdb) p *(vSF.boundaryField_.ptrs_.v_[0].v_)
@(vSF.boundaryField_.ptrs_.v_[0].size_)
```

complex access to information.

gdbOF solves the inconvenience of knowing the attribute's place and using long statements. Using *gdbOF* commands, as it is shown in Example 3, the same results are obtained. Note the simplification of the statement, this is the *gdbOF* spirit, reducing the work needed to debug and perform the same tasks more simply and transparently.

Example 3 View Boundary Field values with *gdbOF*.

```
$(gdb) ppatchvalues vSF 0
```

There are many examples in OpenFOAM® like the previous one in which the necessity of a tool that simplifies the access to the complex class diagram can be useful. Note that in the last example it wasn't mentioned how the index of the desired patch was known. Usually OpenFOAM® user knows only the string that represents the patch, but not the index by which it is ordered in the list of patches. Here *gdbOF* simplifies the task again, providing the **ppatchlist** command which displays the list of patches with the corresponding indexes. Regarding to other basic gdfOF tools please refer to the *gdbOF User's Manual*, Chapter 2.

A.3 Advanced Debugging

A.3.1 System matrix

Increasing the complexity of debugging, there can be found cases involving not only the search and dereference of some plain variables. A typical case is the dumping of the linear system, $Ax = b$, generated by the discretization of a set of differential equations which are being solved. This is stored using the

LduAddressing technique which takes advantage of the sparse matrix format and saves the coefficients in an unusual way. This storing format and the necessity of accessing to individual matrix coefficients lead to trace the values one by one and to apply a decoding algorithm. There are two commands to do this task, one to dump the data as full matrices and the other to dump the data as sparse matrices.

In order to implement the necessary loops over the matrix elements, gdb provides a C-like syntax to use iterative (while, do-while) and control structures (if, else). These commands have a very low performance, so the iteration over large blocks of data must be done externally. *gdbOF* becomes independent of gdb for the assembly of matrices using another platform: the *lDUAddressing* vectors are exported to auxiliary files and the calculation is performed in another language through calls to the shell. Thus, python is chosen due to its ability to run scripts from console and having a simple file management, both to load and to save data. This is performed by the *pfvmatrixfull/pfvmatrixsparse* commands whose structure is presented in Pseudo-code 1

Pseudo-code 1 Structure of *gdbOF* Command *pfvmatrixfull/pfvmatrixsparse*.

1. Get parameters
 2. Get upper and lower arrays with gdb
 3. Redirect data to an auxiliary file
 4. Format the auxiliary files: gdb format → python format
 5. Call python script to assemble the matrix
 - (a) Read auxiliary files
 - (b) Set limits
 - (c) Do lduAddressing
 - (d) Complete with zeros
 6. Format auxiliary files: python format → gdb format
 7. Show output or/and save file in octave format. Add header (sparse case)
-

A.3.2 Mesh Search

Another group of macros are those which search in the mesh. The aforementioned inability of *gdb* to perform loops on large blocks of data extends to the case of meshes, forcing thus to do the searching tasks using external tools. In order to circumvent this issue OpenFOAM®’s mesh methods are used to accomplish these tasks. Thus *gdbOF* includes *ad hoc* stand-alone applications to which call at debugging time to search in the mesh. Even though this way means creating a new instance of the mesh in memory, the cost in time and development is lower than that required to accomplish the search on the mesh in *gdb*, implementing the loops in the *gdb* C-like syntax, or in another language such as python. These OpenFOAM® applications are included in *gdbOF* package and they are compiled when the *gdbOF* installer is run.

Cases of searching on the mesh typically covered by *gdbOF* are those which start with a point defined by $[x, y, z]$, returning a cell index or values in some field, either in the center of cell (`volFields`) or at each of its faces (`surfaceFields`).

Regarding to obtaining the value of a field at some point there is no more inconvenient than finding the index of the cell or index of the cell containing the point (via `pfindcell` command), whose centroid is nearest to it. The corresponding `volFields` command returns two indexes: the index of the cell that contains the point, and the index of the cell which has the nearest centroid. Afterwards, the user put one of these indexes in the command `pinternalvalueslimits` to extract the field value in the cell centroid, or to observe the equation assembled for that cell with the command `pfvmatrix`.

The algorithm of this tool is presented in Pseudo-code 2, where it may be noted that it does not exist any communication between *gdb* and other platforms more than the shell call. The return of the results is through temporal files, which must be generated in a particular format to be readable by *gdbOF*. This particular technique is used since it is not possible to access from a given process to the data of a second process which is being debugged.

Pseudo-code 2 Structure of *gdbOF* Command `pfindcell`.

1. Get parameters
 2. Call FOAM app. to make the search
 - (a) Start new case
 - (b) Do search
 - (c) Save results in a temporal file
 3. Read temporal file using a shell script
 4. Show the indexes by standard output
-

Another kind of searching through the mesh is to find a list of indexes of faces belonging to a cell. This task operates in a similar way. The user invokes a *gdbOF* command and this uses a back-end application. Despite the simplicity of using the commands, the code is more intricate because the storage of faces in a cell is not correlated, and the faces are subdivided in internal or boundary faces (this requires walking through the list of faces in the mesh). It is also needed to identify whether these faces are in the `internalField` or in one of the patches in the `boundaryField`: the last option requires seeking the patch which the face belongs to and the local index of the face within this patch. With this information it is possible to obtain the field's value at that face. For more information see *gdbOF User's Manual Appendix C*.

The *gdbOF* command `psurfacevalues` performs this search: given a cell, find the indexes of the faces that make up it and the value of the chosen field in each of these faces.

In `pfindcell`, the result stored on disk was only necessary to parse and display it on console, but in this case, the indexes that returns the application should be used to access to an array containing the values of the field. To do that, this implementation requires to generate a temporal gdb macro (using a shell script) because it is not possible in gdb to assign the result of extracted data from a file to a variable. The Pseudo-code 3 presents this implementation.

Pseudo-code 3 Structure of *gdbOF* Command `psurfacevalues`.

1. Get parameters and check if it is a `surfaceField`
 2. Call FOAM application to make the search
 - (a) Start new case
 - (b) Do search
 - (c) Save results in a temporal file
 3. Read temporal file using a shell script
 4. Through each index:
 - (a) Generate temporal macro
 - (b) Call macro (this macro prints the results)
-

Note that the temporal gdb macro is generated on the fly and it is only functional for the parameters generated in the temporal code of the macro (field name and location of the desired value), then the loop in all faces of the cell is transparent to the user and it is not a problem for debugging.

A.3.3 Graphical debugging

Having in mind that the aim of these tools is the debugging of field manipulation software, the most powerful tool is finally presented. It consists on the spatial visualization of fields in a graphical way.

This is a widely spread concept which reminds the first efforts in graphical debugging (Dewar and Cleary, 1986). An usual application of graphical debugging are general data structures (Waddle, 2001; Korn and of Computer Science, 1999), and particularly linked-lists (Shimomura and Isoda, 1991) and graphs (Parker et al., 1998). Data Display Debugger (Zeller and Lutkehaus, 1996; Cruz et al., 2008) can be cited as an useful and general tool for these purposes. Respect to the field manipulation software debugging, it requires mesh manipulation and more sophisticated data analysis tools which

drives to specific implementations (Grimm, 2002; Abramson et al., 1996).

In the *gdbOF* particular case, this objective summarizes previously presented tools, and it is particularly tailored for `volField` debugging. Basically it consists in an OpenFOAM® format data dump tool callable from any debugging point with optional `.vtk` file format for exporting (via `foamToVtk` tool) and Paraview® (Squillacote and Ahrens, 2006) on the fly running. The algorithm to achieve this goal is presented in Pseudo-code 4.

Pseudo-code 4 Structure of *gdbOF* Command `pexportfoamformat`.

1. Get parameters and check if it is a `volField`
 2. OS environment setting (first run)
 - (a) Creation of data dump directories
 - (b) Symbolic linkage of `constant/` and `system/` to avoid data duplication
 3. Get actual time-step and last data written name
 4. Write OpenFOAM® file format header and set field dimensions
 5. Write `internalField`
 6. Identification of boundary patches via `ppatchlist` calling.
 7. For each patch, write boundaries' `surfaceFields`.
 8. Close file.
 9. Call optional parameters (`.vtk` exporting and Paraview® running)
-

A.4 Tests

A.4.1 Scalar Transport Test

The first test consists of the unsteady advective-diffusive equation, in a two dimensional geometry with a mesh of 3×3 cells, which is shown in Figure A.1.

The partial differential equation solved is presented in Eqn. (A.1).

$$\frac{\partial \rho\phi}{\partial t} + \vec{\nabla} \cdot (\rho\vec{v}\phi) - \vec{\nabla} \cdot (\rho\nu\vec{\nabla}\phi) = S(\phi) \quad (\text{A.1})$$

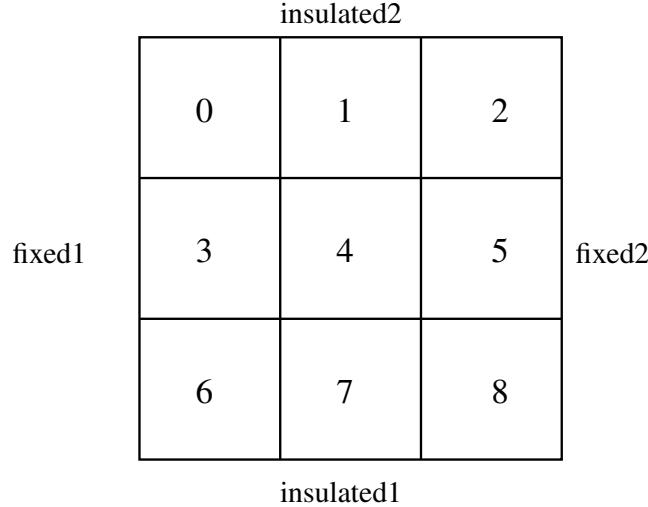


Figure A.1: Geometry and patches in scalar transport test (numbers identify cells, figure credits to Juan M. Giménez).

with the boundary conditions shown in Eqns. (A.2-A.4).

$$\vec{\nabla}\phi \cdot \vec{n}|_{\text{insulated}} = 0 \quad (\text{A.2})$$

$$\phi_{\text{fixed1}} = 373[\text{K}] \quad (\text{A.3})$$

$$\phi_{\text{fixed2}} = 273[\text{K}] \quad (\text{A.4})$$

where $\vec{n} = \frac{\vec{S}_f}{|\vec{S}_f|}$. To solve this problem, the following parameters are selected: $\vec{v} = [1, 0]$, $\Delta t = 0.005$, $\rho = 1$, $\nu = 0.4$, $S(\phi) = 0$ and $\phi^0 = 273$ uniform along the whole domain as initial solution.

In the Finite Volume Method, each cell is discretized as it is shown in Eqn. (A.5) (Jasak, 1996).

$$\frac{\phi_P^n - \phi_P^0}{\Delta t} V_p + \sum_f F \phi_f^n - \sum_f \nu \vec{S}_f (\vec{\nabla} \phi)_f^n = 0 \quad (\text{A.5})$$

where ϕ is the unknown field, V_p the cell volume, Δt the time-step, F the flux of the advective field at the faces \vec{S}_f . Finally the superscripts n and 0

represent different time-steps and f subscript indicates a face value.

It is known that the assembly of a problem that includes convection using the upwind method results in a non-symmetric matrix, in addition, increasing the diffusive term and decreasing the time step, this matrix will tend to be diagonal dominant.

Assembling the Eqn. (A.5) in each cell for the initial time ($t = 0.005$), the system of equations presented in Eqn. (A.6) is obtained.

$$\begin{aligned}
 202.6\phi_0 - 0.4\phi_1 - 0.4\phi_3 &= 55271.4 \\
 -1.4\phi_0 + 202.2\phi_1 - 0.4\phi_4 &= 54600 \\
 -1.4\phi_1 + 201.6\phi_2 - 0.4\phi_5 &= 54545.4 \\
 -0.4\phi_0 + 203\phi_3 - 0.4\phi_4 - 0.4\phi_6 &= 55271.4 \\
 -0.4\phi_1 - 1.4\phi_3 + 202.6\phi_4 - 0.4\phi_5 - 0.4\phi_7 &= 54600 \\
 -0.4\phi_2 - 0.14\phi_4 + 202\phi_5 - 0.4\phi_8 &= 54545.4 \\
 -0.4\phi_3 + 202.6\phi_6 - 0.4\phi_7 &= 55271.4 \\
 -0.4\phi_4 - 1.4\phi_6 + 202.2\phi_7 - 0.4\phi_8 &= 54600 \\
 -0.04\phi_5 - 1.4\phi_7 + 201.6\phi_8 &= 54545.4
 \end{aligned} \tag{A.6}$$

OpenFOAM® Assembly

The above system, which was assembled manually, can be compared with the system obtained by running the OpenFOAM® solver `scalarTransportFoam`.

Establishing a breakpoint in the proper code line, and calling the `gdbOF pfvMatrixfull` command, the system matrix \bar{A} is printed on the console. This matches the manually generated system, showing the right performance of the tool.

An additional feature of this command and others, is the ability to

Example 4 View system matrix with *gdbOF*

```

$(gdb) b fvScalarMatrix.C:144
$(gdb) run
$(gdb) pfvmatrixfull this fileName.txt
$(gdb) shell cat fileName.txt
202.60    -0.40     0.00   -0.40   ...
-1.40     202.20    -0.40     0.00   ...
 0.00     -1.40     201.60    0.00   ...
-0.40      0.00     0.00    203.00   ...
...
...
...
...
...
(gdb) p *totalSource.v_@9
{55271.4, 54600, 54545.4, 55271.4 ...

```

export data in a file format compatible with the calculation software Octave and Matlab®. To do this only one more parameter is needed in the command invocation, indicating the file name. Thus, *gdbOF* is responsible for exporting the values in the correct format, using rows, columns and values in [row,col,coeff] format. `pfvmatrixsparse` exports the matrix of the system in this format which has a header that identifies the file as a sparse matrix. This method greatly reduces the size needed to store the matrices in the case of medium or large meshes.

Regarding to patch commands this example is also useful to show their potentiality. Suppose that checking a boundary condition is wanted, for example the value $\phi = 373^2$ in the *fixed1* patch. First of all, it is necessary to know the index of this patch. Once the patch index is known, it is possible to see its values (See Example 5). The output is an array with three values corresponding to the boundary condition on each one of the three faces that make up this patch.

Appendix B of the *gdbOF* User’s Manual shows how the internal and boundary values (in `volFields` and in `surfaceFields`) are stored in OpenFOAM®.

²In the case, T is used to represent the `scalarField` instead of ϕ because OpenFOAM® preserves ϕ for a `surfaceScalarField` which represents the flux through each face ($\phi = S_f \cdot U_f$)

Example 5 View patches list with *gdbOF*

```
(gdb) ppatchlist T
PatchName    --> Index to Use
FIXED1      --> 0
FIXED2      --> 1
INSULATED2   --> 2
INSULATED1   --> 3
FRONT_AND_BACK --> 4
(gdb)
(gdb) ppatchvalues T 0
(gdb) $1 = {373,373,373}
```

A.4.2 Laplacian Test

In this problem, *gdbOF* is used to monitor the field values and the resulting equations system, in order to realize how the correction method for non-orthogonal mesh used in OpenFOAM® works³ (Jasak, 1996; Versteeg and Malalasekera, 2007).

The problem to solve is defined in the Eqn. (A.7), with the boundary conditions shown in Eqns. (A.8-A.10), and the non-orthogonal mesh presented in Figure A.2.

$$\vec{\nabla} \cdot (\rho\nu\vec{\nabla}\phi) = 0 \quad (\text{A.7})$$

$$\vec{\nabla}\phi \cdot \vec{n}|_{\text{insulated}} = 0 \quad (\text{A.8})$$

$$\phi_{\text{fixed1}} = 273 \quad (\text{A.9})$$

$$\phi_{\text{fixed2}} = \phi_{\text{fixed1}} \quad (\text{A.10})$$

Constants and initial conditions are: $\rho = 1$, $\nu = 1$ and $\phi^0 = 0$ in the whole domain.

³The diffusive term in a non-orthogonal mesh is discretized in the following way: $\vec{S}_f \cdot (\vec{\nabla}\phi)_f = \vec{\Delta}_f \cdot (\vec{\nabla}\phi)_f + \vec{k}_f \cdot (\vec{\nabla}\phi)_f$, where $\vec{S}_f = \vec{\Delta}_f + \vec{k}_f$. The correction methods propose different forms to find $\vec{\Delta}_f$.

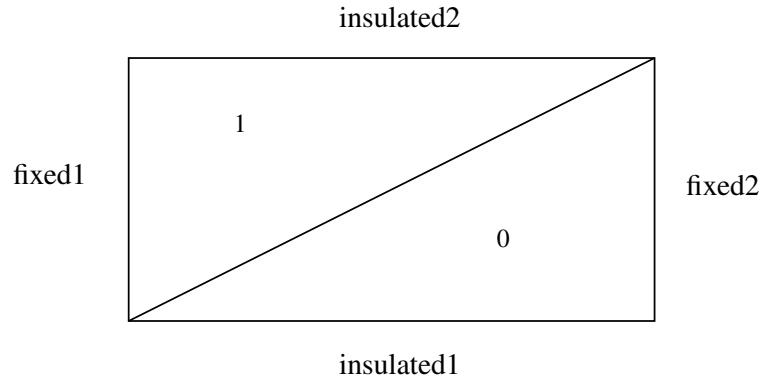


Figure A.2: Geometry and patches in Laplacian test (numbers identifies cells, figure credits to Juan M. Giménez).

Example 6 allows to verify the proper initialization of the internal field. The list shown presents the values of the field.

Example 6 View internalField values with *gdbOF*

```
(gdb) pinternalvalues T
(gdb) $1 = {0,0}
```

It can be shown analytically that the solution to this problem is a linear function $\phi(x) = ax + b$. If $\phi_{\text{fixed2}} = \phi_{\text{fixed1}}$ then $a = 0$ and the solution is constant, doing unnecessary the second term in non-orthogonal correction [$\vec{k}_f \cdot (\vec{\nabla} \phi)_f = 0$]. It allows to compare the systems generated by the different approaches in respect to the obtained in OpenFOAM®, and to determine which one is used as default.

Using minimum-correction approach ($\vec{\Delta}_f = \frac{\vec{d}_{PN} \cdot \vec{S}_f}{|\vec{d}_{PN}|} \vec{d}_{PN}$):

$$\begin{aligned} -3.29\phi_0 + 1.79\phi_1 &= -409.5 \\ 1.79\phi_0 + -3.29\phi_1 &= -409.5 \end{aligned}$$

Using orthogonal-correction approach ($\vec{\Delta}_f = \frac{\vec{d}}{|\vec{d}|} |\vec{S}|$):

$$\begin{aligned} -4.5\phi_0 + 3\phi_1 &= -409.5 \\ 3\phi_0 + -4.5\phi_1 &= -409.5 \end{aligned}$$

Using over-relaxed approach ($\vec{\Delta}_f = \frac{\vec{d}}{\vec{d} \cdot \vec{S}} |\vec{S}|^2$):

$$\begin{aligned} -5.25\phi_0 + 3.75\phi_1 &= -409.5 \\ 3.75\phi_0 + -5.25\phi_1 &= -409.5 \end{aligned}$$

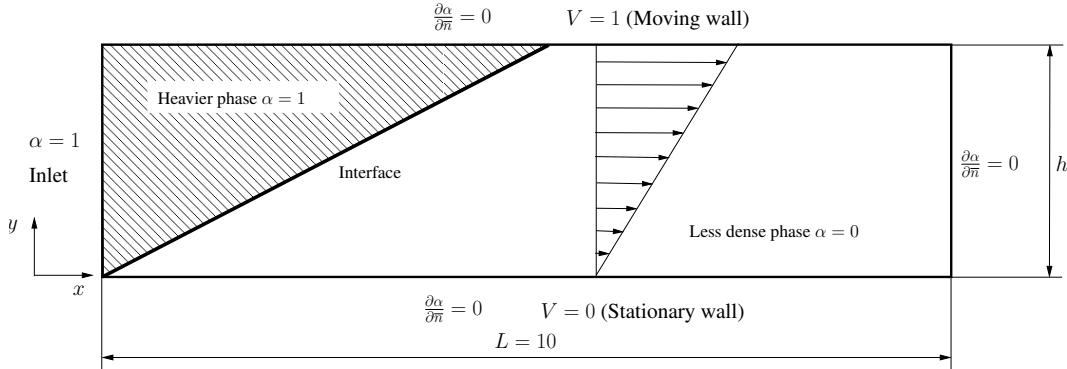
Example 7 shows how *gdbOF* extracts the equation system. Here, the reader can verify that the over-relaxed approach is implemented as default in OpenFOAM®.

Example 7 Equation System debugging in LaplacianTest

```
$ (gdb) b fvScalarMatrix.C:144
Breakpoint 1 at 0xb71455dc: file fvMatrices/fvScalarMatrix... line 144
$ (gdb) run
...
$ (gdb) pfvmatrixfull this this.txt
Saved correctly
$ (gdb) shell cat this.txt
-5.25    3.75
3.75    -5.25
(gdb) p *totalSource.v_@2
{-409.5, -409.5}
```

A.4.3 Multiphase Test

As the last example, a multiphase solver, namely `interFoam` is used showing *gdbOF* functionality. In this case a 2D reference problem is solved, which has analytical solution. Let be a rectangular domain with a Couette velocity profile (see Figure A.3), and filled with a light fluid as initial condition and the domain inlet with a heavy fluid in all its extension. The problem to solve is the evolution of the heaviest phase through the domain along the time.

Figure A.3: Geometry in `interFoam` test.

This two phase system is solved by means of a momentum equation [see Eqn. (A.11)] and an advection equation for the void fraction function α [see Eqn. A.12)] (Berberovic et al., 2009)

$$\frac{\partial \rho \vec{u}}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u} \vec{u}) - \vec{\nabla} \cdot \left(\mu \vec{\nabla} \vec{u} \right) - \left(\vec{\nabla} \vec{u} \right) \cdot \vec{\nabla} \alpha_q = -\vec{\nabla} p_{rgh} - \vec{g} \cdot \vec{x} \vec{\nabla} \rho + \sigma \kappa \vec{\nabla} \alpha_q \quad (\text{A.11})$$

$$\frac{\partial \alpha_q}{\partial t} + \vec{\nabla} \cdot (\vec{u} \alpha_q) + \vec{\nabla} \cdot [\vec{v}_{qp} \alpha_q (1 - \alpha_q)] = 0 \quad (\text{A.12})$$

In this case, $\vec{g} = 0$ and it can be shown that $\vec{\nabla} p_{rgh}$ and κ are both null (no pressure gradient is needed in a velocity driven flow and curvature vanishes due to a linear interface). Taking this in mind, an initial linear velocity profile is an spatial solution of Eqn. (A.11) so it reduces to Eqn. (A.13).

$$\frac{\partial \vec{u}}{\partial t} = 0 \quad (\text{A.13})$$

From this conclusion it is clear that streamlines are horizontal, and the heaviest phase advances more quickly as streamlines are closer to the top region, giving a linear interface front (See Figure A.3). This advancement is governed by an advective equation for the indicator function which includes

an extra term, suitable to compress the interface (OpenCFD, 2005).

Using Finite Volume Method Eqn. A.12 can be discretized as in Eqn. A.14 (Bohorquez R. de M., 2008)

$$\frac{\alpha_q^{n+1} - \alpha_q^n}{\Delta t} V + \sum_f \left\{ (\alpha_q)_f^n F^n + (\alpha_q)_f^n \left[1 - (\alpha_q)_f^n \right] F_{qp}^n \right\} = 0 \quad (\text{A.14})$$

where $F^n = \vec{u}^n \cdot \vec{S}_f$, $F_{qp}^n = \vec{v}_{qp}^n \cdot \vec{S}_f$ and superindex n implies the timestep. The compressive velocity, \vec{u}_r , and is computed directly as a flux: $F_{qp} = n_f \min \left[C_\alpha \frac{|\phi|}{|\vec{S}_f|}, \max \left(\frac{|\phi|}{|\vec{S}_f|} \right) \right]$. C_α is an adjustment constant and $n_f = \frac{(\vec{\nabla}\alpha)_f}{[(\vec{\nabla}\alpha)_f + \delta_n]} \cdot \vec{S}_f$ is the face unit normal flux with δ_n as a stabilization factor (Berberovic et al., 2009). F_{qp} values are variable only vertically in this example and will be checked at debugging time against those calculated from theory, using *gdbOF* tools. In this case, because of how the advective terms are calculated, it is necessary to show values at faces.

The domain was meshed as a 3D geometry due to OpenFOAM® requirements (OpenCFD, 2012) with a $100 \times 10 \times 1$ elements in the grid, so each hexahedron has edges of 0.1 units in size. Since its definition and taking $C_\alpha = 1$, $|\vec{v}_{qp}| = |\vec{u}|$, therefore $F_{qp} = \vec{u}_{qp} \cdot \vec{S}_f = 0.01 |\vec{v}_{qp}| (\check{\vec{v}}_{qp} \cdot \check{\vec{S}}_f)$. So taking three distances from the bottom edge of the domain, $y = 0.05$, $y = 0.45$ and $y = 0.95$, values for F_{qp} in faces with \vec{S}_f aligned with x direction must be $|F_{qp}| = 0.005$, $|F_{qp}| = 0.045$ and $|F_{qp}| = 0.095$ respectively.

Again, it is necessary to find the indexes of three cells with such y coordinates, taking for example $x = 0.05$, and using `pFindCell` tool the results shown in Example 8 can be obtained.

As it was explained in Section A.3.2 using only the index of the cell is not enough to address the values in a `surfaceField` of a given field. Each cell has

Example 8 View cell index in multiphase problem.

```
(gdb) pfindcell 0.05 0.45 0.05
RESULTS:
Nearest cell centroid cell number: 400
Containing point cell number (-1=out) : 400
```

as many surface values as faces in the cell, therefore it is necessary to show all these values, extracting the information from faces whose indexes are not necessarily correlative. The `psurfacevalues` *gdbOF* command simplifies this task. Knowing the index of the cell to analyze, it returns the information on each face about the field indicated in the command line parameters: boundary face or internal face (categorized according to whether it has a neighbour or not) and field value. If it is working with a 2D mesh, information is also returned as in a 3D mesh, but it indicates which of these faces has an empty boundary condition (see *gdbOF User's Manual*, Appendix C or the Subsection A.3.2).

Example 9 Example of usage of `psurfacevalues` for face defined field.

```
(gdb) psurfacevalues phir 400
internal Face:
$5 = 0
internal Face:
$6 = -0.0045
internal Face:
$7 = 0
empty Face
empty Face
boundary Face:
$8 = 0.0045
```

Therefore, applying this command to the cell previously found, makes it possible to show ϕ_{rf} in all faces of that cell (See Example 9). Results are consistent with the original problem. Two faces are marked as *empty* because the mesh has only one cell in depth. This boundary condition is used by OpenFOAM® to represent no variability in direction perpendicular to the face, allowing a 2D calculation. Faces 5 and 7 corresponds to top and bottom

faces of the cell where flux is null. Finally, faces 6 and 8 have faces with normals aligned with the velocity and the flux values are those predicted theoretically for $y = 0.45$. Values have different sign due to the normals orientation.

Regarding graphical debugging presented in Section A.3.3 `pexportfoamformat` is a useful tool to inspect the α field as in Figure A.3. To do so, command is invoked as in Example 10 and results are shown in Figure A.4.

Example 10 Field exporting to .vtk by means of `pexportfoamformat`. Paraview® is invoked as well

```
(gdb) pexportfoamformat alpha1 VTK Paraview
Including internal field...
Including boundary field(s)
fixedWall
movingWall
inlet
outlet
frontAndBackPlanes
-----
Field saved to gdbOF_dump/alpha1.0.dump
-----
Exporting to VTK...
Launching Paraview...
```

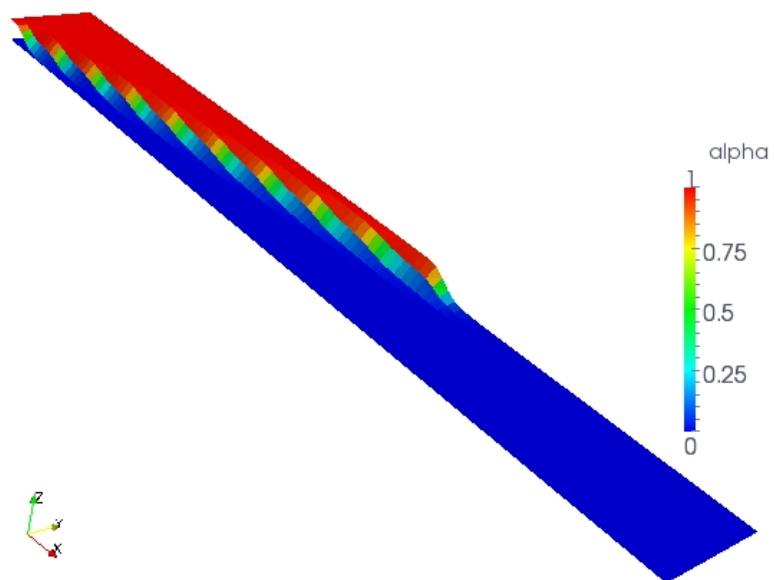


Figure A.4: α field representation in Paraview[®] using `pexportfoamformat` (with `VTK` option).

Appendix B

Resumen extendido en castellano

B.1 Un modelo de mezcla extendido para el tratamiento simultáneo de interfases de pequeña y larga escala

B.1.1 Antecedentes

El estudio de los sistemas multifluido y multifásicos es de gran interés en el ambiente científico y tecnológico. La representación de este tipo de sistemas es clave en la industria automotriz, energía atómica, petroquímica, hidráulica, etc.

Para el abordaje de simulación de estos fenómenos se recurre generalmente a diversos modelos, entre los más utilizados se encuentran: la Simulación Numérica Directa (DNS), el Método de Volumen de Fluido (VOF) (Hirt and Nichols, 1981), el Modelo Multifluido (Drew, 1983) y el Modelo de Mezcla (Manninen et al., 1996). En la Simulación Numérica Directa aplicada a flujos multifásicos el modelo es capaz de representar todas las escalas geométricas de las distintas fases así como también las de la turbulencia. Para ello, las ecuaciones completas de momento y conservación de masa se resuelven en cada fase, imponiendo en las interfases y fronteras las

condiciones de borde correspondientes. Es fácil ver que la aplicabilidad de esta técnica queda circunscripta a casos simples o bien a problemas de laboratorio, dado los grandes recursos computacionales que requiere, al menos en la actualidad.

En el caso del método del Volumen de Fluido el enfoque geométrico es similar al DNS, siendo el tamaño de la malla el parámetro que determina qué escalas serán representadas. Respecto de la turbulencia, por lo general esta es tratada utilizando el filtrado, ya sea temporal (RANS) o espacial (LES).

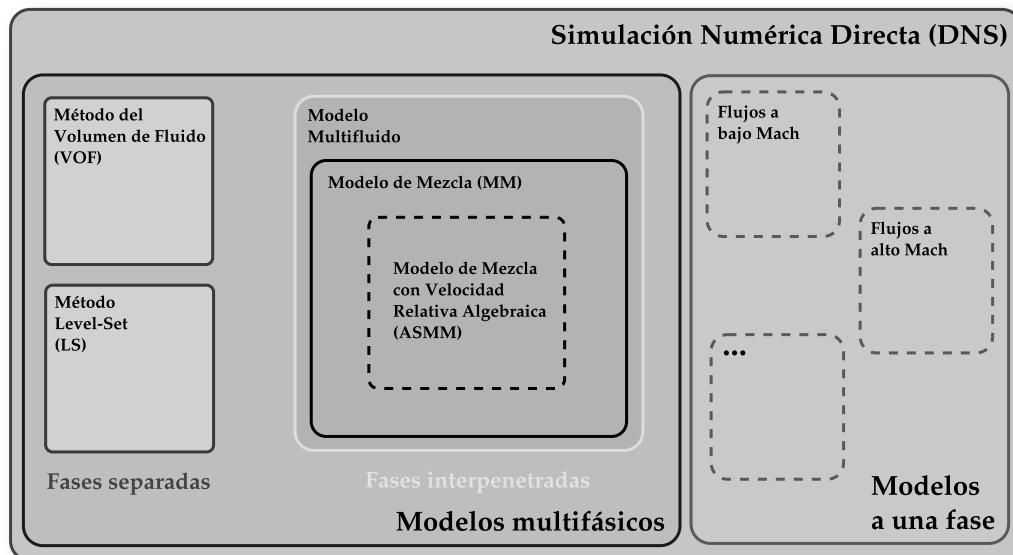


Figure B.1: Relación entre los diferentes tipos de modelos utilizados en flujo multifásico.

El Modelo Multifluido representa un grado de simplificación mayor que los dos casos anteriores. Este se utiliza, por lo general, cuando ya no es posible capturar mediante la malla las estructuras geométricas del flujo, considerando que las distintas fases, sólidas, líquidas y gaseosas se interpenetran. A tal fin se resuelven ecuaciones de momento y conservación de masa para cada fase, en las que, desde ya, aparecen términos que representan el intercambio de momento entre fases (no considerando el cambio de fase ni el intercambio de energía).

Finalmente el Modelo de Mezcla se vale de una simplificación adicional, esta es que las distintas fases interpenetradas pueden considerarse directamente una mezcla, para la cual se resuelve una sola ecuación de momento, y una ecuación de conservación de masa por fase. Las propiedades físicas utilizadas en las ecuaciones son las de la mezcla (caracterizada por las distintas fracciones de volumen).

Cabe destacar que si bien el Modelo de Mezcla representa una simplificación respecto del Modelo Multifluido, el grado de aplicación de cada uno y la calidad de los resultados depende, en gran medida, del problema. Una gran debilidad del Modelo Multifluido reside en su mala postura matemática (Zanotti et al., 2007) y en la falta de ecuaciones de cierre para los términos de transferencia de momento entre fases (Manninen et al., 1996).

Así pues el modelo VOF es utilizado en problemas donde la captura de interfase es crucial, manifestándose con importancia los fenómenos de tensión superficial y adhesión (formación de gotas, fenómenos de capilaridad, *break-up* de *jets*, etc.) o bien en casos donde la predicción de la posición de la superficie libre es de interés primordial (toberas, problemas de hidráulica de superficies libres, industria naval, reservorios, etc.). En estos casos las interfaces se consideran como de gran escala, tomando como referencia alguna medida del tamaño de la malla (véase Figura B.2.a).

En el caso de los modelos Multifluido y de Mezcla su interés radica en su capacidad de predecir el comportamiento de flujos con interfaces de pequeña escala –“interfaces dispersas”–(véase Figura B.2.b) cuando no es posible o deseable un modelado completo. Este tipo de interfaces se presenta por lo general en problemas de tanques de sedimentación, separación por ciclones, flujo anular en refinerías, flujos con burbujas finas en intercambiadores de calor, etc.

Tal como se han presentado, la Simulación Numérica Directa representa

el único modelo con el que es posible abordar problemas de fluidos en general (Scardovelli and Zaleski, 1999; Tryggvason et al., 2006), sin embargo, las limitaciones actuales en recursos computacionales no hacen posible su aplicación directa. Por otra parte, dado su falta de generalidad, el resto de los modelos funciona normalmente en compartimentos estancos, atendiendo al cumplimiento de las hipótesis asumidas en su formulación.

En esta situación queda abierta la discusión respecto del desarrollo de nuevos modelos que permitan abordar fenómenos donde las escalas de las interfases son variadas, o bien donde se producen transiciones entre unas y otras (véase Figura B.2.c). Se plantean así casos como el flujo anular con niebla (*annular mist flow* ó *droplet annular flow*), la transición de flujo tapón a flujo con burbujas (*churn flow to bubbly flow*) (Ishii and Hibiki, 2010) y la evolución del fluido en un *jet* antes y después del *break-up* (Masuda and Nagaoka, 2006), etc.

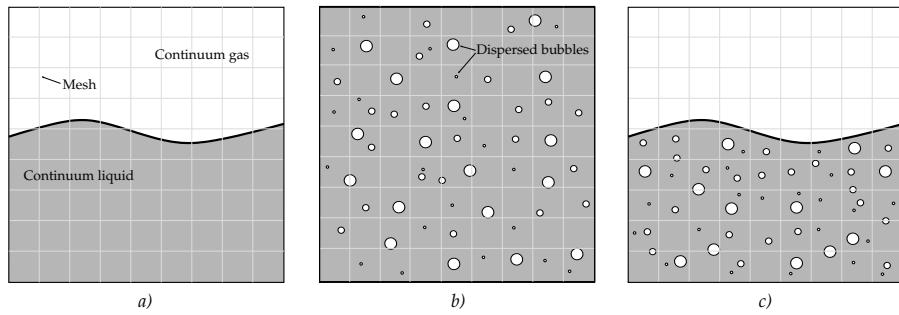


Figure B.2: Representación de interfaces de pequeña y gran escala en un flujo con burbujas. a) Interfaces de gran escala, b) interfaces de pequeña escala, c) presencia de interfaces de ambas escalas en simultáneo.

Respecto a estos casos, el estado del arte arroja aún pocas respuestas. Uno de los primeros trabajos en el tema corresponde a Cerne *et. al.* (Cerne et al., 2001). En el se plantea el acoplamiento del modelo VOF con el Multifluido (Dos Fluidos en este caso) a los efectos de resolver el problema de la inestabilidad de Rayleigh-Taylor. Ambos modelos son tratados por separado,

tomando un criterio de selección para uno u otro mediante la detección de interphasas de pequeña o gran escala. Más adelante Strubelj y Tiselj (Štrubelj and Tiselj, 2011) continúan la línea de trabajo anterior utilizando para la captura de interfase el Método de Conjunto de Niveles (Level Set) y escribiendo el modelo obtenido completamente en términos del modelo de dos fluidos.

Otros autores como Yan y Che (Yan and Che, 2010) y Masuda y Nagaoka (Masuda and Nagaoka, 2006) avanzan sobre la escritura de las ecuaciones para VOF y Dos Fluidos en forma acoplada, cerrando el modelo por medio de los términos de transferencia de momento y masa. Esta metodología es utilizada para resolver flujos con burbujas de pequeña y gran escala y *break-up* de *jets*.

Un aporte de interés es el propuesto por Bohorquez (Bohorquez R. de M., 2008). En él, el Modelo de Mezcla es acoplado a VOF para la resolución de flujos a superficie libre con sedimentos en suspensión en la fase líquida. Se plantean así interphasas de larga escala (líquido-gas) y de pequeña escala (sedimento-líquido). El sistema se resuelve utilizando una formulación de mezcla general donde la clave reside en la definición de las velocidades relativas entre fases.

Teniendo en cuenta entonces el estado actual del arte se propone la formulación de un Modelo de Mezcla Extendido, tal que sea capaz de capturar las interphasas de larga escala (mediante una metodología similar a VOF) y aquellas de pequeña escala para las cuales el modelo se reduce al Modelo de Mezcla clásico. Estas estrategias se desarrollan en un entorno de trabajo dado por la librería OpenFOAM®, la cual es una herramienta de computación paralela orientada a objetos, capaz de manejar mallas estructuradas y no estructuradas en dos y tres dimensiones y atendiendo a la modelización de la turbulencia. En el contexto actual presentado estas estrategias se consideran innovadoras y originales, realizando un aporte significativo al estado del arte. Este tipo de estrategias se encuentra actualmente en discusión y, particularmente en el caso del VOF combinado con el Modelo de Mezcla, no se encuentran publicaciones salvo la citada.

Así pues, la formulación en el continuo gas-líquido resulta ser (Bohorquez R. de M., 2008; Manninen et al., 1996):

$$\left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho_m \vec{v}_m) = 0 \\ \frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{v}_m \alpha_q) + \nabla \cdot [\alpha_q (1 - \alpha_q) \vec{v}_{qp}] = 0 \\ \frac{\partial \rho_m \vec{v}_m}{\partial t} + \nabla \cdot (\rho_m \vec{v}_m \otimes \vec{v}_m) = -\nabla p_{rgh} + \nabla \cdot (\bar{\bar{\tau}}) - \sigma \kappa \nabla \alpha - \vec{g} \cdot \vec{x} \nabla \rho_m \end{array} \right. \quad (B.1)$$

donde la primera de las ecuaciones representa la conservación de masa, la segunda el transporte de la fracción de volumen para la fase líquida y la tercera la conservación de cantidad de movimiento, $\rho_m \equiv \alpha_q \rho_q + (1 - \alpha_q) \rho_q$ la densidad de la mezcla, $\vec{v}_m \equiv [\alpha_q \rho_q \vec{v}_q + (1 - \alpha_q) \rho_p \vec{v}_p] / \rho_m$ la velocidad de la mezcla o velocidad del centro de masa, α_q es la fracción de volumen de la fase primaria, $\vec{u} \equiv \alpha_q \vec{v}_q + (1 - \alpha_q) \vec{v}_p$ es el flujo volumétrico de la mezcla, $\vec{u}_{qp} \equiv \vec{v}_q - \vec{v}_p$ la velocidad relativa entre fases (resultante de una ley de cierre relacionada con el fenómeno físico en estudio), $p_{rgh} \equiv p - \rho \vec{g} \cdot \vec{x}$ es la presión reducida, σ la tensión superficial actuante en las interfases de larga escala, κ la curvatura de dichas interfases, \vec{g} la gravedad y \vec{x} el vector posición.

La relación entre la velocidad del centro de masa \vec{v}_m y el flujo volumétrico \vec{u} está dada por:

$$\vec{v}_m = \vec{u} + \alpha_q (1 - \alpha_q) \frac{\rho_q - \rho_p}{\rho_m} \vec{v}_{qp} \quad (B.2)$$

Respecto al tensor de tensiones $\bar{\bar{\tau}}$, éste puede definirse como:

$$\bar{\bar{\tau}} = \bar{\bar{\tau}}' + \bar{\bar{\tau}}'' + \bar{\bar{\tau}}''' \quad (B.3)$$

donde $\bar{\bar{\tau}}'$ representa el tensor viscoso, $\bar{\bar{\tau}}''$ el tensor de esfuerzos debidos a la turbulencia y $\bar{\bar{\tau}}'''$ el tensor de difusión de momento debido al movimiento relativo del gas respecto del líquido. Este último se define como:

$$\vec{\tau}''' \equiv \alpha_q (1 - \alpha_q) \frac{\rho_q \rho_p}{\rho_m} \vec{v}_{qp} \otimes \vec{v}_{qp} \quad (\text{B.4})$$

Si en el planteo de las ecuaciones (B.1--B.4) tal como se han presentado se consideran las interfases de larga escala solamente y el ancho de estas como infinitésimo, con continuidad en velocidades ($\vec{v}_{qp} \equiv 0$), el sistema se reduce al método VOF clásico. Si la velocidad relativa se considera nula solamente en las ecuaciones de momento, pero se retiene el término no lineal de la ecuación transporte de la fracción de volumen se tiene el método VOF tal como se implementa en OpenFOAM® (Berberovic et al., 2009) (Weller-VOF). Este término no lineal artificial permite la compresión de la interfase más allá de los esquemas de advección utilizados para tratar el término lineal de divergencia. En esta formulación el valor de la velocidad relativa \vec{u}_{qp} queda como parámetro libre seleccionándose tal que comprima la interfase de manera adecuada.

A los fines de plantear una formulación extendida se retienen entonces las ecuaciones completas, sin embargo, dependiendo se trate de interfaces de pequeña o gran escala, se utiliza la definición física de \vec{v}_{qp} o la numérica respectivamente (para el caso del Figura B.2, refiere a aquellas burbujas más pequeñas o más grandes al tamaño característico de la malla). El mismo criterio de selección se utiliza para aplicar el modelo de mezcla completo o la implementación de Weller-VOF. Se hace necesario entonces adoptar o diseñar un criterio de selección de modelos basado en las características geométricas del flujo. Uno de los disponibles es el criterio de Cerne (Černe et al., 2001), basado en el análisis de un cuadro compuesto por una celda y todas su vecinas por caras y aristas.

B.1.2 Desarrollo

La implementación y validación del modelo de mezcla extendido propuesto requiere de varios bloques constructivos, muchos de los cuales se encuentran disponibles actualmente, sin embargo es necesario también diseñar o adaptar algunos de ellos. El marco de desarrollo mediante el Método de Volúmenes

Finitos en entornos de alto rendimiento se encuentra dado por la herramienta OpenFOAM®. Si bien se trata de una herramienta de código abierto, las bases numéricas y su implementación no se encuentran completamente documentadas, con lo cual se hace necesario la revisión de algunos conceptos que se presentan en el capítulo 2. El desarrollo de aplicaciones requiere la posibilidad de depuración para lo cual se implementó un entorno utilizable dentro de la herramienta gdb la cual es presentada en el apéndice A.

Dado que se plantea la implementación de un modelo de mezcla extendido basado en el modelo de mezcla clásico, se hace necesario entonces una revisión profunda de éste método, desde su derivación desde el método de Dos Fluidos hasta su solución en forma semi-analítica. Es aquí donde se realiza un aporte en la conceptualización del método y en la generación de soluciones semi-analíticas que permitan la posterior validación de los métodos numéricos propuestos. Vale aclarar que la correcta solución del sistema dado por el Método de Mezcla está fuertemente ligada al tratamiento de las ecuaciones de conservación de masa de las fases dispersas. Estas ecuaciones son de carácter hiperbólico presentando en su solución ondas de choque y rarefacción y ondas compuestas. Es necesario pues abordar la solución de este tipo de ecuaciones, conocer su estructura de autovalores y los métodos necesarios para capturar las ondas que se presentan. Esta discusión se presenta en el capítulo 3.

Siendo que las ecuaciones de conservación de masa para las fases dispersas se resuelven en un contexto dado por un *solver* conservativo, donde el correcto manejo del flujo en las caras es vital para lograr soluciones acotadas, se hace necesario entonces adaptar los métodos conocidos al uso de flujos previamente calculados en caras. La solución de este tipo de problemas es conocida en el campo de los limitadores tipo Flux Corrected Transport (FCT), pero no en el caso de los *solvers* tipo *Riemann-free*. Así pues en el capítulo 4 se propone un nuevo método Riemann-free basado en la reconstrucción de valores en caras utilizando métodos de alta resolución y el uso de un flujo previamente calculado en las caras, dado por el módulo de acoplamiento entre presión y velocidad. Este método se extiende además al caso multidimensional. Por otra

parte, dado que en algunas soluciones se permite la acumulación de una fase más allá de los máximos y mínimos de la condición inicial, se hace necesario ajustar el método FCT para trabajar con extremos globales.

La unificación del método VOF con el modelo de mezcla requiere primeramente la escritura de ambos métodos en un entorno unificado. Se procede entonces en el capítulo 5 a derivar el método VOF a partir del modelo mezcla, escribiendo éste último en una formulación combinada entre velocidad de centro de masa y velocidad de centro de volumen (flujo volumétrico) [véase Eqn. (B.1)]. Una vez que se ha encontrado la formulación de VOF se presenta el algoritmo implementado en el solver `interFoam` de OpenFOAM® tal como se encuentra en la distribución original del software. Este solver es utilizado para resolver dos casos paradigmáticos en la problemática multifásica como son la inestabilidad de Rayleigh-Taylor y la rotura de presa (Dam-Break).

En el capítulo 6 se presenta la implementación de un *solver* de mezcla basado en esquemas Riemann-free y en métodos de alta resolución. Esta herramienta es capaz de resolver con precisión problemas académicos centrados en la presencia de ondas de diferente tipo, como así también problemas de tipo industrial, lo cual es validado en los ejemplos presentados.

Habiendo revisado y comprendido los conceptos de base y desarrollado las herramientas faltantes se aborda finalmente la solución de problemas multifásicos atendiendo las diferentes escalas de la interfase. Para ello se hace una revisión del estado del arte actual, resaltando las fortalezas y deficiencias de cada método. La motivación para el desarrollo de métodos extendidos surge de la falta de resolución que presentan los métodos de captura o seguimiento de interfase, esta fenomenología resulta evidente del análisis pormenorizado de los problemas resueltos mediante el método VOF en el capítulo 5. Se concluye entonces que una vez que las estructuras de fluido caen por debajo de la resolución de la malla comienzan a presentarse problemas de tensión superficial numérica y errores en el cálculo de las fuerzas de flotación. Esto último se debe a que no es posible mantener los valores de la fracción de volumen en los

extremos de 0 y 1 (indicando un fluido u otro), apareciendo entonces nuevos fluidos con densidades y viscosidades dadas por las propiedades de la mezcla.

Se presenta entonces un modelo de mezcla extendido capaz de capturar las interfas es de larga escala mediante el uso de la técnica VOF, mientras que la dinámica de las escalas no resueltas por la malla es dada por el método de mezcla. Para ello es necesario determinar que zona del dominio deben ser resueltas con un modelo u otro, lo cual se logra mediante una función indicadora basada en el gradiente de la fracción de volumen.

El modelo extendido es entonces aplicado a cuatro casos de prueba: una pluma de burbujas en un tanque, un reactor de burbujas, el caso de la rotura de presa y finalmente la inestabilidad de Rayleigh-Taylor. En el caso del reactor de burbujas se propone además un modelo unidimensional que permite tener un acercamiento al problema completo tanto en la fracción de volumen como en la posición de la superficie libre. Los resultados demuestran que el modelo extendido brinda una respuesta satisfactoria a los problemas de interfas es de múltiple escala logrando un reducción en los requerimientos del tamaño de malla y mayor libertad en las condiciones de borde.

Bibliography

- Abramson, D., Foster, I., Michalakes, J., Sosić, R.: Relative debugging: A new methodology for debugging scientific applications. *Communications of the ACM* **39**(11), 69–77 (1996)
- Babik, F., Gallouët, T., Latché, J., Suard, S., Vola, D.: On two fractional step finite volume and finite element schemes for reactive low mach number flows. In: *The International Symposium on Finite Volumes for Complex Applications IV-Problems and Perspectives-Marrakech* (2005)
- Battaglia, L., Storti, M.A., D'Elía, J.: Simulation of free-surface flows by a finite element interface capturing technique. *International Journal of Computational Fluid Dynamics* **24**(3), 121–133 (2010)
- Berberovic, E., van Hinsberg, N., Jakirlic, S., Roisman, I., Tropea, C.: Drop impact onto a liquid layer of finite thickness: dynamics of the cavity evolution. *Physical Review E* **79** (2009)
- Bohorquez, P.: Finite volume method for falling liquid films carrying monodisperse spheres in newtonian regime. *AIChE Journal* **58**(8), 2601–2616 (2012)
- Boris, J., Book, D.: Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works. *Journal of Computational Physics* **11**, 38–69 (1973)
- Brackbill, J., Kothe, D., Zemach, C.: A continuum method for modeling surface tension. *Journal of Computational Physics* **100**(2), 335–354 (1992)
- Brennan, D.: The numerical simulation of two-phase flows in settling tanks. Ph.D. thesis, Department of Mechanical Engineering Imperial College of Science, Technology and Medicine (2001)

- Bürger, R., Wendland, W.: Sedimentation and suspension flows: Historical perspective and some recent developments. *Journal of Engineering Mathematics* **41**(2), 101--116 (2001)
- Buscaglia, G., Bombardelli, F., Garcia, M.: Numerical modeling of large-scale bubble plumes accounting for mass transfer effects. *International Journal of Multiphase flow* **28**(11), 1763--1785 (2002)
- Carrica, P., Wilson, R., Stern, F.: An unsteady single-phase level set method for viscous free surface flows. *International Journal for Numerical Methods in Fluids* **53**(2), 229--256 (2006)
- Cartland Glover, G., Generalis, S.: The modelling of buoyancy driven flow in bubble columns. *Chemical Engineering and Processing: Process Intensification* **43**(2), 101--115 (2004)
- Cary, J., Shasharina, S., Cummings, J.: Comparison of C++ and Fortran 90 for object-oriented scientific programming. *Computer Physics Communications* **105**(1), 20--36 (1997)
- Choi, S.: Note on the use of momentum interpolation method for unsteady flows. *Numerical Heat Transfer: Part A: Applications* **36**(5), 545--550 (1999)
- Cloete, S., Olsen, J., Skjetne, P.: CFD modeling of plume and free surface behavior resulting from a sub-sea gas release. *Applied Ocean Research* **31**(3), 220--225 (2009)
- Coquel, F., El Amine, K., Godlewski, E., Perthame, B., Rascle, P.: A numerical method using upwind schemes for the resolution of two-phase flows. *Journal of Computational Physics* **136**(2), 272--288 (1997)
- Cruchaga, M., Celentano, D., Tezduyar, T.: Collapse of a liquid column: numerical simulation and experimental validation. *Comput. Mech.* **39**, 453--476 (2007)
- Cruz, D., Henriques, P., Pereira, M.: Alma versus DDD (2008)

- Dagaleesan, S.: Fluid dynamic measurements and modelling of liquid mixing in bubble columns. Ph.D. thesis, Washington University, Saint Luis (1997)
- Darwish, M., Moukalled, F.: TVD schemes for unstructured grids. International Journal of Heat and Mass Transfer **46**(4), 599--611 (2003)
- De Villiers, E.: The potential of large eddy simulation for the modeling of wall bounded flows. Ph.D. thesis, Imperial College, London (2006)
- Dewar, A., Cleary, J.: Graphical display of complex information within a prolog debugger. International Journal of Man-Machine Studies **25**(5), 503--521 (1986)
- Drew, D.: Mathematical Modeling of Two-phase Flow. Annual Review of Fluid Mechanics **15**(1), 261--291 (1983)
- Eckel, B.: Thinking in C++, vol. 1 Introduction to Standard C++. 2nd. edn. Prentice Hall Inc. (2000)
- Evje, S., Flåtten, T.: Hybrid flux-splitting schemes for a common two-fluid model. Journal of Computational Physics **192**(1), 175--210 (2003)
- Ewer, J., Knight, B., Cowell, D.: Case study: an incremental approach to re-engineering a legacy fortran computational fluid dynamics code in C++. Advances in Engineering Software **22**(3), 153--168 (1995)
- Ferziger, J., Peric, M.: Computational Methods for Fluid Dynamics, vol. I. Springer-Verlag (2002)
- Fluent Inc.: Fluent 6.3. User's Guide. Fluent Inc. (2006)
- Friedl, M.: Bubble plumes and their interactions with the water surface. Ph.D. thesis, Swiss Federal Institute of Technology (ETH), Zürich (1998)
- Friedl, M., Fanneløp, T.: Bubble plumes and their interaction with the water surface. Applied Ocean Research **22**(2), 119--128 (2000)

- Gastaldo, L., Herbin, R., Latché, J.: An entropy preserving finite-element/finite-volume pressure correction scheme for the drift-flux model. Arxiv preprint arXiv:0803.2469 (2008)
- Gastaldo, L., Herbin, R., Latché, J.: A discretization of the phase mass balance in fractional step algorithms for the drift-flux model. IMA Journal of Numerical Analysis **31**(1), 116--146 (2011)
- Gopala, V., van Wachem, B.: Volume of fluid methods for immiscible-fluid and free-surface flows. Chemical Engineering Journal **141**(1-3), 204--221 (2008)
- Gray, C., Gubbins, K.: Theory of molecular fluids, vol 1. Clarendon. Oxford (1984)
- Greenshields, C., Weller, H., Gasparini, L., Reese, J.: Implementation of semi-discrete, non-staggered central schemes in a colocated, polyhedral, finite volume framework, for high-speed viscous flows. International Journal for Numerical Methods in Fluids **63**(1), 1--21 (2010)
- Grimm, V.: Visual debugging: A way of analyzing, understanding and communicating bottom-up simulation models in ecology. Natural Resource Modeling **15**(1), 23--38 (2002)
- Harten, A.: High resolution schemes for hyperbolic conservation laws. Journal of Computational Physics **49**(3), 357--393 (1983)
- Hirsch, C.: Numerical computation of internal and external flows: fundamentals of computational fluid dynamics, vol. 1. Butterworth-Heinemann (2007)
- Hirt, C., Nichols, B.: Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comput. Phys. **39**, 201--225 (1981)
- Ishii, M.: Thermo-Fluid Dynamic Theory of Two-phase Flow. NASA STI/Recon Technical Report A **75** (1975)
- Ishii, M., Hibiki, T.: Thermo-fluid dynamics of two-phase flow. Springer Verlag (2010)

- Issa, R.: Solution of implicitly discretised fluid flow equations by operator splitting. *Journal of Computational Physics* **62**, 40--65 (1986)
- Jasak, H.: Error analysis and estimation for the finite volume method with applications to fluid flows. Ph.D. thesis, Department of Mechanical Engineering Imperial College of Science, Technology and Medicine (1996)
- Jasak, H., Weller, H., Gosman, A.: High resolution NVD differencing scheme for arbitrarily unstructured meshes. *International Journal for Numerical Methods in Fluids* **31**(2), 431--449 (1999)
- Knio, O., Najm, H., Wyckoff, P.: A semi-implicit numerical scheme for reacting flow: II. stiff, operator-split formulation. *Journal of Computational Physics* **154**(2), 428--467 (1999)
- Korn, J., of Computer Science, P.U.D.: Abstraction and visualization in graphical debuggers. Princeton University Princeton, NJ, USA (1999)
- Kurganov, A., Tadmor, E.: New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. *Journal of Computational Physics* **160**(1), 241--282 (2000)
- Kuzmin, D., Moller, M., Turek, S.: Multidimensional FEM-FCT schemes for arbitrary time stepping. *International Journal for Numerical Methods in Fluids* **42**(3), 265--295 (2003)
- Kynch, G.: A theory of sedimentation. *Transactions of the Faraday Society* **48**, 166--176 (1952)
- van Leer, B.: Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of Computational Physics* **32**(1), 101--136 (1979)
- Leonard, B.P.: The ultimate conservative difference scheme applied to unsteady one-dimensional advection. *Computer Methods in Applied Mechanics and Engineering* **88**, 17--74 (1991)

LeVeque, R.: Finite volume methods for hyperbolic problems. Cambridge Univ Press (2002)

L.F.Barceló, P.A., C., Larreteguy, A., Gayoso, R., Gayoso, F., Lavalle, G.: Análisis del comportamiento de equipos separadores de agua-petróleo usando volúmenes finitos y el modelo de drift-flux. In: Mecánica Computacional Vol XXIX (2010), pp. 8463--8480 (2010)

Bohorquez R. de M., P.: Study and Numerical Simulation of Sediment Transport in Free-Surface Flow. Ph.D. thesis, Málaga University, Málaga (2008)

Mangani, L., Bianchini, C., Andreini, A., Facchini, B.: Development and validation of a C++ object oriented CFD code for heat transfer analysis. In: ASME-JSME, Thermal Engineering and Summer Heat Transfer Conference (2007)

Manninen, M., Taivassalo, V., Kallio, S.: On the mixture model for multiphase flow. Technical Research Centre of Finland (1996)

Marinescu, D.: Stl-views-1.0.3.gdb (2008)

Márquez Damián, S., Giménez, J., Nigro, N.: gdbOF: A debugging tool for openFOAM®. Advances in Engineering Software **47**(1), 17--23 (2012)

Márquez Damián, S., Nigro, N., Buscaglia, G.: Solving the algebraic slip mixture model as a hyperbolic system with restrictions. In: Mecánica Computacional Vol XXXI, pp. 651--671 (2012)

Márquez Damián, S., Nigro, N.M.: An approach to the algebraic slip mixture model using OpenFOAM(r). In: Mecánica Computacional Vol XXX, pp. 2257--2257 (2011)

Martin, J., Moyce, W.: An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane. Philos. Trans. R. Soc. Lond. **244**, 312--324 (1952)

- Masuda, R., Nagaoka, M.: A Coupled Interface-capturing and Multi-fluid Model Method for Computing Liquid Jet from Nozzle Flow. In: International Congress on Liquid Atomization and Spray Systems (2006)
- Mathur, S., Murthy, J.: A pressure-based method for unstructured meshes. Numerical Heat Transfer **31**(2), 195–215 (1997)
- Matloff, N., Salzman, P.: The Art of Debugging with GDB, DDD, and Eclipse. 1st edn. No Starch Press (2008)
- Najm, H., Wyckoff, P., Knio, O.: A semi-implicit numerical scheme for reacting flow: I. stiff chemistry. Journal of Computational Physics **143**(2), 381--402 (1998)
- Nessyahu, H., Tadmor, E.: Non-oscillatory central differencing for hyperbolic conservation laws. Journal of Computational Physics **87**(2), 408--463 (1990)
- Nigam, M.: Numerical simulation of buoyant mixture flows. International Journal of Multiphase Flow **29**(6), 983–1015 (2003)
- OpenCFD: OpenCFD Technical report no. TR/HGW/02 (2005)
- OpenCFD: OpenFOAM, The Open Source CFD Toolbox, User Guide. OpenCFD Ltd. (2012)
- Parker, G., Franck, G., Ware, C.: Visualization of large nested graphs in 3D: Navigation and interaction. Journal of Visual Languages and Computing **9**(3), 299–317 (1998)
- Peng Karrholm, F.: Numerical Modelling of Diesel Spray Injection, Turbulence Interaction and Combustion. Ph.D. thesis, Chalmers University of Technology, Goteborg (2008)
- Pilon, L., Viskanta, R.: Minimum superficial gas velocity for onset of foaming. Chemical Engineering and Processing: Process Intensification **43**(2), 149–160 (2004)

- Puckett, E., Almgren, A., Bell, J., Marcus, D., Rider, W.: A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *Journal of Computational Physics* **130**(2), 269–282 (1997)
- Rhie, C.M., Chow, W.L.: Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* **21**(11), 1525–1532 (1983)
- Rider, W., Kothe, D.: Reconstructing volume tracking. *Journal of Computational Physics* **141**(2), 112–152 (1998)
- Rudman, M.: Volume-tracking Methods for Interfacial Flow Calculations. *International Journal for Numerical Methods in Fluids* **24**(7), 671–691 (1997)
- Rusche, H.: Computational fluid dynamics of dispersed two-phase flows at high phase fractions. Ph.D. thesis, Imperial College of Science, Technology and Medicine, London (2002)
- Saad, Y.: Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics (2003)
- Sanyal, J., Roy, S., Dudukovic, M., et al.: Numerical simulation of gas-liquid dynamics in cylindrical bubble column reactors. *Chemical Engineering Science* **54**(21), 5071–5083 (1999)
- Scardovelli, R., Zaleski, S.: Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* **31**(1), 567–603 (1999)
- Schiller, L., Naumann, Z.: A drag coefficient correlation. *Z. Ver. Deutsch. Ing.* **77**, 318 (1935)
- Shimomura, T., Isoda, S.: Linked-list visualization for debugging. *Software, IEEE* **8**(3), 44–51 (1991)
- Squillacote, A., Ahrens, J.: The Paraview guide. Kitware (2006)
- Sweby, P.: High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis* pp. 995–1011 (1984)

- Toro, E.: Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Springer Verlag (2009)
- Trontin, P., Vincent, S., Estivalezes, J., Caltagirone, J.: Detailed comparisons of front-capturing methods for turbulent two-phase flow simulations. International Journal for Numerical Methods in Fluids **56**(8), 1543--1549 (2008)
- Tryggvason, G., Esmaeeli, A., Lu, J., Biswas, S.: Direct Numerical Simulations of Gas/liquid Multiphase Flows. Fluid Dynamics Research **38**(9), 660--681 (2006)
- Černe, G., Petelin, S., Tiselj, I.: Coupling of the Interface Tracking and the Two-fluid Models for the Simulation of Incompressible Two-phase Flow. Journal Computational Physics **171**, 776--804 (2001)
- Versteeg, H., Malalasekera, W.: An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Prentice Hall (2007)
- Štrubelj, L., Tiselj, I.: Two-fluid Model with Interface Sharpening. International Journal for Numerical Methods in Engineering **85**(5), 575--590 (2011)
- Waddle, V.: Graph layout for displaying data structures. In: Graph Drawing, pp. 98--103. Springer (2001)
- Wallis, G.: One-dimensional Two-phase Flow, vol. 1. McGraw-Hill New York (1969)
- Weller, H.: A new approach to VOF-based interface capturing methods for incompressible and compressible flow (2008)
- Weller, H.G., Tabor, G., Jasak, H., Fureby, C.: A tensorial approach to computational continuum mechanics using object-oriented techniques. Computer in Physics **12**(6), 620--631 (1998)
- Yan, K., Che, D.: A coupled model for simulation of the gas-liquid two-phase flow with complex flow patterns. International Journal of Multiphase Flow **36**(4), 333--348 (2010)

Zalesak, S.: Fully multidimensional Flux-Corrected Transport algorithms for fluids. *Journal of Computational Physics* **31**, 335--362 (1979)

Zanotti, A., Mendez, C., Nigro, N., Storti, M.: A Preconditioning Mass Matrix to Avoid the Ill-posed Two-fluid Model. *Journal of Applied Mechanics* **74**, 732 (2007)

Zanotti, A.L.: Modelado del flujo multifase en la produccion de acero por colada continua. Ph.D. thesis, Facultad de Ingeniería y Ciencias Hídricas, UNL. Santa Fe, Argentina (2007)

Zeller, A., Lutkehaus, D.: DDD A free graphical front-end for unix debuggers. *ACM Sigplan Notices* **31**(1), 22--27 (1996)

**Doctorado en Ingeniería
mención mecánica computacional**

Título de la obra:

**An Extended Mixture Model
for the
Simultaneous Treatment of
Short and Long Scale Interfaces**

Autor: Santiago Márquez Damián

Lugar: Santa Fe, Argentina

Palabras Clave:

Finite Volume Method, Mixture Model,
Hyperbolic Systems with Restrictions,
Volume of Fluid Method, Multiphase Flow,
Coupled Models, OpenFOAM