

UNIVERSIDAD NACIONAL DEL LITORAL



**Enlarging time-steps  
for solving one and two phase flows  
using the Particle Finite Element Method**

Juan Marcelo Gimenez

FICH

FACULTAD DE INGENIERÍA

Y CIENCIAS HÍDRICAS

CIMEC

CENTRO DE INVESTIGACIÓN

DE MÉTODOS COMPUTACIONALES

Tesis de Doctorado en Ingeniería  
Mención Mecánica Computacional

2015



UNIVERSIDAD NACIONAL DEL LITORAL

Facultad de Ingeniería y Ciencias Hídricas

Centro de Investigación de Métodos Computacionales

**ENLARGING TIME-STEPS  
FOR SOLVING ONE AND TWO PHASE FLOWS  
USING THE PARTICLE FINITE ELEMENT METHOD**

**Juan Marcelo Gimenez**

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

**2015**

Comisión de Posgrado, Facultad de Ingeniería y Ciencias Hídricas, Ciudad Universitaria, Paraje “El Pozo”,

S3000, Santa Fe, Argentina.

**ENLARGING TIME-STEPS  
FOR SOLVING ONE AND TWO PHASE FLOWS  
USING THE PARTICLE FINITE ELEMENT METHOD**

**Juan Marcelo Gimenez**

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  
Ph.D in ENGINEERING  
mention Computational Mechanics

**2015**

## **Author Legal Declaration**

This dissertation has been submitted to the Postgraduate Department of the Facultad de Ingeniería y Ciencias Hídricas in partial fulfillment of the requirements 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.

Some parts of the work presented in this thesis have been published in the following journals: Computer & Fluids, Computational Particle Mechanics, Computational Physics and Computer Methods in Applied Mechanics and Engineering.

Juan Marcelo Gimenez

©Copyright by  
Juan Marcelo Gimenez  
2015

ENLARGING TIME-STEPS  
FOR SOLVING ONE AND TWO PHASE FLOWS  
USING THE PARTICLE FINITE ELEMENT METHOD

**Thesis advisor:**

- Dr. Norberto M. Nigro  
CIMEC (CONICET-UNL)  
FICH - UNL

**Evaluation committee members:**

- Dr. Rainald Löhner  
George Mason University - USA
- Dr. Gustavo Buscaglia  
Instituto de Ciências Matemáticas e de Computação - USP - Brazil
- Dr. Mario Storti  
CIMEC (CONICET-UNL)  
FICH-UNL
- Dr. Enzo Dari  
CNEA, CAB, CONICET

FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS  
UNIVERSIDAD NACIONAL DEL LITORAL

2015



*A mis viejos*



# Acknowledgments

This thesis has mainly received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Universidad Nacional del Litoral (UNL) and Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT). Others contributors were the program ERASMUS MUNDUS action 2 ARCOIRIS project and the European Research Council under the Advanced Grant: ERC-2009-AdG “Real Time Computational Mechanics Techniques for Multi-Fluid Problems”. The research work was mainly carried out at the Centro de Investigación de Métodos Computacionales (CIMEC) Santa Fe, Argentina, dependent of UNL and CONICET. Also, six months were done at the Escuela Técnica Superior de Ingenieros Navales (ETSIN) of the Universidad Politécnica de Madrid (UPM), Spain.

An special acknowledgement is given to all the Free Software community and particularly to the Ubuntu, PETSc, libMesh, OpenFOAM®, Gnome and LaTEX projects.

I would like to thank my advisor Norberto M. Nigro 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 laboratories.

Finally and the most important, this thesis is dedicated to my parents who have guided me and unconditionally have walked beside me. This is in my mother's memory.



# Abstract

The analysis, comprehension and study of incompressible flows is a field of great interest in industry and academia. Several categorizations can be found in order to organize and simplify the modeling, being the most important the distinction about the number of fluid-phases involved in the scenario. The development and analysis of numerical methods for one-phase flows, which are found in areas as aerodynamics, civil engineering, among many examples, have undergone a tremendous progress and impressive results. Multi-phase flows, despite the fact of being present in hydraulics, petrochemical industry, oceanography, siderurgy, atomic energy and many other human activities, it is a field placed far from being completely understood and the available numerical tools are still in a developing stage. Beyond the number of phases involved, a general limitation of most of the current numerical strategies for incompressible flows is the considerable computational effort required to solve problems of industrial interest where complex geometries are present requiring high level of detail and large simulation times.

In this context, the current thesis presents a highly efficient numerical tool to solve one- and two-phase flows called Particle Finite Element Method Two (PFEM-2). The proposal, which is an evolution from its predecessor PFEM, is a numerical method with a hybrid spatial discretization, which uses Lagrangian particles and Eulerian mesh, for the numerical solution of the transport equations written in a formulation based on the material derivative, or Lagrangian formulation. The improved methodology includes a novel strategy of explicit integration of particles trajectories following the streamlines, which gives it the capability of using large time-steps without significant accuracy loss. In addition, this thesis presents a high performing implementation, able to be executed over scientific clusters, and the extension of PFEM-2 to solve two-phase flows considering the treatment of discontinuities at the interfaces and surface tension without loosing the feature of employing large time-steps.

New strategies of projection from particles to mesh, mathematical formulation of

integration errors, solution of complex academical and industrial problems where the capability of using large time-steps is enforced and compared with classical numerical alternatives, among other novel topics treated, conform the body of this thesis giving a robust frame of analysis for PFEM-2 and positioning it among the fastest methods to solve incompressible flows.

# Resumen

El análisis, compresión y estudio de flujos incompresibles es un campo de gran interés tanto en ambientes académicos como en la industria. Con el fin de organizar y simplificar el modelado, existen diversas maneras de clasificar estos flujos, siendo la más importante la distinción por el número de fases involucradas. El desarrollo y análisis de métodos numéricos para flujos a una fase, el cual encuentra aplicaciones en áreas como aerodinámica, ingeniería civil, entre otros, ha sufrido un tremendo progreso y resultados impactantes. En el caso de los flujos a varias fases, pese a estar presentes en una mayor diversidad de situaciones como en hidráulica, industria petroquímica, oceanografía, siderúrgica, energía atómica entre otras varias actividades humanas, conforman un campo que aún no está totalmente comprendido y en el cual las herramientas numéricas disponibles se encuentran en una etapa de desarrollo. Más allá del número de fases involucradas, un problema general de las estrategias actuales para la solución de problemas de flujo incompresible es el considerable esfuerzo computacional requerido para resolver problemas de interés industrial donde se utilizan geometrías complejas y se requiere un alto nivel de detalle.

En este contexto, ésta tesis presenta una metodología numérica altamente eficiente para la solución de problemas a una y dos fases denominada como Método de Partículas y Elementos Finitos (PFEM-2). Ésta propuesta, la cual es una evolución de su predecesor PFEM, es un método numérico que consiste en una discretización espacial híbrida, que utiliza partículas Lagrangianas y una malla fija Euleriana, para la solución numérica de ecuaciones de transporte escritas en una formulación basada en la derivada material, o formulación Lagrangiana. En su nueva generación se incluye una novedosa estrategia de integración explícita de la trayectoria de las partículas, la cual concede al método la posibilidad de utilizar grandes pasos de tiempo sin sacrificar demasiada precisión. Además, esta tesis presenta la implementación del método para cómputo de alto rendimiento y la extensión del mismo a flujos a dos fases atendiendo al tratamiento de discontinuidades en las interfaces y la tensión

superficial, todo esto sin perder la facultad de emplear grandes pasos de tiempo.

Una novedosa derivación del método, nuevas estrategias de proyección de partículas a mallas, formulación matemática de los errores de integración, resolución de problemas complejos de interés industrial y/o académico, dónde la capacidad de utilizar grandes pasos de tiempos es analizada y comparada con alternativas numéricas clásicas, conforman el cuerpo de esta tesis otorgando un marco robusto de análisis a PFEM-2 y posicionandolo entre los métodos más rápidos para la solución de flujos incompresibles.

# Contents

<b>Acknowledgments</b>	<b>III</b>
Abstract . . . . .	V
Resumen . . . . .	VII
<b>Figure List</b>	<b>XIII</b>
<b>Table List</b>	<b>XV</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background and motivation . . . . .	1
1.2 Objectives . . . . .	4
1.3 Thesis outline . . . . .	5
<b>I One-phase flows</b>	<b>7</b>
<b>2 Fluid motion equations</b>	<b>9</b>
<b>3 Numerical formulation</b>	<b>13</b>
3.1 Segregated methods . . . . .	14
3.1.1 Splitting of viscosity in FSM . . . . .	16
3.1.2 Drawbacks with convective terms . . . . .	16
3.2 Enhanced Particle Finite Element Method (PFEM-2) . . . . .	17
3.2.1 eXplicit Integration following the Velocity Streamlines (X-IVS) . . . . .	19
3.2.2 Projection from particles to mesh nodes . . . . .	20
3.2.3 Variational form of Eulerian stages . . . . .	30
3.2.4 Algorithm synopsis . . . . .	33

<b>4 High performing implementation</b>	<b>35</b>
4.1 Domain decomposition . . . . .	36
4.2 Load balancing . . . . .	39
4.3 Other stages . . . . .	41
4.4 Performance of the implementation . . . . .	41
<b>5 Analysis of Lagrangian and Eulerian errors</b>	<b>45</b>
5.1 Solving a scalar transport equation . . . . .	46
5.1.1 Eulerian errors . . . . .	47
5.1.2 Lagrangian errors . . . . .	48
5.1.3 Validation test . . . . .	51
5.2 An extension to viscous incompressible fluid flow . . . . .	55
5.2.1 General comments . . . . .	55
5.2.2 Validation test . . . . .	56
<b>6 Industrial application: residence time distributions</b>	<b>59</b>
6.1 Background . . . . .	59
6.2 Turbulent dispersion . . . . .	61
6.3 Contact tank . . . . .	63
6.4 Skimmer tank . . . . .	65
<b>II Two-phase flows</b>	<b>71</b>
<b>7 Fluid motion equations</b>	<b>73</b>
<b>8 Extension of PFEM-2 to two-phase flows</b>	<b>77</b>
8.1 Internal interfaces evolution . . . . .	77
8.2 Enriched shape functions . . . . .	81
8.2.1 Pressure calculation . . . . .	83
8.3 Surface tension treatment . . . . .	86
8.4 Algorithm synopsis . . . . .	90
<b>9 Interface tracking problems</b>	<b>93</b>
9.1 Rigid body rotation of Zalesak's disk . . . . .	93
9.2 Single vortex case . . . . .	94

9.3 LeVeque deformation case . . . . .	96
<b>10 Inertial-dominant problems</b>	<b>97</b>
10.1 Rayleigh-Taylor instability . . . . .	98
10.1.1 Extending the time-step . . . . .	100
10.1.2 Three-dimensional simulation . . . . .	102
10.2 Non-linear sloshing in a rectangular container . . . . .	102
10.2.1 Enrichment and density jump issues . . . . .	104
10.3 Viscous standing waves . . . . .	107
10.4 Dam-break problem . . . . .	110
10.4.1 Three-dimensional simulation . . . . .	111
<b>11 Surface tension-dominant problems</b>	<b>117</b>
11.1 Stability Analysis . . . . .	118
11.2 Rising bubble case . . . . .	120
11.2.1 Zero gravity condition . . . . .	122
11.2.2 Ellipsoidal regime test . . . . .	124
11.2.3 Skirted regime test . . . . .	124
11.2.4 Extending time-step . . . . .	126
11.3 Standing capillary wave . . . . .	129
11.4 Towards a simulation of 3D jet atomization . . . . .	132
<b>12 Conclusions</b>	<b>137</b>
12.1 Published works . . . . .	138
12.2 Open lines of research . . . . .	139
<b>Appendices</b>	<b>141</b>
<b>A Resumen extendido de la Tesis</b>	<b>143</b>
A.1 Introducción . . . . .	143
A.2 Flujos a una fase . . . . .	144
A.2.1 Implementación en paralelo . . . . .	147
A.2.2 Análisis de errores de integración y pruebas . . . . .	148
A.3 Flujos a dos fases . . . . .	150
A.3.1 Extensión de PFEM-2 a flujos bifásicos . . . . .	151

A.3.2 Resultados y validación	153
A.4 Trabajos futuros	154
<b>Bibliography</b>	<b>157</b>

# List of Figures

3.1 Trajectory integration approaches . . . . .	20
3.2 Surrounding particles . . . . .	21
3.3 Projections comparison: Log RMS . . . . .	26
3.4 Projections comparison: smooth and sharp functions . . . . .	29
4.1 Four cases tracking a particle . . . . .	39
4.2 Implementation speed-up . . . . .	42
5.1 Order of error in projections . . . . .	50
5.2 Errors in scalar transport case. Convergence . . . . .	53
5.3 Errors in scalar transport case. Evolution . . . . .	54
5.4 Errors in incompressible flow case . . . . .	58
6.3 Skimmer: mesh . . . . .	67
6.5 Skimmer: tracer . . . . .	70
7.1 Two-phase flow configuration . . . . .	74
8.1 Interface issues . . . . .	79
8.2 Enrichment 2d: proposal A . . . . .	82
8.3 Enrichment 2d: proposal B . . . . .	83
9.1 Zalesak Disk results . . . . .	94
9.2 Single Vortex results . . . . .	95
9.3 Leveque's test results . . . . .	96
10.1 Rayleigh Taylor results . . . . .	99
10.2 Rayleigh Taylor evolution . . . . .	99
10.3 Rayleigh Taylor 3d evolution . . . . .	103

10.4 Sloshing configuration	104
10.5 Sloshing heights	105
10.6 Sloshing heights. Enrichment options	107
10.7 Standing wave configuration	108
10.8 Standing wave results	109
10.9 Dam-Break configuration	111
10.10 Dam-break water levels	112
10.11 Dam-break pressures	113
10.12 Dam-break snapshots	114
10.13 Dam-break 3d pressures	115
11.1 Stability analysis: chart	119
11.2 Rising bubble case configuration and boundary conditions.	121
11.3 Rising Bubble case: zero gravity test	124
11.4 Rising bubble case: ellipsoidal test	125
11.5 Rising bubble case: skirted test	127
11.6 Rising bubble case: ellipsoidal test with large time-step	128
11.7 Rising bubble case: skirted test with large time-step	129
11.8 Rising bubble case: stability chart	130
11.9 Configuration setup of standing capillary wave case	131
11.10 Standing capillary wave kinetic energy	132
11.11 Jet 3d atomization: geometry	133
11.12 Jet 3d atomization: results	135
A.1 Implementation speed-up	148

# List of Tables

3.1 Projections comparison: RMS . . . . .	25
4.1 Computing times in an Infiniband cluster . . . . .	43
5.1 Parameters for the manufactured 2d scalar transport case . . . . .	53
5.2 Errors validation cases. Incompressible flow . . . . .	57
6.2 Skimmer tank main features and operative conditions . . . . .	68
10.1 Rayleigh-Taylor snapshots . . . . .	101
10.2 Rayleigh-Taylor computing times . . . . .	101
11.1 Physical parameters for the rising bubble case . . . . .	120
11.2 Rising bubble case: pressure and errors . . . . .	123
11.3 Standing capillary wave errors . . . . .	131
11.4 Configuration Jet . . . . .	134

# Chapter 1

## Introduction

*Life is what happens  
while you are busy making other plans.*

### 1.1 Background and motivation

As Moore predicted in the mids of 1960s, during the last fifty years the number of transistors in an integrated circuit has doubled approximately every two years [Moore, 1965]. This huge advance in the hardware also allowed increasing, by the same rate, the numerical grid sizes used in the simulation of physical phenomena giving the possibility of performing more realistic simulations [Voller and Porté-Agel, 1965]. Over those decades, computer simulation of incompressible fluid flows has been mainly based on the Eulerian formulation of the fluid mechanics equations on fixed domains [Donea and Huerta, 2003]. A trend-line in the Computational Fluid Dynamics (CFD) research of the last years tries to improve the numerical strategies to achieve faster solvers which make use of new technologies such as parallelism, and more recently, the Graphical Processing Units (GPU). On the other hand, while the computing capability increases, the traditional numerical methods lack ability to solve more complex problems. A manifest example in the field of CFD is the case of heterogeneous flow simulation, where the Finite Element Method (FEM) or Finite Volume Method (FVM), originally used to solve homogeneous flows, must be enhanced and adapted to capture and manage the intricate phenomena, such as free-surface behavior, that appear in those type of problems.

Methods based on the Lagrangian formulation were born as an alternative to traditional Eulerian methods. Researchers found that the Lagrangian approach allows to solve naturally

simulations in which deformations are not negligible and, in addition, its explicit formulation is in better position regarding the computational efficiency when is employed in parallel computing. Lagrangian formulations aim to represent the behavior of a physical problem by a collection of particles. Each particle moves according to its own mass, and the internal and external forces applied on it. External forces are evaluated by the interaction with the neighbor particles. All physical and mathematical properties are attached to the particle itself. For instance, physical properties like viscosity or density, physical variables like velocity, temperature or pressure and also mathematical variables like gradients or volumetric deformations are assigned to each particle and they represent an average of the property around the particle position. In order to evaluate the forces among particles any classical approximation method may be used, including FEM, finite differences, meshless methods, etc. This means that a particle method may be used with or without a mesh, depending on the method chosen to evaluate the interaction forces.

Although the capabilities regarding to manage free-surfaces, fluid-solid interaction and explicit computing, pure Lagrangian formulation also presents its drawbacks and requires expensive processing unneeded in Eulerian frameworks. In the case of meshless methods, constant tracking of all the moving points must be done, a task that is non-trivial when there are several million particles in a large domain. For meshed methods in fluid problems, the extra task is the construction of a new non-distorted mesh at each time-step.

Nowadays, hybrid methods gain traction in CFD, where the Lagrangian framework given by a particle method, is combined with an Eulerian methodology. In these hybrid methods, a fixed or reconstructed grid supports part of the pressure and velocity calculation. The original idea, proposed by Monaghan [Gingold and Monaghan, 1977] and later works applied to fluid mechanics [Monaghan, 1988], where a pure Lagrangian perspective was used during the whole meshless computation, has been in some cases completed using other well known discretization methods, such as FVM [Nestor et al., 2009] or FEM [Idelsohn et al., 2003a]. The first combination of Lagrangian and FEM methods can be found in [Idelsohn et al., 2003b], where an extended Delaunay Tessellation is used to reconstruct the mesh while the fluid evolves. In this method, known as MFEM, the construction of the shape functions inside each polyhedron is based on a non-Sibsonian interpolation.

The next step in this evolution was the first version of the Particle Finite Element Method (PFEM) [Idelsohn et al., 2004] which is a robust method designed to solve fluid-structure interaction problems including free-surface, breaking waves, flow separations, etc... where

---

Lagrangian particles and meshing processes are alternated with the advantage of having a FEM structure that supports the differential equation solvers. An interesting difference between the PFEM and other hybrid methods as PIC (Particle-In-Cell) [Harlow, 1955], MAC (Marker-And-Cell) [Harlow and Welch, 1965] and MPM (Material Point Method) [Wieckowsky, 2004] is that while in latter methods the particles transport mass and consequently have a volume, PFEM particles are non-material points that transport the fixed intensive properties of the fluid.

Other methods, that also combine both Eulerian and Lagrangian perspectives, are the arbitrary Eulerian-Lagrangian (ALE) [Donea, 1983] or semi-Lagrangian methods [Allievi and Bermejo, 2000]. The Lagrangian perspective makes it possible to use a material derivative formulation where the absence of the non-linear convective terms transform the Navier-Stokes system into a linear coupled problem. Methodologies, such as the backward Characteristics method [Allievi and Bermejo, 2000], also give this possibility but if the process is done in a fixed mesh without any distortion, and unless high order polynomials are used, a dissipative process appears due to the interpolation of the feet of characteristics.

In contrast to the backward characteristic method where the feet of the characteristic line was searched and located in a mesh element based on the known velocity fields at past time-steps, a new strategy known as X-IVS (eXplicit Integration following the Velocity Streamlines) was developed by Idelsohn et al. [Idelsohn et al., 2012]. This methodology of integrating the convection of fluid particles is based on following the streamlines of the flow in the current time-step instead of the particle trajectories, which represents an alternative way to solve the non-linearities of the flow equations. Adding this strategy to the original PFEM method a new methodology appears which is called Particle Finite Element Method Second Generation (PFEM-2) [Idelsohn et al., 2013b]. The X-IVS strategy gives the possibility of solving complex flows with large time-steps, as well as the presence of the mesh allows accurate solutions of the fractional step method. Author's master thesis [Gimenez, 2014] presents a deep review of PFEM-2 where implementation issues regarding to parallelism are analyzed and several cases of homogeneous incompressible flows are solved with impressive results.

Therefore, the current thesis presents a review of the novelty, features of the implementation and analysis of main characteristics of the method, including a mathematical analysis of stability and errors. Several benchmarks are solved so as to validate the theoretical advantages of PFEM-2 in both, one- and two-phase systems. Finally complex problems with

application in oil and process industries are presented confirming the capabilities of the method of being a more efficient alternative to the classical methodologies for a wide range of problems.

## 1.2 Objectives

The main objective of this thesis is the development and implementation of a fast and accurate method to simulate both one-phase and two-phase incompressible flows. The strategy developed is expected to be more efficient solving inertia dominant problems than classical numerical methodologies.

In order to achieve the main objective, several original contributions and novelties in this thesis must be cited:

- Development of a hybrid methodology based on the material derivative formulation of the equations which combines Lagrangian particles and a fixed Eulerian mesh, taking as starting point the work of Idelsohn et. al. [Idelsohn et al., 2013b].
- Strong mathematical development of fundamentals of the method such as the projection particle-nodes algorithms and the integration errors, where is analytically determined and numerically validated the range of problems where this methodology is preferable to be employed.
- Implementation employing high performance computing (HPC) strategies which obtains good parallel efficiency when large problems are solved.
- Extension of the method to the case of two-phase flows, allowing to keep the use of large time-steps, through:
  - Modification of particle convection: adding of a marker function and behavior modeling based of density differences.
  - Interface evolution and determination of its position.
  - Treatment of jumps and kinks of the unknowns.
  - Modeling of surface tension. Smoothing of the marker function to reduce parasitic velocities in FEM environment.
  - Large set of validation tests comparing accuracy against reference data and efficiency against other numerical tools.

- 
- Simulation of complex cases: Residence Time Distribution in a skimmer tank for water-oil separation and the atomization of a liquid jet with droplet spray formation.

### 1.3 Thesis outline

The thesis is split into two main parts, dedicated to the analysis of one-phase and two-phase flows respectively.

The Part I, which deals with one-phase flows, is organized as follows: Chapter 2 introduces the theory of one-phase flows focusing on the differences between the Eulerian and Lagrangian continuum formulations. In Chapter 3 a novel derivation of PFEM-2 is presented where the method is branched from the classical Eulerian fractional-step strategy but the convection is solved using a Lagrangian framework. Special emphasis is done in the strategies of projection of states from particles to nodes. Chapter 4 shows aspects about the computational implementation of the method employing high performance strategies in order to be able to simulate with PFEM-2 large problems over computing clusters. Chapter 5 is devoted to demonstrate why PFEM-2 is more accurate than a classical Eulerian FEM in certain cases when large time-step and/or coarse meshes are used. An error formula is obtained and validated through numerical tests, both in scalar transport and incompressible flow cases. Finally, Chapter 6 presents a case of oil industry where a skimmer tank devoted to oil-water separation is analyzed. The target is to find the residence time distribution to determinate the efficiency of the equipment, and PFEM-2 solutions are evaluated solving the same problem with large time-steps.

The Part II, which is concerned with two-phase flows, presents the following structure: Chapter 7 shows the theory of two-phase flows and the main physical phenomena that may occur at the interface, focusing specially on the surface tension. The chapter also presents the governing equations, together with the boundary and interface conditions, and describes the possible discontinuities of the flow variables. In Chapter 8 the extension of PFEM-2 to two-phase flows is shown, where topics as the treatment of interface evolution, capturing of variables discontinuities and modeling of the surface tension are deeply analyzed, deriving in a general algorithm. Chapter 9 deals with an exhaustive validation of the proposed PFEM-2 method to transport arbitrary shapes with neither interface disturbances nor mass loss. In Chapter 10 a wide range of two-phase problems, in which the inertial term is dominant, with different ratio between densities and viscosities of the involved fluids is

solved using the PFEM-2 method and the results compared with experimental and numerical reference. Chapter 11 presents the stability and accuracy of problems where the surface tension force dictates the interface behavior, always forcing the time-step and comparing with other numerical alternatives.

Last chapter summarizes the main conclusions and achievements of the thesis, and describes open lines of research.

# **Part I**

## **One-phase flows**



# Chapter 2

## Fluid motion equations

The governing equations in fluids have been commonly formulated in a spatial or Eulerian description, where the frame of reference is fixed. Lagrangian description makes use of the material formulation, in which the frame of reference coincides with the particles of the medium and thus, it is convected with the fluid. In the continuum, any point is the average behavior of the molecules in a suitable neighborhood of the point, then the macroscopic properties of the bulk fluid can be associated to any fluid particle.

Let  $\Omega \subset \mathbf{R}^d$ ,  $d \in \{2, 3\}$ , be a bounded domain. We denote time by  $t$ , the Cartesian spatial coordinates by  $\mathbf{x} = \{x_i\}_{i=1}^d$ , and  $\nabla = \{\partial_{x_i}\}_{i=1}^d$ . We consider an homogeneous viscous flow with constant density  $\rho$  and dynamic viscosity  $\mu$ . The evolution of the velocity  $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$  and the pressure  $p = p(\mathbf{x}, t)$  is governed by the Navier-Stokes equations:

$$\nabla \cdot \mathbf{u} = 0 \quad (2.1)$$

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}. \quad (2.2)$$

Equation (2.2) is the differential momentum balance equation. Its expression corresponds to the Eulerian framework, where the two terms on the left hand side describe the movement of the fluid: the first one is the acceleration and the second one is the convective term. On the right side the terms defining the forces action on the fluid: the first one is the surface force where the stresses tensor  $\boldsymbol{\sigma}$  for an isotropic and incompressible fluid is defined as  $\boldsymbol{\sigma} = -p\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ , with  $\mathbf{I}$  the identity matrix. The equations are restricted to the case where the only volume force acting on  $\Omega$  is gravity  $\mathbf{g} \in \mathbf{R}^d$ . On the other hand, Equation (2.1) is the differential expression of the mass balance, conforming the so-called continuity equation or incompressibility condition.

The boundary  $\Gamma$  of  $\Omega$  can be considered as the union of two boundary types  $\Gamma = \Gamma_D \cup \Gamma_N$ :

$\Gamma_D$ , where Dirichlet boundary conditions are imposed for the velocity and homogeneous Neumann boundary conditions for pressure and  $\Gamma_N$  where homogeneous Dirichlet boundary conditions are imposed for the pressure and homogeneous Neumann boundary conditions are used for the velocity. Typically, the condition over  $\Gamma_D$  corresponds to the presence of a no-slip wall (imposing  $\mathbf{u} = 0$ ) or an inlet (imposing  $\mathbf{u} \cdot \mathbf{n}_{\Gamma_D} \neq 0$ ). On the other hand, the condition over  $\Gamma_N$  represents an outlet condition where the pressure is set to a reference value. Other boundary conditions could be considered, as slip wall ( $\mathbf{u} \cdot \mathbf{n}_{\Gamma} = 0$ ), symmetry plane, periodic, etc, but its employment does not modify substantially the conclusions obtained in this Chapter.

The relationship between the time derivatives expressed in the material (Lagrangian) and the spatial (Eulerian) domains is written as:

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f \quad (2.3)$$

being the left-hand side the so-called material derivative, and  $f = f(\mathbf{x}, t)$  any scalar or vectorial physical quantity. In the case of  $f = \mathbf{u}$ , the Equation (2.3) employed in (2.2) gives the Navier-Stokes equations expressed in the Lagrangian framework:

$$\nabla \cdot \mathbf{u} = 0 \quad (2.4)$$

$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} \quad (2.5)$$

Due to the multiplication of the velocity and its gradient, the convective term in Equation (2.2) is non-linear. The presence of this non-linear term demands iterative algorithms to converge to the solution including linearization techniques to solve the system. On the other hand, the Navier Stokes equations written in a Lagrangian framework lead to a system of linear equations due to the absence of the convective term. Moreover, the resulting system changes the non-symmetric equations in the Eulerian frame into a symmetric and positive definite one. The drawback in the Lagrangian framework is that a kinematic problem has to be solved in order to follow the particle trajectories:

$$\frac{D\mathbf{x}}{Dt} = \mathbf{u}(\mathbf{x}, t) \quad (2.6)$$

then, the convective non-linearity of Equation (2.2) is transformed into a geometrical non-linearity in Equation (2.6). In this way, the governing equations in Lagrangian description present an initial value problem, and one is forced to include time as an independent variable even for steady flows.

It must be noted that in the case of constant viscosity in the entire domain, the viscous term expressed in *divergence form*  $\nabla \cdot \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)$  can be simplified to  $\mu \nabla^2 \mathbf{u}$ , achieving the called *Laplace form*.

An adimensional number, which is used to determine dynamic similitude between two different cases of fluid flow, is the Reynolds number  $Re$ . This quantity is defined as the ratio between the momentum forces and the viscous forces, and consequently quantifies the relative importance of these two types of forces for given flow conditions. The Reynolds number is mathematically defined as

$$Re = \frac{\text{inertial forces}}{\text{viscous forces}} = \frac{\rho U D}{\mu} \quad (2.7)$$

being  $D$  a characteristic linear dimension and  $U$  a characteristic velocity. Therefore, according to the value of  $Re$  a widely used categorization divide the flows into two main types:

- Laminar flow, which occurs at low Reynolds numbers, in which viscous forces are dominant, characterized by smooth fluid motion.
- Turbulent flow, which occurs at high Reynolds numbers and is dominated by inertial forces, which tend to produce chaotic eddies and other flow instabilities.

The non-linearity of the convective term is responsible for the instabilities that could appear in the flow. For a certain value of  $Re$ , which depends on the configuration of the problem, the stability of the system is lost. Small perturbations are amplified leading to an irregular, chaotic and unpredictable behavior of the flow. The nature of turbulent flow leads to the idea of randomness, then statistical and/or modeling techniques can be employed in its study.

Among the numerical techniques available to simulate turbulent flows, can be listed:

- **DNS** (Direct Numerical Simulation): The most accurate and conceptually simplest approach. It solves the Navier-Stokes equations with neither averages nor approximations more than the discretization, whose error can be estimated and controlled. In order to ensure capturing all the turbulence scales, the grid size must be in the order of the Kolmogorov scale  $D/\eta \sim Re^{3/4}$  and the temporal step must respect the time scale  $\Delta t_D / \Delta t_\eta \sim Re^{1/4}$ . These characteristics lead DNS to be affordable only with simple geometries with low Reynolds numbers.
- **LES** (Large Eddy Simulation): Although the turbulent flows have a wide range of temporal and spatial scales, the biggest ones have more energy for transporting conservative properties. Therefore, in the LES approach the large scales are solved and

---

the small scales are modeled. The variables which are solved are filtered versions of the real ones, i.e.  $\bar{\mathbf{u}} = \int G(x, x') \mathbf{u}(x) dx'$  where  $G(x, x')$  is a filter with length scale  $\Delta$ . After filtering the Navier-Stokes equations, a tensor  $\tau_{ij}^s = -\rho(\bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j)$  appears, which is known as Sub-Grid Scale Reynolds stress (SGS). The SGS tensor must be modeled, leading to several variants of the LES approach.

- **RANS** (Reynolds Averaged Navier-Stokes): This approach solves the time-averaged equations of motion for fluid flow. The idea behind the equations is Reynolds decomposition, whereby an instantaneous quantity is decomposed into its time-averaged and fluctuating parts. In order to modelate the fluctuations tensor that appears after averaging, the Boussinesq hypothesis is employed leading to the called *closure models*. These models estimate the eddy viscosity solving transport equations of turbulent variables as kinetic turbulent energy, dissipation rate, among others.

In addition, owing to the presence of a no-slip wall the flow behavior and turbulence structure are considerably different from free turbulent flows where the inertial forces dominate. Taking into account the Reynolds number based on a distance  $y$  away from the wall ( $Re_y = \frac{\rho |\mathbf{u}| y}{\mu}$ ), just before  $y$  reaches zero in the wall, there will be a range of values of  $y$  for which  $Re_y$  is on the order of 1. At this distance from the wall and closer, the viscous forces will be equal in order of magnitude to the inertia forces or larger. Therefore, close to the wall the flow is influenced by viscous effects leading to a laminar behavior which conforms the so-called *linear sub-layer*. After that, the transition from laminar to turbulent follows a logarithmic law up to the zone far from the wall which is free from direct viscous effects. This transitional sub-layer is named *log-law sub-layer*. If the grid-size is not able to capture the mentioned transition, modeling should be done employing the so-called *wall functions*. Readers interested in a deep explanation oriented to developing should read the book of Ferziger and Peric [Ferziger and Peric, 2002].

# Chapter 3

## Numerical formulation

There is a large number of strategies to solve numerically the incompressible flow equations. A first classification can be done attending to the reference system employed, i.e. Eulerian or Lagrangian. In the former group can be found two subgroups of methods: those which solve velocity and pressure in a coupled way, named *monolithic* and those which decouple the unknowns in order to solve them in a segregated way, named *projection methods*. The second subgroup has won popularity due to the resulting algebraic equation system being easier, and faster, to solve than that obtained with the monolithic strategy, in which usually ill-conditioned matrices with a large bandwidth appear.

In the case of methods which employ the Lagrangian framework, a sub-grouping is done according to the way to afford the severe distortion of the original domain, i.e. using remeshing algorithms and meshless methods. While the former approach introduces a new mesh and transfers the information from the old distorted mesh to the new one, being this process source of numerical diffusion, the latter abandons the mesh and represents the problem domain by a finite number of interacting particles.

In the current chapter, the segregated methods will be presented and the numerical method PFEM-2 will be derived from it. A comparison between the employment of segregated or monolithic strategies in the context of PFEM-2 is shown in the thesis of Becker [Becker, 2015]. Regarding Lagrangian alternatives, the introduction of the current thesis granted a wide review of the current strategies, but along the work several references will appear.

### 3.1 Segregated methods

A major difficulty for the numerical simulation of incompressible flows is that the velocity and the pressure are coupled by the incompressibility constraint. The groundbreaking works of Chorin [Chorin, 1968] and Temam [Témam, 1969] started the development of segregated or Fractional-Step Methods (FSM) to overcome this difficulty in time-dependent viscous incompressible flows. The most attractive feature of FSM is that, at each time-step, one only needs to solve a sequence of decoupled parabolic and elliptic equations for the velocity and the pressure respectively, making it very efficient for large scale numerical simulations. There are several ways to perform such splitting and therefore a variety of fractional-step methods for the unsteady Navier-Stokes equations do exist. The books by Quartapelle [Quartapelle, 1993], Quarteroni and Valli [Quarteroni and Valli, 1994], and references therein, should be consulted for a detailed exposition of fractional-step methods.

Fractional-step methods perform time discretization before the spatial discretization. When this approach is adopted, a controversy arises on which boundary conditions must be imposed at each step, because the intermediate semi-discrete problems must be well-posed. Another important feature is the overall order of accuracy with respect to time discretization. Most fractional-step methods are first-order accurate, but some second-order accurate methods have also been developed [Donea and Huerta, 2003].

The principle of the projection method is to compute the velocity and pressure fields separately through the computation of an intermediate velocity, which is then projected onto the subspace of solenoidal vector functions. The Helmholtz decomposition states that any vector field  $\mathbf{w}$  in  $\Omega$  admits the unique orthogonal decomposition

$$\mathbf{w} = \mathbf{v} + \nabla\phi \quad (3.1)$$

into the sum of  $\mathbf{v}$  a solenoidal field with zero normal component on the domain boundary (i.e.,  $\nabla \cdot \mathbf{v} = 0$  and  $\mathbf{v} \cdot \mathbf{n} = 0$  on  $\Gamma$ ), and the gradient of a scalar function  $\phi$ . In the incompressible flow context, an intermediate velocity field,  $\hat{\mathbf{u}}^{n+1}$ , is decomposed into the sum of a solenoidal velocity field,  $\mathbf{u}^{n+1}$ , and the gradient of a scalar function proportional to the unknown pressure, namely,  $\nabla p^{n+1}$ .

In the Eulerian framework, the so-called *momentum predictor* is the first step of FSM, which includes the viscous and convective terms in the Laplace form of the Navier-Stokes equations (2.2) and, given the previous time-step velocity field  $\mathbf{u}^n$ , consists in finding an intermediate

velocity field,  $\hat{\mathbf{u}}^{n+1}$ , such that

$$\begin{cases} \rho \frac{\hat{\mathbf{u}}^{n+1} - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^n \cdot \nabla) \hat{\mathbf{u}}^{n+1} - \mu \nabla^2 \hat{\mathbf{u}}^{n+1} = \rho \mathbf{g}^{n+1} - \nabla p^n & \text{in } \Omega \\ \hat{\mathbf{u}}^{n+1} = \mathbf{u}_D^{n+1} & \text{on } \Gamma \end{cases} \quad (3.2)$$

conforming a semi-implicit treatment which linearizes the convective term. Explicit and fully implicit treatment are also other possibilities, but while the former has conditional stability enforcing to reduce the size of  $\Delta t$ , the latter needs to solve nonlinear and non-symmetric systems at each time-step. On the other hand, due to the fact that this step includes the viscous term, the Dirichlet boundary conditions are imposed here.

The second step, also named as *projection step* consists in determining the end-of-step velocity  $\mathbf{u}^{n+1}$  and the pressure  $p^{n+1}$  solving

$$\begin{cases} \rho \frac{\mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1}}{\Delta t} + \nabla (p^{n+1} - p^n) = 0 & \text{in } \Omega \\ \nabla \cdot \mathbf{u}^{n+1} = 0 & \text{in } \Omega \\ \mathbf{u}^{n+1} \cdot \mathbf{n} = \mathbf{u}_D^{n+1} \cdot \mathbf{n} & \text{on } \Gamma \end{cases} \quad (3.3)$$

Applying the divergence operator to the first equation in (3.3), since  $\nabla \cdot \mathbf{u}^{n+1} = 0$  (second equation), the projection step is reformulated in terms of a Poisson equation for the pressure:

$$\begin{cases} \nabla^2 (p^{n+1} - p^n) = \frac{\rho}{\Delta t} \nabla \cdot \hat{\mathbf{u}}^{n+1} & \text{in } \Omega \\ \mathbf{n} \cdot \nabla (p^{n+1} - p^n) = 0 & \text{on } \Gamma \end{cases} \quad (3.4)$$

Once the end-of-step pressure is obtained thought the *Poisson step*, it can be used to determinate the end-of-step velocity  $\mathbf{u}^{n+1}$  using the explicit relation:

$$\mathbf{u}^{n+1} = \hat{\mathbf{u}}^{n+1} - \frac{\Delta t}{\rho} \nabla (p^{n+1} - p^n) \quad (3.5)$$

which is also known as *correction step*. Note that the boundary condition on pressure implies that  $\nabla p^{n+1} = \nabla p^n = \dots = \nabla p^0 = 0$ . This non-physical Neumann boundary condition enforced on the pressure that introduces a numerical boundary layer and consequently limits the accuracy of the scheme [Guermond et al., 2006]. Together with the weak representation of the tangential velocity on the boundary, the non-physical boundary conditions for the pressure represents a potential drawback of FSM.

Note that the projection scheme presented in this section is called *incremental* due to the use of the pressure increment ( $p^{n+1} - p^n$ ), instead of the total pressure, in the incompressibility phase, which leads to a semi-implicit method. However, the drawback of this method is the necessity of using finite elements passing the Ladyzhenskaya-Babuška-Brezzi

(LBB) compatibility condition [Donea and Huerta, 2003]. An alternative solution is the employment of pressure stabilizing which will be analyzed in SubSection 3.2.3. A deep review about projection methods and high-order alternatives was done by Guermond [Guermond et al., 2006].

Finally, a time-step in the classical FSM method (both incremental and non-incremental versions) consists of three main stages:

1. Calculating Momentum predictor.
2. Calculating Poisson equation.
3. Calculating Velocity correction.

### 3.1.1 Splitting of viscosity in FSM

In order to be able to impose the Dirichlet boundary conditions in the second step of FSM, Blasco, Codina and Huerta [Blasco et al., 1998] proposed a method which introduces a diffusion term in the equation of the second step, which consequently loses its inviscid character responsible for preventing control of the prescribed tangential component of the velocity at the boundary. The second step then consists in determining the end-of-step velocity  $\mathbf{u}^{n+1}$  and pressure  $p^{n+1}$  as solution of

$$\left\{ \begin{array}{ll} \rho \frac{\mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1}}{\Delta t} - \mu \nabla^2 (\mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1}) + \nabla (p^{n+1} - p^n) = 0 & \text{in } \Omega \\ \nabla \cdot \mathbf{u}^{n+1} = 0 & \text{in } \Omega \\ \mathbf{u}^{n+1} \cdot \mathbf{n} = \mathbf{u}_D^{n+1} \cdot \mathbf{n} & \text{on } \Gamma \end{array} \right. \quad (3.6)$$

allowing to impose the Dirichlet boundary conditions in both phases of FSM. The drawback of this method is the necessity of using finite elements passing the LBB condition with both *incremental* or *non-incremental* approaches.

### 3.1.2 Drawbacks with convective terms

Although the Eulerian formulation for homogeneous incompressible flows has been used for the last forty years as a standard strategy, both segregated and monolithic approaches have several problems particularly when solve convective dominant problems.

When using an Eulerian framework, the convective term has four main drawbacks:

- *Linearization:* In order to solve the non-linear problem, a linearization of the convective term must be proposed. The strategy presented above decouples the carrier and

transported velocities assuming that the former is fixed at time  $n$ . Although this semi-implicit strategy is not as restrictive as the explicit strategy regarding to the time-step size, the mentioned assumption could also generate large errors.

- *Stabilization:* Using second order approximations for the convective term (i.e. Galerkin FEM, central differences in FDM, linear interpolation in FVM) is not optimal when convection dominates diffusion (the viscosity effects), that is for high Reynolds number flows. In such cases, one must reduce the order of approximation or use stabilization techniques. Particularly, in FEM environment, the stabilization needed to circumvent the LBB condition, named Pressure-Stabilizing/Petrov-Galerkin (PSPG), must be coupled with other terms to stabilize the Galerkin formulation in presence of a dominating convective term in the momentum equation such as Streamline-upwind Petrov-Galerkin (SUPG) , Galerkin Least-Squares (GLS), Sub-Grid Scale (SGS) or Least-Squares (LS). Reader interested in a deeper analysis should read the book of Donea and Huerta [Donea and Huerta, 2003] and the review paper by Tezduyar and Osawa [Tezduyar, 1992]. The main drawback of stabilization procedures is the introduction of artificial diffusion which reduces the accuracy of the method.
- *Non-symmetric system:* The discretization of Eulerian formulation with convective term leads to a non-symmetric algebraic equation system. This type of matrices has lower convergence rate than symmetrical ones due to its inherent bad-conditioning. Therefore, the computational effort to solve these systems is relevant.
- *Hyperbolic problem:* Equation systems from the discretization of elliptic problems with constant physical properties can be pre-factorized in order to accelerate the solution at each time-step [Gimenez et al., 2014]. However when the convective term appears, the system turns hyperbolic which impossibilities the pre-factorization despite using constant physical properties.

### 3.2 Enhanced Particle Finite Element Method (PFEM-2)

The simple fact of using the Lagrangian formulation of the incompressible flow equations (2.5) instead of the Eulerian ones (2.2) transforms the equations system from non-linear to linear because the convective term is not present. Therefore, the Lagrangian Framework naturally avoids the Eulerian drawbacks explained in 3.1.2

The new generation of the Particle Finite Element Method (PFEM-2) was born as an evolution of the original PFEM with the inclusion of a novel explicit integration strategy for the particles called eXplicit Integration Following the Velocity and Acceleration Streamlines (X-IVS or X-IVAS depending on the use or not of the acceleration streamlines) [Idelsohn et al., 2012] and the use of a fixed background mesh allowing to employ large time-steps without compromise the accuracy and avoiding the hard task of remeshing each time-step [Idelsohn et al., 2013a]. In this section, instead of following the historical-driven presentation of PFEM-2, an alternative description of the method is shown, which is focused on alleviate the drawbacks of Eulerian framework through the inclusion of particles in order to solve convective terms.

In the Lagrangian framework, the non-linearity associated with the convective term is replaced by the solution of the non-linear trajectory equation (2.6). A collection of discrete points is employed to solve that equation. A points  $p$ , known as particle, could have associated physical properties like viscosity  $\mu_p$  or density  $\rho_p$ , physical variables like velocity  $\mathbf{u}_p$ , temperature  $T_p$  or pressure  $p_p$  and also mathematical variables like gradients or volumetric deformations and they represent an average of the property around the particle position. In the case of PFEM-2, massless particles are employed (they do not have diameter associated), allowing to use variable density of particles though the domain for example, in order to refine the approximation in some interest regions. This last feature gives to the method proposed in this thesis a relevant advantage over other particle-based methods as SPH or MPM. Also, in PFEM-2 the particles have a protagonist role because they carry the original data which is never destroyed but only updated according to the computing obtained by the mesh, which plays a secondary role.

Finally, in contrast to the sequence of steps presented for FSM, a time-step in the PFEM-2 algorithm consists of five main stages:

1. Calculating convection over particles.
2. Projecting states from particles to mesh.
3. Calculating Momentum predictor over mesh.
4. Calculating Poisson equation over mesh.
5. Calculating Velocity correction over mesh and interpolate it to particles.

It must be noted that two initial steps are included which calculates the convection over particles and project it over the mesh, respectively. Moreover, the last step also includes a correction of the velocity over the particles. Next subsection will deal with each one of these stages in order to present a detailed description of the numerical method.

### 3.2.1 eXplicit Integration following the Velocity Streamlines (X-IVS)

As it was mentioned, the trajectory integration strategy used by PFEM-2 is named X-IVS. The idea of the method is to use the velocity streamlines obtained at time-step  $t^n$  to approximate the final position of the particle  $p$ , i.e.  $\mathbf{x}_p^{n+1}$ , and its approximation to the velocity  $\hat{\mathbf{u}}_p^{n+1} \approx \mathbf{u}_p^n$ :

$$\begin{cases} \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{u}^n(\mathbf{x}_p^\tau) d\tau \\ \hat{\mathbf{u}}_p^{n+1} = \mathbf{u}_p^n \end{cases} \quad (3.7)$$

Equation (3.7) is explicit because is only using information at time-step  $t^n$ . Note that it is not using neither a constant nor a linear approximation of the velocity field. Instead, it is employing the same high order approximation the velocity field has at time  $t^n$ . The only difference with the exact integration is that here we are performing the integral (within each time-step) following a pseudo trajectory of the particles calculated with the velocity streamline, instead of following the true trajectory. Once the field  $\mathbf{u}^n$  is discretized using linear piecewise functions supported by a background mesh, Equation (3.7) may be integrated analytically or using any standard time integration scheme like explicit Runge-Kutta or alternatively by any sub-stepping technique. The way to integrate analytically Equation (3.7) inside each triangular element is presented in [Idelsohn et al., 2012]. Later in [Nadukandi, 2015] the extension to tetrahedra, based on stable formulas, was shown. However, in spite of being the most accurate strategy, the analytical integration requires large computing times when the particles cross several elements mesh. Figure 3.1 presents a graphical representation of the trajectory integration, comparing the exact trajectory, the approximated by X-IVS and a first-order approximation using forward Euler. The more transient is the velocity the more differences appear between exact and X-IVS. Such differences are null when the velocity is steady. First-order approximation can not solve without error even in the steady case.

To sum up, this new integration proposal provides an efficient strategy to employ

time-steps which allow a Courant-Friedrich-Levy (CFL) number larger than one, being

$$\text{CFL} = \frac{|\mathbf{u}| \Delta t}{\Delta x}$$

In practical terms, this means that any particle will be able to travel more than one element by time-step without compromising the stability of the method. However, as it will presented in Chapter 5, the integration error of this strategy will be smaller as the velocity is more steady.

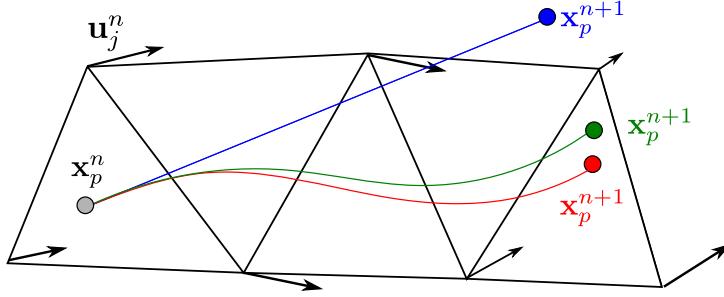


Figure 3.1: Comparison of trajectory integration approaches. Green line represents the exact trajectory, the red line represents the X-IVS approximation and blue line shows the path calculated with a first-order integration.

Originally, this integration strategy also included the use of acceleration streamlines in order to update the particle velocity (X-IVAS) [Idelsohn et al., 2013a]. However, the presence of the viscous term implies a practical limitation to the time-step size, namely that the Fourier number, defined as

$$\text{Fo} = \frac{2\mu \Delta t}{\rho \Delta x^2}$$

must remain smaller than one. An implicit correction was proposed in order to guarantee stability beyond that limit [Gimenez et al., 2012], but that approach has shown several drawbacks in certain cases and moreover its extension to two-phase flows is not straightforward, then X-IVS is the approach chosen for explicit integration leaving the diffusion calculation as a full-implicit step calculated over the mesh.

### 3.2.2 Projection from particles to mesh nodes

As aforementioned, the PFEM-2 spatial discretization consists of a collection of particles and a background fixed mesh. The latter is based on the Finite Element Method [Zienkiewicz and Morgan, 1983], which consists in dividing the domain into elements, whose geometry is defined by nodes. The unknown variables are the values at the nodes and

the solution is interpolated from the nodal values inside each element. Particularly, in PFEM-2 linear piecewise element functions are employed to approximate the unknowns. After convection step, calculated through the movement of the collection of particles, the nodal data of the velocity field must be updated with the value of the convected particles. This section presents several approaches to project particle data to nodal data, analyzing accuracy and convergence rate.

Let  $\mathbf{x}_p^{n+1}$  be the position of a particle  $p$  after convection with  $\phi_p$  the value of the transported variable on the particle,  $\mathbf{x}_j$  the fixed position of the node  $j$  and  $\phi_j$  the nodal value of the variable, and  $J$  the number of nodes of the mesh. The problem consists in determining the value of  $\phi_j$  projecting the values of  $\phi_p$  of the surrounding  $P$  particles employing a projection function  $\Pi_{p \rightarrow j}$ . The *surrounding particles* of the node  $j$  are defined as those which belong to an element of which the node  $j$  is vertex. Figure 3.2 presents a graphical interpretation of this definition.

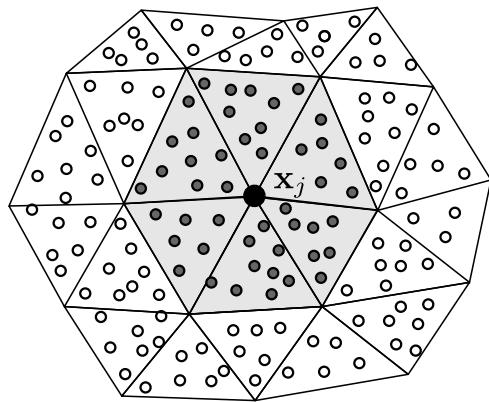


Figure 3.2: Set of  $P$  surrounding particles, dark ones, to a node  $j$  at position  $\mathbf{x}_j$ . Particles not used by node  $j$  are not filled. Grey zone identifies the elements where the node  $j$  is vertex.

In one-phase flow simulations, the particles transport and must project the values of the approximate velocity  $\hat{\mathbf{u}}_p^{n+1}$ . In the case of transporting temperatures, phase-fraction, etc, these states are also projected. Since the beginnings of the method, several projection strategies have been employed and this subject is still a challenge.

There are several drawbacks regarding the particles likely to appear during a simulation. Among the most important can be cited: lacking of particles surrounding a node after the convection step, particles which leave (or enter) the domain, and elevated number of total particles in the simulation. Several tricks are employed to overcome these problems. Those actions can be categorized into preventive and palliative.

- Preventive: After every time-step, a recounting over each mesh-element is made. If the number of particles contained by the element is lower than a predetermined number, some particles are seeded to reach that minimum. The state of the new particle results from the linear interpolation from the nodes to the position of the new particle. On the other hand, if the number of particles exceeds the maximum limit, the last particles which arrived to the element are removed from the simulation, currently losing its information. The decision of deleting the last ones is only for efficiency purposes due to the list of particles of the element is easily cut.
- Palliative: During the convection stage, a particle can leave the domain (as when it goes through an outlet boundary), therefore these particles must be deleted from the inventory. In addition, at the moment of the projection stage, if a node does not have particles surrounding it, which typically happens at elements near the inlet boundary when using  $CFL > 1$ , some state must be imposed to that node. The strategy employed is to seed a new particle at node position and search its state following the streamlines in backward direction, similar as the Characteristics method [Allievi and Bermejo, 2000]. This strategy is highly diffusive because linear elements are employed to interpolate the state, however in most of cases when the particle goes backward it reaches the inlet where the boundary condition can be used avoiding that diffusion.

Regarding the projection strategy itself, a desired property is a global second order of convergence of the error. In addition, projection introduces a degradation of the solution accuracy proportional to the number of time-steps employed. There is a smaller impact when the time-step is large and a bigger one when the time-step tends to reduce. This fact will be observed experimentally in this Section and mathematically described in Chapter 5. On the other hand, a desired property of a projection algorithm is the commutativity, i.e. the projection satisfying the inverse property of operators:

$$\Pi_{p \rightarrow j}^{-1} \left( \Pi_{p \rightarrow j} (\phi) \right) = \mathbf{I}(\phi) \quad (3.8)$$

If Equation (3.8) is not accomplished it strongly endangers the solution accuracy and may also be responsible for excessive diffusion of the primal variables or for noisy secondary (dual) fields computed on the mesh, like forces, heat fluxes, etc. Next, several projection strategies are presented and compared, focusing on the experimental convergence order and the commutativity property.

## Weighted Average

The simplest strategy of projection from particles to nodes used by PFEM-2 performs a weighted average of the particle values

$$\phi_j = \frac{\sum_{p=1}^P \phi_p W_j(\mathbf{x}_p^{n+1})}{\sum_p W_j(\mathbf{x}_p^{n+1})} \quad (3.9)$$

where the function  $W_j$ , associated with the node  $j$ , can be either the typical kernel functions used in particle methods such as SPH [Gingold and Monaghan, 1977] or the linear shape functions raised to a power  $\alpha > 0$  (it is  $W_j(\mathbf{x}) = N_j(\mathbf{x})^\alpha$ ), however none of them accomplish the condition presented in Equation (3.8). Previous works [Gimenez and González, 2015, Idelsohn et al., 2014, Becker et al., 2014] have presented the weighted average strategy, but without analyzing its accuracy and convergence.

## Local Minimization

Minimizing some error measure is an excellent strategy to reduce projection errors. In the local minimization approach, every nodal value can be obtained by solving an independent Least Squares approximation problem, which is defined as follows: find the constants  $a_1, a_2, \dots, a_k$  such as minimizing a local error function  $E_l$  which computes the quadratic difference between the particle states  $\phi_p$  at positions  $\mathbf{x}_p$  surrounding a node at position  $\mathbf{x}_j$  and a polynomial function  $\mathbf{P}(\mathbf{x}) = a_1 P_1(\mathbf{x}) + a_2 P_2(\mathbf{x}) + \dots + a_{N_p} P_{N_p}(\mathbf{x})$ , with

$$E_l(a_1, a_2, \dots, a_k) = \frac{1}{2} \sum_{p=1}^P \left( \phi_p - \sum_{k=1}^{N_p} a_k \mathbf{P}_k(\mathbf{x}_p) \right)^2. \quad (3.10)$$

According to the polynomial function used, the method receives different names. If a linear function is employed, i.e. an approximation plane in 2d projection, the strategy is called *Local Least Squares Linear* (LLSL), while if a quadratic function is used, i.e. an approximation paraboloid in 2d projection, the approach is named *Local Least Squares Quadratic* (LLSQ). Once found the coefficients  $a_k$  that minimize the local error, the nodal value is obtained following

$$\phi_j = \sum_k^{N_p} a_k \mathbf{P}_k(\mathbf{x}_j) \quad (3.11)$$

## Global Minimization

Another approach consists in solving a minimization problem where the approximation functions are the same linear shape functions used by the FEM discretization. This leads to solve a unique global system where the unknown coefficients are directly the nodal values searched. In this case the minimization problem is defined as: find the constants  $\phi_1, \phi_2, \dots, \phi_J$  such as minimizing a global error function  $E_g$  which computes the quadratic difference between the particle states  $\phi_p$  at positions  $\mathbf{x}_p$  surrounding a node at position  $\mathbf{x}_j$  and the nodal value  $\phi_j$ , with

$$E_g(\phi_1, \phi_2, \dots, \phi_J) = \frac{1}{2} \sum_{p=1}^P \left( \phi_p - \sum_{j=1}^J \phi_j N_j(\mathbf{x}_p) \right)^2. \quad (3.12)$$

In order to minimize  $E_g$ , Equation (3.12) is derived and equaled to zero, which leads to an equation system of  $J$  unknowns

$$\mathbf{M}\boldsymbol{\phi} = \mathbf{f} \quad (3.13)$$

where  $J$  is the number of nodes,  $\mathbf{M}_{ij} = \sum_p N_i(\mathbf{x}_p)N_j(\mathbf{x}_p)$  is a consistent like-mass matrix,  $\boldsymbol{\phi}$  is the unknowns vector of nodal values, and  $\mathbf{f}_i = \sum_p N_i(\mathbf{x}_p)\phi_p$ . The projection by solving this equation is named *Global Least Squares Consistent* (GLSC).

It should be noted that lumping the matrix, it is  $\tilde{\mathbf{M}}_{ii} = \sum_p \sum_j N_i(\mathbf{x}_p)N_j(\mathbf{x}_p)$ , the original projection approach presented in Equation (3.9) with  $W_j = N_j$  and  $\alpha = 1$  is recovered. This strategy is named *Global Least Squares Lumped* (GLSL).

## Comparison Test

In order to compare the projection methods presented above, a 2D test is designed where a geometrical domain by  $[-1, -1] \times [1, 1]$  is discretized with a reference grid with  $50 \times 50$  nodes conforming 4802 triangles leading to a mesh-size  $H = 0.04$ . Over each element twelve particles are seeded at random positions with the exact solution proposed, which is a four-hill Gaussian represented by the formula

$$\phi^{ex}(x, y, t) = \sin(\pi x) \sin(\pi y). \quad (3.14)$$

For any projection method it is desirable that the solution calculated at nodes corresponds to the exact one. As error measure the root-mean-square (RMS) formula is used for any time

$t$ , which reads

$$RMS(t) = \sqrt{\frac{1}{N} \sum_{j=1}^N (\phi_j^{ex}(t) - \phi_j^{ap}(t))^2} \quad (3.15)$$

where  $N$  is the number of nodes on the mesh,  $\phi^{ex}$  is the analytic reference solution and  $\phi^{ap}$  is the approximated numerical solution.

Table 3.1 shows the RMS error measured employing the projection methods presented in this section which are graphically presented in Figure 3.3 focusing on the convergence rate obtained by each method. Results show that the most accurate strategy is LLSQ followed by GLSC, also both reaching second order of convergence with the mesh-size. The approaches LLSL and GLSL are more inaccurate obtaining also less convergence order, particularly the latter.

A drawback of local minimization methods is that they require the inversion of  $N$  matrices of  $N_p \times N_p$  (one matrix by node), which leads to large cpu-times. Moreover, with these local approaches, at least  $N_p$  particles around each node are required which leads to increase excessively the number of particles in the simulation or need to visit not only the first neighbor layer. Another problem appears on parallel calculations because at nodes over the boundary of the domain decomposition, intensive communication between processors is required. For example, in 3d simulations and using second order polynomials, a matrix of  $10 \times 10$  for each boundary node is needed to communicate between processors which share these nodes. The mentioned drawbacks turn the Local minimization alternative unaffordable in the context of fast algorithms.

In the case of global minimization strategies it is desirable to employ the non-lumped version of the equation system in order to obtain more accurate projections. If the lumped version is used, the convergence with the mesh-size is only linear, whereas using the consistent matrix the convergence is quadratic.

	LLSL	LLSQ	GLSC	GLSL
$\Delta x = 2H$	0.01377	0.00333	0.0052	0.00650
$\Delta x = H$	0.00454	0.00090	0.0014	0.00328
$\Delta x = H/2$	0.00153	0.00023	0.00034	0.00141

Table 3.1: RMS errors for the projection methods presented in this section with different mesh refinements. Twelve particles per element are employed in each test.

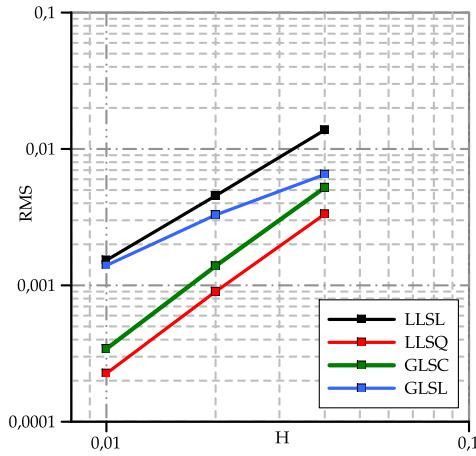


Figure 3.3: Logarithmic plot of the RMS errors using different projection algorithms.

### Bounded minimally-diffusive conservative projection

Last subsection presented that using global minimization techniques is the most efficient strategy to project values from particles to nodes, while the accuracy can be improved employing higher-order approaches, i.e., using GLSC seems to be the best option because of the second order accuracy. However, that assumption could fail in certain cases depending on the shape of the functions to interpolate: when the function is non-smooth (sharp or discontinued) the second order projection presents overshoots and undershoots [Farrell and Maddison, 2011] which can be dangerous when the advected solution must be restricted in a values range, for example the phase fraction. On the other hand, the employment of first order projections leads to conservative and bounded but extremely diffusive solutions and low convergence rate.

The aim of this section is presenting a new strategy which leads to projections which keep the conservativeness and obtain bounded results adding the minimum amount of numerical diffusion. The equation system of the minimization problem (Equation (3.13)), can be derived following the work of Farrel [Farrell et al., 2009] which proposes a conservative interpolation between two meshes  $T_A$  and  $T_B$ . Unlike the mentioned work where the shape functions used by both meshes are piecewise linear, in the current case the source mesh  $T_A$  is a point-collocated mesh being the particles its nodes, while the target mesh  $T_B$  is the background fixed-mesh used by PFEM-2 which has the same discretization as Farrel proposed. Let  $\phi$  be a

function whose integral is to be conserved, i.e.

$$\int_{\Omega} \phi(\mathbf{x}) d\Omega = \int_{\Omega} \Pi(\phi(\mathbf{x})) d\Omega, \quad (3.16)$$

where  $\Pi$  is the projection operator to be described. Considering Equation (3.16) in a weak integral sense:

$$\int_{\Omega} \phi_A N_B^j d\Omega = \int_{\Omega} \phi_B N_B^j d\Omega, \quad (3.17)$$

for each basis function  $N_B^j$  associated with mesh  $T_B$ . This will conserve the integral if the constant function 1 is contained in the span of  $\{N_B^j\}_j$ , which assume no strong Dirichlet boundary conditions are applied within the interpolation procedure. Replacing  $\phi_A$  and  $\phi_B$  with their finite element representations:

$$\int_{\Omega} \sum_p \hat{\phi}_{A_p} \delta(\mathbf{x} - \mathbf{x}_p) N_B^j d\Omega = \int_{\Omega} \sum_i \hat{\phi}_{B_i} N_B^i N_B^j d\Omega, \quad (3.18)$$

for  $p$  ranging over the basis functions of  $T_A$  (i.e. the particles),  $i$  and  $j$  ranging over the basis functions of mesh  $T_B$ ,  $\delta$  the Kronecker delta function, and  $\hat{\phi}$  the discretized functions values. This gives rise to the matrix equation

$$\mathbf{M}_B \hat{\phi}_B = \mathbf{M}_{BA} \hat{\phi}_A \quad (3.19)$$

where  $(\mathbf{M}_B)_{ij} = \int_{\Omega} N_B^i N_B^j d\Omega$  and  $(\mathbf{M}_{BA})_i = \int_{\Omega} N_B^i \delta_A(\mathbf{x} - \mathbf{x}_p) d\Omega = \int_{\Omega} N_B^i(\mathbf{x}_p) d\Omega$ . The system of equations obtained in (3.19) is exactly the same as the obtained in (3.13) but its derivation allows to demonstrate conservativeness. Employing consistent version the boundedness is not guaranteed. On the other hand, using the lumped version of the like-mass matrix  $\mathbf{M}_B$  has no effect on the conservation properties of the interpolation for Lagrange elements, but it has two other major effects: it bounds the resulting solution, and it adds an artificial numerical diffusion such as if the maximum is attained only at a single point, that maximum will be lost [Farrell et al., 2009].

A conservative projection which guarantee boundedness and minimum diffusion (named BCMD for Bounded Conservative Minimum Diffusive projection) can be formulated as follows: computing the consistent Galerkin interpolation, and then selectively apply the minimum of numerical diffusion to the resulting projected values to bound them within the bounds of the field  $\phi_A$  on the particles. This strategy requires to know *a priori* the lower and

upper bound values  $\phi_{max}$  and  $\phi_{min}$  respectively. A *deviation* field  $\phi_{dev}$  is constructed as:

$$\phi_{dev} = \begin{cases} \phi_B - \phi_{max}, & \phi_B > \phi_{max} \\ \phi_B - \phi_{min}, & \phi_B < \phi_{min} \\ 0, & \phi_{min} \leq \phi_B \leq \phi_{max} \end{cases} \quad (3.20)$$

which measures how much unbounded is the consistent approximation  $\phi_B$ , obtaining boundedness when  $\phi_{dev} = 0$  everywhere. The main idea of the algorithm is to spread the overshoot or undershoot nodes onto bounded nodes thought as an iterative process. At each iteration a new field  $\phi_{alt}$  is found solving the diagonal system

$$M_B^L \phi_{alt} = M_B \phi_{dev}. \quad (3.21)$$

Finally, the new interpolating is constructed as:

$$\phi_B \leftarrow \phi_B - \phi_{dev} + \phi_{alt}. \quad (3.22)$$

The iterative sequence solves (3.20), (3.21) and (3.22) where, at each diffusion step, the deviation is spread to neighboring nodes. This update is iterated until the deviation field is zero, or some tolerance is reached. These iterations are cheap, each requiring only a matrix-vector multiplication, an array division, and two vector additions, however it is reported the needed of hundreds of iterations [Farrell et al., 2009].

A simple 1d case is presented. The mesh consists of four nodes conforming three linear elements with  $\Delta x = 1$ , whereas six particles per element are seeded. Two initial solutions are placed over the particles: a smooth solution represented by a sinusoidal and a sharp solution represented by a step. In the Figure 3.4 can be shown the results over mesh after 1 iteration and 10 iterations of a loop consisting on projection-interpolation successive operations, and using a smooth initial solution. It should be noticed that the GLSC approach obtains more accurate results than GLSL, moreover applying the BCMD algorithm the results are similar to the former. Another important feature of the GLSC and BCMD approaches is that both the solution over mesh does not change after the first iteration, which represents the desirable characteristic mentioned in Equation (3.9), i.e. the projection operator is the inverse of the interpolation operator. However, the position of the maximum amplitude of the sinusoidal signal is displaced from the original position. On the other hand, after ten iterations the lumped strategy losses the original shape reaching unacceptable results.

Figures 3.4c and 3.4d shows the results over mesh after 1 and 10 iterations of the loop projection-interpolation using a sharp initial solution. In this case the consistent approach

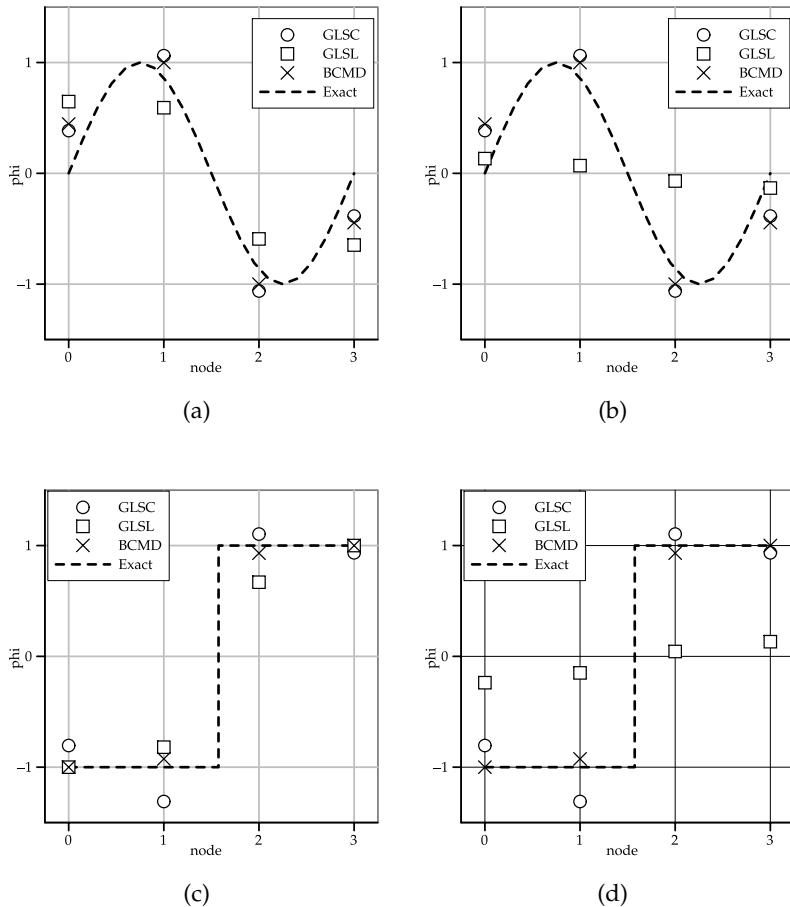


Figure 3.4: Comparison of GLSC, GLSL and BCMD projection strategies. Case with a smooth initial solution after 1 projection-interpolation iteration (Figure 3.4a) and after 10 iterations (Figure 3.4b). Case with a sharp solution after 1 projection-interpolation iteration (Figure 3.4c) and after 10 iterations (Figure 3.4d).

has overshoots and undershoots which can be dangerous when the advected solution must be restricted in a values range. The lumped strategy again suffers from successive deterioration of the solution, but the error after the first step can be acceptable for some cases. The case of transporting sharp functions over the particles is usual in flow simulations. A typical example is the volume fraction in VoF, an overshoot or an undershoot leads to unphysical fluid properties then they are not acceptable. In these cases a diffusive strategy is a proper choice.

Regarding to the BCMD strategy, it has demonstrated to be the most accurate method in every case. However its computational cost is prohibitive when the main attempt is to achieve a fast numerical code. A possibility is to accelerate the convergence with over-relaxation

techniques or upwinding strategies, but this work is out of scope in the current thesis then this projection strategy is left as an ideal proposal. The GLSC projection does not suffer from diffusion, but its unbounded behavior leaves its choice only for fields which its large gradients are represented by several elements, i.e. smooth functions. When the last is not possible, i.e. the change in the function must be captured in a band of a couple of elements, GLSL is the best choice.

### 3.2.3 Variational form of Eulerian stages

Once the convection term is implicitly solved by the particle movement in a Lagrangian framework, the momentum predictor step (Equation (3.2)) of FSM is reduced to

$$\begin{cases} \rho \frac{\widehat{\mathbf{u}}^{n+1} - \widehat{\mathbf{u}}^n}{\Delta t} - \mu \nabla^2 \widehat{\mathbf{u}}^{n+1} = \rho \mathbf{g}^{n+1} - \nabla p^n & \text{in } \Omega \\ \widehat{\mathbf{u}}^{n+1} = \mathbf{u}_D^{n+1} & \text{on } \Gamma \end{cases} \quad (3.23)$$

where the convection term was already solved.

In the space discretization built with the Finite Element Method, the key step is to construct the discrete linear subspaces  $\mathbf{V}_h \in \mathbf{V}$ ,  $\mathbf{V}_{0h} \in \mathbf{V}_0$  and  $Q_h \in Q$  that approximate the continuous spaces. Let  $\mathbf{V}_h$  and  $Q_h$  be the finite element spaces to interpolate vector and scalar functions, respectively, constructed in the usual manner from a finite element partition  $\Omega = \cup_e \Omega^e$  and  $\emptyset = \cap_e \Omega^e$ ,  $e = 1, \dots, nel$ , where  $nel$  is the number of elements. In the case of one-phase flows the same interpolation is employed for both the velocity and the pressure. In particular continuous and linear  $P1 - P1$  interpolations. The subspace  $\mathbf{V}_{0h}$  incorporates the Dirichlet conditions for the velocity.

Using the mentioned spaces, the problem (3.2) can be written as: Given a convected velocity  $\widehat{\mathbf{u}}^{n+1} \in \mathbf{V}_h$  at time  $t^{n+1}$  and a old pressure  $p^n \in Q_h$ , find the predicted velocity  $\widehat{\mathbf{u}}^{n+1} \in \mathbf{V}_{0h}$ , such that for all  $\mathbf{N} \in \mathbf{V}_{0h}$

$$\int_{\Omega} \mathbf{N} \rho \frac{\widehat{\mathbf{u}}^{n+1} - \widehat{\mathbf{u}}^n}{\Delta t} d\Omega + \int_{\Omega} \nabla \mathbf{N} \mu \nabla \widehat{\mathbf{u}}^{n+1} d\Omega - \int_{\Gamma} \mathbf{N} \mu \nabla \widehat{\mathbf{u}}^{n+1} \cdot \mathbf{n} d\Gamma = \int_{\Omega} \nabla \mathbf{N} \rho \mathbf{g}^{n+1} - \nabla p^n d\Omega \quad (3.24)$$

On the other hand, the problem to find the new pressure using the Equation (3.4) can be written as: Given a predicted velocity  $\widehat{\mathbf{u}}^{n+1} \in \mathbf{V}_{0h}$  at time  $t^{n+1}$  and a old pressure  $p^n \in Q_h$ ,

find  $p^{n+1} \in Q_h$  by solving the discrete variational problem:

$$\int_{\Omega} \nabla N \frac{\Delta t}{\rho} \nabla (p^{n+1} - p^n) d\Omega = \int_{\Omega} N \nabla \cdot \hat{\mathbf{u}}^{n+1} d\Omega \quad (3.25)$$

Finally, the third Eulerian stage to solve is the velocity correction (Equation (3.5)) which can be expressed as: Given a predicted velocity  $\hat{\mathbf{u}}^{n+1} \in \mathbf{V}_{0h}$  and a pressure  $p^{n+1} \in Q_h$  both at time  $t^{n+1}$ , find  $\mathbf{u}^{n+1} \in \mathbf{V}_h$  by solving the discrete variational problem:

$$\int_{\Omega} \mathbf{N} \rho \mathbf{u}^{n+1} d\Omega = \int_{\Omega} \mathbf{N} \rho \hat{\mathbf{u}}^{n+1} d\Omega - \Delta t \int_{\Omega} \mathbf{N} \nabla (p^{n+1} - p^n) d\Omega \quad (3.26)$$

Using the Galerkin method [Zienkiewicz and Morgan, 1983] where the shape functions employed to approximate the variables are the same as the test functions, i.e.  $\mathbf{u} = \sum_i \mathbf{N}_i \mathbf{u}_i$  for the velocity and  $p = \sum_i N_i p_i$  for the pressure, the FEM problem for an incompressible flow can be written as:

$$\left( \mathbf{M} \left( \frac{\rho}{\Delta t} \right) + \mathbf{K}(\mu) \right) \hat{\mathbf{u}}^{n+1} = \mathbf{M} \left( \frac{\rho}{\Delta t} \right) \mathbf{u}^n - \mathbf{G} p^n \quad (3.27)$$

$$\mathbf{K} \left( \frac{\Delta t}{\rho} \right) p^{n+1} = \mathbf{B} \hat{\mathbf{u}}^{n+1} + \mathbf{K} \left( \frac{\Delta t}{\rho} \right) p^n \quad (3.28)$$

$$\mathbf{M}(\rho) \mathbf{u}^{n+1} = \mathbf{M}(\rho) \hat{\mathbf{u}}^{n+1} - \Delta t \mathbf{G} [p^{n+1} - p^n] \quad (3.29)$$

where  $\mathbf{M}_{ij} = \int_{\Omega} N_i N_j d\Omega$  and  $\mathbf{K}_{ij} = \int_{\Omega} \nabla N_i \nabla N_j d\Omega$  are the classical finite element mass and stiffness matrices, while  $\mathbf{B}_{ij} = \int_{\Omega} \nabla N_i N_j d\Omega$  and  $\mathbf{G}_{ij} = \int_{\Omega} N_i \nabla N_j d\Omega$  are the discrete expression of the divergence and gradient operators respectively, accomplishing that  $\mathbf{B}_{ij} = \mathbf{G}_{ji}$ .

## Pressure Stabilization

Because of not including the convective term, the momentum equation (3.27) does not need any stabilization. Equation (3.28) on the contrary must be stabilized in space since an unstable velocity-pressure pair is chosen. Any spatial stabilization method can be used in principle [Codina, 2000], then in this work is chosen the one presented by Codina in [Codina, 2001] which consists in adding to the variational form the term  $\tau [\nabla p^{n+1} - \boldsymbol{\pi}^n]$  where  $\boldsymbol{\pi}$  is the continuous function of  $\nabla p$ , and  $\tau$  is a stabilization parameter which depends on the time-step, mesh-size, viscosity and local velocity. Therefore, Equation (3.28) can be rewritten as

$$\mathbf{K} \left( \frac{\Delta t}{\rho} + \tau \right) p^{n+1} = \mathbf{B} \hat{\mathbf{u}}^{n+1} + \mathbf{K} \left( \frac{\Delta t}{\rho} \right) p^n + \mathbf{G}(\tau) \boldsymbol{\pi}^n \quad (3.30)$$

with

$$\tau = \left[ \left( \frac{1}{\Delta t} \right)^2 + \left( \frac{2|\hat{\mathbf{u}}^{n+1}|}{\Delta x} \right)^2 + 9 \left( \frac{4\mu}{\Delta x^2} \right)^2 \right]^{-\frac{1}{2}} \quad (3.31)$$

It must be remarked that the possibility of treating explicitly the projection  $\pi$  greatly simplifies the numerical implementation of this stabilization technique because only an  $L_2$  projection needs to be performed at the end of each time-step. This is computationally very inexpensive, especially if the like-mass matrix  $\mathbf{M}$  involved in the process is approximated by the diagonal one obtained from a standard lumping technique.

### Particle updating

Once found the nodal values of the unknowns at time  $n + 1$ , the states transported by the particles must be updated to synchronize the field representations. As mentioned before, particles transport the actual information of the fields. Then, a problem appears at the moment of updating particles states because it always implies interpolation from nodal states, task which never is diffusive-free.

A first option consists in using a simple interpolation with the linear shape functions basis. The state of the particle  $\phi_p^{n+1}$ , i.e. the value of the unknown  $\phi$  on the particle  $p$  at time  $n + 1$ , is obtained as

$$\phi_p^{n+1} = \sum_j \phi_j^{n+1} N_j(\mathbf{x}_p^{n+1}) \quad (3.32)$$

which would replace the particle value with the interpolation. This strategy destroys the original information of the particle which leads to excessive diffusion of the method. An improved option is to correct but not replace the particle states. The correction is done interpolating over the particle only the difference between the nodal values after projection  $\hat{\phi}_j^{n+1}$  and the current state  $\phi_j^{n+1}$ , i.e. after velocity correction. This leads to an incremental modification as

$$\phi_p^{n+1} = \hat{\phi}_p^{n+1} + \sum_j (\phi_j^{n+1} - \hat{\phi}_j^{n+1}) N_j(\mathbf{x}_p^{n+1}) \quad (3.33)$$

which not only reduces the artificial diffusion but also allows for a different scales to be used in the particles and the mesh without losing information. This means that even if sudden changes in element size exist and therefore the mesh is unable to see the smaller scale, the particles will conserve the resolution of this scale and it will be possible to recover. This

treatment must be done with every unknown transported by particles, being in the case of incompressible flows  $\phi = \mathbf{u}$ , the velocity.

The advantage of using the particle-mesh projection strategy GLSC or its bounded version presented in the examples in Figure 3.4, i.e. the minimization of the numerical diffusion when mesh-particles interpolation is done, is also relevant in this stage in order to achieve more accurate results.

### 3.2.4 Algorithm synopsis

It is assumed that all fluid variables are known at time  $t^n$  for both the particles and the mesh nodes. Subindexes  $()_i$  and  $()_j$  represent a generic mesh node  $j$  while subindex  $()_p$  a generic particle  $p$ . Let  $N$  the finite element linear basis functions. According to this notation, the steps are:

---

**Algorithm 1** - Time-Step PFEM-2 for one-phase incompressible fluids.

---

1. Convective Stage:

$$\begin{cases} \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{u}^n(\mathbf{x}_p^\tau) d\tau \\ \hat{\mathbf{u}}_p^{n+1} = \mathbf{u}_p^n \end{cases}$$

2. Projection Stage:

$$\mathbf{M}_{ij} \hat{\mathbf{u}}_j^{n+1} = \mathbf{M}_{ip} \hat{\mathbf{u}}_p^{n+1}$$

3. Momentum Stage:

$$\left( \mathbf{M}\left(\frac{\rho}{\Delta t}\right) + \mathbf{K}(\mu) \right) \hat{\mathbf{u}}^{n+1} = \mathbf{M}\left(\frac{\rho}{\Delta t}\right) \hat{\mathbf{u}}^{n+1} - \mathbf{G} p^n$$

4. Poisson Stage:

$$\mathbf{K}\left(\frac{\Delta t}{\rho} + \tau\right) p^{n+1} = \mathbf{B} \hat{\mathbf{u}}^{n+1} + \mathbf{K}\left(\frac{\Delta t}{\rho}\right) p^n + \mathbf{B}(\tau) \pi^n$$

5. Correction Stage:

$$\begin{aligned} \mathbf{M}(\rho) \mathbf{u}^{n+1} &= \mathbf{M}(\rho) \hat{\mathbf{u}}^{n+1} - \Delta t \mathbf{G} (p^{n+1} - p^n) \\ \rho_p \mathbf{u}_p^{n+1} &= \rho_p \hat{\mathbf{u}}_p^{n+1} + \sum_j \delta \mathbf{u}_j^{n+1} N_j(\mathbf{x}_p^{n+1}) \end{aligned}$$


---

where  $\mathbf{x}$  is a spatial coordinate, and  $\delta \mathbf{u} = \mathbf{u} - \hat{\mathbf{u}}$ .



# Chapter 4

## High performing implementation

The hybrid spatial discretization used by PFEM-2 allows to use the optimum strategy to calculate each equation term. Convection terms are solved by particles in a Lagrangian way using the X-IVS method. On the other hand, forces and diffusive terms are solved on a mesh in an Eulerian way using a classic FEM approach. However, keeping in memory the data of the mesh and of the cloud of particles represents an important storage cost that limits implementations that run only on a single computer. Therefore, it is imperative to develop a distributed-memory implementation.

In order to develop a multi-machine implementation for PFEM-2, it is essential not to reinvent the wheel. Reusing code amortizes the formidable software development effort required to support parallel unstructured mesh-based simulations. In the open-source community there are several object-oriented toolkits and libraries [Mackerle, 2004], [Sbalzarini et al., 2006], [Dadvand et al., 2010], which include domain decomposition, shape functions of many orders, numerical integration, assembling and solving equation systems, etc. These packages can be extended by developers for their specific application libraries.

Particularly, the home-made parallel numerical framework developed uses as basis the library libMesh [Kirk et al., 2006]. It is an object-oriented library written in C++ to solve FEM problems with adaptive refinement and coarsening (AMC), which performs the communication between nodes through the standard MPI (Message Passing Interface). Several libraries are also included in the suite, but the main ones are PETSc [Balay et al., 2012] for the solution of linear systems on both serial and parallel platforms, and METIS [Karypis and Kumar, 1999] and ParMETIS [Karypis and Kumar, 1998] which implement a domain-decomposition based on graph partitioning schemes for serial or distributed meshes,

respectively. Two main classes are added to the libMesh library: `Particle` and `PFEM2`. The former, which derives from the class `Point`, represents individually each particle of the system, whereas the latter encapsulates the entire library and has two main attributes: the cloud of particles (a `Particle`'s sequential array) and an instance of the libMesh class `EquationSystem`, which contains the mesh data and FEM systems to solve.

Next sections present a summary about features of the implementation in order to guarantee good parallel performance, mainly measuring it in terms of scalability. It must be taken into account that this thesis is the continuation work of the Master degree thesis of the author [Gimenez, 2014], which was focused in the high performing implementation of the PFEM-2 method, then interested readers should address the mentioned thesis and the published paper [Gimenez et al., 2014] for a deeper explanation and validation.

## 4.1 Domain decomposition

Critical to any implementation of distributed computing is the methodology used to distribute the global computational task to the local processor resources. In a numerical simulation, the tasks are generally aligned with integration points on a body in space; hence dividing the physical space may be used to parallelize a problem. That strategy is adopted by most FEM parallel implementations through domain-decomposition methodology, where a problem domain is geometrically divided into sub-domains that can then be distributed across the available computational resources. The sub-domains exchange data with one another through their boundaries.

On the other hand, most of the particle methods, including the one presented in this work, have a natural parallelism because force calculations and position updates can be done simultaneously for all particles. Two main ideas have been exploited to achieve this parallelism [Kacianauskas et al., 2010]. The first method is called atom decomposition of the workload, since the processor computes forces on its particles no matter where they move in the simulation domain because the assignment remains fixed for the duration of the simulation. The second method consists simply in the above mentioned spatial decomposition. Oriented to PFEM-2, the former has shown good performance for shared memory computers [Gimenez, 2014], but the global character of the employed algorithms produces inter-processor communication overhead on distributed memory machines, because an updated copy of the entire mesh in each processor is needed.

---

Therefore, the domain-decomposition is also selected for particles. Finally, the selection of domain-decomposition techniques for both mesh and particles in the implementation of PFEM-2 is here called dual particle-mesh distribution.

Regarding the update calculations, domain decomposition requires significant communication between the sub-domains and/or some degree of zone duplication to ensure accuracy. These zones along the segmented planes are called *ghost* or *virtual*. For the typical first order strategy of FEM used by PFEM-2, the zone simply refers to any immediately adjacent nodes. However, to calculate a particles trajectory using large time-steps, this dimension may extend through several layers of elements, unless a particle leaves the sub-domain and is immediately sent to the neighbor processor to continue the calculation. This approach adds inter-processor communication in the X-IVS stage, but it eliminates the uncertainty of not knowing how many layers of ghost elements are needed to perform the trajectory. Moreover, layers definition can be a forbidding problem when unstructured meshes are used.

As it was mentioned above, the basic principle of parallelizing the X-IVS algorithm in the domain-decomposition manner is that each CPU calculates the trajectories of its set of particles (those that reside in it at the given time-step). When a particle, due to its movement, changes the partition where it belongs, the particle data are transferred to the CPU in control of the partition the particle has entered and the control of that particle is given to that CPU [Sbalzarini et al., 2006].

An option to implement particle transfer could be an asynchronous transfer, in which the particle data is packed in a continuous data buffer and sent to the CPU that is in control of the partition on the other side of the partition boundary. When there are no more particles to be tracked on the partition, the algorithm leaves the tracking loop and enters in a dummy loop where it checks if a particle data message has been sent to it from another CPU. If so, it accepts the message, unpacks the data, proceeds to the tracking loop and starts tracking the newly acquired particle. This approach has an issue to calculate the termination condition because a synchronous point for all CPUs is required [Kaludercic, 2004]. Then, attempt to minimize the communication and due to the need for a synchronous point, the buffer of particles is only packed but not transferred until the particle loop ends. This strategy transfers particles and synchronizes simultaneously, and is selected for the current implementation.

To perform parallel particle tasks, a `ParticleCommunication` class is implemented, which allows each sub-domain to send and receive a set of particles through the MPI method

AllToAllv. This strategy is easy to implement; however, because it is a collective operation, only useful when few processors are used. Other possibilities must be analyzed, trying to reduce the communication when a large number of processors are used. Interchanging information only with the neighbor sub-domains through point-to-point communications can be the best alternative.

---

**Algorithm 2 - X-IVS manager over distributed-memory**


---

```

int ini_ip = 0;
while(1){
    std::map<int, std::vector<int> >particles2send;
    np = vP.size();
    for(unsigned int ip=ini_ip;ip<np;ip++){
        int c=0, pid_send;
        bool next_ddt = true;
        while(next_ddt){
            c = integrateSubStep(vP[ip],pid_send);
            switch(c){
                case 0:// substep complete
                    break;
                case 1://out of domain
                    next_ddt = false;
                    break;
                case 2://out of sub-domain
                    next_ddt = false;
                    particles2send[pid_send].push_back(ip);
                    break;
                case 3://time-step complete
                    next_ddt = false;
                    break;
            }
        }
        ini_ip=np;
        next = particles2send.size();
        Parallel::max(next); //max All_reduce
        if(!next) break; //no particles to send
        ParticleCommunication().interchangeParticles(particles2send,vP); //Alltoallv
    }
}

```

---

Because a particle can cross more than two sub-domains in a time-step, an external loop is needed which breaks when no more particles are transferred. Algorithm 2 presents a transcription of the code where the external loop is observed alongside the loop over the particles and the stop condition (when all processors do no have any more particles to

transfer). The `for` loop, whose iteration starts from the last particle analyzed to the last particle in the current array, integrates the trajectory of each particle computing each sub-time-step in a `while` loop. There are four options at each sub-time-step: in the first case, the particle has completed its sub-time-step and must calculate the rest of the trajectory; in the second case the particle has left out the domain boundary and its computation finishes; in the third case, the particle has crossed to other sub-domain (in this case, it is queued to be sent to the other processor); finally, in the last case the particle finishes the entire time-step. Figure 4.1 represents graphically every case after moving a particle.

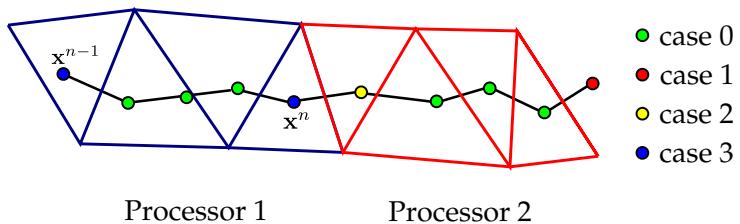


Figure 4.1: Graphical representation of every option when a particle is transported. In case 0 the substep was done successfully, in case 1 the particle leaves the domain, in case 2 the particle leaves the subdomain and must be sent to the new owner, and in case 3 the particle ends successfully the time-step integration.

## 4.2 Load balancing

In a parallel FEM calculation, the domain distribution must be done so that the number of elements assigned to each processor is the same, and the number of adjacent elements assigned to different processors is minimized. The goal of the first condition is to balance the computations among the processors. The goal of the second condition is to minimize the communication resulting from the placement of adjacent elements to different processors.

The graph partitioning strategy implemented by the library Metis can be used to successfully satisfy these conditions by first modeling the finite element mesh with a graph, and then partitioning it into equal parts. The user can control the distribution associating a positive weight  $\eta(v)$  with each edge  $v$  of the graph.

For the case of FEM simulations, a typical selection of the weights is  $\eta(v) = \eta_n(v) = \#dofs\_of\_element$ , because the amount of computational task of each element is directly proportional to the number of degrees of freedom. Then, because the current implementation uses only linear simplices, after a domain decomposition each processor has approximately

---

the same number of elements. At the beginning of the simulation, each processor creates a fixed number of particles on each own element; this guarantees that the initial load of particles over processors is also balanced. However, when the simulation starts, this distribution can be modified according to the particle movement itself.

The number of particles on each element rarely keeps constant, being in general very common to find some elements empty of particles or some elements with many more particles than that desired. Beyond the balancing problem, the main drawback is the loss of accuracy of the solution; then PFEM-2 solves this issue seeding and removing particles, as explained in Section 3.2.2 leading to keep the number of particles approximately constant.

An approximately constant number of particles in each sub-domain only guarantees a proper load of particles on each processor. However, as in this thesis an adaptive sub-stepping integrator to move the particles is usually employed, that balanced distribution does not ensure a balanced work-load over each CPU, similarly to the case of static scheduling over shared-memory.

The parallel architecture in multi-core environments makes possible a dynamic scheduling to distribute the work over processors because each particle has access to the entire domain data. This feature is not possible in multi-node environments with the domain geometrically divided but, as it was explained above, the atom decomposition is not a proper choice because adds more serious problems.

A possible solution to the work-load balancing of the X-IVS stage on distributed-memory consists in a dynamic solution which, starting from an initial partition, interchanges few elements between processors according to how the solution varies in time. In this sense both, the number of elements that are moved and the edge-cut of the resulting partitioning are optimized. However, many other parameters must be optimized, as the time interval between those repartitions and partition quality; therefore, exhaustive research is needed to achieve successful results with this solution.

An easier option to implement is using a static approach which includes information about the  $CFL_h$  on the partition algorithm through a weights array  $\eta_w(V) \neq \eta_n(V)$ , where the weighted partition is done only at the beginning of the simulation. It requires knowing or estimating the solution previously, which is impossible in most of the cases. Therefore, an initial simulation must be done in which the weights are calculated, saved, and used in the partitioning algorithm of a new run.

### 4.3 Other stages

X-IVS implementation on distributed-memory is the main novelty of this Chapter. However, there are other relevant stages of the algorithm, whose implementation should be detailed.

There are many mesh-based calculations in a PFEM-2 time-step: momentum predictor, pressure calculation and velocity correction that are calculated by FEM. Employing a fixed background mesh, the benefits of having a constant matrix for the pressure equation may be exploited. This matrix is initially factorized and then used as a preconditioner. This feature also appears for the implicit diffusion step if the viscosity is not time-dependent. Then, libMesh library must be used smartly, assembling only once those matrices and specifying parameters to reuse the preconditioner. A complete Cholesky factorization is not affordable with large problems or in parallel executions with PETSc. Hence, iterative preconditioners like diagonal, incomplete Cholesky or algebraic multigrid are employed.

On the projection stage, the main parallel issue is the calculation of the state of the nodes over the boundary between sub-domains. To perform this task, ghost nodes are also calculated by processors. In the projection algorithm, at the end of the loop over particles, the processor owner of the boundary node receives the partial state of ghost node from the neighbor processor and calculates the complete nodal state. Because projection methods are based on weighted averages (which is a non associative operation), the parallel calculation must be done carefully: they must not be calculated as partial averages on each processor and later averaging the result again on the owner processor. Proper implementation requires, for each ghost node, calculating and sending two partial summations (numerator and denominator) and performing the division only in the owner CPU.

### 4.4 Performance of the implementation

As a test case, the results solving incompressible flow around a cylinder in three dimensions at  $Re = 1000$  are presented in this Section. Numerical and physical parameters employed and numerical results obtained (drag and lift coefficients among others) are reported in [Gimenez, 2014] and [Gimenez et al., 2014] where proper accuracy is obtained.

In order to tests the scalability of the implementation over different networks, two different clusters were employed. The first one has nodes Intel(R) Core(TM) i7-3930K CPU @ 3.20GHz with 16Gb RAM interconnected with 1Gb Ethernet, while the second has dual socket nodes, with Intel Xeon E5-2600 CPUs and 64Gb RAM, interconnected with IB-QDR 40 Gbps. For

each case the load distributor had employed  $\eta_n(V)$ , ensuring that the lower memory capacity in the first cluster does not interfere on the performance.

Figure 4.2 presents the speedup  $S_n$  obtained, which is defined as

$$S_n = \frac{T_1}{T_n}, \quad (4.1)$$

where  $T_1$  and  $T_n$  are the total computing time using 1 and  $n$  processes respectively. Almost two million of degrees of freedom were used and the total simulation time was 100[s]. The performance of the implementation decays fast when moderate or large number of processes are used in a simulation over a slow network, being largely improved by those obtained over an Infiniband network. The reduction of latency, the increment of transfer bandwidth and the dedicated hardware are relevant advantages of the Infiniband network that must be taken into account for choosing a simulation platform.

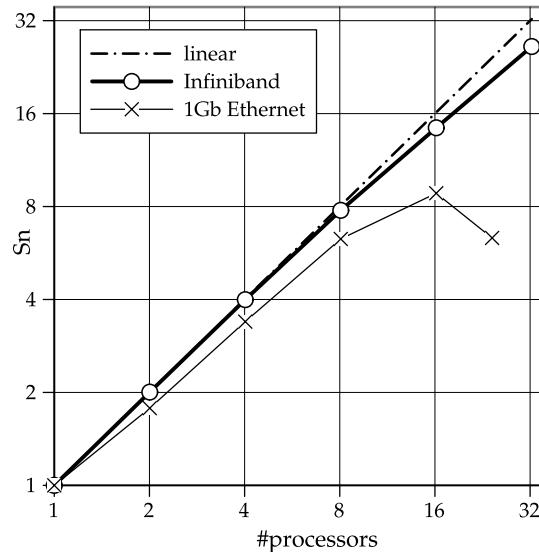


Figure 4.2: Speed-Up comparison between simulations over an Infiniband cluster and a 1 Gigabit Ethernet cluster. Test: flow around a cylinder in 3d.

On the other hand, Table 4.1 shows the percentage of computing time which requires each time-step in the entire simulation. In the particular case of large CFL, the convection stage dominates the computing time. That stage and the projection conform the Lagrangian part, which spends around the 50% of the simulation. It should be noticed that the weights array employed in the partitioning favors the scalability Eulerian stages, diminishing the obtained by particles, but it guarantees similar speed-ups for every stage, allowing to obtain an appreciable  $S_{32} = 26\times$ .

---

	Convection	Projection	Momentum	Poisson	Correction	Total
percentage	40%	12%	15%	17%	10%	100%
$S_{32}$	$26\times$	$22\times$	$27\times$	$24\times$	$29\times$	$26\times$

Table 4.1: Relative computing time and speed-up for each stage of the algorithm PFEM-2 over an Infiniband cluster.

More details about the simulation can be found in [Gimenez et al., 2014], where a comparison between the total time spent by PFEM-2 compared with the widely used CFD software called OpenFOAM® [Weller et al., 1998] is also presented. The speed-up reached by PFEM-2 is similar to that achieved by the CFD software, but a factor approximately  $3\times$  was obtained at the same level of accuracy thanks to the possibility of using larger time-steps. In this mode, the implementation has demonstrated good behavior over a cluster dedicated to scientific computing showing efficiencies of about 80% with 32 processes, which guarantees a good enough starting point of a massively parallel code.



## Chapter 5

# Analysis of Lagrangian and Eulerian errors

The objective of this section is to demonstrate why PFEM-2 is more accurate than a classical Eulerian FEM in certain particular cases when large time-step and/or coarse meshes are used. This shows that a proper manner to improve the efficiency of the algorithms in order to take advantages of the increasing computer power is the use of this particle-based method. The analysis presented in this section is valid for the case of a spatial discretization using piecewise linear finite elements.

The mathematical model of scalar and vector field transport equation has several complexities which lead to almost unaffordable analytic solutions to be used as reference for formal error analysis. In order to circumvent such a drawback in this section manufactured test cases are developed to be used as good candidates for getting exact solutions to our test problems. Several control parameters for the exact solution are used to place the problem in various conditions from some beneficial to Lagrangian schemes to other more beneficial to an Eulerian one. Therefore, in first place the case of scalar transport will be used to present the comparison methodology and some manufactured examples which validate the error formulas. Then, the Navier-Stokes case is studied presenting also some validation tests based on the manual construction of an analytical solution.

## 5.1 Solving a scalar transport equation

The differential equation to solve is the scalar transport equation, which written in Eulerian (fixed) frame reads

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = Q(\mathbf{x}) \quad (5.1)$$

where  $T = T(\mathbf{x}, t) = T^t(\mathbf{x})$  is the scalar unknown,  $\mathbf{u}$  is the velocity and  $Q$  includes the source term  $q(\mathbf{x})$ , the diffusion term  $\nabla \cdot (k \nabla T)$ , and the linearized version of the reaction term  $cT$ , etc, being

$$Q(\mathbf{x}) = q(\mathbf{x}) + \nabla \cdot (k \nabla T) + cT + \dots \quad (5.2)$$

To write Equation (5.1) in the Lagrangian (mobile) frame, it is necessary to use the material derivative  $D/Dt$ , which condenses the temporal derivative and the convective term into a unique term. However, as presented in Section 3.2.1, the transport equation must be coupled with the particles trajectory, obtaining

$$\begin{cases} \frac{DT}{Dt} = Q(\mathbf{x}_p) \\ \frac{D\mathbf{x}_p}{Dt} = \mathbf{u} \end{cases} \quad (5.3)$$

where  $T = T(\mathbf{x}_p^t, t) = T^t(\mathbf{x}_p^t)$  is the scalar unknown. Here the superscript  $t$  means the time dependency and the subscript  $p$  represents the particle itself.

Equations (5.1) and (5.3) are integrated in time on an Eulerian frame using a linear method named  $\theta$ -method:

$$T^{n+1}(\mathbf{x}) = T^n(\mathbf{x}) + \int_n^{n+1} (Q^t - \mathbf{u}^t \nabla T^t) dt \approx T^n(\mathbf{x}) + (Q - \mathbf{u} \nabla T)^{n+\theta} \Delta t \quad (5.4)$$

where  $f^{n+\theta} = (1 - \theta)f^n + \theta f^{n+1}$  is the linear in time interpolation for any function  $f$ . Therefore, the time integration error of a function  $f^t$  for  $\theta = 1/2$  is proportional to the second derivative of the function and  $\Delta t^2$ , i.e.:

$$\int_n^{n+1} f^t dt \approx f^{n+\theta} \Delta t \pm \epsilon \Delta t \text{ with } \epsilon = \mathcal{O}\left(\frac{\partial^2 f}{\partial t^2} \Delta t^2\right). \quad (5.5)$$

In the case of time integration in the Lagrangian frame, there are several options to evaluate that integral [Idelsohn et al., 2015]. The standard  $\theta$ -method reads:

$$\begin{cases} T^{n+1}(\mathbf{x}_p^{n+1}) \approx T^n(\mathbf{x}_p^n) + Q^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \\ \mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + \mathbf{u}^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \end{cases} \quad (5.6)$$

Another option is performing the integration with the moving particles following the streamlines (taking into account intermediate positions), splitting the integration into an explicit part and an implicit part. Therefore, the trajectory and temperature integrations are decoupled employing two different values for  $\theta$ , i.e.,

$$\begin{cases} T^{n+1}(\mathbf{x}_p^{n+1}) \approx T^n(\mathbf{x}_p^n) + (1 - \theta_1) \int_n^{n+1} Q^n(\mathbf{x}_p^t) dt + \theta_1 Q^{n+1}(\mathbf{x}_p^{n+1}) \Delta t \\ \mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + (1 - \theta_2) \int_n^{n+1} \mathbf{u}^n(\mathbf{x}_p^t) dt + \theta_2 \mathbf{u}^{n+1}(\mathbf{x}_p^{n+1}) \Delta t \end{cases} \quad (5.7)$$

The case of performing the unknown integration following streamlines with  $\theta_1 = 1/2$ , and the trajectory, also following streamlines, but with  $\theta_2 = 0$ , has been named X-IVAS+implicit correction [Idelsohn et al., 2013b]. On the other hand, if the choice for unknown integration is  $\theta_1 = 1$  the method is named X-IVS, which was presented in Section 3.2.1. The latter integration will be used in all the examples presented here when using a Lagrangian frame. For the error analysis, the standard  $\theta$ -method will be considered, assuming that the X-IVS integration improves the results and decreases the integration errors because they take into account all the intermediate positions of the particles, but without evaluating this difference.

### 5.1.1 Eulerian errors

As presented in Equation (5.5), the linear  $\theta$ -method introduces a minimum error for  $\theta = 1/2$  which is proportional to the second derivative of the integrated function and the square of the time-step, i.e.:

$$T^{n+1}(\mathbf{x}) = T^n(\mathbf{x}) + \int_n^{n+1} (Q^t - \mathbf{u}^t \nabla T^t) dt \approx T^n(\mathbf{x}) + [(Q - \mathbf{u} \nabla T)^{n+\theta} \pm \epsilon_t] \Delta t \quad (5.8)$$

with the time integration error

$$\epsilon_t = \mathcal{O}\langle \frac{\partial^2}{\partial t^2} [Q - \mathbf{u} \nabla T] \Delta t^2 \rangle \Delta t = \mathcal{O}\langle [Q - \mathbf{u} \nabla T] \cdot \Delta t^2 \rangle \Delta t. \quad (5.9)$$

On the other hand, the FEM approximation of the functions and the space derivatives introduces spatial errors. As it was mentioned, this analysis only considers linear finite elements approximations of the unknown, therefore the spatial errors are proportional to the second derivative of the functions and the square of the mesh-size, i.e.:

$$T^{n+1}(\mathbf{x}) = T^n(\mathbf{x}) + \int_n^{n+1} (Q^t - \mathbf{u}^t \nabla T^t \pm \epsilon_x) dt \approx T^n(\mathbf{x}) + [(Q - \mathbf{u} \nabla T)^{n+\frac{1}{2}} \Delta t \pm \epsilon_x] \Delta t \pm \epsilon_t \quad (5.10)$$

with the spatial discretization error

$$\epsilon_x = \mathcal{O}\left(\frac{\partial^2}{\partial x^2} [Q - \mathbf{u} \nabla T] \Delta x^2\right) \Delta t. \quad (5.11)$$

Finally, avoiding higher order terms, the unknown function after a time-step using the Eulerian framework is:

$$T^{n+1}(\mathbf{x}) = T^n(\mathbf{x}) + (Q - \mathbf{u} \nabla T)^{n+\theta} \Delta t \pm \epsilon_x + \epsilon_t = T^n(\mathbf{x}) + (Q - \mathbf{u} \nabla T)^{n+\theta} \Delta t \pm \epsilon^E \quad (5.12)$$

with  $\epsilon^E = \epsilon_x + \epsilon_t$ .

### 5.1.2 Lagrangian errors

In this subsection first a general analysis taking into account the whole problem involved in a Lagrangian formulation, i.e. solving not only for the scalar field, also for the particle trajectories, is done. Afterwards a specific analysis of projection errors for PFEM-2 scheme due to the mapping of the field between particles and the mesh is presented considering the importance of this projection stage in the global error of the method.

#### General analysis

In the case of the Lagrangian frame, the errors in the numerical evaluation of the unknown function and the particle position are:

$$\begin{cases} T^{n+1}(\mathbf{x}_p^{n+1}) \approx T^n(\mathbf{x}_p^n) + Q^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \pm \mathcal{O}\langle Q'' \Delta x^2 \rangle \Delta t \pm \mathcal{O}\langle Q'' \Delta t^2 \rangle \Delta t \\ \mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + \mathbf{u}^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \pm \mathcal{O}\langle \mathbf{u}'' \Delta x^2 \rangle \Delta t \pm \mathcal{O}\langle \mathbf{u}'' \Delta t^2 \rangle \Delta t \end{cases} \quad (5.13)$$

The error in the evaluation of the particle position  $\epsilon_x^p = \mathcal{O}\langle \mathbf{u}'' \Delta x^2 \rangle \Delta t \pm \mathcal{O}\langle \mathbf{u}'' \Delta t^2 \rangle \Delta t$  also introduces an error in the evaluation of the unknown function. Performing a series expansion around  $\mathbf{x}_p^{n+1}$ , i.e.

$$T^{n+1}(\mathbf{x}_p^{n+1} + \epsilon_x^p) = T^{n+1}(\mathbf{x}_p^{n+1}) + \nabla T \epsilon_x^p + \mathcal{O}\langle T''(\epsilon_x^p)^2 \rangle \quad (5.14)$$

and replacing in the first equation of (5.13), it is

$$T^{n+1}(\mathbf{x}_p^{n+1}) \approx T^n(\mathbf{x}_p^n) + Q^{n+\theta}(\mathbf{x}_p^{n+\theta}) \Delta t \pm \mathcal{O}\langle Q'' \Delta x^2 \rangle \Delta t \pm \mathcal{O}\langle Q'' \Delta t^2 \rangle \Delta t \pm \nabla T \epsilon_x^p \quad (5.15)$$

the expression for the Lagrangian error  $\epsilon^L$  can be found

$$\epsilon^L = \pm \mathcal{O}\langle (Q'' \pm \mathbf{u}'' \nabla T) \Delta x^2 \rangle \Delta t \pm \mathcal{O}\langle (Q'' \pm \mathbf{u}'' \nabla T) \Delta t^2 \rangle \Delta t \quad (5.16)$$

Comparing the Eulerian errors expression in (5.12) with the Lagrangian ones in (5.16), the main differences are in the terms:

$$\underbrace{[\mathbf{u} \nabla T]'' \Delta x^2 \pm [\mathbf{u} \nabla T] '' \Delta t^2}_{\text{Eulerian}} \neq \underbrace{\mathbf{u}'' \nabla T \Delta x^2 \pm \mathbf{u}'' \nabla T \Delta t^2}_{\text{Lagrangian}} \quad (5.17)$$

The difference presented in Equation (5.17) leads to a big advantage of the Lagrangian framework against its Eulerian counterpart for some problems. For instance, in the standard convection-diffusion problem of a non-constant unknown, where the convective field is known and has a constant or nearly constant velocity, Equation (5.17) reads:

$$\underbrace{[\mathbf{u} \nabla T]'' \Delta x^2 \pm [\mathbf{u} \nabla T] '' \Delta t^2}_{\text{Eulerian}} \neq 0 ; \quad \underbrace{\mathbf{u}'' \nabla T \Delta x^2 \pm \mathbf{u}'' \nabla T \Delta t^2}_{\text{Lagrangian}} = 0 \quad (5.18)$$

i.e., the Lagrangian integration does not have error because  $\mathbf{u}'' = \mathbf{u}'' = 0$ , but the Eulerian framework presents error due to  $[\mathbf{u} \nabla T]'' \pm [\mathbf{u} \nabla T] '' = \mathbf{u} [\nabla T]'' \pm \mathbf{u} [\nabla T] '' \neq 0$

### Projection errors

In order to obtain a complete formula for the errors obtained for the particular Lagrangian approximation employed by PFEM-2, the projection errors should be taken into account. As it was mentioned along this work, PFEM-2 is a hybrid method using particles and mesh in order to exploit the advantages of Lagrangian and Eulerian methods. This duality between mesh and particles requires that data should be sent back and forth along the whole computation several times. If this projection is not designed properly a huge source of inaccuracies is introduced and the error degrades a lot. Moreover, the projection is normally responsible of getting noisy values represented by the mesh, like forces in a fluid structure interaction problems. The reference work of Idelsohn et. al. [Idelsohn et al., 2015] conclude that these errors depend on the distance between particles which is denoted by  $h$ . Therefore, the importance of using a fine particle mesh with  $h \ll \Delta x$  becomes obvious for decreasing projection errors. However that assumption relies in an ideal projection strategy which consists in generating a mesh with the particle positions. That approach is unaffordable because of the computational cost involved. The analysis of the projection error term employing an affordable projection technique (one of those presented in Section 3.2.2) was experimentally done in that Section, where convergence results have shown second order convergence with the mesh-size. In this Subsection, a mathematical procedure is presented in order to establish theoretically the error when the projection from particles to mesh is done.

Let  $\phi$  be the solution of the minimization problem of the Equation (3.12), with  $\phi \in V_h$  and  $\Phi$  a  $C^2$  function, then

$$\|\phi - \Phi\|^2 := \frac{1}{N_p} \sum_{p=1}^{N_p} \|\phi(\mathbf{x}_p) - \Phi(\mathbf{x}_p)\|^2 \leq \|V - \Phi\|^2, \quad \forall V \in V_h \quad (5.19)$$

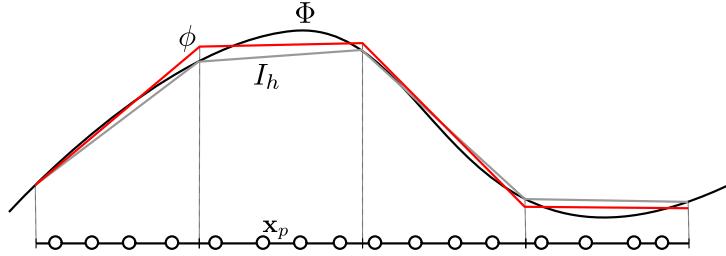


Figure 5.1: Graphical example of the analytical function  $\Phi$  and its two possible FEM approximations  $\phi$  and  $I_h$ .

**Theorem 1.** *Exists a constant  $C$ , independent from the mesh and from the function  $\Phi$  such as*

$$\|\phi - \Phi\| := \left[ \frac{1}{N_p} \sum_{p=1}^{N_p} \|\phi(\mathbf{x}_p) - \Phi(\mathbf{x}_p)\|^2 \right]^{1/2} \leq C \|D^2\Phi\|_\infty \Delta x^2 \quad (5.20)$$

where  $\|D^2\Phi\|_\infty$  means the maximum of the second derivatives of  $\Phi$  and  $\Delta x$  is the mesh size.

*Proof.* Let  $I_h$  be the Lagrange interpolating function of  $\Phi$  at mesh nodes over the Finite Element space  $V_h$ . Knowing that  $\phi$  minimizes  $\|\phi - \Phi\|$  among every FEM function, then  $\|\phi - \Phi\| \leq \|I_h - \Phi\|$ . According to the standard results of interpolation [Burden and Faires, 2005], at each particle it results

$$\|I_h(\mathbf{x}_p) - \Phi(\mathbf{x}_p)\| \leq C \|D^2\Phi\|_\infty \Delta x^2$$

therefore

$$\left[ \frac{1}{N_p} \sum_{p=1}^{N_p} \|\phi(\mathbf{x}_p) - \Phi(\mathbf{x}_p)\|^2 \right]^{1/2} \leq \left[ \frac{1}{N_p} \sum_{p=1}^{N_p} \|I_h(\mathbf{x}_p) - \Phi(\mathbf{x}_p)\|^2 \right]^{1/2} \leq C \|D^2\Phi\|_\infty \Delta x^2$$

□

The theorem 1 allows to limit the error measured over the particles. Next, is necessary to limit the error in  $L^2$  norm over the domain, then the error over the mesh nodes must be limited. For completeness, but without demonstration, the Theorem 2 and its corollary are presented<sup>1</sup>.

<sup>1</sup>At the moment of the publication of this thesis, proofs are next to be published by Morin et.al.

**Theorem 2.** Let  $\mathbf{x}_j$  be the position of the node  $j$ ,  $J$  the number of nodes, and considering next hypothesis:

- One dimension: if there are at least two particles by element separated more than  $c\Delta x$ .
- Two (three) dimensions: if there are at least three (four) particles by element which conform a non-degenerated triangle (tetrahedron) of diameter greater than  $c\Delta x$ .
- The number of particles inside each element is limited.

Therefore

$$\|\phi(\mathbf{x}_j) - \Phi(\mathbf{x}_j)\| \leq C \|D^2\Phi\|_\infty \Delta x^2 \quad \forall j \in J$$

where  $C$  depends on the quality of the triangles (tetrahedron) that conform the particles inside each element.

**Corollary 1.** Under the hypothesis of the previous theorem,

$$\|\phi - \Phi\|_{L^2} \leq C \|D^2\Phi\|_\infty \Delta x^2$$

with  $C$  as in the previous theorem.

The projection error is introduced at each projection step independently of the time-step size. Then, in a total period of time  $(t_f - t_0)$  the Lagrangian error can be extended to

$$\epsilon^L = \pm \mathcal{O}\langle (Q'' \pm \mathbf{u}'' \nabla T) \Delta x^2 \rangle (t_f - t_0) \pm \mathcal{O}\langle (Q'' \pm \mathbf{u}'' \nabla T) \Delta t^2 \rangle (t_f - t_0) \pm \mathcal{O}\langle T'' \Delta x^2 \rangle \frac{(t_f - t_0)}{\Delta t} \quad (5.21)$$

### 5.1.3 Validation test

The differences presented in Equation (5.17) should be analyzed in depth. In the Eulerian case the error formula includes the spatial and temporal derivative of the product between the velocity and the scalar gradient, while as in the Lagrangian formula they only affect the velocity. It has important theoretical consequences, which can be enumerated:

1. Eulerian frames are better for diffusion dominant problems. In these cases, the errors between the Lagrangian and Eulerian approaches are of the same order, but in the Lagrangian frames the projection errors must be added.
2. Lagrangian frames are better for convective dominant problems when the convective flow is constant or nearly constant  $\mathbf{u}'' \approx \mathbf{u}'' \approx 0$ . The remaining cases, when the convective flow presents high variations the Lagrangian or the Eulerian frame will be better or worse depending on the projection errors.

3. Eulerian frames are better for stationary problems. In these cases,  $[\nabla T]'' = 0$  and they do not have projection errors.

The aim of this section is verifying these consequences with numerical simulations employing both frames. The modification of some parameters in these tests will increase or decrease the terms which introduce errors in each formula, being the objective matching practical and theoretical results. Therefore, a particular 2D problem was tailored with an analytical solution to compare the results.

The proposed domain is  $[-1, -1] \times [1, 1]$ . The velocity field  $\mathbf{u} = u\hat{\mathbf{i}} + v\hat{\mathbf{j}}$  has variation both in time and space: a rigid rotation with a periodic temporal variation of its angular velocity which is modulated in space by a parabola. The proposed velocity field is:

$$\begin{cases} u(x, y, t) &= -\omega_1 y(1 - x^2)(1 - y^2)(1 + C \sin(\omega_2 t)) \\ v(x, y, t) &= \omega_1 x(1 - x^2)(1 - y^2)(1 + C \sin(\omega_2 t)) \end{cases} \quad (5.22)$$

where  $\omega_1$  is the mean angular velocity of the rotation field,  $\omega_2$  is the frequency at which the field increases/decreases its rotation. The parameter  $C$  allows to control the amplitude of the variation of the rotation direction, such as if  $|C| > 1$  the rotation is inverted for certain time, if  $C = 1$  the movement vanishes at some instant and if  $|C| < 1$  the field does not change its rotation direction.

A source term  $q$  was included such that the unknown function becomes:

$$T(x, y, t) = \sin(\omega_3 t) \sin(\pi x) \sin(\pi y) \quad (5.23)$$

which is identically zero at boundaries and has four hills at points  $[\pm 0.5, \pm 0.5]$ . An initial value of  $T = 0$  and Dirichlet boundary conditions  $T = 0$  were used. The diffusivity  $k$  and the oscillation frequency  $\omega_3$  can take independent values allowing to analyze different situations.

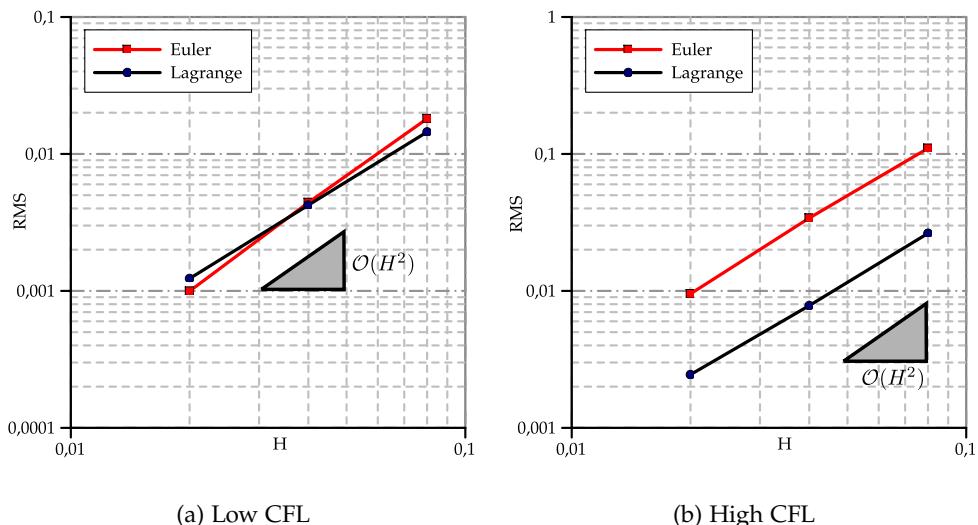
The reference grid used has  $50 \times 50$  nodes conforming 4802 triangles, which gives a mesh-size  $\Delta x = H = 0.04$ . In the Eulerian case, Crank-Nicholson as time discretization scheme was employed. For Lagrangian simulations four equispaced particles per element were initially seeded. The particle grid size  $h$  is defined as  $h = \sqrt{2A/N_p}$ , where  $A$  is the area of the container element and  $N_p$  is its number of particles. With this definition, the area of the element is divided by the number of particles contained.

Four cases were designed to cover a wide range of problems types. The table 5.1 presents the parameters employed in each case. Two time-steps are employed in order to simulate with small  $\text{CFL}_{max} \approx 1$  and large  $\text{CFL}_{max} \approx 10$  respectively, being  $\text{CFL} = |\mathbf{u}| \Delta t / \Delta x$  the Courant-Friedrich-Levy number.

Case	$C$	$\omega_1$	$\omega_2$	$\omega_3$	$k$
1	1	$2\pi$	0	$2\pi$	0
2	1	$2\pi$	$2\pi$	$2\pi$	0
3	1	$2\pi$	$2\pi$	$3\pi$	0
4	$1/2$	$2\pi$	$2\pi$	$2\pi$	0

Table 5.1: Parameters for the manufactured 2d scalar transport case

Results are presented in Figure 5.3. In the Case 1, which has a steady velocity field ( $\mathbf{u}'' = 0$ ) and no diffusion, as anticipated in previous section, the Lagrangian framework have better results than the Eulerian one, being specially remarkable when large Courant numbers are employed. In the mentioned case, the PFEM-2 error does not depend on the time-step size showing a periodic variation due the spatial error which is proportional to the temporal harmonic function  $\nabla T$ . On the other hand, FEM shows large errors when the time-step is increased.

Figure 5.2: Convergence of RMS Error. Case A.  $CFL_{max} = 0.6$  in 5.2a and  $CFL_{max} = 6$  in 5.2b.

In Case 2, the velocity field is also unsteady having an harmonic variation in its amplitude. Here  $\mathbf{u}'' \neq 0$ , then the Lagrangian solutions shows similar problems to the Eulerian. However, if the oscillation frequency of the solution is larger than the amplitude variation of the velocity ( $\omega_3 > \omega_2$ ), as happens in Case 3, the temporal error term in the Eulerian case ( $[\mathbf{u} \nabla T]'' \Delta t^2$ ) increases, while the Lagrangian ( $\mathbf{u} [\nabla T]''$ ) remains almost equal. In this mode, Case 2 and

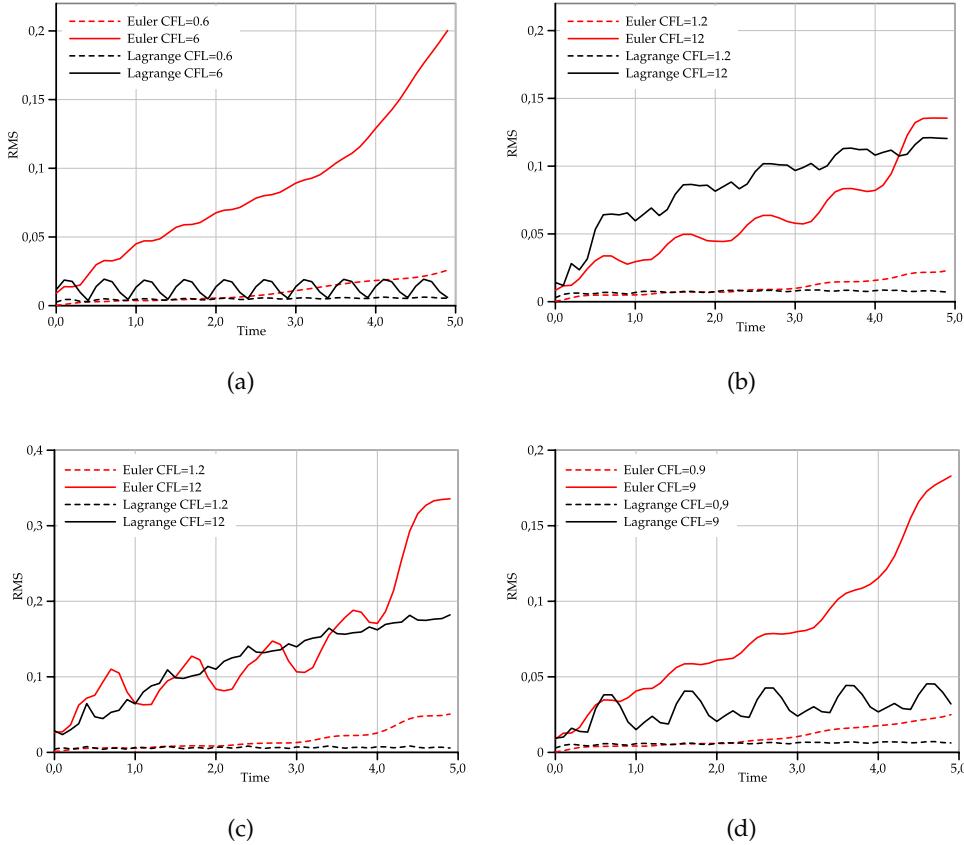


Figure 5.3: Case Four-hills without diffusion. RMS error evolution for Eulerian and Lagrangian simulations with steady (Figure 5.3a) and unsteady (Figures 5.3b, 5.3c and 5.3d)) velocity fields. The Lagrangian error is presented with black lines and the Eulerian with red lines.

Case 3 also prove experimentally the theoretical error formula presented in Equation (5.17).

The first three cases use a velocity field which vanishes every  $T = 1/\omega_2$ , this gives to the Eulerian formulation some advantages because the spatial error term ( $[\mathbf{u} \nabla T]'' \Delta x^2$ ) also vanishes. That advantage is, in some cases, recovered by the Lagrangian when the time-step is increased. The Case 4 overcomes this fact using another value for the constant C which avoids that  $\mathbf{u}(t) = 0$  for any  $t$ . Figure 5.3d shows that the errors differences now are more visible, proving again the theoretical error formula.

As a preliminar conclusion, the results presented in Figure 5.3 confirms the proposed error formula, showing that the Lagrangian error due to use large time-steps is affected by the unsteadiness of both the velocity field and the source. As a footnote based on experimental facts, it can be mentioned that a X-IVAS calculation of the unsteady source is mandatory

to obtain accurate results, which is not possible with the time-step selected (it only samples seven points of the source's period). Regarding to Eulerian simulations, they show larger errors when the CFL grows. Also, this error increases when high  $\nabla T^{\prime\prime}$  are employed.

On the other hand, Figure 5.2 presents a convergence analysis for Case A. Three mesh refinement were used  $\Delta x = 2H$ ,  $H$  and  $H/2$ , modifying the  $\Delta t$  in order to keep the  $CFL_{max}$  in a constant value. The RMS error presented in each case is the average value of RMS from  $T = 0$  to  $T = 2[s]$  of simulation time. As presented in Figure 5.3a, using  $CFL_{max} \approx 0.6$  the Lagrangian and Eulerian errors are almost the same. However, when the adimensional number is increased to  $CFL_{max} \approx 6$ , the Lagrangian solution obtains better accuracy than Eulerian even using both time-step and mesh-size twice larger. This fact represents a notable advantage of the Lagrangian approach over the Eulerian in this type of problems, showing that PFEM-2 simulation can obtain the same precision than FEM even solving a problem eight times smaller.

Regarding to convergence orders, both Lagrangian and Eulerian solutions show concordance with theoretical formula, presenting a second order convergence in every case. It is emphasized that in the Lagrangian approach this behavior is reached if the projection operator fulfill this requirement.

## 5.2 An extension to viscous incompressible fluid flow

In this case the model to be solved is represented by the Navier-Stokes equations added with the mass conservation that imposes a constraint on the velocity field to be divergence free. This condition is enforced by the pressure acting as a Lagrange multiplier. Inspecting the momentum equation and comparing this with the scalar transport equation just analyzed, here the same unknown variable acts as the velocity field that drives the convection term. This fact produces a non-linear term that is the responsible for the chaotic nature of the model that pretend to emulate the physical effects produced by the turbulence.

### 5.2.1 General comments

The error equation looks similar like the presented in the case of scalar transport, but the conclusions are fundamentally different. Generalizing Equation (5.17), the difference between

the errors in the Eulerian and Lagrangian frames are:

$$\underbrace{[\mathbf{u} \nabla \mathbf{u}]'' \Delta x^2 \pm [\mathbf{u} \nabla \mathbf{u}] \cdot \Delta t^2}_{Eulerian} \neq \underbrace{\mathbf{u}'' \nabla \mathbf{u} \Delta x^2 \pm \mathbf{u} \cdot \nabla \mathbf{u} \Delta t^2}_{Lagrangian} \quad (5.24)$$

Adding the projection errors for a period  $(t_f - t_0)$  reads:

$$\underbrace{([\mathbf{u} \nabla \mathbf{u}]'' \Delta x^2 \pm [\mathbf{u} \nabla \mathbf{u}] \cdot \Delta t^2) (t_f - t_0)}_{Eulerian} \neq \underbrace{(\mathbf{u}'' \nabla \mathbf{u} \Delta x^2 \pm \mathbf{u} \cdot \nabla \mathbf{u} \Delta t^2) (t_f - t_0) \pm \mathcal{O}(\mathbf{u}'' \Delta x^2) \frac{(t_f - t_0)}{\Delta t}}_{Lagrangian} \quad (5.25)$$

In this case, it is not anymore possible to separate the case in which the convective field is constant from the case in which the gradient of the transported field ( $\nabla \mathbf{u}$ ) is not constant. Nevertheless, some particular cases may be analyzed in order to draw some conclusions:

- Eulerian frames are better for low Reynolds number. In these cases, the errors between the Lagrangian and the Eulerian approaches are of the same order, but on Lagrangian frames the projection errors must be added.
- Lagrangian frames are better for convective dominant problems when the velocity has a smooth variation in time but the gradient of the velocity has high spatial variations. This case is very common in fluid mechanics problems, such as in shock waves. The remaining cases are better or worse depending on the projection errors.
- Lagrangian frames are better for multi-fluid flows. This is because Eulerian frames need to solve a level set equation to know the position of the interface. The level set equation [Osher and Fedkiw, 2001] is a convection equation that requires small time-steps to yield accurate results due to the considerations concluded in the previous section, i.e.:

$$\mathbf{u} \cdot \nabla \mathbf{u} < 1 \quad \text{and} \quad [\mathbf{u} \nabla \mathbf{u}] \cdot \nabla \mathbf{u} \approx \mathbf{u} [\nabla \mathbf{u}] \cdot \nabla \mathbf{u} \gg 1$$

### 5.2.2 Validation test

Following the same criteria as the validation test in Section 5.1.3, a manufactured case is employed where a solution is proposed and the external force  $\mathbf{f}$  must be adapted in order to satisfy the differential equation.

The case is adapted from the book of Donea&Huerta [Donea and Huerta, 2003] where the authors solved a stationary Stokes flow. In the current thesis, the problem includes also the

convective term, leading to recalculate the external force needed to satisfy the Navier Stokes equations. A two-dimensional problem in the square domain  $[0, 0] \times [1, 1]$  is considered, which possesses a closed-form analytical solution. The problem consists in determining the velocity field  $\mathbf{u}$  and the pressure  $p$  such that

$$\left\{ \begin{array}{ll} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot (\nu \nabla \mathbf{u}) + \nabla p = \mathbf{b} & \text{in } \Omega \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \\ \mathbf{u} = 0 & \text{in } \Gamma \end{array} \right. \quad (5.26)$$

where the fluid viscosity  $\nu$  can be changed in order to simulate with different Reynolds numbers, being  $Re = |\mathbf{u}|L/\nu$  with  $L = 1$ . In order to solve the previous equation system, an analytical solution is proposed:

$$\left\{ \begin{array}{l} u(x, y, t) = x^2(1-x)^2(2y-6y^2+4y^3)(1+0.5 \sin(\omega t)) \\ v(x, y, t) = -y^2(1-y)^2(2x-6x^2+4x^3)(1+0.5 \sin(\omega t)) \\ p(x, y, t) = x(1-x) \end{array} \right. \quad (5.27)$$

which allows to find the expression for  $\mathbf{b}$ . Varying  $\nu$  and  $\omega$  it is possible to modify the Reynolds number and the unsteadiness of the solution respectively. Cases were solved using a  $50 \times 50$  Cartesian mesh split into triangles and setting the time-step such as  $CFL_{max} \approx 10$ . Table 5.2 shows the configuration and the RMS errors at  $T = 1000[s]$  of the simulated cases.

Case	$\omega$	$Re$	Eulerian			Lagrangian		
			RMS $\mathbf{u}_x$	RMS $\mathbf{u}_y$	RMS $p$	RMS $\mathbf{u}_x$	RMS $\mathbf{u}_y$	RMS $p$
1	$\pi/5000$	10	0.75	0.73	0.036	2	1.9	0.058
2	$\pi/5000$	1000	0.012	0.011	0.014	0.0071	0.0062	0.013
3	$\pi/50$	10	0.8	0.82	0.026	0.83	0.85	0.023
4	$\pi/50$	1000	$\gg 1$	$\gg 1$	$\gg 1$	1.3	1.5	0.06

Table 5.2: Configuration and RMS  $\times 10^3$  errors. Every case was run with  $CFL_{max} \approx 10$ .

Figure 5.4 presents graphically the evolution of the RMS for Cases 1 and 2. In the first case, a low Reynolds numbers was selected, therefore the Eulerian solution obtains better results although simulating with large CFL. This is an equivalent problem to that diffusion dominated scalar transport problem. However, when the Reynolds number is increased, the Lagrangian framework recovers its advantage. In cases 3 and 4, the unsteadiness of the solution is increased, leading to an increment of the temporal derivatives. In the Lagrangian

case the error enlargement is lower than the Eulerian case, mainly because  $\mathbf{u} \cdot \nabla \mathbf{u} < [\mathbf{u} \nabla \mathbf{u}]$ . This fact is of extreme importance in Case 4 where the Eulerian solution diverges, concluding that beyond the increasing of the Reynolds number, the grow of the unsteadiness also affects Eulerian simulations. On the other hand, Lagrangian framework is not the adequate in the case of Low Reynolds number, which is also consistent with the theoretical formula.

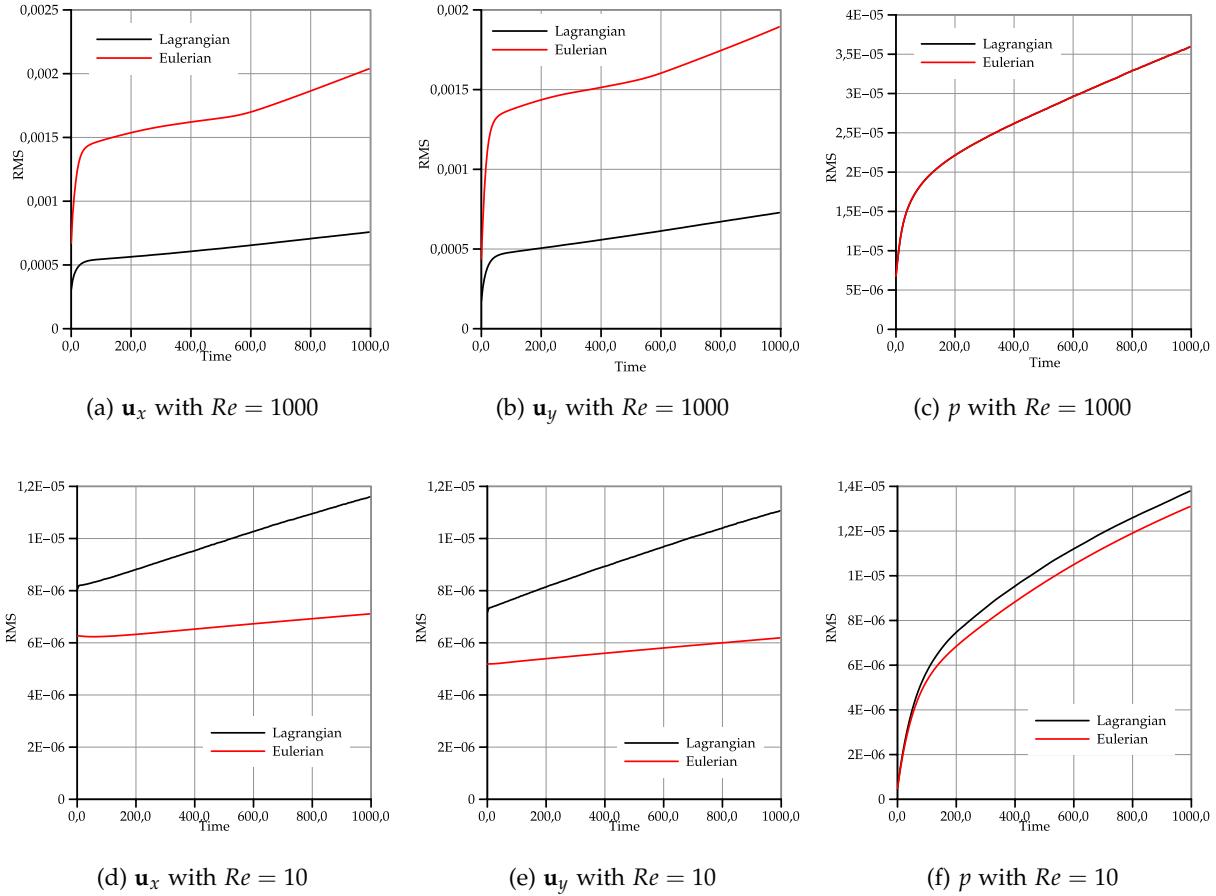


Figure 5.4: RMS measured for a Navier-Stokes analytical solution using Lagrangian and Eulerian schemes. Cases with  $\omega = \pi/5000$ .

The analysis presented allows to design better conditions to improve the accuracy of the results obtained with the PFEM-2 method that combined with the enhanced efficiency shown in previous works complete the target of introducing a new method for solving the more challenges problems of the next decades. As a side-effect it has been shown that in problems governed by high speeds, as those occurring in fluid mechanics, this approximation not only allows to enlarge the time-steps but also allow to get good solutions with coarser meshes when compared with pure Eulerian codes with a significant saving of computational costs.

# Chapter 6

## Industrial application: residence time distributions

### 6.1 Background

In the oil & gas industry the water-oil separators are essential part of the superficial installations downstream of the wells. Particularly, in secondary oil recovery the extraction requires large amount of water that need to be reinjected into the wells in order to attend environmental issues. This produced water is reinjected to help to extract crude by drag and buoyancy forces. The produced water must be extremely cleaned of remaining oil in order to avoid of possible obstructions leaving the well out of service. This cleaning process is carried out in part with a buoyancy equipment called skimmer, which basically consists of a big tank with internals where the water-oil mixture enters and the crude is separated by buoyancy and collected on the free-surface while the water is removed from the bottom.

A parameter characterizing the mixing and flow within tanks and to compare the behavior of real skimmers to their ideal models is the residence time distribution (RTD). It is a probability distribution function that describes the time a fluid element could remains inside the tank, for its purification. The RTD is useful, not only for troubleshooting in existing skimmers allowing an optimization of them, also in estimating the yield of a given separation and improving the future tank designs. The standard function  $E(t)$ , normally employed for RTD, has the units of  $\text{time}^{-1}$  and is defined such that

$$\int_0^{\infty} E(t) dt = 1 \quad (6.1)$$

In absence of dispersion and for a constant volumetric flow ( $Q = Q_0$ ) the theoretical

residence time  $\tau$  is defined as:

$$\tau = \frac{V}{Q'} \quad (6.2)$$

with  $V$  the tank volume.  $\tau$  must be equal to the mean residence time,  $t_m$ . As is the case with other variables described by distribution functions, the mean value of the variable is equal to the first moment of the RTD function,  $E(t)$ . Thus, the mean residence time is

$$t_m = \int_0^\infty t E(t) dt \quad (6.3)$$

Tracer injection from the inlet is the strategy which allows to experimentally or numerically measure the RTD function. The kind of material and amount of the tracer is selected such as neither modify the fluid physical properties nor its hydrodynamic conditions. There are two mainly used options for the tracer inlet distribution: a pulse or a step. In the first case a small volume of tracer, which approximates to the Dirac delta function, is introduced. The injection time  $t_{inj}$  must be much smaller than the theoretical residence time, i.e.  $t_{inj} \ll \tau$ . Then, considering  $C(t)$  the tracer concentration at the outlet at any time  $t$ , the RTD function can be obtained as:

$$E(t) = \frac{C(t)}{\int_0^\infty C(t) dt} \quad (6.4)$$

In an experiment with a step function the tracer concentration at the inlet changes suddenly at a given time  $t$  from 0 to  $C_0$ . The tracer concentration is measured at the outlet and normalized in order to obtain the cumulative RTD function  $F(t)$  with values between 0 and 1:

$$F(t) = \frac{C(t)}{C_0} \quad (6.5)$$

The responses to the pulse and to the step are related as:

$$\begin{cases} F(t) = \int_0^t E(t) dt \\ E(t) = \frac{dF(t)}{dt} \end{cases} \quad (6.6)$$

The efficiency of a skimmer tank is measured as the capacity of separating water-oil. Due to the equipment works only by buoyancy, the more time the injected mixture is in the tank, the more separation is obtained. Thus, it is expected that the skimmer works as a plug-flow reactor (PFR) where no dispersion of  $E(t)$  is obtained. However, because of geometrical or economic constrains, real tanks have channeling (bypassing or short circuiting) and dead zones which tend to modify the desired ideal behavior reducing the residence time of some injected fluid portions. Moreover, the presence of flow turbulence leads to obtain non-ideal

behaviors. In this context, numerical experiments are widely used to optimize geometrical designs and operation conditions without the need of large set of experimental tests. Finding  $E(t)$  and/or  $F(t)$  for several tank prototypes using numerical experiments requires very long simulations (for standard skimmers several hours of real time). As aforementioned, PFEM-2 is able to employ large time-steps being useful to solve in a faster way this type of problems without stability compromise. In addition, due to the flow behavior inside a skimmer tank has regions dominated by inertia and a proper tracer transport mainly depends on the convective term, it has demonstrated that the method will introduce less numerical error than other Eulerian alternatives.

## 6.2 Turbulent dispersion

The concentration  $C$  of the tracer follows an advective-diffusive equation, i.e.

$$\frac{DC}{Dt} = \nabla \cdot (\alpha_T \nabla C) \quad (6.7)$$

Most of real skimmers present turbulent flow regime, i.e. the experimental RTD can be described by the plug flow with axial dispersion model. In this context, the simulated dispersion coefficient  $\alpha_T$  can be described as the sum of three contributions, two of them are physical: molecular ( $\alpha_m$ ) and turbulent dispersion ( $\alpha_t$ ), and the third term is the numerical diffusion ( $\alpha_n$ ):

$$\alpha_T = \alpha_m + \alpha_t + \alpha_n \quad (6.8)$$

The tracer employed must guarantee negligible molecular diffusion. In general it is known that molecular diffusion has little effect on ensemble mean concentrations in high Reynolds number flows [Thomson, 1988]. However, experiments show that the turbulent contribution  $\alpha_t$  is not negligible for the skimmer flow regimes being analyzed, requiring the numerical computation of the diffusion to accurately predict the behavior of the tracer. Regarding to the third term, as presented in Chapter 5, Lagrangian integration does not introduce excessive numerical diffusion, while Eulerian strategies introduce more or less diffusion according to the discretization strategy selected and the mesh employed.

With the PFEM-2 method, the advection (i.e. left hand side of Eq. 6.7) is automatically done through the movement of the particles. Each particle carries its own character or concentration value  $C_p$ , marking with  $C_p = 1$  to those that belong to the tracer, leaving the others with  $C_p = 0$ . The diffusive part of the equation can be solved in PFEM-2 either explicitly or

implicitly. The former, which computes the diffusion over particles during its streamline integration (X-IVAS method [Idelsohn et al., 2012]), is not able to be employed with time-steps larger than  $\alpha_T \Delta t / \Delta x^2 > 1$ . On the other hand, the latter is unconditionally stable and consists on projecting the concentration field from particles to nodes, solving a Laplacian equation on the mesh, and interpolating the updated values again over the particles. When only the tracer is solved, this strategy adds three extra stages per time-step, increasing the computational work mainly due to the solution of the equation system for diffusion. It can be noted that both approaches loss the particle character, allowing it to take values  $0 \leq C_p \leq 1$ .

In this thesis, an alternative approach, which can be solved explicitly over particles without any stability issue and without modifying the particle character, is presented. It is based on the traditional idea of modeling the motion of single particles in a turbulent flow assuming that the turbulent motions of the particles are similar to the Brownian motions of particles or molecules, i.e. a stochastic strategy. Therefore, the movement of the particles due to the turbulent part of the flow field is modeled with a random walk scheme. The tracer position  $\mathbf{x}(t)$  in a continuous random-walk model (CRW) can be described by the non-linear Langevin equation:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}(\mathbf{x}, t) + \bar{\bar{B}}(\mathbf{x}, t)\xi(t) \quad (6.9)$$

where the vector  $\mathbf{A}(\mathbf{x}, t)$  represents the variation of  $\mathbf{x}(t)$  due to the mean velocity field (i.e. deterministic part), while the second term is a stochastic term which depends on a tensor  $\bar{\bar{B}}(\mathbf{x}, t)$  that characterizes the random motion and a vector  $\xi(t)$  with random numbers normalized between 0 and 1. Following the assumptions presented by Fabbroni [Fabbroni, 2009] and considering isotropic turbulent diffusivity, the Eq. 6.9 can be applied to each particle leading to:

$$\Delta\mathbf{x}_p = \mathbf{u}_j(\mathbf{x}_p)\Delta t + \sqrt{2\alpha_t\Delta t} \mathbf{w} \quad (6.10)$$

where  $\mathbf{w}$  is a random Gaussian variable with zero mean and standard deviation of 1. An additional term that allows to incorporate physical anisotropy to the model was introduced in recent works [Hryb et al., 2009]. The key idea is to move the particles from high turbulent zones to lower ones in order to represent the diffusion in a more realistic way.

$$\Delta\mathbf{x}_p = \mathbf{u}_j(\mathbf{x}_p)\Delta t - \nabla\alpha_t(\mathbf{x}_p)\Delta t + \sqrt{2\alpha_t\Delta t} \mathbf{w} \quad (6.11)$$

There are other alternatives in order to model the turbulence over particles, being the most used the called Discrete Random Walk (DRW). This model, which is also known as Eddy Interaction Model (EIM) [Graham and James, 1996], is based on reconstructing the

instantaneous field from the local mean values of velocity and turbulent intensity. The discrete time intervals between velocity updates are dictated by the *eddy lifetime* which varies spatially according to the turbulent fields. Due to the explicit and parallel integration employed by PFEM-2, this approach is far from being efficient when time-steps larger than eddy lifetime are used. Finally, in this Chapter the CRW, under the approach assumed by Eq. 6.11, is selected for modeling turbulent dispersion.

### 6.3 Contact tank

Although the purpose is rather different, the optimization of chlorine contact tanks follows similar design strategies as the presented at the beginning of this Chapter for the case of skimmer tanks. The target of a chlorine contact tank, based on the volume displacement criterion, is to achieve enough detention time to successfully conclude the chlorination process. This criterion relies upon the assumption that plug-flow conditions are hold along the tank. However, the existence and arrangement of baffles in the tank, and the inlet and outlet configurations can result in a much more complex flow pattern, where turbulent mixing, dead-zones and short-circuiting exist. Therefore, a RTD analysis is usually employed to evaluate and optimize the design.

Following the experimental work of Shiono [Shiono and Teixeira, 2000], an example is presented to show the performance of the CRW compared with the calculation of turbulent diffusion over the mesh. The case consists on a 2D model of a contact tank with 7 baffles and 8 compartments, as presented by Fig. 6.1. The discharge entering the tank is 1.17[L/s], resulting in a mean cross-sectional velocity of 0.0104[m/s]. This flow configuration leads to a  $\tau \approx 750[\text{s}]$ . The geometry is discretized with an unstructured mesh composed by triangles. Elements have an almost uniform size of  $\Delta x \approx 0.01[\text{m}]$ .

In order to obtain the time-averaged flow quantities, a first simulation is done. Turbulence is modeled employing the  $k - \epsilon$  model. This flow, shown in Fig. 6.1b, is in good agreement with the published by other works [Gaultieri, 2006, Wilson and Venayagamoorthy, 2010]. The turbulent kinematic viscosity  $\nu_t$  can be obtained as  $\nu_t = C_\mu k^2/\epsilon$ , being  $C_\mu = 0.09$  a constant parameter. Then, the turbulent dissipation is found as  $\alpha_t = \nu_t/\sigma_t$ , where  $\sigma_t$  is the named turbulent Schmidt number. The value of  $\sigma_t$  is adjusted from experiments. Previous works have verified that a turbulent Schmidt number of 0.44 would give a good agreement between experimental and simulated RTD.

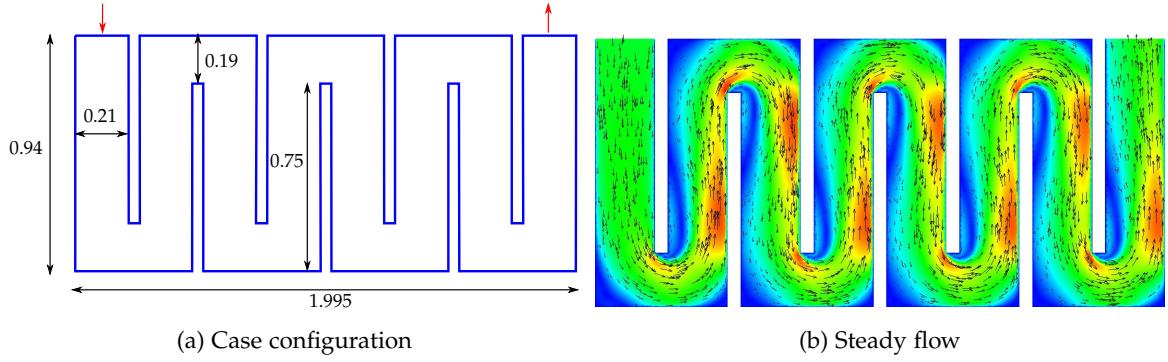


Figure 6.1: Contact tank case. Geometrical configuration and steady-state incompressible flow field calculated with  $k - \epsilon$ . Geometry dimensions are expressed in meters. Magnitude of velocity is scaled from 0[m/s] (blue) to 0.22[m/s] (red).

Next step is computing the tracer transport. In order to solve Eq. (6.7) the steady state velocity and turbulent fields are employed. The simulation is carried on using the two numerical strategies available, which vary according to the turbulent dispersion approach employed, i.e. an *implicit* calculation isotropic over the mesh, or a stochastic simulation with CRW. Table 6.1 presents tabulated snapshots for the tracer evolution for both strategies also comparing with a pure advective solution.

Without turbulent diffusion treatment the solutions obtained are nonphysical leading to unrealistic RTDs because  $\lim_{t \rightarrow \infty} F(t) \neq 1$ . On the other hand, the inclusion of dispersion modeling let to the entire volume will be filled by tracer. However the transient evolution from the initial state (without tracer) to the final state is simulated differently with implicit or CRW strategies. The random-walk solution shows greater axial diffusion while the implicit one presents larger diffusion at front of tracer step.

Fig. 6.2 compares among the cumulative RTDs calculated with PFEM-2 and a numerical reference calculated with the CFD software OpenFOAM®2.4.0 which implements the Finite Volume Method (FVM). Finer mesh ( $\Delta x = 0.005[\text{m}]$ ) and high order methods for convection and other interpolations were employed in order to obtain a reference solution. The reference cumulative RTD obtained is in agreement with other numerical results reported in literature [Wilson and Venayagamoorthy, 2010]. Comparing the reference with PFEM-2 simulations, a better agreement with the implicit option can be seen. The stochastic approach tends to transport a lower dissipated step leading to a cumulative RTD too sharp.

Regarding to computing times, if only the transport equation for the tracer is solved, the random-walk approach is cheaper. The need of projection, solution of an equation system and

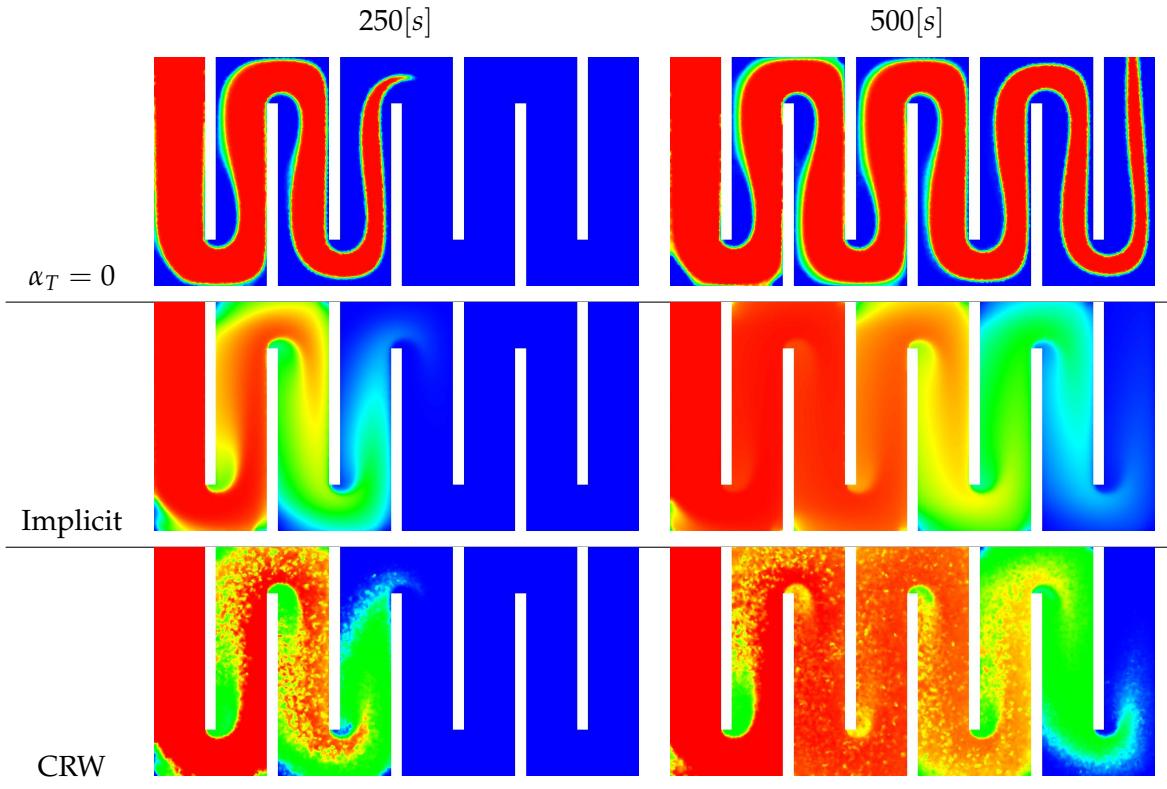


Table 6.1: Screen-shots for tracer distribution at  $t = 250[\text{s}]$  (left) and  $t = 500[\text{s}]$  (right) simulated with several PFEM-2 variants. Scale goes from 0 (blue) to 1 (red).

correction over particles turns the implicit strategy approximately 40% more expensive than the stochastic strategy. Although this 2D case was solved considering an steady state flow, most of tracer transports must be simulated together with the unsteady flow. In such cases, PFEM-2 already requires the three mentioned steps for the velocity and pressure calculation, therefore the inclusion of the tracer treatment does not add significant extra work. Attending to the above mentioned, the implicit treatment of the turbulent dispersion is chosen as a modeling strategy for the next example.

## 6.4 Skimmer tank

The skimmer selected for the numerical experiments is a cylindrical tank with aspect ratio between the height  $H$  and diameter  $D$  of  $H/D = 1.5$ . It has three plates. Two of them are hold by the central column which is also used for inlet and outlet, and the third is supported by the cylindrical tank wall named ring plate. Every one has a three degree inclination. The

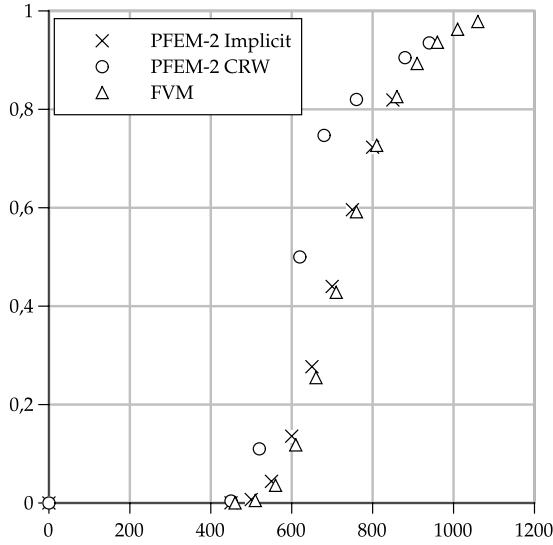


Figure 6.2: Cumulative Residence Time Distribution for contact tank case. Comparison between both options for turbulent dispersion in PFEM-2 (implicit and random-walk) and an Eulerian reference solution.

inlet is through two pipes located below the upper plate that injects the load tangentially to the tank walls in order to produce a rotational movement concentric with the vertical axis of the tank. Table 6.2 presents the main characteristics of the tank and the operative condition. The mesh employed can be observed in Fig. 6.3, which consists of 306,768 nodes conforming 1,683,575 tetrahedral elements.

Due to the tracer injection is done when the flow is established, a start-up simulation is carried out with a hydrostatic initial condition along the tank, filled with water without tracer. Concerning the boundary conditions, a fixed and uniform velocity is imposed at the inlet to satisfy the required flow rate, the pressure is fixed to a reference value ( $p = 0$ ) at the outlet, the free-surface is assumed as slip, and the remaining boundaries are considered as walls.

Due to the turbulent behavior of the flow, the static Smagorinsky Large Eddy Simulation (LES) turbulence model [Smagorinsky, 1963] is employed to capture the effects unresolved by the scale of the mesh. This approach models the turbulent viscosity as

$$\mu_t = C_s^2 \rho \Delta^2 |\mathbf{S}| \quad (6.12)$$

being  $\mathbf{S} = \nabla \mathbf{u} + \nabla \mathbf{u}^T$  and  $C_s = 0.2$  the Smagorinsky constant. More references about the turbulence modeling in PFEM-2 can be found in the Master's thesis of the author [Gimenez, 2014].

The start-up simulation was done employing a  $\Delta t = 2[\text{s}]$  which leads to a  $\text{CFL}_{max} \approx 20$ .

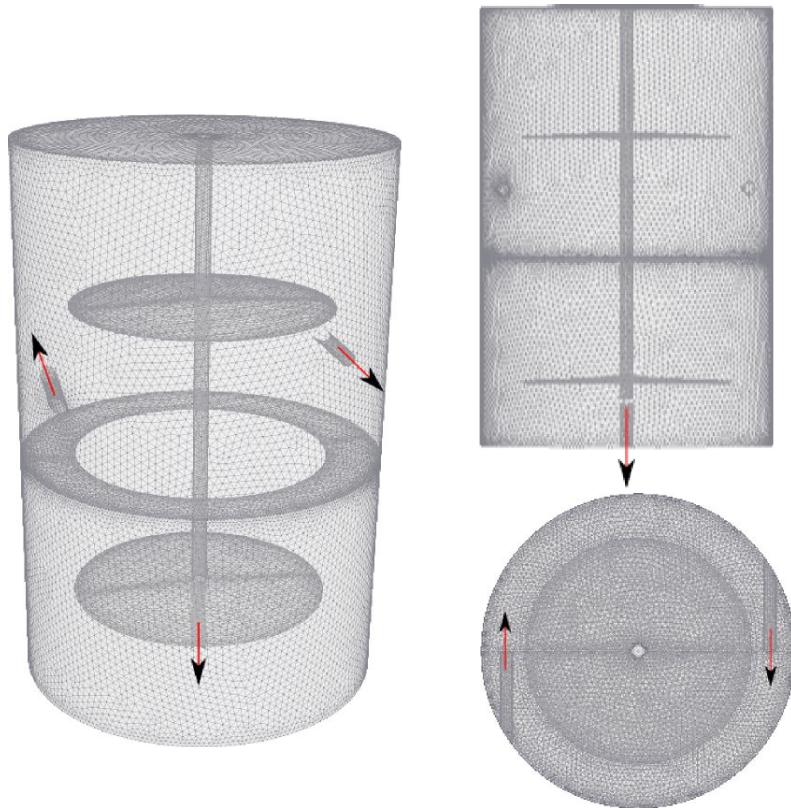


Figure 6.3: Skimmer tetrahedral mesh employed in the numerical experiments.

The final time  $T_f$  is chosen when the integral of the kinetic energy reaches a converged value, as shown by Fig. 6.4a. At this moment no significant changes are observed at the main flow, beyond small fluctuations due to its own unsteadiness. In Fig. 6.4b the magnitude of the velocity field is presented. The tank plates are not shown for clarity but three horizontal and one vertical slices are included to visualize the obtained solution. With a mean inlet velocity about  $0.24[m/s]$ , it is possible to see the flow surrounding the tank going upwards and downwards close to the walls. Using a limited scale adjusted between 0 and 0.01 (Fig. 6.4b), it is also possible to identify a toroidal flow pattern.

Considering the solution presented in Fig. 6.4 as the initial condition, a new simulation is done but now injecting a fully concentrated tracer at the inlet and solving an extra transport equation for it (Eq. (6.7)). The entering particles through the inlet are marked with  $C_p = 1$ , leaving the others with  $C_p = 0$ . Each particle carries its own concentration value  $C_p$ , which is projected to the mesh in order to calculate the turbulent dispersion implicitly. The advective-diffusive tracer equation is solved together with the incompressible flow equations, remarking that the former does not produce any modification into the latter (passive scalar).

Feature	Symbol	Value
Height	$H$	17.89[m]
Diameter	$D$	11.93[m]
Volume	$V$	2000[m <sup>3</sup> ]
Inlet Flow	$Q$	1/9[m <sup>3</sup> s <sup>-1</sup> ]
Theoretical Mean Residence Time	$\tau$	5[hs]
Fluid viscosity	$\mu$	0.001[Kgs <sup>-1</sup> m <sup>-1</sup> ]
Fluid density	$\rho$	1000[Kgm <sup>-3</sup> ]
Tracer diffusivity	$\alpha$	0[m <sup>2</sup> s <sup>-1</sup> ]

Table 6.2: Skimmer tank main features and operative conditions

A qualitative analysis of the numerical experiment with step-type injection is presented in Fig. 6.5a. There, a capture of the tracer concentration employing a scale limited to  $0 < C < 0.25$  after 80 minutes is shown. The tracer is concentrated mostly above the ring plate because the main flow is ascending, but some tracer overflows that plate and gradually fills the lower tank section, which allows to obtain concentration at the outlet. On the other hand, a quantitative analysis is presented in the Fig. 6.5b. The temporal evolution of the concentration measured at the outlet corresponds directly to the function  $F(t)$  (Eq. (6.5)). The same tests were simulated employing different time-steps  $\Delta t = 2, 10$  and  $50$  which leads to  $CFL_{mean} \approx 4$  and  $CFL_{max} \approx 500$  in the last case. The three cases show similar overall solutions.

A relevant result in this type of analysis is the fraction of tracer that has been in the tank for less time than  $\tau$ , i.e.  $F(\tau - \epsilon)$  with  $\epsilon$  small enough. An ideal PFR would obtain  $F_{PFR}(\tau - \epsilon) = 0$ , but considering current simulations of the real tank, the obtained value is  $F_{\Delta t=2}(\tau - \epsilon) \approx 0.598$  (taken from the simulation with smallest time-step). This value should be compared with the obtained by simulating other skimmer designs. This allows to determinate the relative quality of the tank, characterizing the troubleshooting of this particular design. The more relevant problem is the reduction of the effective volume mainly because of the dead zone around the axial axis of the tank where the tracer rarely enters. Simulations with larger time-step predicts similar values for  $F(\tau - \epsilon)$ , being the solution for  $\Delta t = 10$  very similar to the reference,  $F_{\Delta t=10}(\tau - \epsilon) \approx 0.6$ , but slightly different from the solution employing the largest time-step ( $F_{\Delta t=50}(\tau - \epsilon) \approx 0.608$ ).

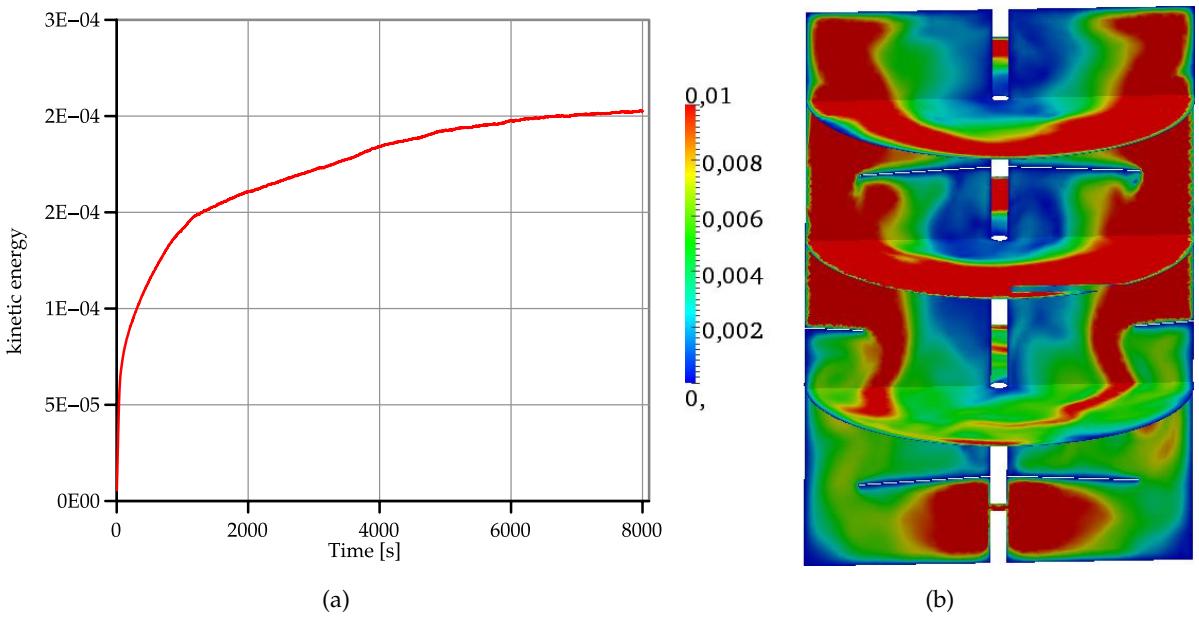


Figure 6.4: Velocity magnitude of the final solution in the start-up simulation sampled at slices and integral of the kinetic energy convergence.

The first arrival time  $t_0$  is defined as

$$t_0 = \min(t) \quad \forall t \mid F(t) > 0.01, \quad (6.13)$$

which gives information of the shortest path between the inlet and the outlet (i.e. short-circuiting) in the tank. In this case, solutions with  $\Delta t = 2[s]$  and  $10[s]$  presents similar results ( $t_0 \approx 38$  minutes), while the simulation with largest time-step underestimate the arrival time at 25 minutes.

Regarding to computational times, as mentioned above, the benefits of employing large time-steps are very appreciated for solving this type of problems. In this particular case, and considering at least 500 minutes of simulation time required, if the numerical scheme is restricted to  $CFL_{max} < 1$  more than  $3e6$  iterations are needed. A moderate  $\Delta t = 2[s]$  which is able to be used with other implicit Eulerian alternatives, was selected as a PFEM-2 reference solution demanding one order less of iterations, although spending more than two days of cpu time running in twelve processes in parallel. Employing a larger time-step  $\Delta t = 10[s]$ , PFEM-2 achieves to a similar solution, but spending only 12 hours.

As conclusion, this section presented an application of the PFEM-2 method to solve problems of industrial interest. The error formulas, presented in Chapter 5, for both scalar transport and incompressible flow, guarantee less error in PFEM-2 than employing

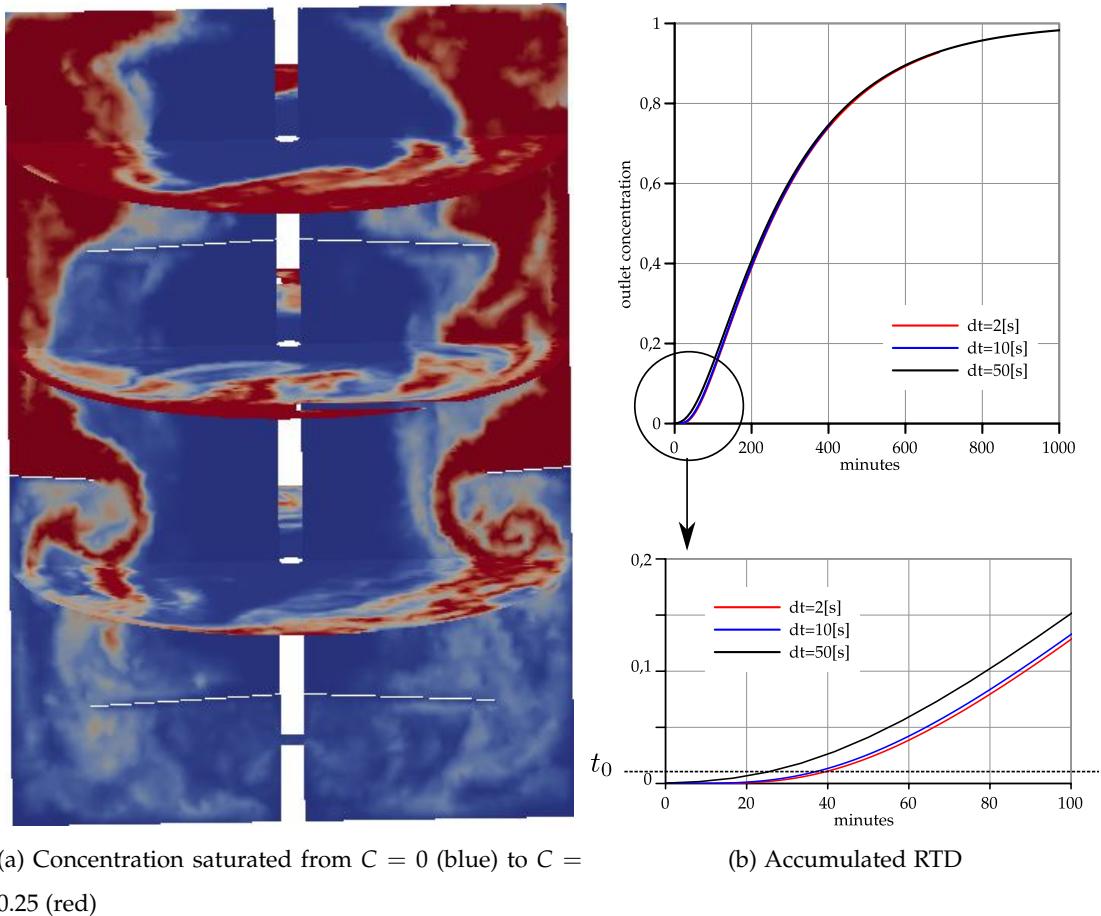


Figure 6.5: Left: snapshot with the concentration field at  $T = 80$  minutes. Right: Cummulative RTD using different  $\Delta t$ .

classical Eulerian strategies when large time-steps and coarse meshes are used to solve convective-dominant problems. Also, this type of problems where the Reynolds number is not large enough to require a finer mesh but the process is very long in time, makes this methodology very useful. Remembering that it is very straightforward to parallelize the spatial coordinates, but it is no so obvious how to parallelize the time scale. Finally, these results confirm that PFEM-2 is not only an interesting research subject but also a valuable tool for design engineers at the moment of making decisions.

## **Part II**

### **Two-phase flows**



# Chapter 7

## Fluid motion equations

The simultaneous presence of multiple fluids with different properties in external or internal flows is found in daily life, environmental problems, and numerous industrial processes, among many other practical situations [De Mier Torrecilla, 2009]. Examples are fluid-fuel interaction in enhanced oil recovery, blending of polymers, bubble column reactors, and the liquid atomization process which has application in fields such as aerospace propulsion systems, automotive engines, food processing and ink-jet printing, in order to name only a few.

Multiphase flows in general involve a combination of phases: gas, liquid and solid, however in most cases as in this thesis, only a mixture of two phases is considered. The complex nature of two or multiphase flow originates from the existence of multiple, deformable and moving interfaces, together with significant discontinuities of fluid properties and intricated flow field near the interface.

A classification based on the interface structures and the topographical distribution of each phase is done by Ishii & Hibiki [Ishii and Hibiki, 2011], who classify two-phase flows in dispersed, separated and mixed flows:

- Disperse flows consist of a disperse phase (finite particles, drops or bubbles) distributed in a connected volume of the continuous phase. As examples can be cited water droplets in air, air bubbles in water or sand grains in water.
- Separated flows consist of two continuous immiscible fluids separated by interfaces. Common examples are a free-surface (water and air) and water and oil mixture at rest.
- Mixed flows has a simultaneous presence of a free-surface and a disperse phase. Examples can be extracted from the nuclear and chemical industry as the case of

bubbly/droplet annular flow [Ishii and Hibiki, 2011] or the interaction of bubble plume with a free surface, which is frequent in oceanography [Márquez Damián, 2013].

The interface between the two phases takes more relevance in the case of separated flows, while in dispersed flows the interface position is difficult to track and usually not relevant, being needed only some averaged or homogenized quantity fluid flow analysis. This thesis is focusing on two-phase separated flows, and therefore, on the interfaces between fluids. For interested readers , reviews of numerical strategies to solve dispersed and/or mixed flows can be found in [Márquez Damián and Nigro, 2014; Márquez Damián, 2013].

In the case of two-phase flows,  $\Omega$  contains two different immiscible incompressible phases (liquid-liquid or liquid-gas) with different material properties  $\rho_i$  and  $\mu_i$   $i = 1, 2$ . The domain is subdivided into two subdomains  $\Omega_1$  and  $\Omega_2$  each of them containing one of the phases, respectively, accomplishing that  $\Omega = \Omega_1 \cup \Omega_2$  and  $\Omega_1 \cap \Omega_2 = \emptyset$ . Isothermal conditions are assumed, and reaction mass transfer and phase transition are not considered.

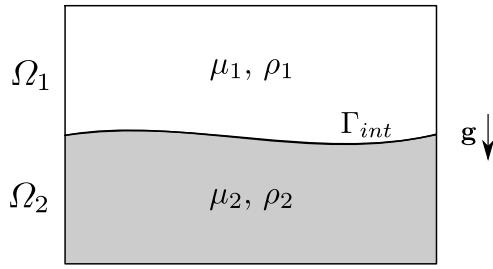


Figure 7.1: Two-phase flow configuration

In each of the phases conservation of mass and momentum has to hold, yielding separate Navier-Stokes equations in the two sub-domains  $\Omega_i$ ,  $i = 1, 2$ :

$$\begin{cases} \rho_i \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \nabla \cdot \boldsymbol{\sigma}_i + \rho_i \mathbf{g} \\ \frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho_i \mathbf{u}) = 0 \end{cases} \quad \text{in } \Omega_i \quad i = 1, 2 \quad (7.1)$$

with  $\boldsymbol{\sigma}_i = -p\mathbf{I} + \mu_i \left( \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right)$ . It must be noted that in the Eulerian framework the mass conservation equation only can be simplified into the usual incompressibility condition  $\nabla \cdot \mathbf{u} = 0$  used in homogeneous fluids if the domains are considered separately, as in Equation (7.1). Considering an unique domain the  $\frac{\partial \rho}{\partial t}$  term changes in time because material points of different densities transit along the fixed points of the domain.

At the internal interfaces, as the phases are viscous and no phase transition takes place,

the velocity can be assumed to be continuous

$$[\![\mathbf{u}]\!] = 0 \text{ on } \Gamma_{int}. \quad (7.2)$$

with  $[\!.\!]$  representing the jump of the variable across the interface  $\Gamma_{int}$ , i.e. the value close to the interface in  $\Omega_1$  minus the value close to the interface in  $\Omega_2$ .

The other condition, expressed by Equation (7.3) at the internal interface expresses that the jump in the normal stresses is balanced with the surface tension force. This force  $\mathbf{F}_\Gamma$  is proportional to the interface curvature and points to the center of the osculating circle that approximates  $\Gamma_{int}$ . The surface tension coefficient  $\gamma$  is assumed constant in this work and its value depends on the two fluids at the interface. A more general definition of  $\gamma$  could be function of the temperature and/or concentration leading place to phenomena as thermocapillary, formation of Bernard-Marangoni cells, etc.

$$\mathbf{F}_\Gamma = [\![\sigma \cdot \mathbf{n}_\Gamma]\!] = \gamma \kappa \mathbf{n}_\Gamma \text{ on } \Gamma_{int}. \quad (7.3)$$

where  $\kappa$  is the local curvature of the interface and  $\mathbf{n}_\Gamma$  is the unit normal to the interface.

Idelsohn et. al. [Idelsohn et al., 2010] affirms that the pressure jump at the interface is not only a consequence of the surface tension, but also occurs in cases where surface tension is neglected. Later, De Mier [De Mier Torrecilla, 2009] demonstrates this fact projecting the Equation (7.3) along the unit normal  $\mathbf{n}$  and unit tangent  $\tau$  vectors attached to the interface and replacing the expression of  $\sigma$ . The jump in the pressure field in these cases is a function of the viscosity jump and the derivative of the normal velocity to the interface, i.e.

$$p_1 - p_2 + \gamma \kappa = 2 (\mu_1 - \mu_2) \frac{\partial u_n}{\partial n} \quad (7.4)$$

Following same fundamentals as in Chapter 2, i.e. employing the concept of material derivative (Equation (2.3)), Equation (7.1) can be written in its Lagrangian expression as follows

$$\begin{cases} \rho_i \frac{D\mathbf{u}}{Dt} = \nabla \cdot \sigma_i + \rho_i \mathbf{g} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad \text{in } \Omega_i \quad i = 1, 2 \quad (7.5)$$

In contrast with the Eulerian framework, the Lagrangian is able to express as  $\nabla \cdot \mathbf{u} = 0$  the incompressibility condition because the total temporal variation (or material derivative) of the density is null for any particle.

Due to the characteristics of the method developed in this thesis, it is better to introduce an alternative formulation of the problem presented in Equation (7.5). If the flow is considered

as a single mixture of phases, where the fluid intensive properties depend on an indicator function  $\lambda$  used to distinguish between the two phases, i.e.  $\rho = \rho(\lambda)$  and  $\mu = \mu(\lambda)$ , the Equation (7.6) can be expressed as

$$\left\{ \begin{array}{l} \rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} + \mathbf{F}_\Gamma \\ \nabla \cdot \mathbf{u} = 0 \\ \frac{D\lambda}{Dt} = 0 \\ \frac{D\mathbf{x}}{Dt} = \mathbf{u} \end{array} \right. \quad \text{in } \Omega \quad (7.6)$$

which is known as the interface capturing system due to the inclusion of an equation for the interface movement. This set of equations has the advantage of not requiring the constant splitting of the domain into two sub-domains. In addition, the surface tension is included as a forced concentrated load along the interface, redefined as

$$\mathbf{F}_\Gamma = \gamma \kappa \delta_\Gamma \mathbf{n}_\Gamma \quad (7.7)$$

where  $\delta_\Gamma$  is the Dirac delta function that localizes the surface tension force on the interface.

# Chapter 8

## Extension of PFEM-2 to two-phase flows

There are several, albeit small, differences between the PFEM-2 algorithm for homogeneous flows and the two fluid version. Those differences stem from the density and viscosity discontinuities that appear in the fluid, added to the surface tension force treatment. Consequently most of the implemented changes are related to the strategies followed to correctly capture the interface between both fluids. Taking into account the methodology presented in Section 3.2, important considerations to manage and improve each one of the stages for the particular case of two-phase problems are presented hereafter. Three sections related with aspects of the simulation are presented, namely: the kinematic description of the interface through fluid particles during the X-IVS stage, the discontinuities treatment at the interface employing enrichment functions, and the surface tension force computing.

### 8.1 Internal interfaces evolution

Methods used to describe the evolution of interfaces can be clustered in two classes, namely: interface capturing and interface tracking methods. While in the former the interface is determined by an implicit function that is advected in an Eulerian frame, such as Volume of Fluid (VoF) [Hirt and Nichols, 1981] and Level Set Method (LSM) [Osher and Fedkiw, 2001], in the latter the interface evolution equation is solved in a Lagrangian fashion, for example, by evolving marker particles [Enright et al., 2002].

In the case of LSM, the strategy consists in using a distance function that is convected according to the fluid velocity. This function represents the distance from a point to the

interface. By definition, the interface will be located where its value is zero. This level function is variable in the space, but if it has large variations in time, after some time-steps it does not represent the distance to the interface anymore, leading to diffusion of the interface and mainly loss of mass. For this reason a reinitialization of the level set must be done to recover a distance function which guarantees that the properties are better conserved. Moreover, an Eulerian advection of the level set function produces large diffusion and requires small time-steps to achieve accurate solutions.

On the other hand, VOF is based on the conservative nature of the FVM, where instead of tracking an interface, it is more natural to save the content of each cell and define the shape from this data later on. The method defines a function that is the fraction occupied by one of the phases in each cell of the domain. Therefore the interface position is not tracked, but the fraction of fluid instead. Once fluids have been convected between cells, the interface position can be reconstructed accepting some accuracy loss. Integrating VoF with FVM is likely to be the most used strategy for two-phase flows in commercial/widespread codes, as an examples in OpenFOAM® [Weller et al., 1998] and ANSYS® Fluent.

On the other hand, formulations clustered in the Lagrangian framework are a more natural choice for simulations where there are large deformations. Using particles that are advected carrying its own properties over the domain almost avoid any numerical diffusion. In this sense, the calculation of the interface evolution is naturally done using particles with a marker value assigned.

The spatial domain discretization using mesh and particles allows PFEM-2 to select an appropriate combination of those interface evolution approaches. In this work, a strategy where the interface position information is shared and interchanged by the particles and the fixed mesh is adopted.

The proposed strategy is the following: considering two different fluid phases separated by an interface, each particle  $p$  carries the information of the phase to which it was initially assigned. This quantity, represented by a scalar function  $\lambda_p$ , assumes integer values  $-1$  or  $1$  depending if it belongs to the first or second fluid. This value is advected solving

$$\frac{D\lambda_p}{Dt} = 0 \quad (8.1)$$

which is calculated in the PFEM-2 framework just adding an equation at X-IVS stage [3.7] then

$$\left\{ \begin{array}{l} \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_{t^n}^{t^{n+1}} \mathbf{u}^n(\mathbf{x}_p^\tau) d\tau \\ \hat{\mathbf{u}}_p^{n+1} = \mathbf{u}_p^n \\ \lambda_p^{n+1} = \lambda_p^n \end{array} \right. \quad (8.2)$$

leading to each particle to keep its marker value during the entire simulation.

Therefore, the function values are projected from the particles to the mesh nodes to determine the interface position. Due to the sharp nature of the interface, care must be taken with the selection of the projection algorithm: overshoots or undershoots generate nonphysical properties around the interface. Mesh nodes thus obtain real values after the projection which are different to the integer values  $\pm 1$  that the particles transport. As mentioned previously, the projection strategy consists in a least squares minimization, and in the case of this sharp function the resulting l.h.s. matrix is lumped to avoid new maxima or minima. Finally, the interface is defined as the set of points that satisfy the equation  $\lambda = 0$ , then inside each element an iso-line (an iso-plane in 3D) is determined (see Figure 8.1a). It must be noted that this approach leads to a continuous interface through the elements.

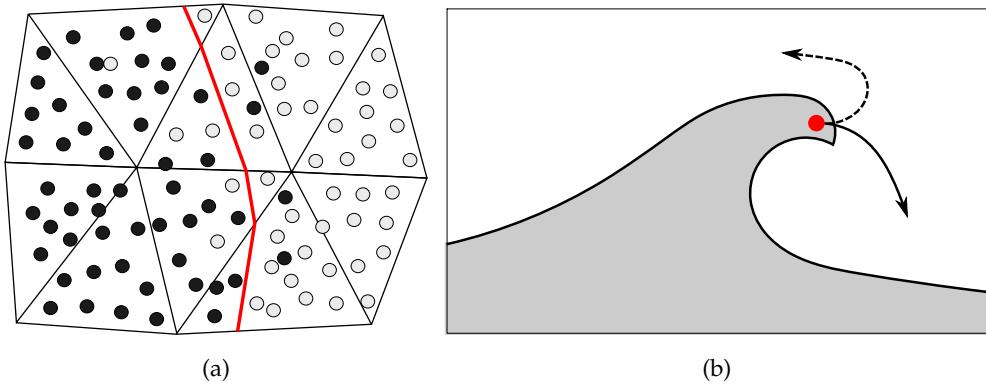


Figure 8.1: Considerations for solving two-phase flows with PFEM-2. Figure 8.1a presents a graphical description of the interface position calculation where  $\lambda(\mathbf{x}) = 0$ . Figure 8.1b shows the necessity of the modification of the particle integration in a breaking wave. Dashed line is the path following the streamline at time  $t^n$ , and continuous line the actual particle trajectory.

In general, the movement of a particle is done by sub-steps when the Equation (8.2) is solved numerically. The velocity used in the particle movement at position  $\mathbf{x}_p$  is calculated by

the equation:

$$\mathbf{u}^n(\mathbf{x}_p^\tau) = \frac{\sum_i \mathbf{u}_i^n N_i(\mathbf{x}_p^\tau)}{\sum_i N_i} \quad (8.3)$$

where the nodes included in the interpolation are the nodes of the hosting element, and  $N$  their linear shape functions associated.

Two situations could happen to any particle when its position changes: all the nodes of the hosting element have the same sign value as the fluid particle, or one or more nodes have a different sign than the particle. Whilst in the first case a typical finite element interpolation is performed, the second situation clearly represents a case where the fluid particle is close to the interface. In the particular case that the density ratio  $\rho_1/\rho_2$  is larger than a first numerical parameter  $\alpha$ , i.e.  $\rho_1/\rho_2 > \alpha$ , two situations can appear:

- $\rho_p = \rho_2$  (light particle tracking). The velocity used in the particle movement will be computed using Equation (8.3).
- $\rho_p = \rho_1$  (heavy particle tracking). Depending on the value of  $A = \sum_{i(\rho_i=\rho_p)} N_i$  where the sum is limited to the hosting nodes that have the same sign as the particle, we can have 2 possibilities:
  - $A < \beta$  the gravity force will be included in the computation of the particle trajectory, which will finally be computed as a parabolic motion.
  - $A > \beta$ , the sums that appear in Equation (8.3) are both restricted to the hosting nodes  $i$  that have the same density as the particle  $\rho_i = \rho_p$ .

being  $\beta$  a second numerical parameter independent on the density ratio, which regulates the minimum amount of information needed to move a heavy particle following the streamlines of its own phase. On the other hand,  $\rho_p$  is the density corresponding to the sign of the particle  $p$ , where if  $\lambda_p = 1$  then  $\rho_p = \rho_1$  else  $\rho_p = \rho_2$ .

This means that if a water particle is momentarily in an air regime, it will remain as a water particle for further determination of the interface position. While a better approximation to the real particle trajectory defined by the acting forces is searched, the parabolic motion is used herein as the simplest trajectory when only gravity forces are acting (see Figure 8.1b), an interesting alternative for the particle motion (not used in this work) could be using a water droplet drag model, but it requires a mass assignment to the particle. It must be noticed that for similar densities  $\rho_1/\rho_2 < \alpha$ , Equation (8.3) is used for every particle.

## 8.2 Shape function enrichments for pressure gradient discontinuity capturing

In typical finite element methods, the shape functions  $N_j$  and their gradients  $\nabla N_j$  are continuous within each element, and therefore any interpolated unknown is also continuous. When the interface crosses an element, the discontinuity in the material properties leads to discontinuities in the unknowns and/or its gradients that classical interpolations do not capture. These are:

- pressure gradient jumps where density discontinuities are present [Coppola-Owen and Codina, 2005].
- pressure jumps where viscosity discontinuities are present [Idelsohn et al., 2010].
- gradient velocity jumps where viscosity discontinuities are present [Márquez Damián, 2013].
- pressure jumps where surface tension is present [Ausas et al., 2012].

The focus of this thesis is put on the treatment of discontinuity of the pressure gradient due to the presence of two different density fluids. Under a gravitational flow and when this difference is large, such as air-water simulation, the interpolation errors in the pressure gradient using the classical FEM strategy give rise to spurious velocities that can render the solution meaningless. In this context, enrichment methods add degrees of freedom to elements that are cut by the interface in order to reduce interpolation errors. In this work, two space enrichment methodologies are proposed in order to treat those pressure gradient discontinuities.

The first enriched space is based on the one presented by Coppola [Coppola-Owen and Codina, 2005], which is illustrated, for the two-dimensional case, in Figure 8.2. The triangle conformed by the nodes at positions  $x_1$ ,  $x_2$  and  $x_3$  is cut by the interface at points  $x_A$  and  $x_B$ , dividing the element into two regions  $\Omega_1$  and  $\Omega_2$  and leaving isolated the node  $x_1$ . The construction of the enrichment function  $N^*$  must satisfy

$$\begin{cases} N^*(x_A) = 1 \\ N^*(x_1) = N^*(x_2) = N^*(x_3) = 0 \end{cases} \quad (8.4)$$

where  $x_A$  is a point over the edge  $x_1 - x_2$  such as  $\lambda(x_A) = 0$ .

It can be demonstrated [Coppola-Owen and Codina, 2005] that the enrichment function, which accomplishes the previously presented restrictions, can be expressed as a linear combination of the traditional shape functions, being:

$$N^*|_{\Omega_2} = k_1 N_1 \quad (8.5)$$

$$N^*|_{\Omega_1} = k_2 N_2 + k_3 N_3 \quad (8.6)$$

where  $k_1 = \frac{\lambda_2 - \lambda_1}{\lambda_2}$ ,  $k_2 = \frac{\lambda_1 - \lambda_2}{\lambda_1}$  and  $k_3 = -k_1 \frac{\lambda_3}{\lambda_1}$ , being  $\lambda_j$  the value of the marker function at node  $j$ . The unique new degree of freedom could be statically condensed within each element in the pressure equation and then recovered in the correction step.

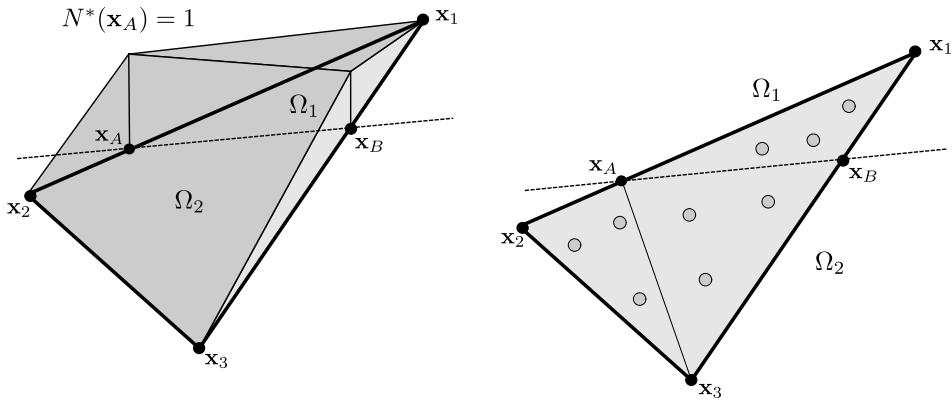


Figure 8.2: 2D interface element. The interface is calculated cutting the element at the segment  $A - B$ . The enrichment proposed by Coppola (left) and the partition of the triangle into three sub-triangles with its own Gauss points (right).

The second set of enrichment functions used in this work is described in Figure 8.3. As in the previous case, the two new degrees of freedom can also be statically condensed. However, using this new space, it is possible to ensure continuity between elements at the cost of having to rebuild the system matrix at each time-step, which could be an expensive task due to memory allocation. This enrichment space is constructed following

$$\begin{cases} N_A^*(\mathbf{x}_A) = N_B^*(\mathbf{x}_B) = 1 \\ N_A^*(\mathbf{x}_1) = N_A^*(\mathbf{x}_2) = N_A^*(\mathbf{x}_3) = N_A^*(\mathbf{x}_B) = 0 \\ N_B^*(\mathbf{x}_1) = N_B^*(\mathbf{x}_2) = N_B^*(\mathbf{x}_3) = N_B^*(\mathbf{x}_A) = 0 \end{cases} \quad (8.7)$$

Therefore, using any of the enriched spaces, the pressure is now interpolated in the cut element  $h$  following:

$$p_h(\mathbf{x}) = \sum_{i=1}^n N_i(\mathbf{x}) p_i + \sum_{i=1}^e N_i^*(\mathbf{x}) p_i^* \quad (8.8)$$

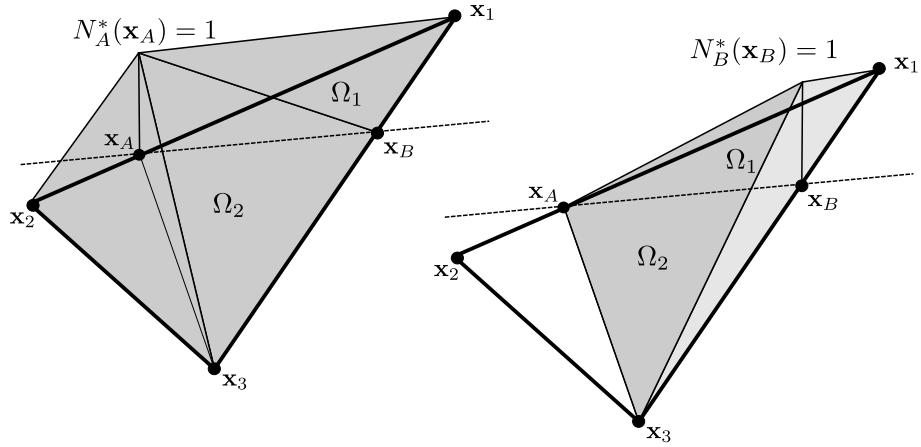


Figure 8.3: 2D interface element. The interface is calculated cutting the element at the segment  $A - B$ . An enrichment space with two functions per interface element, which can be used to ensure continuity between elements, is presented. The integration partition is the same as presented above (Figure 8.2).

where  $N_i$  are traditional linear shape functions (a total of  $n$ ), and  $N_i^*$  are the enrichment shape functions (a total of  $e$ ).

In order to capture the discontinuities and taking advantage of the enrichment functions used, the integration rules need to be modified in elements cut by the free surface. The method used is to divide each tetrahedra (triangles when represented in 2D) element into up to four tetrahedral (three triangular in 2D) sub elements. For each sub element, the same integration rule as for the non-cut elements is used. Figure 8.2 shows such a partition where the small circles represent the Gauss points for the integration. When using enrichment functions for the pressure, the material properties  $\rho, \mu$  are taken as  $\rho_1, \mu_1$  or  $\rho_2, \mu_2$ , depending on which part of the domain ( $\Omega_1$  or  $\Omega_2$ ) the integration point is found.

### 8.2.1 Pressure calculation

In contrast to the momentum predictor calculation in the one-phase case (Equation (3.24)), in two-phase flows the particles, which move across several elements, can cross the interface during a time-step, consequently, the pressure gradient of the previous time-step would introduce a poor and even unstable approximation of the new pressure forces. In order to avoid these large errors in the evaluation of the pressure gradients in the predicted velocity value, the pressure field is set to zero at the beginning of each time-step. Therefore, the acceleration at the momentum predictor stage is only due to the gravitational force and

viscous forces.

The formulation presented to find the new pressure in the case of one-phase flows (Equation (3.25)) requires imposing the pressure gradient at boundaries, which typically is neglected or zeroed. This approach is not valid in the case of two-phase flows, leading to severe errors near to the boundaries unless a prediction of the unknown gradient is imposed. A solution to this drawback consists in weakening both pressure laplacian and velocity divergence terms in order to obtain the system:

$$\left[ \Delta t \int_{\Omega} \frac{1}{\rho} \nabla N^T \nabla N \, d\Omega \right] \delta p^{n+1} = \left[ \int_{\Omega} \nabla N^T N \, d\Omega \right] \hat{\mathbf{u}}_j^{n+1} \quad (8.9)$$

where the integral at boundary resulting from both weakenings

$$\int_{\Gamma} N \left[ \hat{\mathbf{u}}_j^{n+1} + \Delta t \frac{1}{\rho} \nabla \delta p^{n+1} \right] \cdot \mathbf{n} \, d\Gamma = \int_{\Gamma} N \mathbf{u}_j^{n+1} \cdot \mathbf{n} \, d\Gamma \quad (8.10)$$

is zeroed to ensure the impenetrability on walls, is added to the l.h.s in the case of outflow, or, finally, is added to the r.h.s if velocity is imposed on inflow boundaries.

Then, in the case of split elements where enrichment shape functions are used as trial functions, the resulting local system becomes:

$$\begin{pmatrix} L_{N,N} & L_{N,*} \\ L_{N,*}^T & L_{*,*} \end{pmatrix} \begin{pmatrix} \delta p^{n+1} \\ \delta p^{*n+1} \end{pmatrix} = \begin{pmatrix} D_N \\ D_* \end{pmatrix} (\hat{\mathbf{u}}^{n+1}) \quad (8.11)$$

where

- $(L_{N,N})_{n \times n} = \Delta t \int_{\Omega^e} \frac{1}{\rho} \nabla N_i^T \nabla N_j \, d\Omega$
- $(L_{N,*})_{n \times e} = \Delta t \int_{\Omega^e} \frac{1}{\rho} \nabla N_i^T \nabla N_j^* \, d\Omega$
- $(L_{*,*})_{e \times e} = \Delta t \int_{\Omega^e} \frac{1}{\rho} \nabla N_i^{*T} \nabla N_j^* \, d\Omega$
- $(D_N)_{n \times n} = \int_{\Omega^e} \nabla N_i^T N_j \, d\Omega$
- $(D_*)_{e \times n} = \int_{\Omega^e} \nabla N_i^{*T} N_j \, d\Omega$

At this point, is possible to choose between two options. First, if the enriched space expressed by equation (8.7) is used, the new degrees of freedom can be included in the global system, guaranteeing continuity between elements. The second option is to follow the classical procedure of static condensation of the system through Gaussian elimination[Felippa, 2004], where the following reduced system is obtained:

$$\left[ L_{N,N} - L_{N,*} (L_{*,*})^{-1} L_{N,*}^T \right] (\delta p^{n+1}) = \left[ D_N - L_{N,*} (L_{*,*})^{-1} D_* \right] (\hat{\mathbf{u}}^{n+1}) \quad (8.12)$$

with  $\delta p^{n+1} = p^{n+1} - p^n$ , where the continuity between elements is not ensured.

In the case of an enriched space, see Equation (8.4), when condensation is used, the assumption of neglecting the inter-elemental boundary terms from the integration by parts of the Poisson equation is not true. From our experience, and as reported by Coppola [Coppola-Owen and Codina, 2005], when the Froude Number is high (that is, a low density ratio), the neglected term from the weakening of the divergence of the velocity introduces severe problems of numerical diffusion.

A way to solve this problem is not to integrate by parts the divergence term. Nevertheless, this requires imposing a pressure gradient on boundaries, whose value can not be easily predicted when there is a gravitational field, large density jump, and the fluid is not at rest.

Finally, if condensation is used, the weakening of the velocity divergence in the Poisson step is only chosen when a large density jump is considered. Continuous enrichment does not suffer from the above mentioned problems and the same formulation can be used for every Froude number. However, a loss of computational performance appears due to the necessity of memory management to assemble the variable-size pressure equation system.

If we are interested in solving the system in absolute pressure terms, the final system is

$$\left[ L_{N,N} - L_{N,*} L_{*,*}^{-1} L_{N,*}^T \right] (p^{n+1}) = \left[ D_N - L_{N,*} L_{*,*}^{-1} D_* \right] (\hat{\mathbf{u}}^{n+1}) + \left[ L_{N,N} - L_{N,*} L_{*,*}^{-1} L_{N,*}^T \right] (p^n) \quad (8.13)$$

After obtaining the new pressure, it is necessary to correct the velocity prediction using the pressure gradient. However, in this step, the new degree of freedom for the pressure must be taken into account when calculating the enriched pressure gradient. In the case of condensed calculation, the value of  $p^{*n+1}$  must be recovered from the nodal pressures calculated in the poisson step with:

$$p^{*n+1} = (L_{*,*})^{-1} [D_* \hat{\mathbf{u}}^{n+1} - L_{N,*}^T p^{n+1} + L_{N,*}^T p^n + L_{*,*} p^{*n}] \quad (8.14)$$

Finally, the equation system presented in Equation (8.15) must be solved.

$$\int_{\Omega} N \rho \mathbf{u}^{n+1} d\Omega = \int_{\Omega} N \rho \hat{\mathbf{u}}^{n+1} d\Omega - \Delta t \left[ \int_{\Omega} N \nabla \delta p^{n+1} d\Omega + \int_{\Omega} N \nabla \delta p^{*n+1} d\Omega \right] \quad (8.15)$$

It was previously mentioned that the pressure value is zeroed at the beginning of the time-step, one of the main reasons can be found in the recovering step. It must be noted that, not only the standard nodal pressures but also the enrichments pressures of the previous iteration are required in this stage. Since the enrichment pressures depend on the interface location inside the element, using the latest  $p^*$  of the previous time-step may introduce several

problems due to the movement of the interface. This leads to poor results that in most cases become unstable.

Although this pressure reinitialization leads to a first order temporal approximation, successive iterations on a pressure-correction loop improve the incompressibility of the solution. Idelsohn [Idelsohn et al., 2014] ensures that the stabilization effect of the first order fractional step is lost when higher order schemes are used. The same conclusion was obtained here, where instead of using a stabilization technique, a limited number of iterations (two or three) were used in order to obtain a converged pressure field without the presence of any pressure oscillations. More iterations would tend to make the scheme unstable.

### 8.3 Surface tension treatment

In a PFEM-2 simulation, the Lagrangian particles are integrated following the integration strategy called X-IVS, where the streamlines fixed at time  $n$  are employed to update the particle movements and velocities. In order to extend this approach to track the interface, each particle is marked with a sign function  $\lambda_p$  depending if it belongs to the first or second phase, and this value is preserved over the particle during the entire simulation guaranteeing boundedness. After X-IVS step, the particle values must be projected to the nodes to continue with next algorithm steps. Mesh nodes thus obtain real values after the projection which can be different to the integer values  $\pm 1$  that the particles transport. Finally, the interface is defined as the set of points that satisfy the equation  $\lambda = 0$ .

Although the enrichment proposed in previous Section is a strategy which allows to capture with great accuracy kinks of unknowns at interface, which is of interest on physical problems dominated by gravitational forces with large density ratios or fluid-structure simulations with large viscosities ratios [Becker et al., 2014], its use is not necessary in other problems where the flow regime is different, such as in the case of surface tension dominant, where the strong requirement is reducing spurious currents when the surface term is applied.

Once projected over nodes, the function  $\lambda$  has similar properties to a VoF function: the mass is preserved but the discontinuous shape impossibilities an accurate gradient calculation, from which is obtained the normal to the interface. If that normal is employed to estimate the curvature, the poor results lead to unphysical flows around the interface when surface tension is included, resulting in unrealistic interface shapes. Strategies to reduce the spurious currents based on either interface reconstruction or smoothing kernels are available, a literature

review can be found in [Lam, 2009], but most of them must be employed only on structured meshes. The coupling achieved by advecting the interface using the conservative VoF function, calculating the interface normal using the smoothed LS function and updating the physical properties from a smoothed Heaviside function is an improved strategy called CLSVOF [Sussman and Puckett, 2000]. In [Albadawi et al., 2013], Albadawi presents a less expensive option called S-CLSVOF which uses an one-way coupling strategy without need to advect the level set function, and in this work its approach is selected. Then the initial level set-like function  $\phi_0$  is obtained following

$$\phi_0 = \frac{3}{4} \Delta x \lambda \quad (8.16)$$

The main criterion in choosing this value is to satisfy an initial value of  $\phi$  which is close to the mesh step size. This initial function is a signed function since it has a positive value in the denser fluid and a negative value in the lighter. However, in order to obtain a  $|\nabla\phi| = 1$  around the interface, the function is then re-distanced by solving the re-initialization equation:

$$\frac{\partial \phi}{\partial \tau} = \text{sign}(\phi_0)(1 - |\nabla\phi|) \quad (8.17)$$

with the initial condition of  $\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x})$ , being  $\tau$  an artificial time discretized with  $\Delta\tau = 0.1\Delta x$ . Because the re-distancing starts from the initial interface and moving towards both fluids, and we are interested only on the zone around the interface, only few iterations  $\phi_{corr} = \frac{\epsilon}{\Delta\tau}$  are required, with  $\epsilon$  representing the width around the interface, typically  $1.5\Delta x$ . This reinitialization process does not leads to the typical breakdown of the mass conservation due to the original information is kept over the particles.

After solving 8.17 the  $\phi$  is now a continuous function smooth around the interface, which helps in determining accurately the interface normal  $\mathbf{n}$  as usual in LSM

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \quad (8.18)$$

Hence, it provides a more precise and smoother interface curvature

$$\kappa = \nabla \cdot \mathbf{n} \quad (8.19)$$

Some details are important to remark when Equations 8.18 and 8.19 are solved in the FEM framework. An initial strategy is to replace 8.18 into 8.19, and to use weighted residuals with the linear piecewise trial functions  $\mathbf{N}$  solving directly for  $\kappa$ , it is

$$\int_{\Omega} \mathbf{N} \kappa \, d\Omega = - \int_{\Omega} \nabla \mathbf{N} \frac{\nabla\phi}{|\nabla\phi|} \, d\Omega - \int_{\Gamma} \mathbf{N} \frac{\nabla\phi}{|\nabla\phi|} \cdot \boldsymbol{\eta} \, d\Gamma \quad (8.20)$$

However this approach lead to spurious results because  $\nabla\phi$  is discontinuous between the elements. A further option which obtain better results is first obtain a field  $\hat{\mathbf{n}}$  with continuous  $\nabla\phi$  between the elements (linear field in the elements), doing

$$\int_{\Omega} \mathbf{N}\hat{\mathbf{n}} d\Omega = \int_{\Omega} \mathbf{N}\nabla\phi d\Omega \quad (8.21)$$

then, obtain the curvature as usual

$$\int_{\Omega} \mathbf{N}\kappa d\Omega = - \int_{\Omega} \nabla\mathbf{N}\mathbf{n} d\Omega - \int_{\Gamma} \mathbf{N}\mathbf{n} \cdot \boldsymbol{\eta} d\Gamma \quad (8.22)$$

with  $\mathbf{n} = \frac{\hat{\mathbf{n}}}{|\hat{\mathbf{n}}|}$  and  $\boldsymbol{\eta}$  the normal to the boundary  $\Gamma$ .

One of the most difficult tasks in front-capturing techniques is to accurately identify the interface to directly impose the term  $\mathbf{F}_{\Gamma}$ . This difficulty can be alleviated by interpreting the surface tension as a continuous body force spread across a transition region of thickness avoiding the need of reconstructing the interface explicitly. In this way, the continuum surface force model (CSF) of Brackbill et al. [Brackbill et al., 1992] provides an approach to approximate the term of surface tension force  $\mathbf{F}_{\Gamma}$  as a force per unit volume as

$$\mathbf{F}_{\gamma} = \gamma\kappa\nabla\phi\delta_s(\phi) \quad (8.23)$$

where  $\delta_s$  is the regularized interface delta function defined as follows

$$\delta_s(\phi) = \begin{cases} 0, & \phi > |\epsilon| \\ \frac{1}{2\epsilon}(1 + \cos(\frac{\pi\phi}{\epsilon})), & \phi \leq |\epsilon| \end{cases} \quad (8.24)$$

It has been presented [Brackbill et al., 1992, De Mier Torrecilla, 2009] that if the surface tension term on Equation (8.23) is discretized explicitly, i.e. the surface tension forces are evaluated on the interface at the previous time-step, the stability of the scheme imposes the following restriction on the time-step size  $\Delta t_{max}$ :

$$\Delta t_{max} = \sqrt{\frac{\rho\Delta x^3}{\gamma}} \quad (8.25)$$

With this restriction the propagation of capillary waves is resolved and their unstable amplification avoided. The Equation (8.25) can be rather limiting for fine meshes and large surface tension coefficients then is a relevant issue in order to preserve the large time-step proposed by PFEM-2. A solution to partially overcome this limitation is to treat the force term (8.23) implicitly. In this proposal, the surface tension term is included into the implicit calculation of the momentum equation over the mesh using the updated interface position.

This helps to extend the time-step limitation but is not a fully implicit approach because the interface movement is not coupled with the surface tension imposition. An analysis of the stability of this proposal is presented in Section 11.1.

The material properties over the mesh ( $\rho$ ,  $\mu$ ) need to be somehow approximated in elements cut by the interface. A possibility mentioned in previous section consists in using an enhanced integration rule. However that approach, designed to be applied with enrichment, is in conflict with the smoothing proposed by Brackbill in CSF. Hence, when surface tension is not negligible and enrichment is not employed, the material properties are calculated with the smoothed Heaviside function  $H$

$$H(\phi) = \begin{cases} 0, & \phi < \epsilon \\ \frac{1}{2} \left[ 1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\epsilon}\right) \right], & \phi \leq |\epsilon| \\ 1, & \phi > \epsilon \end{cases} \quad (8.26)$$

$$\rho(\phi) = \rho_1 + (\rho_2 - \rho_1)H(\phi) \quad (8.27)$$

$$\mu(\phi) = \mu_1 + (\mu_2 - \mu_1)H(\phi) \quad (8.28)$$

It must be noticed the relevance of the parameter  $\epsilon$  in this strategy. This parameter, which determines the extent of the interface smearing, has been analyzed in other works [Albadawi et al., 2013], concluding on the necessity of using the previously mentioned value that preserves a narrow thickness. There are alternative sharp interface methods such as the Ghost Fluid approach [Fedkiw et al., 1999] which respects jump discontinuities across the interface and avoids an interface thickness. However, in these type of strategies the extension to unstructured meshes is far from being straightforward. A FEM framework strategy to treat surface tension without thickness is employing enriched shape functions to treat pressure jumps as proposed by Ausas [Ausas et al., 2012]. In spite of the possibility of capturing jumps, the curvature calculation still being a difficult task. Height functions [Lorstad et al., 2004] seems to be the best option, but its formulation for 3d unstructured meshes is still an open challenge. The Laplace-Beltrami formulation [Bänsch, 2001] appears as an interesting alternative because the curvature does not appear explicitly. However, this strategy also presents drawbacks: it is only accurate with small surface deformations, and requires the computing of the interfacial mesh leading to expensive computations.

## 8.4 Algorithm synopsis

It is assumed that all fluid variables are known at time  $t^n$  for both the particles and the mesh nodes. Subindexes  $(\cdot)_j$  y  $(\cdot)_p$  represent a generic mesh node  $j$  and a generic particle  $p$  respectively. Let  $N$  the finite element linear basis functions. According to this notation, the steps are presented in Algorithm 3, where  $\mathbf{x}$  is a spatial coordinate,  $\delta p = p^{n+1} - p^n$  is the pressure correction,  $\delta \mathbf{u} = \mathbf{u} - \hat{\mathbf{u}}$  is the velocity correction computed on the mesh,  $\theta_p$  can be 0 or 1 depending on the pressure restart choice affecting the approximation order of the pressure. Also,  $\theta_\mu$  can be 0 or 1 depending on the necessity or not of an accurate diffusion calculation when large Fourier numbers are employed,  $\mathbf{F} = \int_{\Omega} N_j(\rho \mathbf{g} + \mathbf{F}_\gamma) d\Omega$ , and  $\mathbf{M}$  and  $\mathbf{K}$  are the standard mass and stiffness matrices of any FEM assembling.

---

**Algorithm 3** - Time-Step PFEM-2 for two-phase incompressible fluids.

---

1. Convective Stage:

$$\begin{cases} \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{u}^n(\mathbf{x}_p^\tau) d\tau \\ \hat{\mathbf{u}}_p^{n+1} = \mathbf{u}_p^n \\ \lambda_p^{n+1} = \lambda_p^n \end{cases}$$

2. Projection Stage:

$$\begin{aligned} \mathbf{M}_{ij}\hat{\mathbf{u}}_j^{n+1} &= \mathbf{M}_{ip}\hat{\mathbf{u}}_p^{n+1} \\ \mathbf{M}_{ij}^L \lambda_j^{n+1} &= \mathbf{M}_{ip} \lambda_p^{n+1} \end{aligned}$$

3. Momentum Stage:

$$\left( \mathbf{M}\left(\frac{\rho}{\Delta t}\right) + \mathbf{K}(\mu) \right) \hat{\mathbf{u}}^{n+1} = \mathbf{M}\left(\frac{\rho}{\Delta t}\right) \hat{\mathbf{u}}^{n+1} - \theta_p \mathbf{G} p^n + \mathbf{F}^{n+1}$$

4. Poisson Stage:

$$\mathbf{K}\left(\frac{\Delta t}{\rho} + \tau\right) p^{n+1} = \mathbf{B}\hat{\mathbf{u}}^{n+1} + \theta_p \left( \mathbf{K}\left(\frac{\Delta t}{\rho}\right) p^n + \mathbf{B}(\tau) \pi^n \right)$$

5. Correction Stage:

$$\begin{aligned} \mathbf{M}(\rho)\mathbf{u}^{n+1} &= \mathbf{M}(\rho)\hat{\mathbf{u}}^{n+1} - \Delta t \mathbf{G} \left( p^{n+1} - \theta_p p^n \right) + \theta_\mu \mathbf{K}\left(\frac{\mu}{\rho}\right) \left( \mathbf{u}^{n+1} - \hat{\mathbf{u}}^{n+1} \right) \\ \rho_p \mathbf{u}_p^{n+1} &= \rho_p \hat{\mathbf{u}}_p^{n+1} + \sum_j \delta \mathbf{u}_j^{n+1} N_j(\mathbf{x}_p^{n+1}) \end{aligned}$$


---

where  $\tau$  is a stabilization parameter for the pressure and  $\pi^n$  is the recovery of the pressure gradient on mesh nodes enforcing its continuity.

The computational implementation is done extending the work presented in Chapter 4 for the two-phase modeling. Regarding to efficiency, the code preserves the same performance as presented for the case of one-phase flows. Maybe the most heavy task among the novelties for the two-phase flows is the reinitialization algorithm, which adds around of 30% plus of CPU time per time-step when is enabled, i.e. when the surface tension force can not be neglected.



# Chapter 9

## Interface tracking problems

This section will deal with an exhaustive validation of the proposed PFEM-2 method to transport arbitrary shapes with neither interface disturbances nor mass losses. It is well known that employing the Lagrangian scheme is relatively easy to solve pure-advection problems as presented in this section, but we consider that it is important for the reader to reach a strong conclusion about the goodness of this framework in contrast with the problems observed with the typical Eulerian schemes. The latter are represented by the suite OpenFOAM® which implements a VoF strategy with interface compression. As will be shown, the larger time-step is employed, the more relevant are the differences between both frameworks.

### 9.1 Rigid body rotation of Zalesak's disk

This test consists in the advection of a region composed of a circle with a slot [Zalesak, 1979]. If the interface tracking is accurate enough, after several revolutions, the shape must remain identical. The computational domain employed is  $\Omega \in R^2 : [0;100] \times [0;100]$ . The advected region is a circle centered at (50;75) with a radius of 15 and a slot of width 5 and height 25. The velocity field is a rigid body rotation around the center of the domain with a period of 628 time units:

$$u = (\pi/314)(50 - y), \quad (9.1)$$

$$v = (\pi/314)(x - 50) \quad (9.2)$$

The grid has 100 points in each direction, conforming a cartesian mesh (in the case of PFEM-2 simulations the mesh was split into 20000 triangles). The Courant number used in simulations is approximately  $CFL = 4.5$ . Both the initial field and the solution after two

revolutions are shown on Figure 9.1. In the case of PFEM-2 simulation, approximately five particles by element were used. Most relevant OpenFOAM® settings are: SuperBee as the divergence scheme for the linear term in volume fraction advection equation, MULES as the time integration scheme, the number of `alphaSubCycles` is 20 (to guarantee interface Courant number less than 0.5) and the interface-compression factor `cAlpha` is set to 1.

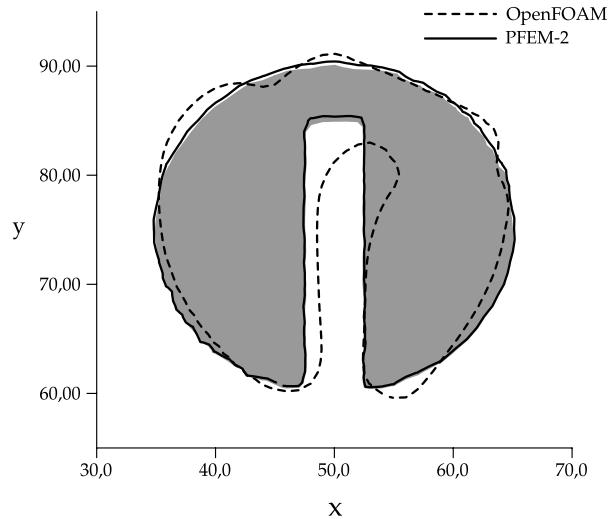


Figure 9.1: Zalesak’s disk results after two full revolutions with 100 grid point per direction and  $CFL = 4.5$ . The grey region represents the initial condition.

PFEM-2 evolution shows a good agreement with the expected result (shape preservation). Some small errors, which are more evident when the magnitude of velocity is higher, appear due to approximate a curve with a sequence of straight trajectories. Even though in OpenFOAM® simulation the interface-compression method combined with the advection scheme avoids numerical diffusion, they modify the disk shape excessively, finishing in a poor prediction of the final status.

## 9.2 Single vortex case

While Zalesak’s disk test is a good indicator of numerical diffusion in an interface-capturing method, it does not test the ability to preserve small scale structures of the fluid flow. A well known test to evaluate the ability of the method to solve structures of different sizes and their evolution is given by the *vortex-in-a-box* problem introduced by Puckett et al. [Puckett et al., 1997]. The difficulty of this tests is that requires the solution of an interface

stretching problem. The computational domain is  $\Omega \in R^2 : [0;1] \times [0;1]$ , where the interest region is a circle centered at  $(0.5; 0.75)$  with a radius of 0.15, advected with a velocity field defined by the stream function

$$\psi(\mathbf{x}) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right)$$

being the velocity components

$$u = \psi_x = \sin^2(\pi x) \sin(2\pi y) \cos\left(\frac{\pi t}{T}\right)$$

$$v = -\psi_y = -\sin^2(\pi y) \sin(2\pi x) \cos\left(\frac{\pi t}{T}\right)$$

The grid has 256 points in each direction, and the Courant number used in simulations is approximately  $CFL = 4.8$ .

The setting employed for each numerical method in this case is almost equal to the previous test, with the only one difference that in OpenFOAM® the interface-compression factor `cAlpha` is set to 0.25 to give more stability through relaxing in some level the strong sharpness imposition. Using a larger factor, the simulation turns unstable.

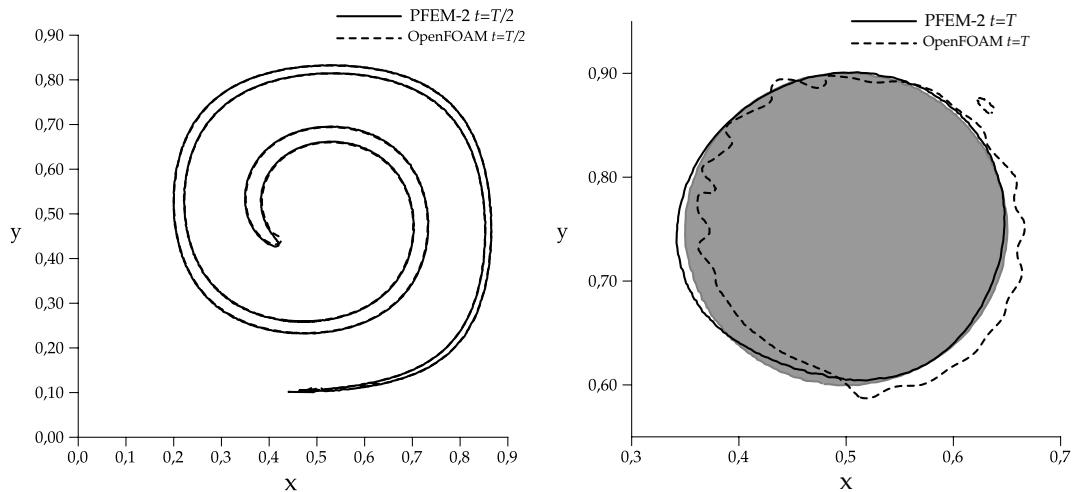


Figure 9.2: Single vortex test using 256 grid points per direction and  $CFL = 4.8$  ( $T = 8$ ). Grey region represents the initial condition.

The results presented in Figure 9.2 show, for PFEM-2, good agreement with the expected result (shape preservation) after the cycle. Although the first half of the evolution is well captured by OpenFOAM®, the reconstruction of the original shape is not good enough.

### 9.3 LeVeque deformation case

LeVeque [LeVeque, 1996] proposed a three dimensional incompressible flow field which combines a deformation in the  $x - y$  plane with one in the  $x - z$  plane. This problem can be considered an extension of the previous case, requiring the correct capturing of the stretching phenomenon in three dimensions.

- The computational domain is  $\Omega \in R^2 : [0; 1] \times [0; 1] \times [0; 1]$ .
- The advected region is a circle centered at  $(0.35; 0.35; 0.35)$  with a radius of 0.15.
- The velocity field is given by

$$\begin{aligned} u &= 2 \sin^2(\pi x) \sin(2\pi y) \sin(2\pi z) \cos\left(\frac{\pi t}{T}\right) \\ v &= -\sin(2\pi x) \sin^2(\pi y) \sin(2\pi z) \cos\left(\frac{\pi t}{T}\right) \\ w &= -\sin(2\pi x) \sin(2\pi y) \sin^2(\pi z) \cos\left(\frac{\pi t}{T}\right) \end{aligned}$$

- The grid has 50 points in each direction.
- Courant number used in simulations is approximately  $CFL = 4$ .

Since the flow is reversed for  $t > T/2$ , after one period the function must return to its original shape. Figure 9.3 shows that PFEM-2 successfully recovers almost the initial shape, which is a very complicated task for other numerical strategies [Enright et al., 2005, Costarelli et al., 2013].

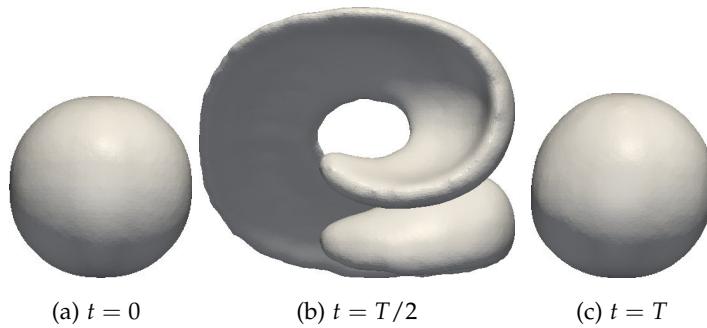


Figure 9.3: Snapshots of 3D deformation field test with PFEM-2. There were used 50 points per direction and  $CFL = 4$ . Results were smoothed by post-processing purposes.

# Chapter 10

## Inertial-dominant problems

In this section, a wide range of two-phase problems, where the inertial term is dominant, with different ratio between densities and viscosities of the fluids involved are solved using the PFEM-2 method and the results compared with reference ones. Reference data includes numerical results, experimental analysis and/or analytical solutions. Due to the features of the problems evaluated in this section, the enrichment strategy is enabled for every case in order to capture more accurately the free-surface, this leads to employ the integration rule presented in 8.2. On the other hand, surface tension force is neglected.

The first case is the widely known Rayleigh-Taylor instability, where a small perturbation must generate complex fluid structures that are well reported in literature. This problem focuses on the capabilities of the method to deal with large time-steps. These results are compared with those obtained with a well reputed Eulerian strategy, named Volume of Fluid (VoF), which adds limiters as a method of guaranteeing boundedness of phase-fractions, and interface compression numerical terms to keep the interface sharp. The second test is a sloshing problem which allows the numerical strategy behavior to be tested when different density ratios are simulated, the upcoming results are compared to well validated codes. In this second test, a discussion about the enrichment strategies is presented. Next, the ability of the method to deal with a highly dissipative free surface flow is tested with a standing wave problem, results are compared to semi-analytical solutions of the decayment of the total energy of the system. Finally, the last example is a direct comparison against results coming from a dam-break experiment. This case shows that the method can solve large motions of the interfaces and splashing of waves, while maintaining acceptable levels of accuracy when a comparison against pressure and height measurements is performed.

The main aim of this section is to show the capability of the method to work with large Courant numbers without stability loss and with negligible resign accuracy, reasonable large time-steps are then selected for each test. As presented in author master thesis (see [Gimenez, 2014]), the computational cost of each time-step is almost equal to traditional Eulerian solvers, and the increase of time-step decreases the duration of the global computation without loss of accuracy. Herein, the efficient distributed-memory implementation presented in [Gimenez et al., 2014] is extended to the two-phase treatment and used to simulate each of next cases presented.

## 10.1 Rayleigh-Taylor instability

This problem is based on the evolution of two layers of fluids initially at rest in a gravity field. The density of the upper most fluid is larger than the one placed underneath. Due to a little disturbance in the contact surface the more dense fluid moves down and the less dense fluid does the opposite. During the evolution of the problem, 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 growth and evolution of the instability has been investigated among others by Tryggvason [Tryggvason, 1988] for inviscid incompressible flows, and by Guermond & Quartapelle [Guermond and Quartapelle, 2000] for viscous flows.

The starting point is the problem documented by Guermond [Guermond and Quartapelle, 2000]. The computational domain is  $[-d/2, d/2] \times [-2d, 2d]$  and the initial position of the perturbed interface is  $\eta(x) = 0.1d \cos(2\pi x/d)$ . The density ratio is 3, which corresponds to an Atwood number of 0.5 according to Tryggvason's definition  $At = (\rho_{max} - \rho_{min}) / (\rho_{max} + \rho_{min})$ . Other physical parameters are selected to obtain a Reynolds number  $Re = \rho_{min} d^{3/2} g^{1/2} / \mu = 1000$ . The computational domain is discretized into 80000 structured triangles ( $\Delta x = 0.01$ ) setting slip boundary conditions on each wall. The time-step selected is  $\Delta t = 0.01$ [s], which allows for  $CFL_{max} \approx 8$ . Between five and eight particles per element are used and two pressure iterations are required.

To compare solutions with reference results, the time is made dimensionless by using  $\tilde{t} = t \sqrt{g At}$ . Results on the vertical position of the tip of the falling and rising fluid (spike and bubble, respectively) are shown in Figure 10.1. It can be observed that current solution is in good agreement with the reference results.

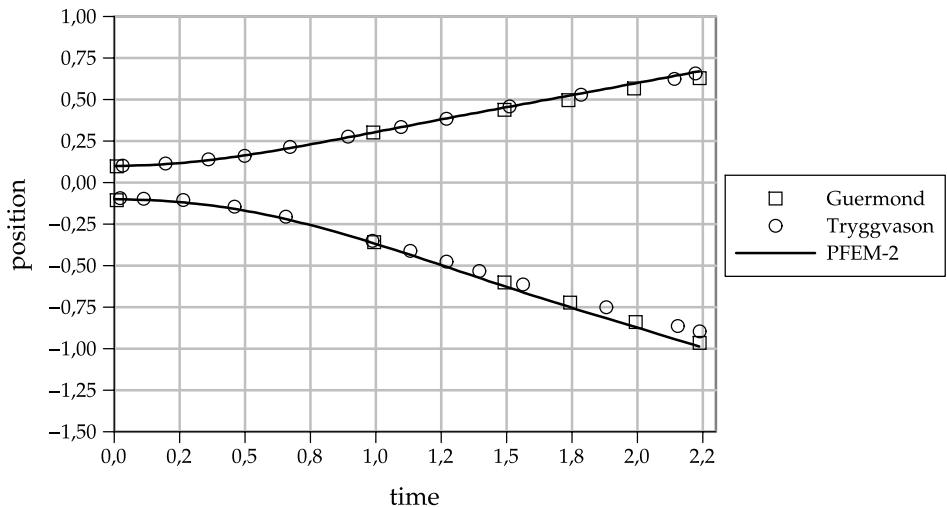


Figure 10.1: Position of rising and falling bubbles versus time. Case with  $Re = 1000$ .

On the other hand, the evolution of the instability is shown in Figure 10.2 at dimensionless times  $\tilde{t} = 0, 1, 1.5, 2$ . Around  $\tilde{t} = 1.5$  the heavy fluid begins to roll up into two counter-rotating vortices. Later, around  $\tilde{t} = 2$ , these two vortices become unstable and a pair of secondary vortices appear at the tails of the roll-ups. These shapes of the fluid interface obtained with PFEM-2 are similar to the ones shown as reference results, see [Guermond and Quartapelle, 2000].

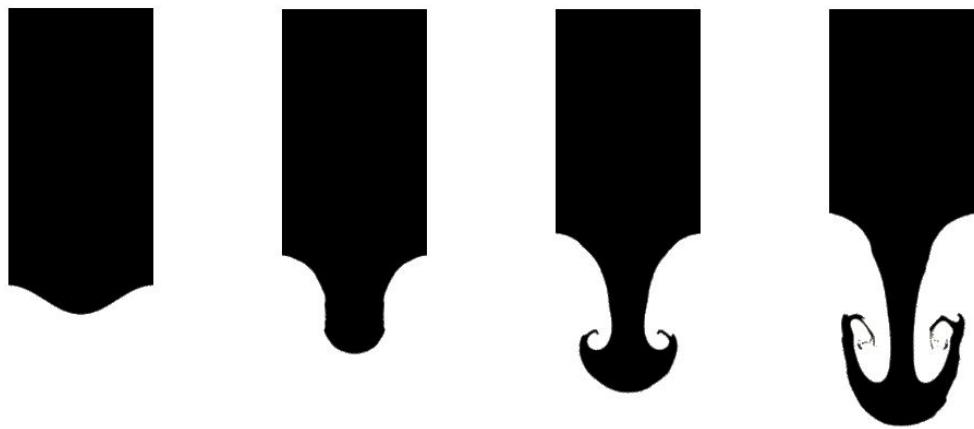


Figure 10.2: Rayleigh-Taylor instability evolution. Case with  $Re = 1000$ . From left to right  $\tilde{t} = 0.0, 1.0, 1.5, 2.0$ .

### 10.1.1 Extending the time-step

In order to emphasize the capability of the method to manage large time-steps, the current case is also simulated with a large range of  $\Delta t$  using the in-house implementation of PFEM-2 and comparing with results obtained by the widely known OpenFOAM® suite. The problem setup and domain discretization is the same as presented above and the PFEM-2 settings are not modified.

In the case of OpenFOAM®, the solver `interFoam` is chosen, which implements a Volume of Fluid (VoF) algorithm for multiphase flow [Berberović et al., 2009][Márquez Damián and Nigro, 2014]. It includes the multi-dimensional limiter for explicit solution (MULES) as a method of guaranteeing boundedness of scalar fields, in particular phase/mass-fractions (more information about MULES can be found in [Márquez Damián, 2013]). Since OpenFOAM® version 2.3, a new semi-implicit variant of MULES has been introduced which combines operator splitting with application of the MULES limiter to an explicit correction rather than to the complete flux. This approach would maintain boundedness and stability at an arbitrarily large Courant number. In the next simulation, the following recommended simulation schemes have been used: CrankNicolson (second order, implicit) time integration, Gauss linear (second order, Gaussian integration with linear interpolation) discretization for the gradient, divergence and Laplacian operators (corrected with two `nNonOrthogonalCorrectors` due to the triangular mesh, for the later). Relevant VoF settings are: `nAlphaSubCycles` is set in order to keep the maximum local CFL of the sub-cycling around 0.5, `cAlpha= 0.25` to give more stability through relaxing on some level the strong sharpness imposition, and `MULESCorr` is enabled to calculate the limiter in a semi-implicit way.

Table 10.1 presents snapshots comparing the solutions with PFEM-2 and OpenFOAM® at a particular time ( $\hat{t} = 2.25$ ) using several fixed time-steps, i.e. global CFL numbers. For normalization purposes of this particular case, that non-dimensional number is defined as  $CFL_{rt} = U\Delta t/\Delta x$ , being  $U = \sqrt{gd}$  a reference velocity and  $d$  the fluid column of one of the phases. From captures, it can be shown that PFEM-2 keeps approximately the same solution with each  $CFL_{rt}$ , but `interFoam` can not solve with any accuracy using  $CFL_{rt} > 0.5$  because the evolution of the mushroom-like interface differs from the reference results and this divergence is increased with larger time-steps.

Another relevant feature to take into account when comparing both algorithms is that similar CPU times are required to solve a time-step. Table 10.2 summarizes the CPU Times

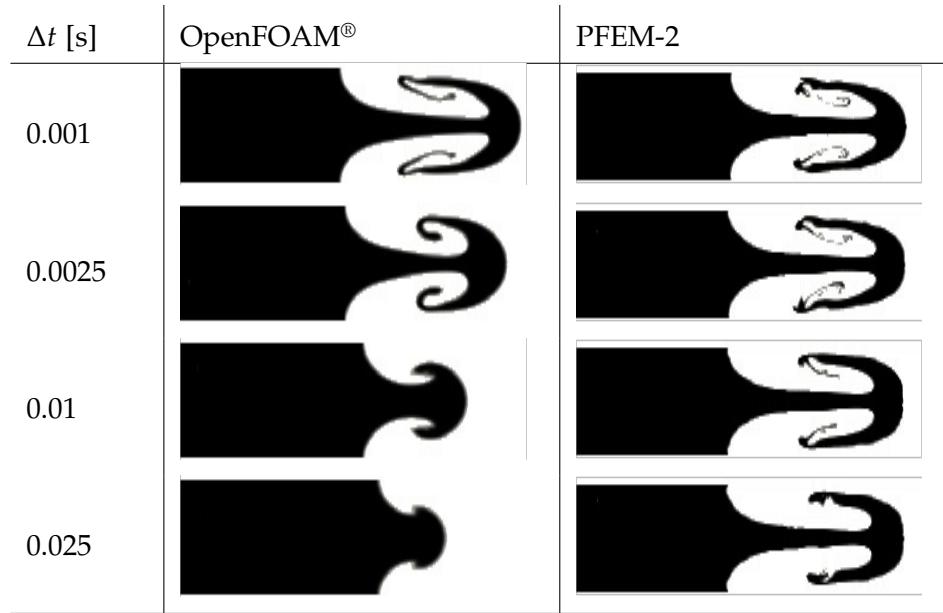


Table 10.1: Rayleigh-Taylor instability captures for  $\tilde{t} = 2.25$ . OpenFOAM® simulation implements VoF+MULES simulation (interFoam solver).

required to complete 1[s] of real time in the current case. Results show that, using the same time-step, both solvers have similar performance, being OpenFOAM® faster. However, due to the capability of time-step enlargement with PFEM-2, shorter CPU times are achieved with similar accuracy.

Solver	$\Delta t$	CPU Time
OpenFOAM®	0.001	1121[s]
PFEM-2	0.0025	1011[s]
PFEM-2	0.01	288[s]
PFEM-2	0.025	123[s]

Table 10.2: Total computing times to simulate 1[s] of real time of the Rayleigh-Taylor instability 2d. Running on an Intel i5-3230M CPU @ 2.60GHz with 8Gb of RAM and one processor.

### 10.1.2 Three-dimensional simulation

In this section, the extension of the two dimensional problem to three dimensions is presented. The third dimension is generated as a surface of revolution from the previous 2d geometry, conforming a cylindrical volume in 3d of radius  $R = 0.5$ . This allows the same problem configuration to be kept, this is, a slip boundary condition on the wall,  $At = 0.5$ , and an initial perturbation of the surface  $\eta(r) = 0.1d \cos(2\pi r/d)$ , with  $0 < r < R$ .

The computational domain is discretized with a mesh size of  $\Delta x = 0.03[m]$  conforming a non-structured mesh with around 1.2 million of tetrahedral elements. An average of eight million particles are used during the simulation that move across the light-phase and heavy-phase domains. Simulation was carried out with a  $\Delta t = 0.025$  which peaks to  $CFL_{max} = 15$ . Figure 10.3 shows the evolution of the heavy-phase. It must be noticed that the simulation is extended until reaching the stable condition with the heavy-phase at the bottom and at rest, which is approximately 30[s] of simulation time. To complete the entire simulation, the implementation requires around three wall-clock hours running on an AMD Opteron 6376 @ 2.3GHz with a 64Gb RAM using 16 processors.

The spirit of this section is to show the stability of the 3D simulation with a realistic progress. However, a way to prove the validity of the solution is that, during the simulation, the initial mass quantity is preserved. The conservation is theoretically expected since the analyzed case does not have a mass inlet and/or outlet. A more detailed analysis about the accuracy of 3d simulations will be presented in Section 10.4.1.

## 10.2 Non-linear sloshing in a rectangular container

Free surface oscillations of a liquid confined in a closed container (sloshing phenomenon) are an important issue when large amounts of liquid are industrially transported. The phenomenon involves two fluids that share a free surface boundary separating them, normally the density of the upper fluid is several orders of magnitude less than the bottom one. This phenomenon has proven of great interest due to the fact that violent impacts of the fluid can affect the structural integrity of the container.

For the studied cases in this section, the sloshing phenomenon is produced by a horizontal harmonic excitation  $x = a_h \sin(\omega_h t)$ , where  $a_h$  is the excitation amplitude and  $\omega_h$  is the excitation frequency of the rectangular tank where the two fluid phases are contained. The tank is divided in two parts, the bottom part with water with a density of  $\rho_I = 1000[kg/m^3]$

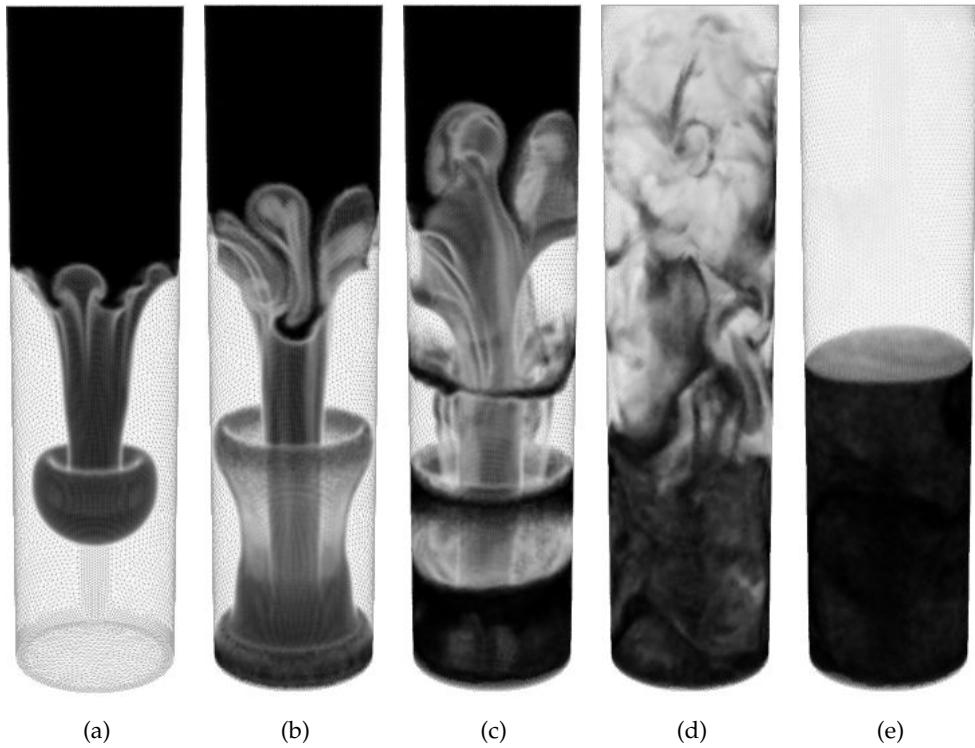


Figure 10.3: Snapshots of the heavy-phase in the Rayleigh-Taylor instability solved in three-dimensions with PFEM-2. From left to right  $\tilde{t} = 2[s], 3.3[s], 4.4[s], 8, 8[s]$ , and  $27.5[s]$ .

and the top part which contains a fluid with different densities  $\rho_{II} = 1.3, 50, 200, 800[\text{kg}/\text{m}^3]$ , depending on the studied case. The dimensions of the tank are  $a$ (width) by  $b$ (height) and the initial free surface is at height  $h$  from the bottom of the tank, see Figure 10.4. The free surface starts the simulation as a horizontal line and is subsequently deformed by the tank excitation and the flow dynamics.

For the different cases, a 2D rectangular tank with a width  $a = 1.0[\text{m}]$  and a height of  $b = 1.0[\text{m}]$  is used. The initial height of the interface is  $h = 0.5[\text{m}]$  and the lateral excitation applied is  $x = 0.05 \sin(3t)$ . The simulations were performed considering the flow as laminar and non-viscous, hence, no turbulence model was included and slip boundary conditions were used. The density jump  $\sigma = \frac{\rho_{II}}{\rho_I}$  was modified to study its influence on the free surface evolution. A two dimensional Cartesian mesh of  $450 \times 225$ , splitted into triangles, has been used in all cases.

Reference results for this case are taken from [Gómez-Goñi et al., 2013] which uses the codes STARCCM+ and OpenFOAM® to obtain numerical solutions and reports the free surface displacement on the left wall of the container. Those simulations use the same grid as

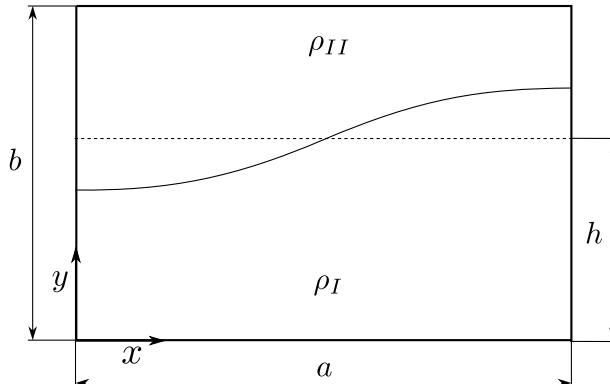


Figure 10.4: Configuration of the Non-linear sloshing in a rectangular container case. Initial condition is represented by a dashed line. The continuous line represents the position of the free-surface at a certain time.

presented above, but, in order to avoid numerical instabilities, the  $CFL$  number was limited to  $CFL_{max} = 0.5$  which implies  $\Delta t \approx 0.001$ . In PFEM-2 such restrictions do not exist,  $\Delta t$  is therefore fixed to 0.01, reaching a  $CFL_{max} \approx 5$ .

Regarding to the numerical parameters  $\alpha$  and  $\beta$  presented in Section 8.1, the simulations were done using a value of  $\alpha = 10$  and  $\beta = 0.1$ . In this way, two cases where heavy particles on lighter media are moved with parabolic motion ( $\sigma = 0.0013$  and  $\sigma = 0.05$ ), and two cases where the trajectory prediction of those particles is done following streamlines ( $\sigma = 0.02$  and  $\sigma = 0.8$ ), are analyzed.

Figure 10.5 presents the free surface displacement reported on the left wall of the container for different values of  $\sigma$ . For each one of them, PFEM-2 simulations show a good agreement with reference solutions. It is worth mentioning that the time-step used is around ten times bigger than the one used in [Gómez-Goñi et al., 2013].

### 10.2.1 Enrichment and density jump issues

The PFEM-2 results presented in the previous section, have been obtained using the continuous enrichment strategy, which allows for the same formulation to be used, independently of the Froude number. However, as was mentioned before, without using a continuous formulation between elements for the enriched shape functions, the assumption about the inter-elemental boundary terms of the Poisson equation formulation should be revisited. In this subsection, problems that appear when the formulation combines non-continuous enriched shape functions and small density jump ( $\rho_I/(\rho_I - \rho_{II}) \sim 1$ ) are

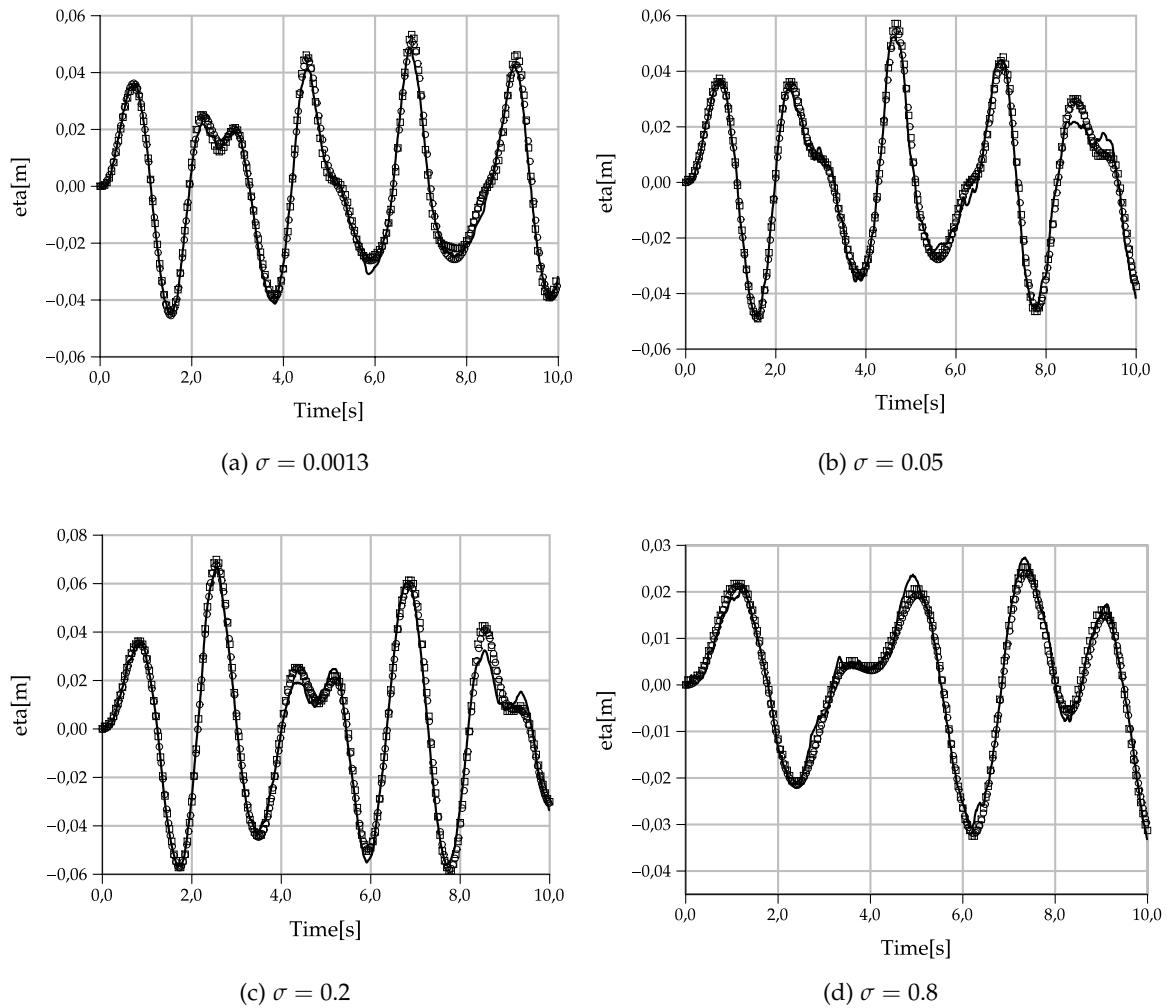


Figure 10.5: Level height on the left wall for a two phase flow for different density jumps.

References: ○ STARCCM, □ OpenFOAM® and filled line PFEM-2.

presented. The latter is somewhat quantified by the Froude number, which also includes the ratio between inertial and gravitational forces, this is  $Fr = \frac{U^2}{gL} \frac{\rho_I}{\rho_I - \rho_{II}}$ . This phenomenon is mentioned, but not investigated, by Coppola [Coppola-Owen and Codina, 2005].

Figure 10.6 shows a comparison between the level height calculated by PFEM-2 using continuous enrichment, discontinuous enrichment and no-enrichment for two extreme density jump cases. It must be noted that next results were obtained with the enrichment shape functions presented in 8.4, however, using 8.7, and condensing the elemental matrices, similar conclusions were reached. The reference solution used in Figure 10.6 was obtained using the formulation presented in 8.7 without condensing, which is equivalent to continuous enrichment. As was mentioned, that reference solution was previously validated in section

## 10.2

When both fluids have similar densities  $\sigma \sim 1$ , larger  $Fr$  values are obtained. In Figure 10.6b, a comparison with  $\sigma = 0.8$  between different PFEM-2 formulations is presented. When no enrichment strategy is used, the solution presents a noisy behavior where, according to the level heights, a typical mass-loss appears, deteriorating the overall solution. When discontinuous enrichment is added (this is, condensing the equation system (8.9)) the solution is smoother but still excessively dissipative compared to the continuous enrichment formulation used as reference. Meaning that, when the discontinuous enrichment formulation is used, the assumption of avoiding the inter-elemental boundary terms in the right hand side, as shown in 8.10, is not correct and, consequently, a diffusive behavior appears. This conclusion is confirmed when discontinuous enrichment is used (but without weakening the divergence of the velocity) and a solution very similar to the continuous enrichment one is obtained (not presented in the figure due to being almost identical to the reference). By not integrating by parts, the divergence term requires imposing a pressure gradient on the boundaries: if small density jumps are used, the previous gradient value  $\nabla p^n$  could be imposed on the boundaries, although this approximation is not valid when there is a gravitational field, large density jumps and the fluid is not at rest.

For the other limit  $\sigma \ll 1$ , the case when  $\sigma = 0.0013$ , is presented in Figure 10.6a. The graphic shows that both the continuous and the discontinuous enrichment solutions present a very similar behavior. As before, when no enrichment is used, results become noisy, showing that a spurious velocity field appears close to the free surface when no improvements are introduced in this region. Discontinuous enrichment with a strong form of divergence of the velocity cannot be used in this case due to the wrong prediction of the pressure gradients on the boundaries which turns the simulation unstable.

In the current implementation, choosing the continuous enrichment formulation increases the total CPU time by 50% compared to the other discontinuous formulations, albeit with the great advantage of being applicable to a wide range of situations. If the discontinuous enrichment formulation is used, better computing times are obtained. Nonetheless, some particularities, such as integrating the velocity divergence term by parts or not, must be taken into account depending on the density jump of the problem.

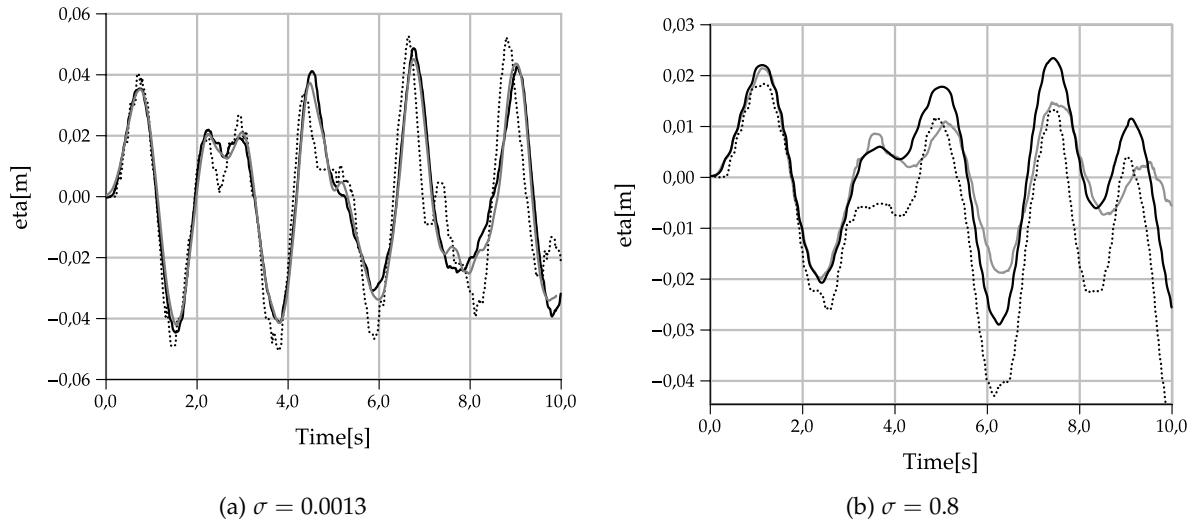


Figure 10.6: Level height on the left wall of a two phase flow for different density jumps. References: filled black line continuous enrichment, filled gray line discontinuous enrichment, and dotted line without enrichment.

### 10.3 Viscous standing waves

Computing the dissipation due to wave-breaking remains a challenging problem in the computational fluid mechanics context. In order to analyze the PFEM-2 solution when two-phase viscous incompressible flows are simulated, the evolution of a viscous standing wave has been chosen. An approximate analytical solution is available for small amplitude perturbations in the scientific literature [Lighthill, 2001] and it is of practical interest since it is related to the propagation of gravity waves.

The chosen standing wave configuration consists in a rectangular tank with length  $L$  and a water filling height of  $H = L/2$ . This setup has been extracted from [Colagrossi et al., 2013], for a sketch of this configuration see Figure 10.7. The wave length is  $\lambda = L$ ,  $k$  is the corresponding wave number (i.e.  $k = 2\pi/\lambda$ ) and  $A$  is the wave amplitude that defines the ratio  $\epsilon = 2A/H$ .

When the fluid is viscous, neglecting the wall boundary layer dissipation, and assuming small-amplitude waves (i.e small  $\epsilon$ ) and small wave steepness (i.e  $2A/\lambda \ll 0.1$ ); an approximate analytical solution of the standing wave evolution can be obtained through the

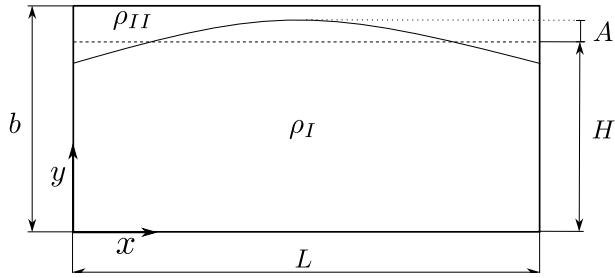


Figure 10.7: Configuration scheme of the standing wave case. Initial condition is represented by dotted lines. The continuous line presents an intermediate state where the maximum amplitude  $A$  is reached.

linearization of the Navier-Stokes equations for traveling waves:

$$\varphi(x, y, t) = \varphi_0(x, y) \cos(\omega t) \quad (10.1)$$

$$\varphi_0(x, y) = -\epsilon \frac{Hg}{2\omega} \frac{\cosh[k(y + H)]}{\cosh(kH)} \cos(kx) \quad (10.2)$$

where the circular frequency  $\omega$  is given by the dispersion relation of gravity waves, that is,  $\omega^2 = gk \tanh(kH)$  where  $g$  is the acceleration of gravity. At time  $t = 0$ , the free surface is horizontal while the initial fluid velocity is given by  $\varphi_0$ .

It can be demonstrated that the approximate solution is well posed only for  $Re \gg 1$  and for  $Re^{-1} \ll k \ll Re^{2/3}$ , where  $Re = H\sqrt{gH}/\nu$  is the Reynolds number for this problem. From that solution, it is possible to obtain the formula that gives the approximate decay of the kinetic energy [Lighthill, 2001]:

$$\varepsilon_K(t) = \epsilon^2 g \frac{\lambda H^2}{32} e^{-4\nu k^2 t} [1 + \cos(2\omega t)] \quad (10.3)$$

The kinetic attenuation is governed by the parameter  $\beta_l = 4\nu k^2$ , which depends on the wave number and on the kinematic viscosity  $\nu = \mu/\rho_I$ . Lately, work [Antuono and Colagrossi, 2013] has demonstrated that generally, Equation (10.3) overestimates the dissipations, especially when the Reynolds number is not very large. Then, an improved damping rate was proposed in [Antuono and Colagrossi, 2013]  $\beta = 4\nu k^2 - 2\sqrt{2}k^{11/4}Re^{-3/2} + O(Re^{-2})$ , which is used next for comparisons.

To accomplish the linear solution result, the PFEM-2 simulations have been implemented using a free-slip condition for the velocity and a Neumann condition for the pressure on each tank boundary. Also, the parameters  $L = 2$ ,  $A = 0.05$  and  $g = 1$  have been selected.

Several Reynolds number ( $Re = 25, 50, 250, 2500$ ) have been chosen to compare with the approximate analytic dissipation. Problems were solved using a grid size  $\Delta x$  such that  $H/\Delta x = 100$  and varying  $\Delta t$  in order to solve using a Fourier number  $Fo = \frac{v\Delta t}{\Delta x^2}$  with a local maximum of  $Fo_{max} \approx 10 - 50$ . Figure 10.8 shows the comparison between the expected energy dissipation (which includes the improved damping rate) and numerical results for the kinetic energy calculated with PFEM-2. Large Fourier numbers were used in order to reduce the computation times required to complete the simulations, showing that the accuracy of the method is maintained with large CFL numbers, even when diffusion is considered.

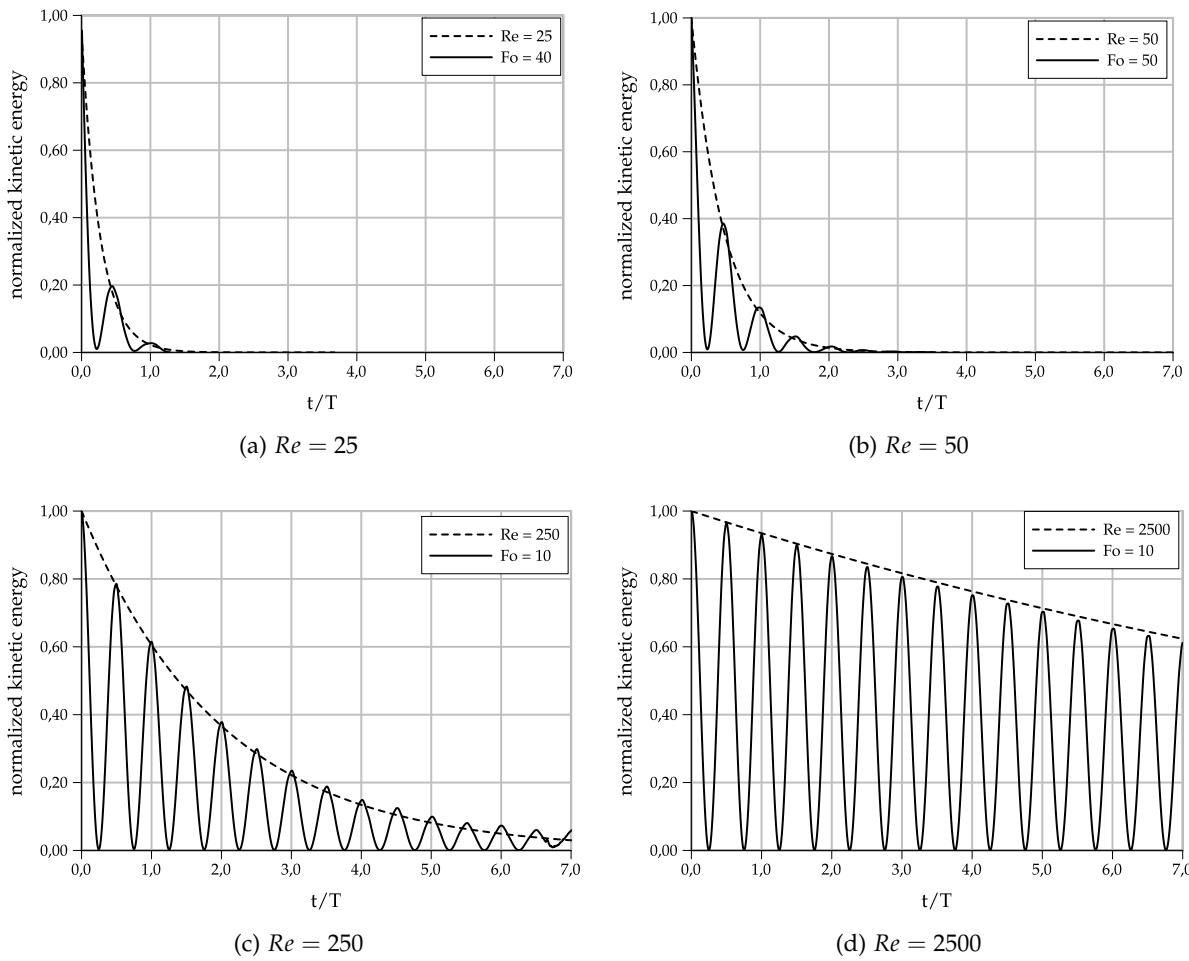


Figure 10.8: Kinetic Energy decay for standing wave problem with different Reynolds numbers. Dashed lines are approximate analytical solutions for total energy and continuous lines are the kinetic energy calculated with PFEM-2. Legend in each figure indicates the maximum Fourier number used in each numerical simulation.

Regarding the computational effort, it is noticeable that the complete set of PFEM-2

simulations only required a few hours on an Intel i7-2600k 3.4GHz processor. This is a large difference with respect to the CPU times reported in other works, see [Colagrossi et al., 2013], where a similar set of problems were solved with similar accuracy using SPH and a 30 CPUs cluster, each CPU with 8 Intel Xeon 2.33GHz cores, running continuously for several days. The mentioned difference of computing times can be associated with the strong limitation in the time-step size of the SPH method.

## 10.4 Dam-break problem

The objective of this section is to compare experimental measurements of a dam-break flow over a dry horizontal bed with the numerical approximation carried out with the PFEM-2 algorithm. The extensive set of experimental data is extracted from [Lobovský et al., 2014], where the dynamics of the dam break wave impacting a vertical wall downstream, with emphasis on the pressure loads and surface evolution after the dam burst, are presented.

A computational configuration of the tank used in experimental cases is presented in Figure 10.9, where the locations of water level measuring points and pressure sensors are shown. In this report, only the case with  $H = 300[\text{mm}]$  is analyzed. A two-phase non-viscous flow simulation is carried out, with  $\rho_{\text{water}} = 1000[\text{kg}/\text{m}^3]$ ,  $\rho_{\text{air}} = 1[\text{kg}/\text{m}^3]$  and gravity force  $\mathbf{g} = -10 \hat{j}[\text{m}/\text{s}^2]$ . The 2D computational grid used has  $322 \times 120$  nodes, conforming a mesh with around 80000 triangles. Boundary conditions are slip on all walls, and  $\Delta t$  is fixed to 0.1, which allows for  $CFL_{\max} \approx 20$  when the free surface impacts the downstream wall.

Figure 10.10 shows the comparison between experimental and numerical results for each water-level measurement. A good agreement can be observed, moreover when taking into account the capture of the back wave and splashing start events.

The impact pressure was measured with four sensors on the vertical wall at the end of the downstream flume, as described in Figure 10.9. The statistical analysis of the pressure peaks, rise times and the occurrence time, i.e. the time between the opening of the dam gate and the occurrence of the impact, are presented in Figure 10.11. The shown pressure  $P$  is non-dimensionalized with regards to the hydrostatic pressure at the bottom of the reservoir.

In the reference work, the analysis is focused on peak events. It can be noticed that the highest peak is recorded by sensor number 1 which is the sensor receiving the full impact, whilst the pressure of the other sensors is given by the run up of the flow. It can also be observed that sensor number 4, i.e. the sensor located at the highest position, does not show

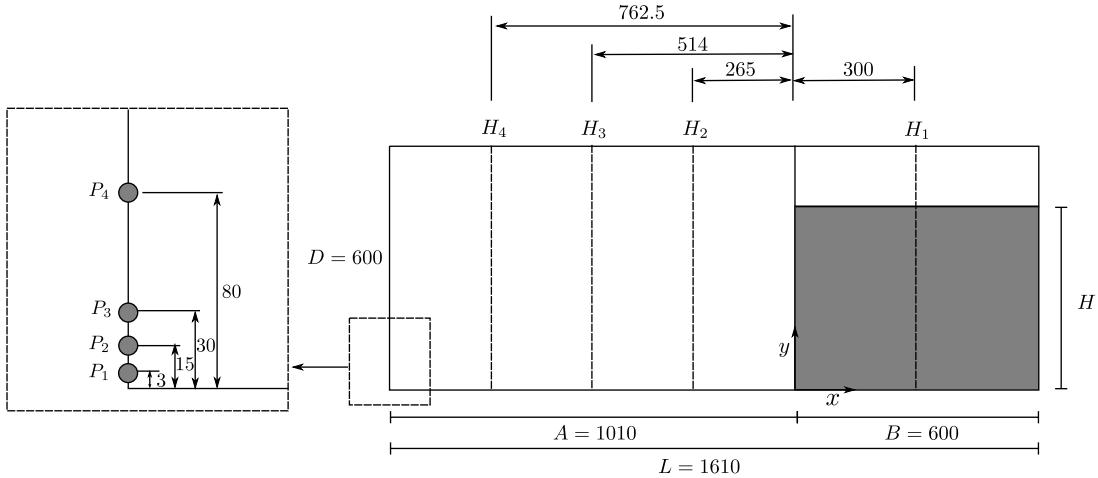


Figure 10.9: Configuration scheme of the dam-break case.  $H_1, H_2, H_3, H_4$  present the locations of water level measuring positions. Also,  $P_1, P_2, P_3, P_4$  show the locations of pressure sensors at the impact wall downstream from the dam. The grey zone represents the initial water condition. Dimensions are in millimeters.

a pure impact event, see Figure 10.11d, and the maximum for this sensor is actually obtained later in time, when the water falls back after running along the wall. Numerical solution behavior follows the mentioned conclusions, although the pressure values are not between the statistical limits of experimental data. Also, a discrepancy can be observed with the peaks arrival times for sensors 1 to 3. This difference can be assigned to the numerical simplification which does not model the gate movement. However, the pressure magnitude of the peaks is well predicted giving confidence to PFEM-2 calculations.

Finally, Figure 10.12 presents snapshots for the evolution of the simulated free-surface. Initial condition is shown in Figure 10.12a. Pressure peaks are related with the impact event observed in Figure 10.12c, which generates the back wave propagation that is displayed in the remaining figures.

#### 10.4.1 Three-dimensional simulation

Despite being a problem that can accurately be simulated in 2D, the same example was run in 3D to test the ability of the PFEM-2 solver to deal with larger geometries and 3D problems. The 3D mesh, which adds a third dimension of a thickness of  $0.15[m]$  with slip walls, has six million elements. These have an average  $h = 0.07$ , demanding more than 25 million particles for the fixed mesh approximation that fill the air and water domains. The same physical and

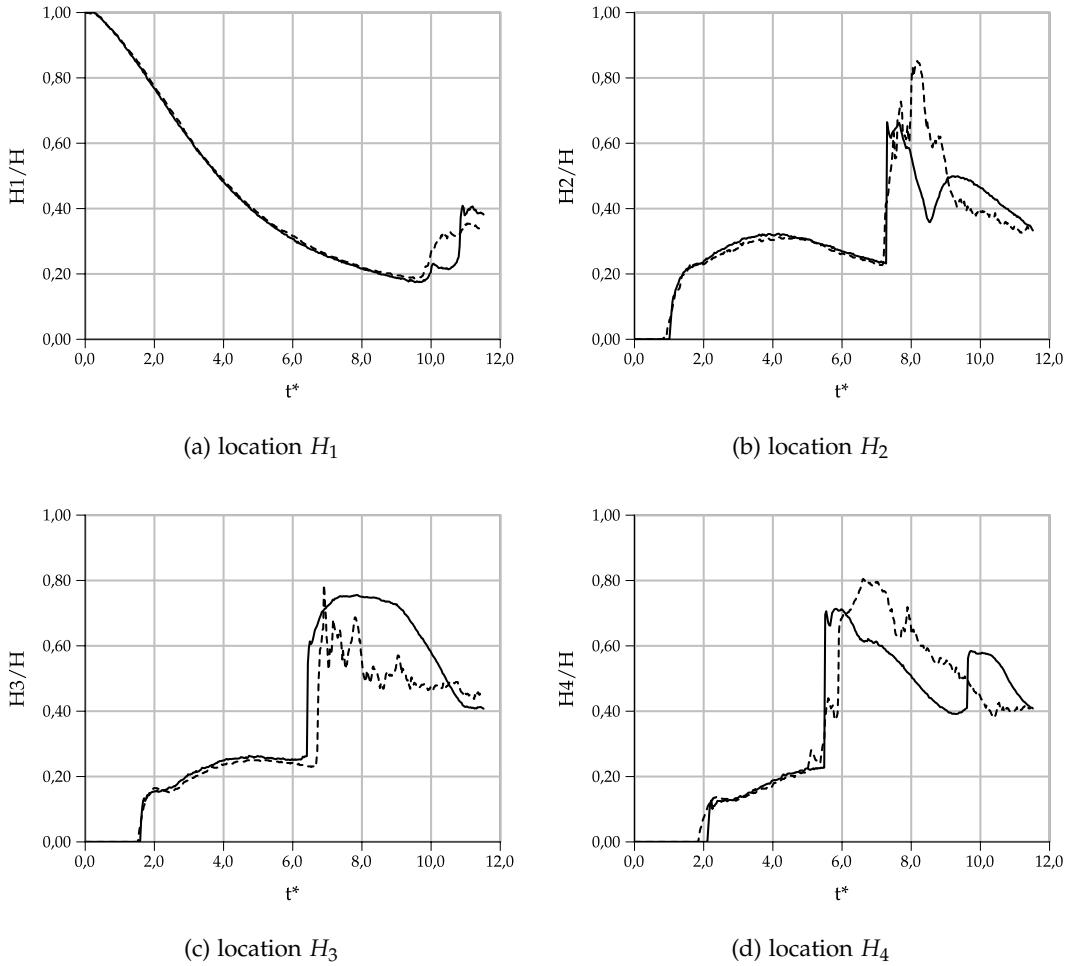


Figure 10.10: Water levels for tests with initial filling height  $H = 300[\text{mm}]$  compared to data from literature experimental results [Lobovský et al., 2014] (dashed lines) and numerical results with PFEM-2 (continuous lines). Time normalization is  $t^* = t(g/h)^{1/2}$ .

numerical parameters of the 2D simulation were used. Sensors are placed in the same position on the left wall and in the middle position of the third dimension, see Figure 10.9.

In this example, the time-step used, the element sizes and the velocity of the fluid lead to a simulation with some time-steps having a Courant number larger than 12, mainly when waves impact with walls. This shows once more, the capability of the method to manage 3D geometries with large time-steps. In Figure 10.13, the pressure history for each sensor is presented, showing a similar appearance to the two dimensional simulation.

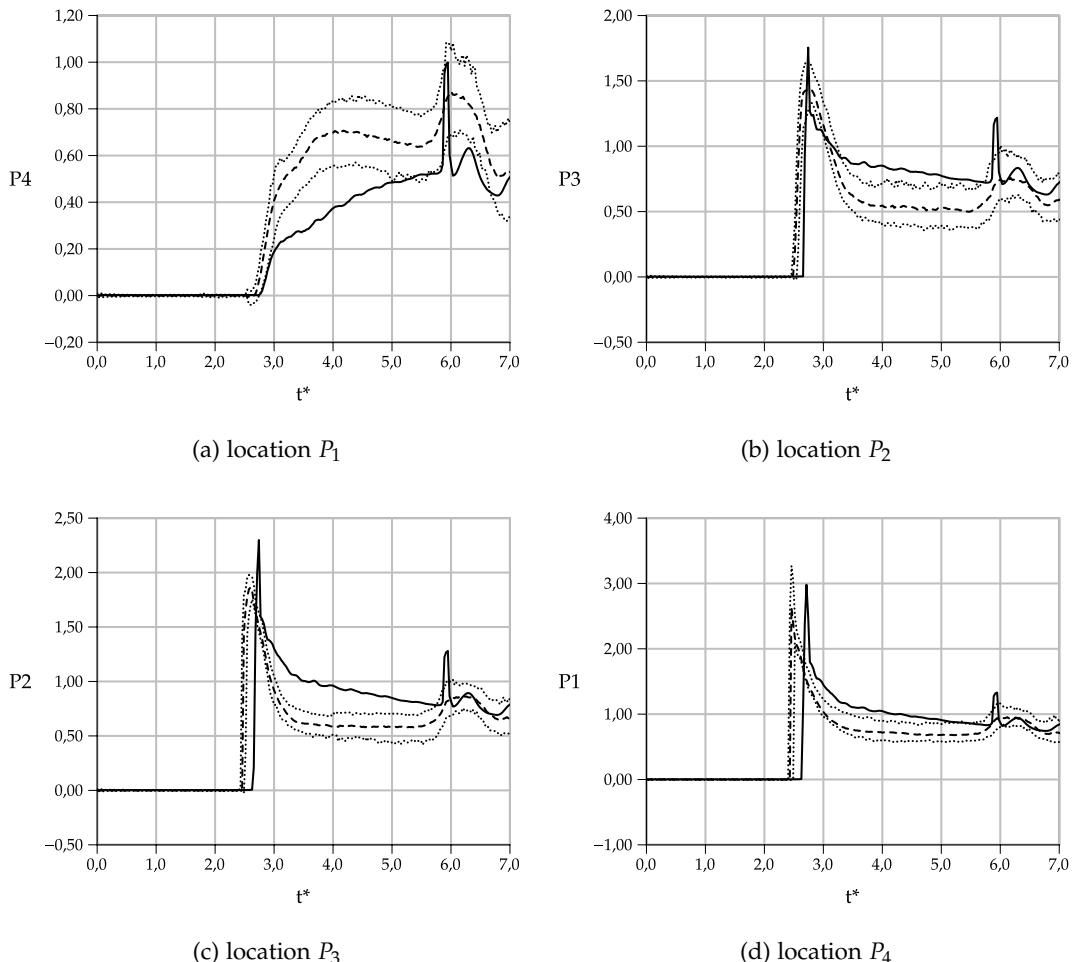


Figure 10.11: Pressure time histories comparison between experimental results [Lobovský et al., 2014] (discontinuous lines) and numerical results with PFEM-2 (continuous lines). Experimental results shows the median (dashed lines) and percentiles 2.5 and 97.5 (dotted lines).

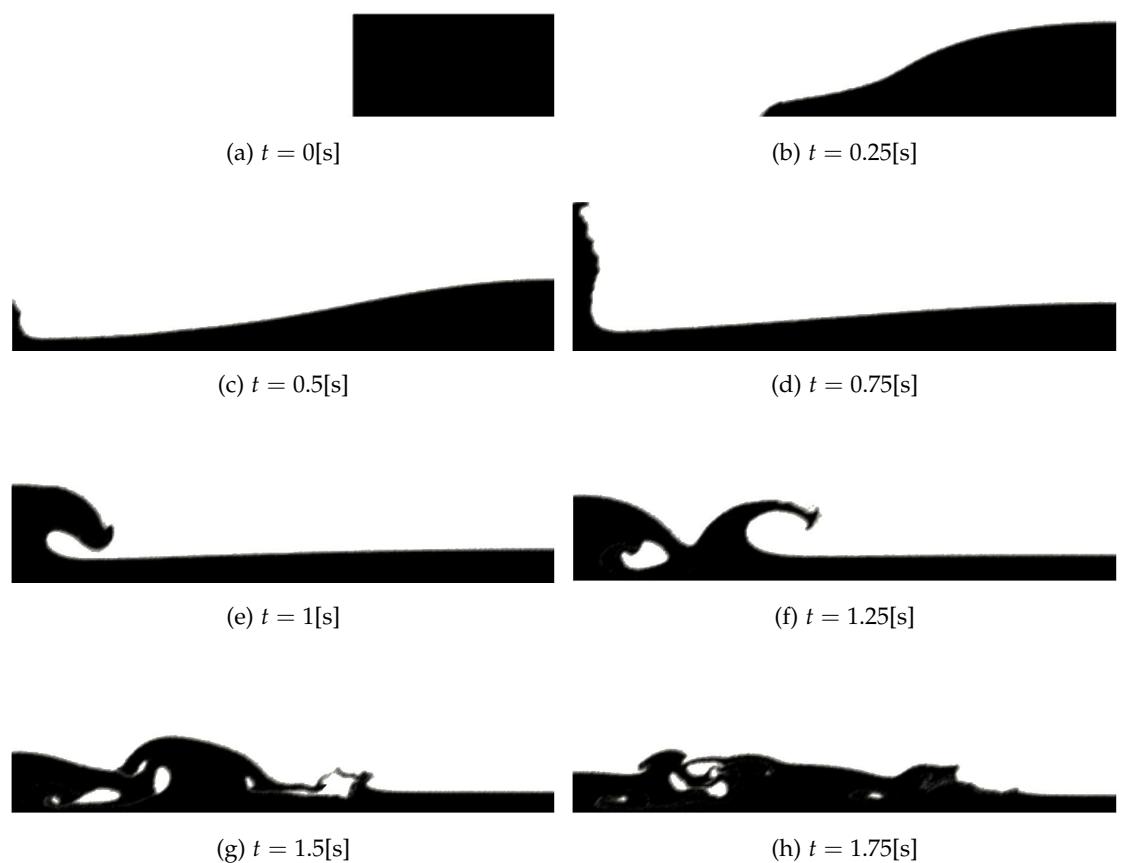


Figure 10.12: Snapshots of the two-dimensional simulation of dam-break.

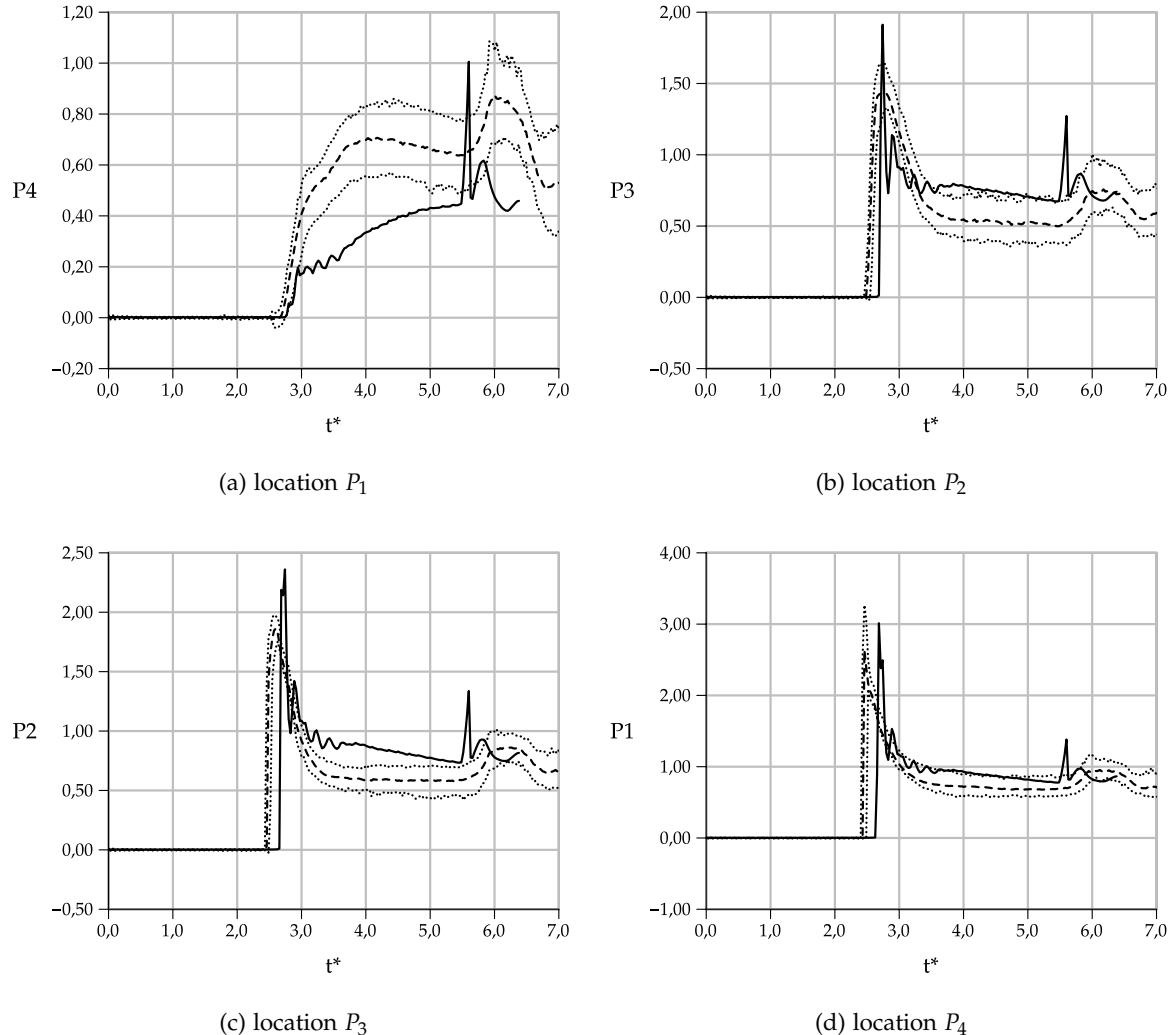


Figure 10.13: A pressure time history comparison between experimental results [Lobovský et al., 2014] (discontinuous lines) and numerical results with PFEM-2 in three dimensions (continuous lines). Experimental results shows the median (dashed lines) and percentiles 2.5 and 97.5 (dotted lines).



# Chapter 11

## Surface tension-dominant problems

The solution of problems where the surface tension force dictates the interface behavior is of different nature from the inertial dominant presented in Chapter 10. In this kind of problems is more relevant to compute accurately, and without noise, the curvature of the interface than its exact position and/or the improved treatment of jumps and kinks. Therefore, during the simulation the decision of using or not enrichment does not have as much influence in the results as the selection of a smooth or sharp marker function. As mentioned before, in order to reach an accurate curvature calculation (Equation (8.22)) is mandatory to use a smoothing strategy over the original marker function  $\lambda$ . Moreover, the integration rule should go along with this choice, employing then the Heaviside functions stated in Equation (8.26) to calculate fluid properties.

Regarding to numerical parameters,  $\theta_p$  is set to 0 in every case so as to restart the pressure at each time-step to allow larger time-steps and 3 iterations of steps 4 and 5 are done to improve the global first order. On the other hand, the parameter  $\theta_\mu$  is set to 0 except when the Fourier number  $Fo = \mu\Delta t/\Delta x$  is greater than 10 where is set to 1. The latter allows to increase the accuracy of the fractional step strategy for highly diffusive problems.

The preliminary case analyzes the stability of the surface tension modeling measuring the spurious currents and its dissipation level. Being the advection almost negligible, this case allows to show that the Eulerian parts of the algorithm works as other standard codes. Next case consists in a bubble which rises due to buoyancy force under two different regime, one more rigid where the surface tension is stronger and other more inertial where a skirted shape must be found. An analysis of the parasitic currents and mesh convergence is done for the case where the gravity is neglected. Moreover, the same cases simulated with larger time-steps are

stable but with more errors when surface tension increases, in contrast to Eulerian algorithms which tends to turn unstable. The second test is a standing wave dominated by capillarity. Although this problem is not the most indicated to be solved by PFEM-2 due to the lack of inertial dominance, the method shows good accuracy even though using reasonable large time-steps. Finally, a preliminary simulation of a primary atomization of a liquid jet is done. Being an inertia dominated case, large time-steps can be employed but a proper capture of drops and ligaments depends on the local Weber number.

## 11.1 Stability Analysis

The most critical numerical artifact introduced by the modeling of the surface tension is the generation of spurious currents which appear in the form of vortices around the interface. The employed method, CSF, is not excepted from that drawback. These flows, also named as parasitic currents, are generated solely due to numerical artifacts through the discrete approximation of the interface which acts as a perturbation on the physically smooth interface.

If the surface tension term is discretized explicitly, i.e. the surface tension forces are evaluated at the interface at the previous time-step, the stability of the scheme places a stability condition on a time-step [Brackbill et al., 1992] as

$$\Delta t < \sqrt{\frac{\rho \Delta x^3}{\gamma}} \quad (11.1)$$

which results in a limiting for fine meshes and large surface tension coefficients. The implicit treatment of surface tension terms is shown to alleviate this restriction [Hysing, 2006]. Instead of evaluating the surface tension with the interface at the previous time-step  $n$ , the PFEM-2 employs the interface at time  $n + 1$ . However, the interface movement is not coupled with the surface tension force calculation leading to a sort of semi-implicit scheme. Then, to evaluate the range of stability of this methodology, an analysis similar to that presented by Deshpande et al. [Deshpande et al., 2012] is done here.

In order to include the effect of viscosity in the case of low Reynold number, Galusinski and Vigneaux [Galusinski and Vigneaux, 2008] have revisited the time-step constrain leading to the following generalized time-step criterion

$$\Delta t \leq \tau_\gamma = \frac{1}{2} \left\{ C_2 \frac{\mu \Delta x}{\gamma} + \sqrt{\left( C_2 \frac{\mu \Delta x}{\gamma} \right)^2 + 4C_1 \frac{\rho \Delta x^3}{\gamma}} \right\} = \frac{1}{2} \left\{ C_2 \tau_\mu + \sqrt{(C_2 \tau_\mu)^2 + 4C_1 \tau_\rho^2} \right\} \quad (11.2)$$

with  $\tau_\mu$  and  $\tau_\rho$  two independent time-scales depending on the viscosity and density respectively. The constants  $C_1$  and  $C_2$  are independent of fluid properties and are only solver specific and in this work are determined experimentally from the simulations.

Following [Deshpande et al., 2012], the proposed case has a domain of  $1 \times 1$ [mm] discretized with a uniform grid of  $\Delta x = 10^{-3}$ m. Centered is a droplet of radius  $R = 0.25$ [mm]. Both density  $\rho$  and viscosity  $\mu$  are the same for the fluid inside or outside the drop and their value depend on the time scale considered. The coefficient of surface tension ( $\gamma = 0.01$ [N/m]) and simulation time-step ( $\Delta t = 10^{-4}$ [s]) are always maintained constant. Gravity is neglected in all simulations. The final simulation time is set to  $T_f = 10$ [s].

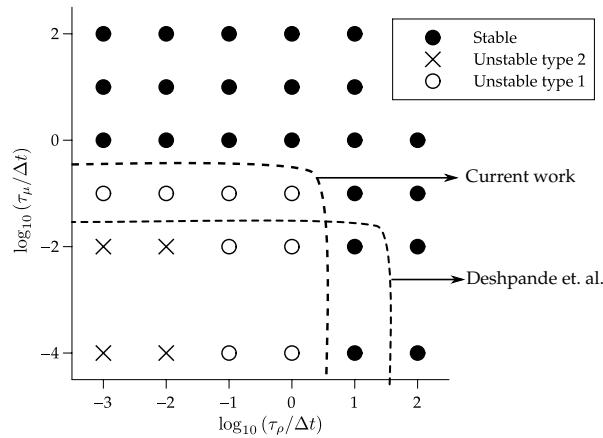


Figure 11.1: Stability chart for integration times  $T_f = 10$ [s]. Dashed lines represent the boundary between the stable and unstable computations found by Deshpande [Deshpande et al., 2012] and by the current work with PFEM-2.

The set of cases simulated covers the values of  $\tau_\rho/\Delta t$  and  $\tau_\mu/\Delta t$  desired varying  $\rho$  and  $\mu$  properly. The Figure 11.1 presents the stability charts for the behavior of each test. Three categories of simulations are taken into account:

- Stable: kinetic energy calculated over the entire domain decays and center of mass of the droplet remains fixed.
- Unstable type 1: simulation ends, but the center of mass of the droplet finishes displaced more than the size of one element of the mesh and/or the kinetic energy does not decay.
- Unstable type 2: simulation crashes after completion.

From the results, the constants  $C_1 = 0.1$  and  $C_2 = 1$  can be obtained. It should be noted that this set of simulations has an ideal number of Reynolds of  $Re = 0$  which is in the opposite

side of a proper application range of Lagrangian strategies. Then, which is actually tested is the performance of the fractional step method employed by PFEM-2 to couple velocity and pressure plus a constant projection/interpolation from/to particles. Comparing with the reference work of Deshpande et al. [Deshpande et al., 2012], PFEM-2 is more robust against the density time scale than FVM+VoF, but it is weaker regarding to viscosity time scale. The former is expectable because the density is related to the unsteady and inertia terms. The latter can be understood due to either excessive noise at interface because of the use of particles, which requires more viscosity to dissipate that phenomenon than Eulerian strategies and due to failures of the first order fractional step where the iterations do not recompose the solution. Anyway, it has been shown by Deshpande that the generation of spurious currents is only secondary for a moving interface, therefore the time-step analysis shown here represents a conservative estimate. It is expected when the convection takes part in the simulation, the advantages of PFEM-2 will be clearer.

## 11.2 Rising bubble case

A widely used surface-tension benchmark is the case of an air bubble rising in a liquid column. Beyond qualitative results, as the bubble shape, Hysing et. al. [Hysing et al., 2009] have presented a set of quantitative results obtained with several CFD multiphase codes solving two cases varying some physical properties. The first one considers a bubble in the ellipsoidal regime which undergoes moderate shape deformation, while in the second one the bubble belongs to the skirted regime and experiences much larger deformation. Both fluids are Newtonian, incompressible and isothermal, with properties listed in Table 11.1.

Test	$\rho_1$	$\rho_2$	$\mu_1$	$\mu_2$	$g$	$\gamma$	Re	Eo
1	1000	100	10	1	0.98	24.5	35	10
2	1000	1	10	0.1	0.98	1.96	35	125

Table 11.1: Physical parameters for the rising bubble case

In the Fig. 11.2 the case configuration and the boundary conditions are presented. The initial condition is null velocity with the phase marker imposed as shown. In comparison with other reported works, for example [Klostermann et al., 2013], where the initial condition had to be relaxed in order to smooth the interface between the two regions, with the current strategy this pre-processing is not necessary because the initial marker field is imposed over

particles then projected to the nodes, obtaining a naturally smoothed field over the mesh which diminishes, but not remove, the typical parasitic current of staircase profiles.

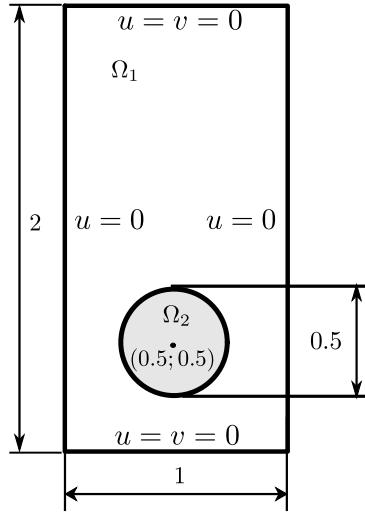


Figure 11.2: Rising bubble case configuration and boundary conditions.

The reference solutions presented in [Hysing et al., 2009] have been run with three different numerical approaches: the TP2D of Turek [Turek, 1998], the FreeLIFE of Parolini & Burman [Parolini and Burman, 2005], and the MooNMD of Ganesan et al. [Ganesan et al., 2007]. They all use the finite element method, but the two first approaches describe the interface with the level set, while the latter tracks it in an arbitrary Lagrangian-Eulerian way. In [Klostermann et al., 2013] Klostermann et al. validated the results of the open source library OpenFOAM®, which implements the finite volume method, and particularly for two-phase flows a VoF strategy with interface compression. The following bubble quantities are used to compare the results:

- Shape at the final time  $t = 3[\text{s}]$
- Center of mass:  $\mathbf{x}_c = \frac{\int_{\Omega_2} \mathbf{x} d\Omega_2}{\int_{\Omega_2} 1 d\Omega_2}$
- Rise velocity:  $\mathbf{u}_c = \frac{\int_{\Omega_2} \mathbf{u} d\Omega_2}{\int_{\Omega_2} 1 d\Omega_2}$

The computations have been performed on structured meshes divided in triangles with element sizes of  $h = 1/40, 1/80, 1/160$  (levels 1, 2 and 3 respectively), to reach the final simulation time ( $T_f = 3[\text{s}]$ ). During the first group of tests a grid size-dependent time-step of  $\Delta t = h/2$  is employed in order to calibrate the simulation to obtain similar results to the

reference. Once proved, the time-step is increased to analyze the stability and accuracy of the method when it is enforced.

### 11.2.1 Zero gravity condition

In order to estimate some first errors and uncertainties of the numerical model a transient simulation with surface tension but without gravity is carried out. The simulations were done up to reach  $T_{final} = 3[\text{s}]$ . The pressure jump over the droplet interface and parasitic velocities are analyzed in the simulation. The value of the pressure jump over the interface due to surface tension in two dimensions can be analytically calculated as

$$\Delta p = \frac{\gamma}{R} \quad (11.3)$$

where  $R = 0.25$  is the bubble radius. Using the physical conditions of the most surface-tension dominant case (Test 1), it leads to  $\Delta p = 98[\text{Pa}]$ . A normalized pressure  $P$  can be obtained in the case of a static bubble, it should read in  $\Omega_1$ :  $P = 0$  and in  $\Omega_2$ :  $P = 1$  with a sharp pressure jump at the surface.

Various numerical methods are known to generate spurious artificial numerical flows instead of keeping steady cylindrical drops [Strubelj et al., 2009]. The order of magnitude of parasitic velocities  $u_p$  can be estimated according to the surface tension coefficient  $\gamma$  and dynamic viscosity  $\mu$  of the bubble:

$$u_p = \frac{C_p \gamma}{\mu} \quad (11.4)$$

where  $C_p$  is a numerical constant, a characteristic of the quality of the numerical modeling of surface tension forces (a non-dimensional number similar to a capillary number). The optimal value of  $C_p$  is zero. Typical values of  $C_p$  are found between  $10^{-3}$  and  $10^{-10}$ .

A set of simulations were done employing different numerical strategies and various meshes in order to calculate the curvature and comparing results. The Table 11.2 presents the tests and its numerical results, while Figure 11.3 shows the final pressure field for some simulations. In the set of cases presented, spurious velocities are found on both sides of the interface, which are interpreted as parasitic currents. These observations are in agreement with those found in, for example, [Klostermann et al., 2013], [Strubelj et al., 2009] for a static viscous droplet in equilibrium. The current proposal of PFEM-2 with S-CLSOV + CSF was tested employing both curvature FEM calculation strategies above cited, i.e. using discontinuous and continuous normals (Equations 8.20 and 8.22) showing a clear advantage for the latter option in the parasitic current indicator, i.e.  $C_p$ , and similar results about the

pressure jump found  $P$ . On the other hand, the strategy that solves with VoF + interface compression + CSF is also included in the analysis. This approach, employed by the open source library OpenFOAM® and analyzed in [Klostermann et al., 2013], shows acceptable accuracy only when grids composed by quads are used. In the case of meshes of triangles, the curvature results are noisy leading to unphysical spikes (overshoots and undershoots) of the pressure and large spurious currents.

So as to completion, the Table 11.2 also presents the results obtained with PFEM-2+CSF but without smoothing, i.e. calculating the normal and curvature directly with the marker function  $\lambda$ . As expected, the spurious currents are one order above than the results with smoothing and the pressure jump is over estimated.

The employment of meshes composed by triangles leads to more efficient implementations of the PFEM-2 method, therefore a solution for the curvature in this type of meshes is essential so as not to resign performance. The results presented in this subsection guarantee accurate enough solutions for PFEM-2. However, one of the most important drawbacks of this strategy is that almost no grid convergence was found because the values of  $P$  and  $C_p$  remain almost constant even the mesh is refined. It is known that the integral effect of curvature (i.e. average pressure jump) actually converges to a value that is systematically different from the analytical value [Deshpande et al., 2012].

Method	$1/h$	$P$	$C_p$
PFEM-2 with $\kappa$ as Eq. 8.20	40	0.975	$1.9 \cdot 10^{-3}$
PFEM-2 with $\kappa$ as Eq. 8.22	40	0.95	$4.1 \cdot 10^{-4}$
PFEM-2 with $\kappa$ as Eq. 8.22	80	0.954	$5.3 \cdot 10^{-4}$
PFEM-2 with $\kappa$ as Eq. 8.22	160	0.955	$5.8 \cdot 10^{-4}$
VoF quads (ref [Klostermann et al., 2013])	40	0.83	$4.6 \cdot 10^{-4}$
VoF quads (ref [Klostermann et al., 2013])	320	0.7	$2.1 \cdot 10^{-4}$
VoF triangles	40	0.78	$6.7 \cdot 10^{-3}$
PFEM-2 with no smoothing	40	1.81	$3.0 \cdot 10^{-3}$

Table 11.2: Pressure jump and parasitic currents comparison between PFEM-2 and the simulations of [Klostermann et al., 2013].

As a footnote comment, in contrast with the set of simulations presented in [Klostermann et al., 2013], in our case we did not find the noisy behavior of

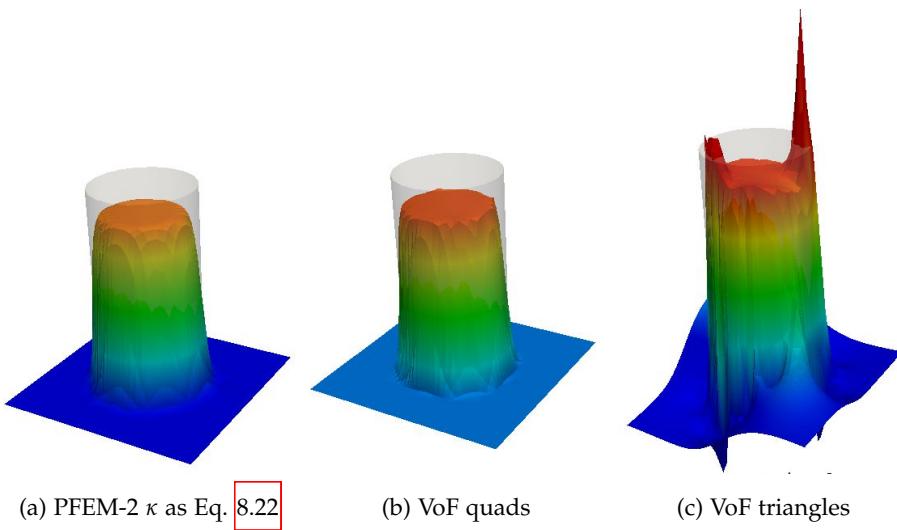


Figure 11.3: Zero Gravity test with a mesh of  $1/h = 40$ . Analytical pressure is shown in grey.

OpenFOAM® with the quads mesh when finer meshes are used. However, as in that publication, there is not found grid convergence of the method.

### 11.2.2 Ellipsoidal regime test

For the Test 1, Figure 11.4b shows the PFEM-2 bubble shapes at final time  $T_f$  for the meshes  $h = 1/40, 1/80, 1/160$ , the convergence to the shape of the finest mesh can be observed, which is in good agreement with the OpenFOAM® solution reported in [Klostermann et al., 2013] as shown in Figure 11.4a. PFEM-2 shape is less similar to FreeLIFE solution, but keeps good agreement. The plots of the bubble rise velocity in Figure 11.4c show that our bubble reaches a slightly larger maximum, but the evolution of the center of mass in Figure 11.4d is again in good agreement.

### 11.2.3 Skirted regime test

The same type of results are shown for test 2 in Figures 11.5a to 11.5d. Although the bubbles in both test cases rise with similar velocity, the decrease in surface tension as well as higher viscosity and density ratios causes bubble 2 to undergo a much larger deformation and to develop thin filaments. In both FreeLIFE and OpenFOAM® solutions these filaments break up, which also happens in PFEM-2 simulation (Figure 11.5c). In the physical reality, breakup occurs due to capillary waves present on the interface, which trigger the three-dimensional

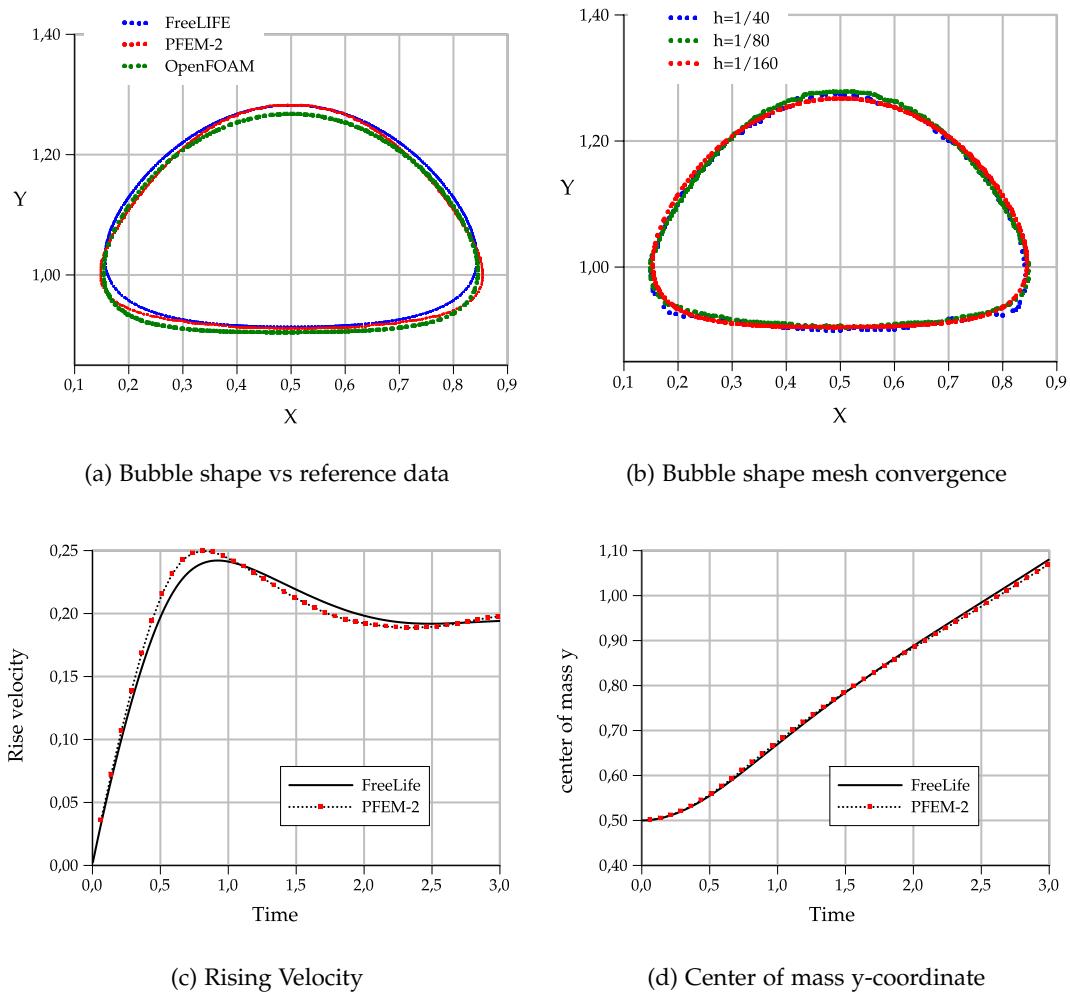


Figure 11.4: Rising bubble Test 1. Comparison of benchmark quantities: PFEM-2 vs. FreeLIFE and OpenFOAM®results. Mesh size  $h = 1/160$ , excepting in the mesh convergence.

Plateau-Rayleigh instability when the filament radius is small enough. Thus, capillary waves can cause the skirt filament to fragment during flow, though this response requires very large elongations, typically greater than 20 times the initial bubble radius [De Mier Torrecilla, 2009]. The Figure 11.5b shows that the PFEM-2 solution converges to the shape of the finest mesh, mainly the size of the two bubbles detached from the filaments (the coarser mesh is employed the larger unphysical satellite bubbles are obtained). The problem here is the use of the interface thickness parameter  $\epsilon$  which is mesh-dependent and introduce several distortions in coarser ones.

#### 11.2.4 Extending time-step

In order to emphasize the capability of the method to manage large time-steps, the current case is also simulated with a range of  $\Delta t$  using the in-house implementation of PFEM-2 and comparing with results obtained by the widely known OpenFOAM® suite which implement, as it was mentioned, VoF with interface compression. The problem setup and domain discretization is the same as presented above. In the case of OpenFOAM®, the solver and the configuration used in [Klostermann et al., 2013] is used in this subsection, which ensured good results in the rising bubble case. Compression flux treatment, time schemes and momentum predictor employment are analyzed in the mentioned work, deriving a recommended solver configuration for this case. The time-step employed is  $\Delta t = 3h$ , which enforce to obtain  $CFL = |\mathbf{u}|\Delta t/\Delta x > 1$  number that is critical for Eulerian framework solvers, mainly when it is measured at the interface.

Figure 11.6 presents PFEM-2 solutions with  $\Delta t = h/2$ ,  $3h$  and  $6h$  solving the most surface-tension dominant case, i.e. Test 1. Although the solution is stable for each time-step, the higher surface tension relevance respect to Test 2 generates non accurate solutions in the interface zone: unphysical disturbances like Rayleigh-Taylor instabilities are observed showing that the surface tension term is not imposed properly when the largest time-step is used. However, the solution with  $\Delta t = 3h$  is good enough and can be used as an accurate initial appearance of the solution. This preliminary and fast solution can not be done with OpenFOAM®, because the solution diverges when  $\Delta t > h$  is employed, due to the strong interface compression imposition.

On the other hand, Figure 11.7 the solutions obtained at  $T_f = 3[s]$  for the Test 2 can be shown. PFEM-2 solution when the time-step is increased is stable and keeps similar shape and quantitative values as rise velocity and center of mass, but loosing some definition of

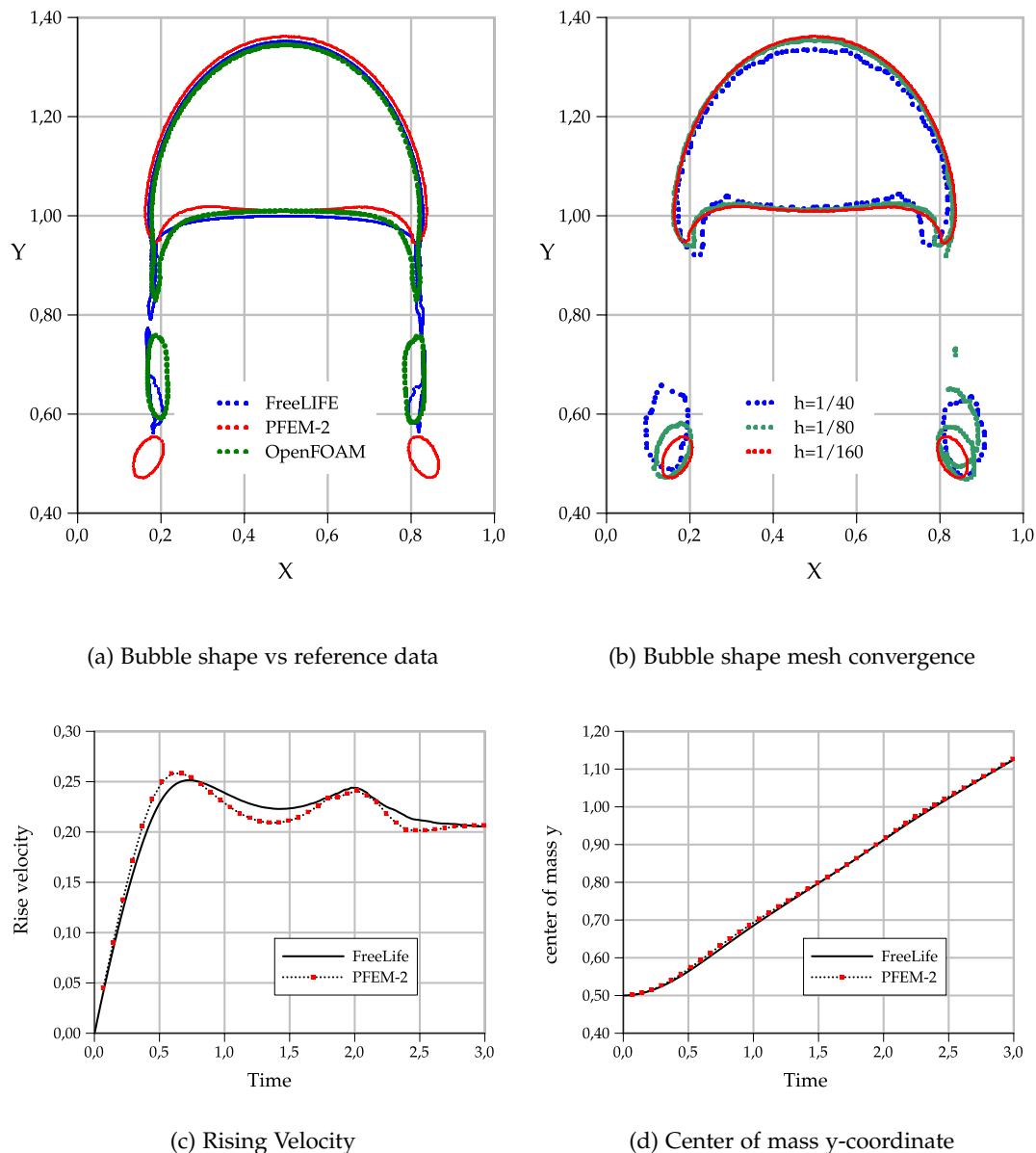


Figure 11.5: Rising bubble Test 2. Comparison of benchmark quantities: PFEM-2 vs. FreeLIFE and OpenFOAM® results. Mesh size  $h = 1/160$ , excepting in the mesh convergence.

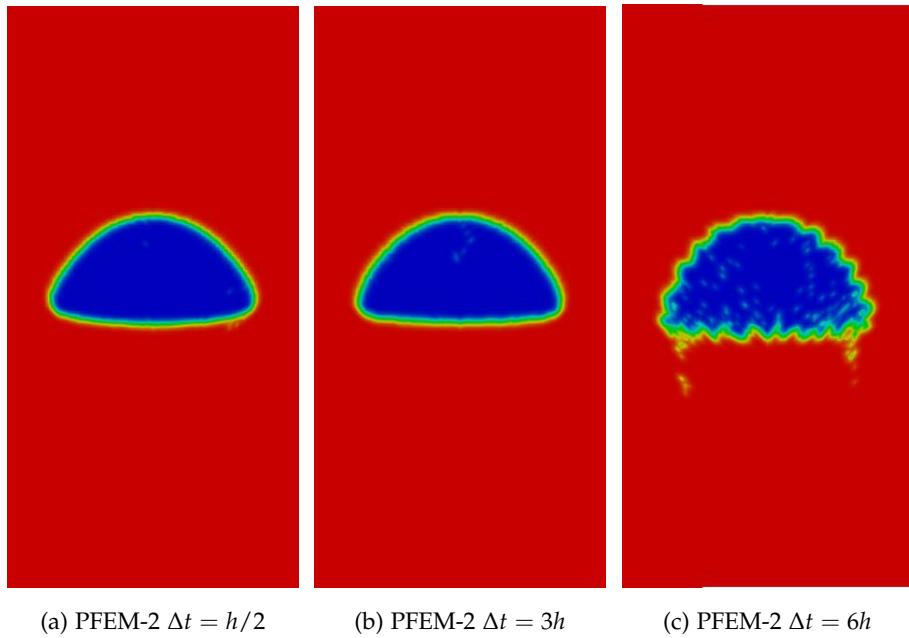


Figure 11.6: Rising bubble Test 1. Comparison of PFEM-2 solutions when the time-step is increased.

the satellite bubbles. On the other hand, OpenFOAM®(OF) solution with large time-step diverges approximately at  $t = 1.1[\text{s}]$  because of the disturbance introduced by the interface compression term trying to force a sharp interface. Reducing the interface compression coefficient could preserve the stability, but the final shape is highly diffusive, as presented by [Klostermann et al., 2013].

Figure 11.8 shows an application of the stability analysis to the case of the rising bubble. Same axis are presented and in this case the simulations presented correspond to the ellipsoidal and skirted regimes with the three time-step employed. As it was mentioned before, the stability limits are a conservative estimation due the generation of spurious currents is only secondary for a moving interface. The simulated cases prove this fact: although every test fall into the unstable region (see Figure 11.1), when convective term is included these numerical artifacts, which are not dissipated in unstable type 1 simulations, does not produce large errors in the results. Therefore, a stronger limit is used here which separates divergent (crashing) and non-divergent (no crashing) simulations.

It should be noted that the region of divergence of PFEM-2 is smaller than the obtained with OpenFOAM®. Therefore, almost every rising bubble simulation falls over the non-divergent region in PFEM-2, but only those that use the smallest time-step do not

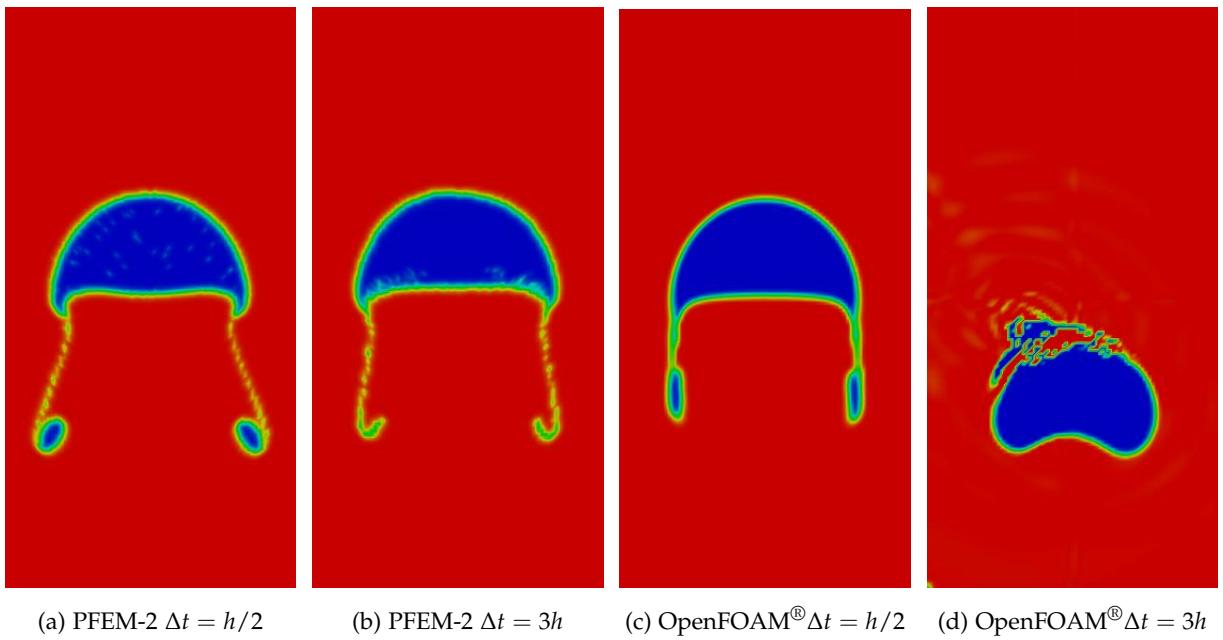


Figure 11.7: Rising bubble Test 2. Comparison of solutions when the time-step is increased.

blow-up with OpenFOAM<sup>®</sup>. This results is proven experimentally in the cases presented above. Although both large time-step tests theoretically are almost into the unstable region in PFEM-2, only the ellipsoidal test is experimentally unstable. The inclusion of new tests around that region could improve the determination of this limit.

### 11.3 Standing capillary wave

In Section 10.3 the results of the PFEM-2 method solving the case of the standing gravity wave were presented. In those simulations, several number of Reynolds were analyzed obtaining good agreement with analytical solutions. The flow were dominated by the gravity force, therefore the enrichment strategy to capture properly the density jump (the pressure gradient) was mandatory.

In the current work, the case to analyze is governed by a totally different force but leading to similar results. In this test the density jump is secondary, instead a good resolution of the surface tension forces, which dominate the flow behavior, is of transcendental relevance.

The setup used is taken from [Deshpande et al., 2012] in this simulation is shown in Figure 11.9. A perturbation of amplitude  $A = \lambda/20$  where  $\lambda$  is the wavelength width is imposed as initial condition of the phases positioning, which is allowed to evolve under the influence of

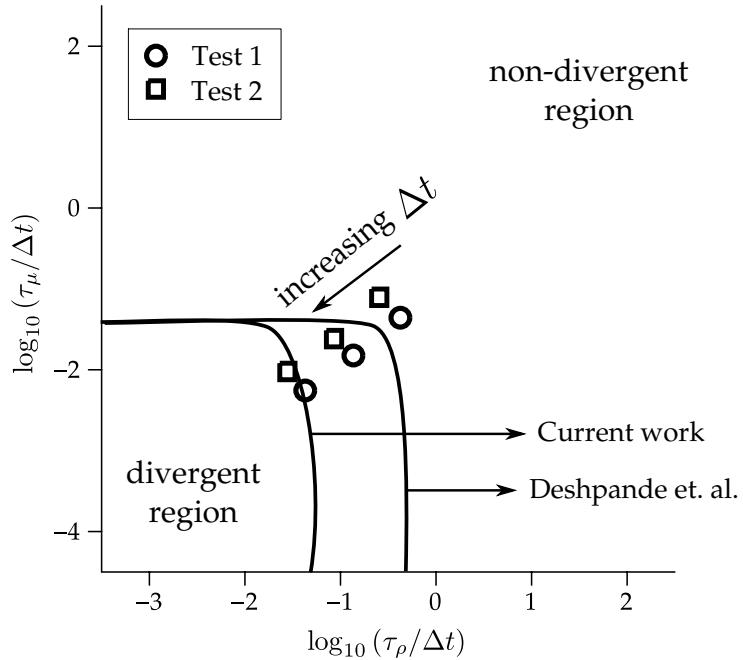


Figure 11.8: Rising bubble simulation in the stability chart. Filled line represent the boundary between the non divergent and divergent computations (unstable type 2 simulations) found by Deshpande [Deshpande et al., 2012] and by the current work with PFEM-2. Points represent the placement of tests ellipsoidal and skirted employing different time-steps.

surface tension alone. The heavier phase (fluid  $I$ ) has a density of  $\rho_I = 1000[\text{kg}/\text{m}^3]$  and a kinematic viscosity of  $\nu_I = 10^{-6}[\text{m}^2/\text{s}]$ , while the lighter phase has  $\rho_{II} = 1[\text{kg}/\text{m}^3]$  and  $\nu_{II} = 0[\text{m}^2/\text{s}]$  respectively. Regarding to boundary conditions, bottom is set as slip, left and right sides have a symmetry conditions and the top is considered as atmosphere fixing the pressure as  $p = 0$ .

Lamb [Lamb, 1993] presented an analytical solution of this problem when small-amplitude waves are considered. In this regime, the standing wave evolution can be obtained through the linearization of the Navier-Stokes equations for traveling waves. Therefore, the frequency of oscillation in this linear limit is

$$\omega = \sqrt{\frac{\gamma\kappa^3}{\rho_I + \rho_{II}}} \quad (11.5)$$

where  $\kappa = 2\pi/\lambda$ . Analytically, the frequency of change in kinetic energy  $K$  is twice the frequency of oscillation of the free surface. The analytical period of oscillation of kinetic energy is therefore  $\tau_A = 1/2(2\pi/\omega) = 2.385 \times 10^{-5}[\text{s}]$ . In addition, the rate of decay of

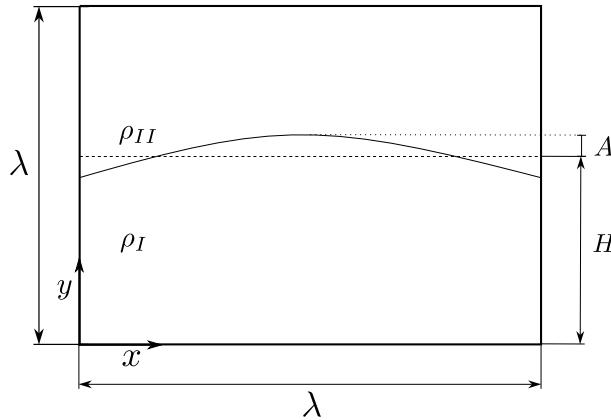


Figure 11.9: Configuration setup of standing capillary wave case

kinetic energy due to viscous effects is given as

$$\frac{K(t)}{K(0)} = e^{-4\nu^* t^*} \quad (11.6)$$

where  $\nu^* = \nu_I \kappa^2 \sqrt{\rho_I / (\kappa^3 \gamma)}$  and  $t^* = t / \sqrt{\rho_I / (\kappa^3 \gamma)}$ .

Three different grid densities were tested with  $\lambda / \Delta x = 40, 80$  and  $160$ . In order to obtain a more accurate dissipative forces calculation, the numerical parameter  $\theta_\mu$  is set to 1. This selection introduces a diffusion term in the equation of the corrector step as proposed Blasco, Codina and Huerta [Blasco et al., 1998]. Under our experience, without that term would not be possible to obtain the proper decay of kinetic energy with the classical fractional step method used due to the large Fourier number involved in this case.

The evolution of kinetic energy for these grids is plotted in Figure 11.10 along with the exponential decay of kinetic energy due to viscous effects, calculated from Equation (11.6).

Name	$\lambda / \Delta x$	$\Delta t [\text{s}]$	$\tau_A [\text{s}]$ calculated
Test A	40	$5 \cdot 10^{-7}$	$2.56 \cdot 10^{-5}$
Test B	80	$2.5 \cdot 10^{-7}$	$2.49 \cdot 10^{-5}$
Test C	160	$6.25 \cdot 10^{-8}$	$2.47 \cdot 10^{-5}$

Table 11.3: Errors of the computed period of kinetic energy for different grids. Case: standing capillary wave.

The numerical parameters employed in each test with its corresponding computed oscillation periods, averaged over 8 cycles, are shown in Table 11.3. The period of oscillation for the coarsest grid has the largest error and it is reduced employing finer meshes, achieving

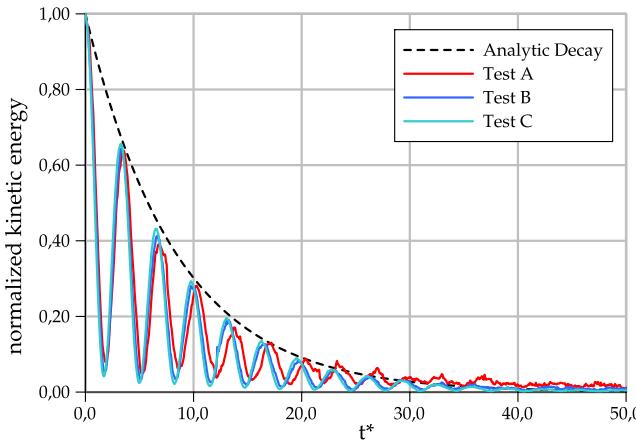


Figure 11.10: Comparison of the period of kinetic energy for the configuration in Figure 11.9, with the analytical solution of [Lamb, 1993].

an error in period with respect to the analytical solution of 3% with the finer grid. The last shows that, in spite of the inaccuracies in curvature, the results show trends of convergence to a value close to the theory. The observed loss in rate of convergence can be explained as a combination of systematic error in curvature, as it was shown in the zero gravity test, and also the fact that the analytical solution also contains a systematic error, since it is based on the linearized version of the equations, whereas PFEM-2 solves the full version of the Navier-Stokes equations.

## 11.4 Towards a simulation of 3D jet atomization

Liquid atomization is an important process which found interest in several engineering applications such as aerospace propulsion systems, automotive engines, food processing, and ink-jet printing. Its numerical simulation allows to investigate physical processes of the atomization because our understanding on physical mechanisms of such phenomenon is still not sufficient. Our investigation group is doing its first steps in this research area and we report in this work our early results using the numerical method presented in this work contrasted with the use of the widely validated tool OpenFOAM® [Weller et al., 1998].

The main properties of the case analyzed are the following: the size of the domain is (2.1[mm], 0.3[mm], 0.3[mm]), where the first dimension is the streamwise direction and the other two, the spanwise directions. At the injection level, the jet diameter  $D$  is equal to 0.1[mm], while the liquid jet Reynolds number is equal to  $Re = 4659$ . A summary of the

physical parameters, for this configuration, can be found in Table 11.4. Also, the geometry and boundary conditions are presented in Figure 11.11. Boundary condition over *borders* is slip, over *bottom* zero gradient velocity and pressure equal to zero. Top boundary has two patches: on *inlet* a mean inlet of  $\hat{U}_z = -100[\text{m/s}]$  is imposed and over *wall* no-slip condition is set.

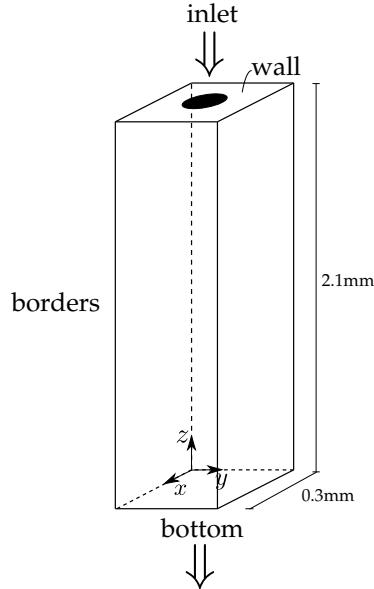


Figure 11.11: Geometry and boundary conditions for the case of the 3d jet.

As a first reference result, we can cite the work of Ménard et al. [Ménard et al., 2007; Chesnel et al., 2011], which employ the LSM to track the interface added to the Ghost Fluid Method (GFM) to describe the interface discontinuities and manage the pressure, density and viscosity jumps. Also, the Level Set method is coupled with the Volume of Fluid method (VoF) to ensure mass conservation. The mesh used by Chesnel and Ménard and co-workers in [Chesnel et al., 2011] is a  $2048 \times 256 \times 256$  Cartesian grid with regularly spaced nodes ( $\Delta x = 1.17\mu\text{m}$ ). Liquid surface instabilities close to the injector are visible. Their deformation leads to the formation of ligaments and droplets of various sizes. At the end of the domain, the liquid core has almost disappeared and a dense spray of droplets leaves the computational domain. The key of the quickly drop production is the use of a space-time correlated turbulent flow at the inlet: Ménard uses a synthesized correlated turbulence with a method proposed by Klein et al. [Klein, 2003]. In the work of Desjardins et al. [Desjardins et al., 2008], authors employ a forerunner simulation to impose the inlet turbulent boundary condition, obtaining similar results to the above mentioned. Both works have a relevant conclusion: by the end of the computational domain, the liquid core has been fully disintegrated.

Parameter	Symbol / Unit	Value
Gas density	$\rho_g$ [kg/m <sup>3</sup> ]	25
Liquid density	$\rho_l$ [kg/m <sup>3</sup> ]	696
Gas viscosity	$\mu_g$ [kg/m s]	$1 \times 10^{-5}$
Liquid viscosity	$\mu_l$ [kg/m s]	$1.18 \times 10^{-3}$
Surface Tension Coefficient	$\gamma$ [N/m]	0.06
Injection Diameter	$D_0$ [ $\mu\text{m}$ ]	100
Liquid Reynolds	$Re_l$	4659
Liquid Weber	$We_l$	7239
Turbulent Intensity	$\overline{u'u'}/U^2$	0.05
Turbulent Scale	$L_t$ [m]	$0.1D_0$

Table 11.4: Simulation parameters.

Another approach in the numerical characterization of jet atomization is reported by Shinjo et al. in [Shinjo and Umemura, 2010, Shinjo and Umemura, 2011]. In this work, the authors report that the grid resolution used by Ménard was coarse for the chosen Reynolds and Weber numbers, so this was not a direct numerical simulation in a true sense: the produced ligaments and droplets did not exhibit smooth shapes or wave dynamics driven by surface tension, but the overall liquid jet motion was captured in that simulation. Shinjo solved with a mesh with 400 million of cells ( $\Delta x = 0.3$  [m]). In contrast to Ménard, the ligament drop is done far from the inlet, being the main responsible the plain velocity front imposed at the inlet by Shinjo instead of using a turbulent-induced flow [Trinh and Chen, 2005].

Our initial simulations using PFEM-2 and OpenFOAM® employ plain inlet, therefore more similarities with Shinjo results are found. It must be taken into account that in the most refined case simulated with OpenFOAM®, the geometry was meshed with a cartesian base grid of  $32 \times 32 \times 256$  but the solver employed, named `interDyMFoam`, works with adaptive refinement at interface reaching a minimum grid size of  $\Delta x \approx 1.15\mu\text{m}$ . On the other hand, PFEM simulation has an uniform mesh size of  $\Delta x \approx 2.75\mu\text{m}$  conforming a mesh with 24 millions of tetrahedra, but even far from the refinement degree used in reference works.

Figure 11.12 shows a comparison between the simulations. The picture shows that the droplet formation and the like-mushroom shape are comparable, but the minimum drop size is better described using a finer mesh. The great advantage of using PFEM-2 is when

the computing time is analyzed because simulation was done employing a Courant number at interface  $CFL_{int} \approx 10$  while OpenFOAM® crashes when  $CFL_{int} > 1$  was tried. As it was shown in the rising bubble case, the simulation of drops can not be accurate when these large time-steps are used, however the stability of the method allows to obtain a very approximate solution, mainly of the jet core, even spending 10% of time comparing with Eulerian methodologies.

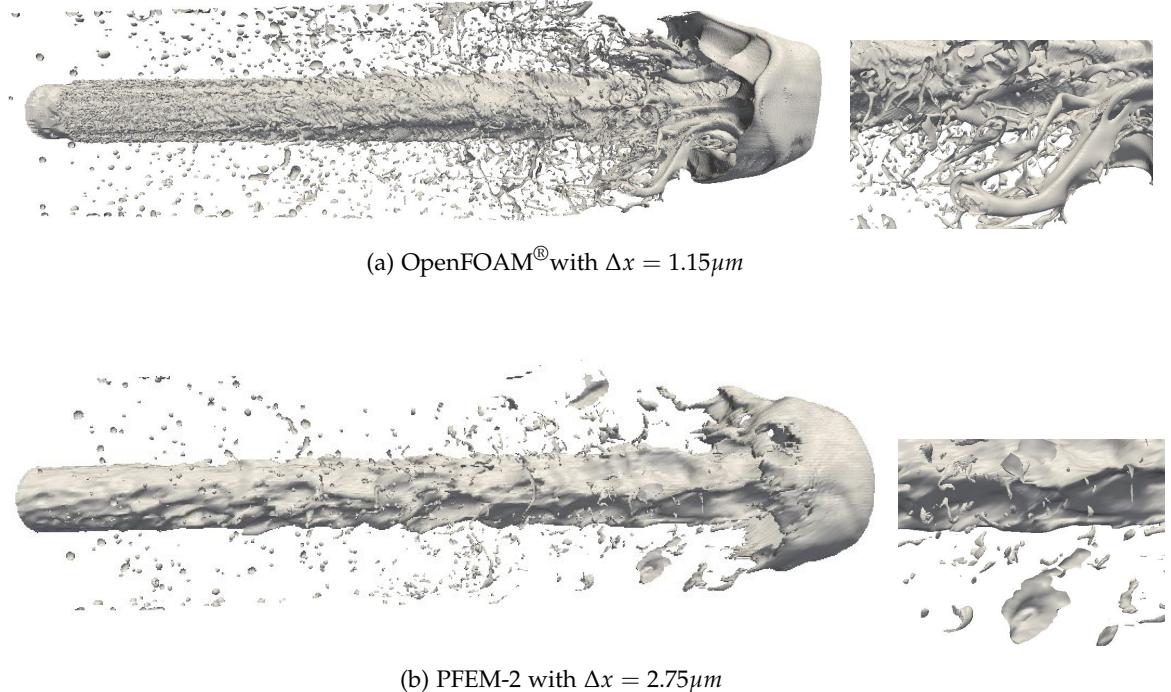


Figure 11.12: Overall shape of the liquid jet atomization. Figures correspond to iso-surfaces of  $\lambda = 0$  (interface). PFEM-2 simulation with  $CFL_{int} \approx 10$  and OpenFOAM® with  $CFL_{int} < 1$

Future works must enhance the simulation with PFEM-2, preferably employing finer meshes to contrast more adequately with reference works. An analysis of droplet size distribution is a relevant pending task which must be done in a future analysis.



# Chapter 12

## Conclusions

*And in the end,  
the love you take  
is equal to the love you make*

In this work the enhanced Particle Finite Element Method (PFEM-2) has been presented, extended and validated. This numerical method combines a fixed Eulerian mesh with a mobile cloud of particles and solves a material derivative formulation of the continuum equations through a derivation of the classical pressure-velocity segregation method named fractional-step. The methodology was successfully applied in the simulation of one-phase and two-phase incompressible flows. Employing a house-made high performing implementation developed during this thesis, notable advantages, regarding to the computational efficiency, were found comparing with pure Eulerian alternatives when solving problems dominated by convection. These advantages are based on the possibility of employing large time-steps because of the low error committed by the Lagrangian integration of the convective term when the X-IVS method is used.

In addition, a set of strategies to project the states of the particles to the mesh nodes were analyzed obtaining formulations with second order convergence. Employing these novel approaches for projection, a mathematical formula for the integration error was derived where is demonstrated that the error of the method obtains a second order of convergence as in the case of classical FEM approach. However, due to the error vanishes when the velocity is almost constant in time, this particularity confirms the large advantages of PFEM-2 solving these type of flows. Beyond the academic cases which validates the formulas, the conclusions were confirmed in a complex case from the oil industry obtaining accurate enough results

---

which are not deteriorated even using large time-steps.

In addition, this thesis was also focused on understanding the basic physical principles of two-phase flows and the difficulties that arise in their numerical simulation. The Lagrangian methods are specially well suited for tracking any kind of interfaces and this fact was exploited. Preserving the same base formulation of the one-phase algorithm, the extension to two-phase flows involved endowing the particles of new features. First, the necessity to transport a marker function to follow the interface and, second, switching off from following the streamlines when the particle has different density than the environment. Regarding to the problem on the mesh, a set of modern strategies were employed to satisfy the balance at interface. In the case of problems dominated by gravitational forces, a proper capture of the density jump at the interface, i.e. the pressure gradient jump, is required. Using enrichment this phenomenon is well captured, but it is found that condensing the new degrees of freedom can introduce extra dissipative behavior due to vanishing inter-elemental forces. In the case of surface-tension dominant problems, the main requirement is to find an accurate approximation to the curvature in order to impose properly the surface force term. Therefore, a smoothing of the marker function based on the level set reinitialization is employed giving impressive results even using unstructured meshes. Several tests were simulated grouping them into interface evolution flows, inertia dominant flows and surface tension flows. The results showed good agreement with experimental, analytical or numerical reference data, being relevant to highlight the advantages in terms of efficiency of PFEM-2 over pure Eulerian strategies when convective dominant flows were solved.

## 12.1 Published works

As a result of the work developed during this thesis, five original articles and two book chapters have been written:

- Idelsohn S., Nigro N., Gimenez J., Rossi R. and Marti J. (2013). *A fast and accurate method to solve the incompressible navier-stokes equations.* Engineering Computations, 30-Iss:2:197–222
- Gimenez J., Nigro N. and Idelsohn S. (2014). *Evaluating the performance of the particle finite element method in parallel architectures.* Journal of Computational Particle Mechanics, 1:103–116.

- Nigro, N., Gimenez J. and Idelsohn S. (2014). *Recent advances in the particle finite element method towards more complex fluid flow applications*. In Idelsohn, S. R., editor, Numerical Simulations of Coupled Problems in Engineering, volume 33 of Computational Methods in Applied Sciences, pages 267–318. Springer International Publishing.
- Gimenez J. and González L. (2015). *An extended validation of the last generation of particle finite element method for free surface flows*. Journal of Computational Physics, 284(0):186 – 205.
- Idelsohn S., Oñate E., Nigro N., Becker P. and Gimenez J. (2015). *Lagrangian versus eulerian integration errors*. Computer Methods in Applied Mechanics and Engineering, 293(0):191 - 206.
- Gimenez J., Morin P., Nigro N. and Idelsohn S. (2015) *Numerical comparison of the Particle Finite Element Method against an Eulerian formulation*. Book Chapter ASFI 2014, Modeling and Simulation in Science, Engineering and Technology Book Series. In press
- Gimenez J., Nigro N., Oñate E. and Idelsohn S. (2016). *Surface tension problems solved with the Particle Finite Element Method using large time-steps*. Computers & Fluids. Accepted. doi:10.1016/j.compfluid.2016.04.026.

Some parts of this PhD work have been done during the research visits at the Escuela Técnica Superior de Ingenieros Navales from the Universidad Politécnica de Madrid - ETSIN/UPM - (Madrid, Spain, September 2013 to March 2014) and the Centro Internacional de Métodos Numéricos en Ingeniería - CIMNE - (Barcelona, Spain, October 2013 and March 2014).

## 12.2 Open lines of research

The analysis presented in the case of the sloshing in a rectangular container has shown several drawbacks of the enrichment technique when the new degrees of freedom are condensed. Because using continuous enrichment is far from being efficient, the study of the inclusion of inter-elemental forces when the velocity divergence is weakened must be carried out.

On the other hand, it is desirable to obtain a uniform formulation to treat both surface tension and gravitational forces with the same confidence in a same simulation. A solution can be to extend the enrichment strategy to treat also the pressure jump dictated by the surface

tension. However, this approach is not compatible with the CSF method currently used to model the surface tension force. An alternative could be employing the Laplace-Beltrami formulation, but the requirement of obtaining every time-step a interfacial mesh should be relaxed in order not to compromise the efficiency of the method. If the previous is overcame, an implicit formulation for the treatment of the surface tension could be developed leading to a completely stable solver.

Currently, PFEM-2 is limited to employ meshes composed by simplices (triangles or tetrahedrons). It would be interesting to be capable of running with any type of meshes and particularly Cartesian grids, which could gives some advantages as reducing the number of elements, better definition of boundary layers and improved prediction of the curvature. The main drawback to solve is to integrate the particle trajectory fast, but it could be solve splitting locally the element which contains the particle into triangles and performing the integration as usual. The fact of being a local split avoids the necessity of synchronize the faces between neighbor elements. Remaining steps of the algorithm, i.e. the Eulerian ones, would be done over the original mesh.

A last open line of research is to implement the method in the package OpenFOAM®. This library has several algorithms, classes and methods to manage particles, then the implementation of the X-IVS strategy would be straightforward. Instead of employing FEM, the Eulerian stages would be solved by FVM because OpenFOAM® has an efficient machinery that implements this numerical framework. This approach would be named PFVM.

# **Appendices**



# Appendix A

## Resumen extendido de la Tesis

### A.1 Introducción

Durante los últimos 30 años, la simulación de problemas fluidodinámicos ha sido dominada por los métodos numéricos basados en la formulación Euleriana de las ecuaciones sobre dominios fijos [Donea and Huerta, 2003]. En este contexto los métodos Lagrangianos surgen como una alternativa a aquellos métodos tradicionales ya que utilizando este marco de referencia es posible resolver de forma natural simulaciones en donde la deformación del dominio no puede ser despreciada, esto sumado a que su formulación explícita los ubica en mejor posición respecto al rendimiento computacional utilizando las tecnologías hardware y principalmente el cómputo en paralelo. Pese a su ductilidad a la hora de simular superficies libres, interacción fluído-sólido o grandes deformaciones, los algoritmos Lagrangianos puros presentan sus propios problemas ya que requieren cómputos costosos que no son requeridos en el modelo Euleriano. En el caso de métodos sin malla deben manipularse varios millones de partículas lo que representa una tarea no trivial. En el caso de métodos con malla la tarea extra es construir una nueva malla a cada paso de tiempo que preserva una calidad aceptable para el cómputo.

Hoy en día los métodos híbridos ganan fuerza en el ámbito de la fluidodinámica computacional (CFD), en dónde los métodos de partículas Lagrangianos se combinan con metodologías Eulerianas. En estos métodos, las mallas fijas o reconstruidas soportan parte del cómputo de la presión y la velocidad. Como primeras ideas, la estrategia Lagrangiana original propuesta por Monaghan [Monaghan, 1988], ha sido completada con métodos de discretización Euleriana tales como el método de los volúmenes finitos (FVM) [Nestor et al., 2009], o elementos finitos (FEM) [Idelsohn et al., 2003a] en dónde una

triangulación de tipo Delaunay es utilizada para la reconstrucción de la malla a medida que el fluido evoluciona. Restringiendo a los elementos a utilizar funciones de forma lineales y alternando movimiento de partículas, mallado y solución implícita de sistema de ecuaciones, surge el método de elementos finitos y partículas (PFEM) [Idelsohn et al., 2004], el cual ha demostrado gran capacidad de resolución de problemas de superficie libre, rompimiento de olas, separación de flujo, etc. Una interesante diferencia entre PFEM y otros métodos híbridos como PIC (particle-in-cell) [Harlow, 1955], MAC (marker-and-cell) [Harlow and Welch, 1965] y MPM (material point method) [Wieckowsky, 2004] es que mientras en estos últimos las partículas transportan masa y consecuentemente tiene volumen, en PFEM los puntos son no materiales que transportan propiedades intensivas fijas del fluido. Metodologías como el método de las características (CM) [Allievi and Bermejo, 2000] también permiten el uso de derivadas materiales que eviten el cómputo del término no lineal convectivo, sin embargo su utilización resulta muy disipativa a menos que se utilicen polinomios de alto orden.

Con el fin de evitar la difusión producida por el método de las características, una nueva estrategia conocida como X-IVAS fue desarrollada por Idelsohn et.al. [Idelsohn et al., 2012]. Esta metodología de integración de la convección del fluido esta basada en seguir las líneas de corriente del paso de tiempo actual en vez de las trayectorias de las partículas tal como CM, siendo esto una alternativa novedosa para resolver las no linealidades de las ecuaciones de flujo. Incorporando esta estrategia al método PFEM original, aparece un nuevo método que recibe el nombre de PFEM-2 [Idelsohn et al., 2013b]. Esta estrategia dota al método de la posibilidad de resolver flujos complejos utilizando grandes pasos de tiempo y la utilización de mallas permite obtener soluciones precisas del método de pasos fraccionados [Gimenez, 2014].

## A.2 Flujos a una fase

Las ecuaciones de gobierno de la dinámica de fluidos y en particular para fluidos newtonianos incompresibles, llamadas ecuaciones de Navier-Stokes, pueden ser representadas utilizando la formulación Euleriana o la formulación Lagrangiana. Mientras en la primera se utiliza un marco de referencia fijo, la segunda utiliza un marco de referencia móvil con una partícula del fluido. Esto repercute a la hora de expresar las ecuaciones: mientras en el Euleriano aparece una derivada temporal de la velocidad sumada a un término convectivo no lineal, en el marco Lagrangiano aparece solamente una derivada material y la no-linealidad de la ecuación pasa a ser geométrica ya que se requiere la solución de un problema cinemático para la integración

de la trayectoria de las partículas.

Dentro de los métodos numéricos que resuelven utilizando el marco de referencia Euleriano pueden diferenciarse aquellos que resuelven un gran sistema de ecuaciones que incluya todas las variables involucradas, llamados *métodos monolíticos*, y aquellos esquemas que desacoplan la velocidad y la presión con el fin de resolver la misma problemática en una secuencia de pasos más simples, los cuales reciben el nombre de *métodos segregados*. Dentro de este último grupo una estrategia de referencia es el método propuesto por Chorin [Chorin, 1968] y Temam [Témam, 1969] denominado método de los pasos fraccionados (FSM).

Un paso de tiempo del método FSM consiste en primer lugar calcular una velocidad *predictora* la cual tiene en cuenta los efectos convectivos y difusivos y cumple con las condiciones de borde, en segundo lugar hallar una presión que garantice la incompresibilidad del flujo y en tercer lugar modificar la velocidad predictora en base al nuevo cálculo de la presión, es decir, proyectar la velocidad a un espacio de divergencia cero. Pese a ser un método simple, robusto y eficiente, el método FSM sufre de los típicos problemas de los métodos Eulerianos al resolver los términos convectivos, lo que repercute en grandes inconvenientes a la hora de resolver problemas de convección dominante:

- *Linearización*: El sistema de ecuaciones debe ser transformado a uno lineal, desacoplando la velocidad transportadora de la velocidad transportada.
- *Estabilización*: Utilizar aproximaciones de segundo orden para los términos convectivos en problemas donde la convección domina sobre los efectos viscosos requiere de la adición de técnicas de estabilización [Donea and Huerta, 2003, Tezduyar, 1992] la cual introduce difusión artificial que reduce la precisión del método.
- *Sistema no-simétrico*: matrices con baja tasa de convergencia y usualmente mal condicionadas, requiriendo un importante esfuerzo computacional para resolver el sistema asociado.
- *Problema hiperbólico*: Aunque se utilicen propiedades del fluido constantes, no es posible pre-factorizar el sistema como sí lo es en problemas elípticos [Gimenez et al., 2014].

Una manera de arribar a la formulación del método PFEM-2 es partir del esquema típico FSM, y reemplazar el tratamiento Euleriano estándar del término convectivo con una formulación Lagrangiana basada en partículas. Para ello se requiere de una malla Euleriana que permanece fija durante la simulación y de un conjunto de partículas que se advectan de

acuerdo al propio fluido. Finalmente, un paso de tiempo del algoritmo PFEM-2 consiste en cinco etapas:

1. Calcular la convección sobre las partículas.
2. Proyectar los estados desde las partículas a la malla.
3. Calcular la predicción de la velocidad velocidad en la malla.
4. Calcular la presión final en la malla.
5. Calcular la velocidad final en la malla proyectando la predicción de la velocidad sobre un espacio de divergencia nula. Luego interpolar la corrección sobre las partículas.

En notación formal, se asume que todas las variables son conocidas a tiempo  $t^n$  tanto para las partículas (posición  $\mathbf{x}_p$  y velocidad  $\mathbf{u}_p$ ) como para los nodos (velocidad  $\mathbf{u}_j$  y presión  $p_j$ ), donde los subíndices  $()_i$  y  $()_j$  representan un nodo de la malla genérico  $j$  mientras que el subíndice  $()_p$  una partícula genérica  $p$ . Además  $N$  son las funciones base lineales de elementos finitos. De acuerdo a esta notación, los pasos son:

---

**Algorithm 4** - Paso de tiempo en PFEM-2 para flujos incompresibles a una fase.

---

1. Etapa convectiva:

$$\begin{cases} \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{u}^n(\mathbf{x}_p^\tau) d\tau \\ \widehat{\mathbf{u}}_p^{n+1} = \mathbf{u}_p^n \end{cases}$$

2. Etapa de proyección:

$$\mathbf{M}_{ij} \widehat{\mathbf{u}}_j^{n+1} = \mathbf{M}_{ip} \widehat{\mathbf{u}}_p^{n+1}$$

3. Etapa de predicción de velocidad:

$$\left( \mathbf{M}\left(\frac{\rho}{\Delta t}\right) + \mathbf{K}(\mu) \right) \widehat{\mathbf{u}}^{n+1} = \mathbf{M}\left(\frac{\rho}{\Delta t}\right) \mathbf{u}^n - \mathbf{G} p^n$$

4. Etapa de presión:

$$\mathbf{K}\left(\frac{\Delta t}{\rho} + \tau\right) p^{n+1} = \mathbf{B} \widehat{\mathbf{u}}^{n+1} + \mathbf{K}\left(\frac{\Delta t}{\rho}\right) p^n + \mathbf{B}(\tau) \boldsymbol{\pi}^n$$

5. Etapa de corrección:

$$\begin{aligned} \mathbf{M}(\rho) \mathbf{u}^{n+1} &= \mathbf{M}(\rho) \widehat{\mathbf{u}}^{n+1} - \Delta t \mathbf{G} (p^{n+1} - p^n) \\ \rho_p \mathbf{u}_p^{n+1} &= \rho_p \widehat{\mathbf{u}}_p^{n+1} + \sum_j \delta \mathbf{u}_j^{n+1} N_j(\mathbf{x}_p^{n+1}) \end{aligned}$$


---

en dónde  $\delta p = p^{n+1} - p^n$  y  $\delta \mathbf{u} = \mathbf{u} - \widehat{\mathbf{u}}$ .

### A.2.1 Implementación en paralelo

El principal objetivo del método PFEM-2 es encontrar algoritmos para simular de forma precisa problemas de CFD tan rápido como sea posible. Esta metodología intenta reducir los tiempos de cálculo que hoy en día presentan tanto códigos comerciales como códigos abiertos en éste área. Esta meta no significa aún alcanzar simulaciones en tiempo real (Real Time CFD), pero es esperable cambiar días de simulación por horas o horas por minutos, haciendo posible satisfacer las actuales necesidades del diseño ingenieril. Para ello es vital contar con un código eficiente y que sea desarrollado utilizando las mejores características del software y hardware de hoy en día. Entonces, la estrategia de aumentar el tamaño de los pasos de tiempo de nada serviría si no se posee una implementación eficiente ni paralela.

El principal problema del método PFEM-2 radica en la avidez respecto al uso de memoria para el almacenamiento de malla y partículas en comparación con métodos que utilizan sólo malla. Esta limitación acota el tamaño máximo de problemas a resolver en una sola unidad de cómputo, por lo que es necesario pensar en el paralelismo de memoria distribuida inmediatamente. La tesis de maestría del autor [Gimenez, 2014] presenta la implementación del método sobre arquitecturas paralelas. En la presente tesis se retoman algunos conceptos claves tales como el paradigma de programación, el lenguaje y las librerías utilizadas. Particularmente la implementación paralela en memoria distribuida toma como base la librería libMesh [Kirk et al., 2006], la cual es una librería orientada a objetos escrita en C++ para resolver problemas FEM con refinamiento adaptativo. La comunicación entre los nodos de cómputo se realiza a través del standard MPI. Otras librerías que se utilizan son PETSc [Balay et al., 2012], para la solución de sistema de ecuaciones lineales, y METIS [Karypis and Kumar, 1999] y ParMETIS [Karypis and Kumar, 1998] que implementan estrategias de descomposición del dominio basadas en esquemas de particionamiento de grafos para mallas seriales o distribuidas, respectivamente.

La Figura 4.2 presenta el speedup  $S_n$  obtenido resolviendo un caso test estándar de flujo incompresible en tres dimensiones: el flujo alrededor de un cilindro a  $Re = 1000$ . El speedup se define como

$$S_n = \frac{T_1}{T_n}, \quad (\text{A.1})$$

dónde  $T_1$  y  $T_n$  son el total de tiempo de cómputo empleando 1 y  $n$  procesos respectivamente. Alrededor de dos millones de grados de libertad fueron utilizados y el total de tiempo simulado fue de 100[s]. El clúster empleado tiene nodos dual socket, con procesadores

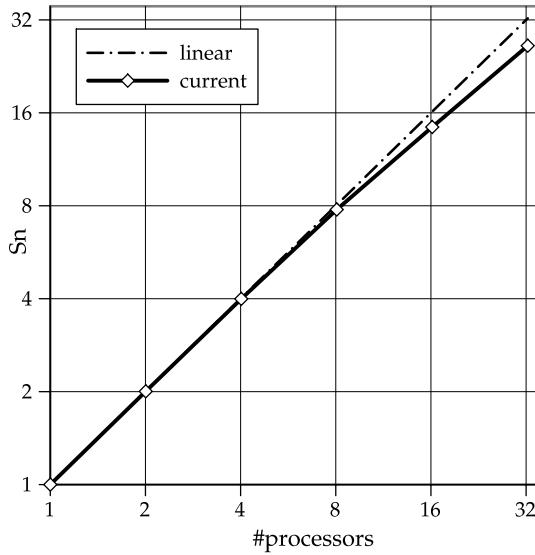


Figure A.1: Speed-Up over an Infiniband cluster solving the flow around a cylinder in 3d.

Intel Xeon E5-2600 y 64Gb de memoria RAM, interconectados con IB-QDR 40 Gbps. El speedup alcanzado por PFEM-2 es similar al obtenido por el software CFD contra el que se comparó (OpenFOAM®), sin embargo se logra un factor de reducción de los tiempos totales de  $3\times$  manteniendo el mismo nivel de precisión en los resultados gracias a la posibilidad de utilizar mayores pasos de tiempo. De esta manera, la implementación paralela demuestra un buen comportamiento al ejecutarse sobre un clúster dedicado al cómputo científico, logrando eficiencias de aproximadamente 80% con 32 procesos, lo que garantiza un buen punto de partida de un código masivamente paralelo.

## A.2.2 Análisis de errores de integración y pruebas

En la tesis se realiza una comparación entre los errores de integración, tanto temporal como espacial, de las metodologías PFEM-2 y FSM estándar. En primer lugar se analizan los errores en el caso de un problema de transporte escalar

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = Q(\mathbf{x}) \quad (\text{A.2})$$

donde  $T = T(\mathbf{x}, t)$  es la incógnita escalar,  $\mathbf{u}$  es la velocidad y  $Q$  incluye el término fuente  $q(\mathbf{x})$ , el término difusivo  $\nabla \cdot (k \nabla T)$ , el término reactivo  $cT$ , etc. Reemplazando con las discretizaciones empleadas por ambas metodologías, pueden derivarse las siguientes fórmulas

de error para una integración desde el tiempo  $t_0$  hasta  $t_f$ :

$$\epsilon^E = \pm \mathcal{O}([Q - \mathbf{u} \nabla T]'' \Delta x^2)(t_f - t_0) \pm \mathcal{O}([Q - \mathbf{u} \nabla T]'' \Delta t^2)(t_f - t_0) \quad (\text{A.3})$$

$$\epsilon^L = \pm \mathcal{O}((Q'' \pm \mathbf{u}'' \nabla T) \Delta x^2)(t_f - t_0) \pm \mathcal{O}((Q'' \pm \mathbf{u}'' \nabla T) \Delta t^2)(t_f - t_0) \pm \mathcal{O}(T'' \Delta x^2) \frac{(t_f - t_0)}{\Delta t} \quad (\text{A.4})$$

donde  $\epsilon^E$  es el error con el marco de referencia Euleriano,  $\epsilon^L$  es el error en el marco de referencia Lagrangiano, y se utilizan los símbolos  $''$  y  $''''$  para expresar las derivadas segundas respecto al espacio y al tiempo respectivamente. De estas expresiones puede concluirse que:

- Los esquemas Eulerianos son mejores para problemas de difusión dominante. En estos casos, los errores son del mismo orden, pero en el caso Lagrangiano deben agregarse los errores de proyección.
- Los esquemas Lagrangianos son mejores para problemas de convección dominante donde el flujo convectivo cumple que  $\mathbf{u}''' \approx \mathbf{u}'' \approx 0$ . En el resto de los casos, cuando el flujo convectivo presenta altas variaciones, alguno de los esquemas será mejor que el otro dependiendo de la influencia de los errores de proyección del Lagrangiano.
- Los esquemas Eulerianos son mejores para problemas estacionarios. En dichos casos,  $[\nabla T]''' = 0$  y no poseen errores de proyección.

Cada una de estas aseveraciones han sido demostradas experimentalmente a través de una serie de test presentados a lo largo de la tesis. Los resultados hallados para el caso de transporte escalar pueden generalizarse a las ecuaciones de flujo incompresible resultando en expresiones similares a las presentadas en las Ecuaciones A.3 y A.4, pero con conclusiones particularmente diferentes:

- Los esquemas Eulerianos son mejores para problemas a bajos números de Reynolds. En estos casos, los errores son del mismo orden, pero en el caso Lagrangiano deben agregarse los errores de proyección.
- Los esquemas Lagrangianos son mejores para problemas de convección dominante donde la velocidad posee una variación temporal suave pero el gradiente de velocidad tiene altas variaciones espaciales. Este caso es muy común en problemas de fluido-dinámica como en las ondas de choque. En el resto de los casos alguno de los esquemas será mejor que el otro dependiendo de la influencia de los errores de proyección del Lagrangiano.

- Los esquemas Lagrangianos son mejores para problemas de más de una fase fluída involucrada. Esto es porque ambos esquemas necesitan resolver una ecuación de advección pura de un escalar para encontrar la posición de la interfase y una ecuación de este tipo es resuelta mucho mejor por el Lagrangiano:

$$\mathbf{u} \cdot \nabla \mathbf{u} < 1 \quad \text{and} \quad [\mathbf{u} \nabla \mathbf{u}] \approx \mathbf{u} [\nabla \mathbf{u}] \gg 1$$

Dado que las fórmulas de error garantizan que utilizar grandes pasos de tiempo en PFEM-2 no perjudica la calidad de la solución en el caso de flujos de baja variabilidad temporal, se propone la resolución de un problema industrial de este tipo. El caso resuelto se trató de un tanque de separación agua-petróleo en donde interesa conocer parámetros que califiquen la eficiencia de dicho tanque en base a los tiempos de residencia obtenidos y su distribución a lo largo del tiempo. Dado que distribuciones de este tipo requieren la simulación de grandes extensiones de tiempo, utilizar grandes pasos de tiempo representa la solución para reducir los tiempos de cómputo necesarios. Los resultados numéricos hallados muestran soluciones similares en PFEM-2 utilizando pasos de tiempo tanto conservativos  $CFL_{max} < 1$  como exigentes  $CFL_{max} > 10$ , reduciendo el tiempo de cómputo de varios días a sólo unas cuantas horas, con lo que demuestra la efectividad del método numérico propuesto en un caso de aplicación industrial referido al cálculo de la distribución de tiempos de residencia.

### A.3 Flujos a dos fases

Los problemas en dónde se presentan simultáneamente múltiples fluidos con diferentes propiedades, tanto en flujos externos como internos, ocurren más a menudo que los casos de flujo homogéneo [De Mier Torrecilla, 2009]. Entre las diversas situaciones que puede hallarse este tipo de flujos se encuentra la interacción agua-crudo en la recuperación mejorada de petróleo, mezcla de polímeros, reactores de columnas de burbujas y en la atomización de líquidos, que encuentra aplicación en sistema de propulsión aeroespacial, motores de combustión interna, procesamiento de alimentos, impresoras a chorro de tinta, por nombrar sólo algunos ejemplos.

Una clasificación de este tipo de flujo basada en las estructuras de interfase fue realizada por Ishii & Hibiki [Ishii and Hibiki, 2011], quienes agruparon estos flujos en:

- Flujos dispersos, que consisten en una fase dispersa (pequeñas partículas, gotas o

burbujas) distribuidas en un volumen conectado de la fase continua.

- Flujos separados, en donde dos fases continuas se encuentran separadas por una interfase.
- Flujos mixtos: presencia simultánea de flujos dispersos y separados.

La presente tesis está enfocada en la resolución de flujos separados a dos fases inmiscibles con condiciones isotérmicas, en dónde el cómputo de la interfase adquiere una relevancia notable. En los trabajos de Márquez Damián [Márquez Damián and Nigro, 2014; Márquez Damián, 2013] pueden encontrarse estrategias numéricas para flujos dispersos o mixtos.

### A.3.1 Extensión de PFEM-2 a flujos bifásicos

Para extender el método PFEM-2 a la solución de problemas a dos fases se debieron considerar diferentes complejidades que ocurren principalmente en la zona de la interfase. La selección de las diferentes estrategias se basa fundamentalmente en aquellas que no perjudican la capacidad de PFEM-2 de extender los pasos de tiempo sin comprometer la estabilidad de la simulación.

- *Evolución de las interfaces internas:* Los métodos clásicos como Volume of Fluid VoF [Hirt and Nichols, 1981] y Level Set (LSM) [Osher and Fedkiw, 2001] determinan la interfase a través de la advección Euleriana de una función marcadora. Esta estrategia sufre de los errores numéricos presentados en la sección anterior, mas aún si se desean utilizar grandes pasos de tiempos como es el espíritu de esta tesis. Aprovechando la ductilidad de las partículas, en PFEM-2 se utiliza la estrategia de partículas marcadoras para indicar que zonas del dominio se corresponden a un tipo de fluido y cuales al otro. Entonces, las partículas ahora transportan una función  $\lambda = \pm 1$  la cual es proyectada a la malla luego del paso convectivo. Una vez en la malla, la interfase queda determinada por los puntos  $x$  tales que  $\lambda(x) = 0$ . Debido a la advección de la función marcadora es realizada de manera Lagrangiana, queda demostrados que el error es menor a su par Euleriano, mas aún utilizando grandes pasos de tiempo.
- *Discontinuidades de las variables y/o sus gradientes:* La discontinuidad de las propiedades materiales de los fluidos en la interfase producen discontinuidades de las incógnitas o

sus gradientes que las funciones de forma lineales utilizadas por PFEM-2 no pueden capturar:

- saltos del gradiente de presión debido a saltos de densidad [Coppola-Owen and Codina, 2005]
- saltos de presión debido a saltos de viscosidad [Idelsohn et al., 2010]
- saltos del gradiente de velocidad debido a saltos de viscosidad [Márquez Damián, 2013]
- saltos de presión debido a la tensión superficial [Ausas et al., 2012]

En esta tesis se adopta el enriquecimiento de las funciones de forma propuesto por Coppola [Coppola-Owen and Codina, 2005] para los gradientes de presión. Debido a que en algunas situaciones resulta insuficiente, en esta tesis se propone otro tipo de funciones de forma enriquecidas que admiten tanto resolución de forma condensada que no garantiza continuidad entre los elementos, como así también admiten continuidad inter-elemental a costa de agregar incógnitas extras al sistema de ecuaciones. Además, el caso correspondiente al cálculo de la presión al condensar los nuevos grados de libertad es profundamente analizado en esta tesis, presentando las expresiones directas para su implementación.

- *Tratamiento de la tensión superficial:* El problema de utilizar funciones marcadoras de tipo signo es que en la interfase dicha función adopta una morfología abrupta de tipo escalón. Al calcular la curvatura a través de las derivadas de dicha función marcadora, esto es  $\kappa = \nabla \cdot \nabla \lambda$ , se encuentra un resultado demasiado impreciso que genera corrientes parásitas y suele imposibilitar continuar con la simulación. Entre las varias soluciones propuestas en la literatura [Cummins et al., 2005], en esta tesis se elige la propuesta de Albadawi [Albadawi et al., 2013] el cual halla una función distancia  $\alpha$  a partir de operaciones algebraicas de la función  $\lambda$ , que se utiliza para el cómputo de una curvatura mucho más suave  $\kappa = \nabla \cdot \nabla \alpha$ . Finalmente, el término de la fuerza tensión superficial  $\mathbf{F}_\gamma$  se introduce en el sistema utilizando el modelo propuesto por Brackbill et al. [Brackbill et al., 1992] denominado Continuum Surface Force (CSF), haciendo

$$\mathbf{F}_\gamma = \gamma \kappa \mathbf{n} \delta_s(\phi) \quad (\text{A.5})$$

Finalmente, el método PFEM-2 para la solución de problemas de flujos a dos fases queda expresado como se presenta en el algoritmo 5 en donde aparece un parámetro numérico  $\theta$

el cual puede tomar los valores 0 o 1 dependiendo de la elección de utilizar un esquema de primer orden temporal estable o un esquema de segundo orden que puede ser inestable, respectivamente; y un término de fuerzas volumétricas  $\mathbf{F} = \int_{\Omega} N_j (\rho \mathbf{g} + \mathbf{F}_{\gamma}) d\Omega$  que aporta la fuerza de gravedad y la de tensión superficial incluida siguiendo la estrategia CSF.

---

**Algorithm 5** - Paso de tiempo en PFEM-2 para flujos incompresibles a dos fases.

---

1. Etapa convectiva:

$$\begin{cases} \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{u}^n(\mathbf{x}_p^\tau) d\tau \\ \hat{\mathbf{u}}_p^{n+1} = \mathbf{u}_p^n \\ \lambda_p^{n+1} = \lambda_p^n \end{cases}$$

2. Etapa de proyección:

$$\begin{aligned} \mathbf{M}_{ij} \hat{\mathbf{u}}_j^{n+1} &= \mathbf{M}_{ip} \hat{\mathbf{u}}_p^{n+1} \\ \mathbf{M}_{ij}^L \lambda_j^{n+1} &= \mathbf{M}_{ip} \lambda_p^{n+1} \end{aligned}$$

3. Etapa de predicción de velocidad:

$$\left( \mathbf{M}\left(\frac{\rho}{\Delta t}\right) + \mathbf{K}(\mu) \right) \hat{\mathbf{u}}^{n+1} = \mathbf{M}\left(\frac{\rho}{\Delta t}\right) \hat{\mathbf{u}}^{n+1} - \theta \mathbf{G} p^n + \mathbf{F}^{n+1}$$

4. Etapa de presión:

$$\mathbf{K}\left(\frac{\Delta t}{\rho} + \tau\right) p^{n+1} = \mathbf{B} \hat{\mathbf{u}}^{n+1} + \mathbf{K}\left(\frac{\Delta t}{\rho}\right) p^n + \mathbf{B}(\tau) \pi^n$$

5. Etapa de corrección:

$$\begin{aligned} \mathbf{M}(\rho) \mathbf{u}^{n+1} &= \mathbf{M}(\rho) \hat{\mathbf{u}}^{n+1} - \Delta t \mathbf{G} (p^{n+1} - p^n) \\ \rho_p \mathbf{u}_p^{n+1} &= \rho_p \hat{\mathbf{u}}_p^{n+1} + \sum_j \delta \mathbf{u}_j^{n+1} N_j(\mathbf{x}_p^{n+1}) \end{aligned}$$


---

### A.3.2 Resultados y validación

Una vez presentado el algoritmo PFEM-2 para flujos a dos fases, los capítulos siguientes presentan casos de prueba en dónde se verifica la precisión y eficiencia del mismo, comparando contra soluciones experimentales o numéricas empleando otras estrategias:

- En el Capítulo 9 se presentan resultados para casos de prueba que consisten en la advección de funciones escalares e interfaces que presentan ciertas complejidades como ser la captura de interfaces abruptas (sharp) y la resolución del estrechamiento de la misma. Es todos los casos presentados se utilizan pasos de tiempo tales que el

$CFL_{max} \gg 1$ , en donde el método PFEM-2 aprovecha sus características Lagrangianas para obtener resultados mucho más precisos que los Eulerianos.

- En el Capítulo 10 se presenta una batería de problemas dónde la inercia del flujo prevalece por sobre la difusión los cuales se resuelven utilizando siempre  $CFL_{max} \gg 1$ . Un primer ejemplo resuelve un problema de sloshing para diferentes ratios de densidad demostrando la necesidad del uso de funciones de forma enriquecidas para evitar corrientes parásitas en la interfase. Se resuelve también un problema de rotura de presa del cual se tienen resultados experimentales, obteniendo un buen acuerdo para muestreos de altura de superficie libre y de presión de impacto, tanto en simulaciones 2D como 3D. Un tercer caso, el de ondas estacionarias, muestra que la solución es correcta aún cuando el problema se convierte en difusivo dominante. El caso más relevante muestra la solución de la inestabilidad de Rayleigh-Taylor, en donde PFEM-2 logra extender el tamaño del paso de tiempo veinticinco veces en comparación de la estrategia VoF preservando similar calidad de resultados.
- En el Capítulo 11 se resuelven problemas en dónde la tensión superficial domina el comportamiento de los flujos. En primer lugar se realiza un análisis de estabilidad que determina que para ciertos tipos de problemas el algoritmo puede diverger. En el caso de la burbuja ascendente, al utilizar un paso de tiempo controlado PFEM-2 obtiene resultados similares a otras alternativas, y además mantiene la ventaja sobre métodos tipo VoF al poder utilizar pasos de tiempo varias veces mayor sin comprometer la estabilidad, pese a encontrarse cierto deterioro en la solución. Utilizando PFEM-2 tambien se resuelve la atomización de un jet líquido en donde un flujo altamente inercial genera desprendimientos, filamentos y gotas sobre las cuales domina la tensión superficial. La utilización de grandes pasos de tiempo permite obtener resultados precisos sólo en las zonas de alta convección, mientras que en zonas dominadas por la tensión superficial (gotas desprendidas) conlleva a una solución incorrecta de las mismas que se refleja a través de la pérdida de masa de los desprendimientos.

## A.4 Trabajos futuros

La presente tesis deja abierta varias líneas de investigación y desarrollo en la evolución del método. Entre ellas se pueden listar:

- Obtención de una formulación única para el tratamiento de forma eficiente de problemas donde convivan de manera simultánea zonas de alta tensión superficial junto a un flujo global dominado por inercia o fuerzas gravitatorias.
- Corrección del enrichment cuando los grados de libertad son condensados utilizando fuerzas inter-elementales.
- Extensión del método para poder utilizar mallas de cuadrángulos (hexaedros) o polígonos (poliedros) en general.
- Implementación del método sobre la librería OpenFOAM®utilizando el método de los volúmenes finitos para las etapas Eulerianas del algoritmo.



# Bibliography

- [Albadawi et al., 2013] Albadawi, A., Donoghue, D., Robinson, A., Murray, D., and Delauré, Y. (2013). Influence of surface tension implementation in volume of fluid and coupled volume of fluid with level set methods for bubble growth and detachment. *International Journal of Multiphase Flow*, 53(0):11 – 28.
- [Allievi and Bermejo, 2000] Allievi, A. and Bermejo, R. (2000). Finite element modified method of characteristics for Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 32:439–464.
- [Antuono and Colagrossi, 2013] Antuono, M. and Colagrossi, A. (2013). The damping of viscous gravity waves. *Wave Motion*, 50:197–209.
- [Ausas et al., 2012] Ausas, R. F., Buscaglia, G. C., and Idelsohn, S. R. (2012). A new enrichment space for the treatment of discontinuous pressures in multi-fluid flows. *International Journal for Numerical Methods in Fluids*, 70(7):829–850.
- [Balay et al., 2012] Balay, S., Brown, J., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H. (2012). PETSc Web page. <http://www.mcs.anl.gov/petsc>.
- [Becker, 2015] Becker, P. (2015). *An enhanced Particle Finite Element Method with special emphasis on landslides and debris flows*. PhD thesis, Universitat Politècnica de Catalunya.
- [Becker et al., 2014] Becker, P., Idelsohn, S., and Oñate, E. (2014). A unified monolithic approach for multi-fluid flows and fluid–structure interaction using the particle finite element method with fixed mesh. *Computational Mechanics*, pages 1–14.
- [Berberović et al., 2009] Berberović, E., van Hinsberg, N. P., Jakirlić, S., Roisman, I. V., and Tropea, C. (2009). Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. *Phys. Rev. E*, 79:036306.

- [Blasco et al., 1998] Blasco, J., Codina, R., and Huerta, A. (1998). A fractional-step method for the incompressible navier-stokes equations related to a predictor-multicorrector algorithm. *Int. J. Numer. Methods Fluids*, 28(10):1391–1419.
- [Brackbill et al., 1992] Brackbill, J., Kothe, D., and Zemach, C. (1992). A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335 – 354.
- [Burden and Faires, 2005] Burden, R. L. and Faires, J. D. (2005). *Numerical Analysis*. Thompson.
- [Bänsch, 2001] Bänsch, E. (2001). Finite element discretization of the navier–stokes equations with a free capillary surface. *Numerische Mathematik*, 88(2):203–235.
- [Chesnel et al., 2011] Chesnel, J., Menard, T., Reveillon, J., and Demoulin, F.-X. (2011). Subgrid analysis of liquid jet atomization. *Atomization and Sprays*, 21(1):41–67.
- [Chorin, 1968] Chorin, A. (1968). Numerical solution of the navier-stokes equations. *Mathematics of Computation*, 22:745–762.
- [Codina, 2000] Codina, R. (2000). Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 190(13–14):1579 – 1599.
- [Codina, 2001] Codina, R. (2001). Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics*, 170(1):112 – 140.
- [Colagrossi et al., 2013] Colagrossi, A., Souto-Iglesias, A., Antuono, M., and Marrone, S. (2013). Smoothed-particle-hydrodynamics modeling of dissipation mechanisms in gravity waves. *Phys. Rev. E*, 87:023302.
- [Coppola-Owen and Codina, 2005] Coppola-Owen, A. H. and Codina, R. (2005). Improving eulerian two-phase on finite element approximation with discontinuous gradient pressure shape functions. *International Journal for Numerical Methods in Fluids*, 49:1287 – 1304.
- [Costarelli et al., 2013] Costarelli, S. D., Storti, M. A., Paz, R. R., Dalcin, L., and Idelsohn, S. R. (2013). Gpgpu implementation of the bfcc algorithm for pure advection equations. *Cluster Computing*, accepted.
- [Cummins et al., 2005] Cummins, S. J., Francois, M. M., and Kothe, D. B. (2005). Estimating curvature from volume fractions. *Computers and Structures*, 83(6–7):425 – 434.

- [Dadvand et al., 2010] Dadvand, P., Rossi, R., and Oñate, E. (2010). An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Archives of Computational Methods in Engineering*, 17:253–297.
- [De Mier Torrecilla, 2009] De Mier Torrecilla, M. (2009). *Numerical Simulation of multi-fluid flows with the Particle Finite Element Method*. PhD thesis, Universitat Politecnica de Catalunya.
- [Deshpande et al., 2012] Deshpande, S., Anumolu, L., and Trujillo, M. (2012). Evaluating the performance of the two-phase flow solver interfoam. *Comput. Sci. Disc.*, 5:014016 (36pp).
- [Desjardins et al., 2008] Desjardins, O., Moureau, V., and Pitsch, H. (2008). An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395 – 8416.
- [Donea, 1983] Donea, J. (1983). Arbitrary lagrangian-eulerian finite element methods. In Belytschko, T. and Hughes, T., editors, *In Computational Methods for Transient Analysis*, pages 474–516.
- [Donea and Huerta, 2003] Donea, J. and Huerta, A. (2003). *Finite Element Method for Flow Problems*. Wiley, Chichester England.
- [Enright et al., 2002] Enright, D., Fedkiw, R., Ferziger, J., and Mitchell, I. (2002). A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83 – 116.
- [Enright et al., 2005] Enright, D., Losasso, F., and Fedkiw, R. (2005). A fast and accurate semi-lagrangian particle level set method. *Comput. Struct.*, 83(6-7):479–490.
- [Fabbroni, 2009] Fabbroni, N. (2009). *Numerical simulations of passive tracers dispersion in the sea*. PhD thesis, Alma Mater Studiorum – Università di Bologna.
- [Farrell and Maddison, 2011] Farrell, P. and Maddison, J. (2011). Conservative interpolation between volume meshes by local galerkin projection. *Computer methods in applied mechanics and engineering*, 200(1-4):89–100.
- [Farrell et al., 2009] Farrell, P., Piggott, M., Pain, C., Gorman, G., and Wilson, C. (2009). Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering*, 198(33–36):2632 – 2642.

[Fedkiw et al., 1999] Fedkiw, R. P., Aslam, T., Merriman, B., and Osher, S. (1999). A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457 – 492.

[Felippa, 2004] Felippa, C. (2004). *Introduction to Finite Element Methods*. University of Colorado, Chapter 10.

[Ferziger and Peric, 2002] Ferziger, J. and Peric, M. (2002). *Computational Methods for Fluid Dynamics*. Number 3rd Edition. Springer-Verlag Berlin Heidelberg.

[Galusinski and Vigneaux, 2008] Galusinski, C. and Vigneaux, P. (2008). On stability condition for bifluid flows with surface tension: Application to microfluidics. *Journal of Computational Physics*, 227(12):6140 – 6164.

[Ganesan et al., 2007] Ganesan, S., Matthies, G., and Tobiska, L. (2007). On spurious velocities in incompressible flow problems with interfaces. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1193 – 1202.

[Gimenez, 2014] Gimenez, J. (2014). Implementación del método PFEM sobre arquitecturas paralelas. Master's thesis, Facultad de Ingeniería y Ciencias Hídricas - Centro de Investigación de Métodos Computacionales - Universidad Nacional del Litoral.

[Gimenez et al., 2012] Gimenez, J., Nigro, N., and Idelsohn, S. (2012). Improvements to solve diffusion-dominant problems with PFEM-2. In *Mecánica Computacional*, volume XXXI, pages 137–155.

[Gimenez et al., 2014] Gimenez, J., Nigro, N., and Idelsohn, S. (2014). Evaluating the performance of the particle finite element method in parallel architectures. *Journal of Computational Particle Mechanics*, 1:103–116.

[Gimenez and González, 2015] Gimenez, J. M. and González, L. M. (2015). An extended validation of the last generation of particle finite element method for free surface flows. *Journal of Computational Physics*, 284(0):186 – 205.

[Gingold and Monaghan, 1977] Gingold, R. A. and Monaghan, J. J. (1977). Smoothed particle hydrodynamics, theory and application to non-spherical stars. *Royal Astronomical Society*, 181:375–389.

- [Gómez-Goñi et al., 2013] Gómez-Goñi, J., Garrido-Mendoza, C. A., Cercós, J. L., and González, L. (2013). Two phase analysis of sloshing in a rectangular container with volume of fluid (vof) methods. *Ocean Engineering*, 73:208–212.
- [Graham and James, 1996] Graham, D. and James, P. (1996). Turbulent dispersion of particles using eddy interaction models. *International Journal of Multiphase flow*, 22:157 – 175.
- [Gualtieri, 2006] Gualtieri, C. (2006). Numerical simulation of flow and tracer transport in a disinfection contact tank. In *Conference: Third Biennial Meeting: International Congress on Environmental Modelling and Software (iEMSS 2006)*.
- [Guermond et al., 2006] Guermond, J., Minev, P., and Shen, J. (2006). An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44–47):6011 – 6045.
- [Guermond and Quartapelle, 2000] Guermond, J. and Quartapelle, L. (2000). A projection fem for variable density incompressible flows. *Journal of Computational Physics*, 165:167–188.
- [Harlow, 1955] Harlow, F. (1955). A machine calculation method for hydrodynamic problems. Technical report, Los Alamos Scientific Laboratory report LAMS-1956.
- [Harlow and Welch, 1965] Harlow, F. H. and Welch, J. E. (1965). Numerical Calculation of Time Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids* (1958-1988), 8(12):2182–2189.
- [Hirt and Nichols, 1981] Hirt, C. and Nichols, B. (1981). Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comput. Phys*, 39(1):201–225.
- [Hryb et al., 2009] Hryb, D., Cardozo, M., Ferro, S., and Goldschmit, M. (2009). Particle transport in turbulent flow using both lagrangian and eulerian formulations. *International Communications in Heat and Mass Transfer*, 36(5):451 – 457.
- [Hysing, 2006] Hysing, S. (2006). A new implicit surface tension implementation for interfacial flows. *International Journal for Numerical Methods in Fluids*, 51(6):659–672.
- [Hysing et al., 2009] Hysing, S., Turek, S., Kuzmin, D., Parolini, N., Burman, E., Ganesan, S., and Tobiska, L. (2009). Quantitative benchmark computations of two-dimensional bubble dynamics. *International Journal for Numerical Methods in Fluids*, 60(11):1259–1288.

[Idelsohn et al., 2003a] Idelsohn, S., Calvo, N., and Oñate, E. (2003a). Polyhedrization of an arbitrary 3d point set. *Computer Method in Applied Mechanics and Engineering*.

[Idelsohn et al., 2010] Idelsohn, S., Mier-Torrecilla, M., Nigro, N., and Oñate, E. (2010). On the analysis of heterogeneous fluids with jumps in the viscosity using a discontinuous pressure field. *Computational Mechanics*, 46(1):115–124.

[Idelsohn et al., 2013a] Idelsohn, S., Nigro, N., Gimenez, J., Rossi, R., and Marti., J. (2013a). A fast and accurate method to solve the incompressible Navier-Stokes equations. *Engineering Computations*, 30-Iss:2:197–222.

[Idelsohn et al., 2013b] Idelsohn, S., Nigro, N., Gimenez, J., Rossi, R., and Marti., J. (2013b). A fast and accurate method to solve the incompressible navier-stokes equations. *Engineering Computations*, 30-Iss:2:197–222.

[Idelsohn et al., 2012] Idelsohn, S., Nigro, N., Limache, A., and Oñate., E. (2012). Large time-step explicit integration method for solving problems with dominant convection. *Comp. Meth. in Applied Mechanics and Engineering*, 217-220:168–185.

[Idelsohn et al., 2003b] Idelsohn, S., Oñate, E., Calvo, N., and Pin, F. D. (2003b). The meshless finite element method. *Int. J. Num. Meth. Engng.*, 58,6:893–912.

[Idelsohn et al., 2004] Idelsohn, S., Oñate, E., and Del Pin, F. (2004). The particle finite element method a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal of Numerical Methods*, 61:964–989.

[Idelsohn et al., 2015] Idelsohn, S., Oñate, E., Nigro, N., Becker, P., and Gimenez, J. (2015). Lagrangian versus eulerian integration errors. *Computer Methods in Applied Mechanics and Engineering*, 293(0):191 – 206.

[Idelsohn et al., 2014] Idelsohn, S. R., Marti, J., Becker, P., and Oñate, E. (2014). Analysis of multifluid flows with large time steps using the particle finite element method. *International Journal for Numerical Methods in Fluids*, 75(9):621–644.

[Ishii and Hibiki, 2011] Ishii, M. and Hibiki, T. (2011). *Thermo-Fluid Dynamics of Two-Phase Flow*. Springer, United States.

- [Kacianauskas et al., 2010] Kacianauskas, R., Maknickas, A., Kaceniauskas, A., Markauskas, D., and Balevicius, R. (2010). Parallel discrete element simulation of poly-dispersed granular material. *Advances in Engineering Software*, 41:52–63.
- [Kaludercic, 2004] Kaludercic, B. (2004). Parallelisation of the lagrangian model in a mixed eulerian-lagrangian cfd algorithm. *Parallel Distrib. Comput.*, 64:277–284.
- [Karypis and Kumar, 1998] Karypis, G. and Kumar, V. (1998). A parallel algorithm for multilevel graph partitioning and sparse matrix reordering. *Parallel and Distributed Computing*, 48:71–95.
- [Karypis and Kumar, 1999] Karypis, G. and Kumar, V. (1999). A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20:359–392.
- [Kirk et al., 2006] Kirk, B. S., Peterson, J. W., Stogner, R. H., and Carey, G. F. (2006). libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, 22(3–4):237–254. <http://dx.doi.org/10.1007/s00366-006-0049-3>.
- [Klein, 2003] Klein (2003). A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations. *Journal of Computational Physics*, 186(2):652 – 665.
- [Klostermann et al., 2013] Klostermann, J., Schaake, K., and Schwarze, R. (2013). Numerical simulation of a single rising bubble by vof with surface compression. *International Journal for Numerical Methods in Fluids*, 71(8):960–982.
- [Lam, 2009] Lam, K. W. (2009). *A Numerical Surface Tension Model for Two-Phase Flow Simulations*. PhD thesis, University of Groningen.
- [Lamb, 1993] Lamb, H. (1993). *Hydrodynamics*. Number 6th Edition. Cambridge Mathematical Library.
- [LeVeque, 1996] LeVeque, R. (1996). High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.*, 33(2):627 – 665.
- [Lighthill, 2001] Lighthill, J. (2001). *Waves in fluids*. Cambridge University Press.

- [Lobovský et al., 2014] Lobovský, L., Botia-Vera, E., Castellana, F., Mas-Soler, J., and Souto-Iglesias, A. (2014). Experimental investigation of dynamic pressure loads during dam break. *Journal of Fluids and Structures*, 48(0):407 – 434.
- [Lorstad et al., 2004] Lorstad, D., Francois, M., Shyy, W., and Fuchs, L. (2004). Assessment of volume of fluid and immersed boundary methods for droplet computations. *International Journal for Numerical Methods in Fluids*, 46(2):109–125.
- [Mackerle, 2004] Mackerle, J. (2004). Object-oriented programming in fem and bem: a bibliography (1990–2003). *Advances in Engineering Software*, 35(6):325 – 336.
- [Márquez Damián, 2013] Márquez Damián, S. (2013). *An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces*. PhD thesis, Facultad de Ingeniería y Ciencias Hídricas (FICH) Instituto de Desarrollo Tecnológico para la Industria Química (INTEC) Universidad Nacional del Litoral.
- [Márquez Damián and Nigro, 2014] Márquez Damián, S. and Nigro, N. (2014). An extended mixture model for the simultaneous treatment of small-scale and large-scale interfaces. *International Journal for Numerical Methods in Fluids*, 75(8):547–574.
- [Monaghan, 1988] Monaghan, J. (1988). An introduction to SPH. *Computational Physics Communications*, 48:89–96.
- [Moore, 1965] Moore, G. (1965). Cramming more components onto integrated circuits. *Electronics Magazine*, pages 114–117.
- [Ménard et al., 2007] Ménard, T., Tanguy, S., and Berlemont, A. (2007). Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet. *International Journal of Multiphase Flow*, 33(5):510 – 524.
- [Nadukandi, 2015] Nadukandi, P. (2015). Numerically stable formulas for a particle-based explicit exponential integrator. *Computational Mechanics*, 55(5):903–920.
- [Nestor et al., 2009] Nestor, R. M., Basa, M., Lastiwka, M., and Quinlan, N. J. (2009). Extension of the finite volume particle method to viscous flow. *Journal of Computational Physics*, 228(5):1733 – 1749.
- [Osher and Fedkiw, 2001] Osher, S. and Fedkiw, R. (2001). Level set methods: an overview and some recent results. *Journal of Computational Physics*, 169:463 – 502.

- [Parolini and Burman, 2005] Parolini, N. and Burman, E. (2005). A finite element level set method for viscous free-surface flows. In *7th Conference on Applied and Industrial Mathematics in Italy*, pages 416–427.
- [Puckett et al., 1997] Puckett, E., Almgren, A., Bell, J., Marcus, D., and Rider, W. (1997). A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *J. Comput. Phys.*, 130(2):269 – 282.
- [Quartapelle, 1993] Quartapelle, L. (1993). *Numerical solution of the incompressible Navier-Stokes equations*. Birkhäuser, Basel.
- [Quarteroni and Valli, 1994] Quarteroni, A. and Valli, A. (1994). *Numerical approximation of partial differential equations*. Springer-Verlag, Germany.
- [Sbalzarini et al., 2006] Sbalzarini, I., Walther, J., Bergdorf, M., Hieber, S., Kotsalis, E., and Koumoutsakos, P. (2006). {PPM} – a highly efficient parallel particle–mesh library for the simulation of continuum systems. *Journal of Computational Physics*, 215(2):566 – 588.
- [Shinjo and Umemura, 2010] Shinjo, J. and Umemura, A. (2010). Simulation of liquid jet primary breakup: Dynamics of ligament and droplet formation. *International Journal of Multiphase Flow*, 36(7):513 – 532.
- [Shinjo and Umemura, 2011] Shinjo, J. and Umemura, A. (2011). Surface instability and primary atomization characteristics of straight liquid jet sprays. *International Journal of Multiphase Flow*, 37(10):1294 – 1304.
- [Shiono and Teixeira, 2000] Shiono, K. and Teixeira, E. C. (2000). Turbulent characteristics in a baffled contact tank. *Journal of Hydraulic Research*, 38(6):403 – 416.
- [Smagorinsky, 1963] Smagorinsky, J. (1963). General circulation experiments with the primitive equations. *Mon. Wea. Rev.*, 91:99–164.
- [Strubelj et al., 2009] Strubelj, L., Tiselj, I., and Mavko, B. (2009). Simulations of free surface flows with implementation of surface tension and interface sharpening in the two-fluid model. *International Journal of Heat and Fluid Flow*, 30(4):741 – 750.
- [Sussman and Puckett, 2000] Sussman, M. and Puckett, E. G. (2000). A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301 – 337.

- [Tezduyar, 1992] Tezduyar, T. (1992). Stabilized finite element formulations for incompressible flow computations. *Adv. Appl. Mech.*, 28:1–44.
- [Thomson, 1988] Thomson, D. J. (1988). *Random Walk models of turbulent dispersion*. PhD thesis, Department of Mathematics and Statistics, Brunel Universit.
- [Trinh and Chen, 2005] Trinh, H. P. and Chen, C. P. (2005). Modeling of turbulence effects on liquid jet atomization. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*.
- [Tryggvason, 1988] Tryggvason, G. (1988). Numerical simulations of the rayleigh-taylor instability. *Journal of Computational Physics*, 75:253–282.
- [Turek, 1998] Turek, S. (1998). *Efficient Solvers for Incompressible Flow Problems*. Springer.
- [Témam, 1969] Témam, R. (1969). Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (ii). *Archive for Rational Mechanics and Analysis*, 33(5):377–385.
- [Voller and Porté-Agel, 1965] Voller, V. and Porté-Agel, F. (1965). Moore's law and numerical modeling. *Journal of Computational Physics*, 179(0):698–703.
- [Weller et al., 1998] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620–631.
- [Wieckowsky, 2004] Wieckowsky, Z. (2004). The material point method in large strain engineering problems. *Computer methods in applied mechanics and engineering*, 193.39:4417–4438.
- [Wilson and Venayagamoorthy, 2010] Wilson, J. and Venayagamoorthy, S. (2010). Evaluation of hydraulic efficiency of disinfection systems based on residence time distribution curves. *Environmental Science and Technology*, 44(24):9377 – 9382.
- [Zalesak, 1979] Zalesak, S. (1979). Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31(3):335 – 362.
- [Zienkiewicz and Morgan, 1983] Zienkiewicz, O. and Morgan, K. (1983). *Finite Elements and Approximation*. Wiley, New York USA.