

Programa de Doctorat en Anàlisi Estructural

Doctoral Thesis

An enhanced Particle Finite Element Method with
special emphasis on landslides and debris flows

Pablo Agustín Becker

Supervisors:

Sergio Idelsohn

Eugenio Oñate Ibáñez de Navarra

Departament de Resistència de Materials i Estructures a l'Enginyeria
Universitat Politècnica de Catalunya

A mis padres

Acknowledgements

I would like to thank all the people that contributed to the realization of this thesis. Without their support this would not have been possible.

To begin with, I would like to thank my supervisors, Prof. Sergio Idelsohn and Prof. Eugenio Oñate for giving me the opportunity to come to Barcelona to work with them. Looking back at all the things I've learned during these years, I couldn't be possibly more grateful to them. In the countless hours we spent with Sergio discussing the PFEM in front of his blackboard, he was not only helping me understand the method, he was also teaching me to think critically how to tackle the engineering problems I might face in my future.

And as I talk about Sergio, I would also like to express my eternal gratitude to Lelia. They both were always there making themselves sure that I had everything I needed from the very first moment I arrived. They made me feel my home was not so far away. Also Riccardo, Pooyan, Antonia, Kike and Abel helped me feel at home from the very beginning. Alessandro, Francisco, Jordi, Lorenzo, Stefano and all the guys of the liveliest office in CIMNE: Ferran, Ignasi, M.Angel, Miquel, Roberto and Salva helped me both discussing engineering problems and relaxing when outside the office. Also Rosa for her help in all the paperwork.

I would also like to thank Prof. Yongqi Wang and Prof. Lorenzo Sanavia for their help during my secondments in Germany and Italy. In these periods I was able to advance considerably in the model for the landslides of debris flows, as part of the MUMOLADE team. Also the support, friendship and technical discussions with all the members of the MUMOLADE project made this thesis possible. Thank you guys!

And last but not least I would like to thank my family: my parents Carlos and Marta, who always supported me in all my decisions, even when I decided to "cross the puddle". My brothers Carlos and Victoria for being there, completing the family any brother could possibly ever dream of. My grandmother Nurie and my grandfather Norberto. Their love and endless patience with my (noisy) experiments in their backyard are one of the reasons why I chose to follow engineering. Also all the *domingueros*! And finally I would like to thank Maria for her love. She did not only gave me support during these years, but also gave me a new family here in Barcelona. Thank you Marina, Jaume and Irene.

The research has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n 289911.

Abstract

This thesis presents the implementation of a solution strategy to solve multi-material problems. The algorithm is based on the Particle Finite Element Method, second generation (PFEM-2). It consists on using a set of Lagrangian particles to convect properties, together with a fixed Finite Element (FE) mesh to solve the Lagrangian equations. One of the main goals in this research is to create an algorithm that is as general as possible, in order to be capable of solving a wide variety of problems. The resulting method is flexible enough as to solve both multi-fluid flows and Fluid-Structure Interaction (FSI) problems with minimum changes.

The work developed in this thesis takes the single fluid PFEM-2 method as the starting point. This strategy uses concepts of both Lagrangian and Eulerian strategies. In order to tackle the large deformation problems, a set of Lagrangian particles with no connectivities are used to convect the properties. Having convected the particles, information is projected to a fixed FE mesh, where the Lagrangian equations are solved. Finally the corrections are sent back to the particles.

In order to generalize the PFEM-2 to tackle multi-phase problems, sharp interfaces are defined, at which the material properties are considered discontinuous. Since this leads to discontinuities in the unknowns, the Finite Element space is enriched at those interfaces, later using a condensation procedure to maintain the number of unknowns in the system of equations. On the other hand, the convection strategy for the particles is also modified to take into account the change in the material properties, leading to a better approximation. The numerical examples show the accuracy on both multi-fluid and FSI problems.

The final chapter of this thesis covers the implementation of a strategy to solve landslides and debris flows. All the tools developed in this thesis, together with special customizations for this particular problem are used. The numerical experiments show that the physics of the problem is correctly captured, with good correlation against experimental results.

Resumen

Esta tesis presenta la implementación de un método computacional para la solución de problemas dinámicos con multi materiales. El algoritmo está basado en el Particle Finite Element Method de segunda generación (PFEM-2), que consiste en utilizar una serie de partículas Lagrangianas junto a una malla fija de Elementos Finitos (FE) para resolver las ecuaciones Lagrangianas. Uno de los objetivos fundamentales de este trabajo es la creación de un algoritmo tan general como sea posible, de forma tal que sea capaz de resolver un amplio espectro de problemas. El fruto de esta investigación es un método lo suficientemente flexible para resolver tanto flujos de multi fluidos como problemas de Interacción Fluido-Estructura (FSI).

Este trabajo toma como punto de partida el PFEM-2 para un único fluido. Este método utiliza conceptos tanto de estrategias Lagrangianas como Eulerianas. Una serie de partículas Lagrangianas sin interconectividades entre ellas son utilizadas para resolver la convección. Una vez se han convectado estas partículas, su información se proyecta a una malla fija de Elementos Finitos, donde las ecuaciones son resueltas. Finalmente, las correcciones se proyectan a las partículas para calcular el paso siguiente.

Con el objetivo de generalizar el PFEM2 para problemas con multi materiales, se definen interfaces agudas donde las propiedades de los materiales se consideran discontinuas. Debido a que esto ocasiona discontinuidades en las incógnitas, el espacio de Elementos Finitos es enriquecido para poder capturar estos saltos. Posteriormente estas nuevas variables son condensadas estáticamente para evitar un incremento en el número de incógnitas. Por otro lado, la estrategia de convección de partículas también es modificada para mejorar la aproximación cuando existen discontinuidades materiales. Los casos numéricos analizados muestran buena precisión tanto en problemas de multi fluidos como en casos FSI.

El último capítulo presenta la implementación de una estrategia para simular deslizamientos y corrientes de derrubios (riadas). El problema es tratado utilizando todas las herramientas generales desarrolladas en esta tesis junto con estrategias particulares para este problema. Las simulaciones numéricas muestran que la física del problema está bien representada, con buena correlación con resultados experimentales.

Contents

Contents	I
List of Figures	V
List of Figures	VII
List of Symbols	1
1 Introduction	1
1.1 Common solving strategies for multi-fluid flows	1
1.1.1 Eulerian Frameworks	2
1.1.2 Pure Lagrangian Frameworks	4
1.1.3 Mixed Frameworks	7
1.2 Objectives	7
1.3 List of publications	8
1.4 Organization	9
2 The PFEM-2 for Multi-materials	11

2.1	Introduction	11
2.2	The Finite Element Method	12
2.2.1	Discretization	12
2.2.2	Eulerian scalar transport	13
2.3	The PFEM-2 method	16
2.3.1	Circumventing the difficulties of convective transport	16
2.3.2	Transferring information onto the fixed mesh	20
2.3.3	Continuum equations	21
2.3.4	Transferring corrections back to the particles	22
2.4	Interface Description	23
2.4.1	Interface capturing	23
2.4.2	Detecting the boundaries of bodies	24
2.4.3	Improving the fields of unknowns: Enrichments	27
2.4.4	Condensing the new degrees of freedom	30
2.5	Particle management	39
2.5.1	Particle reseeding	39
2.5.2	Particle database	40
2.6	Summary and Conclusion	42
3	Solution Strategies for multi-fluids	45
3.1	Introduction	45
3.2	Navier Stokes equations	46
3.3	Fractional Step Method	48
3.3.1	Motivation	48

3.3.2 Weak Form	48
3.3.3 Improvements for the first order Fractional Step scheme	52
3.3.4 Boundary conditions	54
3.3.5 Special considerations on the interface	56
3.3.6 Complete algorithm	61
3.4 Monolithic strategy	62
3.4.1 Motivation	62
3.4.2 Spatial Discretization	62
3.4.3 Stabilized problem	63
3.4.4 Pressure enrichments on the interface	67
3.4.5 Further enrichments for the velocity field	68
3.4.6 Complete algorithm	72
3.5 Test Examples	72
3.5.1 Fractional Step	72
3.5.2 Monolithic Strategy	78
3.6 Summary and Conclusions	91
4 Solution strategies for solids and FSI problems	95
4.1 Introduction	95
4.2 Solids Formulation	96
4.2.1 Continuum model	96
4.2.2 Elastic Response: Hypoelastic model	98
4.2.3 Drucker Prager yield criterion for granular materials	101
4.2.4 Plastic Stress Update algorithm	104

4.2.5	Spatial discretization and weak form	109
4.3	Coupled solver for solids and fluids	114
4.3.1	Enrichment degrees of freedom	114
4.3.2	Complete algorithm	116
4.4	Numerical Examples	117
4.4.1	Elastic problems	117
4.4.2	Plasticity problems	126
4.5	Summary and Conclusions	127
5	Landslides and debris flows simulation	131
5.1	Introduction	131
5.2	Non-Newtonian Rheology	133
5.3	Granular material with seepage	136
5.3.1	Problem statement	137
5.3.2	Water model	138
5.3.3	Granular material model	145
5.3.4	Complete coupled model	146
5.4	Two resolutions approach	147
5.4.1	Spatial Domains	147
5.4.2	Frame of Reference	148
5.4.3	Complete Algorithm	150
5.5	Test Examples	151
5.5.1	Granular material collapse experiment	151
5.5.2	Small scale debris Flow with erosion	152

5.5.3	Water flow through a porous dam	155
5.5.4	Porous dam damage by water flow	156
5.5.5	Vajont landslide	158
5.6	Summary and Conclusions	162
6	Conclusions and future work	165
6.1	Developed strategies	166
6.2	Summary	166
6.3	Future lines of research	167
Bibliography		169

List of Figures

2.1	a) Real domain, b) Discretized Geometry	12
2.2	Three node triangle shape functions	13
2.3	Simple 1-D Convection Diffusion problem	15
2.4	PFEM-2 streamline integration	19
2.5	Streamlines of a falling water droplet	19
2.6	Contribution of i particle to j node	20
2.7	Contribution of i particle to j node	24
2.8	Lagrangian body in fixed mesh	25
2.9	Boundaries detection. In yellow, first layer of inactive elements; yellow: first layers of nodes outside the body; blue: interface elements	25
2.10	Setting particle distance	26
2.11	Particle distribution and approximate location of the free surface, shown in yellow.	27
2.12	Enrichments for discontinuous gradient	29
2.13	Pressure shape functions: Top: Replacement functions. Bottom: Enrichments	30

2.14 Two-phase thermal problem domain. Left: real problem. Right, discretized geometry	31
2.15 Two-phase problem with non-matching interfaces. Interface marked with the green line.	34
2.16 Two neighbour elements and their normals	35
2.17 Enrichment test example geometry. Up: real domain, bottom: discretized problem. The enrichment location is marked with the green dotted line . .	36
2.18 Thermal problem solution for $h = 1$ mesh	37
2.19 Thermal problem mesh convergence	39
2.20 Elements and Particles databases	41
2.21 Streamline integration. The first element candidates for the successive particles are shown in grey.	42
3.1 Pressure field of a water droplet before and after hitting a wall	53
3.2 Velocity Boundary terms in the pressure equation	55
3.3 Pressure distribution for the hydrostatic case	56
3.4 Enrichment for the pressure field	57
3.5 Hydrostatic test example. Up: real domain, bottom: discretized problem. The enrichment location is marked with the green dotted line	58
3.6 Hydrostatic problem mesh convergence with full integration for the divergence evaluated with N^*	59
3.7 Stencil required for the L2 projection in the nodes of element K	66
3.8 Pressure shape functions for the Monolithic strategy: Top: Replacement functions. Bottom: Enrichments	67
3.9 Real and simulated velocity profiles with restrictive interface conditions	69
3.10 Enrichments for the velocity field	69

3.11 Global and local coordinate systems.	70
3.12 Rayleigh-Taylor instability configuration	73
3.13 Vertical position of the advancing front(lower line) and rising left and right bubbles (upper line) in the Rayleigh-Taylor instability problem	74
3.14 Snapshots for the Rayleigh-Taylor instability problem at $t_{adim} = 1.0, 1.5, 1.75, 2.0, 2.25, 2.5$. Top: $Re = 256$; Bottom: $Re = 1000$	74
3.15 Sloshing Geometry	75
3.16 Snapshots for the sloshing simulation at $t = 0.55, 1.15, 1.70$. Top: PFEM-2, Bottom: OpenFoam	76
3.17 Dam Break experiment configuration	77
3.18 Initial Geometry and snapshots for the experimental and numerical results at $t = 0.3, 0.4, 0.5$	78
3.19 Snapshot for the simulation in 3D	78
3.20 Snapshot for manufactured problem	80
3.21 Time convergence for the manufactured problem	80
3.22 Mesh convergence for the manufactured problem	81
3.23 Sloshing Geometry	81
3.24 Couette flow problem geometry	82
3.25 Couette problem velocity contour	84
3.26 Couette problem mesh convergence	85
3.27 Rigid cylinder in viscous fluid	86
3.28 Vertical velocity of the cylinder	87
3.29 Snapshot of velocity contour at $t=1s$. The location of the interface is shown in grey	87
3.30 Relative velocity mesh convergence error for the falling cylinder case	88

3.31 Boat Geometry	89
3.32 Boat Computational domain	89
3.33 Boat Simulation snapshots at t=0.5s. Top, side and front views. Free surface and water particles in cyan.	92
4.1 Perfect Plasticity material response	98
4.2 Drucker Prager yield surface	102
4.3 Return Mapping	103
4.4 Return Mapping slope	103
4.5 Return Zones	104
4.6 Elastic stress update	105
4.7 Elastic stress update	105
4.8 Pressure distribution of a solid body surrounded by air under gravitational force.	114
4.9 Close-up of a linear element located at the interface showing the interpolated pressure field	115
4.10 Cantilever geometry	118
4.11 Mesh convergence for the cantilever beam	118
4.12 Cantilever beam with fluid geometry	119
4.13 Beam interface and pressure distribution	120
4.14 Convergence for the cantilever beam in fluid	120
4.15 Time evolution of the cantilever beam for different viscosities	122
4.16 Tridimensional cantilever geometry	122
4.17 Oscillation frequency of the 3D cantilever beam for different fluid densities	124
4.18 Mesh convergence for the oscillation frequency of the 3D cantilever beam	124

4.19 Turek FSI velocity contour	126
4.20 Tip displacement of the Turek FSI	126
4.21 Prandtl's problem geometry	127
4.22 Prandtl's problem computational domain	127
4.23 Footing velocity contour at $t = 1s$. Top: $\theta = 0^\circ$, middle: $\theta = 15^\circ$, bottom: $\theta = 30^\circ$	128
5.1 Discretization of the debris flow	132
5.2 Mean solid pressure of the slope	132
5.3 Material constitutive models	134
5.4 Bingham fluid for Drucker-Prager materials. Solid line: ideal response. Dotted line: regularization approximation	134
5.5 Water and granular material domains	137
5.6 Granular material regime. Top: collisional. Bottom:Frictional. Only the latter can be simulated as a continuum occupying the red region.	138
5.7 Average fluid velocity and Darcy velocity	139
5.8 Fluid volume balance in a porous media	140
5.9 Stored information from the fluid domain required to compute the coupling terms in the granular domain system	146
5.10 Discretizations on Ω_{topo} and Ω_{calc}	147
5.11 Calculation Domain displacement and particle distribution	149
5.12 Granular material dam break geometry	151
5.13 Comparison between experimental and numerical results for the 2D granular dam break at $t=0.7$ s. Top: Experimental results. Middle: Numerical results. Bottom: Comparison of the free surface. Experimental results in red and numerical in grey.	152

5.14 Topological Domain	153
5.15 Ω_{topo} pressure and initial position of Ω_{calc}	153
5.16 Debris flow evolution. Colour Description: Orange: Water, Blue: Solid .	154
5.17 Porous dam experimental setup	155
5.18 Porous dam 2D computational domain	156
5.19 Experimental and numerical results for the porous dam	156
5.20 Variable water inflow for the dam experiment	157
5.21 Damage measure for the dam experiment	158
5.22 Shape of the dam and water free surface for different inflows. Top: 51.75l/s, Middle: 69.07l/s, Bottom: 90.68l/s	158
5.23 Vajont landslide section before the collapse. The light grey area is the mobilized mass. The dark grey area is the soil that remained static.	160
5.24 Vajont landslide computational mesh	160
5.25 Vajont Landslide: Water free surface at time different times. From top to bottom: $t = 100s$, $t = 200s$, $t = 300s$	161
5.26 Vajont landslide mean velocity at failure time	161
5.27 Vajont Landslide: Snapshots during the landslide at $t = 500s$, $t = 510s$, $t = 520s$	162
5.28 Vajont landslide final deposition shape. Light grey: numerical result. Dotted line: Real profile.	162

Chapter 1

Introduction

1.1 Common solving strategies for multi-fluid flows

The problem of accurately solving multi fluid flows is still an open challenge. Despite the mechanics of single phase flows are well understood and can be solved efficiently without sacrificing accuracy, the modelling of two or more fluids is still a fast evolving field. One of the reasons for this is that the behaviour of the flow at each of the phases depends on the interface, and at the same time its shape depends on the flow (Brennen, 2005). This complex coupling makes solving multi-phase flows a challenging task.

Among the possible frameworks to derive a formulation, there are 3 main categories (Maitre, 2006): Eulerian (fixed framework) (Zienkiewicz and Taylor, 2000a), Lagrangian (moving framework) (Idelsohn et al., 2004) or a mixed framework (a combination of both) (Donea et al., 1982).

Eulerian formulations are a popular alternative to solve multi-fluid problems, developed from Eulerian single-fluid solvers. There exist several possibilities to deal with the problem of the moving boundary, but we will concentrate here on the most commonly used alternatives; the Finite Element Method (FEM) with the level set method to locate the interface (Osher and Sethian, 1988) and Finite Volume Method (FVM) with Volume Of Fluid (VOF) to determine the fraction of fluid in each cell (Hirt and Nichols, 1981). It must be noted that these combinations of discretization and tracking methods are not the only possible ones (for example they could be exchanged).

1.1.1 Eulerian Frameworks

Eulerian Finite Element Method

Finite Elements are nowadays a standard tool to solve both structural and fluid problems. The main idea is to discretize the spatial domain into elements whose geometry is defined by nodes. This reduces the number of unknowns from the original continuous problem to only the value of the unknowns at the nodes, a discrete problem. Within each element, the variables are obtained by interpolation using finite element *shape functions*. For example, classical low order elements use linear shape functions to reconstruct the values inside of them (Zienkiewicz and Taylor, 2000b).

It must be noted that a solution calculated by FEM can only be as good as the interpolation used. If the exact solution cannot be represented in the space spanned by the shape functions, then they will not be able to capture it accurately, but instead result in an averaged or even distorted solution. This is particularly important for multi-fluid models, since domains composed by different materials usually lead to discontinuities in the properties along the interface, which translates into discontinuities in the unknowns or in their gradients. For example, the pressure field of a water drop inside an air domain is discontinuous due to surface tension and standard Finite Elements are incapable of capturing this since no discontinuity is allowed in the shape functions field. An alternative to overcome this limitation is to use Enriched Finite Elements, which augment the standard shape functions with extra shape functions to capture the part of the solution that escapes from the standard shape function field. Coppola-Owen and Codina (2005) proposed a simple enrichment function that is capable of capturing accurately gradient discontinuities (for example in the pressure field for two fluids with different densities) . On the other hand, to capture discontinuities in the value of the unknown Ausas et al. (2011) proposed a set of 3 enrichment functions that are able to capture both a gradient discontinuity and a jump.

Level Set Method Even though enrichments are essential to model discontinuities, they are still insufficient to simulate multi-fluids unless they are coupled with a second tool to locate precisely the position of the interface, necessary to build the extra shape functions. A common tool to track the seam between the phases in FEM is the so called Level Set Method (LSM).

The LSM (Osher and Sethian, 1988) (Sethian and Smereka, 2003) consists of using a level set scalar function, often the so called *distance function*, that is convected according to the velocity. This function represents the distance from a point to the interface . By definition, the interface will be located where its value is zero. Once the geometry of the interface has been calculated and the properties at each side of it have been assigned, a fluid solver calculates velocities in the whole domain. Despite a naïve implementation of the level set method is not complex, its use would lead to certain problems. The main drawback is that the level set function field degrades when advancing in time and no longer represents the distance to the interface, leading to diffusion of the interface and loss of mass. This is why a recalculation of the level set must be done periodically to guarantee the properties are better conserved (Chang et al., 1996).

Still, when convecting a scalar function with Eulerian methods, numerical diffusion is inevitable. This means that the recalculation must predict this degradation to fix it. But for other convected properties, such as temperature, the diffusion effects after several time steps can be harder to track and fix (Ubbink, 1997) (Idelsohn et al., 2013).

Finite Volume Method

Other common choice, more popular than the FEM for fluid dynamics (Ashgriz and Mostaghimi, 2002), is the Finite Volume Method (FVM) (Eymard et al., 2000). The solution in the FVM is obtained by calculating fluxes through the faces of the cells, therefore leading to a formulation that is automatically conservative on the fluxes, unlike FE.

Volume Of Fluid For multi-phase fluids, due to the conservative nature of the FVM, 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 most popular method that uses this concept is the Volume Of Fluid (VOF) (Hirt and Nichols, 1981) and consists in defining a function that is the fraction occupied by one of the phases in each cell of the domain. Therefore we do not track the interface position, but the fraction of fluid instead. Once fluids have been convected between cells, the interface position has to be reconstructed, loosing precision in the process (Ubbink, 1997). This *sharing* between cells, called donor-acceptor scheme (Ubbink, 1997), inherits the conservative nature of the FVM, so the

mass of each of the fluids is guaranteed to be conservative in the fluxes. This is a key advantage with respect to the LSM in terms of mass conservation. Moreover, the FVM is very robust and is likely to be the most used one in commercial/widespread codes. For example, OpenFoam (Weller et al., 1998) uses this technology to solve multi-fluid problems.

It is worth mentioning that Eulerian methods using explicit strategies are not unconditionally stable. To get a stable scheme, the time step has to be such that a massless particle moving with the fluid velocity, must travel at most one element per time step. This is known as the Courant-Friedrichs-Lowy (CFL) condition. This is a critical restriction because it must be satisfied in the whole domain, so it becomes dominant when there are sections with high velocities and small mesh sizes (Courant et al., 1967). Despite implicit convection schemes are capable of obtaining larger time steps, advancing more than one element per time step may lead to non-physical results (Coppola, 2009).

1.1.2 Pure Lagrangian Frameworks

Lagrangian formulations offer a more natural choice for simulations in which deformations are not negligible, such as in multi-phase problems (Idelsohn et al., 2004) . They provide the key advantage that each 'particle' or node has its own properties and convects it through the domain, therefore reducing or almost eliminating numerical diffusion. For example for structural problems, the stress state, position, strain and historical values such as plasticity of each point of the domain have to be as exact as possible to guarantee accurate results. Convecting those variables through a fixed, indeformable mesh and attempting to rebuild a solution for the next time step would lead to inaccurate results. This is the main reason why structural codes for large deformations usually use Lagrangian frameworks.

However Lagrangian formulations also have their drawbacks and require expensive processing that is not required in Eulerian frameworks. In the case of meshless methods, constant track of all the moving points must be kept, a task that is non trivial when there are several million particles in a large domain. Complex search algorithms have to be used to speed up the searching time, such as bin-based ones (Samet, 1990). These algorithms are necessary since interaction forces are calculated using information from neighbour particles. For Lagrangian mesh methods in fluid problems, the extra task is

the construction of a new mesh at each time step. Among the common Lagrangian formulations currently in use we can cite the Smoothed Particles Hydrodynamics (Gingold and Monaghan, 1977) and the Particle Finite Element Method (Idelsohn et al., 2004).

Smoothed Particle Hydrodynamics

Taking advantage of the recent upraise in the Graphic Processor Units for scientific use, there has been an important development in SPH technology. This method does not use meshes, but employs weighting functions to define interaction forces between each particle and its neighbours. What makes this approach appealing for GPUs is that, if certain compressibility of the fluid is allowed, a completely explicit algorithm is possible, using only information from immediate neighbour particles. This way extremely fast solutions are possible in this type of hardware, though inaccurate for incompressible fluids.

For this reason simple SPH solvers are usually used in simulations where accuracy is not an issue, for example in computer games (Müller et al., 2003). The results look real, yet not precise. Another advantage is that since the reactions are written between each particle and its neighbours, the way of writing these forces has a more natural meaning than, for example, the weak formulation that is the result of the FEM discretization. This eases the implementation of new forces. However it must be noted that, besides the compressibility restriction, smaller time steps must be used due to the explicit nature of the method, and, most importantly, the SPH does not solve balance equation but only estimates reaction forces, meaning that the accuracy in the momentum equations is severely affected.

Particle Finite Element Method

Another choice for fluids in a pure Lagrangian framework is the Particle Finite Element Method (PFEM) (Idelsohn et al., 2004). The PFEM consist of using a set of particles that define the nodes of a finite element mesh. Since fluids have no deformation limit, remeshing must be done at each time step, creating new connectivities and possibly new nodes. Otherwise the mesh would have elements that are too distorted or even inverted.

As all Lagrangian methods, the PFEM offers a more natural solution to problems where

the 'particles' of the domain can move freely. Unlike the level set method, there is no need to recalculate the surface since the location of the interface is obtained trivially; no extra function is needed since each particle is associated with a material. Another advantage is that no artificial smoothing of the surface occurs.

Mier-Torrecilla et al. (2011) used PFEM to simulate several interacting fluids with temperature coupling for the analysis of magmatic chambers. The natural ability of the method to model complex interfaces was enhanced with improvements to deal with sharp variations in the fluids' properties, obtaining results in accordance with the natural phenomena. The thermally coupled PFEM has been also employed for the simulation of industrial applications (Oñate et al., 2014a), (Oñate et al., 2014b). For the particular case of landslides, a full solution of the tridimensional problem is currently being performed by Salazar et al. (2011), using both Newtonian fluid approximations or more accurate, non Newtonian constitutive laws. In the latter models, an effective yet simple alternative is the Bingham plastic. In this approximation the material behaves initially as a solid, and from a certain stress tension threshold it behaves as a Newtonian fluid. This response is kept while the stress is higher than the threshold. Once it has dropped, it behaves again as a solid material.

Another advantage of the PFEM is that since there is no need to define a function to track the interface, the number of phases is unlimited without any modifications. Each particle simply carries the properties of the associated material. For example it is possible to analyse the wave generation caused by a landslide in dams (Salazar et al., 2011) using as many phases as needed.

Despite the advantages of the PFEM method, remeshing and reassembling of the system is necessary at each time step because the particles are constantly moving. Resizing the system is a task that is likely to be computationally expensive than only solving a system that is already assembled. Parallelization of this type of software (to take advantage of supercomputers) is not straightforward due to the moving nature of the domain and the constantly changing connectivities. One of the main reasons for this is that the remeshing stage is not easily parallelizable.

1.1.3 Mixed Frameworks

Algorithms combining both Eulerian and Lagrangian tools have proved to be a good alternative to pure methods. As an example, the work of Enright et al. (2002) uses a pure Eulerian solver for the fluid, but employs a modified LSM to track the interface , that corrects the free surface using Lagrangian particles that fix the errors in the level set function. This method is more accurate than both the pure Eulerian or pure Lagrangian counterparts in the tracking of the interface. Another method that was developed already in the 1950s is the Particle in Cell (PIC) (Evans et al., 1957). This method consists of using moving particles to convect properties and a fixed mesh to calculate forces and pressures. This method not only tracks the interface but also density, viscosity, etc., therefore eliminating the need of the non-linear convective term. An extension of the PIC is the Material Point Method (MPM). The MPM is a popular meshfree method for the simulation of solids in large deformation. Each particle has an associated velocity, position, stresses and mass, which implies that mass conservation is automatically guaranteed and the fixed mesh is used to compute the momentum equations.

A recent enhancement of the PIC method is the so called Particle Finite Element Method second generation (PFEM-2) with fixed mesh (Idelsohn et al., 2013). This method consists of using a fixed FE mesh and particles that are used to estimate the solution by integrating along the streamlines defined by the fixed mesh. The corrections needed are minimal and the accuracy of the solver is excellent for a given mesh size. Moreover, since streamlines are followed in a Lagrangian fashion, there is no limitation in the time step. The only bound for the time-step is the required precision. A key difference between the MPM and the PFEM-2 is that in the latter the particles do not represent a fixed amount of mass. This allows for a variable mesh size to obtain a better resolution in some areas of the domain. On the downside, the absence of a fixed mass implies that mass conservation is not guaranteed.

1.2 Objectives

The aim of this thesis is the development of a strategy to solve multi-material problems, including multi-fluid flows and Fluid-Structure Interaction simulations. The work by

Idelsohn et al. (2013) on the PFEM-2 is taken as the starting point to develop the method, combining a set of Lagrangian particles with a fixed Finite Element mesh. In the original article it was shown the potential of solving single fluid problems with this new strategy, showing both good accuracy and excellent computational times compared to standard Eulerian approaches. Following this line of research, the objective of this work is to generalize the PFEM-2 strategy to tackle multi-phase problems. This is achieved by modifying the convection of particles as well as improving the definition of the unknowns at the interfaces. The resulting algorithm must as general as possible to be capable of simulating multi-phase phenomena with arbitrarily different properties, including the interaction between solids and fluids.

Besides the implementation of a general solver for multi-material problems, the second objective of this thesis is the development of a strategy to simulate landslides and debris flows using the PFEM-2. These phenomena are geological events that occur when a portion of granular material is detached from a slope and flows down the terrain, in many cases eroding the bed and thus increasing their mass as they move. The prediction and simulation of these events is an important field of research due to the damages and high number of casualties that they cause every year.

In order to simulate landslides and debris flows, a solver has to be developed taking into account both the undamaged granular material, the fluidized phase and the intergranular water flow, making it a challenging multi-material problem. Furthermore, the model must be capable of simulating both the initiation and the subsequent displacement of the detached material. In this sense, the general solution strategies of this thesis provide the necessary tools to create the custom solver to simulate this phenomenon.

1.3 List of publications

Part of the work presented in this thesis has been published in academic journals. The following list is ordered chronologically.

- Becker, P., Nigro, N. and Idelsohn, S.R. (2012). Integración temporal explícita con grandes pasos de tiempo de la ecuación de transmisión del calor. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 24(4):187-197.
DOI: 10.1016/j.rimni.2012.07.001

- 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. DOI: 10.1002/fld.3908
- Becker, P., Idelsohn, S.R. 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*, 1-15. DOI:10.1007/s00466-014-1107-0
- Idelsohn, S.R., Oñate, E. , Nigro, N., Becker, P. and Gimenez, J. (2015). Lagrangian vs Eulerian integration errors, *Computer Methods in Applied Mechanics and Engineering*. DOI: 10.1016/j.cma.2015.04.003

1.4 Organization

The main goal of this thesis is to extend the spectrum of capabilities of the PFEM-2 to tackle a wider variety of problems. In order to do so, Chapter 2 presents the key concepts required to solve multi-materials problems. This chapter introduces the convection strategy of the PFEM-2 as well as the tools to enhance the modelling of sharp discontinuities in the material properties of the domain. The accuracy of the method is determined with simple scalar problems, by analysing the source of the errors.

In Chapter 3 the strategies to solve Newtonian fluids are illustrated. In the first section the general problem is presented and the implementation of the PFEM-2 for this problem is described. Next, a segregated (fractional step) strategy is explained. This method is well suited for low viscosity problems, obtaining good computational times with a good accuracy as long as the viscosity remains low. Finally, a monolithic strategy is developed. Unlike the segregated scheme, the monolithic approach is capable of solving accurately problems of arbitrary high viscosity, at the expense of a higher computational cost.

Chapter 4 presents the implementation of a solid solver with the PFEM-2. A simple hypo-elastic constitutive model is chosen for elastic problems that is coupled with an associative Drucker-Prager yield criterion for the plastic regime. Examples in both 2D and 3D are presented to analyse the accuracy of the method. The interaction between solids and fluids is also covered.

As this work is framed within the Multiscale Modelling of Landslides and Debris Flows (MuMoLaDe, <http://www.mumolade.com>) project, the last chapter is devoted to the simulation of landslides and debris flows. To tackle the complex nature of this phenomena, several strategies are presented. First, a non-Newtonian model is introduced for the simulation of plastic flows in high deformation regimes. Also a coupled model for porous granular materials with water flowing through the inter-granular media is developed and finally a moving mesh strategy is presented in order to track the movement of the mass down the slope. Finally, the developed tools are combined together in geomechanics numerical simulations.

Chapter 2

The PFEM-2 for Multi-materials

2.1 Introduction

On single material problems, the solution found must satisfy all the conservation equations that govern the system of interest. For materials with a complex constitutive models this can be a challenging task due to the non linear response, thus difficulting the modelling of these types of problems.

Modelling correctly multi-materials is even a more challenging task, since it is not only necessary to fulfil the conservation equations of each phase, but also achieve equilibrium at the interfaces. This implies that this type of problems requires the use of several tools that must merge together seamlessly in order to solve accurately the problem of interest. On one hand, the continuum equations have to be fulfilled, while at the same time the interaction forces between the different materials must be in equilibrium and finally the interfaces must tracked at every moment.

The use of Lagrangian particles provides an advantage in dealing with coupled problems of this type. Particles can convect all the needed variables plus materials properties at a negligible cost compared to Eulerian methods, in which each species must be convected separately. This chapter presents the basic tools used in this work to simulate multi-materials problems. The selected solving strategy is the Particle Finite Element Method second generation (PFEM-2) with fixed mesh. First, the main characteristics of this method are detailed for any general problem. Secondly, the special considerations for

the interfaces modelling are explained. Only the general methodology will be presented in this chapter, since the implementation strongly depends on the materials that have to be simulated.

The goal in the development of these strategies is to enhance the description of the interfaces between the different materials. This is imperative in many problems where one of the main outputs of the simulation is a detailed tracking of the interface.

2.2 The Finite Element Method

2.2.1 Discretization

The FEM is a popular method to solve continuum media problems, from fluid flows to solid structures. A strong point of the FEM is that since the variables are well defined at every point, it is possible to calculate gradients or integral quantities with a solid mathematical base.

Let us suppose a domain of interest Ω with boundaries $\delta\Omega$, as shown in Figure 2.1a. On the boundaries $\delta\Omega$ the appropriate Dirichlet $\delta\Omega_D$ and Neumann $\delta\Omega_N$ boundary conditions have to be set.

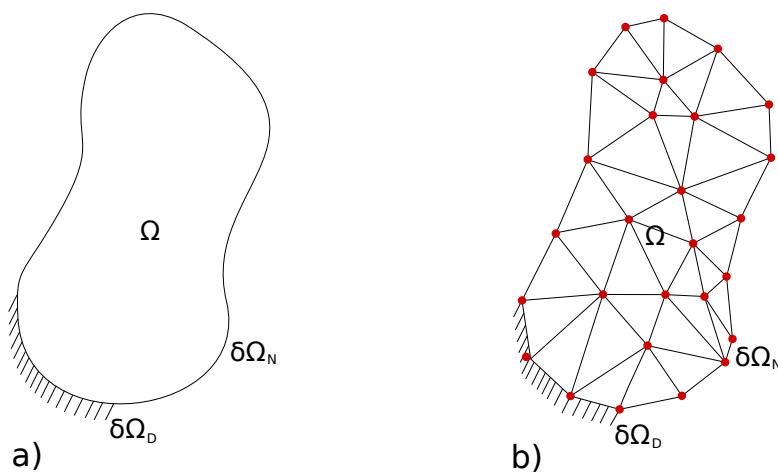


Figure 2.1: a) Real domain, b) Discretized Geometry

The FE discretization consists on subdividing the geometry into smaller portions called *elements*. In Figure 2.1b the discretized geometry can be observed. Inside each of these

elements, the quantities of interest will no longer be unknowns but rather interpolated quantities from the values in the points that define the elements, the *nodes*, shown as red dots. Therefore the problem reduces from the infinite number of unknowns in the continuum to the finite number of nodes that are used in the problem. At any arbitrary point inside Ω , the variables will be obtained using interpolation functions called *shape functions*. As an example, for a given scalar or vectorial variable λ in a three noded linear triangle (see Figure 2.2), its value can be expressed as:

$$\lambda(x, y) = \mathbf{N}(\mathbf{x}, \mathbf{y}) \mathbf{a}^{(e)} \quad (2.1)$$

where

$$\mathbf{N}(x, y) = [N_1(x, y), N_2(x, y), N_3(x, y)] \quad \mathbf{a}^{(e)} = \begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{Bmatrix}$$

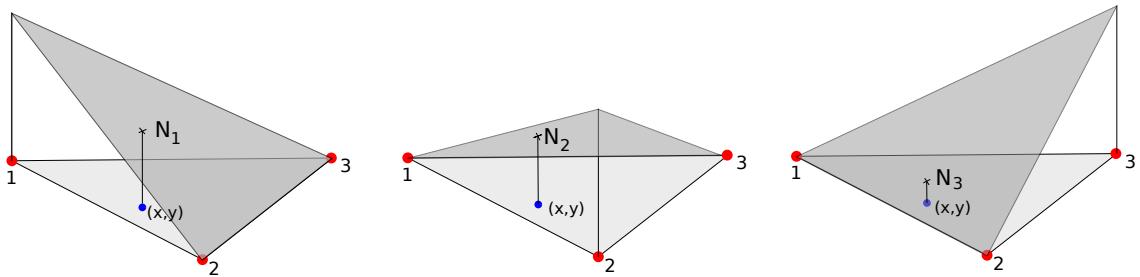


Figure 2.2: Three node triangle shape functions

The shape functions must fulfil the partition of unity in all the points, that is, at any point $N_1 + N_2 + N_3 + \dots + N_n = 1$. In this work only linear elements are used. For two dimensional problems 3-noded triangles are employed, while for tridimensional cases the geometry is discretized using 4-noded tetrahedra. This implies that only continuous interpolations of the variables are possible, unless modifications are added to the standard interpolation field.

2.2.2 Eulerian scalar transport

For most fluid dynamics applications, the domain of interest does not change with time; the outer boundaries remain static. With the goal of solving this type of problems, most FEM fluid solvers are based on Eulerian formulations. A fixed Eulerian mesh is

used and material points flows across it without causing any distortion to the grid. This strategy is optimal for problems in which the only unknowns are the fluid variables, the pressure and the velocity. However, for coupled problems in which additional variables have to be transported, a new set of convection equations have to be added to compute the new field.

Let us suppose that the velocity field is already known and we must find the new distribution of a scalar variable λ whose evolution can be represented by the classical convection-diffusion equation, where D is the diffusion coefficient.

$$\frac{\partial \lambda}{\partial t} = \nabla \cdot (D \nabla \lambda) - \mathbf{V} \cdot (\nabla \lambda) \quad (2.2)$$

Using finite differences in time for a time step Δt and a fully implicit scheme (Backward Euler), where the superscript n denotes the previous time step and $n + 1$ the current time step

$$\frac{\lambda^{n+1} - \lambda^n}{\Delta t} = \nabla \cdot (D \nabla \lambda^{n+1}) - \mathbf{V} \cdot (\nabla \lambda^{n+1}) \quad (2.3)$$

discretizing the geometry using the FE shape functions

$$\frac{N_i \lambda_i^{n+1} - N_i \lambda_i^n}{\Delta t} = \nabla \cdot (D \nabla N_i \lambda_i^{n+1}) - \mathbf{V} \cdot (\nabla N_i \lambda_i^{n+1}) \quad (2.4)$$

and using the Galerkin method for the weak form of the problem, we get the FE system that has to be solved at each time step. Integration by parts has been employed for the diffusion term and the boundary terms have been omitted for simplicity.

$$\int_{\Omega} \frac{N_j N_i \lambda_i^{n+1} - N_j N_i \lambda_i^n}{\Delta t} = - \int_{\Omega} \nabla N_j D \nabla N_i \lambda_i^{n+1} - \int_{\Omega} N_j \mathbf{V}^T (\nabla N_i \lambda_i^{n+1}) \quad (2.5)$$

and in matrix form:

$$\frac{1}{\Delta t} \mathbf{M} \boldsymbol{\lambda}^{n+1} - \frac{1}{\Delta t} \mathbf{M} \boldsymbol{\lambda}^n = -\mathbf{L} \boldsymbol{\lambda}^{n+1} - \mathbf{G}(\mathbf{V}) \boldsymbol{\lambda}^{n+1} \quad (2.6)$$

where the matrices components are

$$\begin{aligned}\mathbf{M}_{ij} &= \int_{\Omega} \mathbf{N}_i \mathbf{N}_j \\ \mathbf{L}_{ij} &= \int_{\Omega} \nabla \mathbf{N}_i \cdot D \cdot \nabla \mathbf{N}_j \\ \mathbf{G}(\mathbf{V})_{ij} &= \int_{\Omega} \mathbf{N}_j \mathbf{V}^T (\nabla \mathbf{N}_i)\end{aligned}$$

Unfortunately, the system (2.6) is only conditionally stable since $\nabla \lambda$ is not bounded. A classical example to test the stability of convection-diffusion algorithm is shown in Figure 2.3. The domain of length L has a constant speed V and Dirichlet boundary conditions at both ends. As initial condition, a linear variation between λ_1 and λ_2 is set.

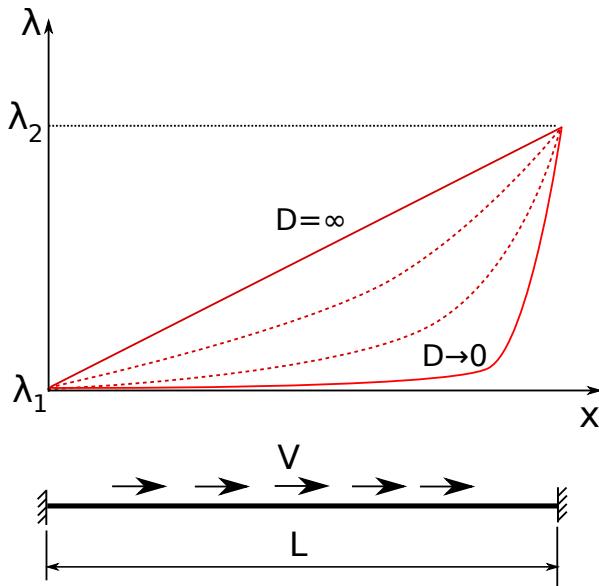


Figure 2.3: Simple 1-D Convection Diffusion problem

This problem must converge to a stationary solution that depends on the diffusion. In the upper part of Figure 2.3 the analytical solution is plotted for different values of the diffusion parameter D . As it can be seen in the plot, this example has the particularity that despite that the initial field is smooth, the stationary solution can be sharp, with structures similar to shock waves when the diffusion tends to zero.

Unfortunately, this property is not unique to this particular problem, but is a known

characteristic of convection-diffusion equations that is linked to the Peclet number (Oñate and Manzan, 2000). For Finite Elements, the elemental Peclet number can be useful to determine the stability of a scheme. This dimensionless number is defined as $Pe_{elem} = \frac{Vl}{2D}$, where l is the element size. When $Pe_{elem} > 1$ i.e., low diffusion and high velocities, the solution is no longer stable in the unstabilized FE implementation.

To prevent instabilities, the system must be stabilized in order to guarantee convergence. The problem lies in the fact that stabilization techniques consist on adding artificial diffusion. Increasing the diffusion to ensure the stability of the problem will always lead to solutions that are more diffusive than the original, unstabilized problem (Patankar, 1980).

2.3 The PFEM-2 method

2.3.1 Circumventing the difficulties of convective transport

The strategy used in this work is based on a mixed Eulerian-Lagrangian framework, combining advantages of both methods. This is achieved by using a set of particles combined with a fixed FE mesh in a solving strategy known as PFEM-2 (Idelsohn et al., 2013). The main steps of the method are obtained as follows. For a given Lagrangian particle p (a material point) and integrating across the streamlines, the position and the velocity at a given timestep $n + 1$ are defined as

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^t(\mathbf{x}_p^t) dt \quad (2.7a)$$

$$\mathbf{V}_p^{n+1} = \mathbf{V}_p^n + \int_n^{n+1} \mathbf{a}^t(\mathbf{x}_p^t) dt \quad (2.7b)$$

where \mathbf{V} is the velocity and \mathbf{a} is the acceleration and the superscript t indicates that both variables have to be evaluated continuously in time.

Equations (2.7) cannot be solved directly in they way they are presented. On one hand they are linked together since one is the derivative of the other. The convection of the particles, defined by (2.7a) depends on the velocity at every moment, while the forces(accelerations) of the momentum equation (2.7b) depend on the position of each of

the particles. On the other hand, the need of a time discretization to solve the equation implies that the variables are not known for any arbitrary time t , but only at discrete time steps $n - 1, n, n + 1$, etc.

For the convection of the material points, these issues can be assessed by using an explicit strategy in (2.7a), uncoupling it from the momentum equations. The main advantage of using Lagrangian particles is that the convection is obtained by simply moving the particles across the space. Note that in the following equation the position of the particle (\mathbf{x}_p^t) is computed approximately, by using explicit information.

$$\mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t) dt \quad (2.8)$$

Idelsohn et al. (2013) applied the same explicit strategy for the momentum Equation (2.7b). The resulting method was the PFEM-2 without implicit viscous correction, which showed good accuracy for low viscosity problems. Unfortunately, this strategy cannot be used for every problem. When the acting forces are large compared to the inertia of the material points, explicit algorithm yield to unstable schemes. The stability of such schemes can be evaluated by the Fourier number. For example, for a diffusion problem with a mesh size h , a time step Δt and a diffusion coefficient D , the limit condition for stable explicit schemes is

$$Fo = \frac{D\Delta t}{h^2} < \frac{1}{2} \quad (2.9)$$

Failing to fulfill this condition leads to unstable algorithms. On the other hand, implicit or semi-implicit algorithms can be unconditionally stable. Defining the parameter $\theta = 0$ for a fully explicit scheme and $\theta = 1$ for fully implicit, the momentum Equation (2.7b) is expressed as

$$\mathbf{V}_p^{n+1} \approx \mathbf{V}_p^n + (1 - \theta) \int_n^{n+1} \mathbf{a}^n(\mathbf{x}_p^t) dt + \theta \int_n^{n+1} \mathbf{a}^{n+1}(\mathbf{x}_p^t) dt \quad (2.10)$$

On the other hand, if the streamline integration was not to be used, the velocity in the new time step could be expressed as

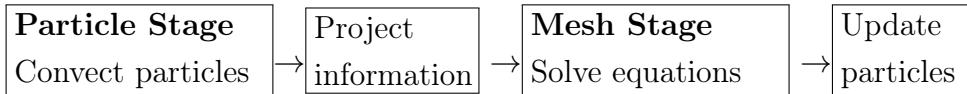
$$\mathbf{V}_p^{n+1} \approx \mathbf{V}_p^n + (1 - \theta)\mathbf{a}^n(\mathbf{x}_p^n)\Delta t + \theta\mathbf{a}^{n+1}(\mathbf{x}_p^{n+1})\Delta t \quad (2.11)$$

Combining the explicit contribution of Equation (2.10) with the implicit term of Equation (2.11), the general algorithm of the PFEM-2 with implicit correction is obtained. This simplifies the implementation of the method since the implicit terms are added directly in the final configuration. Then the final set of equations including the convection reads:

$$\mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t)dt \quad (2.12a)$$

$$\mathbf{V}_p^{n+1} \approx \mathbf{V}_p^n + (1 - \theta) \int_n^{n+1} \mathbf{a}^n(\mathbf{x}_p^t)dt + \theta\mathbf{a}^{n+1}(\mathbf{x}_p^{n+1})dt \quad (2.12b)$$

With only explicit contributions across the streamline integration, it is possible to devise a strategy with two distinct stages. First, the explicit streamline integration is performed. Having done this, results are projected into a fixed mesh to solve the implicit system of momentum equations and finally the corrections are passed to the Lagrangian material points. Then the complete scheme required to solve a step becomes:



The particles used in this scheme do not represent a fixed amount of mass, but rather material points with certain properties and velocity. This allows for a variable number of particles per element depending on the zone, to ensure a better accuracy on selected areas. It must be noted that in the algorithm implemented in this thesis, the particles are only used to transport information (solve the convective term), meaning that the forces will not be integrated across the streamlines ($\theta = 1$). The choice of a fully implicit algorithm was made to ensure that the method remains stable, even when simulating FSI and multi-fluid problems with large time steps. Figure 2.4 shows the velocity streamline integration for a single particle.

Remark 1: It is possible to imagine this explicit convection as the first step of a non linear iteration process. If an iterative process would be used until convergence, once the new velocity has been calculated, each particle should have its coordinates

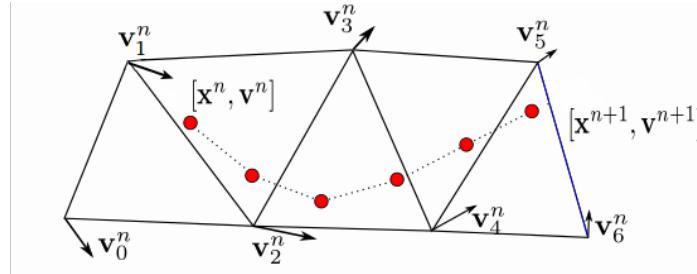


Figure 2.4: PFEM-2 streamline integration

resetted to the initial position and the streamline integration should be performed again to repeat all the tasks. However, practical experience suggests that the first iteration provides results that are accurate enough, with excellent matching between the PFEM-2 simulations and experimental data.

Remark 2: Always following the streamlines in single phase problems provides a good approximation on the convective term, as shown by Idelsohn et al. (2013). However in multiphase problems, in which the properties of the materials differ significantly, this approximation is no longer valid. As an example, the trajectory of a water particle inside a droplet clearly differs from the streamlines generated by both air and water, as shown in Figure 2.5. Estimating the new position with the integral of the previous step velocity would distort the shape of the droplet with an non-physical behaviour. In other words, the explicit streamlines do not provide a good representation for the trajectories. To avoid this, when a particle is not transiting its own phase, the streamline integration will be deactivated and the last known velocity of the correct phase will be used to complete the time integration.

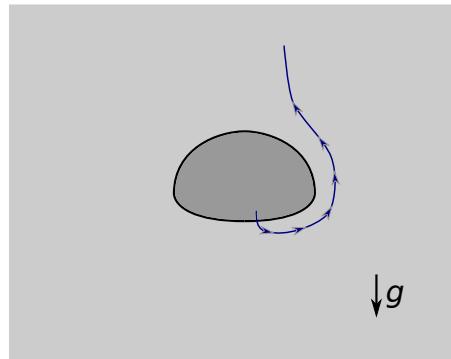


Figure 2.5: Streamlines of a falling water droplet

Once all the particles have been convected, the momentum equation (2.7b) has to be solved. Pure particle methods (as SPH) can solve this equation directly on the material points. However the irregular distribution of particles would lead to solutions that are heavily affected by the particle distribution. To avoid this behaviour, a fixed FE mesh is used in this work.

2.3.2 Transferring information onto the fixed mesh

In order to solve the Lagrangian equations in the mesh, first it is necessary to transfer the information from the particles. To do so, a projection algorithm is used. The simplest projection kernel consists on directly using the shape functions of the elements. As an example, a scalar variable λ is used.

Each i particle has an associated value λ_i . In Figure 2.6 a typical distribution of particles is shown. To calculate the value of λ_j in each j node, all the i particles in the neighbour elements to j will be used. The weighting function is defined as the standard shape function of the element for the node j in the position of each i particle. In this work an explicit algorithm was chosen to project the particles' information on the mesh. The advantage of explicit strategies is their high speed compared to implicit counterparts. However the drawback is that the projected field is heavily diffused compared to the data stored in the particles. Even in a single step going from particles-mesh-particles, it can be easily noticed the created diffusion.

$$\lambda_j = \frac{\sum_i^n \lambda_i N_i^j}{\sum_i^n N_i^j} \quad (2.13)$$

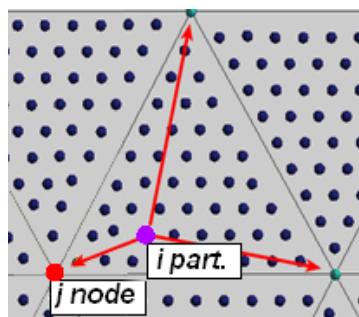


Figure 2.6: Contribution of i particle to j node

The same procedure is extended for all the variables, replacing the λ_i of the particle with the desired variable, either scalar, vectorial or tensorial. Using this kernel, the projected velocity can be written as 2.14. The ($\hat{\cdot}$) in the notation implies that it is the first approximation of the velocity (prior to the addition of any force), taking into account only the convecting term and neglecting all the other contributions. This velocity will be used as the starting point for the Finite Element Fixed Mesh solver.

$$\hat{\mathbf{V}}_j^{n+1} = \frac{\sum_i^n \mathbf{V}_i N_i^j}{\sum_i^n N_i^j} \quad (2.14)$$

It is worth noting that the elemental shape functions N_i are not the only possible kernel. As an example, the function proposed by Wendland (1995) assigns a higher weight to those particles that are closer to the node. Setting h as the element size and $u = R/h$, where R is the distance from the particle to the node, the Wendland kernel is written as (2.15). This kernel provides less *smoothing* to the projection while remaining explicit, but implies that the solution on the closest particle to the node can be taken as exact.

$$W_i^j = \frac{7}{4\pi h^2} (1 + 2u)(1 - u/2)^4; \quad (2.15)$$

Implicit Kernels are also possible, although they are not presented here due its higher computational cost, which requires the solving on a matrix system.

2.3.3 Continuum equations

As explained in the last sections, the use of particles allows us to omit the convective term in the momentum equations, thus becoming a Lagrangian frame. The system of equations to be solved in the physical domain Ω is obtained by coupling the momentum balance for a general material with a mass conservation equation. Together, these two equations provide exact solutions for problems in which thermal terms are considered to be uncoupled from the mechanical equations

$$\rho \frac{D(\mathbf{V})}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho g \quad \text{in } \Omega \quad (2.16a)$$

$$\frac{Dp}{Dt} + \kappa \nabla \cdot \mathbf{V} = 0 \quad \text{in } \Omega \quad (2.16b)$$

Equations (2.16a) are complemented by the standard boundary conditions of prescribed velocities and prescribed tractions at the Dirichlet (Γ_u) boundary conditions and Neumann (Γ_t) boundary conditions, respectively. Also an appropriate constitutive equation is needed to compute the stresses from the deformations.

Equations (4.1) hold for any isotropic material, both fluids and solids, where ρ is the density, κ the compressibility modulus, \mathbf{V} the velocity vector, $\boldsymbol{\sigma}$ the stress tensor and \mathbf{g} the mass force vector. In this work fluids are treated as incompressible, meaning that $\kappa = \infty$ and, therefore, the second equation simplifies for fluids to $\nabla \cdot \mathbf{V} = 0$. It must be noted that anisotropic materials cannot be simulated with this model due to the expression used for the pressure equation, in which the compressibility modulus does not change with the direction of the deformation.

2.3.4 Transferring corrections back to the particles

After the information of the Lagrangian domain has been projected and the system of equations has been solved on the mesh, the velocity and all the variables of the particles have to be updated. At this stage it is important not to replace the particle's properties with the new, updated mesh variables but only a correction. The reason for this is simple; no matter how good the kernel algorithm may be, it is never diffusion-free. Replacing the variables of the particles would destroy valuable information, leading to excessive artificial diffusion of the strategy. This also applies to implicit projection algorithms; despite they offer a clear advantage in terms of accuracy, they still create a different field to the original particle distribution. Based on this, the update stage on the particle will only transfer the variation in the velocity (or any other variable) calculated in the mesh, that is:

$$\delta \mathbf{V}_j^{n+1} = \mathbf{V}_j^{n+1} - \hat{\mathbf{V}}_j^{n+1} \quad (2.17)$$

Transferring only a correction instead of the interpolated value of the variable does not only reduce artificial diffusion but also allows for a different scale 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

it later. This is a clear advantage over Eulerian methods, in which changes of mesh size translate into loss of information.

2.4 Interface Description

The base linear shape functions used in this work are well suited to represent variables whose field is smooth. In other words, there can be no discontinuities on the variables nor its gradients. However in multi-material simulations there can be sharp interfaces which translate into abrupt changes in the unknowns of the problems.

To tackle this problem, two distinct tools will be needed. On one side, a strategy to detect the exact location of the interfaces is used. Having done this, the FE field will be modified in order to improve the interpolation of the variables. The new field must contain the space of the real solution to obtain optimal convergence (Ausas et al., 2012).

2.4.1 Interface capturing

Interface capturing between two materials

To enhance the description of the interfaces in a multi-material problem, it is first necessary to correctly define the location of the boundaries between the different phases. The particles can store auxiliary variables to ease the task of detecting sharp changes in the properties.

Each particle has an associated material and the interface should be located where the material properties change. The distribution of particles inside each element can define complicated curves that become impossible to manage with simple shape functions due to the large number of particles. In Figure 2.7 some particles are in the "wrong" side of the interface. For this reason the instantaneous local interface inside each element is simply defined by a line (or a plane in 3D) taking into account a weighted average using the shape functions. Assuming only two materials are present (e.g. air and water), the lighter one is given a positive sign and the denser one a negative sign. In order to compute the location of the interface, a pseudo φ level set function is used, which defines the interface at the isosurface $\varphi = 0$. To calculate the value of φ in each j node, all the i

particles in the neighbour elements to j will be used. Using the methodology described in the introduction, the weighting function is defined as the standard shape function of the element for the node j in the position of each i particle. Once the contribution of all the particles has been added, the interface location is trivial using Equation (2.18).

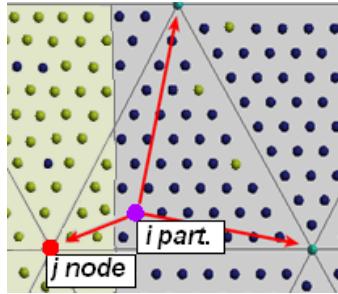


Figure 2.7: Contribution of i particle to j node

$$\varphi_j = \frac{\sum_i^n \text{sign}_i N_i^j}{\sum_i^n N_i^j} \quad (2.18)$$

2.4.2 Detecting the boundaries of bodies

When only part of the mesh is occupied by material bodies, an algorithm is necessary to detect their boundaries. This case is represented in Figure 2.8. After having convected the particles, the information is scattered across the space only at these discrete material points, the particles. However the domain of interest is the continuous grey area, so the solution implies determining the boundaries by relying only in the particles' positions. This section describes an approximate methodology to locate the correct location of these boundaries.

The correct location of the boundaries is needed in order to compute correctly the mass and other extensive properties of the Lagrangian bodies, since each particle does not represent a fixed amount of mass. The following steps can be followed to locate the interfaces between active and inactive areas of the mesh.

1. Detect first layer of inactive elements. These elements can be detected because they have contribution from particles only in some of their nodes. Shown in yellow in Figure 2.9.

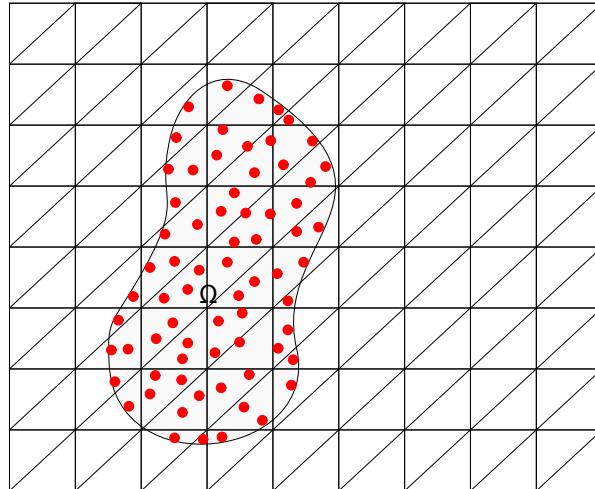


Figure 2.8: Lagrangian body in fixed mesh

2. Flag the first layer of nodes outside the boundaries. These nodes are those that have contribution from particles but belong to the first layer of inactive elements (step 1.). Shown in yellow in 2.9.
3. Flag interface elements. These elements are easily identified because all their nodes have contribution from particles and also they have nodes flagged in step 2. Shown in blue in Figure 2.9.

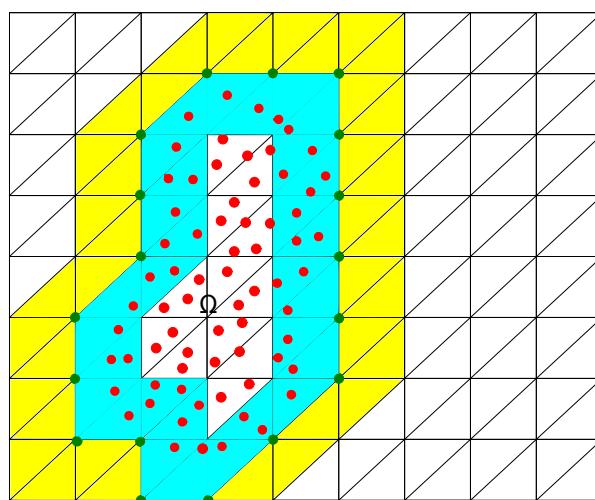


Figure 2.9: Boundaries detection. In yellow, first layer of inactive elements; yellow: first layers of nodes outside the body; blue: interface elements

Having detected the interface elements, now it is necessary to compute the distance from the nodes that define the interface elements to the exact location of the interface. In this sense, a distance function φ will be used. The first layer of internal node will have negative distances while the first layer of external nodes will have positive values.

4. Loop all the nodes of interface elements.

- Set a very large positive distance to external nodes.
- Set a very small negative distance to internal nodes.

5. Loop all the particles inside interface elements.

- Find the closest particles to external nodes. The nodal distances will be directly the node-particle distance. Shown in Figure 2.10 for external nodes j and k , the distance function is φ^+ .
- For the negative distances, it is not possible to use the particle that is furthest away, therefore a different approach must be chosen. Negative distances will be calculated as the average distance from the particles used in φ^+ to nodes to the opposite edge of the element. Shown in Figure 2.10 as $-\varphi^-$. For the external node j , this value can be computed as $\varphi^- = \varphi^+ N_j / (1 - N_j)$, where N_j is the value of the j shape function in the closest particle position. Finally to average the distance, the shape functions will be used: $\varphi_i = -\frac{\sum \varphi^+ N_i}{\sum N_i}$, where N_i is the i shape function in the used particles.

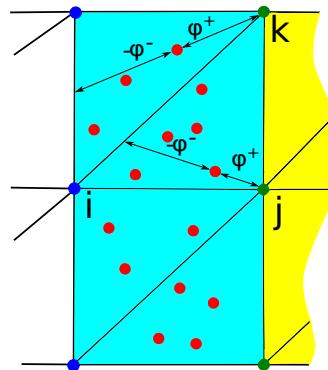


Figure 2.10: Setting particle distance

The previous steps provide an approximate definition of the interface. If the interface is drawn from the closest particles chosen by the algorithm in Figure 2.10, the interface

will not match exactly the boundaries of the particles. In Figure 2.11 the location of the free surface is drawn. As it can be seen, particles can be left outside the free surface.

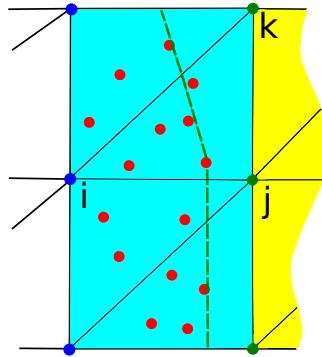


Figure 2.11: Particle distribution and approximate location of the free surface, shown in yellow.

The problem lies in using the Euclidean distance between the particles and the nodes, while in reality it should be the distance from the plane defined by the continuous function $\varphi = 0$ to the nodes. However, this cannot be computed directly at the beginning of the loop since there is no such thing as a plane, only a discrete distribution of particles. To avoid an iterative process and considering that this solution is good enough, no further modifications will be done on the distance function.

2.4.3 Improving the fields of unknowns: Enrichments

Standard FE formulations only allow for continuous interpolations of the variables inside the elements. However discontinuities in the material properties can cause jumps in the gradients of the variables and even complete discontinuities. The chosen strategy must take into account these special cases in order to reproduce correctly the physics of the real problem.

To overcome this issue the Finite Element space must be modified to allow for a more accurate reproduction of the solution. A first alternative would be remeshing the zone where the interface passes through. This would lead to the exact solution but would require adding new degrees of freedom (DoFs) in the new nodes and new connectivities. This strategy, despite possible, is computationally expensive since the system must be resized and new space must be allocated in the memory, task that is likely to be slower than the solution of the system itself.

An alternative to a new mesh is the enrichment of the FE space. Enriching consist on creating new DoFs on each of the interface elements and then statically condensing the new unknowns (Felippa, 2004). Enrichment shape functions will be used to improve the standard, continuous field of the finite elements. When the FE space is enriched, extra shape functions are added to the interpolation field. The combination of the original shape functions and the new ones must contain the space that we want to reproduce. Failing to do so will translate into poor convergence.

In this work two alternatives are presented: gradient discontinuities and complete discontinuities. Using one or the other highly depends on the physics of the problem. In some cases adding extra freedom on the interface only increases the computational time to assemble the system. However if the system is over relaxed with no balance equations to ensure stability, the solution obtained can be non-physical or diverge. For these reasons it is important to chose the correct enrichment functions based on the required field for the problem.

Gradient Discontinuities

For cases in which the material discontinuities only produce a kink in the gradients of the unknowns, it is necessary to add new shape functions that are continuous in the variables but discontinuous in the gradient. The most simple function that can fulfil this criteria is the one proposed by Coppola-Owen and Codina (2005). This function was developed to improve the pressure field in two-fluid problems with low viscosities, such as mould casting simulations. In case, the pressure is still continuous but with a steep change in the gradient, meaning that no discontinuities in this variable are needed. The addition of this function improves considerably mass conservation and the convergence of the solution. Figure 2.12 shows the gradient enrichment for the two-dimensional case. The concept is easily extendible to tridimensional problems; the function is constructed by creating a continuous function whose gradient must be constant on each side of the interface.

This enrichment function must be used together with the standard linear interpolation functions of the elements. The combination of these two fields can capture a smooth field with a sharp change in the slope at the location of the interface.

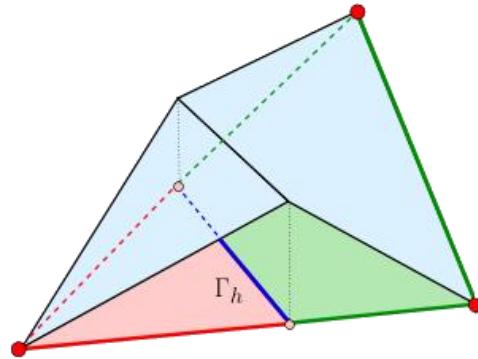


Figure 2.12: Enrichments for discontinuous gradient

Variables Discontinuities

When the variables can have sharp discontinuities at the interface, a gradient enrichment is not enough to capture the solution. In these cases more freedom must be given to the unknowns in order to obtain an accurate result, as shown by Ausas et al. (2012). Following this line, three enrichments shape functions plus special shape functions to replace the standard pressure field are used. Figure 2.13 shows that a total of 6 functions are used for triangles, with the goal of fully uncoupling the pressure from both sides. For tridimensional problems the concept is easily extrapolated, using a total of 8 shape functions in split tetrahedra.

To create this function, first the standard shape functions in the interface elements are replaced by their discontinuous counterparts. These functions are basically the same as the original ones, but the integrals are calculated only with the contribution of the partitions whose sign matches the sign of the node. On the other hand, when partitions and nodes have different signs, the contribution is added to the enrichments functions i^* , j^* or k^* . In other words, the original shape functions are split in two independent functions across the interface. This allows more freedom in the pressure field, while retaining the partition of unity (Zienkiewicz et al., 2006). Conserving the partition of unity is useful when reconstructing the fields after the projection stage, as it will be seen in the next chapters.

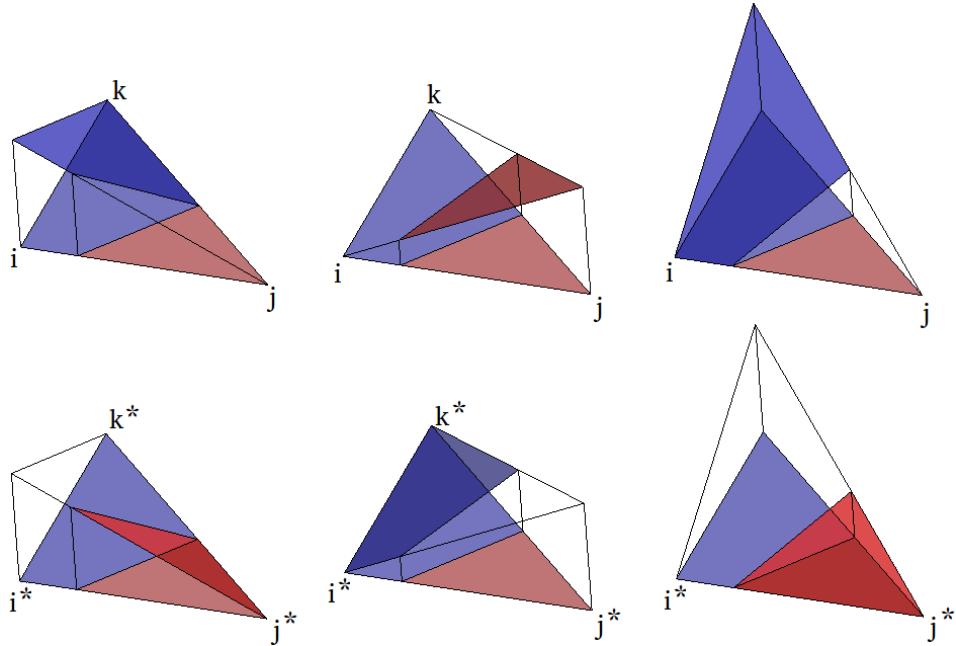


Figure 2.13: Pressure shape functions: Top: Replacement functions.
Bottom: Enrichments

2.4.4 Condensing the new degrees of freedom

Using enrichment shape functions allows to capture more accurately the field of unknowns. The cost of this is that the new problem has a different number of unknowns compared to the original problem. As an example, adding the gradient enrichment adds one extra unknown to the problem per every interface element of the geometry.

Despite the number of unknowns is larger than in the original system, the associated computational cost to solve the sparse system would not be significantly larger. As stated in the last section, the major time increase would be the result of resizing the matrix and changing the graph of connectivities in this matrix. Condensing allows to circumvent this resizing. This process consists on eliminating part of the degrees of freedom (the enrichments) by performing matrix operations in the system of equations. The reduced algebraic system is solved and afterwards the condensed unknowns can be recovered.

Problem statement

Let us suppose a simple thermal (or any other scalar) problem to be solved. Two materials and no loads will be considered to reduce as much as possible the complexity of the problem, as shown in Figure 2.14.

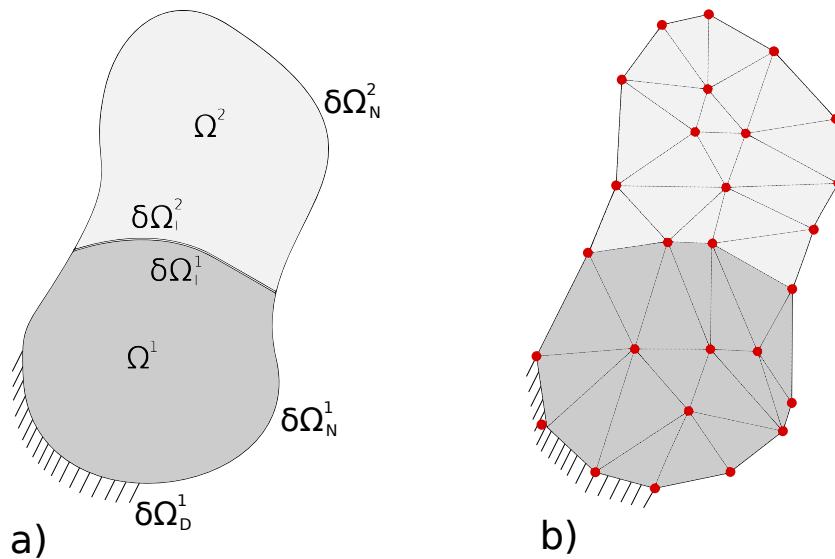


Figure 2.14: Two-phase thermal problem domain.
Left: real problem. Right, discretized geometry

In each of the phases, the following equations have to be fulfilled, where the diffusion is defined by the coefficients D_1 in Ω_1 and D_2 in Ω_2 . Also appropriate boundary conditions have to be set both in the limits of the body and in the interface between the phases. Homogeneous (adiabatic) condition will be imposed for the Neumann boundaries.

$$\begin{aligned}
 \nabla \cdot (D_1 \nabla T) &= 0 && \text{in } \Omega^1 \\
 \mathbf{n}_1 \cdot D_1 \nabla T &= 0 && \text{on } \delta\Omega_N^1 \\
 \mathbf{n}_1 \cdot D_1 \nabla T &= q && \text{on } \delta\Omega_I^1 \\
 T &= \bar{T} && \text{on } \delta\Omega_D^1 \\
 \nabla \cdot (D_2 \nabla T) &= 0 && \text{in } \Omega_2 \\
 \mathbf{n}_2 \cdot D_2 \nabla T &= 0 && \text{on } \delta\Omega_N^2 \\
 \mathbf{n}_2 \cdot D_2 \nabla T &= q && \text{on } \delta\Omega_I^2
 \end{aligned} \tag{2.19}$$

Where \mathbf{n}_1 and \mathbf{n}_2 are the normal vectors to the edges of each subdomain. Since the flux q across the interface has to be conserved, the equation at Ω_I reads:

$$\mathbf{n}_1 \cdot D_1 \nabla T = \mathbf{n}_2 \cdot D_2 \nabla T \quad \text{in } \delta\Omega_I \quad (2.20)$$

Weak form of the problem

Using the Finite Elements shape functions to define the unknowns of the problem and then Galerkin to obtain the weak form, the problem in the subdomains Ω^1 and Ω^2 can be written as:

$$\begin{aligned} \int_{\Omega^1} \mathbf{N} \nabla \cdot (D_1 \nabla \mathbf{T}) &= 0 && \text{in } \Omega^1 \\ \int_{\Omega^2} \mathbf{N} \nabla \cdot (D_2 \nabla \mathbf{T}) &= 0 && \text{in } \Omega^2 \end{aligned} \quad (2.21)$$

where $\mathbf{T} = \sum_i N_i T_i$. Splitting in n elements in the first subdomain and using m elements in subdomain 2, the system is obtained by assembling the contribution of each j element:

$$\begin{aligned} \sum_{j=1}^n \int_{\Omega^j} \mathbf{N} \nabla \cdot (D_1 \nabla \mathbf{T}) &= 0 && \text{in } \Omega^1 \\ \sum_{j=1}^m \int_{\Omega^j} \mathbf{N} \nabla \cdot (D_2 \nabla \mathbf{T}) &= 0 && \text{in } \Omega^2 \end{aligned} \quad (2.22)$$

The problem with this equation is that it requires a second derivative of the linear shape functions used in the temperature field. Since this would lead to the trivial solution ($= 0$), modifications have to be introduced to solve the systems of equations. Writing the vector of unknowns $\mathbf{a} = [T_1, T_2, \dots, T_n]$ and integrating by parts in both domains we obtain:

$$\begin{aligned} \sum_{j=1}^n \left[- \int_{\Omega^j} \nabla \mathbf{N} (D_1 \nabla \mathbf{N}) \mathbf{a} + \int_{\delta\Omega^j} \mathbf{N} \mathbf{n}_1 \cdot D_1 \nabla \mathbf{N} \mathbf{a} \right] &= 0 \\ \sum_{j=1}^m \left[- \int_{\Omega^j} \nabla \mathbf{N} (D_2 \nabla \mathbf{N}) \mathbf{a} + \int_{\delta\Omega^j} \mathbf{N} \mathbf{n}_2 \cdot D_2 \nabla \mathbf{N} \mathbf{a} \right] &= 0 \end{aligned} \quad (2.23)$$

For the chosen Neumann homogeneous boundary conditions these terms have to be zero $\int_{\delta\Omega} \mathbf{N} \cdot D\nabla\mathbf{T} = 0$, so this will translate into weakly imposed BC. On the other hand, for the internal boundaries $\delta\Omega_I$, the boundary terms to guarantee flux conservation lead to:

$$\int_{\delta\Omega_I^1} \mathbf{N} \cdot \mathbf{n}_1 \cdot D_1 \nabla\mathbf{T} = \int_{\delta\Omega_I^2} \mathbf{N} \cdot \mathbf{n}_2 \cdot D_2 \nabla\mathbf{T} \quad (2.24)$$

Equation (2.24) is the weak form of the exact condition (2.20) that must be fulfilled in the internal boundaries. Since the functions \mathbf{T} are the same across both sides of the interface, omitting these terms from (2.23) will lead to the correct approximation of the weak form:

$$\begin{aligned} \sum_{j=1}^n \int_{\Omega^j} \nabla\mathbf{N} \cdot (D_1 \nabla\mathbf{T}) &= 0 \\ \sum_{j=1}^m \int_{\Omega^j} \nabla\mathbf{N} \cdot (D_2 \nabla\mathbf{T}) &= 0 \end{aligned} \quad (2.25)$$

The structure of this problem is the solution of a Laplacian, in which the contribution of each element is multiplied by the diffusion coefficient D of the element. In matrix form, using \mathbf{L} to express the Laplacian, the problem becomes:

$$\mathbf{L}(D) \mathbf{T} = 0 \quad (2.26)$$

The system (2.26) represents a good approximation to the continuous problem when the interfaces can be correctly defined with the selected discretized geometry. Equation (2.20) shows that when the material conductivity changes abruptly, the gradient of the temperature will be discontinuous. Since the selected shape functions are piecewise linear but there is no requirement on the continuity of the derivatives between elements, the solution will be adequate at all points and at the interfaces.

Weak form with enrichments

As shown in the last section, standard finite elements provide good results when the discretized geometry boundaries match the material interfaces. However when the in-

terfaces are located across elements, the linear shape functions will not be able to capture the sharp discontinuity in the gradient of the solution. This is critical since one of the goals of the PFEM-2 is avoiding remeshing or distorting the mesh in order to match the interfaces, as shown in Figure 2.15.

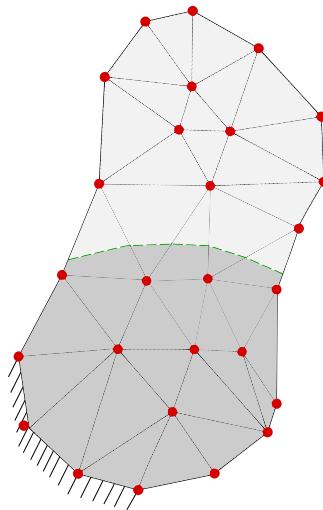


Figure 2.15: Two-phase problem with non-matching interfaces. Interface marked with the green line.

The use of enrichments to capture discontinuities in the gradients allows to override this limitation while maintaining the size of the problem. To do so, the first step is writing the full weak form of the system (2.23) but adding the enrichment DoF. The standard shape functions will be labelled $\mathbf{N}_N = [N_1, N_2, N_3]$ and the enrichment function described in Section 2.4.3 with $\mathbf{N}_* = [N_*]$. For n elements in the domain, the enriched formulation becomes:

$$\sum_{j=1}^n \left[- \int_{\Omega^j} \nabla[\mathbf{N}_N, \mathbf{N}_*] (D_1 \nabla[\mathbf{N}_N, \mathbf{N}_*]) [\mathbf{a}_N, \mathbf{a}_*]^T + \int_{\delta\Omega^j} \mathbf{N} \mathbf{n}_1 \cdot D\nabla[\mathbf{N}_N, \mathbf{N}_*] [\mathbf{a}_N, \mathbf{a}_*]^T \right] = 0 \quad (2.27)$$

The Neumann boundary terms at the interfaces between elements have to fulfil flux conservation. Writing the contribution of both standard and enrichment shape functions, the flux conservation is given by the following equation, where i and j are two neighbour

enriched elements, as shown in Figure 2.16.

$$\int_{\delta\Omega_i} \mathbf{N} \cdot \mathbf{n}_i \cdot D_i \nabla[\mathbf{T}_N, \mathbf{T}_*] = \int_{\delta\Omega_j} \mathbf{N} \cdot \mathbf{n}_j \cdot D_j \nabla[\mathbf{T}_N, \mathbf{T}_*] \quad (2.28)$$

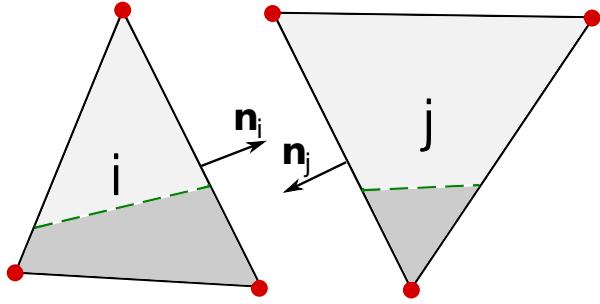


Figure 2.16: Two neighbour elements and their normals

Following the same strategy as in non-enriched elements, these terms will be omitted from the system of equations (2.28), obtaining an extended Laplacian matrix. For each j element the contribution will be:

$$-\int_{\Omega^j} \nabla[\mathbf{N}_N, \mathbf{N}_*] (D_1 \nabla[\mathbf{N}, \mathbf{N}_*]) [\mathbf{a}, a_*]^T = 0 \quad (2.29)$$

And in matrix form:

$$\begin{bmatrix} \mathbf{L}_{NN} & \mathbf{L}_{N*} \\ \mathbf{L}_{*N} & \mathbf{L}_{**} \end{bmatrix} \begin{bmatrix} \mathbf{a}_N \\ a_* \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (2.30)$$

Supposing the 1×1 \mathbf{L}_{**} matrix is invertible, the new unknown can be expressed as:

$$a_* = \mathbf{L}_{**}^{-1} (-\mathbf{L}_{*N} \mathbf{a}_N) \quad (2.31)$$

Replacing (2.31) into (2.30), the enrichment DoF is condensed and the system size recovers its original size. The modified local matrix with the condensed gradient can be expressed as:

$$\tilde{\mathbf{L}} \mathbf{a}_N = \mathbf{0} \quad \text{where} \quad \tilde{\mathbf{L}} = \mathbf{L}_{NN} - \mathbf{L}_{N*} \mathbf{L}_{**}^{-1} \mathbf{L}_{*N} \quad (2.32)$$

The global system will be assembled with this contribution and then solved. After solving the complete system of equations, the a_* of each element can be recovered using Equation (2.31).

Sample heat diffusion enriched problem

To test the accuracy of adding and condensing the gradient enrichment DoF, the following simple example is tested. In Figure 2.17 the geometry of the problem is presented. A pseudo 1-D case is chosen in order to obtain easily the analytical solution. Since there are no source terms in the domain, the heat flux across the length of the geometry, the heat flux must remain constant:

$$\begin{aligned} \nabla T_{left} D_{left} &= \nabla T_{right} D_{right} \\ \frac{\nabla T_{left}}{\nabla T_{right}} &= \frac{1}{10} \end{aligned} \quad (2.33)$$

Taking into account that the interface is located at exactly the middle of the domain, the change in temperature at the right side must be ten times the one at the left side: $\delta T_{right} = 10 \delta T_{left}$. Then the exact solution for the temperature at the interface is $T_{interface} = 10$.

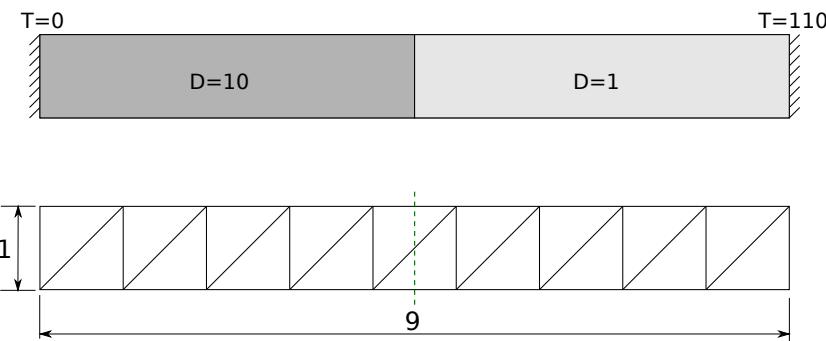


Figure 2.17: Enrichment test example geometry. Up: real domain, bottom: discretized problem. The enrichment location is marked with the green dotted line

In principle the enrichment solution should be able to capture the *exact* solution since the modified finite element space contains the linear field with a discontinuity in the gradient at the interface. Figure 2.18 shows the exact solution and the enriched and

non enriched approximations. In the solution using the gradient discontinuity, the new DoFs have not been recovered after solving the system, so only the linear interpolation is being plotted between $x = 4$ and $x = 5$.

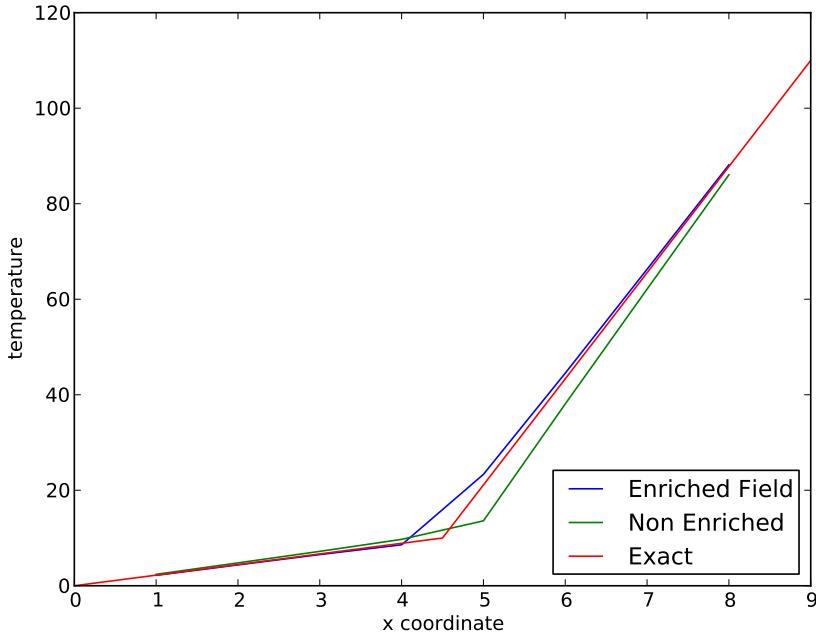


Figure 2.18: Thermal problem solution for $h = 1$ mesh

The results of Figure 2.18 clearly show that despite more accurate, the enriched solution is not exact. Given that the improved temperature field actually contains perfectly the real, continuum solution, this might seem unexpected. The reason behind this is that, despite the chosen shape functions are adequate to capture the solution, an incorrect hypothesis was introduced in the formulation when condensing the enrichment DoFs. Recalling the continuity of fluxes across the elemental boundaries in weak form for two neighbour elements i and j :

$$\int_{\delta\Omega_i} \mathbf{N} \cdot \mathbf{n}_i \cdot D_i \nabla[\mathbf{T}_N, \mathbf{T}_*] = \int_{\delta\Omega_j} \mathbf{N} \cdot \mathbf{n}_j \cdot D_j \nabla[\mathbf{T}_N, \mathbf{T}_*] \quad (2.34)$$

The problem lies in the fact that when the enrichment DoFs \mathbf{T}_* are condensed, continuity is lost between the elemental boundaries. This translates into different boundary conditions, conserving correctly the heat flux generated by the standard shape functions

but imposing homogeneous Neumann BC for \mathbf{T}_* :

$$\begin{aligned} \int_{\delta\Omega_i} \mathbf{N} \cdot \mathbf{n}_i \cdot D_i \nabla [\mathbf{T}_N] &= \int_{\delta\Omega_j} \mathbf{N} \cdot \mathbf{n}_j \cdot D_j \nabla [\mathbf{T}_N] \\ \int_{\delta\Omega_i} \mathbf{N} \cdot \mathbf{n}_i \cdot D_i \nabla [\mathbf{T}_*] &= 0 \\ \int_{\delta\Omega_j} \mathbf{N} \cdot \mathbf{n}_j \cdot D_j \nabla [\mathbf{T}_*] &= 0 \end{aligned} \quad (2.35)$$

This is clearly incorrect since physically there *will* be heat flux produced by the enrichment temperature \mathbf{T}_* . Despite adding these boundary terms should, in principle, recover the quadratic convergence of linear elements, they will not be included in this work. The reason behind this is maintaining the symmetry of the system, thus allowing to use iterative symmetric linear solvers to reduce the computational times. A different alternative will be presented in Chapter 3 to circumvent this limitation for the Poisson pressure equation.

To evaluate the error caused by the lack of boundary terms and the convergence of the enriched and non enriched elements, different meshes were created with structured grids, from $h = 1$ to $h = 0.001$, where h is the element size. The error is computed using the L_2 error norm in regions $h < 4$ and $h > 5$ to avoid computing the contribution of split elements. This way the error can be calculated without recovering the condensed DoFs.

$$L_2 \text{ error} = \sqrt{\frac{\int_{\Omega} (T_{FEM} - T_{exact})^2 d\Omega}{\int_{\Omega} d\Omega}} \quad (2.36)$$

The L_2 error was computed in the discretized domain using the formula (2.37), where n is the number of used elements, m the number of Gauss points per element and W the three Gauss point weights: $W = \text{area}/3$. The *exact* and FE solutions must be evaluated at each Gauss Point.

$$L_2 \text{ error} = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^m (T_{FEM} - T_{exact})^2 W_j}{\sum_{i=1}^n \sum_{j=1}^m W_j}} \quad (2.37)$$

The convergence for both enriched and non enriched solutions can be seen in Figure 2.19. As observed in the graph, the addition of the enrichment shape functions decreases al-

most one order of magnitude the error norm. However, despite this improvement, the convergence with mesh size still remains linear, below the expected quadratic convergence for linear elements. Even worse, in this case the enriched solution should lead to zero error in all the meshes since the analytical solution is contained in the FE space.

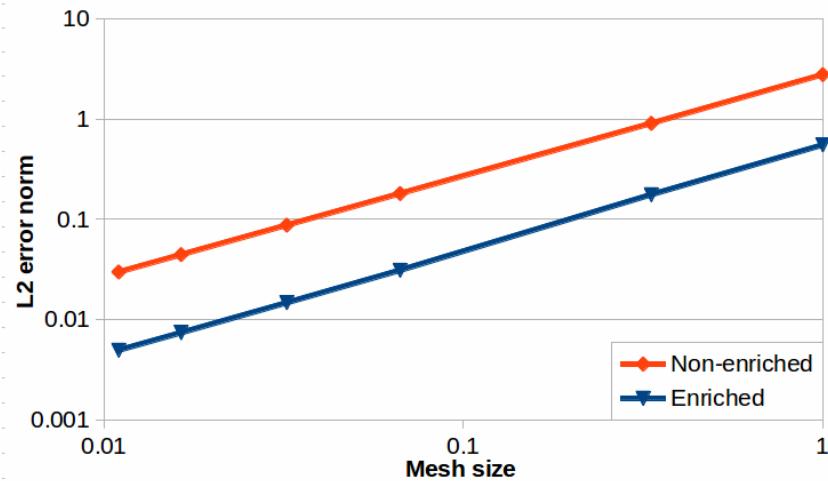


Figure 2.19: Thermal problem mesh convergence

This simple example shows the importance of correctly treating the boundary conditions at the inter-elemental boundaries. Wrong hypothesis, even if they are set in only one shape function of a small fraction of the total elements of the domain can diminish the benefits of the improved model.

2.5 Particle management

2.5.1 Particle reseeding

As stated in previous sections, the particles of the PFEM-2 method do not represent a fixed amount of mass but are rather material points whose mass depends on the number of particles in the area. In large deformation problems (such as fluids) the relative displacements between particles will lead to zones with a different number of particles per unit area. Since areas of a low number of particles per element are not desired in order to ensure a good projection to the mesh, a particle reseeding algorithm is required.

In the paper by Gimenez et al. (2012), a strategy was described to assure that a mini-

mum number of particles per element (around 3) is kept at all times without requiring an excessively large set of particles. The method consists on checking the number of particles before the projection stage. If it is too low, particles are seeded in the required areas and the streamlines are followed in the inverse way to obtain the properties in the origin. This strategy provides good results for single phase fluids but, as stated previously, the streamlines do not always represent a good approximation for the convective term in multiphase problems.

In this work, the reseeding strategy will be simplified to avoid the use of the streamlines. The particles will be seeded at the end of each time step using the updated nodal information. The drawback of this strategy is that a larger number of particles will have to be used to ensure that a minimum number of particles is present at the projection stage. In practical terms, this means that the number of particles will be doubled compared to the single fluid strategy, requiring at least 6 particles per element at all times. It must be noted that, even if this strategy reduces the chance of having elements without enough particles at the projection stage, this situation can still happen if the time-steps are extremely large. If this happens, the inverse streamline integration by Gimenez et al. (2012) will be used to seed particles in empty elements. The complete algorithm reads:

- Step 1)** Convect the particles using the streamlines
- Step 2)** If empty elements → Reseed using inverse streamlines
- Step 3)** Project information into the mesh,
- Step 4)** Assemble and solve system of equations with enrichments
- Step 5)** Reseed with updated information
- Step 6)** Update particles information

2.5.2 Particle database

The large number of particles used in the scheme demands an fast and efficient particle database. Failing to do so would lead to an algorithm that loses the potential advantages of using a combined Lagrangian particles - Fixed mesh strategy. A key aspect

to consider in the design of efficient computational tools is that memory accesses are much more expensive than computations. In fact, the speed of a fluid mechanics solver is bounded by the memory access bandwidth rather than the processor speed, as shown by Mora (2012). This is of particular importance for the PFEM-2 algorithm due to the large number of particles. Memory accesses must be reduced to a minimum to obtain an fast algorithm. In Figure 2.20 the structure of the PFEM-2 database is shown. Each element has direct access to the particles that are inside of it. This database is updated at each time step after the convection stage, so that the elements have updated information of the particle inventory at all times. The strategy also employs a bin search algorithm (Samet, 1990) to speed up the location of particles within the mesh. The Bin structure is a regular grid. Each cell contains the elements that are inside. So in order to locate a particle, instead of searching through all the elements of the domain, only the ones in the cell are checked, reducing considerably the searching time.

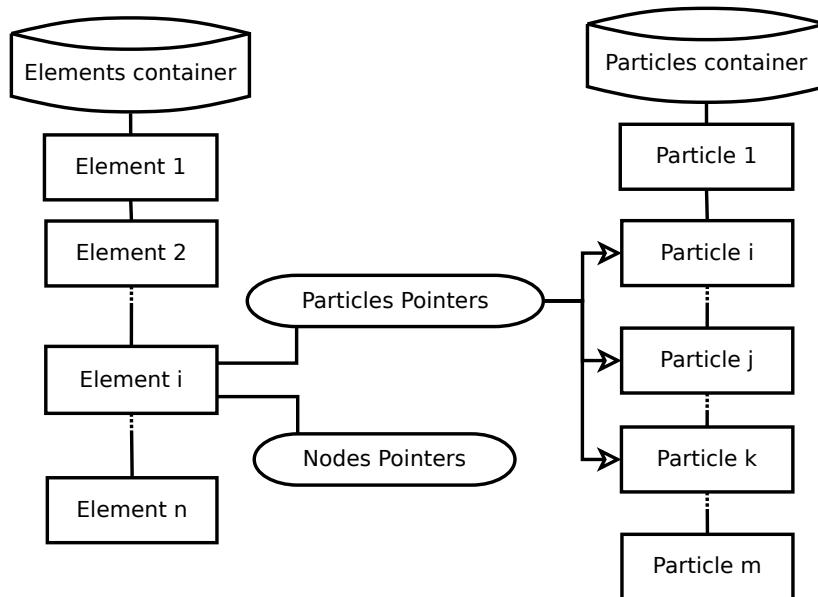


Figure 2.20: Elements and Particles databases

To do all the particle update operations, instead of looping in sequential order through the particle database (e.g. Particle1, Particle2...) , these tasks will be done following the elemental database. Beginning with Element 1, all the particles inside this element will be updated. After this, the particles of Element 2 will be updated and so forth. Knowing

which particles are inside each elements reduces considerably the memory accesses since to do the particle update, nodal information is needed. If a sequential order in the particle database was to be followed, multiple accesses to the nodal database would be needed. In Finite Elements this is particularly expensive since the non-structured grid implies that the nodes that define each element cannot be known a priori and therefore they have to be stored, increasing the database access time.

Accessing the particles using the elemental database has an added benefit in the streamline integration algorithm. Since the particles inside the same element are likely to follow a similar trajectory, it is possible that the elements visited by the first convected particle will be the same as the ones visited by the following particles inside the same element. The successive particles can first attempt to check these elements as possible candidates of the streamline definition. In Figure 2.21 the candidates for the following particles are shown in grey. This aids in the reduction of the computational cost since the normal search algorithm is an expensive task, even when assisted with bin search structures.

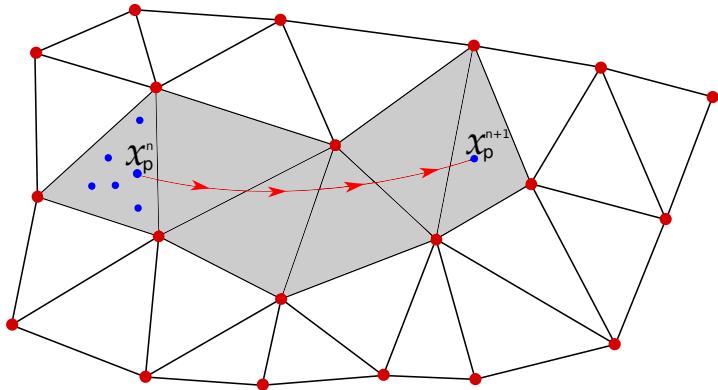


Figure 2.21: Streamline integration. The first element candidates for the successive particles are shown in grey.

2.6 Summary and Conclusion

The standard approach to solve convective equations within an Eulerian framework has been briefly described in this chapter. The implementation of such strategies in finite elements have been explained, together with the stability limits for scalar transport. As observed, stabilization is imperative to ensure stable algorithms, specially for low viscosity/diffusivity materials. This stability problem is even more relevant for fluid or

solid problems, where the convected quantity is the velocity itself, leading to a strongly non linear problem.

The strong non linearity together with the requirement of artificial diffusion to stabilize the convective term represent the main motivations to develop an alternative solving algorithm. The PFEM-2 algorithm seeks to circumvent these limitations by tackling the problem within a Lagrangian framework. By using a set of material points - the particles - together with a fixed mesh, it is possible to create a strategy that retains all conservation of properties without the difficulties of remeshing or the inaccuracies of pure particle methods (meshless methods).

Special emphasis has been placed on the definition of interfaces. Correctly modelling the exact location and the correct equilibrium equations is of vital importance since the behaviour of the complete problems depends on the interaction between the different phases. In this sense, it must be noted that two issues arise from the usage of fixed mesh and particles. On one side, each particle is a discrete material point but yet equations must be solved in a continuum field. On the other hand, standard FE do not represent correctly discontinuities across elements in the material properties.

The problem of the interface detection has been assessed with an interface tracking algorithm. Algorithms to detect both where the particle properties change and the boundaries of the regions occupied by the material domain have been presented. This way it is possible to simulate the interaction of two or more phases that do not fill the entire computational domain.

On the other hand, several enrichment shape functions have been presented to enhance the representation of the interface in the FE field. The boundary terms are carefully analysed to reduce the errors associated to the condensation procedure.

Special attention has been paid on the particle database. The limited bearing of the streamline integration of the multiphase PFEM-2 translate into a larger number of particles compared to the single-phase PFEM-2. This translates into millions of particles, even for relatively small problems. In this sense, the elemental and particles containers have been carefully designed to minimize the particle-associated tasks, taking into consideration that the accessed information has to be used as much as possible before accessing the next data set. The developed database reduces to a minimum the accesses to nodal variable data, thus improving the algorithm speed. It is worth noting that all

the particle tools explained in this chapter do not require any modification in order to be used with any particular problem using the PFEM-2 strategy. Since the elements have direct access to the particles, the update stage is done element-wise and therefore the general strategy remains unmodified. Only the programming of a new element in a Lagrangian frame is required to solve a new problem, making the PFEM-2 a versatile strategy to tackle a broad range of problems.

The tools and strategies discussed in this chapter represent the backbone for the multi-material solvers using the PFEM-2. No customisations or special hypothesis for solids or fluids were performed to obtain the final algorithm, meaning that the strategy is valid for any material that can be modelled as a continuum. In the following chapters, particular implementations for different constitutive equations will be presented.

Chapter 3

Solution Strategies for multi-fluids

3.1 Introduction

The current chapter covers the development of fluid solvers with the PFEM-2 strategy. By exploiting the capacity of particles to convect effortlessly a high number of species, it is possible to track accurately the location of the internal interfaces. An added advantage is that the transport of each particle can be made to depend on the properties of each individual particle in order to improve the first approximation of the convection. This allows the usage of larger time steps in the simulation and therefore the total number of steps is reduced. In this sense, it must be noted that the interface position from one step to the other will differ highly and therefore care must be taken when evaluating information from previous time steps.

This chapter is structured as follows: First, the Navier-Stokes (NS) equations are presented for a general strategy. In the following sections, two different implementations are presented. First, a segregated solver (velocity-pressure) is derived. This strategy is optimal for fluid problems in which the viscosity does not play a dominant role. Secondly, a monolithic implementation is presented. This formulation is suited for both low and high viscosity problems, but more computationally demanding compared to the segregated counterpart.

3.2 Navier Stokes equations

The Navier Stokes (NS) (Donea and Huerta, 2003) equations govern the movement of fluids. To obtain the NS equations for fluids, the first step is writing the momentum equations for a generic continuum, together with a mass conservation equation:

$$\begin{aligned} \rho \left(\frac{\partial(\mathbf{V})}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V} \right) &= \nabla \cdot \boldsymbol{\sigma} + \rho g \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) &= 0 \end{aligned} \quad (3.1)$$

In fluids, the stresses can be easily decomposed into the deviatoric part, produced by viscous effects, and the volumetric component, the pressure. Since the shear stress depends solely on the current deformation rate, it is possible to write the stresses as $\boldsymbol{\sigma} = 2\mu \left(\frac{\nabla \mathbf{V} + (\nabla \mathbf{V})^T}{2} \right) - p \mathbf{I}$. Replacing into (3.1), we obtain:

$$\begin{aligned} \rho \left(\frac{\partial(\mathbf{V})}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V} \right) &= \nabla \cdot \left[2\mu \left(\frac{\nabla \mathbf{V} + (\nabla \mathbf{V})^T}{2} \right) \right] - \nabla p + \rho g \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) &= 0 \end{aligned} \quad (3.2)$$

Equations (3.2) are written in the Eulerian Framework, where the two terms on the left hand side of the first equation describe the movement of the fluid: the first term $\rho \frac{\partial(\mathbf{V})}{\partial t}$ is the acceleration, where ρ is the density and V the velocity, and the second term $\mathbf{V} \cdot \nabla \mathbf{V}$ is the convective term. On the right hand side, the terms define the forces acting on the fluid, with $\nabla \left[2\mu \left(\frac{\nabla \mathbf{V} + (\nabla \mathbf{V})^T}{2} \right) \right]$ the viscous forces depending on the viscosity μ , the gradient of the pressure ∇p and the acceleration of the gravity or other mass forces ρg .

It is interesting to note that all the terms of the first equation are linear except for the convective one (due to the multiplication of the velocity by its gradient). The presence of this non-linear term demands iterative algorithms to converge to the solution. Moreover, the mass conservation equation cannot be simplified into the usual incompressibility condition $\nabla \cdot \mathbf{V}$ used in homogeneous fluids since the term $\frac{\partial \rho}{\partial t}$ changes in time because material points of different densities transit along the fixed points of the domain.

From these observations it is clear that the Eulerian Framework posses several difficulties and would require linearisation techniques to solve the system. On the other hand, the

PFEM-2 strategy allows to use the Navier Stokes equations written in a Lagrangian framework. This leads to a system of equations where the convective term is not present and the material derivatives appear instead of the total derivatives. Then the governing laws in the Lagrangian reference system read:

$$\begin{aligned} \rho \left(\frac{D(\mathbf{V})}{Dt} \right) &= \nabla \cdot \left[2\mu \left(\frac{\nabla \mathbf{V} + (\nabla \mathbf{V})^T}{2} \right) \right] - \nabla p + \rho \mathbf{g} \\ \frac{D\rho}{Dt} + \rho \nabla \cdot (\mathbf{V}) &= 0 \end{aligned} \quad (3.3)$$

Besides the simplification of the first line into a linear equation, one further modification to the system can be introduced to ease the solution strategy. The first term of the mass conservation equation $D\rho/Dt$ can be eliminated since the density of each material point is conserved along the time evolution. Furthermore, the density can be eliminated from the remaining term for the same reason and the incompressibility condition of single-phase fluids is obtained.

$$\begin{aligned} \rho \left(\frac{D(\mathbf{V})}{Dt} \right) &= \nabla \cdot \left[2\mu \left(\frac{\nabla \mathbf{V} + (\nabla \mathbf{V})^T}{2} \right) \right] - \nabla p + \rho \mathbf{g} \\ \nabla \cdot (\mathbf{V}) &= 0 \end{aligned} \quad (3.4)$$

Finally, using a finite difference scheme to advance in time, the system to be solved at each time step becomes (3.5). The parameter θ can be set between 0 (fully explicit viscosity) and 1 (fully implicit viscosity).

$$\begin{aligned} \rho \frac{(\mathbf{V}^{n+1})}{\Delta t} &= \nabla \cdot \left[2\mu \left(\frac{\nabla \mathbf{V}^{n+\theta} + (\nabla \mathbf{V}^{n+\theta})^T}{2} \right) \right] - \nabla p^{n+1} + \rho \mathbf{g} + \rho \frac{\mathbf{V}^n}{\Delta t} \\ \nabla \cdot (\mathbf{V}^{n+1}) &= 0 \end{aligned} \quad (3.5)$$

The developed strategy is capable of solving the NS equations for two or more fluids written in the form (3.5), with completely different properties. The interfaces between the phases are continuously tracked to solve the resulting system accordingly. It must be noted that the missing convective term is still approximated by the convection of the particles, as explained in Chapter 2.

3.3 Fractional Step Method

3.3.1 Motivation

Iterative solvers are the only choice to solve large system of equations. Despite the big advantages in terms of speed over direct solvers, their performance is greatly reduced when the condition number of the system is large, that is, when the system is ill-conditioned.

Solving strategies that compute the pressure and velocity fields at the same time are called monolithic schemes. The advantage of such schemes is that the solution obtained does not make use of any simplification, the linear system is solved exactly as it is. However the big drawback is that since the pressure and velocity have different magnitudes and units, this usually translates into an ill-conditioned system of equations that require several iterations of the solver to converge.

In this section a segregated scheme for multi-fluids is presented. The strategy belongs the *fractional step* family of solvers. This algorithm provides a faster solution compared to monolithic counterparts while maintaining similar accuracy for problems of relatively low viscosity. Still, in this work further improvements over traditional implementations are presented to improve the solution in problems where the viscous forces start playing a more important role.

3.3.2 Weak Form

Using the FE discretization and the test functions \mathbf{w} for the velocity and q for the pressure, the weak form of the system (3.5) becomes (3.6). Note that the viscous term is integrated by parts to avoid second derivatives on the shape functions. The boundary terms that appear when the terms are integrated by parts will be omitted for simplicity.

$$\begin{aligned} \left(\mathbf{w}, \rho \frac{(\mathbf{V}^{n+1})}{\Delta t} \right)_\Omega &= - \left(\nabla w, \left[2\mu \left(\frac{\nabla V^{n+1} + (\nabla V^{n+1})^T}{2} \right) \right] \right)_\Omega \\ &\quad - (\mathbf{w}, \nabla p^{n+1})_\Omega + (\mathbf{w}, \rho \mathbf{g})_\Omega + \left(\mathbf{w}, \rho \frac{\mathbf{V}^n}{\Delta t} \right)_\Omega \quad (3.6) \\ (q, \nabla \cdot (\mathbf{V}^{n+1}))_\Omega &= 0 \end{aligned}$$

Rewriting the system (3.6) in matrix form, the monolithic problem (pressure and velocities together as unknowns) is obtained in (3.7). This is the set of equation that has to be assembled in the fixed FE mesh.

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} & -\mathbf{G} \\ \mathbf{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}(\mathbf{V}^n, \mathbf{g}) \\ 0 \end{bmatrix} \quad (3.7)$$

Where the matrices components are:

$$\begin{aligned} \mathbf{M}(\rho)_{ij} &= \int_{\Omega} \mathbf{N}_i \rho \mathbf{N}_j \\ \mathbf{G}_{ij} &= \int_{\Omega} \nabla \mathbf{N}_i \mathbf{N}_j \\ \mathbf{D}_{ij} &= \int_{\Omega} \mathbf{N}_i \nabla \mathbf{N}_j \\ \mathbf{K}(\mu)_{ij} &= \int_{\Omega} \nabla \mathbf{N}_i \mu \nabla \mathbf{N}_j \\ \mathbf{F}(\mathbf{V}^n, \mathbf{g}) &= \mathbf{M}(\rho)(\mathbf{g} + \frac{1}{\Delta t} \mathbf{V}^n) \end{aligned}$$

The previous system must be solved at each time step to advance in time. As explained above, despite it is possible to solve it directly for both the pressure and velocity simultaneously, in this section the segregated scheme is presented with the aim to develop a faster solver. Besides, the monolithic problem requires stabilization to solve the system due to the equal interpolation of velocity and pressure variables (Codina and Badia, 2006).

The developed strategy is a first order Fractional Step (FS1) solver. Besides providing an easy implementation of the equations, the FS1 remains stable even without stabilization, unlike the monolithic scheme or higher order FS algorithms. It also provides advantages in problems in which the pressure field differs too much from one step to the next.

Rewriting the first equation of (3.6), the velocity reads:

$$\left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right] \mathbf{V}^{n+1} + \mathbf{G} p^{n+1} = \mathbf{F} \rightarrow \mathbf{V}^{n+1} = \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} [\mathbf{F} - \mathbf{G} p^{n+1}] \quad (3.8)$$

Now replacing into the second equation (incompressibility).

$$\begin{aligned} \mathbf{D}\mathbf{V}^{n+1} = 0 \rightarrow \mathbf{D} \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} [\mathbf{F} - \mathbf{G}p^{n+1}] &= 0 \\ \left[\mathbf{D} \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} \mathbf{G} \right] p^{n+1} &= \mathbf{D} \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} \mathbf{F} \end{aligned} \quad (3.9)$$

Up to this point, no assumptions or simplifications were applied to the equations to ease the solving of the system. To begin the solution strategy, the velocity is split in two: A first, compressible, approximation known as Fractional Velocity $\hat{\mathbf{V}}$ (Zienkiewicz and Taylor, 2000a) that neglects the pressure gradient and a correction to ensure that the velocity is incompressible at the end of step. Taking equation 3.9:

$$\begin{aligned} \hat{\mathbf{V}}^{n+1} &= \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} [\mathbf{F} - \cancel{\mathbf{G}p^{n+1}}] \\ \mathbf{V}^{n+1} &= \hat{\mathbf{V}}^{n+1} - \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} [\mathbf{G}p^{n+1}] \end{aligned} \quad (3.10)$$

NOTE More accurate results (second order FS) can be obtained if the pressure from the previous time step is to be used as a first approximation in (3.10), instead of zero. However it would cause severe problems in the case of multi-fluids. In this type of problems, the previous pressure does not provide a good approximation on the current pressure, as will explained the following sections.

On the other hand, the pressure equation reads:

$$\left[\mathbf{D} \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} \mathbf{G} \right] p^{n+1} = \mathbf{D}\hat{\mathbf{V}}^{n+1} \quad (3.11)$$

Finally, reordering the steps we get the three steps of the segregated scheme:

$$\begin{aligned} 1) \quad \hat{\mathbf{V}}^{n+1} &= \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} \mathbf{F} \\ 2) \quad \left[\mathbf{D} \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} \mathbf{G} \right] p^{n+1} &= \mathbf{D}\hat{\mathbf{V}}^{n+1} \\ 3) \quad \mathbf{V}^{n+1} &= \hat{\mathbf{V}}^{n+1} - \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} [\mathbf{G}p^{n+1}] \end{aligned} \quad (3.12)$$

It is interesting to note that the solution 3.12, despite expanded into three steps, remains exact since no simplifications were made and could actually be solved as written. However, the second and third equations demand a high computational effort to find the inverse of the submatrices inside the system. The usual strategy of the Fractional Step is to create an approximation, supposing the viscosity term is negligible against the inertial(mass) term (Donea and Huerta, 2003). While this hypothesis leads to acceptable results in many high Reynolds numbers applications (such as airfoil analysis), it is not valid for the case of debris flows, where viscosity plays a dominant role in the flow behaviour. For this reason a new strategy was developed to simplify the system with a better approximation.

The first equation of (3.12) represents the effect of gravitational and viscous forces and it is solved as it is using an iterative linear solver. On the other hand, in the second equation, it is desirable to find a simpler approximation to the matrix on the left hand side. In general, a matrix of the shape **DIG** can be approximated by the Laplacian matrix **L** (Codina and Badia, 2006), being **I** the identity matrix:

$$[\mathbf{D}][\mathbf{I}][\mathbf{G}] \approx [\mathbf{L}] \quad (3.13)$$

Therefore the problem lies in finding a diagonal approximation for $\left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1}$. The mass matrix $\mathbf{M}(\rho)$ is normally diagonalized in a process called "lumping" (Felippa, 2004), which is a common practice in FE strategies because results are acceptable despite the simplification. On the other hand, to diagonalize the viscosity matrix $\mathbf{K}(\mu)$, a term emulating the diffusion coefficient in the diagonal has to be used, which depends on the diffusion coefficient and the second derivatives along the coordinates. The most straightforward approach to find a correct value is reducing the case to one dimensional problems. Recalling finite differences schemes for any variable T :

$$\mu \frac{\partial^2(T)}{\partial x^2} = \mu \frac{T^{i-1} - 2T^i + T^{i+1}}{h^2} \quad (3.14)$$

As it can be seen, in 1D diffusion problems the input term depends on the square of the element length h . Extending this approximation to the general case and integrating in the volume of the element vol_e , the second equation of the fractional step scheme

becomes:

$$\left[\mathbf{D} \left[vol_e \left(\frac{\mu}{h^2} + \frac{\rho}{\Delta t} \right) \mathbf{I} \right]^{-1} \mathbf{G} \right] p^{n+1} = \mathbf{D} \hat{\mathbf{V}}^{n+1}$$

The matrix $vol_e \left(\frac{\mu}{h^2} + \frac{\rho}{\Delta t} \right) I$ can be seen as a modified lumped mass matrix:

$$\left[\mathbf{D} \left[\mathbf{M} \left(\frac{\mu}{h^2} + \frac{\rho}{\Delta t} \right) \right]^{-1} \mathbf{G} \right] p^{n+1} = \mathbf{D} \hat{\mathbf{V}}^{n+1} \quad (3.15)$$

So finally the simplified system becomes:

$$\mathbf{L} \left(\frac{1}{\frac{\mu}{h^2} + \frac{\rho}{\Delta t}} \right) p^{n+1} = \mathbf{D} \hat{\mathbf{V}}^{n+1} \quad (3.16)$$

Now using the same modified mass matrix of (3.15) into the third equation of (3.12) , the new simplified system ready to be solved is:

$$\begin{aligned} 1) \quad & \hat{\mathbf{V}}^{n+1} = \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} \mathbf{F} \\ 2) \quad & \mathbf{L} \left(\frac{1}{\frac{\mu}{h^2} + \frac{\rho}{\Delta t}} \right) p^{n+1} = \mathbf{D} \hat{\mathbf{V}}^{n+1} \\ 3) \quad & \mathbf{V}^{n+1} = \hat{\mathbf{V}}^{n+1} - \left[\mathbf{M} \left(\frac{\mu}{h^2} + \frac{\rho}{\Delta t} \right) \right]^{-1} [\mathbf{G} p^{n+1}] \end{aligned} \quad (3.17)$$

NOTE: The modification on the mass and pressure matrices, despite providing a better approximation, does not guarantee an exact solution. Due to the diagonalization of the terms, the diffusion effect is lost and therefore the solution is only a (better) approximation. For this reason, when the viscosity forces play a dominant role the solution will not be accurate with this strategy.

3.3.3 Improvements for the first order Fractional Step scheme

The decision to make a first order FS is based on the fact that due to the target large time steps, the pressure field from one step to the next is completely different, specially when portions of the denser fluid hit the walls or two drops collide. Figure 3.1 shows the pressure distribution at two consecutive time steps of a droplet hitting a wall. From

in this image it is clear that estimating the updated field from the previous step would not be a good approach.

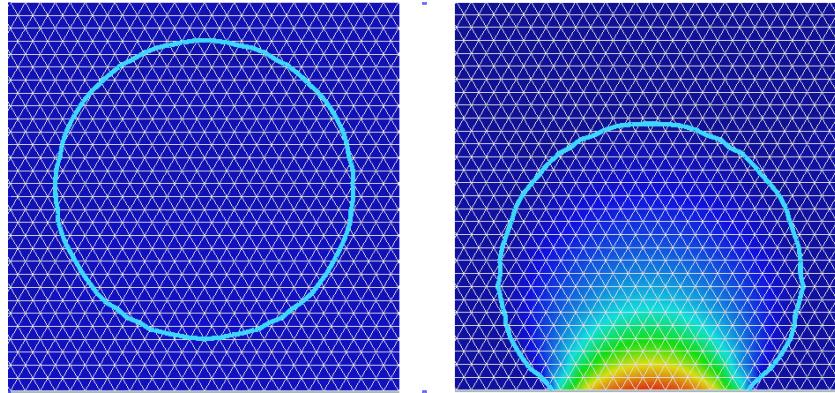


Figure 3.1: Pressure field of a water droplet before and after hitting a wall

The first order FS scheme is only first order accurate (Codina and Badia, 2006), compared to the second order FS step scheme that uses the pressure from the previous time step. While in single phase fluid this may not be an important problem, in multi-fluids the results are unacceptable since severe mass losses appear and therefore the physics of the problem change completely. To overcome this issue, it is possible to change the strategy in order to improve the accuracy towards a second order scheme. The idea consists on, instead of finishing the solution in the third step of (3.17) , treat this velocity as the first step of an iterative process involving steps 2 and 3, a predictor-corrector scheme (Codina and Badia, 2006). This means taking at the end of each iteration $\hat{\mathbf{V}}^{n+i+1} = \mathbf{V}^{n+i}$ and repeat again steps 2 and 3 until convergence using $\delta p^{n+i+1} = p^{n+i+1} - p^{n+i}$. Starting with $p^n = 0$, the full algorithm becomes:

$$\begin{aligned}
 1) \quad & \hat{\mathbf{V}}^{n+1} = \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} \mathbf{F} \\
 2) \quad & \mathbf{L} \left(\frac{1}{\frac{\mu}{h^2} + \frac{\rho}{\Delta t}} \right) p^{n+i+1} = \mathbf{D}\hat{\mathbf{V}}^{n+1} + \mathbf{L} \left(\frac{1}{\frac{\mu}{h^2} + \frac{\rho}{\Delta t}} \right) p^{n+i} \\
 3) \quad & \mathbf{V}^{n+1+i} = \hat{\mathbf{V}}^{n+1+i} - \left[\mathbf{M} \left(\frac{\mu}{h^2} + \frac{\rho}{\Delta t} \right) \right]^{-1} [\mathbf{G} \delta p^{n+i+1}] \\
 & \text{set } \hat{\mathbf{V}}^{n+1} = \mathbf{V}^{n+1+i} \\
 & \text{go to step 2)
 \end{aligned} \tag{3.18}$$

From the second step it is clear that when the divergence of the velocity is zero, the pressure in the new iteration will be equal to that of the previous one (LHS=RHS) . In this case, convergence is achieved and the mass loss in the FE computations will be zero.

It must be noted that the intrinsic stability of the FS1 is optimal for Courant numbers (Courant et al., 1967) $Co \approx 1$ and does not hold for zero velocities. For this reason if extremely small time steps were to be used, spurious pressure variations would appear. Despite this limitation and the fact that the stabilization terms were actually implemented and tested, the current version of the formulation does not include them. The reason for this is that the aim of this work is the development of fast algorithms, meaning that Courant numbers will always be higher than one, in some cases being > 10 . In these conditions, the algorithm will always be stable and therefore there is no need to add stabilization, mainly because it would increase numerical dissipation, reducing the accuracy.

3.3.4 Boundary conditions

Integrating by parts some of the terms of the weak form of the problem leads to the creation of new boundary terms that are not present in the original strong formulation. In particular, the pressure equation (second equation in (3.17)) demands special attention. When the pressure term is weakened and integrated by parts, a term appears in the boundaries Γ that is usually neglected, which leads to algorithms that need several initial cycles before converging to a physically correct solution. Writing the Right Hand Side (RHS) in the full form instead of matrix notation, for a domain Ω ,

$$\mathbf{L} \left(\frac{1}{\frac{\mu}{h^2} + \frac{\rho}{\Delta t}} \right) \delta p^{n+1} = \int_{\Omega} q \nabla \cdot \hat{\mathbf{V}}^{n+1} d\Omega - \int_{\Gamma} q \nabla \delta p^{n+1} \cdot \mathbf{n} d\Gamma \quad (3.19)$$

The term $q \nabla \delta p^{n+1} \cdot \mathbf{n}$ is usually neglected in most second order strategies, which means that the gradient variation between steps is assumed to be small or negligible in the boundaries of the domain. The resulting algorithm usually shows incorrect results on the first steps of a simulation but eventually converges to the solution. Since the strategy described in this work starts with $p^n = 0$ in the first iteration, the previous hypothesis is not suitable. This is specially critical for gravity-driven flows, in which the gradient

of the pressure will be of at least the gravity acceleration multiplied by the density of the fluid ρg . While using several iterations would eventually lead to the correct result (as in standard implementations of the FS), it is not desirable due to the computer resources it would take. To overcome this, the complete equation is written without omitting the boundary terms. Now integrating by parts the divergence of the velocity $\int_{\Omega} u \nabla \cdot \hat{\mathbf{V}}^{n+1}$.

$$\mathbf{L} \left(\frac{1}{\frac{\mu}{h^2} + \frac{\rho}{\Delta t}} \right) p^{n+1} = - \int_{\Omega} \nabla q \hat{\mathbf{V}}^{n+1} d\Omega + \int_{\Gamma} q \hat{\mathbf{V}}^{n+1} \cdot \mathbf{n} d\Gamma - \int_{\Gamma} q \nabla p^{n+1} \cdot \mathbf{n} d\Gamma \quad (3.20)$$

It is interesting to note that the last two terms, being the fractional velocity minus the gradient of the pressure, become the incompressible velocity. That is, the velocity at the end of the time-step $\hat{\mathbf{V}}^{n+1}$. For all internal boundaries the normal of the faces \mathbf{n} sum zero and the terms disappear (Figure 3.2). And for the boundary faces, they will become a new source term that is zero when the imposed velocity is zero. On the other hand, the divergence of the velocity integrated by parts is the transpose of the pressure gradient: $\int_{\Omega} \nabla q \hat{\mathbf{V}}^{n+1} d\Omega = \mathbf{G}^T$.

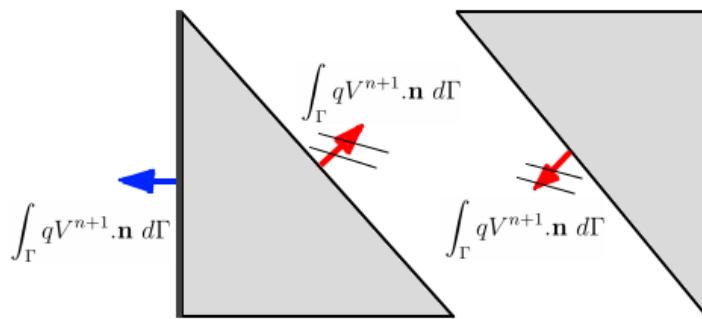


Figure 3.2: Velocity Boundary terms in the pressure equation

One of the advantages of this formulation is that writing no boundary terms leads to naturally impermeable walls. But the main advantage is that the gradient of the pressure boundary term is not taken as zero. As stated before, neglecting this term would lead to important errors in problems dominated by the gravitational force and mass losses, specially at the bottom walls of the domain.

Writing the pressure equation with the divergence of the velocity integrated by parts, the second step becomes (3.21). Note that only the RHS is modified, so the system

remains symmetric.

$$\mathbf{L} \left(\frac{1}{\frac{\mu}{h^2} + \frac{\rho}{\Delta t}} \right) p^{n+1} = -\mathbf{G} \hat{\mathbf{V}}^{n+1} + \int_{\Gamma} q \mathbf{V}^{n+1} \cdot \mathbf{n} \, d\Gamma \quad (3.21)$$

3.3.5 Special considerations on the interface

Despite the advantages that the FE discretization provides, it bears a severe limitation when the variables require abrupt changes that the chosen FE space is not able to reproduce. One of such examples is the pressure along an interface between two different materials. When there is a sharp change of the density, the hydrostatic condition under the gravitational force leads to two different values for the gradient of the pressure. This is specially critical when the fluids have very different densities (i.e. air-water). In these cases, the term $\nabla p \approx \rho g$ changes abruptly on the interface.

Figure 3.3 shows the exact pressure distribution for the hydrostatic case of water and air and the one obtained by three linear elements. It is clear that the solution is very poor and in practice it would lead to an incorrect behaviour of the interface and mass losses (Coppola, 2009).

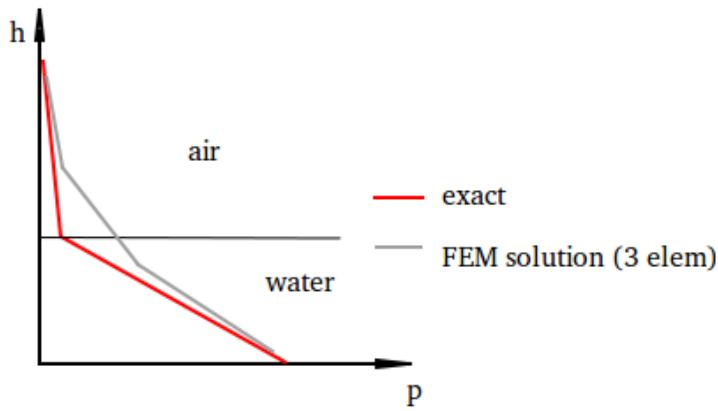


Figure 3.3: Pressure distribution for the hydrostatic case

To account for the gradient discontinuity, the shape function proposed by Coppola (2009) is used. No pressure discontinuities are allowed in the fractional step scheme since there can be no jumps in the pressure field for low viscosities.

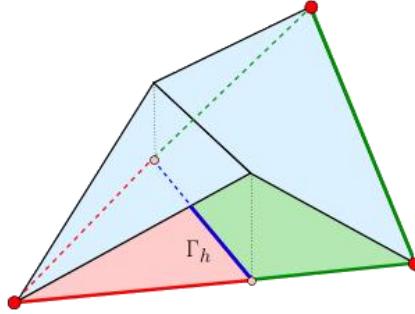


Figure 3.4: Enrichment for the pressure field

Treatment of boundary terms - Divergence terms integration

As it was explained in Section 2.4.4, condensing the degrees of freedom implies that boundary terms might not be correctly assessed in the enrichment DoFs. Following the same procedure of standard shape functions, the divergence of the velocity evaluated with gradient discontinuity can be either fully integrated or integrated by parts. The errors of both methodologies have to be analysed in order to choose the one leading to the solution with the lowest errors.

Standard integration To begin with, the errors of full integration are analysed. Recalling the boundary terms of Equation (3.19) with standard integration and writing for the enrichment shape function:

$$-\int_{\Gamma} q \nabla \delta p_*^{n+1} \cdot \mathbf{n} d\Gamma \quad (3.22)$$

The first order scheme that is used implies that there are no previous estimations for the pressure. This implies that, since this degree of freedom will be condensed, it will not be possible to create a priori estimation for the flux ∇p_* . For this reason it will be considered to be zero. This hypothesis is equivalent to the example analysed in Chapter 2, where poor convergence was observed.

$$0 = \int_{\Gamma} q \nabla \delta p_*^{n+1} \cdot \mathbf{n} d\Gamma \quad (3.23)$$

To check convergence, the same geometry used to assess the error in Chapter 2 is taken, as seen in Figure 3.5. A horizontal acceleration of $-10/4.5 \text{ m/s}$ is set and the initial condition is zero velocity in the whole domain. The exact solution for the pressure equation with these values must be exactly the same as the temperature problem of Chapter 2, as shown in Figure 2.18: The pressure must be exactly 10 Pa at the interface and 110 Pa at the right side.

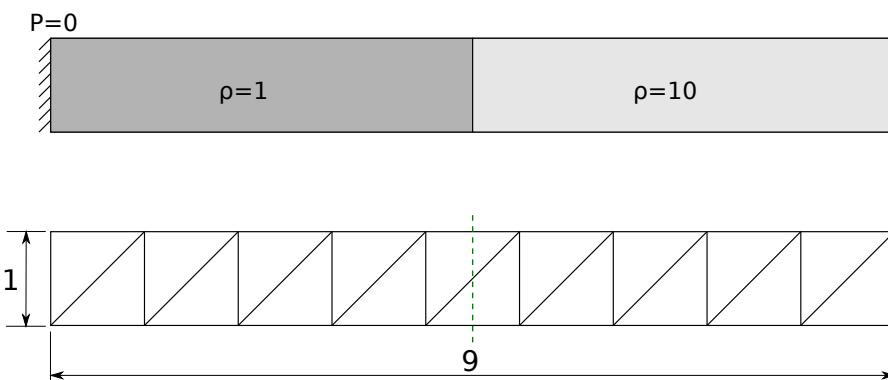


Figure 3.5: Hydrostatic test example. Up: real domain, bottom: discretized problem. The enrichment location is marked with the green dotted line

In Figure 3.6 the mesh convergence of the problem can be observed. As expected from the analysis of the thermal problem, the error is improved considerably compared to the non-enriched problem but the convergence remains linear. Despite being a good improvement, it does not take advantage of the full potential of the extended pressure field; for this case the *exact* solution should be possible for all mesh sizes.

Integration by parts The divergence of the velocity was integrated by parts when evaluated by the standard shape functions. This allowed to obtain an accurate pressure field in a single fractional step iteration. The same strategy can be used for the new pressure shape function N_* . Following Equation (3.21), the new boundary term becomes:

$$\int_{\Gamma} q_* \mathbf{V}^{n+1} \cdot \mathbf{n} \, d\Gamma = 0 \quad (3.24)$$

As it can be seen, there are no terms involving the pressure in this expression. Therefore omitting it from the formulation should not affect the precision for the pressure solution.

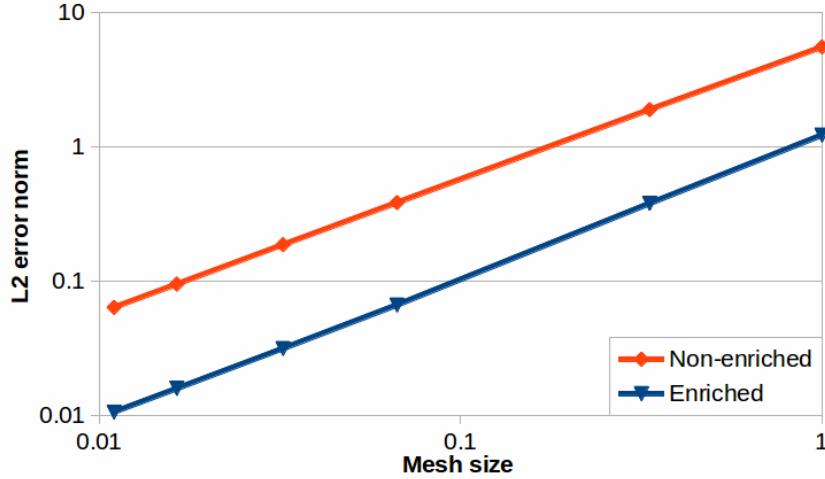


Figure 3.6: Hydrostatic problem mesh convergence with full integration for the divergence evaluated with N^*

Effectively, if the divergence is integrated by parts for the enrichment function, the benchmark for the hydrostatic case is passed with the *exact solution* for all mesh sizes: $L_2 \text{ error} = 0$. For this reason, the convergence plot is not shown.

However, recalling Figure 3.2, the omission of the boundary term (3.24) in the standard shape functions lead to weakly impermeable walls. In the same way, setting it to zero for the enrichment test function q_* will translate into partially impermeable walls and the solution will be degraded, with the inconvenience that measuring the error is not straightforward. To avoid this, an assumption has to be made on the volume flux across the elemental boundaries Γ . Using the fractional velocity \hat{V} , it is possible to create a first estimation on the final velocity. This term has to be added to the Right Hand Side of the pressure system in the enrichment pressure equation:

$$RHS_{p*}+ = \int_{\Gamma} q_* \hat{V}^{n+1} \cdot \mathbf{n} \, d\Gamma \quad (3.25)$$

Since several iterations will be used to reduce mass losses until convergence is achieved, the velocity used for the boundary terms can be updated at each iteration with the previous iteration velocity. This will allow to obtain a solution without incorrect assumptions, both in the pressure and velocity fields.

Full assembled system

The new full enriched system of equations for the pressure degrees of freedom of each of the interface elements can be written in the following way before condensing the extra DoFs (3.26). The source terms, the divergence of the velocity and the previous pressure, are expressed as F for more clarity. To distinguish the different shape functions, the sub-index N is used for the standard FE space and $*$ for the enriched DoFs. To avoid unnecessary indexes, for the moment the ones corresponding to the iteration number are not written.

$$\begin{bmatrix} \mathbf{L}_{NN} & \mathbf{L}_{N*} \\ \mathbf{L}_{*N} & \mathbf{L}_{**} \end{bmatrix} \begin{bmatrix} p_N \\ p_* \end{bmatrix} = \begin{bmatrix} \mathbf{F}_N \\ \mathbf{F}_* \end{bmatrix} \quad (3.26)$$

Expanding the system and rearranging the equations, the enriched DoFs can be obtained. Note that the submatrix L_{**} must be invertible, otherwise it is not possible to condense. Now, for a given iteration $i + 1$, the enriched pressure is (3.27):

$$p_*^{i+1} = \mathbf{L}_{**}^{-1}(\mathbf{F}_* - \mathbf{L}_{*N}p_N^{i+1}) \quad (3.27)$$

Finally replacing (3.27) into (3.26) the new unknowns are condensed and the system recovers the same number of unknowns it originally had. It must be noted that after the pressure has been calculated, the p_* must be recovered using (3.27) in order to calculate the gradient of the pressure correctly in the last step of the segregated strategy.

$$\tilde{\mathbf{L}}p^{i+1} = \tilde{\mathbf{F}} \quad (3.28)$$

where

$$\begin{aligned} \tilde{\mathbf{L}} &= \mathbf{L}_{NN} - \mathbf{L}_{N*}\mathbf{L}_{**}^{-1}\mathbf{L}_{*N} \\ \tilde{\mathbf{F}} &= \mathbf{F}_N - \mathbf{L}_{N*}\mathbf{L}_{NN}^{-1}\mathbf{F}_* \end{aligned}$$

Writing all the terms that account for F^* , the updated enriched pressure becomes (3.29). Notice that the formula includes the enriched pressure from the previous iteration, so having the standard pressures alone is not enough to calculate it, reason why it is

important to keep track of it from one iteration to another.

$$p_*^{i+1} = \mathbf{L}_{**}^{-1} \left[\left(\mathbf{L}_{**} p_n^* + \mathbf{G}_* \hat{\mathbf{V}}^{i+1} \right) - \mathbf{L}_{N,*} (\delta p^{i+1}) \right] \quad (3.29)$$

Finally, having calculated the updated standard pressures with (3.28) and the updated enriched pressure with (3.29), the correction stage (step 3) of the interface element becomes (3.30)

$$\mathbf{M} \mathbf{V}^{i+1} = \mathbf{M} \hat{\mathbf{V}}^{i+1} - \Delta t (\mathbf{G}_N \delta p_N^{i+1} + \mathbf{G}_* \delta p_*^{i+1}) \quad (3.30)$$

3.3.6 Complete algorithm

The algorithm developed in this section provides the Finite Element basis for the Navier Stokes segregated solver in a Lagrangian framework. As it was stated at the beginning, by combining this strategy with the computation of the convective term using the particles, the full Navier Stokes solver is obtained.

A key aspect that must be taken into account is that, since the divergence terms have been integrated by parts, all the domain must be calculated at all times. Even if only one phase is of interest, the other phase must be still calculated. Failing to do so will generate impermeable walls between active and inactive elements, so this solver cannot be used in the same way as a single fluid, level set solver. Below the steps to be followed at each time steps are presented.

Step 1) Convect the particles using the streamlines

Step 2) Project information into the mesh,

with q fluid particles

$$\hat{\mathbf{V}}_j^{n+1} = \frac{\sum_i^{r+q} \mathbf{V}_i N_i^j}{\sum_i^{r+q} N_i^j}$$

$$\varphi_j^{n+1} = \frac{\sum_i^n \text{sign}_i N_i^j}{\sum_i^n N_i^j}$$

Step 3) Detect interface elements

Step 4) Assemble and solve the system of equations with enrichments

$$\begin{aligned}
 3.1) \quad & \hat{\mathbf{V}}^{n+1} = \left[\mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} \right]^{-1} \mathbf{F} \\
 3.2) \quad & \tilde{\mathbf{L}} \left(\frac{1}{\frac{\mu}{h^2} + \frac{\rho}{\Delta t}} \right) p^{n+i+1} = \mathbf{G} \hat{\mathbf{V}}^{n+1} + \tilde{\mathbf{L}} \left(\frac{1}{\frac{\mu}{h^2} + \frac{\rho}{\Delta t}} \right) p^{n+i} \\
 3.3) \quad & p_*^{i+1} = \mathbf{L}_{**}^{-1} \left[\left(\mathbf{L}_{**} p_n^* + \mathbf{G}_* \hat{\mathbf{V}}^{i+1} \right) - \mathbf{L}_{N,*} (\delta p^{i+1}) \right] \\
 3.3) \quad & \mathbf{V}^{n+1+i} = \hat{\mathbf{V}}^{n+1+i} - \left[\mathbf{M} \left(\frac{\mu}{h^2} + \frac{\rho}{\Delta t} \right) \right]^{-1} [\mathbf{G} \delta p^{n+i+1} + \mathbf{G}_* \delta p_*^{i+1}]
 \end{aligned} \tag{3.31}$$

set $\hat{\mathbf{V}}^{n+1} = \mathbf{V}^{n+1+i}$

go to step 3.2) until convergence

Step 5) Recover condensed DoFs

Step 6) Update particles' velocity

3.4 Monolithic strategy

3.4.1 Motivation

When the viscosity of the fluids play a dominant role in the simulation, the fractional step scheme is no longer a good alternative. The hypothesis used to develop the strategy are not valid and therefore the results are not correct using a single fractional velocity iteration. Despite it would be possible to achieve convergence by iterating all the steps of the strategy, it would no longer provide a fast algorithm, the original reason why the fractional step was developed for.

In this high viscosity cases, the monolithic scheme provides a better alternative. The method consists on solving the pressure and velocity equations together, meaning that no simplifications are made to solve the system.

3.4.2 Spatial Discretization

As the first step, the Lagrangian equations of the fluid motion are recalled for convenience (3.32):

$$\begin{aligned} \rho \frac{(\mathbf{V}^{n+1})}{\Delta t} &= \nabla \left[2\mu \left(\frac{\nabla \mathbf{V}^{n+\theta} + (\nabla \mathbf{V}^{n+\theta})^T}{2} \right) \right] - \nabla p^{n+1} + \rho g + \rho \frac{\mathbf{V}^n}{\Delta t} \\ &\quad \nabla \cdot (\mathbf{V}^{n+1}) = 0 \end{aligned} \quad (3.32)$$

A Finite Element discretization directly implemented in the form (3.33) using the same shape functions for the velocity and the pressure would be unstable in the pressure since it does not stand the inf-sup condition (Codina and Badia, 2006). Moreover, even if different functions were to be used to avoid this restriction, the system would not be suitable for several linear solvers since it contains zeros in the diagonal terms of the pressure equation as seen in the matrix form (3.34).

$$\begin{aligned} \left(\mathbf{w}, \rho \frac{(\mathbf{V}^{n+1})}{\Delta t} \right)_\Omega &= - \left(\nabla w, \left[2\mu \left(\frac{\nabla V^{n+1} + (V^{n+1})^T}{2} \right) \right] \right)_\Omega \\ &\quad - (\mathbf{w}, \nabla p^{n+1})_\Omega + (\mathbf{w}, \rho g)_\Omega + \left(\mathbf{w}, \rho \frac{\mathbf{V}^n}{\Delta t} \right)_\Omega \\ &\quad (q, \nabla \cdot (\mathbf{V}^{n+1}))_\Omega = 0 \end{aligned} \quad (3.33)$$

Integrating by parts the gradient of the pressure, the matrix system becomes:

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} & \mathbf{D}^T \\ \mathbf{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}(\mathbf{V}^n, g) \\ 0 \end{bmatrix} \quad (3.34)$$

3.4.3 Stabilized problem

Pressure-Stabilizing/Petrov-Galerkin

In order to raise the limitation of the original problem, stabilization terms can be added to the set of equations (3.33). As stated before, it is only necessary to stabilize the pressure equation. The simplest way to achieve this while modifying as little as possible the solution is the Pressure-Stabilizing/Petrov-Galerkin (PSPG) method (Tezduyar, 1992). It consists on adding a new perturbation term that is $\tau \nabla q$ multiplied by the residual of the momentum equation (first equation in (3.32)), which becomes (3.35):

$$0 = \tau \left(\nabla q, \rho \frac{(\mathbf{V}^{n+1} - \mathbf{V}^n)}{\Delta t} - \nabla [2\mu (\nabla^s V^{n+1})] + \nabla p^{n+1} - \rho g \right)_{\Omega} \quad (3.35)$$

Since the elements used are linear and it is not possible to integrate by parts because the gradient of the test function is being used, the viscosity term vanishes due to the second derivatives. Now adding the perturbation to the unstabilized original weak form and dividing by the density to obtain consistent units, the problem to be solved is obtained (3.36):

$$\begin{aligned} \left(\mathbf{w}, \rho \frac{(\mathbf{V}^{n+1})}{\Delta t} \right)_{\Omega} &= - \left(\nabla w, \left[2\mu \left(\frac{\nabla V^{n+1} + (\nabla V^{n+1})^T}{2} \right) \right] \right)_{\Omega} \\ &\quad - (\mathbf{w}, \nabla p^{n+1})_{\Omega} + (\mathbf{w}, \rho g)_{\Omega} + \left(\mathbf{w}, \rho \frac{\mathbf{V}^n}{\Delta t} \right)_{\Omega} \\ &\quad + \tau \left(\nabla q, \frac{(\mathbf{V}^{n+1} - \mathbf{V}^n)}{\Delta t} + \frac{1}{\rho} \nabla p^{n+1} - g \right)_{\Omega} \\ &\quad (q, \nabla \cdot (\mathbf{V}^{n+1}))_{\Omega} = 0 \end{aligned} \quad (3.36)$$

And in matrix form (3.37)

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{1}{\Delta t} \mathbf{M}(\rho) & \mathbf{D}^T \\ \mathbf{D} + \frac{\tau}{\Delta t} \mathbf{D}^T & \tau L(\frac{1}{\rho}) \end{bmatrix} \begin{bmatrix} \mathbf{V}^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M} \mathbf{V}^n + \mathbf{M} \mathbf{g} \\ \frac{\tau}{\Delta t} \mathbf{D}^T \mathbf{V}^n + \tau \mathbf{D}^T \mathbf{g} \end{bmatrix} \quad (3.37)$$

The system 3.37 can be solved as it is by choosing an appropriate stabilization parameter τ . In this work it is set to $\tau = (1/\Delta t + 4\mu/h^2 + 2V^n/h)^{-1}$ following the work by Codina (2002). The dependency on Δt was added to avoid $\tau \rightarrow \infty$ when the velocity and the viscosity are zero, following the work of Bochev et al. (2007).

Despite it is now stable, the new set of equation lost the symmetry of the system. This reduces the envelope of solvers that can be used, requiring more expensive algorithms. However it is worth noting that the asymmetry originates solely from the stabilization term. Now, considering the acceleration in the perturbation(stabilization) terms to be negligible respect to the acceleration of the gravity:

$$\frac{|V^{n+1} - V^n|}{\Delta t} \ll g$$

the acceleration terms disappear and the new system becomes symmetric (3.38). It must be noted that the previous approximation does not affect directly the momentum equation nor the pressure system but only the stabilization terms, leading to a non-consistent stabilization. Since in general these terms are small compared to the standard terms, the errors originated from this approximation will be negligible. Numerical tests were performed with and without the acceleration in the stabilization and in fact there was no noticeable difference.

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{1}{\Delta t} \mathbf{M}(\rho) & \mathbf{D}^T \\ \mathbf{D} & \tau L\left(\frac{1}{\rho}\right) \end{bmatrix} \begin{bmatrix} \mathbf{V}^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M}\mathbf{V}^n + \mathbf{M}\mathbf{g} \\ \tau \mathbf{D}^T \mathbf{g} \end{bmatrix} \quad (3.38)$$

Orthogonal Sub-grid Scale

The PSPG method provides a good balance between accuracy and speed, without requiring an estimation on the current pressure. However there are other alternatives that are worth being mentioned. The Orthogonal Sub-grid Scale (OSS) stabilization consists on using the projection on the pressure rather than the momentum equation. For the pressure stabilization and using again ∇q as a weighting function, the OSS can be simply written as:

$$0 = \tau (\nabla q, \pi(\nabla p^{n+1}) - \nabla p^{n+1})_{\Omega} \quad (3.39)$$

Where π is the L^2 orthogonal projector. This method is superior to the PSPG in terms of accuracy, however it has the inconvenient that the projector has a stencil wider than the Laplacian of the pressure (Badia, 2012). Figure 3.7 shows the stencil required to calculate the pressure projection for the nodes in the K element. As it can be seen, information from second neighbour elements are needed. Computing the projection implicitly as (3.39) would increase dramatically the computational cost.

Clearly an implicit implementation of this stabilization method is not computationally efficient. Therefore an explicit estimation must be created to implement a fast algorithm. As stated in the beginning of the chapter, the previous step pressure does not provide a

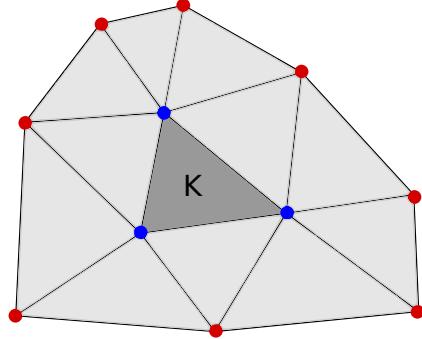


Figure 3.7: Stencil required for the L2 projection in the nodes of element K

good approximation in problems with large time steps and a high density jump. However in certain cases when the density difference between the two phases is not excessively high, it is possible to gather information from the previous time step. This way the projection becomes explicit and therefore the stencil is not extended.

$$0 = \tau (\nabla q, \pi(\nabla p^{n+1}) - \nabla p^n)_\Omega \quad (3.40)$$

Inserting the stabilization terms into the discretized system of equations, the matrix form can be expressed as:

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{1}{\Delta t} \mathbf{M}(\rho) & \mathbf{D}^T \\ \mathbf{D} & \tau L(\frac{1}{\rho}) \end{bmatrix} \begin{bmatrix} \mathbf{V}^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M}\mathbf{V}^n + \mathbf{M}\mathbf{g} \\ \tau \mathbf{D}^T \boldsymbol{\Pi}(p^n) \end{bmatrix} \quad (3.41)$$

It must be noted that the structure of the problem is exactly the same as the one obtained for the PSPG problem, so the complexity is not altered. The only added task is the computation of the explicit projection, whose associated cost is negligible compared to the solution of the system.

Since both explored stabilizations demand a similar computational cost, using OSS or PSPG will depend on the physics of the problem. When the conditions allow for it, that is, the previous pressure is a good approximation of the current pressure, the OSS method will lead to more accurate solutions since no terms are neglected in the development of the formulation.

3.4.4 Pressure enrichments on the interface

Since there is no limitation in the viscosity of the simulated fluids, discontinuities in the pressure can appear when there is a jump in the viscosity. For this reason a full set of discontinuous shape functions will be used for the pressure field, as shown in Figure 3.8 . Recalling the notation N for the standard shape functions and $*$ for the enrichments, the extended system becomes (3.42). The equations are written with the PSPG stabilization, but the procedure is exactly the same for the OSS method.

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{1}{\Delta t} \mathbf{M}(\rho) & \mathbf{D}^T & \mathbf{D}_*^T \\ \mathbf{D}_N & \tau \mathbf{L}(\frac{1}{\rho})_{NN} & \tau \mathbf{L}(\frac{1}{\rho})_{N*} \\ \mathbf{D}_* & \tau \mathbf{L}(\frac{1}{\rho})_{*N} & \tau \mathbf{L}(\frac{1}{\rho})_{**} \end{bmatrix} \begin{bmatrix} \mathbf{V}^{n+1} \\ p_N^{n+1} \\ p_*^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M} \mathbf{V}^n + \mathbf{M} \mathbf{g} \\ \tau \mathbf{D}_N^T \mathbf{g} \\ \tau \mathbf{D}_*^T \mathbf{g} \end{bmatrix} \quad (3.42)$$

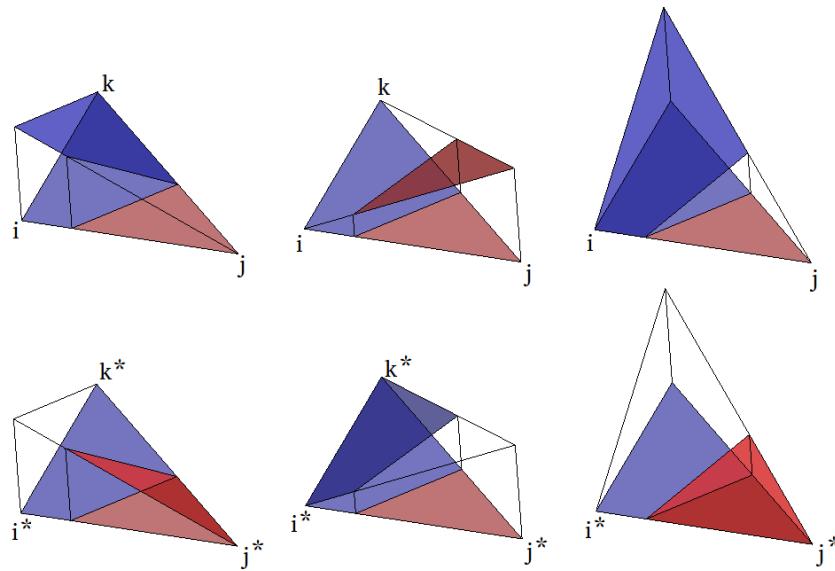


Figure 3.8: Pressure shape functions for the Monolithic strategy:
Top: Replacement functions. Bottom: Enrichments

Condensing the enrichment DoFs the initial size of the system is recovered. For simplification, the $\frac{1}{\rho}$ is omitted in the Laplacian terms.

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{1}{\Delta t} \mathbf{M}(\rho) - \frac{1}{\tau} \mathbf{D}_* \mathbf{L}_{**}^{-1} \mathbf{D}_*^T & \mathbf{D}_N^T - \mathbf{L}_{*N} \mathbf{L}_{**}^{-1} \mathbf{D}_*^T \\ \mathbf{D}_N - \mathbf{L}_{N*} \mathbf{L}_{**}^{-1} \mathbf{D}_* & \tau (\mathbf{L}_{NN} - \mathbf{L}_{*N} \mathbf{L}_{**}^{-1} \mathbf{L}_{N*}) \end{bmatrix} \begin{bmatrix} \mathbf{V}^{n+1} \\ p_N^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M}\mathbf{V}^n + \mathbf{Mg} - \mathbf{D}_* \mathbf{L}_{**}^{-1} \mathbf{D}_*^T \mathbf{g} \\ \tau (\mathbf{D}_N^T \mathbf{g} - \mathbf{L}_{*N} \mathbf{L}_{**}^{-1} \mathbf{D}_*^T \mathbf{g}) \end{bmatrix} \quad (3.43)$$

Unlike the Fractional Step, the above system (3.43) has the advantage of not requiring the pressure to be recovered later, thus simplifying the algorithm. Implemented in this way, accurate results are possible by solving the system only once, unlike the segregated scheme developed, which requires at least two or three iterations of the pressure system to avoid severe mass losses. Taking this into account, the global computation time of the monolithic solver is roughly the same as the FS in most cases. Nonetheless, in bad-conditioned cases the monolithic can be twice as slow or more.

This worse performance may be seen as an important drawback, however it is not a critical issue since simulations can be easily sped up by increasing the number of cores due to the parallel implementation of the algorithm. The simplification provided by the monolithic set of equation outperforms the slightly higher computational cost since less time is required to translate physical laws into the code.

3.4.5 Further enrichments for the velocity field

The previous sections described the general methodology to tackle any fluid problem governed by two or more fluids with different properties. The enriched pressure gradient allows for an accurate reproduction of the pressure field and also reduces the mass losses, which can be an issue in free surface flows (Coppola, 2009) with Finite Elements.

However the velocity field has not been modified, which implies that the velocity gradient is continuous inside each element. While this is not a problem in fluids where one dominates over the other (such as water and air), it can a severe restriction in fluids with similar properties.

Figure 3.9 shows the real profile of a debris flow and the one obtained using a continuous

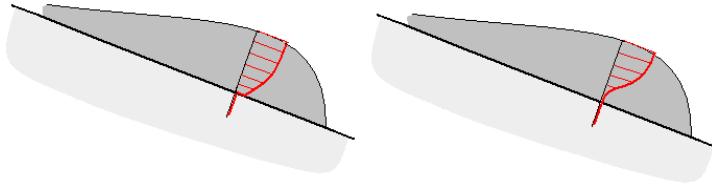


Figure 3.9: Real and simulated velocity profiles with restrictive interface conditions

velocity field of standard Finite Elements where the interface does not match element faces. It is clear that the latter is not able to reproduce the phenomena correctly, leading to zero gradient of velocity at the base of the flow, therefore reducing the run over distance of the flow. Moreover, forces are also miscalculated and erosion and deposition rates are not accurate.

Based on this observation and taking advantage of the easier implementation in the monolithic strategy, an enrichment for the velocity field was added for the velocity. It is critical to avoid over relaxing the system of equations, otherwise the condensed submatrix becomes non invertible and the solution is not possible. Due to the incompressibility constraint, discontinuous values of the velocity in the normal direction are not physically possible. On the other hand, discontinuities in the tangential velocity can cause numerical instabilities the system. Based on this, the problem lies in allowing a different gradient of the velocity at both sides of the interface. This means adding two new degrees of freedom (in 2D) in order to improve the definition of the interface. One in the direction of the interface and another perpendicular to it.

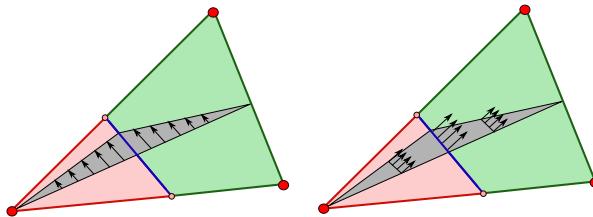


Figure 3.10: Enrichments for the velocity field

The employed DoF can be seen in Figure 3.10. As stated before, it is important not to over relax the system. In this line, the divergence of the enrichment velocities cannot be integrated in the standard way. This would leave the edges of the element as free inlet-outlets. Since this DoFs are condensed, no information is shared between neighbour elements and therefore velocities could raise without control, only limited by the viscous

forces. To avoid this, the divergence of the enrichment velocities has to be integrated by parts: $\int_{\Omega}(-\nabla q, \mathbf{V}_* d) d\Omega = 0$. In the same way as the Fractional Step strategy, omitting the boundary terms will lead to weak divergence-free inside the elements for the enrichment velocities. This will give the element the extra flexibility required while keeping spurious velocities under control.

The system is rotated to the direction of the interface to allow different values of viscosity in the two directions. Therefore they must be locally assembled following the next steps:

- Determine interface position and angle and rotate the element to add enrichments in the local \mathbf{x}' (3.11)
- Assemble standard matrix in local coordinates
- Calculate enrichment DoFs and assemble into the local system (3.44)

$$\begin{bmatrix} \mathbf{K}_{NN} + \frac{1}{\Delta t} \mathbf{M}_N & \mathbf{K}_{N*} & \mathbf{D}_{NN}^T & \mathbf{D}_{N*}^T \\ \mathbf{K}_{*N} & \mathbf{K}_{**} + \frac{1}{\Delta t} \mathbf{M}_* & \mathbf{D}_{*N}^T & \mathbf{D}_{**}^T \\ \mathbf{D}_{NN} & \mathbf{D}_{*N} & \tau \mathbf{L}_{NN} & \tau \mathbf{L}_{N*} \\ \mathbf{D}_{N*} & \mathbf{D}_{**} & \tau \mathbf{L}_{*N} & \tau \mathbf{L}_{**} \end{bmatrix} \begin{bmatrix} \mathbf{V}_N^{n+1} \\ \mathbf{V}_*^{n+1} \\ p_N^{n+1} \\ p_*^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M}_N \mathbf{V}_N^n + \mathbf{M}_{NG} \\ \frac{1}{\Delta t} \mathbf{M}_* \mathbf{V}_*^n + \mathbf{M}_* \mathbf{g} \\ \tau \mathbf{D}_N^T \mathbf{g} \\ \tau \mathbf{D}_*^T \mathbf{g} \end{bmatrix} \quad (3.44)$$

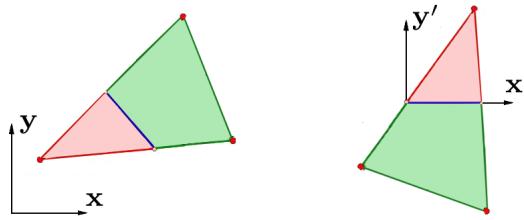


Figure 3.11: Global and local coordinate systems.

Now reordering the equations to ease the condensation procedure:

$$\left[\begin{array}{cc|cc} \mathbf{K}_{NN} + \frac{1}{\Delta t} \mathbf{M}_N & \mathbf{D}_{NN}^T & \mathbf{K}_{N*} & \mathbf{D}_{N*}^T \\ \mathbf{D}_{NN} & \tau \mathbf{L}_{NN} & \mathbf{D}_{*N}^T & \tau \mathbf{L}_{N*} \\ \hline \mathbf{K}_{*N} & \mathbf{D}_{*N}^T & \mathbf{K}_{**} + \frac{1}{\Delta t} \mathbf{M}_* & \mathbf{D}_{**}^T \\ \mathbf{D}_{N*} & \tau \mathbf{L}_{*N} & \mathbf{D}_{**} & \tau \mathbf{L}_{**} \end{array} \right] \begin{bmatrix} \mathbf{V}_N^{n+1} \\ p_N^{n+1} \\ \hline \mathbf{V}_*^{n+1} \\ p_*^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M}_N \mathbf{V}_N^n + \mathbf{M}_{NG} \\ \tau \mathbf{D}_N^T \mathbf{g} \\ \hline \frac{1}{\Delta t} \mathbf{M}_* \mathbf{V}_*^n + \mathbf{M}_* \mathbf{g} \\ \tau \mathbf{D}_*^T \mathbf{g} \end{bmatrix} \quad (3.45)$$

To simplify notation, the system is written as:

$$\left[\begin{array}{c|c} \mathbf{A}_{NN} & \mathbf{A}_{N*} \\ \hline \mathbf{A}_{*N} & \mathbf{A}_{**} \end{array} \right] \left[\begin{array}{c} \mathbf{X}_N \\ \hline \mathbf{X}_* \end{array} \right] = \left[\begin{array}{c} \mathbf{F}_N \\ \hline \mathbf{F}_* \end{array} \right] \quad (3.46)$$

- Condense the new DoFs

$$\tilde{\mathbf{A}}\mathbf{X}_N = \tilde{\mathbf{F}} \quad (3.47)$$

where

$$\tilde{\mathbf{A}} = \mathbf{A}_{NN} - \mathbf{A}_{N*}\mathbf{A}_{**}^{-1}\mathbf{A}_{*N} \quad \tilde{\mathbf{F}} = \mathbf{F}_N - \mathbf{A}_{N*}\mathbf{A}_{**}^{-1}\mathbf{F}_*\tilde{\mathbf{F}} \quad (3.48)$$

- Rotate the matrix to Global coordinates (Oñate, 2013)

$$\tilde{\mathbf{A}}_{ij \text{ global}} = \mathbf{R}^T \tilde{\mathbf{A}}_{ij} \mathbf{R} \quad (3.49)$$

where \mathbf{L} is the rotation matrix composed by the director cosines. $\lambda_{x'x}$ is the cosine of the angle defined between the local axis x' and the global axis x .

$$\mathbf{R} = \begin{bmatrix} \lambda_{x'x} & \lambda_{x'y} & \lambda_{x'z} & 0 \\ \lambda_{y'x} & \lambda_{y'y} & \lambda_{y'z} & 0 \\ \lambda_{z'x} & \lambda_{z'y} & \lambda_{z'z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.50)$$

By adding these new degrees of freedom , it is possible to create a discontinuity in the gradient of the velocity field across the interface, allowing for arbitrary velocity profiles on both sides without interdependence. It is important to note that in the previous formulation no special transmission conditions were written, so stress continuity is being assigned.

3.4.6 Complete algorithm

The monolithic algorithm allows to solve accurately problems with high viscosity, unlike the segregated counterpart. Another advantage is that the strategy to solve the FE system is much more simple since it is not required to recover the solution in order to advance in time. Also, supposing that the explicit convective term is accurate enough, it is only necessary to solve the system only once per time step. As it will be seen in the example sections, this leads to comparable computation times between the monolithic and the segregated strategy. Below are shown the complete steps to advance in time at each step.

Step 1) Convect the particles using the streamlines

Step 2) Project information into the mesh,

with q fluid particles

$$\hat{\mathbf{V}}_j^{n+1} = \frac{\sum_i^{r+q} \mathbf{V}_i N_i^j}{\sum_i^{r+q} N_i^j}$$

$$\varphi_j^{n+1} = \frac{\sum_i^n \text{sign}_i N_i^j}{\sum_i^n N_i^j}$$

Step 3) Detect interface elements

Step 4) Assemble and solve the system of equations with enrichments

Step 5) Update particles' velocity

3.5 Test Examples

3.5.1 Fractional Step

Rayleigh-Taylor instability

The classical benchmark for multi fluids was tested to verify the solver. Despite there are no analytical results or experimental data in 2D to check the error, it is possible to compare against other numerical results in the literature. He et al. (1999) show the results for Reynolds number $Re = 256$ for a Lattice Boltzmann scheme. Using the same initial conditions, results for other Re can be found in (Guermond and Quartapelle, 2000).

The geometry has the dimensions shown in 3.12. The parameters of the simulation are:

$$\rho_q = 3 \text{ kg/m}^3 \quad \nu_q = \frac{\sqrt{g} \cdot d^{3/2}}{Re} \text{ m}^2/\text{s}$$

$$\rho_p = 1 \text{ kg/m}^3 \quad \nu_p = \frac{\sqrt{g} \cdot d^{3/2}}{Re} \text{ m}^2/\text{s}$$

$$g_y = -10 \text{ m/s}^2$$

$$\delta t = 0.01 \text{ s}$$

Initial Conditions:

$$V = 0 \text{ in } \Omega$$

$$\varphi = -y - \delta_0(\cos(2\pi(x) - \pi) + 1) + 2.0 \quad \text{where} \quad \delta_0 = 0.1$$

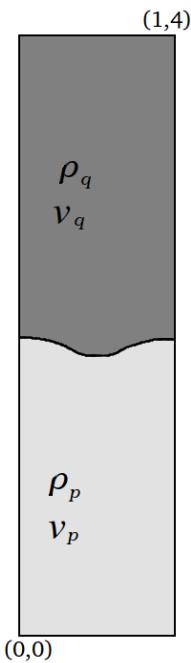


Figure 3.12: Rayleigh-Taylor instability configuration

In Figure (3.14), snapshots at different time steps can be seen. For the Reynolds numbers analysed, the results obtained match the ones found in the literature. Adimensionalized time $t_{adim} = t\sqrt{g/2}$ is used. Finally the position of the advancing front of the denser fluid (centre) and the rising of the bubbles (sides) are compared against the results of Guermond and Quartapelle (2000). Again good matching is observed.

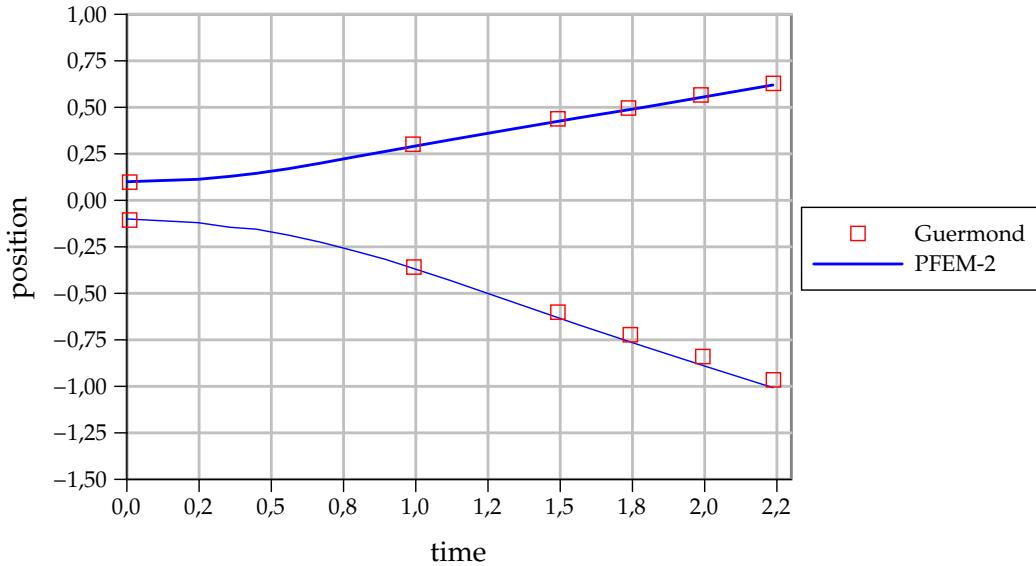


Figure 3.13: Vertical position of the advancing front(lower line) and rising left and right bubbles (upper line) in the Rayleigh-Taylor instability problem

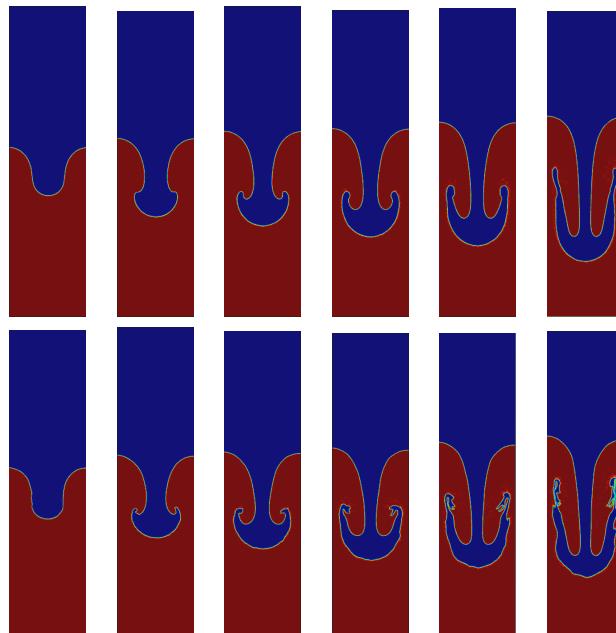


Figure 3.14: Snapshots for the Rayleigh-Taylor instability problem at $t_{adim} = 1.0, 1.5, 1.75, 2.0, 2.25, 2.5$. Top: $Re = 256$; Bottom: $Re = 1000$.

Sloshing

A simple case of sloshing was selected to test the speed of the code against an industry-standard software for the simulation of fluids. The program chosen was OpenFoam (Weller et al., 1998) due to the large community supporting it and the fact that it is Open Source. The geometry and initial conditions of the experiment can be seen in Figure 3.15. The initial velocity is zero in the whole domain, composed of a total of 120,000 unstructured elements.

$$\rho_q = 1 \text{ kg/m}^3 \quad \nu_q = 0 \text{ m}^2/\text{s}$$

$$\rho_p = 1000 \text{ kg/m}^3 \quad \nu_p = 0 \text{ m}^2/\text{s}$$

$$g_y = -10 \text{ m/s}^2$$

$$\delta t = 0.005 \text{ s}$$

$$0 < t < 5\text{s}$$

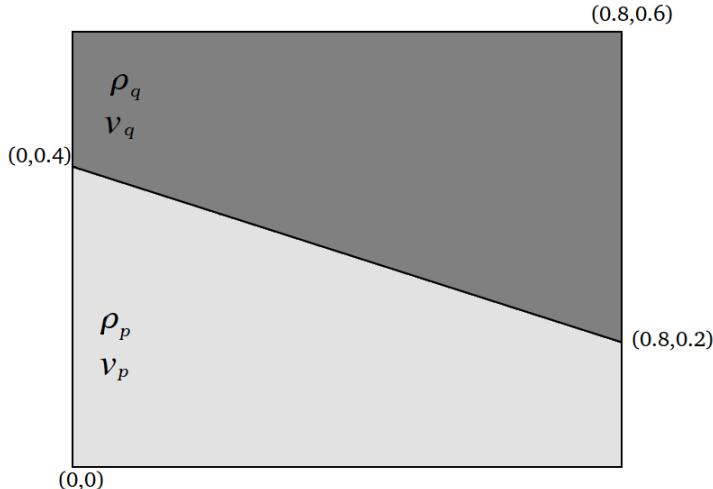


Figure 3.15: Sloshing Geometry

As for the parameters in OpenFoam, they were set to reduce the computation time as much as possible while maintaining accuracy in time evolution of the numerical experiment:

- Mesh size: 0.003 m
- OpenFOAM: variable Dt with a target $Co = 2$

- Solver: interFoam
- Number of PIMPLE correctors: 3
- U tolerance: $1e - 6$
- p tolerance: $1e - 7$

Comparison for different snapshots can be seen in Figure 3.16. The two solvers show the same time evolution. Finally in the Table 3.1 computation times are presented. In all the cases the developed code is faster than OpenFoam due to the larger time steps.

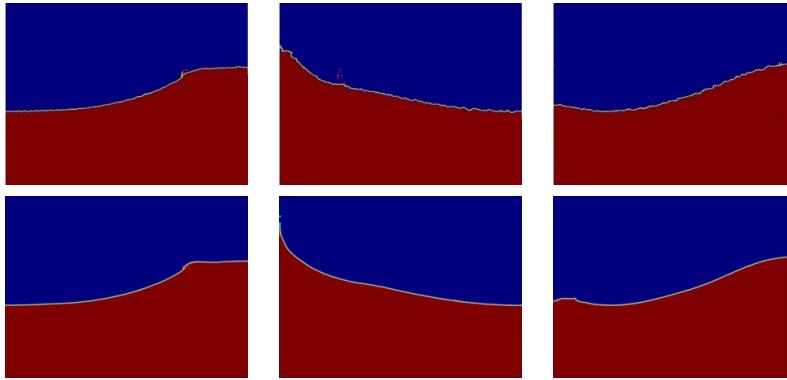


Figure 3.16: Snapshots for the sloshing simulation at $t = 0.55, 1.15, 1.70$. Top: PFEM-2, Bottom: OpenFoam

Code	1 Core	2 Cores	4 Cores
OpenFoam	754	402	286
PFEM-2	284	179	138

Table 3.1: Computation time (in seconds) to simulate 1s of the sloshing experiment

Dam Break with obstacle

Following the experiment proposed by Koshizuka et al. (1995), numerical simulations were done to test the accuracy of the strategy. The domain is a closed box. The water column is held by a wall that is suddenly removed and the flow begins. The initial configuration can be seen in Figure 3.17.

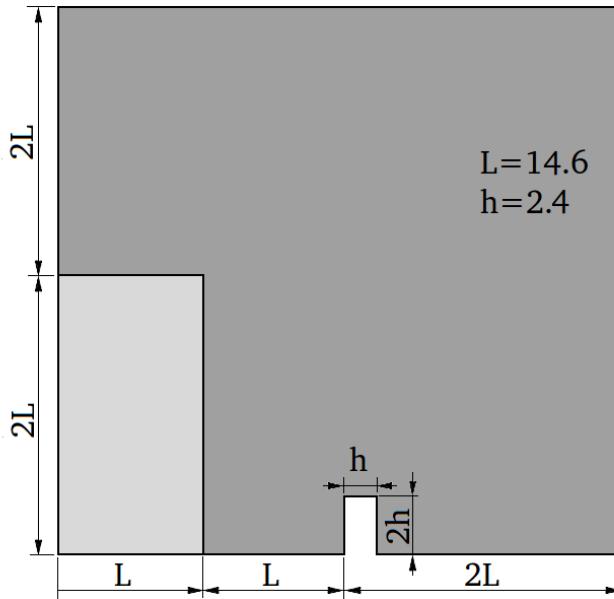


Figure 3.17: Dam Break experiment configuration

The velocity is zero in the whole domain. As for the time step used, it was set in $\delta t = 0.005[s]$. Although the code can support larger values, if $\delta t >= 0.01$ is used, the solution diverges from the experimental data since the physics of the impact are lost. Mesh sizes from $h = 0.02$ to $h = 0.0015$ were used in this example. Unfortunately mesh convergence cannot be analysed in this problem due to the low quality of the pictures and the lack of reliable variables to do so. Still, this example provides results of a real experiment to verify the robustness of the algorithm and independence of stability on the time-step. In fact, in all the meshes $\delta t = 0.005$ was maintained, leading to Courant numbers as high as $Co = 40$ for the finest mesh.

Figure 3.18 shows the comparison between the real experiment and the simulation with the developed formulation for a mesh size $h = 0.03$, with a total of 120,000 elements. Results are qualitatively accurate with short computation times, requiring only 130 seconds to solve the problem despite the fine mesh. It is interesting to note that, despite not visible, air is also calculated with the aid of the enrichment functions for the interface. This allows for the simulation of the interaction between the two fluids, reason why the water is suspended by the trapped air in the last frame of the Figure 3.18. The same example was run in 3D, as seen in figure 3.19. Again the results of the simulation match the real experimental data.

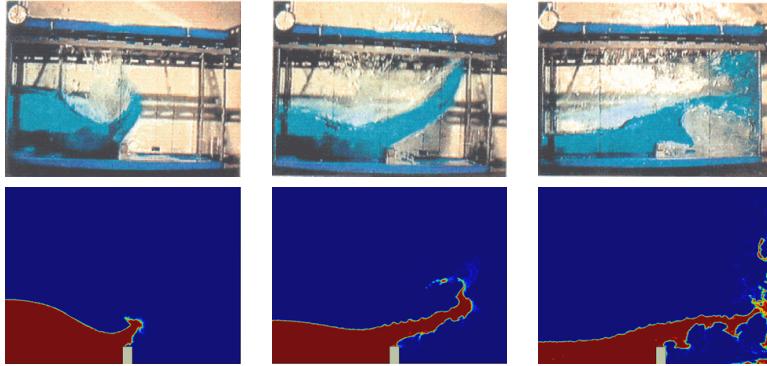


Figure 3.18: Initial Geometry and snapshots for the experimental and numerical results at $t = 0.3, 0.4, 0.5$

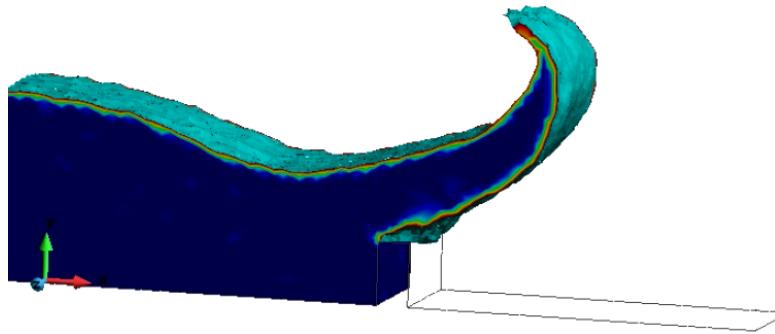


Figure 3.19: Snapshot for the simulation in 3D

3.5.2 Monolithic Strategy

Convergence test

To verify that the implementation of the PFEM-2 is correct, a single fluid test is used as a benchmark. Computing the errors in a simple problem is vital in order to assess correctly the sources of errors in more challenging problems, such as two-fluid and FSI cases. The example proposed by Codina (2001) is a manufactured problem with analytical solution and therefore it is ideal to test the convergence of the solver. The geometry is a square domain of a unique fluid with density $\rho = 1$ and viscosity $\mu = 0.1$ and 1×1 , where a body force is imposed such that the solution must be:

$$\begin{aligned} u_x &= 100 f(x) f'(y) \\ u_y &= 100 f(y) f'(x) \end{aligned}$$

where

$$f(x) = x^2(1 - x)^2$$

The time domain extends in the range $[0, 1]$, and the errors are analysed at $t = 1$. The solution for the finest mesh is shown in Figure 3.20. To analyse the convergence, the first step is determining the required minimum time step to guarantee that the error of the solution is dominated by the spatial discretization rather than time integration errors. The finest mesh ($h = 0.01$) is employed to test the time-step convergence. In Figure 3.21 the results are presented. As it can be seen, the error decreases until $\Delta t = 0.002$, where the error is dominated by the mesh size. Another important aspect from this graph is that three distinct regions can be identified: In the region of $\Delta t < 0.001$, the error is dominated by the mesh discretization and therefore the slope is gentle. In the mid region $0.001 < \delta t < 0.01$, quadratic convergence is observed and finally a linear behaviour is spotted for larger time steps. The reason behind this poor behaviour for $\Delta t > 0.01$ is that this problem is dominated by the body forces and in this work a backward Euler scheme is used to compute these terms, which provides only linear convergence in time. When the position of a particle $\mathbf{x}_p^{(t)}$ changes considerably from one time step to the next, evaluating the integrals only in the final position $\mathbf{x}_p^{((n+1))}$ translates into the observed errors. On the other hand, for most problems of engineering interest, the only acting body force is the gravity, which is constant in the whole spatial and temporal domain and therefore a backward Euler integration scheme is exact.

The mesh convergence for elemental sizes from $h = 0.1$ to $h = 0.01$ is shown in Figure 3.22. The slope lies in between the linear and quadratic convergence rates, at approximately $1.5\times$. The implementation of the PFEM-2 solver done in this work resembles an enhanced Stokes solver coupled with a particle convection strategy. Since linear Stokes elements provide quadratic mesh convergence for smooth problems such as this example, a lower convergence rate can only be explained by the projection of particles to and from the fixed mesh. This stationary example proves that transferring information with the current Kernel implementation is an important source of errors, limiting the convergence rate to a maximum of approximately $1.5x$.

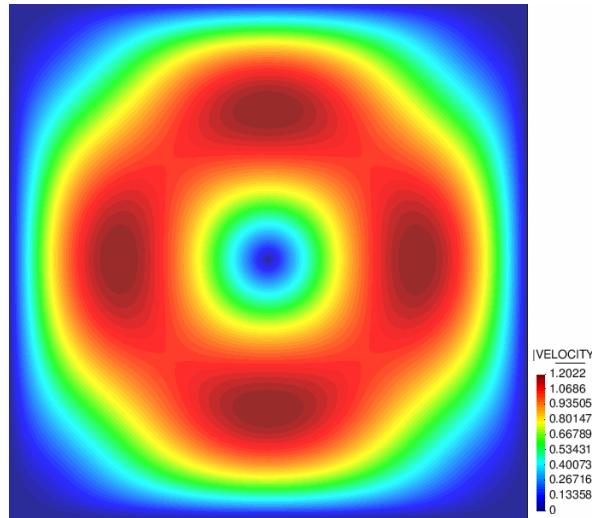


Figure 3.20: Snapshot for manufactured problem

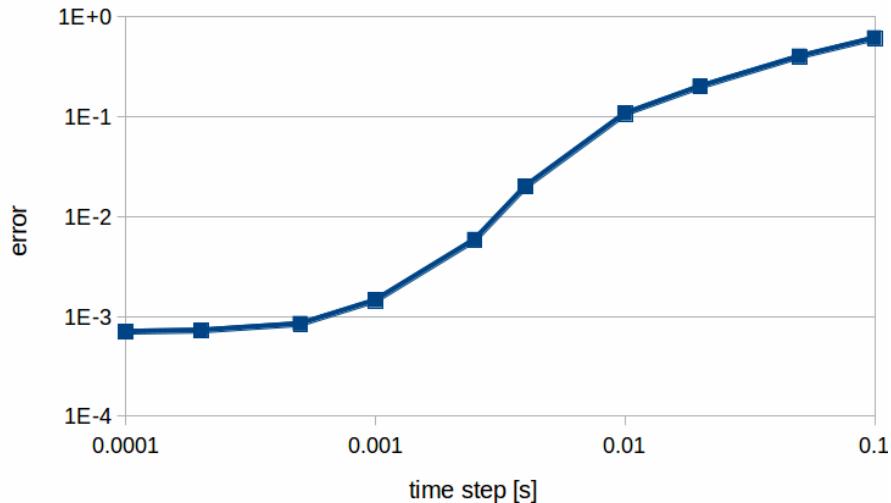


Figure 3.21: Time convergence for the manufactured problem

Sloshing

This problem was used to test the computational time of the fractional step PFEM-2 solver against a well established code. Running the same example is useful to compare the efficiency of the monolithic solver against the FS implementation. The geometry is reproduced again in Figure 3.23. The parameters of the solver are kept identical to the FS case. All enrichment functions (velocity and pressure) are active, although the only extra needed DoF is an improved pressure gradient at the interface. Since the

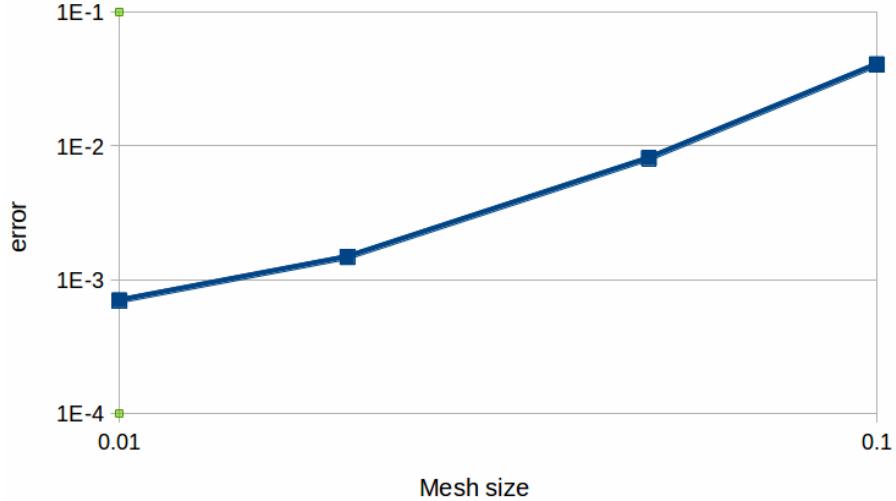


Figure 3.22: Mesh convergence for the manufactured problem

simulated phenomena is relatively calm without braking waves, the OSS stabilization is chosen over the PSPG method. Except for the solving of the linear equations on the mesh, all the other components of the PFEM-2 strategy remain unchanged to ease the comparison of the two implementations.

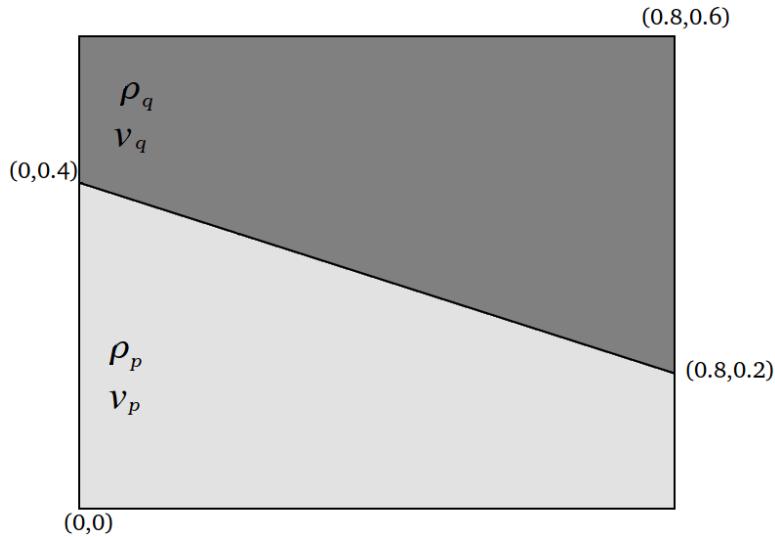


Figure 3.23: Sloshing Geometry

The snapshots of the solution are not presented here since they are almost identical to the FS solution, showing a correct implementation. On the other hand, the computation time required for one second of simulation is larger. Using the monolithic implemen-

tation, 182 seconds of computer computations are needed, against the 138 of the FS strategy, representing an increment of 32 % with four cores of the *i7 – 3820* processor. The time difference is relatively low due to the fact that the analysed computer time includes both the particle tasks and the solving of the linear system. If only the time spent by the solver is considered, the clock time is doubled compared to the FS (120 s. vs 60 s.).

Couette flow

The aim of this example is to determine the improvement in the solution caused by the enriched velocity field. To avoid possible sources of error and focus solely on the effect of the modified velocity field, all tasks related to particles must be deactivated, therefore obtaining the equations of a Stokes solver.

The Couette flow is a good example to test the accuracy of the velocity enriched Stokes solver. This problem consists on a two dimensional channel with a constant pressure gradient along the x coordinate, as seen in Figure 3.24. The pressure is imposed at $p_1 = 10 \text{ Pa}$ in the right and $p_2 = 0 \text{ Pa}$ in the left, obtaining a pressure gradient of $dp/dt = 1 \text{ Pa/m}$. In the stationary case, the pressure forces are balanced by the drag against the upper and lower walls, where no-slip boundary conditions are assigned. On the right side of the picture, the expected velocity profile is visible. The viscosities are set as $\mu_q = 100 \text{ Pa.s}$ and $\mu_p = 10 \text{ Pa.s}$. Due to the discontinuity of this property across the interface, the gradient of the velocity will be discontinuous.

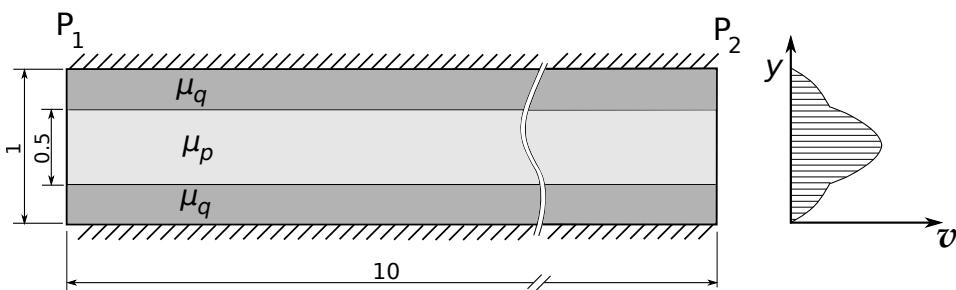


Figure 3.24: Couette flow problem geometry

For the laminar, stationary case shown in the 3.24, the Navier Stokes equations simplify from the original tridimensional nonlinear equation into the following expression, where

μ is the dynamic viscosity:

$$\frac{d^2v}{dy^2} = \frac{1}{\mu} \frac{dp}{dx} \quad (3.51)$$

Integrating the equation two times and knowing that $dp/dx = -1$

$$v = -\frac{1}{2\mu}y^2 + C_1y + C_2 \quad (3.52)$$

Since there is a discontinuity in the material properties at the interface, the equation will have to be solved for two different subdomains p and q . In this sense, two boundary conditions can be used to simplify the procedure to obtain the four constants ($C_{1q}, C_{2q}, C_{1p}, C_{2p}$). On one side, the velocity must be zero in the wall, that is, $v_{y=0} = 0$. On the other hand, since there are no discontinuities in the derivatives at $y = 0.5$ and knowing that the problem is symmetric, the derivative of the velocity must be zero at the interface: $dv/dy|_{y=0.5} = 0$. Using these BC, we obtain:

$$v_q = -\frac{1}{2\mu_q}y^2 + C_{1q}y \quad v_p = -\frac{1}{2\mu_p}(y^2 - y) + C_{2p} \quad (3.53)$$

To obtain the two missing constants, it is possible to write the equations that must be fulfilled at the interface: The velocity cannot have discontinuities and therefore $v_p|_{y=0.25} = v_q|_{y=0.25}$ and also the shear stresses must be in equilibrium. For Newtonian fluid this implies that $\frac{dv_q/dy|_{y=0.25}}{dv_p/dy|_{y=0.25}} = \frac{\mu_p}{\mu_q}$.

$$\begin{cases} -\frac{1}{2\mu_q}(\frac{1}{4})^2 + (\frac{1}{4})C_{1q} = -\frac{1}{2\mu_p}((\frac{1}{4})^2 - (\frac{1}{4})) + C_{2p} \\ \left(-\frac{1}{\mu_q}(\frac{1}{4}) + C_{1q}\right)\mu_q = \left(-\frac{1}{\mu_p}(\frac{1}{4}) + \frac{1}{2\mu_q}\right)\mu_p \end{cases} \quad (3.54)$$

Solving the resulting system of equations, the final velocity profile can be expressed as:

$$v(y) = \begin{cases} -\frac{1}{2\mu_q}(y^2 - y) & y < 0.25 \text{ or } y > 0.75 \\ -\frac{1}{2\mu_p}(y^2 - y) + \frac{3}{32}(1/\mu_q - 1/\mu_p) & 0.25 \leq y \leq 0.75 \end{cases} \quad (3.55)$$

The numerical results can be compared against the analytical solution (3.55), providing the possibility to check the mesh convergence of both enriched and non-enriched solutions. Since the analysed case is a stationary Stokes problem, the use of particles is not necessary and therefore errors related to projection are omitted. Structured meshes are created to ensure that the interface will always cut across the middle of the elements, to test the worst case scenario. In Figure 3.25 the velocity contour is plotted for the mid section of the domain.

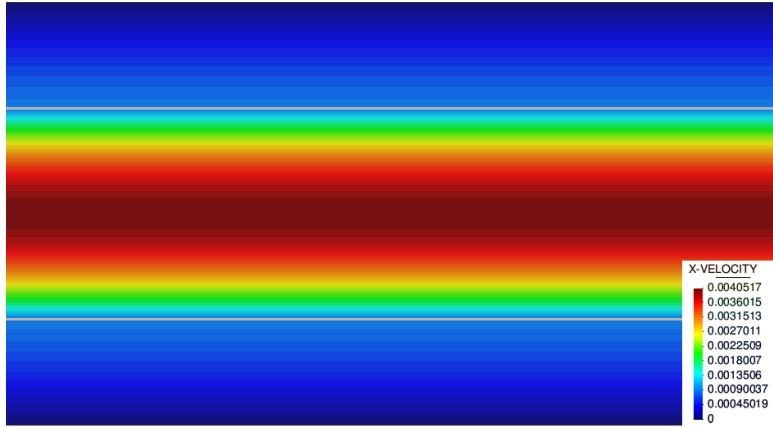


Figure 3.25: Couette problem velocity contour

The mesh convergence of the problem is plotted in Figure 3.26. As expected, the enriched solution offers a better solution compared to the non-enriched case. However, despite the error is one order of magnitude lower, the convergence rate remains linear. The problem lies in the fact that there is no inter-elemental continuity in the viscous shear stresses of the enrichment velocity functions. The boundary terms $\int_{\Gamma} \mathbf{n} \cdot \boldsymbol{\mu} \nabla \mathbf{V} d\Gamma$ are not guaranteed to be in equilibrium across the interface due to the condensation procedure, therefore not improving the linear convergence rate despite the better representation of the field. This case is analogous to the pressure boundary terms of the FS solver. Unfortunately, the strategy to raise the linear convergence used in the FS solver cannot be used in this case; the viscous terms cannot be integrated differently. Still, despite linear convergence is certainly not optimal for linear elements, the proposed solution is still significantly better than the non-enriched methodology.

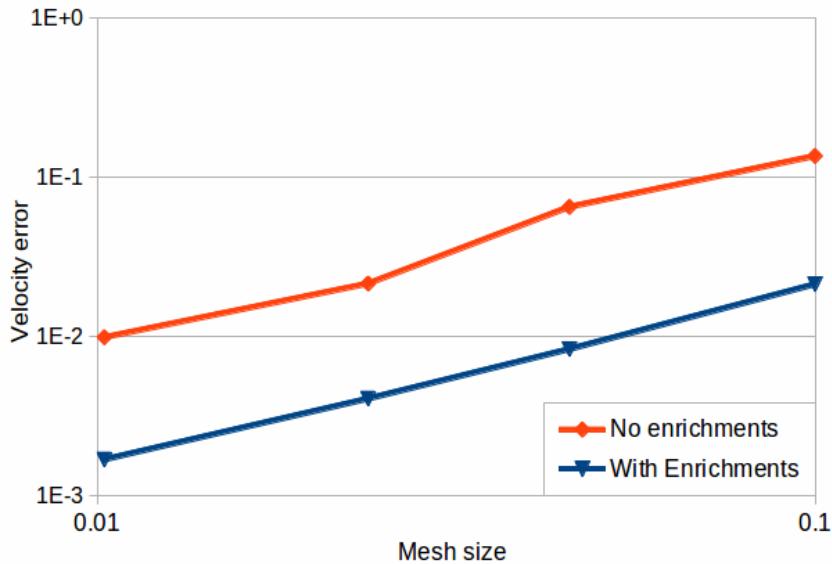


Figure 3.26: Couette problem mesh convergence

Rigid cylinder in viscous fluid

Unlike the FS solver, the monolithic strategy can solve accurately problems of any viscosity. This property can be used to simulate rigid bodies by setting a deliberatively high viscosities in these areas. The purpose of this example is to check the accuracy of the solver to simulate this type of problems, in which neither the stresses nor deformations of the solid body are of interest.

Figure 3.27 shows the geometry of rigid cylinder in a high viscosity fluid. The domain is under gravitational acceleration and the density of the cylinder is larger than the one of the fluid. Slip Boundary conditions are set on the walls, and the gravity value is set to $9.81m/s^2$. The body must accelerate until it reaches a certain terminal velocity, which happens when the drag resistance and force resulting from the density difference are balanced. Analytical expressions for the velocity have been derived by Happel and Brenner (1983) using stream functions. Unfortunately, the range of validity of those expressions are only valid for a small range of low Reynolds numbers and the values of viscosity and density used in this example lie outside the admissible ranges of the hypothesis. On the other hand, Gil et al. (2010), Wang and Liu (2004) and Franci (2015) have obtained numerical results for the parameters used in this example. Despite using different techniques, they have all converged to the same value: the terminal velocity

can be estimated at around $U = 0.0377\text{m/s}$. This magnitude will be used to check the accuracy of the analysed strategy and the mesh convergence rate.

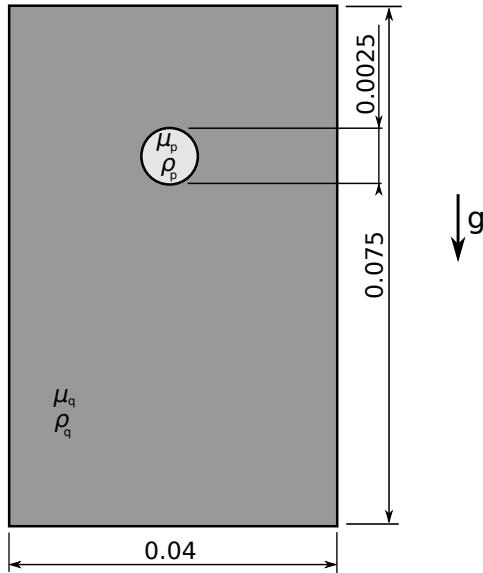


Figure 3.27: Rigid cylinder in viscous fluid

The parameters of the simulation are as follows, where μ is the dynamic viscosity:

$$\rho_q = 1000 \text{ kg/m}^3 \quad \nu_q = 0.1 \text{ Pa.s}$$

$$\rho_p = 1200 \text{ kg/m}^3 \quad \mu_p = 10000 \text{ Pa.s}$$

$$g_y = -9.81 \text{ m/s}^2 \quad \delta t = 0.001 \text{ s}$$

The evolution of the vertical velocity of the cylinder is plotted in Figure 3.28. As it can be seen, there is a fast acceleration at the beginning with an asymptotic behaviour towards the end of the simulation. At approximately $t = 1 \text{ s}$ the velocity is stabilized, with good matching compared to the results of other authors. In Figure 3.29 the velocity contour of the problem at $t = 1 \text{ s}$ can be seen. The small perturbations in the velocity of 3.28 are caused by the errors of the projection from the particles into the mesh.

Mesh sizes from $2 \times 10^{-3} \text{ m}$ to $1 \times 10^{-4} \text{ m}$ were used to check the convergence. For all sizes the time step was fixed at $\Delta t = 0.001 \text{ s}$. As it can be seen in Figure 3.30, the mesh convergence rate is only linear due to the jump in the viscosity across the interface. The same problem was computed without enrichments, showing exactly the same convergence curve as the enriched case. Despite the addition of the velocity enrichments

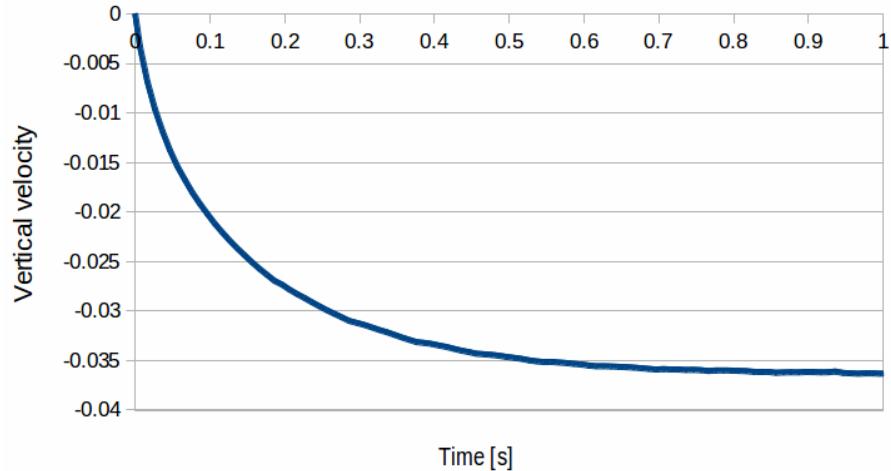
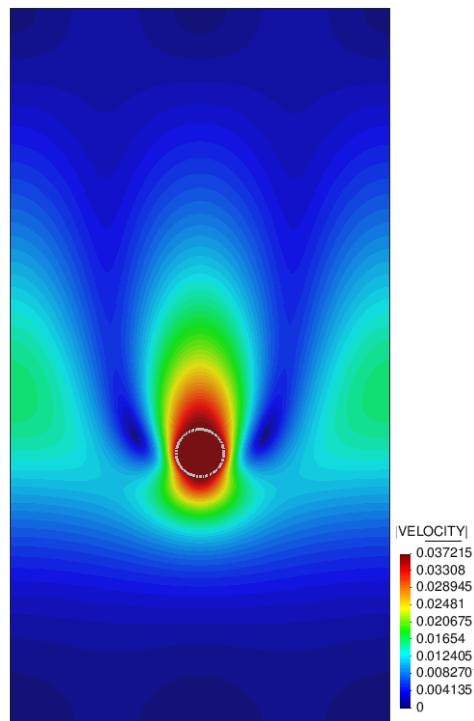


Figure 3.28: Vertical velocity of the cylinder

Figure 3.29: Snapshot of velocity contour at $t=1s$. The location of the interface is shown in grey

should, in principle, improve the solution of the problem, it has not changed significantly the error in the velocity field. While this may seem erroneous since the error was reduced in the Couette problem, it must be noted that for that example the convergence

remained linear, only altered by an arbitrary constant. In the same way, this problem shows the same convergence rate for the enriched and non-enriched formulations, but with a error constant that is likely negligible and cannot be appreciated. This difference might be caused by the projection errors and the lack of previous enrichment velocity $\mathbf{V}_*{}^n = \mathbf{0}$, meaning that it is resetted to zero at each time step. As a result, despite there is an active enrichment velocity DoF, the impossibility to make an estimation on its historical value translates into a more diffusive behaviour, almost equivalent to using no velocity enrichments.

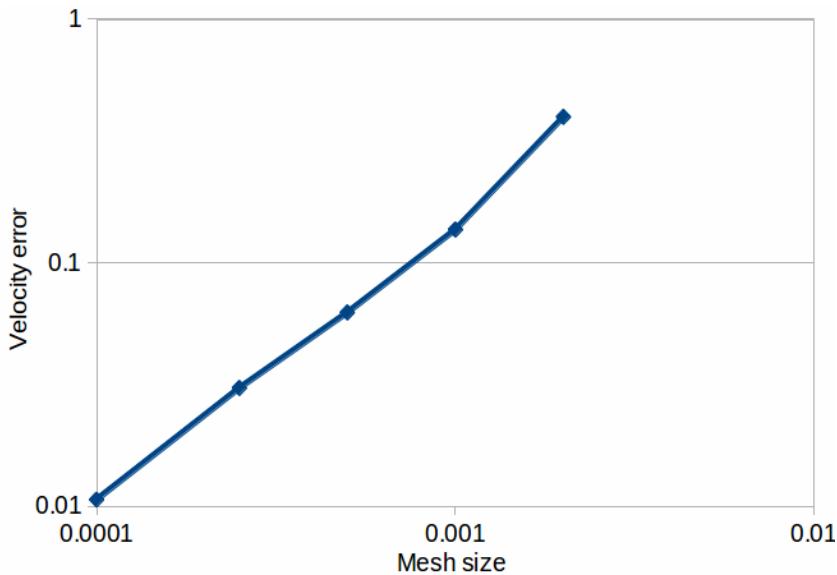


Figure 3.30: Relative velocity mesh convergence error for the falling cylinder case

Non-academic example

The problem of a powerboat sailing over waves is presented to test the code in a non academic example. The geometry of the boat is shown in Figure 3.31. It consists on a twin hull design (catamaran) joined by a link. The dimensions are shown in the same figure.

One of the requirements of the project is that the selected tool must be capable of simulating the spray caused by the interaction between the hull and the waves. The results will be used to evaluate how much spray reaches the central structure of the

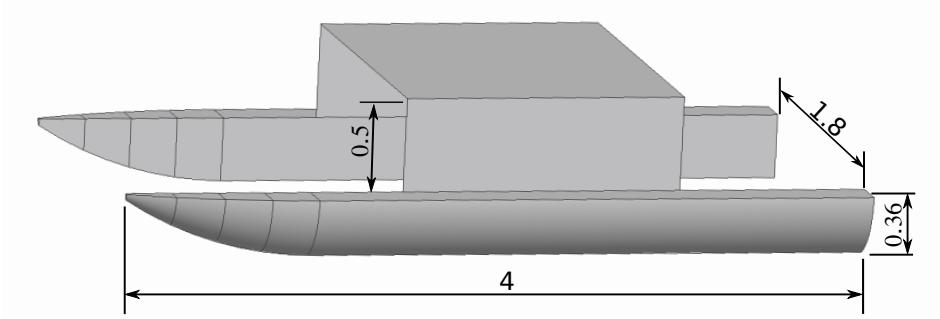


Figure 3.31: Boat Geometry

boat, which is important to asses the visibility in the central section. In this sense, the use of the PFEM-2 strategy allows to represent water volumes that are below the mesh scale, by just plotting individual water particles moving through the air. To improve the behaviour of these particles, a parabolic trajectory was implemented to replace the streamline integration. The convection stage is performed this way until they are back on the water surface, when the streamline integration is recovered. To allow a smaller size of droplets, the maximum number of particles per elements was set to 24.

The simplified geometry for the simulation is shown in Figure 3.32. Due to symmetry, only half is created and slip boundary conditions are set in the mirror wall, at $y = 0$. In order to get a good resolution, an element size of $H = 0.05m$ is chosen, having a total of 3 million elements in the computational domain and more than 50 million particles. Despite most of the memory required by the solver is used to store the particles, the problem can be computed without difficulties in a desktop computer.

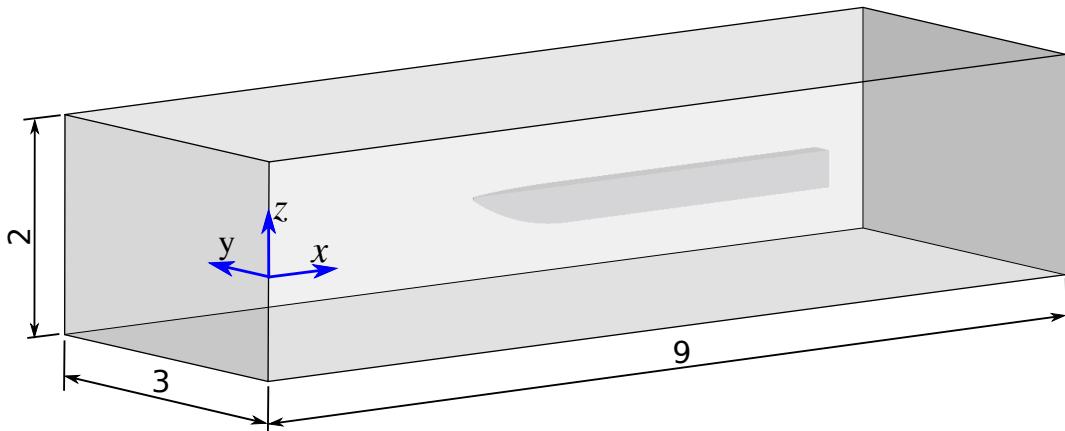


Figure 3.32: Boat Computational domain

Finally, the boat must be able to experience rigid body motions in order to simulate the interaction between the hull and the water surface. Despite the a Fluid-Structure interaction solver is not yet implemented for objects that are not defined by particles, it is possible to impose the deformation to obtain a first approximation. In this sense, the Lagrangian approach of the PFEM-2 strategy allows to solve the problem in a different mesh at each time step without modifications in the FE formulation. In fact, only the convection stage must be adapted to the new, deformed geometry. As for the deformation of the FE mesh, a standard Arbitrary Lagrangian-Eulerian tool is used, solving a Poisson problem to obtain the new coordinates of the nodes. The nodal points are fixed in the boundaries in the domain and in the hull, obtaining the rest of the nodes by solving a diffusion problem. It must be noted that this technique does not guarantee that the elements will not become inverted, so the deformations must be relatively small compared to the complete size of the domain.

Inlet conditions (at $x = 0$) are fixed both in the water level height and in velocity. The velocity is defined by a base vector of $\bar{\mathbf{V}} = [\bar{V}_x, \bar{V}_y, 0]$ and a periodic component to generate waves. The properties of the waves are defined so that the wave transport equation for low amplitudes can be used to compute the wave evolution:

$$A = 0.05m$$

$$\lambda = 1.5m$$

$$k = \frac{2\pi}{\lambda}$$

$$\omega = \sqrt{g \cdot k}$$

Where A is the amplitude and λ is the length of a wave with sinusoidal shape. The k value is the circular wavenumber and g is the gravity in the z direction. Setting the mean free surface level as $z = 0$, the parameters at the inlet for a node with coordinates (x, y, z) it is possible to obtain the free surface position and the velocity field by solving the Helmholtz equation and combining the resulting velocity field with a traslation caused by the continuous speed of the domain relative to a point at infinite depth. For a continuous field of sinusoidal waves advancing with $x+$, the inlet conditons become:

$$\varphi = z - A \cos(k(x - \bar{V}_x t) - \omega t)$$

$$V_x = \bar{V}_x + A \omega \exp(k z) \cos(k (x - \bar{V}_x t) - \omega t)$$

$$V_z = A \omega \exp(k z) \sin(k (x - \bar{V}_x t) - \omega t)$$

The problem was solved for a boat velocity of $\bar{V}_x = 10m/s$ and a periodic rigid body rotation of the boat from 0° to 3.5° . To avoid excessively large Courant numbers, the time step was set to $\Delta t = 0.005$. The time domain extends to $t = [0, 4.0]$. To solve the system in the deformed configuration, before each step the displacements in the boat are imposed and then the Poisson problem is solved to obtain the updated nodal positions. After this, the PFEM-2 Solver is executed and the equations are solved. It must be noted that the particles do not change their position when the mesh is deformed.

In Figure 3.33 two snapshots at time $t = 0.5s$ are shown. The free surface is plotted by creating the isosurface $\varphi = 0$ using the nodal values. On the other hand, to plot the water droplets, only the particles above the free surface are printed. As it can be seen in the figure, the spray produced by the impact of the hull against the waves does not reach much higher than the hull itself, showing that for the analysed speed of $10m/s$ there would be no visibility issues in the vehicle according to this simulation.

Although no comparison against experimental or numerical results has been performed for this test case, it shows the potential to handle large cases from a different approach. Using the PFEM-2 Particles allows to capture details of the simulation that would require extremely finer meshes with other strategies, such as level set or the volume of fluid. Computation times were also acceptable given the mesh size, demanding around 120 seconds per time step including the mesh deformation stage and the regeneration of the mesh search algorithm at each time step due to the new geometry.

3.6 Summary and Conclusions

In this chapter the general tools of the PFEM-2 method shown in Chapter 2 were employed to develop a Navier-Stokes solver. Two distinct strategies were presented for the solution of the NS equations using the PFEM. The objective behind this is having multiple alternatives to tackle a problem depending on each particular case. This allows to minimize the computational resources required while, at the same time, ensuring that the hypothesis of the model do not affect severely the accuracy of the solution.

In the first section the Navier-Stokes equations were introduced. The independence of the shear stresses from the past history of the deformation makes it possible to develop

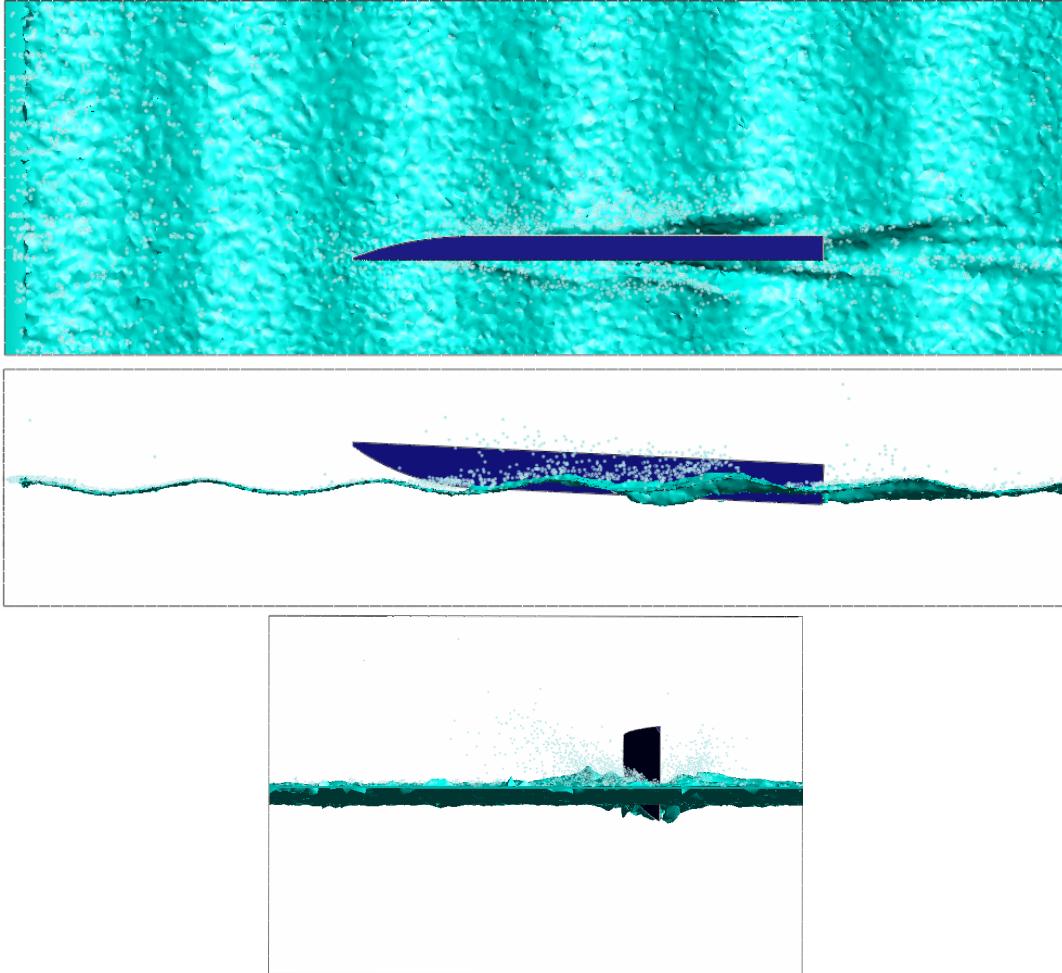


Figure 3.33: Boat Simulation snapshots at $t=0.5$ s. Top, side and front views. Free surface and water particles in cyan.

simple constitutive models, easily uncoupling the volumetric and deviatoric stresses. Finally, the time discretized set of Lagrangian equations were presented, which represent the set of equations that must be solved at each time step. To tackle the solution of this problem, a fractional step (FS) and a monolithic strategy were developed.

The FS implementation was created with the goal of solving the N-S equations as fast as possible. The strategy is based on a first order pressure-velocity segregation scheme, improving this way the conditioning of the matrices. The resulting algorithm is noticeably faster than non-segregated schemes, requiring approximately half the time to solve the linear set of equations. Test examples were presented to test the accuracy, as well as a comparison against a well established fluid solver to test the code speed. From the

results of these simulations, it can be concluded that the implemented FS strategy offers a fast yet accurate alternative to solve the Navier-Stokes equations.

The segregated strategy was also enhanced with a modified pressure field at the interface. To raise the limitations of standard finite elements in problems with sharp interfaces, an enrichment pressure degree of freedom was added to the elements split by the interfaces. Special attention was paid to boundary terms since they are not guaranteed to be in equilibrium across interfaces. In this sense, a modified integration rule was employed to improve the convergence rate. With this modification, the addition of this DoF allows to capture the discontinuity in the pressure gradient that is caused by the density jump, thus leading to the exact solution in hydrostatic cases. The importance of adding this enrichment for multi fluid problems has been assessed in more complex examples by Coppola (2009), comparing the results of enriched and non enriched finite elements formulations.

Despite the advantages of the fractional step solver in terms of the demanded computational time to solve a given problem, the hypothesis used on the derivation bound the range of problems that it can tackle accurately. The decomposition of the velocity into a fractional(non divergence-free) and a final (divergence-free) velocity required certain assumptions that do not hold when the viscosities of the fluids are high. Although modifications to enhance the behaviour for mid-viscosities had been performed, the application of this strategy is still not suited for very-high viscosity problems. In this sense it must be noted that, despite not presented here, segregated schemes that can tackle high viscosity problems have been developed in the scientific literature. However their use requires that the momentum and pressure equations must be solved several times until convergence is achieved, increasing considerably the computation time.

The approach of this work for high viscosity problems is solving monolithically the system of equations . Since the equations are linear, a single iteration is sufficient to achieve the solution. The implementation of this strategy makes use of the same tools to solve the convective transport and location of the interfaces used in the FS strategy, meaning that the solver remains unchanged except for the solution in the FE mesh. Since this strategy is capable of dealing with high viscosity fluids, further enrichments were added at the interface. On one hand, the possibility to solve for discontinuous pressure was added, case that arises when the viscosity of one phase is much higher than the other. Also enrichments in the velocity field were developed to allow a different velocity

gradient at both sides of the interface. By using the enhanced pressure and velocity fields, it is possible to model more accurately the behaviour at the interface.

Low viscosities test cases were simulated to test a correct implementation of the solver and to compare the computational time. The results showed similar accuracy to the FS implementation, but twice the computational time to solve the linear system. Considering that a high viscosity problem would require at least two iterations of a FS strategy to converge while a monolithic a single one, it follows that the monolithic strategy demands at most the same computational time required by the FS counterpart.

High viscosity cases were used to asses the performance of the enhanced interface description. It is interesting to note that despite the better description of the material, the mesh convergence remains linear like the non-enriched case. This is caused by the condensation procedure of the enrichment velocity DoFs, which leads to incorrectly set boundary terms that inhibit a better convergence slope. Still, even if the slope is not improved, the modification of the velocity field proved to be at least as good or even better than the non-enriched implementation, depending on the problem.

As a conclusion, the two implementations developed in this chapter showed good accuracy and adequate performance. While the FS solver is specially suited for cases of low viscosity in which a key goal is reducing the computational cost, the monolithic strategy is better suited for a broader field of application. This flexibility translates into a higher computational cost, requiring twice the time to solve the system. Having these concepts in mind, the decision of choosing one or the other strategy will depend on the problem of interest or the trade off between accuracy and computer time.

Chapter 4

Solution strategies for solids and FSI problems

4.1 Introduction

Simulating correctly the solid phase is a vital part for any FSI algorithm. Unlike fluids, the material response of solids bodies is more complex due to the fact that they can store deformation energy. This implies that the material response does not only depend on the current deformation rate but on all the history of deformation. Since the total deformation of solids bodies is bounded, most methods employ pure Lagrangian strategies to tackle the simulation of solids, even for very large deformations (for solids standards). Currently, Lagrangian Finite Elements represent the most widely used spatial discretization for solids. Extensive literature can be found, from the fundamental mathematical background to specific developments for a wide range of applications.

This chapter covers the implementation of the PFEM-2 solver for solids and FSI problems. The main features of the PFEM-2 method for multi-material problems remain unchanged, only altering the constitutive model of the material and therefore increasing the complexity on the required steps to solve the momentum equations. To allow for an easy coupling between the solid and fluid phases, the same variables will be used in the strategy, namely pressure and velocity. The use of this mixed formulation, as it will be seen later, leads to a splitting in the stresses into its deviatoric and volumetric

components. Since the pressure is obtained directly from the solution of the system, the resulting strategy is more accurate than irreducible counterparts (displacements only) but with a higher complexity system (Cervera et al., 2010).

The current chapter is structured as follows. In the first section, the continuum problem is presented for a generic solid material. Next, the constitutive models for linear and non-linear material responses are explained. Finally, the spatial and temporal discretization is explained together with examples to benchmark the strategy.

4.2 Solids Formulation

In order to simulate solids using the same solving procedure described in the previous chapters, a Lagrangian framework must be used. The algorithm employed in this work follows the strategy developed by Idelsohn et al. (2008) for a pure Lagrangian method, but omitting here the mesh deformation due to the use of particles. In this model the stresses are updated to the new framework at each step and solved in this configuration to obtain the final state of the system. (Updated Lagrangian).

4.2.1 Continuum model

Writing the continuum mechanics problem in a Lagrangian Framework, the momentum and mass conservation equations reads (4.1), where $\boldsymbol{\sigma}$ is the Cauchy stress tensor.

$$\begin{aligned} \rho \left(\frac{D(\mathbf{V})}{Dt} \right) &= \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} \\ \frac{D\rho}{Dt} + \nabla \cdot (\rho \mathbf{V}) &= 0 \end{aligned} \quad (4.1)$$

The mass conservation equation can be replaced by a formulae to relate changes in the volume with changes in the pressure. Using the compressibility modulus κ :

$$\begin{aligned} \rho \left(\frac{D(\mathbf{V})}{Dt} \right) &= \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} \\ \frac{Dp}{Dt} + \kappa \nabla \cdot \mathbf{V} &= 0 \end{aligned} \quad (4.2)$$

This set of equations must be fulfilled at each time step to advance in time. It must be noted that, while the acceleration terms are written in terms of the velocity, the material response will depend on the current deformation and its history.

Stress response

The material stress response can be defined as the increase (or decrease) of the stress state for a given change in the strains. These strains are caused by the deformation experimented by the solid bodies, which can be computed from the displacement at the macroscopic scale. In this sense, it must be noted that the response of the material can be highly non linear and even experiment a reduction of the forces in some cases. This nonlinear behavior means that, unlike the fluid problems analyzed in the previous chapter, the stress response cannot be obtained directly in a single step; it must be tracked carefully to guarantee that the material behavior is modeled correctly.

The total deformation can be split into two distinct parts: the elastic deformation and the plastic deformation. The elastic stress response of a material is characterized by a reversible behaviour. For a given deformation, the elastic stresses will be a function solely on this deformation stress and all the input work is stored as strain energy; a solid in elastic regime will always return to the original shape with the same stress state. On the other hand, the plastic deformation is not reversible. The input mechanical work is not stored as strain energy but rather it is dissipated as heat. This means that the plastic behaviour is equivalent to the viscosity in fluids. In Figure 4.1 the stress response of an elastic-plastic model is plotted. For deformations below the elastic limit σ_0 , the behaviour is reversible. Further increasing the strains does not raise the stress response anymore, converting all input energy into plastic deformations. If later the strains are reduced, the material will recover the original elastic response. In this work only a perfect plastic model will be implemented to represent the response of solids due to its simplicity and faster convergence rate compared to other, more complex, models. Despite it is certainly not the most advanced model to represent granular materials, other authors have used it to model the plastic behaviour, obtaining good results against experimental data (Bui et al., 2008).

In the following sections the employed elastic and plastic models are described. The stress state of the material can be represented as a combination of an elastic and plas-

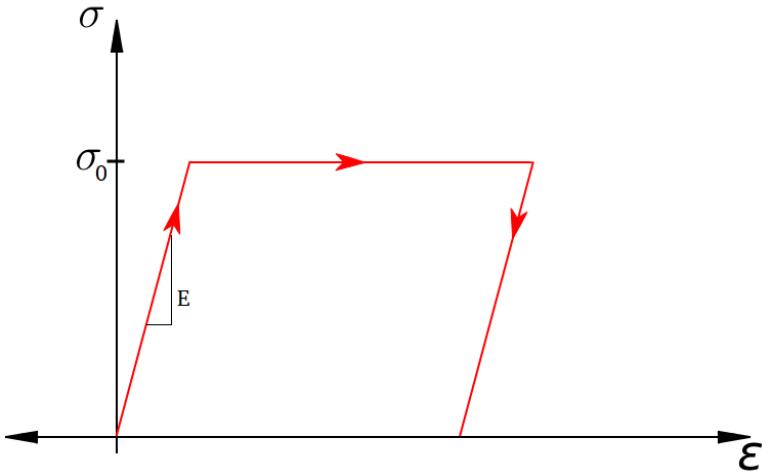


Figure 4.1: Perfect Plasticity material response

tic response; first the reversible part of the deformation will be described -the elastic predictor-, followed by the plastic corrector.

4.2.2 Elastic Response: Hypoelastic model

The hypoelastic model provides an good constitutive model for solids due to its simplicity and direct application for velocity formulations. Hypoelastic materials are those whose stress rate can be defined in the form:

$$\dot{\sigma} = f(\boldsymbol{\sigma}, \mathbf{d}) \quad (4.3)$$

where

$$d_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)$$

The constitutive law of hypoelastic materials returns the rate of stress rather than the stress itself. This is particularly useful in the velocity formulation since the rate stress function f depends on the rate of deformation \mathbf{d} (related to the velocity) instead of the deformation. The problem lies in finding a rate stress measure that is objective, that is, it must hold frame invariance. Using the Truesdell stress rate $\boldsymbol{\sigma}^\nabla$, the elastic stress

rate becomes:

$$\boldsymbol{\sigma}^{\nabla} = f(\boldsymbol{\sigma}, \mathbf{d}) \quad (4.4)$$

The Truesdell rate is related to the PK2 Stress \mathbf{S} and provides the exact stress rate for a given deformation.

$$\boldsymbol{\sigma}^{\nabla} = J^{-1} \mathbf{F}^{(n+1)} \dot{\mathbf{S}}^{(n+1)} \mathbf{F}^{T(n+1)} \quad (4.5)$$

Using the tensor of elastic moduli for the Truesdell stress rate \mathbf{C}^T :

$$\boldsymbol{\sigma}^{\nabla} = \mathbf{C}^T : \mathbf{d} \quad (4.6)$$

$$\boldsymbol{\sigma}^{\nabla} = (\lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbf{I}) : \mathbf{d} - \mathbf{d} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \mathbf{d} + (\mathbf{I} \otimes \mathbf{d}) \boldsymbol{\sigma} \quad (4.7)$$

Where \mathbf{I} is the fourth order identity tensor and λ and μ are the Lamè parameters. As it can be seen, it is not linear since C depends on σ . To override this inconvenience, the Jaumann (Belytschko et al., 2013) stress rate is used, which simplifies the stress rate into a deformation $\boldsymbol{\sigma}^{\nabla J}$ plus a rotation, avoiding all the non-symmetric terms. Note that this model is only valid for isotropic materials.

$$\boldsymbol{\sigma}^{\nabla J} = (\lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbf{I}) : \mathbf{d} \quad (4.8)$$

Using the approximation of the stress rate (4.9) instead of the exact Truesdell stress, Equation (4.5) can be written as:

$$J^{(n+1)} (\lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbf{I}) : \mathbf{d}^{n+1} = \mathbf{F}^{(n+1)} \dot{\mathbf{S}}^{(n+1)} \mathbf{F}^{T(n+1)} \quad (4.9)$$

using finite differences in time to obtain the PK2 stress rate $\dot{\mathbf{S}}$

$$J^{(n+1)} (\lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbf{I}) : \mathbf{d}^{n+1} = \mathbf{F}^{(n+1)} \frac{\mathbf{S}^{(n+1)} - \mathbf{S}^{(n)}}{\Delta t} \mathbf{F}^{T(n+1)} \quad (4.10)$$

$$\mathbf{F}^{(n+1)} \frac{\mathbf{S}^{(n+1)}}{\Delta t} \mathbf{F}^{T(n+1)} = J^{(n+1)} (\lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbf{I}) : \mathbf{d}^{n+1} + \mathbf{F}^{(n+1)} \frac{\mathbf{S}^{(n)}}{\Delta t} \mathbf{F}^{T(n+1)} \quad (4.11)$$

Recalling the relation between the second Piola-Kirchoff stress and the Cauchy stress $J\boldsymbol{\sigma} = \mathbf{F}\mathbf{S}\mathbf{F}^T$, the LHS becomes:

$$J^{(n+1)} \frac{\boldsymbol{\sigma}^{(n+1)}}{\Delta t} = J^{(n+1)} (\lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbf{I}) : \mathbf{d}^{n+1} + \mathbf{F}^{(n+1)} \frac{\mathbf{S}^{(n)}}{\Delta t} \mathbf{F}^{T(n+1)} \quad (4.12)$$

Now, resetting the reference framework at each time step as the last known configuration $\mathbf{x}^{(n+1)} := \mathbf{X}^{(n)}$, the Cauchy stress of the previous time step directly becomes the PK2 stress $\mathbf{S}^{(n)} := \boldsymbol{\sigma}^{(n)}$. Finally the stress in the new configuration becomes (4.14), where the subindex $n+1$ implies that the old stresses have been updated to the new configuration.

$$\boldsymbol{\sigma}^{(n+1)} = J^{-1(n+1)} \mathbf{F}^{(n+1)} \boldsymbol{\sigma}^{(n)} \mathbf{F}^{T(n+1)} + \Delta t (\lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbf{I}) : \mathbf{d}^{n+1} \quad (4.13)$$

$$\boldsymbol{\sigma}^{(n+1)} = \boldsymbol{\sigma}_{n+1}^{(n)} + \Delta t (\lambda \operatorname{tr}(\mathbf{d}^{n+1}) \mathbf{I} + 2\mu \mathbf{d}^{n+1}) \quad (4.14)$$

To mimic the formulation of velocities and pressure used in fluids, it is useful to decompose the stresses into the pressure $p = -\frac{1}{3} \operatorname{tr}(\boldsymbol{\sigma})$ caused by the volumetric deformation $\epsilon_v = \frac{1}{3} \operatorname{tr}(\mathbf{d})$ and deviatoric part $\boldsymbol{\sigma}'$ caused by the deviatoric strain $\mathbf{d}' = \mathbf{d} - \mathbf{I}\epsilon_v$.

$$\boldsymbol{\sigma}^{(n+1)} = \boldsymbol{\sigma}_{n+1}^{(n)} + \Delta t [\lambda \operatorname{tr}(\mathbf{d}^{n+1}) \mathbf{I} + 2\mu (\mathbf{d}'^{n+1} + \mathbf{I}\epsilon_v)] \quad (4.15)$$

$$\boldsymbol{\sigma}^{(n+1)} = \boldsymbol{\sigma}_{n+1}^{(n)} + \Delta t \left[(\lambda + \frac{2}{3}\mu) \operatorname{tr}(\mathbf{d}^{n+1}) \mathbf{I} + 2\mu (\mathbf{d}'^{n+1}) \right] \quad (4.16)$$

$$\boldsymbol{\sigma}^{(n+1)} = \boldsymbol{\sigma}_{n+1}^{(n)} - \Delta p \mathbf{I} + \Delta \boldsymbol{\sigma}' \quad (4.17)$$

where

$$\Delta p = \Delta t (\lambda + \frac{2}{3}\mu) \operatorname{tr}(\mathbf{d}^{n+1}) \quad \text{and} \quad \Delta \boldsymbol{\sigma}' = 2\Delta t \mu (\mathbf{d}'^{n+1}) \quad (4.18)$$

Equation (4.17) updates the old stresses to the new configuration plus the delta stress due to the deformation of the current time step.

4.2.3 Drucker Prager yield criterion for granular materials

The hypo-elastic model described in the previous section provides the algorithm to obtain stresses for a given deformation of a solid with a linear response. However it does not proportionate a yield surface or any other criteria to determine if material points are still in the elastic regime or have undergone inelastic strains. The yield surface is a function in the stress map that inscribes all the possible stress states under elastic behaviour, so that deformations will be reversible. On the other hand, if the material is already on the yield surface and the deformation is further increased, plastic deformation will occur.

It must be noted that the yield surface is not fixed but can be expanded (for plastic hardening materials) or decrease its size with increasing deformation (softening). Still, at all times the stress state must be within the volume inscribed by the yield surface. Failing to do so implies that the material is not able to withstand the imposed load.

Yield Surface

The Drucker Prager (DP) model provides the additional data required to determine whether a granular material is inside the yield surface or has exceeded the admissible stresses. For the particular case of granular materials with cohesion, the DP model yield surface can be expressed as:

$$\sigma_v = A + Bp \quad (4.19)$$

where σ_v is the equivalent tensile stress or Von Misses stress and A and B are parameters that depend on the material. While it is not a true stress, it is a useful tool linked to the deviatoric strain energy. σ_v allows to transform any state of stress to a single scalar measurement, therefore making it possible to use data from simple experimental tests such as uniaxial tension. Writing the j component in the i direction of the deviatoric stress tensor as σ'_{ij} , σ_v can be expressed as:

$$\sigma_v = \sqrt{\frac{2}{3}\sigma'_{ij}\sigma'_{ji}} \quad (4.20)$$

For the particular case of granular materials with cohesion, the expressions for A and B are as follows, where ϕ is the friction angle and c is the cohesion of the material:

$$A = c \quad B = \tan(\phi) \quad (4.21)$$

As it can be seen in equation 4.19, the yield surface is described by a linear function. Figure 4.2 shows the influence of A and B on the line. The admissible stresses are the ones below the yield surface.

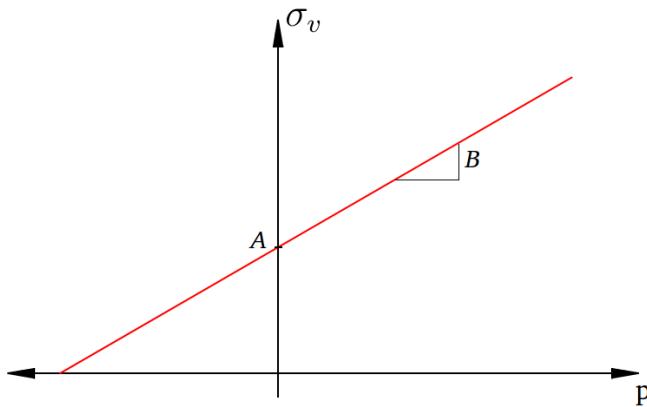


Figure 4.2: Drucker Prager yield surface

Return Mapping

Using Equation (4.17), the hypoelastic model provides an elastic estimation of the stresses called the *elastic predictor* σ^{tr} . In case σ^{tr} was outside the yield surface, it would necessary to provide a path to return the stress state inside the admissible stresses (Figure 4.3). The path must be such that the distance in the plane $p - \sigma_v$ is minimized.

Since the yield surface is described by a line, it is possible to find analytically the closest point that is inside the surface. This path to follow must be perpendicular to the slope of the yield surface. As shown in Figure 4.4, the final state of stress (p^f, σ_v^f) can be found in the intersection between the yield surface and the line defining the return map, which has the slope $-1/B$ and contains the point (p^{tr}, σ_v^{tr}) . Calling A^{tr} the intersection between the blue line and $p = 0$ axis, the set of equations to be solved is:

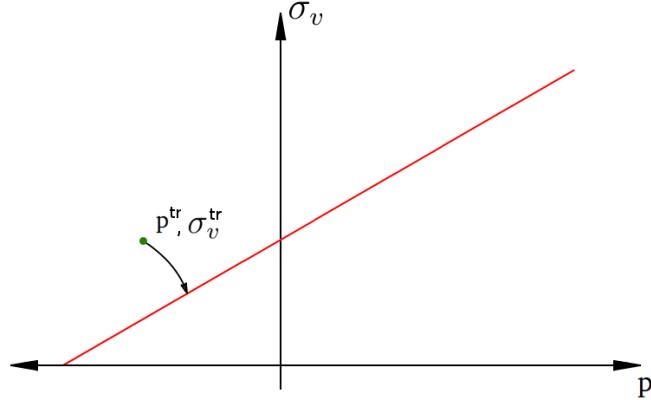


Figure 4.3: Return Mapping

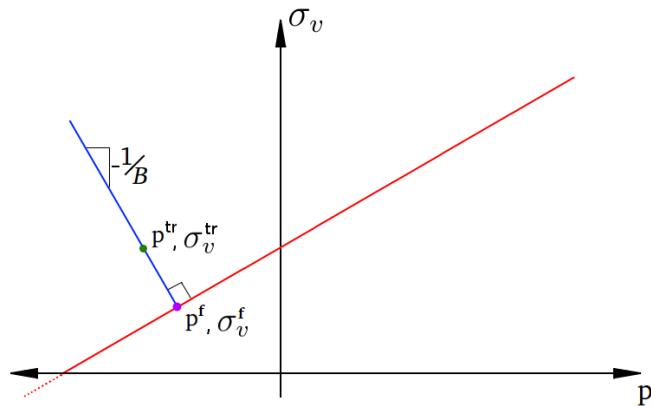


Figure 4.4: Return Mapping slope

$$\begin{cases} -\frac{1}{B} p^{\text{tr}} + A^{\text{tr}} = \sigma_v^{\text{tr}} \\ -\frac{1}{B} p^f + A^{\text{tr}} = A + B p^f \\ A + B p^f = \sigma_v^f \end{cases} \quad (4.22)$$

Solving the system of equations, the return algorithm becomes:

$$\begin{aligned} p^f &= \frac{\sigma_v^{\text{tr}} + \frac{1}{B} p^{\text{tr}} - A}{B + \frac{1}{B}} \\ \sigma_v^f &= A + B p^f \end{aligned} \quad (4.23)$$

It must be noted that the solution (4.23) is mathematically valid for both positive and negative values of σ_v . However negative values (dotted line in Figure 4.4) have no

physical meaning and therefore have to be avoided. In this case, the return map shall go directly to the vertex C of the yield surface (minimum admissible pressure). The two zones are illustrated in Figure 4.5. Finally, the pressure of each material point is updated with (4.23), while the deviatoric tensor σ' is scaled down in order obtain the computed σ_v^f .

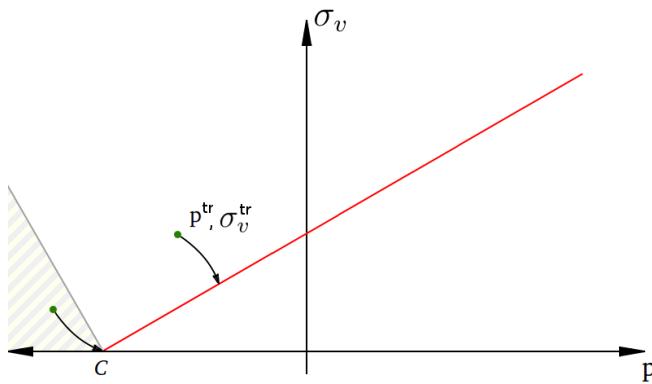


Figure 4.5: Return Zones

Most approaches to debris flow simulation do not simulate the soil beneath the debris flow. This mean that the location of the bed surface is imposed as boundary conditions and as the soil is eroded with empirical laws, this boundary conditions are modified.

On the other hand, simulating the soil bed with an appropriate solid formulation does not only eliminate the need to modify the boundary conditions, but also translates into more accurate estimation of the erosion and deposition rates. Since the soil stresses are computed accurately, the entrained mass is directly the portions of soil that have failed and therefore no empirical formulations of erosion rates are needed.

4.2.4 Plastic Stress Update algorithm

In the previous sections the spatial discretization and the constitutive law for a granular material were described. Since the plasticity models implies a change in the stiffness of the material, the stress response is no longer linear and therefore an iterative procedure is necessary to achieve convergence.

The stiffness matrix of the system for the elastic case provides the exact material response for small deformations. In Figure 4.6 the stress increase from one time to the

next is shown. Since the slope of the stress-strain relation does not change, this matrix can be seen as either a tangent or secant operator due to the constant slope. This linear response also means that the stress update in the material points will compute the same stresses obtained in the solution of the linear system. In other words, it converges in a single iteration.

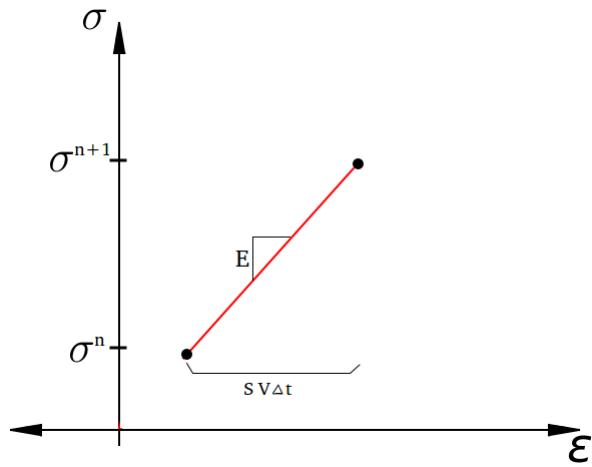


Figure 4.6: Elastic stress update

On the other hand, when the stress response is not linear, as it is in plasticity problems, an iterative procedure is required in order to reach the solution. There are two different approaches to obtain the solution: The secant modulus, shown in the left of Figure 4.7, and the tangent operator the left.

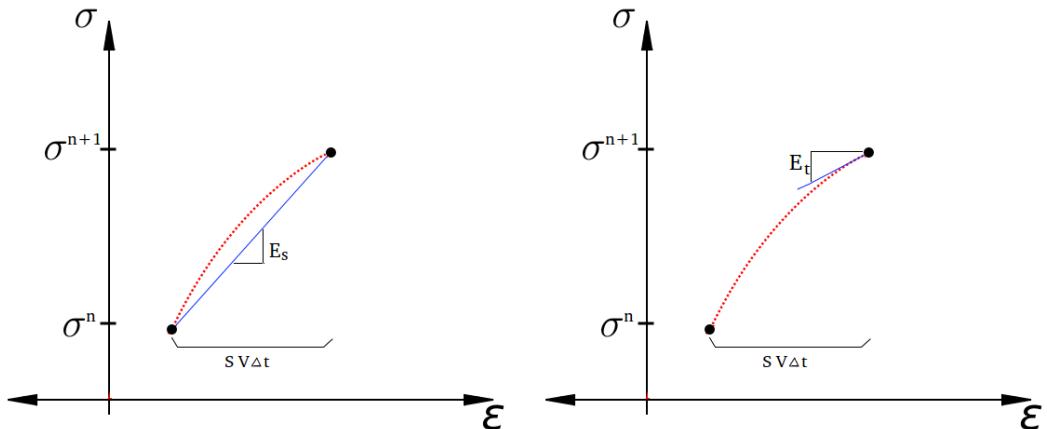


Figure 4.7: Elastic stress update

The secant modulus consists on setting an apparent stiffness so that the displacement multiplied by the modified modulus will return the new stress. Since the exact location of this point in the $\epsilon - \sigma$ plane is not known, in each iteration the secant modulus will be changed until convergence is achieved. The use of this procedure implies that at each iteration the stresses are reset to the last known state of the previous time step.

On the other hand, the tangent operator consists on using the real tangent stiffness of the material at the suspected point where the solution is located at. This means that, unlike in the secant operator, the stresses obtained in the last iteration are used to begin the new iteration. The advantage of the tangent modulus is that it has the potential of higher convergence rates.

Brief introduction to the theory of plasticity

A more convenient form to write the failure criteria is defining a scalar yield function f . The material will be considered on the elastic limit when $f = 0$ and negative for stresses in the admissible elastic region. Positive values have no physical meaning. Recalling the DP criteria (4.19), the yield function becomes:

$$f(\boldsymbol{\sigma}) = \sigma_v - A - B p \quad (4.24)$$

Next, the total deformations can be split into two parts, namely the elastic and the plastic deformation:

$$\mathbf{d} = \mathbf{d}_{el} + \mathbf{d}_{pl} \quad (4.25)$$

Inserting Equation (4.25) into the elastic stress rate equation (4.6):

$$\boldsymbol{\sigma}^\nabla = \mathbf{C}_{el}(\mathbf{d} - \mathbf{d}_{pl}) \quad (4.26)$$

The plastic deformation \mathbf{d}_{pl} can be defined by a modulus, the plastic multiplier λ_{pl} and a direction, a flow rule. The simplest approach to set the flow rule is following the direction of the yield function:

$$\mathbf{d}_{pl} = \dot{\lambda}_{pl} \frac{df(\boldsymbol{\sigma})}{d\boldsymbol{\sigma}} \quad (4.27)$$

Inserting (4.27) into (4.26):

$$\boldsymbol{\sigma}^\nabla = \mathbf{C}_{el} : (\mathbf{d} - \dot{\lambda}_{pl} \frac{df(\boldsymbol{\sigma})}{d\boldsymbol{\sigma}}) \quad (4.28)$$

Now, for the plastic loading condition, the stress state must remain at all times on the yield surface, which translates into $f(\boldsymbol{\sigma}) = 0$. Due to its constant value, its derivative must also remain null and therefore:

$$\dot{f}(\boldsymbol{\sigma}) = \frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} = 0 \quad (4.29)$$

Taking $\dot{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^\nabla$ and inserting (4.29) into (4.28)

$$\frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}} : \mathbf{C}_{el} : (\mathbf{d} - \dot{\lambda}_{pl} \frac{df(\boldsymbol{\sigma})}{d\boldsymbol{\sigma}}) = 0 \quad (4.30)$$

And finally the plastic multiplier becomes:

$$\dot{\lambda}_{pl} = \frac{\frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}} : \mathbf{C}_{el} : \mathbf{d}}{\frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}} : \mathbf{C}_{el} : \frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}}} \quad (4.31)$$

Elastoplastic tangent operator

From the previous definitions it is possible to create a modified stiffness matrix. This way it is possible to have the total deformations as an input parameter instead of splitting it in two, simplifying the solution procedure. Inserting (4.31) into (4.28):

$$\dot{\boldsymbol{\sigma}} = \mathbf{C}_{elpl} : \mathbf{d} = \left(\mathbf{C}_{el} - \frac{(\mathbf{C}_{el} : \frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}}) \otimes (\mathbf{C}_{el} : \frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}})}{\frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}} : \mathbf{C}_{el} : \frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}}} \right) : \mathbf{d} \quad (4.32)$$

Recalling that $p = -1/3\text{tr}(\boldsymbol{\sigma})$, the derivative $\frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}}$ becomes:

$$\frac{d(f(\boldsymbol{\sigma}))}{d\boldsymbol{\sigma}} = \sqrt{\frac{3}{2}} \frac{\boldsymbol{\sigma}'}{\|\boldsymbol{\sigma}'\|} + \frac{\tan(\phi)}{3} \mathbf{I} \quad (4.33)$$

Splitting the elastoplastic tangent operator \mathbf{C}_{elpl} into its deviatoric and volumetric components, the deviatoric tangent operators is obtained. The effect of the change in the plastic multiplier has been omitted for simplicity. (For a more accurate computation of \mathbf{C}_{elpl} , the work of Benedetti et al. (2015) presents the complete operator, including the response for softening and hardening responses)

$$\mathbf{C}'_{elpl} = \mathbf{C}'_{el} - \frac{(2G\sqrt{\frac{2}{3}}\mathbf{n}) \otimes (2G\sqrt{\frac{2}{3}}\mathbf{n})}{\mathcal{D}} \quad (4.34)$$

Where \mathcal{D} and \mathbf{n} are:

$$\mathcal{D} = (\tan(\phi))^2 k + 3G \quad \mathbf{n} = \frac{\boldsymbol{\sigma}'}{\|\boldsymbol{\sigma}'\|}$$

Computation of the plastic flow

Using an associative plastic flow rule, the direction of the plastic deformation can be readily determined by the stress state and the yield surface, as noted in expression (4.27). Using a discrete time increment from step n to $n + 1$, the elastic deformation can be expressed as:

$$\Delta\boldsymbol{\epsilon}_{pl}^{n+1} = \frac{f(\boldsymbol{\sigma}_{trial})}{\mathcal{D}} \left(\sqrt{\frac{3}{2}} \frac{\boldsymbol{\sigma}'}{\|\boldsymbol{\sigma}'\|} + \frac{\tan(\phi)}{3} \mathbf{I} \right) \quad (4.35)$$

Having the expressions for the tangent operators and the plastic flow, it is now possible to write the complete algorithm to solve advance in time from step n to $n + 1$.

4.2.5 Spatial discretization and weak form

The spatial discretization and variables will be the same as the one used in the fluids formulation: linear elements for both the pressure and the velocity. As for the deviatoric stress it will be a secondary variable depending on the unknowns.

Recalling the system of equation for solids (4.2) and splitting the stresses into its two components, deviatoric and volumetric

$$\begin{aligned} \rho \left(\frac{D(\mathbf{V})}{Dt} \right) &= \nabla (\sigma' - p \mathbf{I}) + \rho g \\ \frac{Dp}{Dt} + \kappa \nabla \cdot \mathbf{V} &= 0 \end{aligned} \quad (4.36)$$

Writing the weak form of the problem:

$$\begin{aligned} \left(\mathbf{w}, \rho \frac{D(\mathbf{V})}{Dt} \right) &= (\mathbf{w}, \nabla (\sigma' - p \mathbf{I}) + \rho g) \\ \left(q, \frac{Dp}{Dt} + \kappa \nabla \cdot \mathbf{V} \right) &= 0 \end{aligned} \quad (4.37)$$

Integrating by parts the gradient of the stresses in order to avoid second derivatives on the velocities and to convert the gradient of the pressure into the transpose of the divergence, the system becomes:

$$\begin{aligned} \left(\mathbf{w}, \rho \frac{D(\mathbf{V})}{Dt} \right) &= -(\nabla \mathbf{w}, (\sigma' - p \mathbf{I})) + (\mathbf{w}, \rho \mathbf{g}) \\ \left(q, \frac{Dp}{Dt} + \kappa \nabla \cdot \mathbf{V} \right) &= 0 \end{aligned} \quad (4.38)$$

The boundary terms $\int_{\Gamma} \mathbf{n} \sigma d\Gamma$ that appear after integrating by parts are omitted. This implies that contributions from both sides must cancel each other, therefore ensuring stress continuity.

A first approximation in time for the velocity would lead to an stable but too diffusive algorithm. To avoid this shortcoming, the Newmark's Beta method was selected. Parameters were set for constant acceleration. The resulting expression for the velocity

becomes:

$$2 \frac{\mathbf{V}^{n+1}}{\Delta t} = 2 \frac{\mathbf{V}^n}{\Delta t} - \dot{\mathbf{V}}^n$$

Now using finite differences in time for the pressure, dividing the second equation by the compressibility modulus κ and recalling the definition of the stresses (4.17).

$$\begin{aligned} \left(\mathbf{w}, \rho \frac{\mathbf{V}^{n+1}}{\Delta t} \right) - (\nabla \mathbf{w}, (\Delta \sigma' - \Delta p \mathbf{I})) &= \left(\mathbf{w}, \frac{\mathbf{V}^n}{\Delta t} + \rho g \right) - (\nabla \mathbf{w}, (\Delta \sigma'^n_{n+1} - (p^n) \mathbf{I})) \\ \left(q, \frac{p^{n+1}}{\kappa \Delta t} + \nabla \cdot \mathbf{V} \right) &= \left(q, \frac{p^n}{\kappa \Delta t} \right) \end{aligned} \quad (4.39)$$

To account for the increase in the deviatoric stresses $\Delta \sigma'$, an implicit scheme is used to avoid instabilities:

$$\int_n^{n+1} \mathbf{d} dt = \mathbf{S} \mathbf{V}^{n+1} \Delta t \quad (4.40)$$

where \mathbf{S} is the standard strain-rate velocity matrix

$$\mathbf{S} = \begin{bmatrix} \frac{\partial N}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N}{\partial z} \\ \frac{\partial N}{\partial y} & \frac{\partial N}{\partial x} & 0 \\ 0 & \frac{\partial N}{\partial z} & \frac{\partial N}{\partial y} \\ \frac{\partial N}{\partial z} & 0 & \frac{\partial N}{\partial x} \end{bmatrix}$$

Finally, writing the equations in matrix form, the system can be expressed as:

$$\begin{bmatrix} \left(\frac{2}{\Delta t} \mathbf{M}(\rho) + \Delta t \mathbf{K} \right) & \mathbf{D}^T \\ \mathbf{D} & \frac{1}{\Delta t} \mathbf{M}(\frac{1}{\kappa}) \end{bmatrix} \begin{bmatrix} \mathbf{V}^{n+1} \\ \mathbf{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\rho) \left(\frac{2}{\Delta t} \mathbf{V}^n - \dot{\mathbf{V}}^n + \mathbf{g} \right) + \mathbf{D} \sigma'^n_{n+1} \\ \frac{1}{\Delta t} \mathbf{M}(\frac{1}{\kappa}) \mathbf{p}^n \end{bmatrix} \quad (4.41)$$

In general all the matrix have the same shape as in fluid, except for the stiffness matrix \mathbf{K} . Since it has to account for deviatoric stresses only, it is constructed as

$$\mathbf{K} = \int_{\Omega} \mathbf{S} \mathbf{C}' \mathbf{S} \quad (4.42)$$

where \mathbf{C}' is the deviatoric constitutive matrix. It must be noted that even for 2D simulations, the coefficients in the \mathbf{C}' matrix do not change. The only modification is the elimination of the strains in the third dimension, meaning that a plain strain state is considered. This matrix is obtained by eliminating the volumetric deformation using $\mathbf{m} = [1, 1, 1, 0, 0, 0]^T$. Then it follows $\mathbf{C}'_{el} = \mu (\mathbf{I}_0 - \frac{1}{3}\mathbf{mm}^T)$. For the elastic problem, this yields

$$\mathbf{I}_0 = \begin{bmatrix} 2 & & & 0 \\ & 2 & & \\ & & 2 & \\ & & & 1 \\ & & & & 1 \\ 0 & & & & & 1 \end{bmatrix}; \quad \mathbf{C}'_{el} = \mu \begin{bmatrix} 4/3 & -2/3 & -2/3 & 0 & 0 & 0 \\ -2/3 & 4/3 & -2/3 & 0 & 0 & 0 \\ -2/3 & -2/3 & 4/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The system of equations (4.41) can be solved without the use of stabilized formulations as long as the material is not incompressible ($\kappa < \infty$). Otherwise, the pressure mass matrix becomes zero and therefore stabilization is required. However this would not be a complication since the same stabilization of the previous chapter could be used.

Finally an additional hypothesis has to be taken in order to linearise the system. The transformation of the stresses in the previous configuration to the new configuration requires the updated velocity \mathbf{V}^{n+1} , making the system non-linear. The simplification used in this work consists on using the previous step velocity, therefore assuming small accelerations for the solids. This hypothesis is similar to the one used on the convective term, which showed good accuracy in all the cases tested.

Residual-based solution strategy

In the previous chapter, the solution strategy for Newtonian fluids was presented, both single phase and two phase. Due to the linearity of the equations, it was possible to solve the system directly in the form:

$$\mathbf{M} \dot{\mathbf{x}}^{n+1} + \mathbf{K}_{sec} \mathbf{x}^{n+1} = \mathbf{f}_{ext}^{n+1} \quad (4.43)$$

where \mathbf{x} is the unknown vector (velocities and pressure) and \mathbf{f}_{ext} the external force vector. Note that the stiffness/viscosity matrix \mathbf{K}_{sec} must be written in the secant form since it must account for the mean increment in the timestep. The solution of this linear system directly provided the values of the physical variables since the product $\mathbf{K}_{sec} \mathbf{x}^{n+1}$ returns the exact internal forces: $\mathbf{K}_{sec} \mathbf{x}^{n+1} = \mathbf{f}_{int}^{n+1}$.

On the other hand, solids under plastic deformation have a non-linear behaviour, requiring an iterative strategy in order to obtain the solution of the system. In order to achieve convergence, the tangent operator of the previous section will be used combined with a residual-based strategy. Writing the system (4.43) in the residual form yields:

$$\mathbf{M} \dot{\mathbf{x}}^{n+1} - \mathbf{r}^{n+1} = \mathbf{0} \quad ; \quad \mathbf{r}^{n+1} = \mathbf{f}_{ext}^{n+1} - \mathbf{f}_{int}^{n+1} \quad (4.44)$$

where \mathbf{r} is the residual vector. Note that the acceleration is not included in \mathbf{r} , only the forces are considered in this term. In solids, the internal forces are the sum of the previous step stresses \mathbf{f}_{int}^n plus the increment caused by the new deformations resulting from the displacement increment $\mathbf{V}^{n+1} \Delta t$. Using the secant operator \mathbf{K}_{sec} to estimate stress increment, the residual can be written with the following expression, where the superscript $n+i$ and $n+i+1$ indicate the previous and current iterations respectively.

$$\mathbf{r}^{n+1} = \mathbf{f}_{ext}^{n+1} - (\mathbf{f}_{int}^n + \mathbf{K}_{sec} \mathbf{x}^{n+1+i}) \quad (4.45)$$

Now using the tangent operator to use only the increment of the current iteration, the residual can be expressed as:

$$\mathbf{r}^{n+1} = \mathbf{f}_{ext}^{n+1} - [\mathbf{f}_{int}^{n+i} + \mathbf{K}_{tan}(\mathbf{x}^{n+1+i} - \mathbf{x}^{n+i})] \quad (4.46)$$

Now inserting (4.46) into the residual form of the system of equations (4.44), the linear

system to be solved at each iteration is obtained:

$$\mathbf{M} \dot{\mathbf{x}}^{n+1} - \mathbf{f}_{ext}^{n+1} + [\mathbf{f}_{int}^{n+i} + \mathbf{K}_{tan}(\mathbf{x}^{n+1+i} - \mathbf{x}^{n+i})] = \mathbf{0} \quad (4.47)$$

The advantage of the residual-based strategy using the tangent operator is that the internal force vector \mathbf{f}_{int} can be modified at the end of each iteration in order to fulfil exactly the chosen constitutive model.

Having obtained the system to advance in each iteration and the necessary operators, it is now possible to express the complete algorithm to compute each time step. The following table presents the steps to obtain the solution of the non-linear problem, omitting the particle convection. Note that, since the pressure is calculated nodally, the contribution from all the elements around each node is necessary in order to obtain the correct volumetric strain. Failing to do so would cause locking of the system.

Step 0) Set $\mathbf{K} = \mathbf{K}_{el}$ for the first iteration

Step 1) Compute \mathbf{f}_{int} from particles to elements

Step 2) Solve system of equations (4.47)

Step 3) Compute elemental volumetric and deviatoric strains

$$\epsilon_{vol}^{elem} = \nabla \cdot \mathbf{V} \Delta t$$

$$\boldsymbol{\epsilon}_{dev}^{elem} = \mathbf{B} \mathbf{V} \Delta t$$

Step 4) Compute nodal volumetric strains

$$\epsilon_{vol} = \frac{\int_{\Omega} \epsilon_{vol}^{elem} d\Omega}{\int_{\Omega} d\Omega}$$

Step 5) Update Particle stresses elastically

$$p_{particle}^{n+1} = p_{particle}^n - \epsilon_{vol} k_{el}$$

$$\boldsymbol{\sigma}_{particle}'^{n+1} = \boldsymbol{\sigma}_{particle}'^n + \mathbf{C}'_{el} \boldsymbol{\epsilon}$$

Step 6) Check if particles are outside yield surface

Step 7) Use return mapping to correct $p_{particle}^{n+1}$, $\boldsymbol{\sigma}_{particle}'^{n+1}$

Step 8) Compute \mathbf{K}_{elpl} and go back to 1) until convergence

4.3 Coupled solver for solids and fluids

4.3.1 Enrichment degrees of freedom

In Chapter 3 it was shown that the enrichment velocity degrees of freedom provide a better solution for stationary Stokes problems. However, the addition of the improved field to the PFEM-2 algorithm showed a similar accuracy in numerical experiments, compared to the non-enriched counterpart. Based on these results, the velocity field will not be enriched for the solid strategy, thus simplifying the deviatoric stress updating algorithm. This will translate into an overestimated momentum transfer between the phases.

Despite the velocity field can be used without adding enrichment shape functions in certain problems, it is imperative to enrich the pressure field, even for the simplest cases. The reason for this is the strong discontinuity in the material properties, in particular the stiffness (or viscosity). A simple example to illustrate this is a box of a dense solid material surrounded by air. In this case the pressure distribution would take the shape illustrated in Figure 4.8.

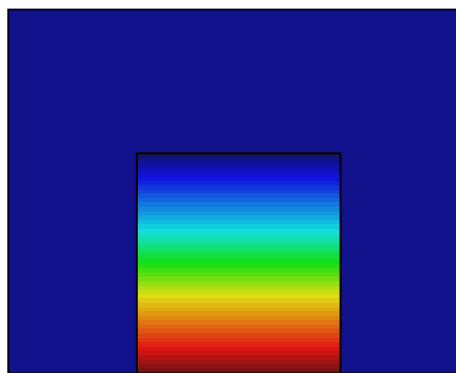


Figure 4.8: Pressure distribution of a solid body surrounded by air under gravitational force.

The fulfillment of the transmission conditions requires that the stresses along the interface must be constant. Neglecting air pressure, this condition is satisfied since $\sigma_{xx} = 0$ at the boundaries of the solid. However the implemented solid model makes use of the mean solid pressure, which means that (taking $\sigma_{zz} = 0$) the pressure in the solid becomes

$$p_{solid} = -\frac{1}{3}\sigma_{yy} \neq p_{air} \quad (4.48)$$

The pressure (4.48) is clearly discontinuous along the interface. If a linear (or any continuous) element was located over the sharp boundary, the discrete pressure field would detect a non-existent gradient that would create normal velocities in the fluid, as seen in 4.9. Since this would violate fluid incompressibility, the solver usually converges to a solution where the last solid node has very low pressure and therefore the structure is more flexible than it should be.

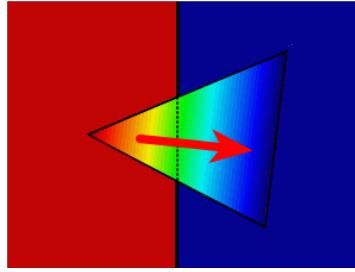


Figure 4.9: Close-up of a linear element located at the interface showing the interpolated pressure field

To avoid this shortcoming, the enrichment shape functions developed for the fluid solver are implemented across the fluid-solid interface. This allows for discontinuous pressures, leading to very accurate results even for coarse meshes as shown in the examples.

On the other hand, it must be noted that stabilizing the pressure equation in the fluid phase is equivalent to adding artificial diffusion to this variable. Therefore, adding stabilization to the interface elements would inevitably lead to an inaccurate behaviour of the solid phase due to dissipation of the stresses in the interface region. For this reason no stabilization is used in these elements. Unfortunately, as stated in Chapter 3, incompressible fluid elements are not stable. To override this problem, both sides of the interface are considered to be slightly compressible (using the compressibility modulus of the solid phase). While this hypothesis is not correct, it leads to small errors only, since the fluid volume is changed only when there is a sharp change in the pressure from one time-step to the next. This strategy yields almost exact results, avoiding pressure diffusion across the interface with minimal errors.

Remark: One of the advantages of this set of discontinuous functions is that the partition of unity is conserved. But most importantly, this also implies that the reconstruction of the pressure p_n after convection is easier to perform accurately. The strategy currently implemented consists on the following steps. First, a standard nodal projection algorithm is used and, therefore, the values of the replacement shape functions are set. Having done this, the enrichment shape functions are defined as the mean of the real nodal values for the previous time-step n . This ensures that the prediction for the gradient in the normal direction is minimized, thus reducing spurious velocities.

4.3.2 Complete algorithm

The complete algorithm is essentially the same for the combined solid-fluid problems. Non-partitioned elements are treated in the way described for either solids or fluids, while the partitioned elements must be enriched across the interface. Finally the complete algorithm can be written as:

Step 1) Convect the particles using the streamlines

Step 2) Project information into the mesh,

with q fluid particles and r solid particles

Common variables

$$\hat{\mathbf{V}}_j^{n+1} = \frac{\sum_i^{r+q} \mathbf{V}_i N_i^j}{\sum_i^{r+q} N_i^j}$$

Solid variables

$$p_j^n = \frac{\sum_i^r p_i N_i^j}{\sum_i^r N_i^j}$$

$$\boldsymbol{\sigma}_{solid}^{rn} = \frac{\sum_i^r \boldsymbol{\sigma}_i^{rn}}{\sum_i^r 1}$$

$$E_{solid} = \frac{\sum_i^r E_i}{\sum_i^r 1}$$

$$\nu_{solid} = \frac{\sum_i^r n u_i}{\sum_i^r 1}$$

$$\rho_{solid} = \frac{\sum_i^r \rho_i}{\sum_i^r 1}$$

Fluid variables

$$\mu_{fluid} = \frac{\sum_i^q \mu_i}{\sum_i^q 1}$$

$$\rho_{fluid} = \frac{\sum_i^q \rho_i}{\sum_i^q 1}$$

- Step 3)** Detect interface elements
- Step 4)** Assemble system of equations with enrichments
- Step 5)** Solve system of equations
- Step 6)** Compute strains
- Step 7)** Update particle stresses
- Step 8)** Go back to 5 until convergence
- Step 9)** Update particles' velocity

4.4 Numerical Examples

4.4.1 Elastic problems

Static cantilever

The first experiment to test the accuracy of the scheme is a cantilever under small deformations. Since material points do not move considerably (*displ << elem size*) it is not necessary to convect properties and therefore particles are only used to store deviatoric stresses. This ensures that the dissipation caused by projecting is eliminated, allowing to test the solids formulation without other possible sources of errors. Also inertial terms are neglected to test the statical solution. The geometry can be seen in Figure 4.10. The material properties are:

$$\rho = 1 \text{ kg/m}^3$$

$$E = 10^7 \text{ Pa}$$

$$\nu = 0$$

The beam's dimensions are $L = 10m$ and $H = 1m$, with a distributed load at the

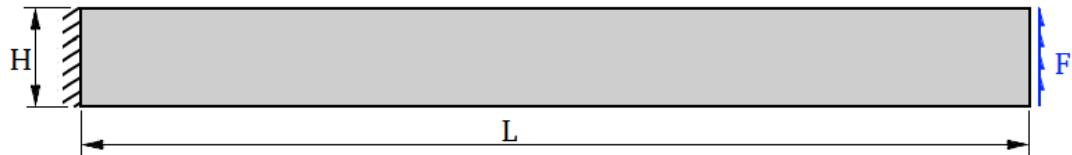


Figure 4.10: Cantilever geometry

right face of $F = 1[N/m]$. According to the Tymoshenko beam theory, the vertical displacement w at the tip must be

$$w_{max} = \frac{F L^3}{3 E I} + \frac{F L}{10/12 h \mu} = 0.0004024m \quad (4.49)$$

Structured meshes with decreasing element size were created to check convergence. Mesh sizes from $h = 1m$ to $h = 0.001m$ were used. Quadratic convergence can be observed in Figure 4.11 for the error in the max displacement of the beam.

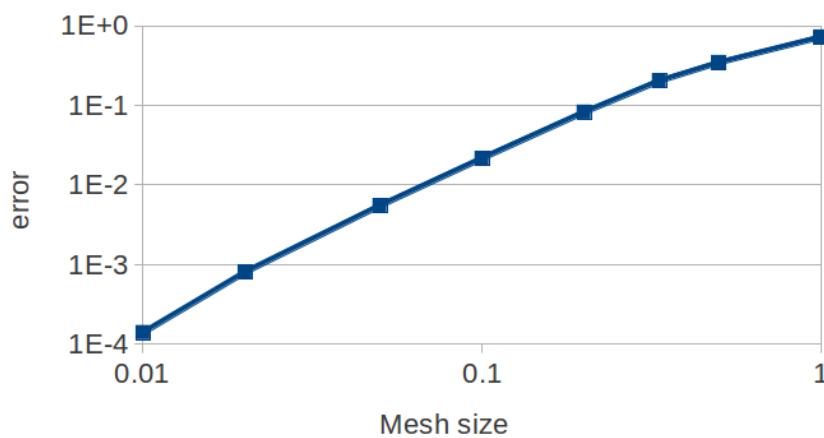


Figure 4.11: Mesh convergence for the cantilever beam

Static cantilever immersed in fluid

To test the accuracy of the solver when both solid and fluid are present, the following numerical was designed. A beam under small deformations is surrounded by a very light fluid. Inertial terms are neglected to test the statical solution, leading to a Stokes

problem in the fluid areas. The viscosity is set as small as possible to allow for a faster convergence to the statical solution while maintaining stability. The geometry can be seen in Figure 4.12. The materials' properties are:

$$\rho_q = 1 \text{ kg/m}^3 \quad \rho_p = 0.001 \text{ kg/m}^3$$

$$E = 10^7 \text{ Pa} \quad \mu_p = 0.01 \text{ Pa.s}$$

$$\nu = 0$$

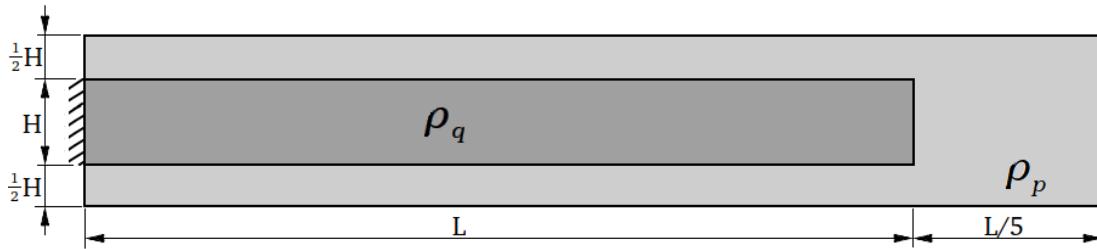


Figure 4.12: Cantilever beam with fluid geometry

The beam's dimensions are $L = 10m$ and $H = 1m$. A vertical mass force of $q = 1[m/s^2]$ is applied to the whole domain. According to the Tymoshenko beam theory and correcting the force to account for the buoyancy effect, the vertical displacement w at the tip must be

$$w_{max} = (\rho_q - \rho_p) \left(\frac{q L^4}{8 E I} + \frac{1}{2} \frac{q L^2}{10/12 h \mu} \right) = 0.001511m \quad (4.50)$$

Structured meshes with decreasing element size were created to check convergence. Mesh sizes from $h = 1m$ to $h = 0.01m$ were used. To test the influence of the projection algorithm, the simulation was run with and without projection. In the latter case, the nodal pressure is kept instead of being obtained from the particles.

In order to test the behaviour of the split elements, in all the meshes the interface between the solid phase and the fluid is located at approximately the middle of the elements, as it can be seen in Figure 4.13. In this case the enrichment shape functions of the pressure are imperative due to the strong discontinuity of the properties, leading to a jump in the pressure field. If this variable was not enriched to allow for discontinuities, a strong pressure gradient would appear at the interface and therefore the resulting velocity field would be inaccurate. It must be noticed that Figure 4.13 does not correctly

represent the pressure at the interface elements since the field is printed with linear interpolation, while in reality it is discontinuous due to the enrichments.

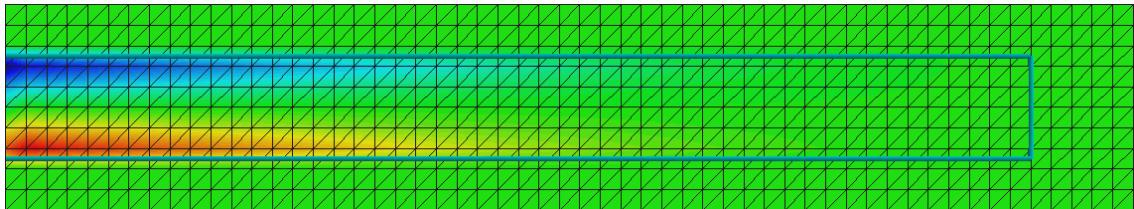


Figure 4.13: Beam interface and pressure distribution

From Figure 4.14 it is clear the detrimental effect of the projection stage. While the algorithm without projection is able to maintain the quadratic convergence of the solid-only case, the results with projection show a convergence slope of only 3/2.

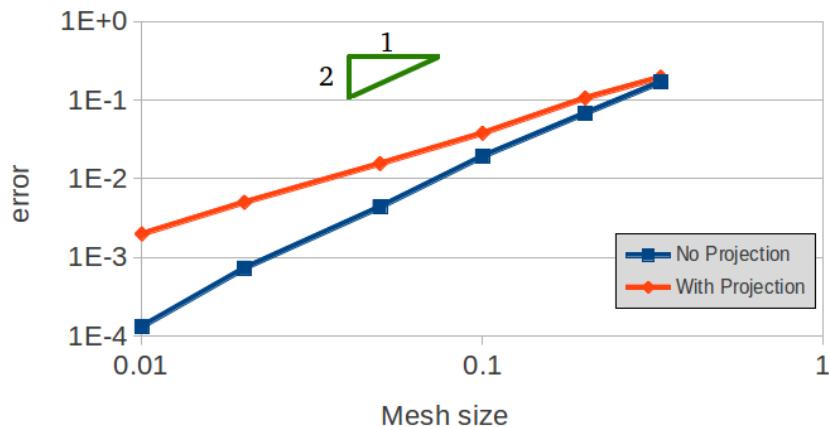


Figure 4.14: Convergence for the cantilever beam in fluid

Despite the solver with the projection stage provides a less accurate solution, it has acceptable errors for all the evaluated mesh sizes. Even for coarse meshes of only 3 elements along the thickness, the errors are still low at around 10 %.

Dynamic cantilever immersed in viscous fluid

As the last example case to test the solver in dynamic problems and to evaluate the effect of the viscosity of the fluid, the same geometry of the previous problem was selected

but adding inertial and viscous terms. Chosen mesh size is $h = 0.1$ and the properties of the materials are:

$$\rho_q = 1 \text{ kg/m}^3 \quad \rho_p = 0.0001 \text{ kg/m}^3$$

$$E = 10^7 \text{ Pa} \quad \mu_p = 0 - 5 \cdot 10^{-3} \text{ Pa.s}$$

$$\nu = 0$$

For the inviscid case and since the density of the second fluid is negligible, it is possible to calculate the analytical vibration frequency following expression (4.54).

$$\omega = \sqrt{\frac{C_n E I}{L \rho}} \quad (4.51)$$

where the subscript n denotes the mode and C_n and the positive roots of equation (4.55)

$$1 + \cos(C_n) \cosh(C_n) = 0. \quad (4.52)$$

Taking the first vibration mode, $C_1 = 1.8751$ and the frequency of the beam becomes (4.56)

$$\omega = \sqrt{\frac{C_1 E I}{L \rho}} = 32.097 \text{ rad/s} \rightarrow 5.108 \text{ Hz} \quad (4.53)$$

The vertical displacements of the tip of the beam can be seen in Figure 4.15. The frequency observed for the inviscid case is 5.17Hz , in good agreement with the theoretical value. On the other hand, for all the cases the displacement oscillates above -0.0015 , slightly stiffer than the analytical value due to the coarse mesh.

As for the influence of the viscosity, in Figure 4.15 it is possible to observe the damping effect of the solution. The inviscid problem conserves perfectly the energy of the system, while the highest viscosity case dampens completely the movement in four oscillations.

3D benchmark. Dynamic cantilever

The last example tests the PFEM-2 algorithm in 3D dynamic problems. The same geometry of the previous 2D cantilever was selected but restoring the inertial terms and

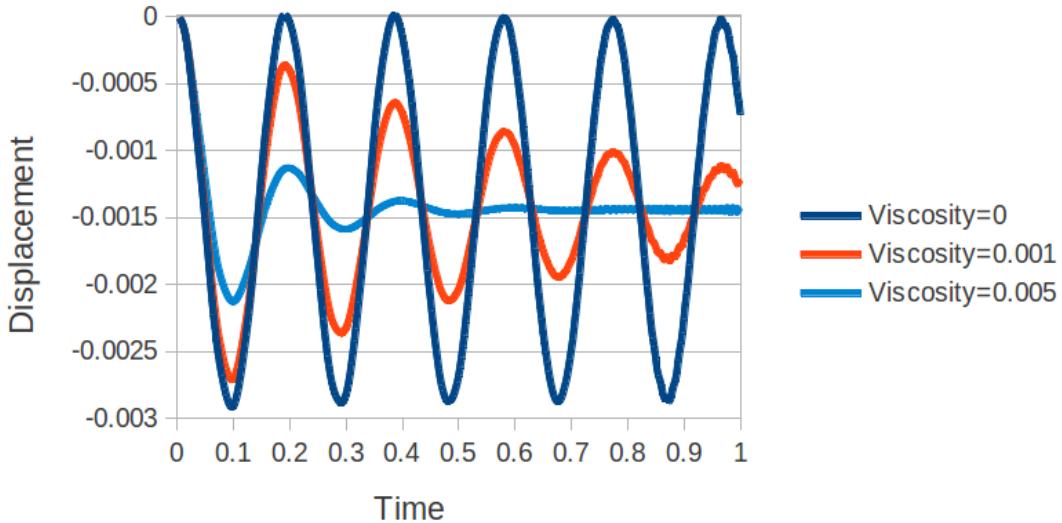


Figure 4.15: Time evolution of the cantilever beam for different viscosities

taking the width $b = 1$ to obtain a square section. The geometry can be seen in Figure 4.16, with a vertical mass force to initiate the movement. The mesh size is $h = 0.125$ in the surroundings of the beam and $h = 0.6$ in the distant fluid areas, for a total of 283500 4-noded tetrahedra and more than four million particles. The time step Δt is set for 0.001 and the properties of the solid are the same as the 2D static cantilever beam example. The sub indexes q and p are used for the solid and fluid properties respectively

$$\rho_q = 1 \text{ kg/m}^3 \quad E = 10^7 \text{ Pa} \quad \nu = 0$$

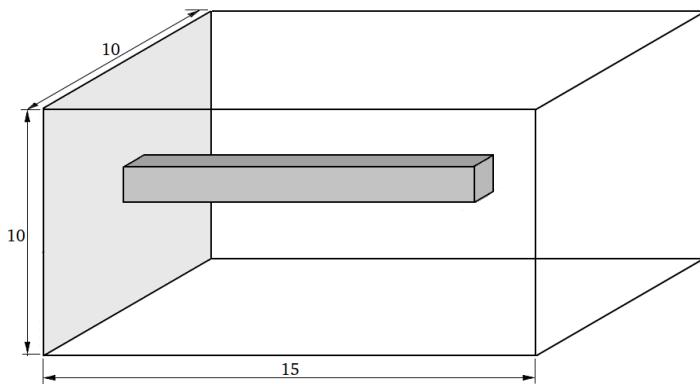


Figure 4.16: Tridimensional cantilever geometry

For cases in which the density of the fluid is negligible, it is possible to calculate the

analytical vibration frequency in vacuum with the following expression:

$$\omega_{vac} = \sqrt{\frac{C_n^4 E I}{L^4 \rho_q}} \quad (4.54)$$

where the subscript n denotes the mode and C_n are the positive roots of the following equation

$$1 + \cos(C_n) \cosh(C_n) = 0. \quad (4.55)$$

Taking the first vibration mode, $C_1 = 1.8751$ and the frequency of the beam becomes:

$$\omega_{vac} = \sqrt{\frac{C_1^4 E I}{L^4 \rho_q}} = 32.097 \text{ rad/s} \rightarrow 5.108 \text{ Hz} \quad (4.56)$$

The frequency observed in the simulation for low fluid densities ($< 0.001 \text{ kg/m}^3$) is 5.175 Hz , in good agreement with the theoretical value. The slightly higher value is caused by the low number of elements in the thickness, which leads to a stiffer behaviour.

To test the influence of the fluid density, according to Van Eysden and Sader (2006) it is possible to estimate the lower vibration modes for rectangular section beams immersed in a fluid using the following formula:

$$\omega_{fluid} = \omega_{vac} \left(1 + \frac{\pi \rho_p b}{4 \rho_q H} \Gamma_f \right)^{-1/2} \quad (4.57)$$

where H and b are the height and width of the beam and Γ_f is the hydrodynamic function which depends on the viscosity and the cross section of the beam. For inviscid flows and square sections $\Gamma_f = 1.513$ Brumley et al. (2010).

Fluid densities ρ_p from 0.0001 kg/m^3 to 0.99 kg/m^3 were selected to validate the algorithm. The results can be seen in Figure 4.17, together with the comparison against Equation (4.57). Analysing this graph, it is clear that the numerical solution diverges from the expected curve as the fluid density increases. This is caused by the limitations of the standard linear elements used for the velocity field, which lead to interface elements that are assembled with averaged stiffness properties between the fluid and the solid and, therefore, the added mass effect is larger than it should be.

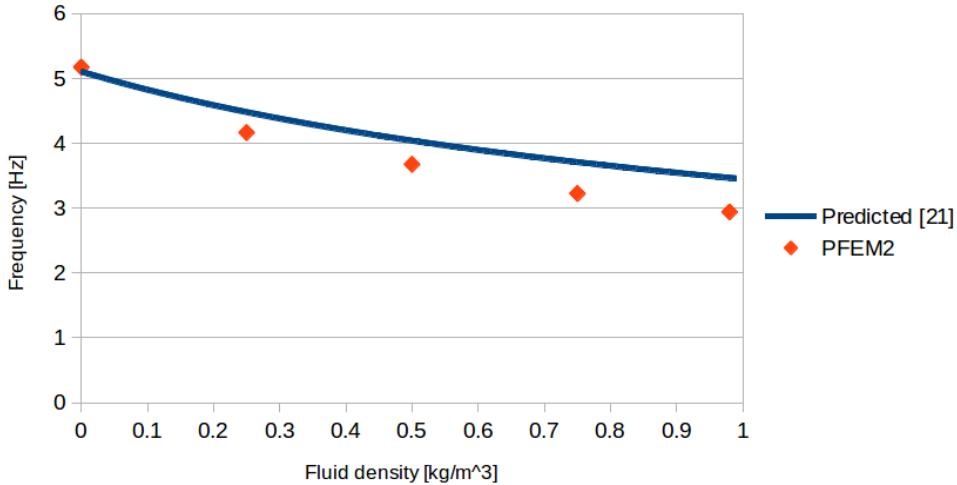


Figure 4.17: Oscillation frequency of the 3D cantilever beam for different fluid densities

To analyse the convergence of the method with the mesh size, the case with a fluid density $\rho_f = 0.5 \text{ kg/m}^3$ was selected and the projection stage was deactivated. Mesh sizes from $h = 0.33$ to $h = 0.08$ were chosen (3 to 12 elements in the thickness of the beam). In all cases the interface was located in the middle of the elements to test the worst scenario. A quadratic convergence with the mesh size can be observed in Figure 4.18, indicating that the interaction between the fluid and the beam is accurately simulated.

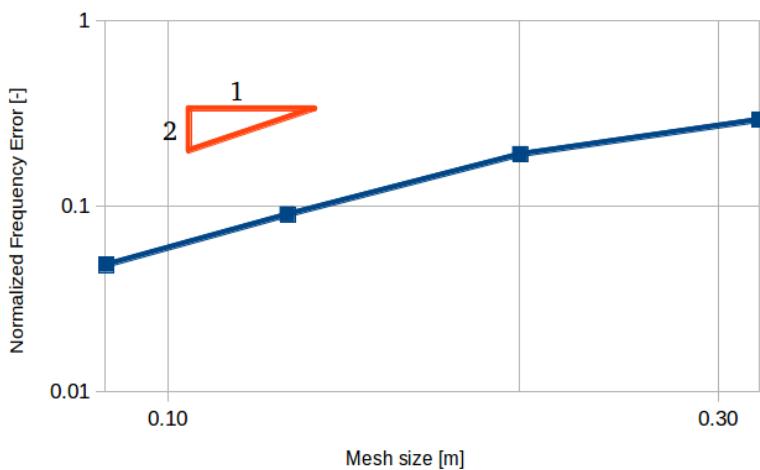


Figure 4.18: Mesh convergence for the oscillation frequency of the 3D cantilever beam

Turek FSI benchmark

To test the PFEM-2 algorithm in a fully coupled scenario, the Turek FSI benchmark Turek and Hron (2006), was simulated. This problem is typically used as a reference to validate new FSI strategies and it is particularly challenging for staggered schemes due to the similar properties of the fluid and the solid. On the other hand, the monolithic solver used in this work is well suited for this benchmark. Unlike the previous examples presented in this work (in which the PSPG stabilization was used) in this example the OSS stabilization is employed, with the explicit projection of the pressure gradient in the RHS. This is done to avoid having zeros in the RHS of the pressure equation of Equation (3.37) due to lack of gravity, which would provide excessive numerical diffusion.

The problem models a 2D rigid cylinder attached to an elastic beam immersed in a fluid stream. The dimensions and problem parameters can be found in Turek and Hron (2006). In this paper a fixed mesh of 48500 3-noded triangles was used. The frequency and amplitude of the oscillations are analysed to test the accuracy. For the case studied we use $Re = 200$, with a parabolic inlet velocity profile and the following parameters:

$$\rho_{fluid} = 1000 \text{ kg/m}^3 \quad \mu_{fluid} = 1 \text{ Pa.s}$$

$$\rho_{solid} = 1000 \text{ kg/m}^3 \quad E_{solid} = 1.4 \text{ MPa}$$

$$\nu_{solid} = 0.4 \quad \delta t = 0.005 \text{ s}$$

In Figure 4.19 the velocity contour is plotted after the stationary regime has been achieved. Finally, in Figure 4.20 the vertical displacement of the tip can be observed. In Table 1 the results are analysed. Compared to Turek's solution, both the amplitude and the frequency are 10 % below the reference value. This result is considered acceptable since only ten 3-noded triangular elements have been used to discretize the beam thickness.

	Frequency	Amplitude
Turek	5.3	0.0707
PFEM2	4.8	0.0610

Table 4.1: PFEM2 and Turek results (Turek and Hron, 2006)

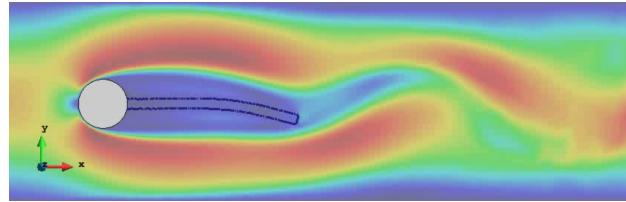


Figure 4.19: Turek FSI velocity contour

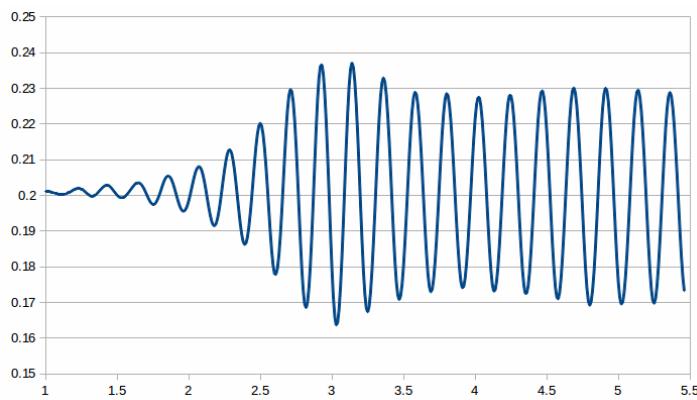


Figure 4.20: Tip displacement of the Turek FSI

4.4.2 Plasticity problems

Prandtl's punch problem with Drucker Prager plasticity

The shallow foundation case is a classic geomechanics problem. This test is used to analyse the stability of solids and consists on a rigid footing with an imposed vertical velocity, as shown in Figure 4.21. Lateral and lower walls have fixed velocities in x and y . In order to reduce the domain size of the problem, the symmetry on the vertical axis is used and therefore the domain reduces as shown in Figure 4.22. The parameters of the simulation are as follows, where the density is not defined since the acceleration and the gravity are neglected in the simulation.

$$V_y = -0.15 \text{ m/s}$$

$$c = 10 \text{ kPa}$$

$$E = 100 \text{ MPa}$$

$$\theta = 0^\circ, 15^\circ \text{ and } 30^\circ$$

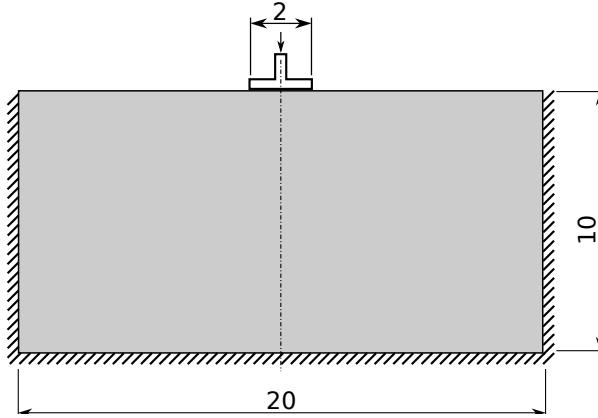


Figure 4.21: Prandtl's problem geometry

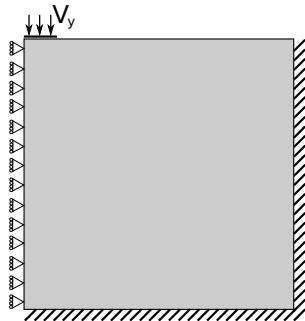


Figure 4.22: Prandtl's problem computational domain

The total simulation time is 1s with a mesh size of $h = 0.05m$, for a total of 2300 elements. Given the low cohesion of the material, the total deformation rate at the end can be considered to be only of plastic origin. The solution behaves as expected when the friction angle is increased.

4.5 Summary and Conclusions

In this chapter the solution strategy to solve solids and FSI problems has been developed. The method is based in the same general tools of the PFEM-2 described in Chapter 2 to solve the convection of particles and the improvements for the interface description. The customizations of the solver has been kept to a minimum in order to limit the complexity of the strategy and reduce the effort to solve solids and fluids within a single, unified frame.

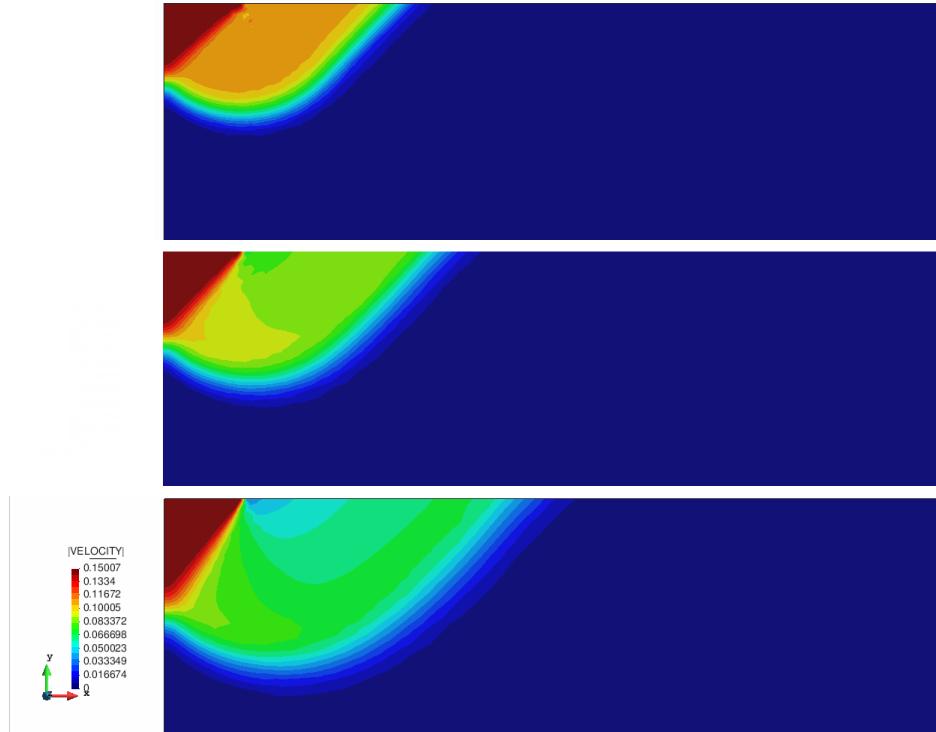


Figure 4.23: Footing velocity contour at $t = 1s$. Top: $\theta = 0^\circ$, middle: $\theta = 15^\circ$, bottom: $\theta = 30^\circ$

In order to ease the coupling between solids and fluids, the velocity and the pressure have been selected as the unknowns for both phases. In the case of solids, the pressure has been defined as the mean volumetric stress to match the definition of fluids, meaning that deviatoric stresses are treated as secondary variables. The material constitutive operator has been developed starting from a hypo elastic solid model. On the other hand, a Drucker Prager yield surface was selected to simulate the behaviour of granular materials. This yield criterion relates the admissible stresses to the mean volumetric stress of the material, providing a good approach to soils despite its simplicity. The DP surface was then coupled with a simple associative perfect plasticity model to compute the plastic flow of the material. Instead of using secant operators, the elastoplastic tangent operators were derived in order to use a residual-based strategy.

The resulting shape of the elemental matrices is highly similar to the one of fluids, thus obtaining a trivial coupling algorithm. In order to enhance the definition of the interfaces, enrichment shape functions were introduced at the split elements. The selected functions allow to obtain discontinuities in the pressure field across the interface, thus

avoiding numerical diffusion in the pressure that would otherwise reduce the stiffness of the solid bodies. In order to minimize this numerical diffusion, the fluid part of the element was not stabilized in the pressure, but rather considered slightly compressible by using a lumped pressure mass matrix, which does not produce pressure diffusion.

Numerical examples of elastic problems showed optimal (quadratic) convergence for the finite element problem when the projection stage was disabled. The pressure interface at the interfaces was correctly captured by using the discontinuous shape functions, showing no numerical diffusion. However, when the full PFEM-2 algorithm was used (including projection), the mesh convergence rate was severely affected, obtaining a slope of only $3/2$. This example clearly shows the detrimental, yet inevitable with the current projection kernel, effect of the numerical diffusion of the projection.

On the other hand, the numerical examples of the plastic problems showed a correct implementation of the solution strategy. The possibility to simulate solid problems with a good accuracy while being able to handle extremely large deformations provides an adequate alternative to simulate geomechanics problems from the initiation to its final deposition with a single computational tool.

Chapter 5

Landslides and debris flows simulation

5.1 Introduction

In the previous chapters a strategy to solve FSI problems was described. The examples show the validity of the solver to deal with the interaction of two materials with completely different properties, both for multi fluid and FSI cases. The monolithic nature of the algorithm allows for direct convergence for simple materials, without the need of iterating between two systems.

While this algorithm alone is suitable for problems in which the mobilized mass has a size similar to the complete domain dimensions, it cannot be applied to problems with relatively small volumes moving across large distances, such as debris flows. Analysing the debris flow displacement compared to its size(Figure 5.1), it becomes clear that expanding the shown discretization to the complete slope would be computationally prohibitively.

Figure 5.1 provides key information to tackle the problem: the size of the debris flow is almost negligible compared to the slope. Moreover, the terrain *upstream* of the debris flow, the one that has not been reached by the fluidized phase, has not been affected and therefore the stress state has not changed either. Figure 5.2 shows that the mean total pressure (trace of the stress tensor) in the slope remains unaltered almost everywhere as the debris flow flows down. The affected area is limited to the soil which is in close proximity to the debris flow.

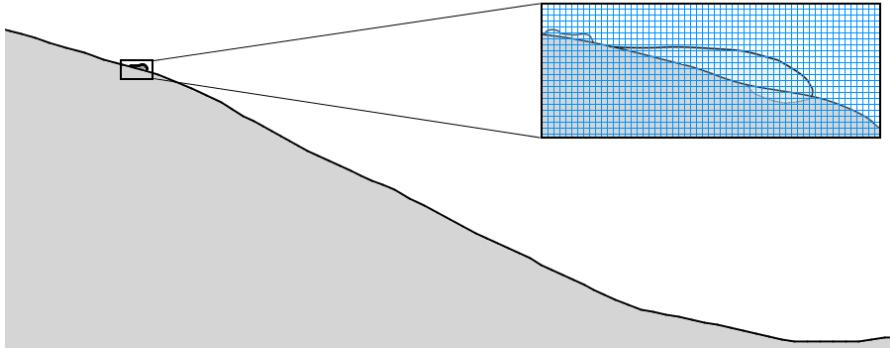


Figure 5.1: Discretization of the debris flow

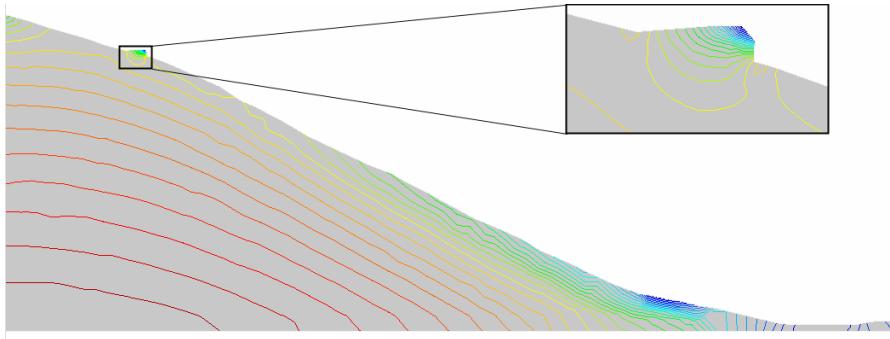


Figure 5.2: Mean solid pressure of the slope

From these observations, two conclusions can be drawn. On one side, it is not necessary to compute the evolution of the variables in the whole domain. Since most of the slope remains unaltered to the flowing mass, it is only necessary to compute the dynamic phenomena in the region surrounding the debris flow. The part of the domain affected by it is much smaller than the total domain but it suffers from extremely large deformations. This required a technique able to update the domain. A two-resolutions approach can reduce drastically the required computational power, using two different spatial domains: a fixed, coarse mesh for the initial, static stress state and a moving one tracking the dynamics of debris flow.

On the other hand, it can be seen that the displacements and the deformations of the moving mass cannot be considered in the *small deformations* regime. This is important to consider because, even though the mass is still a granular material with its corresponding rheology, the stress response does no longer depend on its full history of deformation, but only on the current strain rate. In other words, the deformations

caused at any time-step are much larger than the required strains to initiate the plastic response. Then it follows that the current state alone is enough to define the direction and magnitude of the material response. This rheology corresponds to the behaviour of a non Newtonian fluid, which provides the advantage of being less computationally expensive than solids and also allows to increase dramatically the time-steps, further reducing the computational cost.

As the last addition to enhance the modelling of the debris flows, a coupled model for water - granular material will be developed. Unlike the FSI solver developed in Chapter 4, the aim of this strategy will be simulating the interaction between a porous material and the water flowing through its pores, meaning that in the macro-scale both domains are overlapping. The technique will make use of both the solid and fluid strategies developed in previous chapters.

In the following sections three strategies to enhance the modelling of debris flows will be presented. First a non-Newtonian constitutive model will be explained, then an inter-granular water flow and in the last section a two-resolutions approach will be presented.

5.2 Non-Newtonian Rheology

Simulating the response of granular materials as non-Newtonian fluids has been performed extensively in the literature (Quecedo et al., 2004), (Salazar et al., 2015), . Larese et al. (2012a) used a Mohr-Coulomb yield criterion to simulate the failure in rockfill dams, with excellent matching to experimental results.

To give an insight on the model characteristics, let us recall the definition of the perfect plasticity model and a viscous incompressible fluid. In Figure 5.3(a), the material response of a perfect plasticity Drucker Prager (DP) model under constant pressure is represented in the deformation - shear stress plane ($\epsilon - \tau$) . In Figure 5.3(b), the force of a non-Newtonian fluid is plotted for a given deformation rate.

Using the rheology of a (non-viscoelastic) fluid, it becomes unnecessary to track the historic deformation of the material. Since the force response depends solely on the current deformation rate, the complexity to obtain the stress state of the material point is reduced drastically compared to a complete elastoplastic solid model. For the simulation of a granular materials using the DP yield surface, the elastic part of the curve is

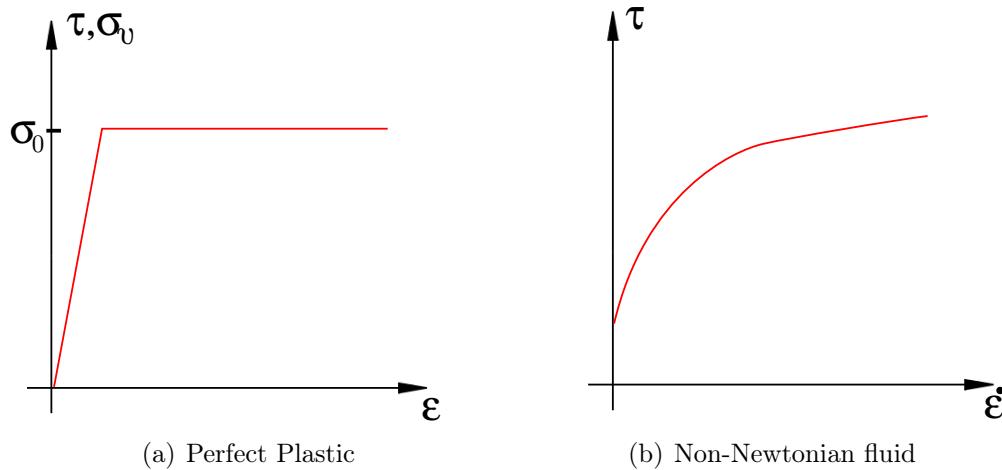


Figure 5.3: Material constitutive models

directly omitted. The material is either undeformed (for loads below the yield surface) or undergoing plastic deformation. In Figure 5.4 the ideal non-Newtonian response to simulate a DP is shown in the red solid line, corresponding to a Bingham fluid whose threshold depends on the pressure. For all the values of the deformation rate, the shear rate remains constant at the yield limit.

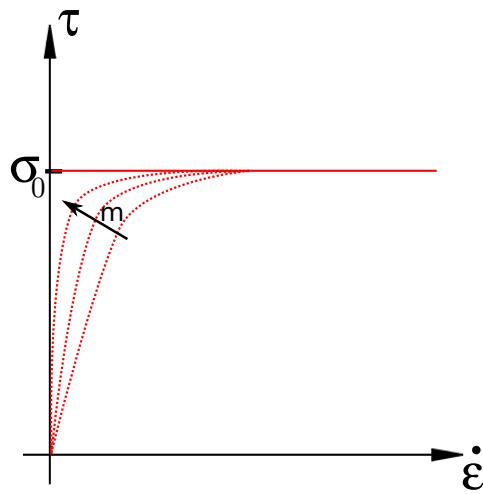


Figure 5.4: Bingham fluid for Drucker-Prager materials. Solid line: ideal response. Dotted line: regularization approximation

Now recalling the definition of the shear stress in a fluid, it is possible to obtain the expression for the viscosity of the DP model, where the maximum shear stress τ depends

on the pressure p . (The viscosity has been taken outside the ∇ operator for more clarity, assuming a constant viscosity in domain of interest)

$$\boldsymbol{\tau}(p) = \mu_{DP} \mathbf{d} \quad (5.1)$$

Defining an scalar measure for the stress $\tau_0 = \sqrt{\boldsymbol{\tau} : \boldsymbol{\tau}}$ and another for the deformation rate $\dot{\gamma} = \sqrt{\mathbf{d} : \mathbf{d}}$, the expression for the viscosity reads:

$$\mu_{DP} = \frac{\tau_0}{\dot{\gamma}} \quad (5.2)$$

The problem with Equation (5.2) is that, since the stress response must have a finite value even for zero velocities, the viscosity μ becomes infinite in this scenario. Moreover, the material response in the unyielded case must be lower than the yield strength and thus the equation is, again, not valid. To overcome this limitation, the material response must be regularized for low strains. In this work, the regularization proposed by Papanastasiou (1987) is used:

$$\mu_{DP} = \frac{\tau_0}{\dot{\gamma}} (1 - e^{-m\dot{\gamma}}) \quad (5.3)$$

where m is the regularization parameter, whose influence in the model is shown in Figure 5.4. The selection of the m parameter is a trade off between accuracy and conditioning of the system. Low values lead to low viscosities and therefore the performance of the iterative linear solver is not affected severely. However this translates into relatively large velocities when the body should actually be static in non-fluidized areas. On the other hand, high values of m allow for negligible velocities in the unyielded condition but can affect the performance of the linear solver. In this work, $m = 10^4$ was chosen.

Since the viscosity parameter depends on the strain rate itself, the system of equations becomes non-linear. Linearising by using information from the previous iteration, the viscosity for the Drucker-Prager rheology can be expressed as:

$$\mu_{DP}^{n+i+1} = \frac{c + \tan(\theta) p^{n+i}}{\dot{\gamma}} (1 - e^{-m\dot{\gamma}}) \quad (5.4)$$

Using the previous constitutive equation in the monolithic fluid solver of Chapter 3

allows to simulate the granular material rheology in the large deformations regime.

5.3 Granular material with seepage

In many cases the initiation of a debris flow is triggered by intense rainfall. As the water penetrates through the pores of the granular material, the increasing fluid pressure decreases the strength of the soil. The apparent stress state of the granular material can modeled by the *effective stress* σ^{eff} (Terzaghi, 1925):

$$\sigma^{eff} = \sigma - I p^w \quad (5.5)$$

where σ the stress tensor of the granular material and p^{water} the pore water pressure. This expression indicates that the inter-granular forces will be directly affected by the water phase. As the water pressure increases, a part of the external forces is balanced by it, therefore reducing the contact pressure between grains. This is a key concept in granular materials, in which the yield strength is strongly related to the granular pressure. A slope that is originally unyielded can easily collapse by increasing the pore water pressure.

By coupling the effect of the pore water pressure with a drag model between water and the soil particles, it is possible to obtain an accurate yet simple approach to simulate porous materials. In this work, the model developed by Larese et al. (2012b) will be implemented. Fixing the porosity (voids fraction in the skeleton), the resulting equations are greatly simplified, allowing to tackle the problem with the already developed PFEM-2 tools. Figure 5.5 shows the problem to be solved. Inside the computational domain Ω , the water volume Ω_w and the granular material body Ω_s are shown. The balance equations of both phases will be solved separately, by using explicit information from the previous iteration of the other phase to compute the coupling terms. In this sense, it must be noted that the areas of the domain not filled by these phases will be still simulated by assuming that air is filling these void areas. This section presents the model to simulate the water flow through the pores, as well as the interaction forces between this phase and the granular material.

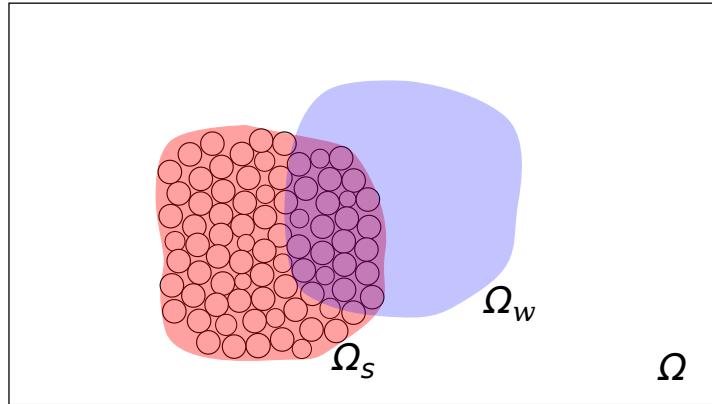


Figure 5.5: Water and granular material domains

5.3.1 Problem statement

An accurate computation of the water flow across the soil skeleton is required to correctly determine the forces on the granular phase and therefore their evolution in time. In this work a simple model will be used, starting by following hypothesis:

- **Constant porosity:** The fraction of voids in the granular material points will remain constant throughout the simulation
- **Water is a Newtonian fluid**
- **Water is incompressible**

The application of the previous hypothesis simplifies drastically the solution of the equations. Since the pore size of the granular material remains constant at all times (neglecting volumetric deformation due to the relatively large bulk modulus), the equations to model the soil remain basically unmodified with the exception of the force terms due to the water presence. In other words, the structure of the system remains unaltered for the granular material.

On the other hand, the constant volume fraction hypothesis implies that the model will be limited to simulations in which the inter-granular forces are dominant and the total volume of the phase does not change. This restricts the envelop of simulation to only non-saltation/collisional regimes, in which the grains behave as a continua rather than disperse distribution of particles in the water phase. In Figure 5.6 both regimes are represented.

5.3.2 Water model

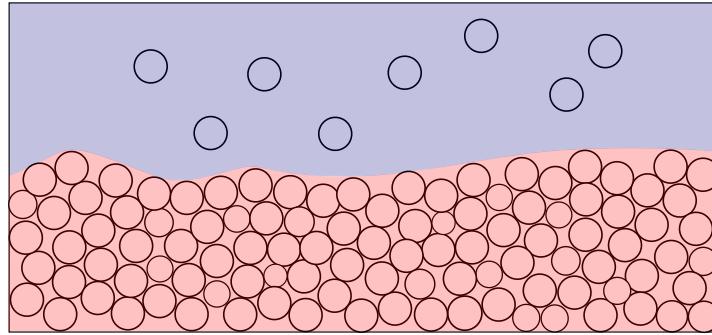


Figure 5.6: Granular material regime. Top: collisional. Bottom: Frictional. Only the latter can be simulated as a continuum occupying the red region.

Balance equations for water through porous media

Having fixed the void ratio related to each granular material point, it becomes an input value. The porosity n is by definition :

$$n = \frac{\Omega_{void}}{\Omega} = \frac{\Omega_{void}}{\Omega_{void} + \Omega_{grains}} \quad (5.6)$$

where Ω is a representative volume and Ω_{void} is the empty part of it. When no solid particles are present, the porosity becomes $n = 1$. In Figure 5.7 the reference and void volume are shown. At this point it is useful to define a two different measures for the fluid velocity. The first concept is the Darcy velocity \mathbf{V} . By definition is the mean velocity averaged in the entire reference volume Ω , as if $n = 1$.

$$\mathbf{V} = \int_{\Omega} \mathbf{u}(x, y, z) \quad (5.7)$$

where $\mathbf{u}(x, y, z)$ is the exact velocity at any point of the domain.

On the other hand, considering that the size of the grains is much smaller than the reference volume, the *fluid velocity* $\bar{\mathbf{V}}$ is the average of the exact velocity respect to the void volume:

$$\bar{\mathbf{V}} = \int_{\Omega_{void}} \mathbf{u}(x, y, z) \quad (5.8)$$

By using the definition of the porosity (5.6), it is possible to relate the two measures of the velocity as:

$$\mathbf{V} = n \bar{\mathbf{V}} \quad (5.9)$$

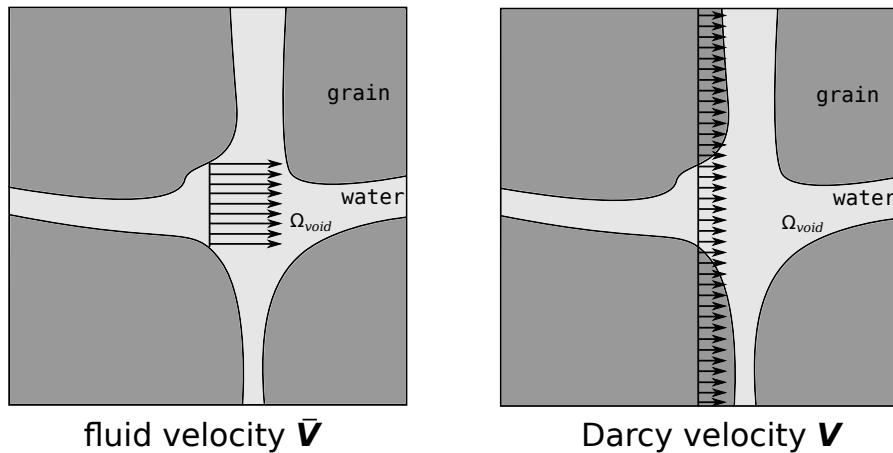


Figure 5.7: Average fluid velocity and Darcy velocity

Volume conservation

In Figure 5.8 the inlet and outlet flows in an infinitesimal volume are shown. The components of the fluid velocity are written as : $\bar{\mathbf{V}} = [\bar{u}, \bar{v}, \bar{w}]$. By assuming that the porosity n remains constant in this control volume $d\bar{x} d\bar{y}$, it is possible to obtain the volume conservation equations.

Writing the flux across the boundaries of the domain (considering the normals point outside):

$$\left(\bar{u} + \frac{\partial \bar{u}}{\partial x} d\bar{x} \right) d\bar{y} - \bar{u} d\bar{y} + \left(\bar{v} + \frac{\partial \bar{v}}{\partial y} d\bar{y} \right) d\bar{x} - \bar{v} d\bar{x} = 0 \quad (5.10)$$

Subtracting like terms and using the definition of the porosity $d\bar{x}d\bar{y} = n dx dy$:

$$\left(\frac{\partial \bar{u}}{\partial x} \right) n dx dy + \left(\frac{\partial \bar{v}}{\partial y} \right) n dx dy \quad (5.11)$$

And finally using the definition of the Darcy velocity, the volume conservation equation

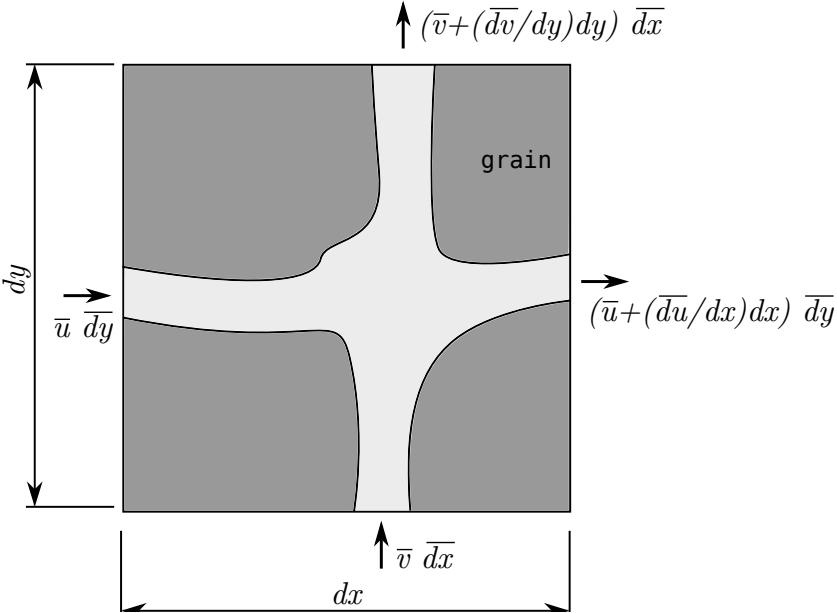


Figure 5.8: Fluid volume balance in a porous media

simplifies to:

$$\left(\frac{\partial u}{\partial x} \right) dx dy + \left(\frac{\partial v}{\partial y} \right) dx dy \quad (5.12)$$

Or:

$$\nabla \cdot \mathbf{V} = 0 \quad (5.13)$$

Momentum conservation

The procedure to obtain the momentum equation in the Lagrangian framework for a fluid in porous media consists on integrating forces and change in momentum only within the void volume $d\bar{x}d\bar{y}$ of the control domain $dxdy$. The acting forces are the same as the case of traditional fluid problems but with the addition of the drag caused by the interaction with the grains of the solid material $\hat{\mathbf{P}}$. Writing the equations for the x direction:

$$\rho \frac{\partial \bar{u}}{\partial t} d\bar{x}d\bar{y} + \frac{\partial p}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} d\bar{x}d\bar{y} - \rho g_x d\bar{x}d\bar{y} + \hat{P}_x d\bar{x}d\bar{y} = 0 \quad (5.14)$$

where the shear forces τ_{xy} are caused by the viscosity. As seen in Chapter 3, this term depends on $\nabla \cdot (2\mu \nabla^S \bar{\mathbf{V}})$

At this point an additional assumption can be made to simplify Equation (5.14). Since the pores between grains are much smaller than the control volume, neither the Darcy velocity nor the fluid velocity $\bar{\mathbf{V}}$ are representative of the actual fluid flow velocity between the pores. In other words, the shear deformation rate at this scale does not provide useful information to compute viscous forces. Based on this, the shear force term is eliminated and the momentum equation reads:

$$\rho \frac{\partial \bar{u}}{\partial t} d\bar{x}d\bar{y} + \frac{\partial p}{\partial x} d\bar{x}d\bar{y} - \rho g_x d\bar{x}d\bar{y} + \hat{P}_x d\bar{x}d\bar{y} = 0 \quad (5.15)$$

Now using the definition of the Darcy velocity and the transformation $d\bar{x}d\bar{y} = n d\bar{x}d\bar{y}$:

$$\begin{aligned} & \left(\rho \frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} n - \rho g_x n + P_x \right) dx dy = \\ & \rho \frac{\partial u}{\partial t} + n \frac{\partial p}{\partial x} - n \rho g_x + P_x = 0 \end{aligned} \quad (5.16)$$

where $P_x = \hat{P}_x n$.

Finally generalizing the momentum and volume conservation equations to the three dimensions of the domain, the system that defines the behaviour of fluids in a porous material becomes:

$$\begin{aligned} & \rho \frac{\partial \mathbf{V}}{\partial t} + n \nabla p - n \rho \mathbf{g} + \mathbf{P} = 0 \\ & \nabla \cdot \mathbf{V} = 0 \end{aligned} \quad (5.17)$$

The previous set of equations define the balance of momentum and the volume conservation in the Lagrangian framework. In order to fully define the movement of the water material points, a drag model has to be defined for \mathbf{P} and the streamline integration of the PFEM-2 must be used to compute the convection of the particles. In this sense, it must be noted that the particles must not be convected with the Darcy velocity but rather with the fluid velocity $\bar{\mathbf{V}}$. The reason behind this is that the Darcy velocity is averaged in the complete control volume (analogous as taking $n = 1$), while in reality it changes according to the material properties. For this reason, even though only the Darcy velocity \mathbf{V} is required to solve the system (5.17), the fluid velocity $\bar{\mathbf{V}} = \frac{1}{n} \mathbf{V}$ will be also be computed. This task will be performed before the streamline integration in

order to correctly asses the effect of the porosity in the convection of the particles.

Drag model

The first empirical law to describe the flow of a fluid across a porous material was formulated by Darcy in 1856. The widely used model states that the loss of energy experimented by the fluid can be modeled by a pressure drop that depends linearly on the Darcy velocity:

$$\frac{dp}{dx} = \frac{\mu}{k} V_x \quad (5.18)$$

where μ is the dynamic viscosity of the water and k the permeability of the soil. This parameter can be obtained experimentally or using the following expression, in which the coefficients are set for rockfill material.

$$k = \frac{n^3 D_p^2}{5(1-n)^2 \theta} \quad (5.19)$$

where D_p is the representative particle size and θ is the shape coefficient of the particles.

Despite its simplicity, this formula provides an accurate approach in many geotechnical applications. Unfortunately, this linear behavior is only valid for low Reynolds numbers, therefore limited to low velocities and small grain sizes. As the particle size increase, this linearity is lost. A physical explanation is that the pressure drag becomes larger than the parasitic(viscous) drag. Since the pressure drag depends on the square of the velocity, the linear Darcy model is no longer valid. It must be noted that splitting the drag into linear and quadratic terms is not the only choice to represent the nonlinearities. While it provides good results in this work, other materials might require different approximations, for example power laws with arbitrary coefficients.

Following the work by Larese et al. (2012b), the Ergun coefficients will be used (Ergun and Orning, 1949) in this thesis. Using the linear and quadratic terms, the resistance law to obtain the pressure drop can be expressed as:

$$\frac{dp}{dx} = E_1 V_x + E_2 V_x^2 \quad (5.20)$$

where for a shape coefficient $\theta = 30$, adequate for rounded rockfill material:

$$E_1 = 150 \frac{(1-n)^2}{n^3} \frac{\mu}{D_p^2} \quad E_2 = 1.75 \frac{(1-n)}{n^3} \frac{\rho}{D_p} \quad (5.21)$$

It must be noted that the velocity V_x of the pressure drop term (5.20) is actually the velocity difference between the fluid and the granular material. Writing the granular material velocity as \mathbf{V}_s , the pressure drop due to viscous effect \mathbf{P} can be expressed as:

$$\mathbf{P} = E_1(\mathbf{V} - \mathbf{V}_s) + E_2||\mathbf{V} - \mathbf{V}_s||(\mathbf{V} - \mathbf{V}_s). \quad (5.22)$$

Fractional step strategy

The viscosity of the water plays a dominant role in the evolution of the flow through a porous media. As seen in the Ergun coefficients (Equation (5.21)), it directly affects the forcing terms. However, the absence of traditional viscous terms in the form $\nabla \cdot (\mu \nabla^s \mathbf{V})$ implies that the system is mathematically equivalent to a perfect fluid (inviscid) fluid flow. As explained in Chapter 3, low or zero viscosity problems are well suited to be solved with a fractional step strategy. By doing so, the computational time can be reduced considerably while maintaining excellent accuracy.

The first step to solve the equations is using finite differences in time to obtain the acceleration and an implicit, first order scheme for the pressure terms. The discretized system in time becomes:

$$\begin{aligned} \rho \frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\Delta t} + \nabla p^{n+1} n - \rho \mathbf{g} n + \mathbf{P}^{n+1} &= 0 \\ \nabla \cdot \mathbf{V}^{n+1} &= 0 \end{aligned} \quad (5.23)$$

Now linearizing the second Ergun coefficient E_2 with the previous velocity difference $\Delta \mathbf{V}^n = \mathbf{V}^n - \mathbf{V}_s^n$ and inserting the definition of \mathbf{P} into the previous equation with an implicit dependence on the fluid velocity:

$$\begin{aligned} \rho \frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\Delta t} + \nabla p^{n+1} n - \rho \mathbf{g} n + \rho(\mathbf{V}^{n+1} - \mathbf{V}_s^n) \left(\frac{E_1}{\rho} + \frac{E_2}{\rho} ||\Delta \mathbf{V}^n|| \right) &= 0 \\ \nabla \cdot \mathbf{V}^{n+1} &= 0 \end{aligned} \quad (5.24)$$

Rearranging the terms and setting $E_{12} = \frac{E_1}{\rho} + \frac{E_2}{\rho} ||\Delta \mathbf{V}^n||$:

$$\begin{aligned} \mathbf{V}^{n+1} + \frac{\nabla p^{n+1} n}{\rho \left(\frac{1}{\Delta t} + E_{12} \right)} &= \frac{\left(\frac{\mathbf{V}^n}{\Delta t} + \mathbf{g} n + \mathbf{V}_s^n E_{12} \right)}{\left(\frac{1}{\Delta t} + E_{12} \right)} \\ \nabla \cdot \mathbf{V}^{n+1} &= 0 \end{aligned} \quad (5.25)$$

Splitting the final velocity into a compressible (ignoring the pressure) and a divergence free part:

$$\mathbf{V}^{n+1} = \hat{\mathbf{V}}^{n+1} - \frac{\nabla p^{n+1} n}{\rho \left(\frac{1}{\Delta t} + E_{12} \right)} \quad (5.26)$$

Inserting the definition into the momentum equation and incompressibility equation of (5.27) and reordering the steps, the fractional step strategy for fluids in porous materials reads:

$$\begin{aligned} 1) \hat{\mathbf{V}}^{n+1} &= \frac{\left(\frac{\mathbf{V}^n}{\Delta t} + \mathbf{g} n + \mathbf{V}_s^n E_{12} \right)}{\left(\frac{1}{\Delta t} + E_{12} \right)} \\ 2) \nabla \cdot \left(\frac{\nabla p^{n+1} n}{\rho \left(\frac{1}{\Delta t} + E_{12} \right)} \right) &= \nabla \cdot \hat{\mathbf{V}}^{n+1} \\ 3) \mathbf{V}^{n+1} &= \hat{\mathbf{V}}^{n+1} - \frac{\nabla p^{n+1} n}{\rho \left(\frac{1}{\Delta t} + E_{12} \right)} \end{aligned} \quad (5.27)$$

Discretized forms of the equations

Using the test functions \mathbf{w} and q and following the same procedure described in Chapter 3, the three steps in the spatial discrete form read:

$$\begin{aligned} 1) \hat{\mathbf{V}}^{n+1} &= (\mathbf{M}(1/\Delta t + E_{12}))^{-1} \left(\mathbf{M}(1) \left(\frac{\mathbf{V}^n}{\Delta t} + \mathbf{g} n \right) + \mathbf{M}(E_{12}) \mathbf{V}_s^n \right) \\ 2) \mathbf{L} \left(\frac{n}{\rho \left(\frac{1}{\Delta t} + E_{12} \right)} \right) \mathbf{p}^{n+1} &= -\mathbf{G} \hat{\mathbf{V}}^{n+1} \\ 3) \mathbf{V}^{n+1} &= \hat{\mathbf{V}}^{n+1} - \left(\mathbf{M} \left(\rho \left(\frac{1}{\Delta t} + E_{12} \right) \right) \right)^{-1} \mathbf{G} p^{n+1} n \end{aligned} \quad (5.28)$$

As in the normal fluid problems, the enrichment shape function proposed by Coppola (2009) is used in the interface between water and air. The procedure described in Chapter 3 has to be followed in order to condense and later recover the extra degree of freedom, being the only difference the coefficients of the mass and Laplacian matrices.

5.3.3 Granular material model

Unlike the modified fluid flow formulation, the balance equations of the granular phase do not require important changes due to interaction between the water and the granular phase. Since the porosity of material points is considered to be constant in time, the only effect is the creation of the drag and buoyancy forces caused by the water. Setting the density of the material as the average density including pores and grains, the balance equations can be expressed using the following expression, where \mathbf{V}_s is the solid velocity:

$$\begin{aligned} \rho \frac{D\mathbf{V}_s}{Dt} &= \nabla \boldsymbol{\sigma} + \rho_s \mathbf{g} - \mathbf{P} + (1 - n) \nabla p_f \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho_s \mathbf{V}_s) &= 0 \end{aligned} \quad (5.29)$$

Since the coupling terms \mathbf{P} and p_f have already been defined and that the structure of the system of equations has not changed, the discrete form of system (5.29) will not be presented in this section. All the matrices can be computed following the expressions of Chapter 4 and adding the contribution of the water coupling terms.

The force contribution from the velocity difference $\Delta \mathbf{V}$ and the buoyancy are both assembled following an elemental-based strategy. Since the solution of the granular material system is to be performed after the water problem has been solved, the exact gradients and velocities of the fluid are readily available at the assembly time. This data is saved for each element when computing the third step of the fractional step fluid strategy (5.28). Figure 5.9 shows all the elemental data to be saved. Since the pressure gradient is constant, only one Gauss Point is required.

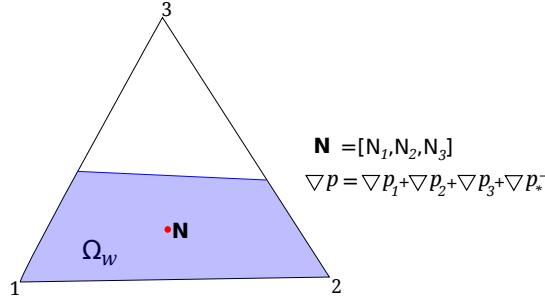


Figure 5.9: Stored information from the fluid domain required to compute the coupling terms in the granular domain system

5.3.4 Complete coupled model

Having developed the solution strategy for both the granular and fluid phases, it is now possible to define the algorithm for the coupled problem. The steps are:

Step 1) Compute $\bar{\mathbf{V}}_f = n\mathbf{V}_f$

Step 2a) Convect the fluid domain particles using $\bar{\mathbf{V}}_f$

Step 2b) Convect the granular domain particles using \mathbf{V}_s

Step 3) Project fluid and solid particles information into mesh

Fluid problem

Step 4) Detect fluid interface elements

Step 5) Solve the fluid system

Step 6) For each element containing water compute: Water volume, ∇p_f

Granular domain problem

Step 7) Detect interface elements

Step 8) Solve the solid system

8.1) Assemble system of equations with coupling terms using ∇p_f , \mathbf{V}_f

8.2) Solve linearized system

8.3) Calculate deformations

8.4) Update particle stresses

8.5) Go to 8.1 until convergence

Step 9) Update particles' velocities

5.4 Two resolutions approach

5.4.1 Spatial Domains

Assuming that the influence of the debris flow only acts in a small moving domain called *Calculation Domain* Ω_{calc} , it becomes sufficient to solve the dynamic problem only in the limited region Ω_{calc} . On the other hand, the stress state of the complete domain *Topographic Domain* Ω_{topo} will be determined by the static mass forces, implying that only one statical simulation will be necessary in this larger Ω_{topo} . Since the solution at this domain must not predict the failure but only a initial approximation of the stresses in equilibrium, a coarse discretization can be used. In Figure 5.10 sample meshes for both domains are shown.

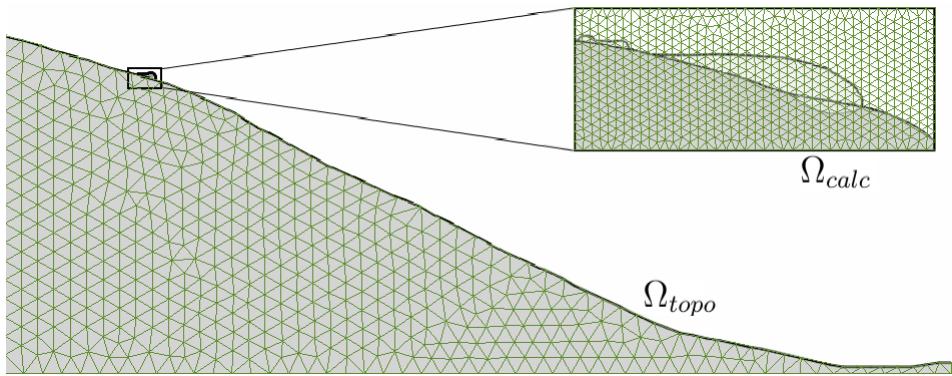


Figure 5.10: Discretizations on Ω_{topo} and Ω_{calc}

Topographic Domain

This domain contains the complete part of the slope affected by the debris flow, from its initiation to its deposition. Since only a single phase is analysed (the soil), any solver capable of dealing with geomaterials can be used to compute the initial stress state. However it is desirable to use the same mixed formulation described in Chapter 4 to avoid possible discrepancies that would create instabilities when passing information from Ω_{topo} to Ω_{calc} .

Besides providing the initial stress state of the complete slope, it also provides an insight on the areas which would be more prone to fail. Locating these areas is critical since the

dimensions of the *Calculation Domain* are very limited so to guarantee higher accuracy in the results.

Calculation Domain

The size of Ω_{calc} must be such that all the mobilized material can fit inside it during the whole evolution of the debris flow. Moreover, it must include the portion of the soil that is affected by the debris flows. However, the dimensions must be kept within certain limits so to allow a sufficiently fine mesh that is able to capture the flow with enough accuracy.

The solver used in this domain is the one described in previous chapters for FSI problems. This allows to simulate the initiation, erosion and deposition processes within a single framework. In this sense, it must be remarked that all the fluidized phase must be contained within this domain. Failing to do so would cause loss of information since the Ω_{topo} lacks both the necessary resolution and solving strategy to save the mobilized mass. Unfortunately, resizing the dimensions of the calculation domain as time advances would be computationally expensive. For this reason, the initial guess of the size must be as accurate as possible to avoid loss of data while maintaining its size small enough to reduce computation times.

5.4.2 Frame of Reference

Inertial Frame of Reference

It is worth noting that in most cases the frame of reference cannot be changed without modifying the equations to be solved. As an example, if everything was calculated using an Eulerian solver at the *Calculation Domain* Ω_{calc} and this mesh had a variable velocity to follow the debris flow, the velocities should be explicitly added to the system of equations, e.g. in the convective term.

However the use of particles to store all the information allows the strategy to remain in an inertial, static frame of reference related to the original coordinate system. Since the calculation mesh linked to Ω_{calc} does not store information but is only used to provide a discretization to solve the equations, it can be placed anywhere in the domain. This

means that it does not have a velocity nor acceleration and therefore the equations remain unaltered.

Together with the lack of a convective term, this is probably one of the main advantages of the use of particles. Since the frame of reference never changes, it is not only possible to move the mesh, but also using a different mesh or even a completely different discretization method while keeping all the data unaltered. This task would be infeasible using particle-less methods.

Particle Management

The PFEM2 uses particles to convect all the variables, from velocities to stresses. As the mobilized mass flows down the slope, the calculation domain must be moved in order to keep the flow within the boundaries. Since the solid particles that have not been removed from the soil remain in their original position, after some time steps they will fall out of the calculation domain, as shown in Figure 5.11 (hatched area). Since this data is not recovered by the Topographical Domain, information will be lost as the particles are deleted to save storage. In the same way, some areas of the calculation domain are left without particles due to the movement. The particle-less elements will have to gather information from the Topographic Domain in order to reseed the particles accordingly.

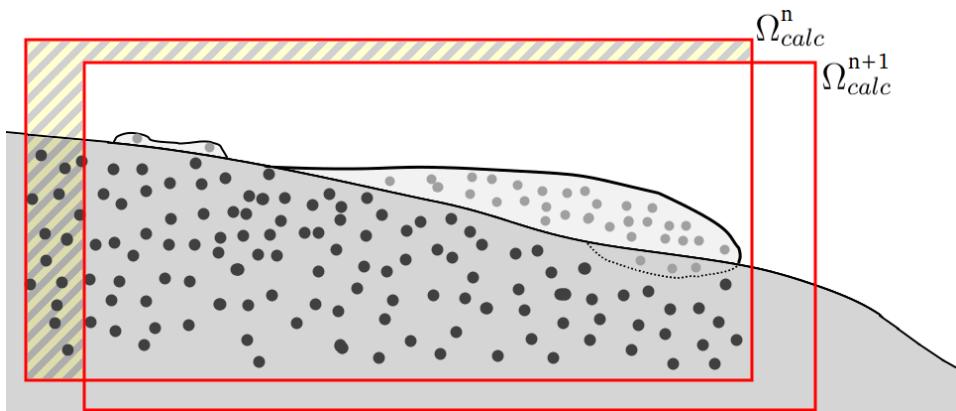


Figure 5.11: Calculation Domain displacement and particle distribution

It must be noted that since the particles and Ω_{calc} are independent from each other, eras-

ing the particles outside Ω_{calc} is not mandatory, but only a tool to reduce computational costs. In fact, if the problem was not memory bound (limited by the required memory), it would be possible to maintain all particles seeded in the whole slope throughout the whole simulation. This way, only the ones inside Ω_{calc} would be used to perform calculations, while the others would only store information until becoming active. Moreover, a Topographic Mesh would not be necessary in this case.

5.4.3 Complete Algorithm

Using the tools described in the previous sections and chapters, it is possible to assemble an algorithm capable of solving debris flow from initiation to deposition. The limiting factor is the Calculation Domain, which has to be sufficiently large to contain all the mobilized mass but with a fine mesh so to be accurate.

Having chosen a correct discretization for Ω_{calc} and having run a static simulation of the whole slope in the Ω_{topo} , the dynamic simulation can start. As the first step, all the volume occupied by Ω_{calc} is seeded with particles by interpolating data from the Topographic domain. After this first step, the algorithm at each step becomes:

- Step 1)** Convect the particles using Ω_{calc} streamlines
- Step 2)** Project information into Ω_{calc} mesh,
- Step 3)** Detect interface elements
- Step 4)** Assemble system of equations with enrichments
- Step 5)** Solve system of equations
- Step 6)** Recover condensed DoFs
- Step 7)** Update particles' velocity, pressure and stresses
- Step 8)** Determine the displacement δ of Ω_{calc} to follow the debris flow
- Step 9)** Move Ω_{calc} the displacement δ
- Step 10)** Erase particles outside Ω_{calc}
- Step 11)** Reseed particles in empty elements of Ω_{calc} using data from Ω_{topo}

5.5 Test Examples

5.5.1 Granular material collapse experiment

The work presented by Bui et al. (2008) provides an excellent example to test two-dimensional solvers for non-cohesive granular materials. By using small aluminium rods (cylinders) instead of tridimensional particles such as sand, the experiment can be supposed as a 2D geometry, without any simplifications. The violent triggering mechanism of this problem translates into an initial failure zone that is simple to define. The total deformations are large enough to ignore the initial elastic deformation, thus allowing to simulate the evolution of the flow using solely a non-Newtonian rheology.

The experimental properties of the material (at macroscopic scale, considering the rods as the microstructure of the material) are as follows.

$$\rho = 1850 \text{ kg/m}^3 \quad E = 1.5 \text{ MPa}$$

$$\phi = 19.5^\circ \quad \nu = 0.35$$

The initial conditions of the setup can be seen in Figure 5.12. Some particles have been marked in blue to ease the tracking of the deformation, forming a square grid of $2 \times 2\text{cm}$. A background grid of $10 \times 10\text{cm}$ is also plotted for comparison, following the experimental setup. The domain has been discretized with a triangular mesh of size $h = 0.0025\text{m}$, leading to a total of 28000 elements. No-slip boundary conditions are set in all the walls. At time $t = 0\text{s}$ the right wall that is supporting the rods is lifted and the movement begins.

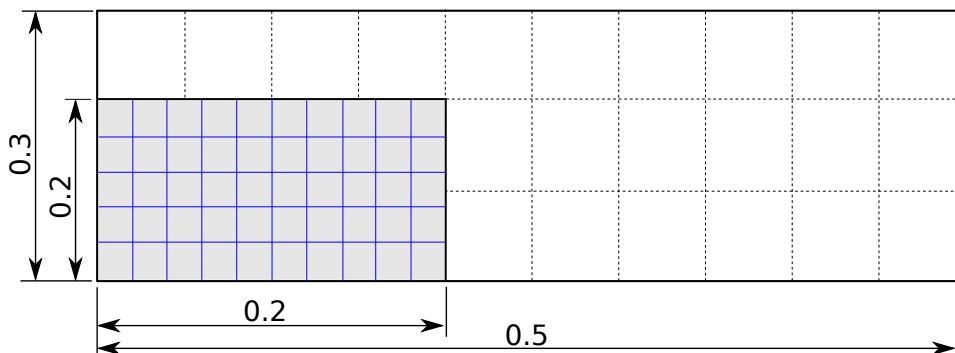


Figure 5.12: Granular material dam break geometry

The granular material rapidly flows to the right until it reaches its final deposition shape at time $t = 0.7\text{s}$. Figure 5.13 shows the shape of the free surface after the movement has stopped. In the experimental results (top), the sheared area is easily recognizable because the grid can no longer be identified. Both the non-sheared area and the shape of the deposition qualitatively match the experimental results. The only difference is a slightly smaller run-out distance in the numerical simulation, which is probably caused by the no-slip condition imposed at all the walls. In reality the rods on the surface experiment a finite shear force rather than an imposed velocity $\mathbf{V} = \mathbf{0}$.

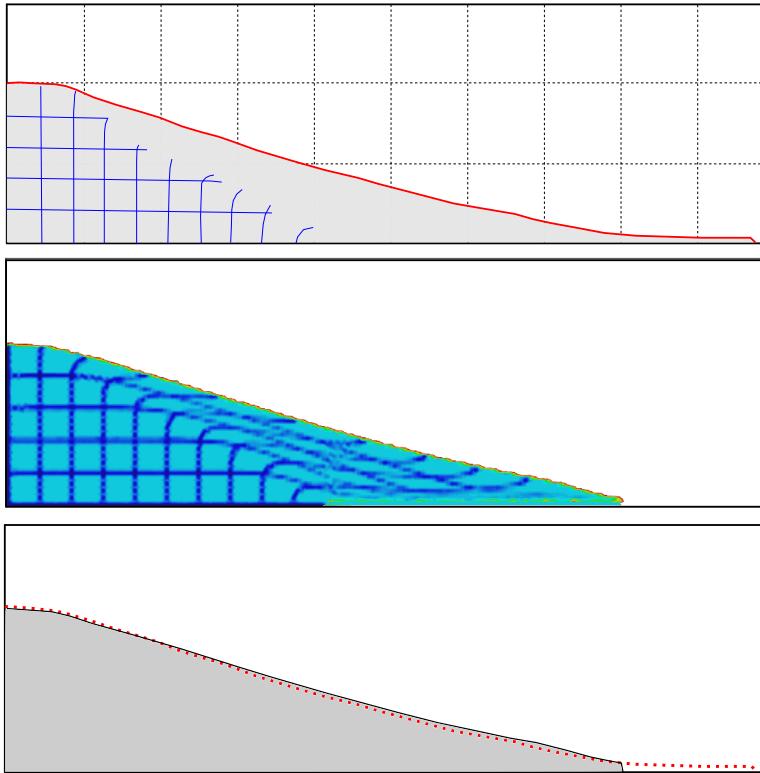


Figure 5.13: Comparison between experimental and numerical results for the 2D granular dam break at $t=0.7\text{s}$. Top: Experimental results. Middle: Numerical results. Bottom: Comparison of the free surface. Experimental results in red and numerical in grey.

5.5.2 Small scale debris Flow with erosion

The use of a solid model with a proper constitutive model circumvents the necessity for an erosion model with special erosion parameters. Two different material properties

are assigned; while the rocks have a high cohesion $c_{rocks} = 50KPa$, the soil bed has only $c_{soil} = 10Kpa$. Setting the same properties would cause the rocks to be destroyed before eroding a significant depth of the soil bed. In this example the materials are not considered porous and therefore only moving mesh strategy combined with the FSI model of Chapter 4 are used. The objective of this example is only testing the robustness of the two strategies working together.

The topographic mesh of the domain can be seen in Figure 5.14, a slope of 30m length by 10 m height. Since the numerical results are not compared with experimental data, the exact geometry is not provided in this document. The whole slope is simulated to simplify setting the boundary conditions, only fixed in the bottom wall of the domain. On the other hand, large elements are used in regions of no interest to reduce computational cost. In Figure 5.15 the statical pressure contour is represented. In the same picture the initial position of the calculation domain is plotted in red, a rectangular box of dimensions 3×1 .

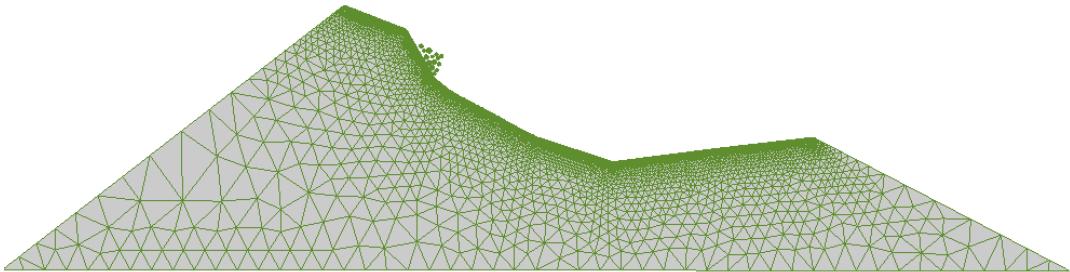


Figure 5.14: Topological Domain

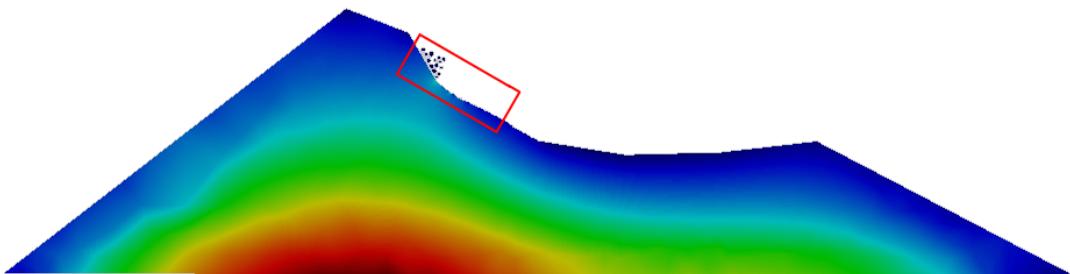


Figure 5.15: Ω_{topo} pressure and initial position of Ω_{calc}

The time evolution of the debris flow can be seen in Figure 5.16. Erosion provides a large mass entrainment in the third snapshot, while in the last picture it is visible the rock front with the fluid rich tail.

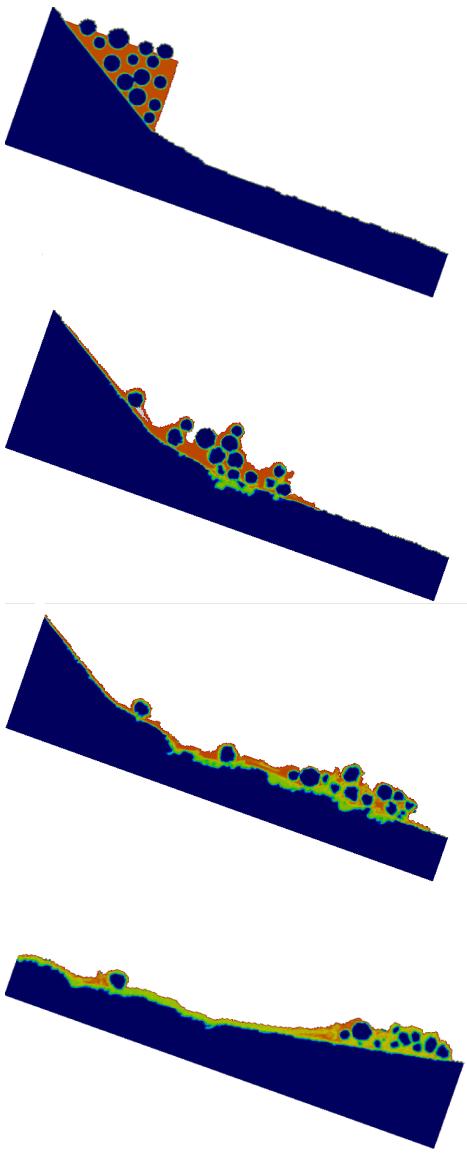


Figure 5.16: Debris flow evolution. Colour Description: Orange: Water, Blue: Solid

Remark: It must be noted that, despite this example showed the potential of the moving mesh strategy, it does not represent accurately a real debris flow. In this sense, the experiments performed by Iverson et al. (2004) provide representative values of real scale debris flows. In the experiments performed in the large scale USGS chute, the depth of the mobilized mass was only of centimetres, while its length extended for several tens of meters. This contrast in the dimensions would cause two main problems. On one

hand, tracking the moving mass would be impossible with important changes in the slope angle and the problem would be even more complicated in 3D. Also, it would lead to calculation domains with poor refinement in the depth, therefore drastically reducing the utility of the method in this type of problems.

5.5.3 Water flow through a porous dam

The experiments performed by Larese et al. (2013) provide good results to test the accuracy of the water model for porous media. The problem consists on a laboratory scale rockfill dam with an water inflow $Q[l/s]$ on the left side, as seen in Figure 5.17. The goal of the experiments was to test the behaviour of the granular material with different mass flows. To validate the PFEM-2 model the lowest Q evaluated the experiments will be used, of $Q = 25.46l/s$. Only this flow will be used since for higher values the dam suffered damage and therefore the geometry was altered, which cannot be tested in this example since the response of the granular material is not simulated.

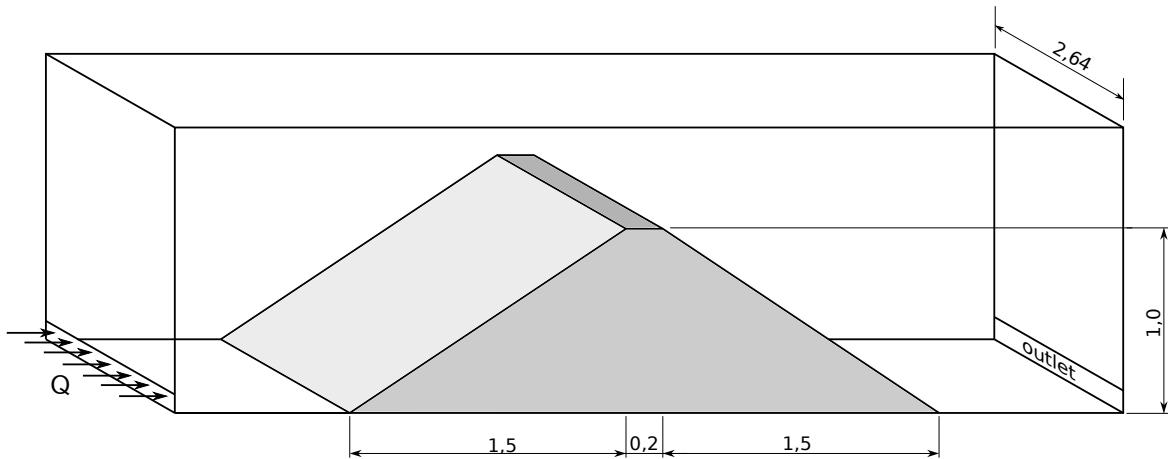


Figure 5.17: Porous dam experimental setup

The computational domain of the problem is plotted in Figure 5.18. A mesh size of $0.02m$ was used, for a total of 9400 elements. Dividing by the width of the domain $2.46m$ and considering the height of the inlet section of $0.1m$, the inlet velocity was set to $V_x = 0.02546/(2.46 \cdot 0.1) = 0.1035m/s$. The water density is set to $\rho = 1000kg/m^3$ and its viscosity to $\mu = 0.0001Pa.s$. The material properties of interest are $n = 0.4$ and the mean particle size $D_{50} = 0.035m$.

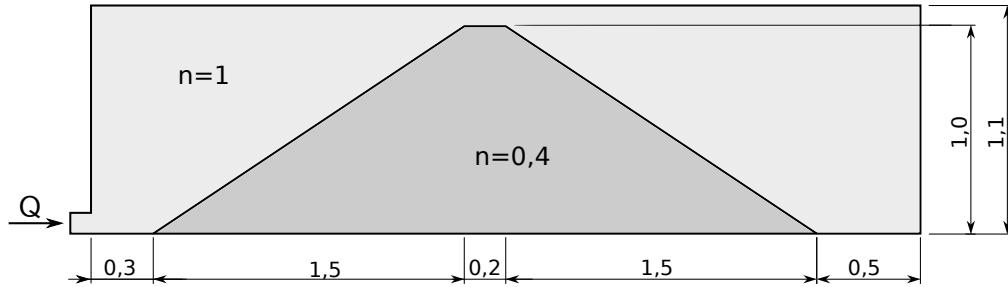


Figure 5.18: Porous dam 2D computational domain

A line of four water pressure sensors were installed in the floor of the experimental setup. The measured values expressed as water column height ($1m = 9800Pa$) can be seen in Figure 5.19. In the same graph the solution obtained with the PFEM-2 algorithm with $\Delta_t = 0.0025s$ are plotted in a continuous line, obtaining a pressure profile slightly below the experimental values. These results are in accordance with Larese et al. (2012b), who found that the Ergun coefficients underestimate drag forces for the material of the dam.

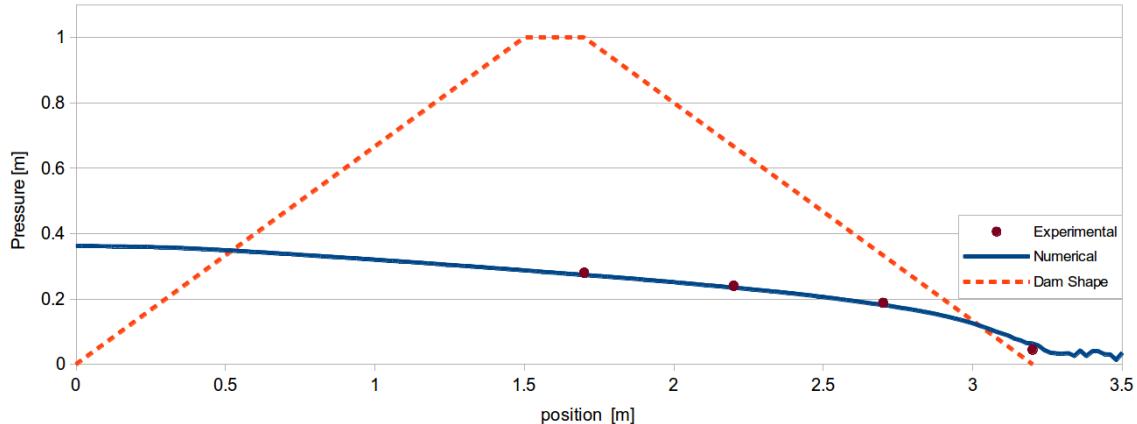


Figure 5.19: Experimental and numerical results for the porous dam

5.5.4 Porous dam damage by water flow

The same geometry of the previous example can be used to test the coupling between the water flow model and the non-Newtonian model to simulate granular materials. In the work by Larese et al. (2012b) , the water inflow was increased in discrete steps, measuring the damage in the dam for each Q . The complete properties of the granular material are expressed in Table 5.1.

Porosity	n	0.4052
Average diameter	D_{50}	0.035
Dry density	ρ_s	1490 kg/m ³
Internal friction angle	ϕ	37° – 42.5°

Table 5.1: Granular material properties

All the material properties are set according to Table 5.1. Given the wide range of values in the friction angle, after a calibration test $\phi = 38^\circ$ was found to obtain the closest results to the experiments. The first numerical test consisted on performing a simulation with an influx of $Q = 25.46 l/s$. According to the experimental results this should cause no damage to the structure and, effectively, the simulations showed no noticeable deformations in the structure of the dam.

The next numerical experiment consisted on using increasing values of the water inflow. The sequence can be found in Figure 5.20. For each tested Q , the value remains stable to ensure that the stationary state is reached.

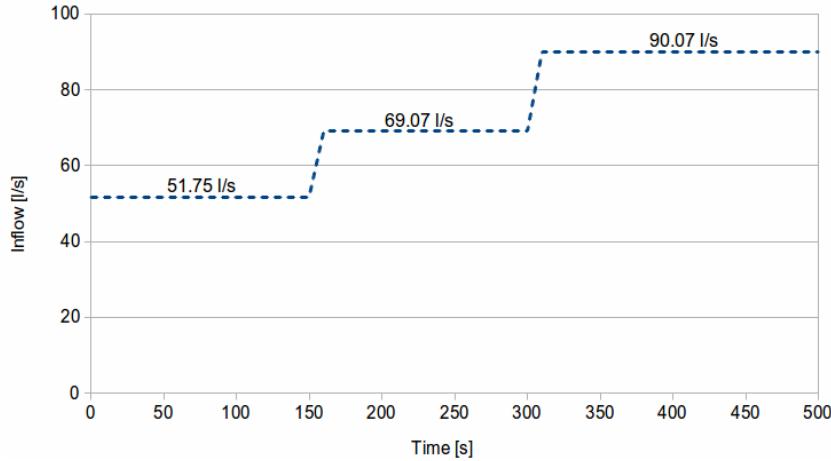


Figure 5.20: Variable water inflow for the dam experiment

Following the work of Larese et al. (2012b), the damage will be measured by the horizontal distance B from the original position of the dam base to the furthest point of displaced material, as shown in Figure 5.21.

Figure 5.22 shows the snapshots of the shape of the dam for the three evaluated water flows. As seen in Table 5.2 the numerical results match the experimental data with a good accuracy.

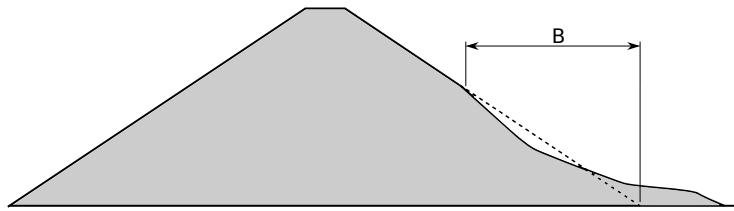


Figure 5.21: Damage measure for the dam experiment

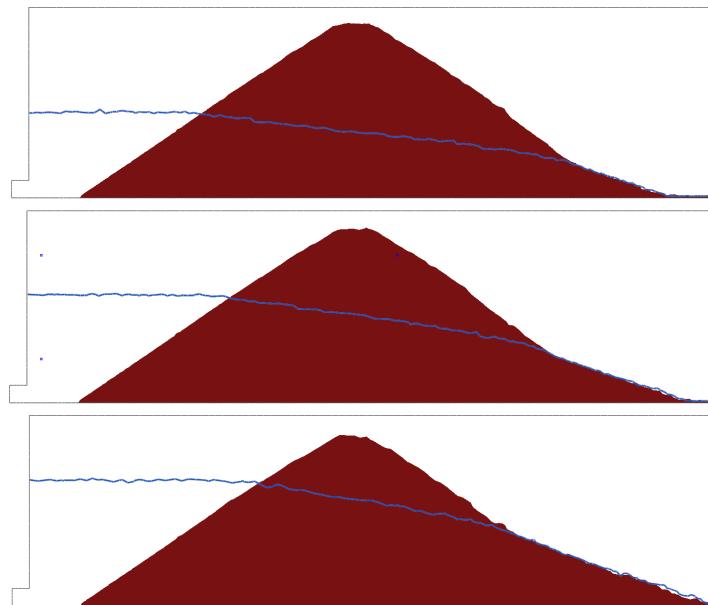


Figure 5.22: Shape of the dam and water free surface for different inflows. Top: 51.75 l/s, Middle: 69.07 l/s, Bottom: 90.68 l/s

Water inflow	Numerical B	Experimental B	error
51.75 l/s	0.75m	0.68	9 %
69.07 l/s	0.81m	1.04	22 %
90.68 l/s	1.60m	1.56	3 %

Table 5.2: Dam damage for different inflows

5.5.5 Vajont landslide

The Vajont landslide was a catastrophic phenomena that occurred in northern Italy in 1963. When the water reservoir of the Vajont Dam was being filled to its maximum capacity, the reduction of the effective strength in the shores of the reservoir caused it to collapse violently, causing a massive landslide reaching speeds of 20 m/s. The violence

of the event was such, that the tsunami wave generated by the moving mass over-topped the dam, reaching the village downstream and killing several people.

The Vajont event is a classic example studied in geomechanics and its simulation is still an open challenge. One of the characteristics that is still not well understood is the unusually high velocity of $20m/s$, which can only be reached if the basal friction is reduced to zero. Thermo-hydro-mechanical theories have been suggested to explain this phenomena (Pinyol and Alonso, 2010). Since the objective of this work is not the creation of a constitutive model that explains the behaviour, the simple granular material presented in this work will be used until a critical strain rate is reached. At this point, the soil at the base will be considered as completely liquefied, with no acting shear forces.

A representative section of the landscape before the event has been chosen for the simulation. The geometry of the problem can be seen in Figure 5.23. The material properties are as follows, where the average particle diameter D_{50} was extremely exaggerated to allow the water to flow between the pores in a time-scale of minutes rather than days. Both the D_{50} and the basal friction angle were calibrated to match the real data.

Porosity	n	0.4
Average diameter	D_{50}	100m
Dry density	ρ_s	1500kg/m ³
Internal friction angle	ϕ	40°
Cohesion	c	10kPa
Basal friction angle	$\phi_{b\ static}$	15°
Dynamic basal friction angle	$\phi_{b\ dyn}$	0°

Table 5.3: Granular material properties for the Vajont problem

Only the area corresponding to the mobilized mass was modelled for the computational model, leaving the interface between the two zones as the boundaries of the domain. The mesh can be seen in Figure 5.24, composed by 9300 triangular elements with a mean size of $h = 20m$. The filling of the dam is performed from the bottom with a inflow of $Q = 150m^3/s$, which translates into approximately 10 minutes for the water free surface to reach 700 mamsl. At this height the landslide began.

Snapshots of the filling can be seen in Figure 5.25 at different time steps. The non-physical mean diameter allows the water to reach a hydrostatic profile with relatively small delay, emulating a slow filling with real porosity parameters.

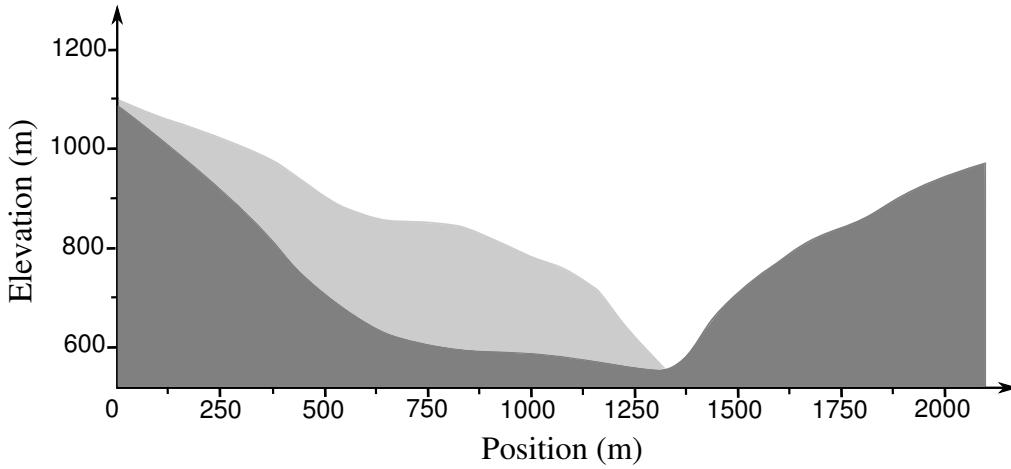


Figure 5.23: Vajont landslide section before the collapse. The light grey area is the mobilized mass. The dark grey area is the soil that remained static.

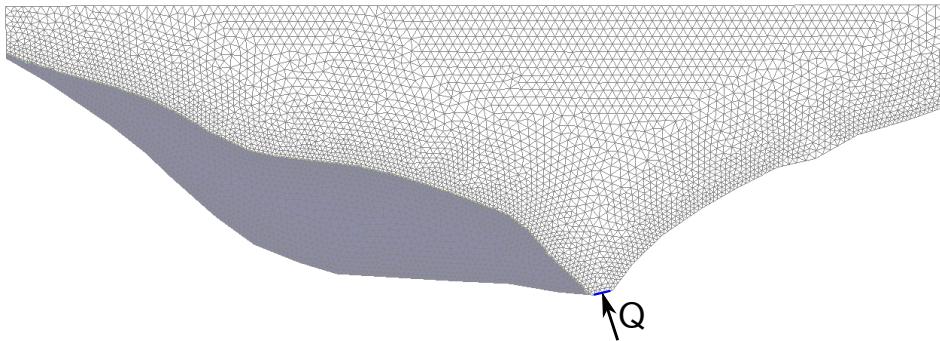


Figure 5.24: Vajont landslide computational mesh

After 480s , the water level reaches 690m^{a.m.s.l} and the velocity of the granular material starts to rise. To simulate the liquefaction of the base, the basal friction angle is reduced from $\phi_b = 15^\circ$ to $\phi_b = 0^\circ$ when the velocity reaches $V = 0.1m/s$. At this moment a low viscosity is assigned to the liquefied elements to avoid numerical instabilities of $\mu = 0.01Pa.s$. In Figure 5.26 the mean velocity of the moving mass is plotted from $t = 485s$ to $t = 525s$. The maximum velocity value and its evolution matches the results of other authors (Zaniboni et al., 2013). Snapshots at different time steps after the movement has begun are shown in Figure 5.27.

The final shape of the deposition is plotted in light grey in Figure 5.28. In the same image the real profile of the surface is plotted in the red dotted line. Excellent agreement

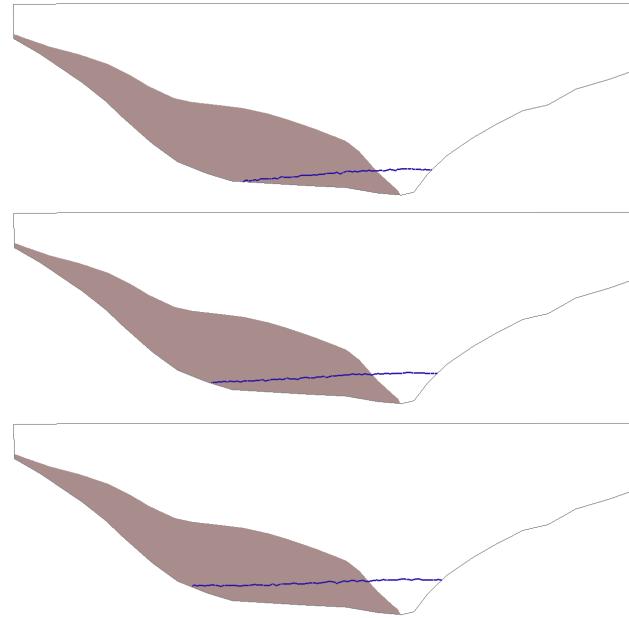


Figure 5.25: Vajont Landslide: Water free surface at time different times. From top to bottom: $t = 100s$, $t = 200s$, $t = 300s$.

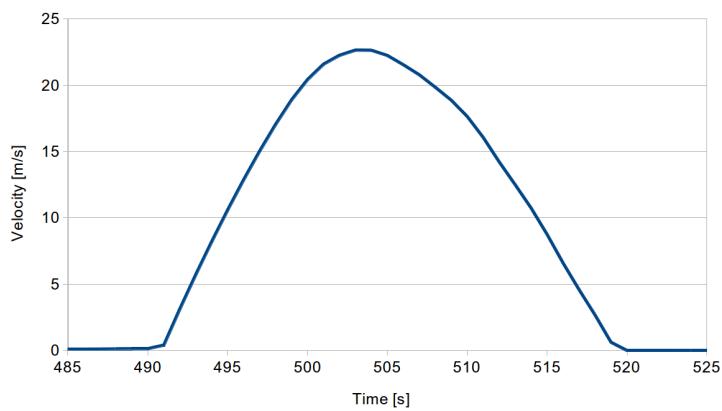


Figure 5.26: Vajont landslide mean velocity at failure time

is shown, proving that the method was capable of simulating the triggering, evolution and final deposition shape of the Vajont landslide event.

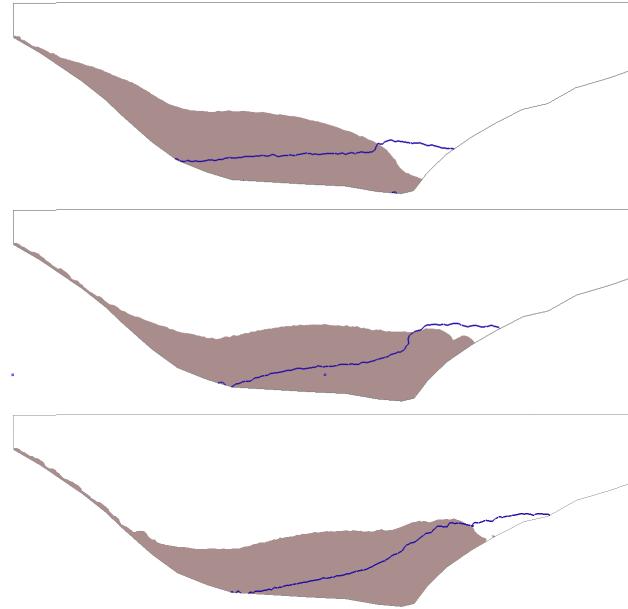


Figure 5.27: Vajont Landslide: Snapshots during the landslide at $t = 500s$, $t = 510s$, $t = 520s$.

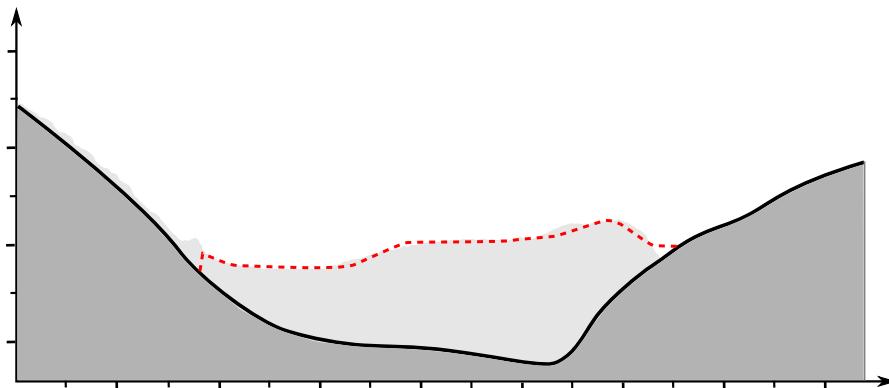


Figure 5.28: Vajont landslide final deposition shape. Light grey: numerical result. Dotted line: Real profile.

5.6 Summary and Conclusions

Accurately simulating debris flows is a challenging task. The multi-phase nature and the large distances they can travel before coming to a stop requires a multi disciplinary approach to tackle this problem. In this chapter the solvers developed in previous chapters were coupled with new numerical tools in order to create a solver suited to simulate this phenomenon. The objective of the developed model is to solve the coupled

problem of a porous granular material with inter-granular water flow that moves across distances much larger than the computational domain.

In the first section, a non-Newtonian model was presented to simulate the rheology of granular materials. The advantages of this model is the lower computational cost compared to the solid solver. On the other hand, using only fluids in the simulation implies that there will always be movement in the material, so the parts that should be in equilibrium will actually be slowly creeping. However, since the objective of this section is the simulation of large deformation problems such as debris flows, this error is not considered an important issue.

A moving domain method was presented in Section 5.2 to tackle the problem of large displacements. Since debris flows are contained in relatively low volumes compared to the total covered distance, using a non fixed mesh that tracks the moving mass allows for fine meshes in the place of interest, while not wasting resources in "inactive" areas. Information on the upcoming or upwind terrain is provided by a larger, background mesh that defines the complete slope.

Section 5.3 introduced a modified fractional step strategy for the simulation of water flow through porous media. The method is based on the FS solver developed in Chapter 3, but modifying the equations and convective terms in order to solve the new problem.

In the last section several numerical experiments were performed to test the different tools developed in this chapter. The results showed a good accuracy of the method and proved that the developed tools can be merged together to create a complete strategy capable of solving landslides in a different approach.

Unfortunately, numerical experiments of debris flows simulations could not be presented in this chapter. The relatively shallow phenomena compared to its total length proved to be a complication for the tracking mesh strategy, leading it to cut the "tail" of the debris flow and therefore destroying the physics of the problem. On the other hand, extending the length of the calculation domain also caused problems. Due to the larger size, the numbers of elements in the thickness of the flow had to be reduced, leading to loss of accuracy, mass losses and non-physical interaction between the air flow and the granular mass. Based on these observations, the current implementation of the PFEM-2 is not capable of solving debris flows problems and further modifications are required.

Chapter 6

Conclusions and future work

A tool to simulate monolithically multi-fluids and Fluid-structure interaction problems was developed. The strategy is based on the PFEM-2 originally formulated by Prof. Idelsohn (Idelsohn et al., 2013) for homogeneous fluids, which allows for fast computations while maintaining accuracy.

The method consists on using a set of Lagrangian particles to transport properties and a fixed mesh where most computations are performed, coupled with a projection algorithm to transfer information from the particles to the mesh . A key feature of the PFEM2 is that, unlike the Material Point Method, the particles do not represent a fixed amount of mass but rather only material points with intensive properties, allowing for strong changes in the mesh sizes and therefore higher accuracy is possible in some areas where desired.

The developed model makes use of several enrichment degrees of freedom (DoFs) to improve both the pressure and velocity fields. Adding extra DoFs allows for a more accurate capturing of the solution where the material properties change abruptly. On the other hand, to avoid resizing the system of equations, the new unknowns are statically condensed, which keeps computational costs reduced.

6.1 Developed strategies

In the first chapter the general strategy of the method to handle multi-materials problems was presented. First the particle convection utility was described with special emphasis on cases in which there are two or more materials. On the other hand, the use of enrichment shape functions and the condensation procedure to enhance the definition of sharp interfaces was explained. Special attention was paid to analyse the source of errors.

In the next two chapters the solution strategies for fluid and FSI problems were described. For the fluid simulation, two different approaches were developed: first a segregated strategy was presented, which is efficient in terms of computational speed. Unfortunately, this strategy is only suited for low viscosity problems. In order to obtain a more general tool, a monolithic strategy was also created. This approach is not bounded to low viscosity problems but, on the downside, requires a longer computational time to obtain the solution. The monolithic formulation was later extended for the simulation of solids in Chapter 4. First a hypo-elastic constitutive model was presented that was later coupled with a plasticity model in order to simulate the behaviour of soils. Finally an FSI solver was created by combining the solid solver with the monolithic fluid solver and the enrichment shape functions.

In the last chapter a strategy to solve debris flows for porous granular materials with inter-granular water flow was presented. This method combines the monolithic and fractional step strategies together with new special tools to enhance the modelling of the phenomena.

6.2 Summary

The enhanced version of the PFEM2 developed in this work provides a new tool to simulate both multi-fluids and FSI problems. The comparison against analytical and experimental results showed the potential of the solver for both the fluid and solid problems analysed. The use of a mixed pressure-velocity formulation for the simulation of solids allowed for a direct coupling algorithm for FSI problems.

From a general point of view, the algorithms developed in this thesis have shown an

excellent flexibility to treat a broad range of problems, from simple single fluid problem to complex coupled problems. The examples of this work have shown the ability of the method to correctly capture the physics involved in all the tested problems. The use of the enrichment shape functions allowed for a good description of material interfaces with excellent accuracy.

Unfortunately, the convergence of the method was below optimal in several cases. By analysing the source of errors it was shown that the projection stage is one of the main causes, reducing the mesh convergence from the expected quadratic rate to only 3/2. The problem lies in the high diffusivity of explicit projection algorithms.

For the particular case of simulating landslides and debris flows, the strategy developed in Chapter 5 presented several tools to tackle the problem. Coupling a granular material model with inter-granular water flow, together with a moving mesh is a relatively simple task due to the Lagrangian nature of the developed method. Numerical experiments showed the ability of the method to predict the initiation of landslides and the subsequent flow. Unfortunately, the attempts to use the moving mesh strategy to simulate debris flow did not yield a viable solution strategy. The extensive length of debris flows (compared to its depth) required compromises in the mesh sizes that ultimately lead to futile results with sever mass loses and non-physical interactions between air and the granular material.

6.3 Future lines of research

The current most important limitation of the PFEM-2 is the non-optimal mesh convergence rate. As a result of this, fine meshes and therefore higher computational power is required to solve problems for a target accuracy. This low convergence rate counters one of the main goals of the PFEM-2, which is minimizing the required computation time. In this line of research, part of the PFEM-2 developing team is already working on alternative projection algorithms. Despite preliminary, the current results with implicit projection schemes show promising outcomes.

Landslide simulations showed great potential, correctly predicting the initiation and post failure displacements. The next step in the development of the method will be extending the plastic constitutive model to three dimensions. This should allow to

simulate landslides following the real soil bed of slopes rather than the two-dimensional simplifications presented in this thesis.

The change of material properties from a solid to a non-Newtonian fluid should be revised. Despite that the transition is feasible and that good results are possible using this methodology, the convergence of the linear iterative solver is heavily impaired when the phase change occurs. For this reason it has not been used in the dam example, but rather relied only on the non-Newtonian model. A careful analysis on the terms should be performed to identify the causes of bad condition number and recover the possibility of using the solid model in the first stages of the deformation without sacrificing computational efficiency.

Finally, the inability of the method to simulate debris flows should be addressed. As the first step, air should no longer be modelled in order to avoid the interaction between the gaseous phase and the granular material at high velocities. On the other hand, mass conservation could be addressed by assigning a fixed mass to each Lagrangian particle, as in the MPM. This way, as long as the total number of particles is conserved, the mass will be conserved too. Finally, in order to track the debris flow as the slope changes, the calculation mesh could be deformed in order to maintain part of the calculation domain both below and above the original ground level, as if it was the mean line of a plate. Deforming of the mesh has already been tested in Chapter 3 with good results. This way the strategy could track the debris flow as long as the slope of the terrain does not change abruptly.

Using these correction measures and improving the projection algorithm should allow the PFEM-2 to simulate correctly debris flows, even with coarse meshes in the thickness of the flow.

Bibliography

- Ashgriz, N. and Mostaghimi, J. (2002). An introduction to computational fluid dynamics. *Fluid flow handbook*.
- Ausas, R., Buscaglia, G., and Idelsohn, S. (2011). A new enrichment space for the treatment of discontinuous pressures in multi-fluid flows. *Int. J. Numer. Meth. Fluids*.
- 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. *Int. J. Numer. Meth. Fluids*, 70(7):829–850.
- Badia, S. (2012). On stabilized finite element methods based on the scott–zhang projector. circumventing the inf–sup condition for the stokes problem. *Computer Methods in Applied Mechanics and Engineering*, 247:65–72.
- Belytschko, T., Liu, W. K., Moran, B., and Elkhodary, K. (2013). *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons.
- Benedetti, L., Cervera, M., and Chiumenti, M. (2015). Stress-accurate mixed fem for soil failure under shallow foundations involving strain localization in plasticity. *Computers and Geotechnics*, 64:32–47.
- Bochev, P. B., Gunzburger, M. D., and Lehoucq, R. B. (2007). On stabilized finite element methods for the stokes problem in the small time step limit. *Int. J. Numer. Meth. Fluids*, 53(4):573–597.
- Brennen, C. (2005). *Fundamentals of multiphase flow*. Cambridge University Press.

- Brumley, D. R., Willcox, M., and Sader, J. E. (2010). Oscillation of cylinders of rectangular cross section immersed in fluid. *Phys. Fluids (1994-present)*, 22(5):052001.
- Bui, H. H., Fukagawa, R., Sako, K., and Ohno, S. (2008). Lagrangian meshfree particles method (sph) for large deformation and failure flows of geomaterial using elastic–plastic soil constitutive model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32(12):1537–1570.
- Cervera, M., Chiumenti, M., and Codina, R. (2010). Mixed stabilized finite element methods in nonlinear solid mechanics: Part i: Formulation. *Computer Methods in Applied Mechanics and Engineering*, 199(37):2559–2570.
- Chang, Y., Hou, T., Merriman, B., and Osher, S. (1996). A level set formulation of eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124(2):449–464.
- Codina, R. (2001). Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics*, 170(1):112–140.
- Codina, R. (2002). Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Comput. Method. Appl. M.*, 191(39):4295–4321.
- Codina, R. and Badia, S. (2006). On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems. *Comput. Method. Appl. M.*, 195(23):2900–2918.
- Coppola, H. (2009). *A finite element model for free surface and two fluid flows on fixed meshes*. PhD thesis, Universitat Politècnica de Catalunya.
- Coppola-Owen, A. and Codina, R. (2005). Improving eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions. *Int. J. Numer. Meth. Fluids*, 49(12):1287–1304.
- Courant, R., Friedrichs, K., and Lewy, H. (1967). On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11(2):215–234.
- Donea, J., Giuliani, S., and Halleux, J. (1982). An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer methods in applied mechanics and engineering*, 33(1):689–723.

- Donea, J. and Huerta, H. (2003). *Finite Element Methods for Flow Problems*. John Wiley and Sons.
- Enright, D., Fedkiw, R., Ferziger, J., and Mitchell, I. (2002). A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183(1):83–116.
- Ergun, S. and Orning, A. A. (1949). Fluid flow through randomly packed columns and fluidized beds. *Industrial & Engineering Chemistry*, 41(6):1179–1184.
- Evans, M., Harlow, F., and Bromberg, E. (1957). The particle-in-cell method for hydrodynamic calculations. Technical report, DTIC Document.
- Eymard, R., Gallouët, T., and Herbin, R. (2000). Finite volume methods. *Handbook of numerical analysis*, 7:713–1018.
- Felippa, C. A. (2004). Introduction to finite element methods. *University of Colorado, Boulder*, <http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d>.
- Franci, A. (2015). *Unified Lagrangian formulation for fluid and solid mechanics, fluid-structure interaction and coupled thermal problems using the PFEM*. PhD thesis, Universitat Politècnica de Catalunya.
- Gil, A. J., Carreño, A. A., Bonet, J., and Hassan, O. (2010). The immersed structural potential method for haemodynamic applications. *Journal of Computational Physics*, 229(22):8613–8641.
- Gimenez, J., Nigro, N., and Idelsohn, S. (2012). Improvements to solve diffusion-dominant problems with pfem-2. *Mecánica Computacional*, 31:137–155.
- Gingold, R. and Monaghan, J. (1977). Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181:375–389.
- Guermond, J.-L. and Quartapelle, L. (2000). A projection fem for variable density incompressible flows. *J. Comput. Phys.*, 165(1):167–188.
- Happel, J. and Brenner, H. (1983). *Low Reynolds number hydrodynamics: with special applications to particulate media*, volume 1. Springer Science & Business Media.

- He, X., Chen, S., and Zhang, R. (1999). A lattice boltzmann scheme for incompressible multiphase flow and its application in simulation of rayleigh–taylor instability. *J. Comput. Phys.*, 152(2):642–663.
- Hirt, C. and Nichols, B. (1981). Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225.
- 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(2):2–2.
- Idelsohn, S. R., Marti, J., Limache, A., and Oñate, E. (2008). Unified lagrangian formulation for elastic solids and incompressible fluids: application to fluid–structure interaction problems via the pfem. *Comput. Method. Appl. M.*, 197(19):1762–1776.
- Idelsohn, S. R., Oñate, E., and Pin, F. D. (2004). The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International journal for numerical methods in engineering*, 61(7):964–989.
- Iverson, R. M., Logan, M., and Denlinger, R. P. (2004). Granular avalanches across irregular three-dimensional terrain: 2. experimental tests. *Journal of Geophysical Research: Earth Surface (2003–2012)*, 109(F1).
- Koshizuka, S., Tamako, H., and Oka, Y. (1995). A particle method for incompressible viscous flow with fluid fragmentation. *Computational Fluid Dynamics Journal*, 4(1):29–46.
- Larese, A., Rossi, R., Oñate, E., and Idelsohn, S. (2012a). A coupled pfem- eulerian approach for the solution of porous fsi problems. *Computational Mechanics*, DOI: 10.1007/s00466-012-0768-9, 50 (6):805–819.
- Larese, A., Rossi, R., Onate, E., and Idelsohn, S. (2012b). A coupled pfem–eulerian approach for the solution of porous fsi problems. *Computational Mechanics*, 50(6):805–819.
- Larese, A., Rossi, R., Oñate, E., Toledo, M., Moran, R., and Campos, H. (2013). Numerical and experimental study of overtopping and failure of rockfill dams. *International Journal of Geomechanics (ASCE)*, pages 10.1061/(ASCE)GM.1943–5622.0000345.

- Maitre, E. (2006). Review of numerical methods for free interfaces. *Les Houches*, 27:31.
- Mier-Torrecilla, M. d., Idelsohn, S., and Oñate, E. (2011). Advances in the simulation of multi-fluid flows with the particle finite element method. application to bubble dynamics. *International Journal for Numerical Methods in Fluids*, 67(11):1516–1539.
- Mora, J. (2012). Do theoretical flops matter for real application performance. In *HPC Advisory Council Spain Workshop*.
- Müller, M., Charypar, D., and Gross, M. (2003). Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 154–159. Eurographics Association.
- Oñate, E. (2013). *Structural Analysis with the Finite Element Method. Linear Statics*, volume 2: Beams, Plates and Shells. Springer.
- Oñate, E., Franci, A., and Carbonell, J. M. (2014a). A particle finite element method for analysis of industrial forming processes. *Computational Mechanics*, 54(1):85–107.
- Oñate, E., Franci, A., and Carbonell, J. M. (2014b). A particle finite element method (pfem) for coupled thermal analysis of quasi and fully incompressible flows and fluid-structure interaction problems. In *Numerical Simulations of Coupled Problems in Engineering*, pages 129–156. Springer.
- Oñate, E. and Manzan, M. (2000). Stabilization techniques for finite element analysis of convection-diffusion problems. *Developments in Heat Transfer*, 7:71–118.
- Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49.
- Papanastasiou, T. C. (1987). Flows of materials with yield. *Journal of Rheology (1978-present)*, 31(5):385–404.
- Patankar, S. (1980). *Numerical heat transfer and fluid flow*. CRC Press.
- Pinyol, N. M. and Alonso, E. E. (2010). Criteria for rapid sliding ii.: Thermo-hydro-mechanical and scale effects in vaiont case. *Engineering Geology*, 114(3):211–227.

- Quecedo, M., Pastor, M., and Herreros, M. (2004). Numerical modelling of impulse wave generated by fast landslides. *International journal for numerical methods in engineering*, 59(12):1633–1656.
- Salazar, F., Irazabal, J., Larese, A., and Oñate, E. (2015). Numerical modelling of landslide-generated waves with the particle finite element method (pfem) and a non-newtonian flow model. *Submitted to the International Journal for Numerical and Analytical Methods in Geomechanics*.
- Salazar, F., Pozo, D., Morán, R., and Oñate, E. (2011). Applications of the particle finite element method in dam engineering. In *II Int. Conference on Particle-based Method - Fundamentals and Applications*, pages 1–11.
- Samet, H. (1990). *The design and analysis of spatial data structures*, volume 85. Addison-Wesley Reading MA.
- Sethian, J. and Smereka, P. (2003). Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics*, 35(1):341.
- Terzaghi, K. (1925). Principles of soil mechanics. *Engineering News-Record*, (95).
- Tezduyar, T. E. (1992). Stabilized finite element formulations for incompressible flow computations. *Adv. Appl. Mech.*, 28:1–44.
- Turek, S. and Hron, J. (2006). *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow*. Springer.
- Ubbink, O. (1997). *Numerical prediction of two fluid systems with sharp interfaces*. PhD thesis, Imperial College of Science, Technology & Medicine.
- Van Eysden, C. A. and Sader, J. E. (2006). Resonant frequencies of a rectangular cantilever beam immersed in a fluid. *J. Appl. Phys.*, 100(11):114916.
- Wang, X. and Liu, W. K. (2004). Extended immersed boundary method using fem and rkpm. *Computer Methods in Applied Mechanics and Engineering*, 193(12):1305–1321.
- 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.

- Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.*, 4(1):389–396.
- Zaniboni, F., Paparo, M. A., and Tinti, S. (2013). The 1963 vajont landslide analysed through numerical modeling. In *International Conference Vajont*.
- Zienkiewicz, O. and Taylor, R. (2000a). *The Finite Element Method: Fluid Dynamics*, volume 3. Butterworth-heinemann.
- Zienkiewicz, O. and Taylor, R. (2000b). *The Finite Element Method: The Basis*, volume 1. Butterworth-heinemann.
- Zienkiewicz, O., Taylor, R., and Zhu, J. Z. (2006). *The Finite Element Method: Its Basic and Fundamentals*, volume 1. Elsevier, 6th edition.