Politecnico di Milano

Department of Civil & Environmental Engineering

POLITECNICO
MILANO 1863

Doctoral Programme in
Structural, Seismic and Geotechnical Engineering
XXXI Cycle

PhD Thesis

A fully explicit Lagrangian Finite Element Method for highly nonlinear Fluid-Structure Interaction problems

---

Simone Meduri

Supervisors:
Prof. Umberto Perego
Prof. Massimiliano Cremonesi

Tutor:
Prof. Claudio di Prisco

Milan - March 2019

A fully explicit Lagrangian Finite Element Method for highly nonlinear Fluid-Structure Interaction problems

A Thesis
Presented to
The Academic Faculty

by

Simone Meduri

In Partial Fullfilment
of the Requirements for the Degree of
Doctor of Philosophy

in
Structural, Seismic and Geotechnical Engineering

March 2019

A fully explicit Lagrangian Finite Element Method for highly nonlinear Fluid-Structure Interaction problems

PhD thesis by Simone Meduri
Supervisors:
Prof. Umberto Perego
Prof. Massimiliano Cremonesi

Doctoral School in Structural, Seismic and Geotechnical Engineering
Department of Civil & Environmental Engineering
Politecnico di Milano
XXXI Cycle - 2015-2018

Acknowledgement

I would like to express my heartfelt gratitude to my supervisors Prof. Umberto Perego and Prof. Massimiliano Cremonesi. I remind now the first meeting when they described me the research project and basically destroyed my previous doubts about starting the PhD. Well, three years later, I sincerely thank them for that meeting, because they just made all this happen. Their constant guide, patience and support have been the engine of this process, an example and a lesson that taught me a lot not only on the professional side, but also as a person.

A special thanks goes to Dr. Victor Oancea, Dr. Vladimir Belsky and Dr. Albert Kurkchubasche for their support during my experience abroad at DS Simulia in Providence, USA (and also after that!). I cannot find the proper words to describe how surprised and honored I was to see all their sincere involvement to help me, not only professionally, despite I was only one of the many interns that a huge company like Simulia hosts every year. I can say that in a huge company I felt welcome as in a small family.

Moreover, in Providence I finally met Dr. Omar Bettinotti, who actually had already started to guide me, even before we knew each other. Indeed, it was because of his suggestion to a common friend of ours that I discovered the possibility of a Master thesis on the PFEM, the starting point of this story. Thank you Omar, because that was the first of a long list of important advice you gave me since then.

I also would like to thank all the other people that made my US experience unique, especially Marta, Sasank, Pushkaraj and Enric.

These three years would not have been the same without the amazing group of friends who shared the daily PhD-life in Politecnico, with all its joys and difficulties, just standing side by side and supporting each other. Among them, a special thanks goes to Pat, Gio, Gian, Andrew, Mary, Marty and Luca. You guys are simply my second family here in Milan.

I also would like to thank my friends-since-forever Rispo, Enes, Toso. It's true that we all live in different cities (even continents) now, but every time we meet we just need 5 minutes to feel as close as if we were still in

the same high school class. I'm sure that this will never change.

The greatest thanks goes to my parents and sister. Thank you for being my constant landmark, my lighthouse. No matter what happens and how distant we are, you are always able to make me feel you are right there on my side to support me, guide me and protect me. This thesis is dedicated to you.

Abstract

The numerical simulation of Fluid-Structure Interaction (FSI) problems is a topic of great relevance because of the wide range of applications in many engineering fields. In this thesis, a partitioned fully explicit and fully Lagrangian Finite Element Method (FEM) for FSI problems is presented. A novel explicit version of the Lagrangian Particle Finite Element Method (PFEM) is employed for the fluid modelling. A distinctive feature of the proposed FSI strategy is that the solid domain is modelled using the explicit FEM of the commercial software SIMULIA Abaqus/Explicit from Dassault Systèmes. This allows to perform simulations with an advanced description on the structural domain, including advanced structural material models and contact interactions. The structure-to-fluid coupling is performed through the SIMULIA built-in Co-Simulation engine and it is based on a technique derived by the Domain Decomposition methods. The method ensures strong coupling and stability of the partitioned solver, retaining at the same time an overall system of fully decoupled explicit equations. Moreover, it allows for the use of different time integration steps and non-conforming meshes in the two subdomains. The fully explicit nature of the coupled solver is appealing for large-scale engineering problems characterized by fast dynamics or high degree of non-linearity. The fully Lagrangian description is particularly effective in the simulation of FSI problems with free surface flows and large structural displacements, since the fluid boundaries are automatically defined by the position of the mesh nodes with no need for interface tracking algorithms.

A novel technique is proposed to simplify the imposition in Lagrangian methods of non-homogeneous boundary conditions which are of practical interest in several engineering applications, e.g., inflow/outflow conditions, fluid slip at boundary walls and symmetry surfaces. The method is based on a mixed Lagrangian-Eulerian description, which introduces fixed Eulerian nodes only on the boundaries where non homogeneous conditions have to be applied, leading to a simple and computationally convenient implementation.

A novel efficient runtime mesh smoothing algorithm for explicit Lagrangian PFEM simulations is proposed. The conditional stability of explicit time integration schemes requires the use of small time increments, proportional to the size of the element in the mesh with the worst geometrical quality. On the other hand, in the 3D framework the Delaunay tessellation employed in the PFEM loses some of its optimality properties holding in 2D, so that badly shaped tetrahedra are frequently added in the triangulation. This leads to unacceptably small stable time step size for explicit solvers. The novel mesh smoothing technique is able to correct overly distorted elements at an acceptable computational cost, so that it can be applied runtime in the frequent remeshing framework of the PFEM. More in general, it could be conveniently applied to regularize the mesh and improve the solution of other Lagrangian methods. This is achieved exploiting an elastic analogy that allows for the use of the same explicit and parallelizable architecture of the fluid solver.

After an extensive validation of the proposed PFEM-FEM FSI approach against analytical, experimental and numerical results presented in the literature, the real engineering application of the automotive airbag deployment is addressed, showing the great potentialities of the fully explicit and Lagrangian approach in this class of challenging industrial problems.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivations

Fluid-Structure Interaction (FSI) problems is a general definition for all the phenomena involving some deformable structures interacting with a surrounding fluid. A wide class of engineering problems, ranging from civil to naval and offshore engineering, from biomedical to aerospace industries, are concerned with FSI [10–15].

In civil engineering, examples of FSI applications are the interaction of a dam with the water in its reservoir or the sloshing of a fluid inside a tank, for instance caused by seismic excitation, the dynamic load of tsunami-like waves on coastal structures or offshore platforms (Figure 1.1-a,b). Moreover, the evaluation of wind excitations on bridges or tall buildings is getting increasingly important as the design is moving towards the use of new materials and/or complex and slender structures (Figure 1.1-c).

Biomedical engineering FSI applications involve the interaction between air and lungs, or the blood flow in the cardiovascular system, possibly involving artificial blood pumps, stents or valves, with complex models which need to be patient-specific (Figure 1.1-d).

Many other FSI examples are related to mechanical engineering applications, for example the study of wind turbines, the vibrations of aircraft wings, or their impact with water in the case of water-landing, the hydroplaning phenomenon which reduces the safety performances of a vehicle tyre when rolling on a wet road, or the flow around a ship hull (Figure 1.1-e). Another challenging application that will be addressed in this work is the automotive airbag deployment, characterized by a complex interaction between the gas inside the chamber and the airbag fabric, as well as the airbag interaction with the vehicle passenger (Figure 1.1-f).

Despite the strong research interest in the FSI in the last decades, the majority of these applications remain a challenging topic because of the

complexity of the involved phenomena: analytical approaches are limited by simplified considerations, while the experimental testing might be costly, time consuming and in some cases even unfeasible. This justifies the continuous search for efficient numerical methods for FSI simulations: the possibility to have a robust and accurate numerical tool in the designing process is of major importance because it can improve the final result in terms of safety, reliability, durability, efficiency and cost.

1.2 Main features of the present approach

There are many different approaches to FSI problems in the literature, which can have advantages and drawbacks according to the peculiarities of the considered application. In this work an Explicit Partitioned Lagrangian Finite Element Method has been developed for FSI problems. An explicit version of the Lagrangian Particle Finite Element Method (PFEM) is used for the fluid side, while the commercial software Abaqus/Explicit, based on standard Lagrangian Finite Element Method (FEM), is used for the structural side. Let us now provide a brief overview on the main peculiarities of such a method, that will be described in details in the following chapters.

Lagrangian description In continuum mechanics, the description of motion can be set in the Eulerian or Lagrangian frameworks. In the former, the observer studies the motion focusing on a fixed control volume through which the continuum moves. In the latter, the observer describes the motion following the material points of the continuum. In the Finite Element context, the choice of one of these kinematics frameworks leads to advantages and drawbacks that will be described in details in Chapter 2 depending on the considered case. The main consequence is that in Eulerian FEM the computational grid discretizes the control volume and it is fixed in time, while in Lagrangian FEM the nodes of the mesh are "attached" to the material points following their motion. The Lagrangian approaches are typical of structural mechanics, while in fluid mechanics the Eulerian approach is usually preferred, because a Lagrangian mesh, which follows the fluid particles motion quickly undergoes unacceptable distortion. However, the Lagrangian description can be very effective in FSI problems involving free surface flows, breaking waves or with structures undergoing large displacements, because the fluid boundaries are naturally defined by the position of the mesh nodes, with no need for interface tracking algorithms.

The Particle Finite Element Method (PFEM) employed in the present work for fluid modelling is a Lagrangian method which exploits the develop-

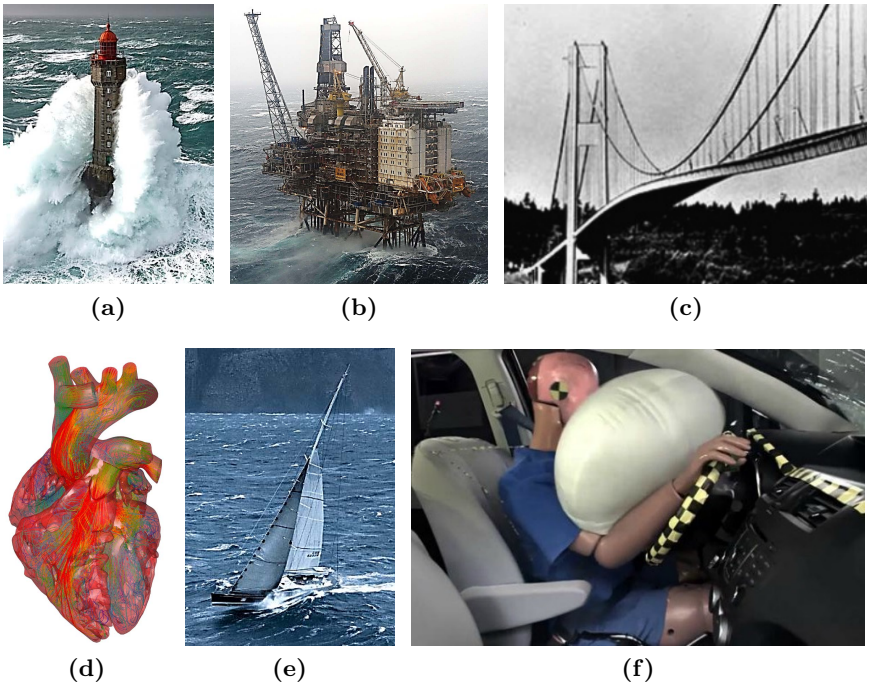


Figure 1.1: FSI problems are relevant in a wide range of engineering applications. (a-b)Wave loads on coastal structures and off-shore platforms. (c) Wind excitation of slender structures: the well-known case of Takoma bridge (USA), which collapsed in 1940. (d) Computational FSI model of a human hearth. (e) Interaction of wind and waves with a sail boat. (e) Crash test: interaction of the airbag with a human dummy.

ment in the past decades of fast and effective triangulation algorithms, such as the Delaunay tessellation. Therefore, a Lagrangian description can be employed, overcoming the problem of mesh distortion with the generation of new meshes every time the current one is too distorted. Consequently, the present coupled FSI method has a fully Lagrangian description both on the fluid and structural sides.

Partitioned FSI solver The computational approaches to FSI can be in general divided into monolithic and partitioned approaches. In the monolithic schemes the governing equations for the structure and the fluid are solved in a unique system. On the contrary partitioned schemes solve separately the governing equations for the fluid and structure subdomains, coupling them together through the exchange of interface conditions at the fluid-structure interface.

The main advantage of the monolithic approach is the preservation of accuracy and stability because the exchange of interface conditions is implicitly included in the solution of the global system of equations. On the other hand, it requires the development of a specific software, not allowing the reuse of existing ones for the fluid and the structure. Moreover, for real engineering problems, it must be underlined that the unified solution of a large system of equations coming from two different physical domains leads to numerical difficulties that can limit its applications.

Partitioned procedures can overcome these limits, since the fluid and structure domains are solved separately and then coupled via interface conditions. This allows for the reuse of existing codes for the physical subdomains (code modularity), solving systems of equations of smaller dimension and better conditioned than in the case of monolithic schemes. However, depending on the employed coupling scheme which may asynchronously enforce the interface conditions, the partitioned approaches may suffer of lack of accuracy, non-optimal convergence rate and numerical instabilities.

In the present work, a partitioned approach based on the Gravouil and Combescure (GC) algorithm is employed. This algorithm preserves the stability of the solvers of each sub-problem and it ensures strong coupling betweeen fluid and structure through the synchronous imposition of interface conditions, leading to a robust and accurate coupled solver. Moreover, the partitioned nature has allowed the choice of coupling with the commercial software Abaqus/Explicit for the structural domain, allowing to introduce all its advanced features in the analyses, such as the wide library of material constitutive relations and finite elements and the possibility to include crack propagation and contact interactions.

Explicit Time Integration One of the novel aspects of the present approach is represented by the use of an explicit time integration both on the fluid and structural subdomains. There are many advantages in the adoption of an explicit dynamics approach. First of all, the solution of the time step problem does not require iterations nor presents convergence problems; secondly the computational burden increases more slowly with the number of degrees of freedom than with implicit solvers. Moreover, it is straightforwardly parallelizable, easily providing high speed-up on multicore platforms. On the other hand, explicit time integration is known to be only conditionally stable, therefore requiring very small time steps. However, there are several large scale engineering applications where the time step is intrinsically small, for example because of a fast dynamics or high level of non-linearity. In such applications, the fully explicit solver presented in this work can be an appealing choice.

The presented main features of the method become particularly important for the industrial application that will be here addressed, namely the deployment of automotive airbags. This phenomenon is characterized by a very fast dynamics (only few tenths of milliseconds for the full deployment) with complex processes of interaction between the gas inside the chamber, the airbag fabric material and the vehicle passengers, which lead to high level of non-linearity. These things considered, in this application there is no doubt that an explicit solver is preferable. At the same time, the large changes in the structural configurations of the airbag from the empty folded initial position to the fully inflated one can underline the effectiveness of a fully Lagrangian approach. Finally, the choice of a partitioned approach may be advantageous for the possibility to reuse existing advanced software (Abaqus/Explicit), thus including in the analysis advanced modelling features.

1.3 Outline

Besides this Introduction, which wanted to briefly highlight the most relevant features and motivations of the present method, this thesis is structured in the following chapters. Let us remark that some of these chapters have been developed on the basis of the papers reported in brackets, which have been published during the PhD programme.

- Chapter 2 introduces the mathematical setting for the fluid modeling. The Navier-Stokes equations under the hypothesis of weakly compressible fluids are presented describing their space and time discretization, as well as the adopted stabilization strategy [16].

- Chapter 3 presents the general aspects of the explicit version of the Particle Finite Element Method employed in the present work for the fluid modeling under the hypothesis of weakly compressible flow. The 2D version of such code has been developed by the author during his Master thesis at Politecnico di Milano and it has been the starting point of the PhD research project. Several 2D examples are reported as a validation of the method [16].

- Chapter 4 presents the coupling algorithm to address FSI problems, namely the Gravouil and Combescure algorithm, as well as some two-dimensional examples to test the accuracy of the 2D PFEM-FEM coupled approach [17].

- Chapter 5 describes a novel strategy for the implementation of non-homogeneous boundary conditions in Lagrangian approaches. This is based on a mixed formulation which introduces fixed Eulerian nodes at the boundary and allows to naturally simulate relevant situations such as: inlet/outlet cross sections, fluid slip at boundary walls and symmetry conditions [18].

- Chapter 6 presents the generalization of the explicit PFEM approach to the 3D framework, where the management of the mesh distortion becomes a major issue. This has led to the development of a novel mesh smoothing technique. Experimental and numerical tests taken from the literature are used for the validation of the 3D PFEM-FEM FSI approach [19].

- Chapter 7 presents the application of the proposed method on a real industrial application, namely the process of deployment of vehicles airbags and their interaction with obstacles which simulate the presence of human passengers.

- Chapter 8 is dedicated to an overall summary of the work, its conclusions and future developments

2 Mathematical modelling of fluid

2.1 The Navier Stokes Equations

The Navier-Stokes Equations (NSE) are a system of partial differential equations describing the motion of a viscous fluid. In the most general form, they represent the mathematical formulation of three physical principles which govern the fluid behaviour, namely the momentum conservation principle, the mass conservation principle and the energy conservation principle. In particular, in the present work only isothermal conditions are addressed, so that only the first two principles are employed to describe the motion of the fluid.

Let us consider a fluid domain $\Omega_f^t \subset \mathbb{R}^d$ moving in the time interval $[0, T]$, being $d = 2, 3$ the space dimension. In the initial (and reference) configuration Ω_f^0, the position of any material point is indicated with $\mathbf{X}$. The corresponding position of the same material point in the current configuration Ω_f^t at time t is defined as $\mathbf{x} = \chi(\mathbf{X}, t)$, where χ represents the mapping from the initial to the current configuration, i.e. the motion [(1)].

Introducing the fluid velocity $\mathbf{v}_f = \mathbf{v}_f(\mathbf{x}, t)$, the Cauchy stress tensor $\boldsymbol{\sigma}_f = \boldsymbol{\sigma}_f(\mathbf{x}, t)$, the density $\rho_f(\mathbf{x}, t)$ and the external body forces per unit mass $\mathbf{b}_f = \mathbf{b}_f(\mathbf{x}, t)$, the momentum and mass conservation of the fluid domain Ω_f^t for a compressible fluid are expressed by:

$$\rho_f \frac{D\mathbf{v}_f}{Dt} = \nabla_{\mathbf{x}} \cdot \boldsymbol{\sigma}_f + \rho_f \mathbf{b}_f \quad \text{in } \Omega_f^t \times [0, T] \tag{2.1}$$

$$\frac{Dp_f}{Dt} + K_f \nabla_{\mathbf{x}} \cdot \mathbf{v}_f = 0 \quad \text{in } \Omega_f^t \times [0, T] \tag{2.2}$$

[(1)] In the following, the subscript letter f is used for all the quantities defined on the fluid domain, while the subscript s will be used for the ones defined on the structural domain.

where K_f is the fluid bulk modulus, $\nabla_{\mathbf{x}}$ represents the gradient operator with respect to the current configuration and the derivative $\frac{D\langle\bullet\rangle}{Dt}$ is the total derivative with respect to time. A suitable set of initial and boundary conditions must be provided to obtain a well-posed problem:

$$\mathbf{v}_f(\mathbf{X}, t=0) = \mathbf{v}_f^0 \quad \text{in } \Omega_f^0 \tag{2.3}$$

$$\mathbf{v}_f(\mathbf{x}, t) = \tilde{\mathbf{v}}_f(\mathbf{x}, t) \quad \text{on } \Gamma_{D,f}^t \times [0, T] \tag{2.4}$$

$$\boldsymbol{\sigma}_f \cdot \mathbf{n}_f = \mathbf{h}_f(\mathbf{x}, t) \quad \text{on } \Gamma_{N,f}^t \times [0, T] \tag{2.5}$$

$$\rho_f(\mathbf{X}, t=0) = \rho_f^0 \quad \text{in } \Omega_f^0 \tag{2.6}$$

where $\mathbf{v}_f^0, \tilde{\mathbf{v}}_f, \mathbf{h}_f, \rho_f^0$ are assigned functions, $\mathbf{n}_f$ is the outward normal to the boundary $\Gamma_f^t = \partial\Omega_f^t$, which is divided into two non-overlapping subsets $\Gamma_{D,f}^t$ and $\Gamma_{N,f}^t$, such that $\Gamma_{D,f}^t \cup \Gamma_{N,f}^t = \Gamma_f^t$ and $\Gamma_{D,f}^t \cap \Gamma_{N,f}^t = \emptyset$.

2.1.1 Kinematic description

Depending on which configuration is chosen to impose the Equations (2.1)-(2.2), two different approaches are defined: Eulerian and Lagrangian. In the Eulerian (or spatial) approach the fluid properties and state variables are a function of the current configuration $\mathbf{x}$ and time t within the defined domain Ω_f^t. On the other hand, in the Lagrangian (or material) approach, the fluid properties and state variables are observed following the evolution of the single particles: they are function of the reference configuration $\mathbf{X}$ and time t, so that the equations are written with respect to the reference domain Ω_f^0.

From the Finite Element point of view, in the Eulerian approach the problem is solved on a fixed computational grid (mesh) through which the material moves; on the contrary Lagrangian methods are based on a mesh that moves according to the material velocity.

The Lagrangian formulation is the natural approach for Computational Structural Dynamics, while the Eulerian one is classically used in Computational Fluid Dynamics (CFD). The main reason is related to the large domain deformation which occurs in phenomena involving fluids (e.g. turbolent flows) which cannot be treated in standard Lagrangian approaches without proper techniques to deal with mesh deterioration. It must be underlined that Eulerian methods are the optimal choice for channel flows (e.g. flow in pipes or vessels) or open domains (e.g. flow past an aircraft wing). However, they do not provide any natural way for treating transient dynamics free surface flows or FSI cases with structure undergoing large displacements, where the shape of the computational domain is continu-

ously evolving. In such cases Lagrangian approaches, combined with proper techniques to overcome the mesh distortion, can be an efficient choice.

The formulation of the Navier Stokes Equations (2.1)-(2.2) is influenced by the choice of the kinematic description in the definition of the total derivative. Considering a generic field variable $h = h(\mathbf{x}, t)$, the Eulerian derivation rule for the total derivative leads to:

$$\frac{Dh}{Dt} = \frac{dh}{dt} + \mathbf{v}_f \cdot \nabla_{\mathbf{x}} h \tag{2.7}$$

where the first term is called local derivative, while the second one is the convective term related to the spatial variation of h. The convective term is a source of non-linearity in the Eulerian formulation.

On the contrary, in a Lagrangian description the convective term disappears because the observer follows the material point motion, namely the total time derivative is equal to the local time derivative. In contrast to the Eulerian description, the non-linearity in Lagrangian Navier Stokes Equations is related to the large displacements in the current configuration with respect to the initial reference configuration.

This work focuses on Lagrangian approaches because they are particularly suited for phenomena involving rapid domain evolution. More in details, instead of a standard Total Lagrangian approach (TL), the Updated Lagrangian formulation (UL) is employed. The UL formulation implies that the reference configuration is updated at every time step and the space derivatives and integrals are computed with respect to the spatial coordinates, so that the non-linearity is given by the fact that the equations are integrated over an unknown domain. The TL formulation could be embraced equivalently. More specifically, in the PFEM framework here considered, characterized by a frequent change in the mesh connectivities (Chapter 3), the TL approach would imply an update of the reference configuration at every mesh change.

2.1.2 Weakly compressible fluids

Another major difference in the formulation of the Navier-Stokes Equations is related to assumption of the fluid incompressibility. In many relevant applications of Computational Fluid Dynamics the hypothesis of incompressible fluid is adopted, due to its high volumetric stiffness, leading to a constant density field. Equation (2.2) then is simply given by:

$$\nabla_{\mathbf{x}} \cdot \mathbf{v}_f = 0 \quad \text{in} \quad \Omega_f^t \times [0, T] \tag{2.8}$$

However, it must be underlined that a small amount of compressibility exists in all the cases idealized as incompressible. In many practical circumstances,

considering this small compressibility can be a convenient choice for problems that would be classified as of incompressible flow. Such an assumption is referred to as weakly compressible fluid hypothesis.

The physical difference between incompressible and compressible fluids is in the propagation of dilational waves (or sound waves). In an incompressible medium, the speed of dilational waves tends to infinity, while in compressible bodies the sound waves speed has a finite value. This is reflected also in the nature of the system of equations: hyperbolic-parabolic for the compressible case and elliptic-parabolic for the incompressible one [20].

Compressibility effects can be characterized by the Mach number (Ma) defined as the ratio between the magnitude of the fluid velocity $|\mathbf{v}_f|$ and the speed of sound in the fluid c_f:

$$Ma = \frac{|\mathbf{v}_f|}{c_f} \tag{2.9}$$

being the dilational wave speed c_f related to the fluid bulk modulus K_f by the following definition:

$$K_f = \rho_f c_f^2 \tag{2.10}$$

The incompressible limit is obtained for a Mach number equal to zero. For Mach numbers much smaller than one (i.e. for small values of the fluid velocities or large values of the fluid bulk modulus) the weakly compressible equations asymptotically approach the incompressible Navier-Stokes equations. Consequently, for suitable small Mach numbers, the weakly compressible model well approximates the incompressible limit. This justifies the fact that a compressible solver is used to solve boundary value problems with incompressible fluids [21]. The real benefit of weakly compressible solvers versus incompressible ones is the possibility to avoid the use of a Poisson solver. Indeed, in the incompressible formulation, the continuity equation represents a constraint on velocity, which, combined with the momentum conservation equation, leads to a Poisson-like equation for pressure. This may lead to numerical issues that will be discussed more in details in Section 2.5. Moreover, in the weakly compressible framework, it is possible to use an explicit time integration that has proved to be robust and very fast for complex hydrodynamic problems [22, 23].

In the compressible framework, an equation of state is necessary to link pressure and density. Following [24,25], in this work the equation Tait equation is used [26, 27]. When a fluid is modeled with a Tait's equation, the energy equation becomes decoupled from mass and momentum conservations [25], and, considering an isothermal flow, it can be neglected.

2.2 Lagrangian NSE for weakly compressible fluids

Let us now consider the Updated Lagrangian version of the Navier-Stokes Equations (2.1))-(2.2) for weakly compressible fluids:

$$\rho_f \frac{d\mathbf{v}_f}{dt} = \nabla_{\mathbf{x}} \cdot \boldsymbol{\sigma}_f + \rho_f \mathbf{b}_f \quad \text{in } \Omega_f^t \times [0,T] \tag{2.11}$$

$$\frac{dp_f}{dt} + K_f \nabla_{\mathbf{x}} \cdot \mathbf{v}_f = 0 \quad in\ \Omega_f^t \times [0,T] \tag{2.12}$$

Initial and boundary conditions (2.3)-(2.6) must be provided. The Cauchy stress tensor $\boldsymbol{\sigma}_f$ can be decomposed into isotropic and deviatoric parts:

$$\boldsymbol{\sigma}_f = -p_f \mathbf{I} + \boldsymbol{\tau}_f \tag{2.13}$$

where $p_f = p_f(\mathbf{x}, t)$ is the pressure field, $\mathbf{I}$ is the identity tensor and $\boldsymbol{\tau}_f$ is the deviatoric stress tensor. Furthermore, for Newtonian and non-Newtonian fluids the deviatoric stress $\boldsymbol{\tau}_f$ is generally related to the deviatoric strain rate $\dot{\boldsymbol{\varepsilon}}_f$ by the following constitutive relation:

$$\boldsymbol{\tau}_f = 2\mu_f(\dot{\boldsymbol{\varepsilon}}_f)\dot{\boldsymbol{\varepsilon}}_f \tag{2.14}$$

where $\mu_f(\dot{\boldsymbol{\varepsilon}}_f)$ is the fluid viscosity and $\dot{\boldsymbol{\varepsilon}}_f = \dot{\boldsymbol{\varepsilon}}_f(\mathbf{x}, t)$ is defined as:

$$\dot{\boldsymbol{\varepsilon}}_f = \frac{1}{2}\left(\nabla_{\mathbf{x}}\mathbf{v}_f + \nabla_{\mathbf{x}}\mathbf{v}_f^T\right) - \frac{1}{3}(\nabla_{\mathbf{x}} \cdot \mathbf{v}_f)\mathbf{I} \tag{2.15}$$

Newtonian fluids are defined by a constant viscosity, namely $\mu_f(\dot{\boldsymbol{\varepsilon}}_f) = \mu_f = const.$

2.2.1 Equation of state

In the compressible framework, an equation of state which relates pressure p_f and density ρ_f is necessary to complete the problem definition. In particular, in the present work, barotropic flows without dependence on temperature are considered; consequently, as suggested in [24, 25], a modified Tait equation of state is used. This empirical relation, originally proposed for water, has proved to represent accurately the behavior of a wide variety of other liquids, gases and solids under high pressures [26]. As reported in [28], Tait's equation can be related to Murnaghan's Equation of state, derived in [29, 30] from an integrated linear theory of finite strains for solids under high pressures. In such conditions, the Murnaghan's Equation of state

can be derived assuming a linear dependence of the bulk modulus K_f with respect to the pressure p_f:

$$K_f(p_f) = K_{0,f} + K'_{0,f} p_f \tag{2.16}$$

where $K_{0,f}$ is the bulk modulus of the material at the reference pressure $p_{0,f}$ (e.g. atmospheric pressure) and $K'_{0,f}$ is the slope of the linear dependence. The definition of the bulk modulus (at constant temperature) reads:

$$K_f(p_f) = \rho_f \left(\frac{\partial p_f}{\partial \rho_f} \right)_{T=const} \tag{2.17}$$

Equating the two above formula and integrating the results, one gets to the Murnaghan's Equation of state:

$$p(\rho) = p_0 + \frac{K_{0,f}}{K'_{0,f}} \left[\left(\frac{\rho_f}{\rho_{f,0}} \right)^{K'_{0,f}} - 1 \right] \tag{2.18}$$

which is equivalent to the expression of the Tait's Equation of state. Experimental tests have shown that the linear dependence of the bulk modulus on pressure with a value of $K'_{0,f} = 7$ is accurate enough to reproduce the behaviour of a wide class of solids and fluids in a range of pressure which is much larger than the ones which arise in the applications here considered. Remarkably, $K'_{0,f} = 7$ is the same value commonly suggested by the empirical Tait's Equation for that parameter, which in that framework is sometimes interpreted as equivalent to the specific heat ratio of the fluid [24, 25].

2.3 Space discretization

A standard Galerkin approach has been followed for the space discretization of Equations (2.11)-(2.12). Let S^v denote the space of admissible functions for the velocity variables and S_0^v the corresponding space of admissible functions with homogeneous boundary conditions, while let S^p be the space of admissible functions for density and pressure [31], namely:

$$\begin{aligned} S^v &= \{ \mathbf{v}_f \in \mathbf{H}^1(\Omega_f^t) \,|\, \mathbf{v}_f = \tilde{\mathbf{v}}_f \quad on \quad \Gamma_{D,f}^t \} \\ S_0^v &= \{ \mathbf{v}_f \in \mathbf{H}^1(\Omega_f^t) \,|\, \mathbf{v}_f = \mathbf{0} \quad on \quad \Gamma_{D,f}^t \} \\ S^p &= L^2\left(\Omega_f^t\right) \end{aligned}$$

2.3.1 Momentum conservation

The weak form of the momentum equation (2.11) is obtained by using a vector test function $\mathbf{w}_f \in S_0^v$ and integrating over the domain Ω_f^t:

$$\int_{\Omega_f^t} \mathbf{w}_f \cdot \rho_f \frac{d\mathbf{v}_f}{dt} d\Omega = \int_{\Omega_f^t} \mathbf{w}_f \cdot \left(\nabla_{\mathbf{x}} \cdot \boldsymbol{\sigma}_f + \rho_f \mathbf{b}_f\right) d\Omega + \\ - \int_{\Gamma_{N,f}^t} \mathbf{w}_f \cdot \left((\boldsymbol{\tau}_f - p_f \mathbf{I}) \cdot \mathbf{n} - \mathbf{h_f}\right) d\Gamma \quad \forall\, \mathbf{w}_f \in S_0^v \tag{2.19}$$

Then, applying the Green formula and using the decomposition of the stress tensor (2.13), as well as the constitutive relation (2.14), the following expression can be obtained:

$$\int_{\Omega_f^t} \mathbf{w}_f \cdot \rho_f \frac{d\mathbf{v}_f}{dt} d\Omega = \int_{\Omega_f^t} p_f \left(\nabla_{\mathbf{x}} \cdot \mathbf{w}_f\right) d\Omega + \\ - \int_{\Omega_f^t} 2\mu_f \nabla_{\mathbf{x}} \mathbf{w}_f : \dot{\boldsymbol{\varepsilon}}_f \, d\Omega + \int_{\Omega_f^t} \mathbf{w}_f \cdot \rho_f \mathbf{b}_f d\Omega + \\ + \int_{\Gamma_{N,f}^t} \mathbf{w}_f \cdot \mathbf{h}_f \, d\Gamma \quad \forall\, \mathbf{w}_f \in S_0^v \tag{2.20}$$

Let us introduce a discrete approximation of the spaces S^v, S_0^v, S^p through the spaces $S_h^v \in S^v$, $S_{0,h}^v \in S_0^v$, $S_h^p \in S^p$. A standard isoparametric finite element discretization of the velocity, pressure and density fields leads to the following interpolations:

$$\mathbf{v}_f^h(\mathbf{x}, t) = \mathbf{N}_f^v(\mathbf{x}) \mathbf{V}_f(t) \tag{2.21}$$

$$p_f^h(\mathbf{x}, t) = \mathbf{N}_f^p(\mathbf{x}) \mathbf{P}_f(t) \tag{2.22}$$

$$\rho_f^h(\mathbf{x}, t) = \mathbf{N}_f^\rho(\mathbf{x}) \mathbf{R}_f(t) \tag{2.23}$$

where $\mathbf{N}_f^v$, $\mathbf{N}_f^p$ and $\mathbf{N}_f^\rho$ are the matrix of shape functions for velocity and the vectors of shape functions for pressure and density, respectively; $\mathbf{V}_f$ is the vector of nodal velocities, $\mathbf{P}_f$ is the vector of nodal pressures and $\mathbf{R}_f$ the vector of nodal densities. For incompressible materials, the spaces of interpolation S_h^v and S_h^p cannot be arbitrarily selected, but they need to satisfy the so called LBB (*Ladyzhenskaya-Babuška-Brezzi*) discrete inf-sup condition [32, 33]:

$$\inf_{q^h \in S_h^p} \sup_{\mathbf{v}^h \in S_h^v} \frac{\int_{\Omega_f^t} q^h \nabla_{\mathbf{x}} \cdot \mathbf{v}^h d\Omega}{||\mathbf{v}^h||_{S_h} ||q||_{Q_h}} \geq \beta > 0 \tag{2.24}$$

where β is a positive constant independent from the mesh size h. Furthermore, the peculiarities of the PFEM method, described in Chapter 3, require the choice of linear interpolations both for velocities and pressures (as well as for densities which are directly connected to pressures by the EOS). Therefore, the same linear shape functions for the three unknown fields are used and linear triangles (in 2D) and tetrahedra (in 3D) are considered in the PFEM approach. Equal order of interpolation for velocity and pressure does not satisfy the LBB condition, so that a stabilization technique need to be added, as it will be discussed in Section 2.5.

Substituting (2.21)-(2.23) in (2.20) one obtains:

$$\mathbf{M}_f \frac{d\mathbf{V}_f}{dt} = -\mathbf{K}_{\mu,f}\mathbf{V}_f + \mathbf{D}_f^T\mathbf{P}_f + \mathbf{F}_{ext,f} = -\mathbf{F}_{int,f} + \mathbf{F}_{ext,f} = \mathbf{F}_f \quad (2.25)$$

The mass matrix $\mathbf{M}_f$ is defined as:

$$\mathbf{M}_f = \int_{\Omega_f^t} \rho_f \mathbf{N}_f^{v\ T} \mathbf{N}_f^v d\Omega \quad (2.26)$$

The viscous matrix $\mathbf{K}_{\mu,f}$ and the discrete gradient operator matrix $\mathbf{D}_f$ of the volumetric term are computed as:

$$\mathbf{K}_{\mu,f} = \int_{\Omega_f^t} \mathbf{B}^T \mathbf{d}_{dev} \mathbf{B}\, d\Omega$$

$$\mathbf{D}_f = \int_{\Omega_f^t} \mathbf{B}^T \mathbf{m} \mathbf{N}_f^p\, d\Omega$$

In the previous expressions, the vector $\mathbf{m}$ is defined as $\mathbf{m} = [1\ 1\ 1\ 0\ 0\ 0]^T$, while $\mathbf{B}$ is the matrix containing the shape functions derivatives which can be divided in n_n nodal blocks (e.g. $n_n = 4$ for linear tetrahedra):

$$\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_{n_n}] \quad (2.27)$$

being each nodal block $\mathbf{B}_i$ a (3×6) matrix defined as:

$$\mathbf{B}_i^T = \begin{bmatrix} \frac{\partial N_i^v}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i^v}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i^v}{\partial z} \\ \frac{\partial N_i^v}{\partial y} & \frac{\partial N_i^v}{\partial x} & 0 \\ 0 & \frac{\partial N_i^v}{\partial z} & \frac{\partial N_i^v}{\partial y} \\ \frac{\partial N_i^v}{\partial z} & 0 & \frac{\partial N_i^v}{\partial x} \end{bmatrix} \quad (2.28)$$

The $\mathbf{d}_{dev}$ matrix contains the material viscous properties of the considered fluid. For a Newtonian fluid it is given by:

$$\mathbf{d}_{dev} = \mu_f \begin{bmatrix} 4/3 & 2/3 & 2/3 & 0 & 0 & 0 \\ 2/3 & 4/3 & 2/3 & 0 & 0 & 0 \\ 2/3 & 2/3 & 4/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.29}$$

Finally the vector of external forces $\mathbf{F}_{ext,f}$ is defined as:

$$\mathbf{F}_{ext,f} = \int_{\Omega_f^t} \rho_f \mathbf{N}_f^{vT} \, \mathbf{b}_f \, d\Omega + \int_{\Gamma_{N,f}^t} \mathbf{N}_f^{vT} \, \mathbf{h}_f \, d\Gamma \tag{2.30}$$

It must be underlined that an explicit integration scheme is employed in the present method (see Section 2.4), so that no global matrices are assembled. The viscous term $\mathbf{K}_{\mu,f}\mathbf{V}_f$ and the volumetric one $\mathbf{D}_f^T\mathbf{P}_f$ are computed at the element level and then directly assembled on the global vector of internal forces $\mathbf{F}_{int,f}$. Finally, let us remark that in practice the vector of nodal resultant forces $\mathbf{F}_f$ is directly computed as the difference between the internal and the external ones.

2.3.2 Mass conservation and Equation of State

The weak form of the mass equation (2.12) is obtained by using a test function $q_f \in S^p$ and integrating over the domain Ω_f^t:

$$\int_{\Omega_f^t} q_f \frac{dp_f}{dt} d\Omega + \int_{\Omega_f^t} K_f q_f \left(\nabla_{\mathbf{x}} \cdot \mathbf{v}_f\right) d\Omega = 0 \quad \forall \, q \in \, S^p \tag{2.31}$$

following the Galerkin isoparametric discretization used for the momentum equation and introducing the interpolations (2.21)-(2.23), one can get to:

$$\mathbf{M}_\rho \frac{d\mathbf{P}_f}{dt} + K_f \mathbf{D}_f \mathbf{V}_f = 0 \tag{2.32}$$

where the volume-like matrix $\mathbf{M}_\rho$ is defined as:

$$\mathbf{M}_\rho = \int_{\Omega_f^t} \mathbf{N}_f^{pT} \, \mathbf{N}_f^p \, d\,\Omega \tag{2.33}$$

Alternatively, the mass conservation can be also enforced starting from the following strong expression [22, 23]:

$$\rho_f(\mathbf{x}, t) J(\mathbf{x}, t) = \rho_f(\mathbf{X}, 0) = \rho_0(\mathbf{X}) \tag{2.34}$$

where $J(\mathbf{x},t) = \det \mathbf{F}(\mathbf{x},t)$ is the determinant of the deformation gradient $\mathbf{F}$. Multiplying equation (2.34) by a test function $q_f \in S^p$ and integrating over the domain Ω_f^0, the following weak form is obtained:

$$\int_{\Omega_f^0} q_f \, \rho_f(\mathbf{x},t) J(\mathbf{x},t) \, d\Omega_0 = \int_{\Omega_f^0} q_f \, \rho_0(\mathbf{X}) \, d\Omega_0 \quad \forall \, q \in \, S^p \tag{2.35}$$

Noting that $J(\mathbf{x},t) d\Omega_0 = d\Omega$, the previous equation can be written as:

$$\int_{\Omega_f^t} q_f \, \rho_f(\mathbf{x},t) d\Omega = \int_{\Omega_f^0} q_f \, \rho_0(\mathbf{X}) \, d\Omega_0 \quad \forall \, q \in \, S^p \tag{2.36}$$

Moreover, let us introduce the space discretization (2.21)-(2.23), leading to:

$$\mathbf{M}_\rho \mathbf{R}_f = \mathbf{F}_\rho^0 \tag{2.37}$$

where $\mathbf{F}_\rho^0$ is defined as the product between $\mathbf{M}_\rho$ and $\mathbf{R}_f$ computed on the reference configuration:

$$\mathbf{F}_\rho^0 = \mathbf{M}_\rho^0 \mathbf{R}_f^0 \tag{2.38}$$

The two versions of the mass conservation equation (2.32),(2.37) have been employed in the present work. Equation (2.37) has the advantage of showing no time derivatives, which avoids any further approximation introduced by the time integration scheme, so that it has been in general preferred. Nonetheless, let us underline that in all the numerical simulations where the two equations (2.32),(2.37) have been tested, no significant difference in the results has been reported. Moreover, in Chapter 5 Equation (2.32) has been preferred because it can be easily cast in a mixed Lagrangian-Eulerian formulation adding the convective term to the pressure time derivative.

To conclude the space discretization, the equation of state (2.18) expresses a pointwise relation between nodal pressure and density. Depending on the use of Equation (2.32) or (2.37), i.e. depending on which variable has been computed from the continuity equation (pressure or density), on the *j-th* node one can compute the unknown variable solving for $P_{f,j}$ or $R_{f,j}$ the following relation:

$$P_{f,j} = P_{f,0} + \frac{K_{0,f}}{K'_{0,f}} \left[\left(\frac{R_{f,j}}{R_{f,0}} \right)^{K'_{0,f}} - 1 \right] \tag{2.39}$$

where $P_{f,0}$ and $R_{f,0}$ are the reference values of pressure and density, respectively.

2.4 Explicit time discretization

Let us consider a generic subdivision of the time interval $[0, T]$ in N time steps Δt^n such that $T = \sum_{n=0}^{N-1} \Delta t^{n+1}$. The Central Difference Scheme (CDS) can be derived considering the general Newmark's method with $\beta = 0$ and $\gamma = 1/2$ [34, 35].

Defining the vector of nodal accelerations $\mathbf{A}_f$, at the beginning of the generic time step $\Delta t^{n+1} = t^{n+1} - t^n$ the CDS computes a mid-step velocity $\mathbf{V}_f^{n+1/2}$ starting from the accelerations of the previous time step:

$$\mathbf{V}_f^{n+\frac{1}{2}} = \mathbf{V}_f^n + \frac{1}{2}\mathbf{A}_f^n \Delta t^{n+1} \tag{2.40}$$

These velocities are used to compute the displacements $\mathbf{U}_f$ and update the nodes positions:

$$\mathbf{U}_f^{n+1} = \mathbf{U}_f^n + \mathbf{V}_f^{n+\frac{1}{2}} \Delta t^{n+1} \tag{2.41}$$

Once the configuration at t^{n+1} is known, the mass conservation is enforced to obtain the new nodal densities $\mathbf{R}_f^{n+1}$:

$$\mathbf{M}_\rho^{n+1} \mathbf{R}_f^{n+1} = \mathbf{F}_\rho^0 \tag{2.42}$$

and the equation of state is used to compute the nodal pressures:

$$P_{f,j}^{n+1} = P_{f,0} + \frac{K_{0,f}}{K'_{0,f}} \left[\left(\frac{R_{f,j}^{n+1}}{R_{f,0}} \right)^{K'_{0,f}} - 1 \right] \tag{2.43}$$

Then, with the updated state of the system, the nodal resultant force vector $\mathbf{F}_f^{n+1}$ is assembled and the momentum equation is solved to compute the new accelerations $\mathbf{A}_f^{n+1}$:

$$\mathbf{A}_f^{n+1} = (\mathbf{M}_f^{n+1})^{-1} \mathbf{F}_f^{n+1} \tag{2.44}$$

Finally, if necessary, the velocity vector at the end of the step can be computed

$$\mathbf{V}_f^{n+1} = \mathbf{V}_f^{n+\frac{1}{2}} + \frac{1}{2}\mathbf{A}_f^{n+1} \Delta t^{n+1} \tag{2.45}$$

Remark 1. *If the continuity equation* (2.32) *is employed, Equation* (2.42) *is replaced by an explicit step to compute the new nodal pressure* $\mathbf{P}_f^{n+1}$*:*

$$\mathbf{M}_\rho^n \frac{\mathbf{P}_f^{n+1} - \mathbf{P}_f^n}{\Delta t^{n+1}} + K_f \mathbf{D}_f^n \mathbf{V}_f^n = \mathbf{0} \tag{2.46}$$

while the EOS (2.43) *is inverted to compute the nodal density* $\mathbf{R}_f^{n+1}$*:*

$$R_{f,j}^{n+1} = R_{f,0}\left[\frac{K'_{0,f}}{K_{0,f}}\left(P_{f,j}^{n+1} - P_{f,0}\right) + 1\right]^{\left(1/K'_{0,f}\right)} \tag{2.47}$$

As usual in explicit approaches, the mass matrices $\mathbf{M}_f^{n+1}$ and $\mathbf{M}_\rho^{n+1}$ are lumped obtaining a global system of fully decoupled equations that can be solved explicitly node-by-node, thus no system of equations has to be solved. Explicit solvers are very appealing for non-linear problems which may suffer of numerical issues of convergence in the solution of the system of governing equations. Another advantage is related to the fact that the computational burden of explicit solvers increases more slowly with the number of degrees of freedom than with implicit solvers. Moreover, their system of decoupled equations is straightforwardly parallelizable, easily providing high speed-up on multi-core platforms. On the other hand, the CDS is only conditionally stable and the choice of the time step size is governed by the CFL (Courant, Friedrichs, Lewy) stability condition [36]. Therefore, an adaptive time step is used in the present method and the stable Δt^{n+1} is computed at every step as:

$$\Delta t^{n+1} = C_N \min_e \left(\frac{h_e}{\tilde{c}_e}\right) \tag{2.48}$$

In the previous expression, C_N is a safety parameter, $\tilde{c}_e$ has the dimension of a velocity and it is the maximum between the actual fluid velocity in the *e-th* element and the speed of dilational waves $c_{f,e}$ depending on the fluid bulk modulus K_f and the elemental density (Relation (2.10)). Finally, h_e is a characteristic size of the *e-th* element in the current configuration. In particular, in the present method the radius r_e^{in} of the element incircle (in 2D) or element insphere (in 3D) is adopted, namely $h_e = r_e^{in}$. Therefore, the mesh quality strongly influences the efficiency of an explicit solver. This may be an issue, especially in the 3D framework, as it will be addressed in details in Chapter 6. The stable time step of explicit solvers is generally much smaller than the time step employed in implicit solvers, which are usually unconditionally stable. However, there are many relevant engineering applications involving fast dynamics or high degree of non-linearity, where the time step is intrinsically small for accuracy or convergence reasons. For those cases the adoption of an explicit solver can be an efficient choice.

Finally, let us remark that in all the examples here considered, besides the airbag deployment applications presented in Chapter 7, the speed of dilational waves c_f is much greater than the fluid velocity. In such cases, a

standard procedure to increase the stable time step size in weakly compressible flows is to use a reduced value of c_f by means of the introduction of a slight compressibility, i.e. a smaller bulk modulus K_f. This can be done without affecting the global behaviour of the fluid if the Mach number, computed with the reduced value of c_f, is much smaller than one [24,37]. Consequently, a reduced value of c_f in the examples here presented is employed, depending on the expected fluid velocity in each example. Nevertheless, the reported density variations due to this additional small compressibility are smaller than 0.1%. On the contrary, this strategy is not employed in the airbag deployment simulations of Chapter 7, where however the fluid velocity is the relevant term to determine the stable time step size.

2.5 Pressure stabilization

In the present section let us address the mass equation in the form (2.32), in order to cast the following description in the framework of mixed velocity-pressure formulation which is commonly adopted in the literature of stabilization methods. Nevertheless, all the following considerations have been applied in the present method also to the mass conservation (2.37).

2.5.1 Instability issues

The finite element solution of Navier-Stokes equations for incompressible and weakly-compressible materials can suffer of numerical instabilities which may arise from two sources. The first one is related to the convective term in the momentum equation, which may lead to spurious oscillation of the velocity field, especially for high values of the Reynolds number, and requires proper stabilization procedures [31]. As previously commented, the absence of the convective term is among the advantages of a fully Lagrangian description and this source of instability does not appear in the present method. The second source of instability arises for incompressible materials when the choice of the spaces for the interpolation of velocity and pressure does not fulfill the LBB inf-sup condition (2.24) [33], leading to spurious oscillations in the pressure field.

As already anticipated, the PFEM approach described in Chapter 3 is based on a frequent use of re-triangulation of the domain, to avoid an excessive mesh distortion caused by the Lagrangian description for the fluid flow. To avoid expensive interpolations to be performed at every change of the mesh, all the variables are stored at the nodes which are located to the vertices of the finite elements. Therefore only linear shape functions can be used for velocity, pressure and density. This choice does not satisfy the

LBB inf-sup condition (2.24) and a stabilization technique is needed.

First of all, it must be noted that the condition (2.24) is related, strictly speaking, only to the fully incompressible formulation and the compressible materials suffer of equivalent issues when tending to the incompressible limit, i.e. for high values of the bulk modulus K_f. In the incompressible framework, the continuity equation (2.8) represents a constraint on velocity and combined with momentum conservation leads to a Poisson-like equation for pressure. The study of the conditions which ensure a unique solution of the consequent algebraic system of discretized equations leads to the stability requirement (2.24) [33].

2.5.2 Stabilization methods

Considering the previous remarks on the causes of the stability issues, the basic idea behind several well-known stabilization methods is to slightly modify the continuity equation with additional terms, relaxing the incompressibility constraint. Among them, let us recall the Laplacian pressure stabilization [38], which adds to the weak form of the continuity equation a term proportional to the Laplacian of the pressure field. The Galerkin least squares method [39] adds a least squares form to the Galerkin formulation. The Finite Increment Calculus (FIC) [40,41] adds terms which come from considering the governing equations in a space-time domain of finite incremental size, instead of infinitesimal size as it is usually done.

Finally, let us enter more into the details of the Direct Pressure Stabilization (DPS) [42, 43], which is here related to the adopted stabilization strategy. The DPS stabilizes an equal velocity-pressure interpolation pair through the application to the pressure field of a L^2 projection operator Π onto an interpolation space of lower order $\tilde{S}_h^p$, and through an additional term to the weak form (2.31) of the continuity equation:

$$\begin{aligned}&\int_{\Omega_f^t} q_f^h \frac{dp_f^h}{dt} d\Omega + \int_{\Omega_f^t} K_f q_f^h \left(\nabla_{\mathbf{x}} \cdot \mathbf{v}_f^h\right) d\Omega + \\ &- \int_{\Omega_f} \left(q_f^h - \Pi q_f^h\right)\left(p_f^h - \Pi p_f^h\right) d\Omega = 0 \quad \forall\, q_f^h \in S_h^p \end{aligned} \tag{2.49}$$

The projector operator Π is defined by the following expression:

$$\int_{\Omega_f} q_f^h \left(\Pi p_f^h - p_f^h\right) d\Omega = 0 \quad \forall q_f^h \in S_h^p \tag{2.50}$$

Combining the two last expressions one can find the final form of the stabi-

lizing term added in the mass equation:

$$-\int_{\Omega_f}\left(q_f^h p_f^h - \Pi q_f^h \Pi p_f^h\right) d\Omega \tag{2.51}$$

Let us consider for the sake of simplicity the case of 2D linear triangles. The projector operator applied to the linear pressure by the DPS leads to a discrete pressure field which is element-by-element piece-wise constant:

$$\Pi p_f^h = \tilde{\mathbf{N}}^p \mathbf{P}_f \tag{2.52}$$

being the vector $\tilde{\mathbf{N}}^p$ given by $\tilde{\mathbf{N}}^p = [1/3; 1/3; 1/3]$. Applying the spatial discretization (2.22), (2.52) to (2.51), one can find that the additional term required by the DPS in the continuity equation is computed on each element e as:

$$-\mathbf{S}_{f,e}^{DPS}\mathbf{P}_{f,e} \tag{2.53}$$

where the matrix $\mathbf{S}_{f,e}^{DPS}$ is defined as [42, 44]:

$$\mathbf{S}_{f,e}^{DPS} = \int_{\Omega_{f,e}^t}\left(\mathbf{N}_f^{pT}\,\mathbf{N}_f^p - \tilde{\mathbf{N}}_f^{pT}\,\tilde{\mathbf{N}}_f^p\right) d\,\Omega = \frac{A_e}{36}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tag{2.54}$$

2.5.3 Stabilization of the present method

It is interesting to observe that the weakly compressible formulation leads itself to the avoidance of the Poisson-like solver, as the continuity equation (2.32) together with the EOS (2.18) allow to relate the pressure to the velocity field and have to some extent a stabilizing effect. Nonetheless, this effect is not enough in the majority of the applications of the weakly compressible framework, which tend to the incompressible case using a high value of the bulk modulus K_f.

In the present work, the stabilization is obtained through a modification of Equation (2.46). For the sake of clarity, in the following this strategy will be referred to as *Consistent-Lumped Stabilization* (CLS), because it introduces the use of both the consistent and lumped version of $\mathbf{M}_\rho^n$ [45]. More in details, the CLS requires the use of the lumped matrix $\mathbf{M}_\rho^n$ when it is multiplied by the vector of unknown pressures $\mathbf{P}_f^{n+1}$, while its consistent version is used when multiplied by $\mathbf{P}_f^n$. Equation (2.46) can be thus modified in:

$$\mathbf{M}_\rho^{l,n}\mathbf{P}_f^{n+1} = \mathbf{M}_\rho^{c,n}\mathbf{P}_f^n - \Delta t^{n+1} K_f \mathbf{D}_f^n \mathbf{V}_f^n \tag{2.55}$$

where the apices c, l are introduced to highlight the choice of a lumped or consistent matrix. The lumped matrix on the left side of the equation is somehow mandatory to retain a system of decoupled equations for the explicit solver involving no matrix inversion. On the other hand, instead of using a lumped matrix also on the right side, the consistent one is employed and this discrepancy has a stabilizing effect.

To better understand the reason of this stabilizing effect, let us define $\tilde{\mathbf{P}}_f$ the vector of nodal pressures computed adopting the CLS on the mass conservation and $\mathbf{P}_f$ the corresponding one computed with the non-stabilized one (i.e. adopting a lumped matrix on both sides of (2.55)). The two vectors of nodal pressures are computed as:

$$\mathbf{M}_\rho^{l,n}\mathbf{P}_f^{n+1} = \mathbf{M}_\rho^{l,n}\mathbf{P}_f^n - \Delta t^{n+1} K_f \mathbf{D}_f^n \mathbf{V}_f^n \tag{2.56}$$

$$\mathbf{M}_\rho^{l,n}\tilde{\mathbf{P}}_f^{n+1} = \mathbf{M}_\rho^{c,n}\mathbf{P}_f^n - \Delta t^{n+1} K_f \mathbf{D}_f^n \mathbf{V}_f^n \tag{2.57}$$

With a trivial substitution, one can get to:

$$\mathbf{M}_\rho^{l,n}\mathbf{P}_f^{n+1} = \mathbf{M}_\rho^{l,n}\tilde{\mathbf{P}}_f^{n+1} - \left(\mathbf{M}_\rho^{c,n} - \mathbf{M}_\rho^{l,n}\right)\mathbf{P}_f^n \tag{2.58}$$

which highlights the additional stabilizing term introduced in the mass conservation (2.32) by the CLS. The operator $\mathbf{M}_\rho^{c,n} - \mathbf{M}_\rho^{l,n}$ is an approximation of the discretized Laplacian operator (the exact equivalence can be shown in 1D), so that one can relate the CLS strategy to the stabilizing approaches adopting the pressure laplacian described in Section 2.5.2.

Furthermore, the Equation (2.58) allows to show that for linear triangles and tetrahedra the stabilizing term added to the continuity equation by the CLS is equivalent to the one stemming from to the Direct Pressure Stabilization (DPS), defined in (2.54). Indeed, computing the difference of the consistent-lumped matrices $\mathbf{M}_\rho^{c,n} - \mathbf{M}_\rho^{l,n}$ one obtains the following additional term for linear triangles, computed at the element level:

$$\mathbf{S}_{f,e}^{PFEM}\mathbf{P}_f^n = -\left(\mathbf{M}_\rho^{c,n} - \mathbf{M}_\rho^{l,n}\right)\mathbf{P}_f^n = \frac{A_e}{12}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}\mathbf{P}_f^n \tag{2.59}$$

The comparison of (2.54) with (2.59) shows a substantial equivalence in between the DPS and the CLS here employed, besides the different scalar factor at the denominators. This equivalence explains the reason of the stabilizing effect of the CLS strategy.

It is important to remark that the additional term related to the CLS does not introduce any variation to the global mass, which can be an issue

of stabilizing methods. On the other hand, this small perturbation to the exact continuity equation has a diffusive effect, as it can be related to a discrete Laplacian operator. This inconsistency however tends to zero as the element size $h \to 0$, when the consistent and lumped matrices coincide, so that better results are obtained refining the mesh, following the spirit of FE.

Finally, let us remark that the stabilizing effect of the CLS can be applied also when the mass conservation is imposed through the Equation (2.42). In such case, the stabilized version of the equation is obtained computing $\tilde{\mathbf{F}}_\rho^0$ using a consistent matrix $\mathbf{M}_\rho^{c,0}$, while the lumped $\mathbf{M}_\rho^{l,n+1}$ is employed on the left side of the equation, to retain a diagonal matrix to be inverted, namely:

$$\mathbf{M}_\rho^{l,n+1}\mathbf{R}_f^{n+1} = \tilde{\mathbf{F}}_\rho^0 = \mathbf{M}_\rho^{c,0}\mathbf{R}_f^0 \tag{2.60}$$

3 An explicit Lagrangian FEM for free-surface fluid flows

3.1 Numerical methods for free surface flows

Many numerical methods have been proposed in the literature to simulate fluid flow problems in the presence of evolving free-surfaces. Following the kinematics frameworks introduced in Section 2.1.1, they can be classified in Eulerian and Lagrangian. Additionally, the ones that employ a combination of the two kinematics descriptions are referred to as Arbitrary Eulerian Lagrangian (ALE) methods.

Eulerian methods As previously underlined, the typical approach in fluid-mechanics is the Eulerian one: among the Eulerian numerical methods, the most used ones are the Finite Difference Method, the Finite Element Method and the Finite Volume Method. Each of these approaches has been successfully used in a wide range of fluid dynamics applications, since the fixed computational grid (mesh) allows to avoid problems of large distortions of the fluid flow. On the other hand, a fixed mesh does not allow a natural treatment of problems where the fluid boundary varies significantly during the analysis, namely free surface flows or FSI problems with structures undergoing large displacements. Moreover, when the nonlinear convective term is dominant, numerical oscillations may arise in the standard Galerkin formulation.

Several strategies have been introduced to track the fluid boundary variation in Eulerian formulations. Some of them are listed below. The Marker and Cell Method (MAC) has been firstly introduced in [46]. This approach combines a fixed Eulerian mesh with a set of massless particles moving according to the velocity computed on the cells to detect which region of the grid are occupied by the fluid. The markers indicate which fluid phase the

cell belongs to, i.e. a cell with no marker particles is considered to contain no fluid, while a cell with marker particles lying adjacent to an empty cell, is called as a surface cell. The MAC method is able to reproduce complex phenomena such as wave breaking. On the other hand, it can be computationally expensive, especially in 3D geometries. This is due to the large number of marker particles to be introduced in addition to the solution of the fluid governing equations, as well as to the fact that usually such particles need to be redistributed during the analysis since they can concentrate in some region and be sparse in some others [47–49].

In the Volume of Fluid Method (VOF) [50], a scalar function is defined as the ratio of the fluid volume to the total volume of the computational cell. Solving the transport equation for the filled fraction function on each control volume, one can follow the evolution of the free surface. Cells with values between zero and one contain a free surface and the normal direction to the boundary is defined by the gradient of the scalar function. Such an approach is able to reproduce complex free surfaces behaviour and wave breaking more efficiently than the MAC method. However, the free surface profile may be not sharply defined, but smeared over one to three cells, similarly to what happens in shocks simulations for compressible flows [47]. Several variants of this approach have been proposed in the literature [24, 51–53], and it is widely used in real engineering problems.

Another strategy is represented by the Level Set Method, described in [49]. This approach is employed in many engineering fields to detect sharp discontinuities that are not conforming to the mesh. The idea is to define a smooth function, which is usually taken as the signed distance to the interface, so that it has positive values inside the fluid domain, negative values outside and null value at the interface. To capture the free-surface one can find the set of points where the function vanishes, while its motion can be analysed by convecting its values with the velocity field [48]. The advantage with respect to the VOF is that the function is smooth across the interface, while the volume fraction in VOF is discontinuous there. Different techniques have been proposed to avoid oscillations and numerical diffusion of the results at the interface [54, 55]. Finally it must be noted that several hybrid strategies have been presented, combining for example the Level-Set and the VOF [56, 57].

Despite the large diffusion of such *ad hoc* strategies for the detection of the free surface, the Eulerian approaches may suffer of problems related to an accurate and sharp definition of the free surface in complex phenomena such as breaking waves with splashes and reconnecting interfaces. For this reason, other approaches adopting a fully Lagrangian or an ALE description have been developed.

Arbitrary Lagrangian Eulerian methods The ALE methods have been developed to combine the advantages of the Eulerian and Lagrangian descriptions [58–62]. According to this idea, the nodes of the computational mesh may be moved within the continuum in a Lagrangian fashion, or be held fixed in an Eulerian manner, or be moved in some arbitrarily specified way to give a continuous rezoning capability. Thus, the movement of the fluid particles is decoupled from the one of the mesh nodes. A convective term is present as in the Eulerian approach, but depending on the relative velocity between material points and mesh nodes. Effectively, the ALE description can be considered as a general framework in which one can find the Eulerian and Lagrangian ones as particular cases: they can be obtained from the ALE formulation setting to zero the grid-nodes velocities or the grid-nodes to fluid particles relative velocities, respectively. The ALE approach has been extensively used to solve free surface fluid and FSI problems with the Finite Element Method. The ALE formulation usually prescribes a fully Lagrangian formulation on a layer along the moving boundaries, while a mixed transition zone separates it from an Eulerian fixed description in the central zone. This is in principle very effective because it exploits the Lagrangian capacity to track the fluid boundary motion and, far from it, the Eulerian one to minimize mesh-distortion. However, the main shortcoming of this technique is that it is essentially limited to geometries where the material flow is relatively predictable and where the fluid boundaries movement is rather limited. For example, it does not allow for new free surfaces creation or merging, without introducing remeshing.

On the contrary, the present work focuses on the most general situation where the motion of the fluid boundaries can be very large and unpredictable, with complex phenomena such as breaking waves, interface merging and splashes. In such a situation, the ALE needs to prescribe a Lagrangian description for a large part of the domain and introduce remeshing. Moreover it has to face all the additional efforts to treat the convective terms.

Lagrangian methods When a purely Lagrangian framework is considered, one can avoid the problems related to the free surface tracking and the non linearities of the convective term. The Navier-Stokes Equations are written in the material coordinates so that no convection is present and the fluid free surface is naturally defined by the position of material points. On the other hand, a computational mesh attached to fluid nodes quickly undergoes large distortions. For this reason, one can define two main types of Lagrangian approaches: the meshless approaches and the mesh based ones which involve remeshing.

The meshless or mesh-free approaches eliminate the dependence on a computational grid (at least partially) by describing the computational domain through a set of particles without mesh constraints, and establishing a system of algebraic equations for the whole problem domain based on particle-derived shape functions. Among meshless methods, let us recall some of the most used ones, such as the Smooth Particle Hydrodynamics method (SPH), the Material Point Method (MPM) and the Element Free Galerkin Method (EFGM). Smooth Particle Hydrodynamics (SPH) was formulated for the first time independently in [63, 64] for astrophysical problems. Each particle refers to a geometric position in the fluid domain and stores hydrodynamic properties such as mass, momentum, density, viscosity. Moreover, these discrete physical variables are smoothed over the fluid domain as a summation on neighbour particles. More in details, they are interpolated from the value on each particle using a Gaussian or spline kernel function which is different from zero only on a localized support domain defined through the characteristic length associated to each particle [65–68]. The governing equations are solved in their strong form. Despite all the advantages of exploiting a Lagrangian mesh-free approach, SPH methods have difficulties in enforcing essential boundary condition, penetration problems between continua when high speed or impact occurs and they need the introduction of artificial viscosity to avoid unstable solutions [69]. As well as SPH, the Element Free Galerkin Method (EFGM) represents the domain as a discrete set of particles. On the other hand, EFGM differs from the SPH in discretizing the governing equations in their weak form. The interpolating functions are constructed using the moving least square approximation, generated at each nodal point and different from zero over the associated local support domain [70]. Moreover, EFGM prescribes the use of a background grid for the integration of the system matrices. The main shortcomings of the EFGM are related to the imposition of boundary conditions, as in the SPH method, and to its computational cost, which can be high especially because of the matrix integration step [71].

The Material Point Method (MPM) is a particle method, proposed for the first time in [46] for the solution of fluid flow problems under large deformations and initially known as the Particle In Cell (PIC) method. It combines the Lagrangian representation of the domain as a set of particle, the so-called material points, with a background computational mesh. All the physical properties and state variables are stored at the particle level [72, 73]. The fixed Eulerian background mesh carry no permanent information and it is employed to solve the governing equations and to determine incremental displacements and strains at the material points. Thus, at the beginning of each time step the information are transferred from the

material points to the background mesh, while at the end of the time step the solution is mapped from the background mesh back to the material points. With respect to the previous particle methods, the MPM has the additional advantage of an easy implementation of the boundary condition, as in standard FEM. Among its shortcomings, it may be expensive both in terms of storage requirements and computing time because of the information transfer from the material points to cell nodes and vice versa at each time step [74].

The Lagrangian mesh-based FEM approach has been rarely employed for the solution of fluid flows before the rise of strategies to treat the mesh distortion [75]. In [76] a mesh re-zoning technique was applied to model solitary wave propagation, with an applicability range limited to small evolution of the free-surface. An adaptive remeshing based on the advancing front algorithm was introduced in [77] to reproduce stronger free surface variations, but it was not able to model complete wave breaking including surface merging and splashes. More recently, the Particle Finite Element Method (PFEM) has been firstly introduced in [78], based on a continuous re-triangulation of the domain whenever the mesh gets too distorted. In [79, 80], a different version of the PFEM, namely the PFEM-2 has been introduced, which is based on a hybrid spatial discretization including a cloud of Lagrangian particles and a fixed background Eulerian mesh. This method has demonstrated its effectiveness in a wide range of applications, especially in convective dominant problems [81].

The present work employs a novel explicit version of the PFEM for the fluid domain, which is described in details the present chapter.

Finally, another approach which can be considered beyond this classification is the Lattice Boltzmann method (LBM) [82, 83]. LBM is widely used for standard CFD problems and it has been extended also to free surface flow. It is outside the classification criteria here considered because, in contrast with classical CFD approaches, which deal with the macroscopic Navier-Stokes equations, the LBM regards CFD problems at the microscopic scale. The primary variable is the particle distribution function, which specifies the probability to encounter a particle at given position and time. The macroscopic variables such as velocity and pressure are obtained by evaluating the hydrodynamic moments of the particle distribution function. The evolution of the particles distribution function is described by the Boltzmann equation. Such equation relates the time evolution and spatial variation of a collection of molecules to a collisional operator that describes the interaction of the molecules. At the limit of a dense distribution of particles, the macroscopic Navier-Stokes Equations are recovered [84]. Consequently, LBM provides stable and efficient numerical calculations for the

macroscopic behaviour of fluids, although describing the fluid in a microscopic way, given that you have the computational power to have a proper amount of particles. Furthermore, the term Lattice is related to the required background computational grid which need to be regular. For this reason, when dealing with free surface flows, it has to be coupled with the aforementioned strategies to capture the fluid interfaces in Eulerian approaches such as the VOF [85,86]. The main difficulties when using the LBM are related to the complex nature of the collisional integral operator and to the previoulsy commented problems of accuracy and sharpness of the free surface profile when interface capturing algorithms are employed. Moreover, despite the architecture of the algorithm is suitable for an efficient parallization which allows for the introduction of the required large number of particles [87], when it is coupled with strategies to deal with free surface flows or with other solvers for FSI problems, a performance drop is observed [85].

3.2 The Particle Finite Element Method (PFEM)

The Particle Finite Element Method (PFEM) is a mesh-based Lagrangian approach particularly suitable for problems characterized by severe changing in the domain topology. It was firstly introduced to model free surface flows [78,88] and subsequently it showed its potentialities in the simulation of FSI problems [1,89–91], granular flows [92–94], multiphase materials flows [95], forming processes [96], thermal coupled problems [97,98], transport, erosion and sedimentation in fluids [99].

The PFEM can be basically summarized as a standard Lagrangian FEM combined with an efficient algorithm for the re-triangulation of the domain which is frequently performed runtime during the analysis to ensure a proper quality of the spatial discretization. The fluid motion is followed by the mesh nodes (particles) according to the governing equations in a Lagrangian fashion. All the physical properties and variables are stored at the node level so that no information is stored at the element level. Whenever the Lagrangian motion of the nodes leads to an overly distorted mesh, such mesh is deleted, retaining its nodes with all the nodal information. A new mesh is then generated redefining the element connectivities starting from the same set of nodes. Having no data stored at the element integration points or at nodes along the element sides permits the use of remeshing without information loss or necessity of mapping between the old and new meshes which can be computationally expensive. At every time step, once the domain mesh is identified, the integral form of the governing equations are solved as in a standard FEM. The Lagrangian description together with the remeshing algorithm are particularly effective to track the fast evolution

of free surfaces involving breaking waves or splashes.

In the following part of this chapter, all the main features of the method will be described in details, together with the discussion of numerical results to highlight advantages and drawbacks of this approach.

3.2.1 Mesh generation algorithm

The regeneration of the mesh is the key feature of the PFEM. In the PFEM literature this process is referred to as "remeshing", "triangulation" or "tessellation". It must be underlined that it involves only changes in the nodal connectivities (and thus in the elements), while the position of the nodes themselves is preserved. The effectiveness of the PFEM is based on the fast and accurate generation of a mesh. The algorithm must be fast, since it is performed very frequently during the analysis. At the same time it must be accurate, in order to track the fast evolution of the fluid boundaries and solve the integral governing equations on the correct domain. Moreover, in the present novel PFEM version based on an explicit integration scheme, an additional requirement to the remeshing algorithm arises: the quality of the mesh must ensure a reasonable stable time step size (Equation (2.48)) for the overall efficiency of the explicit solver. This can be particularly demanding especially in the 3D framework. A novel technique to ensure all these requirements have been developed and presented in Chapter 6.

In the PFEM the remeshing procedure is performed through the combination of the Delaunay tessellation and the Alpha-Shape Method.

3.2.1.1 Delaunay Triangulation

To better understand the geometrical features of the Delaunay tessellation, let us first introduce the definition of the *Voronoi diagram* and *Voronoi cells*. Given a set of N points P_i in $\mathbb{R}^d$ ($i = 1, \ldots, N$; $d = 2, 3$), the *Voronoi diagram* is the partition of $\mathbb{R}^d$ in N convex regions V_i, called *Voronoi cells*. For the sake of clarity, let us now focus on the two dimensional case ($d = 2$). Each *Voronoi cell* is associate to one node n_i and defined as the locus of points $\mathbf{x} \in \mathbb{R}^2$ such that:

$$d(\mathbf{x}, P_i) \leq d(\mathbf{x}, P_j) \quad \forall i \neq j \tag{3.1}$$

where $d(\mathbf{x}, P_i) = \|\mathbf{x} - \mathbf{x}(P_i)\|$ is the Eucledian distance (Figure 3.1). It is worth noting that each side of a *Voronoi cells* belongs to two regions V_i and V_j, consequently it is the locus of points equidistant to P_i and P_j (the axis of the segment $\overline{P_i P_j}$). Furthermore, each vertex of the *Voronoi diagram* belongs to three region V_i, V_j, V_k and for this reason it is equidistant to

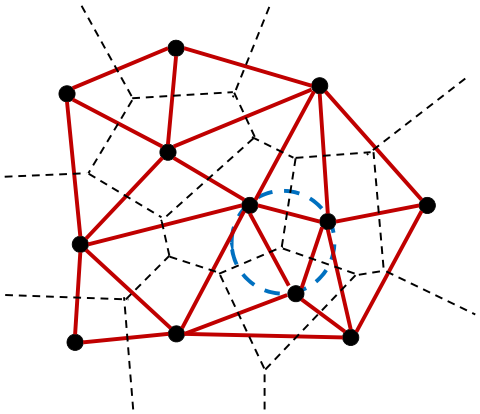


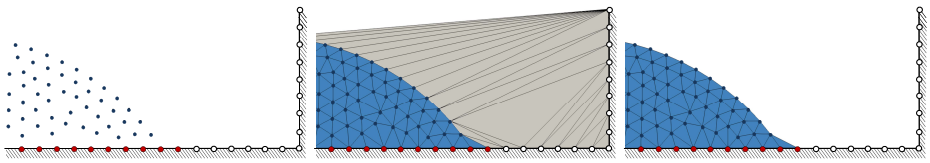
Figure 3.1: *Voronoi cells* (dashed black line) and Delaunay triangulation (solid red line). Every triangle circumcircle (dashed blue line) does not include any other points of the set.

the three points P_i, Pj, P_k, namely, it is the center of the circumference passing through the three points.

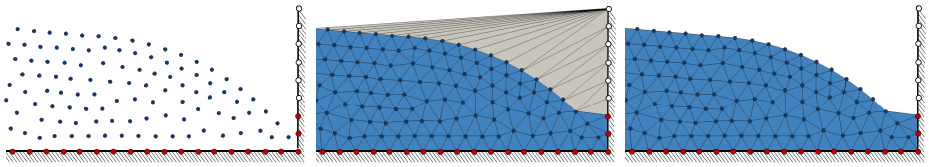
The Delaunay triangulation is the dual of the *Voronoi diagram*. It can be constructed joining the points whose *Voronoi cells* have a common boundary (Figure 3.1). This implies that the Delaunay tessellation generates a mesh of triangles (2D) and tetrahedra (3D) representing the least convex hull enclosing an arbitrary set of points, i.e. the convex figure of minimum area which encloses all the points of the set. A defining property of the resulting mesh is that none of its vertices lay inside any triangle's circumcircle (in 2D) or tetrahedron circumsphere (in 3D). In the 2D framework, the Delaunay triangulation guarantees a lot of remarkable optimal properties such as the minimization of the maximum radius of an element circumcircle and maximization of the minimum angle among all the elements. On the other hand, the 3D Delaunay algorithm loses some of the optimal properties of its 2D counterpart. Unfortunately, this has important consequences on the possible presence of bad quality tetrahedra in the mesh, which will be addressed in details in Chapter 6.

3.2.1.2 Alpha-shape algorithm

In a Lagrangian framework the external boundary $\Gamma_f^t = \partial\Omega_f^t$ and the current volume Ω_f^t are defined by the position of the material particles. In the PFEM approach it is necessary to determine the new boundaries every time a new mesh is generated. The Delaunay Triangulation generates the convex hull of a given set of points. This in general does not coincide with



(a) t^1: set of nodes **(b)** t^1: Delaunay mesh **(c)** t^1: Alpha-shape application



(d) t^2: set of nodes **(e)** t^2: Delaunay mesh **(f)** t^2: Alpha-shape application

Figure 3.2: Mesh generation algorithm at two different instants t^1 (a, b, c) and t^2 (d, e, f). (a, d) Initial set of nodes. (b, e) Delaunay mesh representing the convex hull of the set of particles, with many unphysical distorted elements (grey). (c, f) Removal of the distorted elements through Alpha-shape algorithm. The inactive boundary ghost nodes (white dots) are not included into the governing equations.

the definition of the real fluid domain, which could have concave parts and cavities that cannot be detected by the algorithm, as depicted in Figure 3.2-b,e. To recover the physical domain boundaries, the Alpha-shape method is employed [100, 101]. This method is based on the observation that the unphysical elements which do not belong to the real fluid domain are in general the most distorted ones, because they connect particles that are far one from the other (Figure 3.2-b). Consequently, a geometrical index of distortion α_e is defined for each element e as:

$$\alpha_e = \frac{R_e}{h_{mean}} \tag{3.2}$$

where R_e is the radius of the circumcircle (or circumsphere in 3D) to the considered element and h_{mean} is a characteristic mesh size, for example in the present version the average of the minimum element side among all the elements of the initial mesh. A threshold value $\bar{\alpha}$ for this distortion index is defined in order to remove from the Delaunay mesh all the elements which have a larger distortion index, namely the elements characterized by:

$$\alpha_e \geq \bar{\alpha} \iff R_e \geq \bar{\alpha} h_{mean} \tag{3.3}$$

Clearly, one of the keypoints for the correct result of the method is the proper definition of the distortion limit $\bar{\alpha}$. Different values of $\bar{\alpha}$ can lead to different configurations. If $\bar{\alpha}$ is too large, one is not having the proper resolution to track the fluid domain profile and one can recover, at the limit case, the Delaunay mesh. On the other hand, if $\bar{\alpha}$ is too small, too many elements are removed and one may lose the smoothness of the fluid boundaries or create unphysical holes inside the domain. In the version of the PFEM here presented, two values of $\bar{\alpha}$ are considered, taking advantage of the information about the fluid boundaries coming from the previous mesh. Every node belonging to the fluid boundary has a flag identifier. When a new mesh is generated, elements including nodes which belonged to the boundaries of the previous mesh are checked with a smaller limit $\bar{\alpha}_{bound} = 1.3$, to have the proper resolution on the boundary surface. Conversely, elements including only internal nodes of the previous mesh are checked with a bigger limit $\bar{\alpha}_{int} = 2$, to avoid the creation of unphysical voids and cavities inside the fluid bulk.

It must be underlined that the Alpha-shape method can introduce accuracy errors on the results of the analysis. The local changes of topology related to the insertion/removal of elements may globally alter the mass conservation, introduce local inaccuracies in the representation of the free surfaces and also local perturbations of the equilibrium reached at the previous time steps [102]. Nevertheless, it must be noted that the errors in the boundary definition are proportional to mesh size h and can be reduced down to the desired accuracy by refining the mesh [101]. Moreover, it has been verified in [103] that, if the limit distortion $\bar{\alpha}$ is chosen around the value of $\bar{\alpha} = 1.2$, the variation on the numerical results is negligible.

Finally, it is important to underline that the Alpha-shape method is also able to naturally model fluid particles/elements separating from the bulk, or the contact between flows and structures in FSI problems. The FSI application is described in details in Section 4.3.5. For the fluid separation, let us consider Figure 3.3-a: if a particle of the free surface belongs to a triangle which is overly distorted, the criterion expressed by the Relation (3.3) removes such element. The particle, with all the information saved at the node level, separates from the bulk and starts to move according to its initial velocity and the gravity field (Figure 3.3-b). In the opposite case of a separated particle that has approached enough the fluid free surface boundary (Figure 3.3-c), the triangle generated by a new Delaunay tessellation is not removed and the particle is again merged with the fluid bulk (Figure 3.3-d).

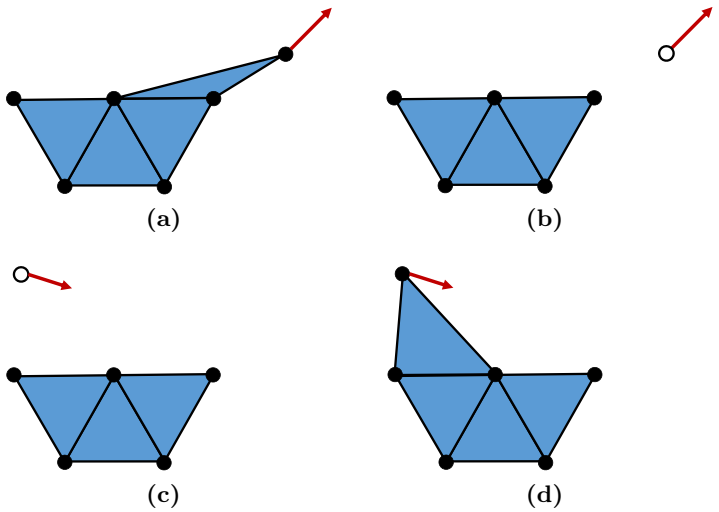


Figure 3.3: (a)-(b) The overly distorted element on the free surface is removed and one particle (white) separates from the bulk starting to move according to its velocity and the gravity field. (c)-(d) A particle approaching the bulk is incorporated when the connecting triangle is not removed.

3.2.1.3 Mesh Improving Tools

The combined action of the Alpha-shape method and the Delaunay triangulation alone cannot ensure a good quality of the mesh throughout the analysis duration. The motion of the fluid nodes can lead to regions with nodes concentration or dispersion, or more in general to nodal distributions which inevitably lead to mesh with bad quality elements. As previously commented, such a situation may affect the accuracy and the effectiveness of the solver, which becomes crucial especially in the 3D framework, because of the poorer properties ensured by 3D Delaunay tessellation. Besides the novel mesh regularization algorithm described in Chapter 6, other algorithms that are standard in the PFEM framework for the improvement of nodal distribution are here addressed. Such algorithms are based on quality checks on the current mesh and the subsequent following actions performed on the regions requiring to be improved:

- A new node is inserted in the centroid of elements whose area exceeds a threshold value (Figure 3.4);
- A node is removed from the mesh if it is too close to the neighbour ones (Figure 3.5);

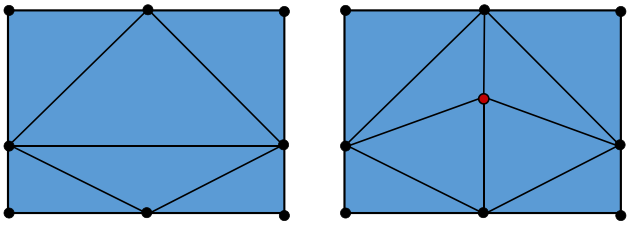


Figure 3.4: Mesh improvements: a new node (red) is inserted in the centroid of elements whose area exceeds a threshold value.

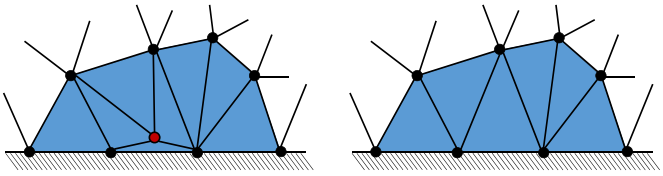


Figure 3.5: Mesh improvements: a node (red) too close to the neighbour ones is removed.

- A node too close to the neighbour ones is re-located at the centroid of the patch of elements sharing that node (Figure 3.6);

It must be underlined that the first two types of improvements need to be followed by a remeshing, while the third one does not alter the mesh connectivities. Moreover, the third node re-location is usually referred to as Laplacian Smoothing in the literature [104]. It is computationally inexpensive and quite effective in the 2D framework. However, there is no general guarantee of an improvement in the quality of the mesh and, especially in 3D, it can lead to mesh deterioration or even inverted elements [104–106]. Finally, let us remark that each of the aforementioned controls is related to predefined threshold quality values defined as functions of the characteristic mesh size h_{mean}: the frequency and the influence of these algorithms can be varied by the user considering the required mesh homogeneity.

3.2.2 Boundary Conditions

A peculiar feature of the PFEM is the way in which the boundary conditions are identified. This is done through a set of nodes distributed on the boundary walls, that will be referred to as *ghost nodes*. The ghost nodes are characterized by assigned material properties and assigned velocity which forces them to remain fixed on the boundary (e.g. zero velocity for fixed

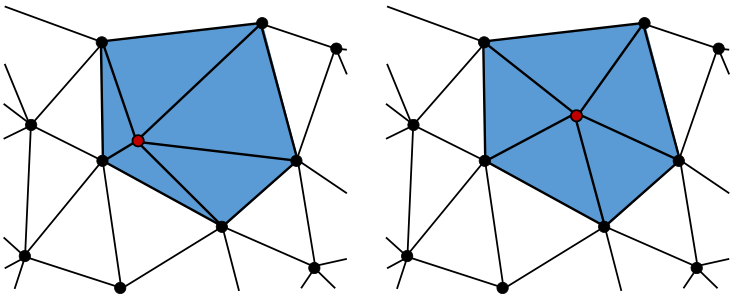


Figure 3.6: Mesh improvements: a node (red) too close to the neighbour ones is re-located at the centroid of the highlighted patch.

boundaries), otherwise the definition of the boundary itself would be lost. For example, in Figure 3.2 one can find a set of particles laying on the boundary walls where a zero velocity is imposed in order to define the limits of the computational domain. Let us remark that such set of ghost nodes covers all the boundary walls that can be possibly involved in the fluid flow, even if they are not initially in contact (e.g. the vertical wall in Figure 3.2). After the Alpha-shape algorithm, if a ghost boundary node is included in any element of the mesh, then its degrees of freedom are considered in the governing equations and it will be referred to as *wet* ghost node (red dots in Figure 3.2-c). Otherwise, the ghost node is inactive and not included in the solution of the governing equations (white dots in Figure 3.2-c).

This strategy is able to optimize the effectiveness of the Alpha-shape algorithm to deal with contact between complex and evolving surfaces. However, it is not easy to be employed in general cases where non-homogeneous conditions are imposed at the boundary walls. Such cases are addressed in details in Chapter 5 with the use of a novel strategy which employs a mixed Lagrangian-Eulerian description at the fluid boundary.

3.2.3 Mass conservation

It is well known that a crucial issue of numerical methods for incompressible (and weakly compressible) fluids is the mass conservation. There are in general two possible sources of mass variations when a free surface flow is modelled through the PFEM [103]. The first one is related to the numerical treatment of the incompressible/weakly compressible Navier-Stokes equations and it mainly depends on the stabilization procedure employed in the method. This is a common issue for numerical methods which use the weak

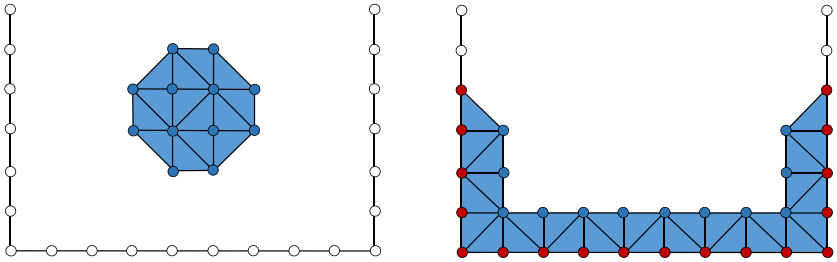


Figure 3.7: Mass variations on a coarse mesh for a fluid drop falling on an empty box.

form of the mass conservation [107]: it is more evident in the Lagrangian framework but it is experienced also in Eulerian and ALE approaches. It must be highlighted that the present version of the weakly compressible PFEM employs the CLS stabilization strategy presented in Section 2.5.3, which allows to consistently impose the global mass conservation equation, so that no mass variation is introduced by the numerical solution of the governing equations.

The second source of mass variation is instead peculiar of the PFEM and in particular of its remeshing procedure combined with the boundary representation strategy. At every remeshing step the connectivities are regenerated, in general with the creation of new elements and the removal of some others. This affects the mass balance and it is more pronounced at the free surface and at the boundary walls. Indeed, if a fluid particle approaches the boundary, new elements are created and inactive ghost boundary nodes can be activated into wet ghost nodes: consequently they become involved into the total mass computation determining a mass variation. Therefore, problems showing a big change in the number of wet boundary nodes during the simulation are intrinsically prone to mass variation. Let us consider the example depicted in Figure 3.7: a drop of fluid is falling onto an empty box. The drop is schematically discretized by 12 particles which initially form a mesh of 14 elements (Figure 3.7-a). The choice of a very coarse mesh is aimed to magnify the mass variation issue and make it more visible. The box walls are initially represented by a set of inactive ghost nodes (white dots in Figure 3.7-a). Let us assume now that the fluid hits the bottom of the box and the particles spread over its walls as shown in Figure 3.7-b. In this configuration, a large amount of ghost nodes are activated and become wet nodes (red dots in Figure 3.7-b) and they can participate to the creation of a greater number of elements. Therefore the mesh turns to 28 elements, so that the mass is doubled. Nonetheless, one should note that this

represents a limit pathological case where the fluid is initially completely detached from the boundaries and a very coarse mesh is employed.

In [103] a detailed study of this type of mass variation has been presented. More in particular, the mass variation related to the PFEM remeshing is strongly dependent on the value of $\bar{\alpha}$ and the mesh size. It has been verified that in a range of values of $\bar{\alpha}$ around $\bar{\alpha} = 1.2$ the differences in the numerical results are negligible. Moreover, it is important to underline that such perturbations can be controlled and reduced refining the mesh size. This is particularly important for the reliability of the PFEM, confirming that the method works in the spirit of the FE, obtaining better results when the mesh size is reduced.

3.3 Solution Scheme

The PFEM solution procedure is summarized in Algorithm 1. At the beginning of every time step of the dynamic analysis, a check on the distortion of the fluid mesh is performed. If the mesh employed in the previous time step is too distorted, all the connectivities are erased, retaining the set of mesh nodes storing of the fluid information. Starting from that cloud of points, the Delaunay triangulation generates a new convex mesh with good quality elements. The detection of the new fluid boundaries is performed with the Alpha-shape method. If necessary, further improvements on the mesh quality can be performed at this stage with the algorithms described in Section 3.2.1.3 (or the one presented in Chapter 6). Once a proper discretization of the fluid domain is recovered, a standard FE Central Difference Scheme step is performed in order to update the physical variables and the position of the nodes to conclude the time step. As it has been already commented, the explicit integration scheme leads to an easily parallelizable algorithm. In the present version, the elemental assemblying operations and the following solution of the global system of decoupled equations are executed in parallel on multiple threads with shared memory.

3.4 Numerical examples

3.4.1 Free Sloshing

Let us consider the sloshing problem presented in [108]. The geometry is depicted in Figure 3.8. In a rigid tank of width $2A = 2m$, an inviscid fluid is characterized by the density $\rho_f = 1000kg/m^3$. Considering the slow dynamics of the present example, a conservative choice for the reduced value of the dilational wave speed is $c_f = 100m/s$ (see Section 2.4), leading to a fluid bulk modulus of $K_f = 1.0\ 10^6 Pa$. The fluid domain has an

Algorithm 1 Explicit PFEM: Solution scheme

for $n = 0, ..., N-1$ **do**
  check mesh distortion
  if mesh too distorted **then**
    generate new mesh
    identify boundaries
    mesh improving algorithms
  end if
  estimate the new stable Δt^{n+1} from eq. (2.48)
  Mid-step velocity update $\mathbf{V}_f^{n+\frac{1}{2}}$: eq. (2.40)
  Displacement update $\mathbf{U}_f^{n+1}$: eq. (2.41)
  Compute density $\mathbf{R}_f^{n+1}$: eq.(2.60)
  Compute pressure $\mathbf{P}_f^{n+1}$: eq.(2.43)
  Compute acceleration $\mathbf{A}_f^{n+1}$: eq. (2.44)
  End-step velocity update $\mathbf{V}_f^{n+1}$: eq. (2.45)
end for

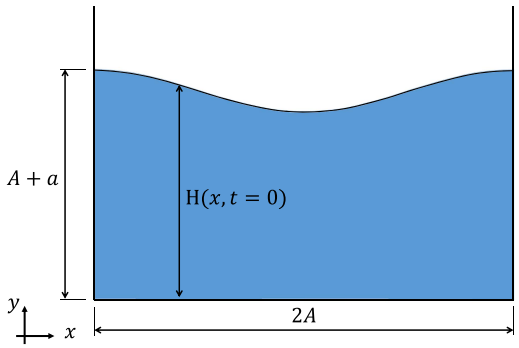
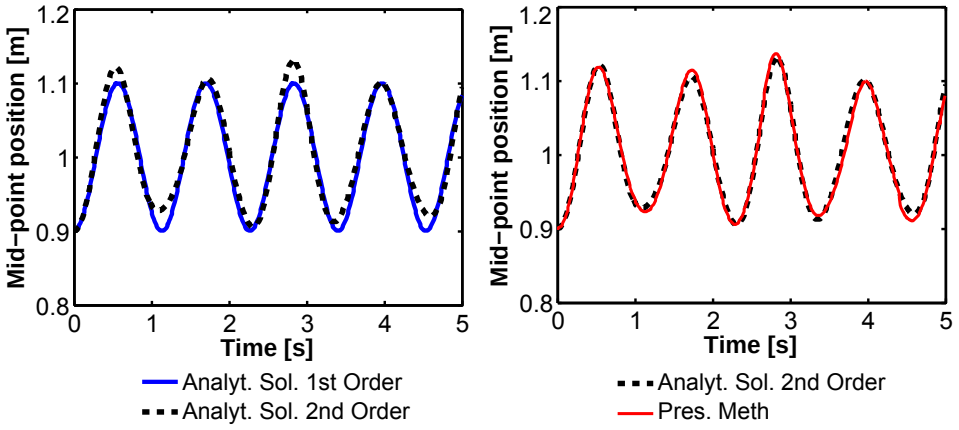


Figure 3.8: Free Sloshing. Geometry of the problem



(a) (b)

Figure 3.9: Free sloshing. (a)Analytical solutions $H_1(A,t)$ and $H_2(A,t)$ for the time history of the elevation of the midpoint of the free surface derived in [108]. (b)Comparison of the second order analytical solution with the results obtained with the present method (with the finest mesh).

initially prescribed free surface elevation $H(x, t = 0)$ with respect to the tank bottom:

$$H(x, t = 0) = A + a\cos\left[k\left(x + \lambda/2\right)\right] \tag{3.4}$$

where $a = 0.1A$ is the amplitude of the wave, $k = \frac{2\pi}{\lambda}$ is the wave number and $\lambda = 2\ m$ is the wave length. The subsequent free surface oscillations of the fluid in the tank are addressed. More in details, the time history of the free surface elevation at the center of the tank $H(A,t)$ is considered. In [108], the analytical solution of non-linear water wave equations up to the second order is derived using the Stokes perturbation expansion. The obtained analytical solution for $H(A,t)$ is the sum of a linear term H_1 and a second order term H_2:

$$H_1(A,t) = A - a\cos\left(\omega_2 t\right) \tag{3.5}$$

$$H_2(A,t) = \frac{1}{8g}\left[2\left(\omega_2 a\right)^2\cos\left(2\omega_2 t\right)\right] + \\ + \frac{1}{8g}\left[\left(\frac{a}{\omega_e}\right)^2\left(k_2^2 g^2 + \omega_2^4 - \left(k_2^2 g^2 + 3\omega_2^4\right)\cos\left(\omega_4 t\right)\right)\right] \tag{3.6}$$

where $\omega_m = \sqrt{k_m g\tanh\left(k_m A\right)}$ and $k_m = m\pi/\lambda$, with $m = 2, 4$. In the case of small waves amplitude the linear term is predominant and sufficient to

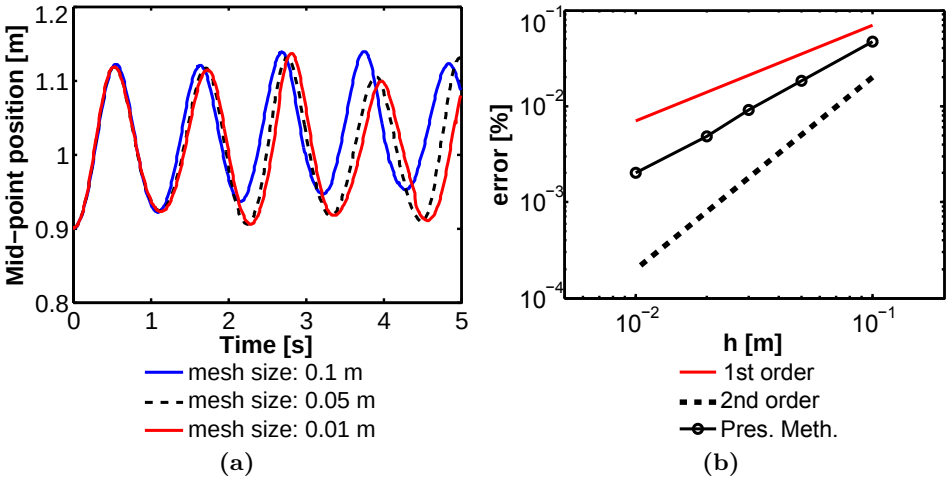


Figure 3.10: Free sloshing. (a)Time history of the elevation $H(A,t)$ of the midpoint of the free surface: comparison of different mesh sizes. (b)Averaged error with respect to the analytical solution for different mesh sizes.

describe the phenomenon represented by harmonic oscillations with constant peaks. In the case here considered, the wave amplitude of $a = 0.1A$ leads to non linear effects and non constant peaks of the oscillations are captured by the second order theory (Figure 3.9-a). Different spatial discretizations have been considered. Figure 3.10-a shows the results for three meshes characterized by an average element size of $h_1 = 0.1m$, $h_2 = 0.05m$ and $h_3 = 0.01m$, leading to mesh of approximately $0.45k$, $1.8k$ and $46k$ elements, respectively. The average stable time steps computed with the stability Relation (2.48) on the three meshes are equal to $\Delta t = 1.2\ 10^{-4}s$, $\Delta t = 7.5\ 10^{-5}s$, $\Delta t = 1.2\ 10^{-5}s$, respectively. The results of the best mesh h_3 are compared with the analytical results in Figure 3.9-b, showing a very good agreement both in terms of oscillations amplitude and frequency. A possible error definition is the following averaged error err_{h_i} with respect to the analytical solution, which takes into account the evolution of the free surface in time. It is computed on each mesh h_i as:

$$err_{h_i} = \frac{1}{N}\sum_{n=1}^{N} \left| H^{ex}(A,t^n) - H^{num}_{h_i}(A,t^n) \right| / H^{ex}(A,t^n) \qquad (3.7)$$

where N is the number of time steps of the time discretization. The averaged error for all the considered meshes are plotted in a log-log graph in Figure 3.10-b, showing a decrease in the magnitude which is more than linear with the average mesh size h.

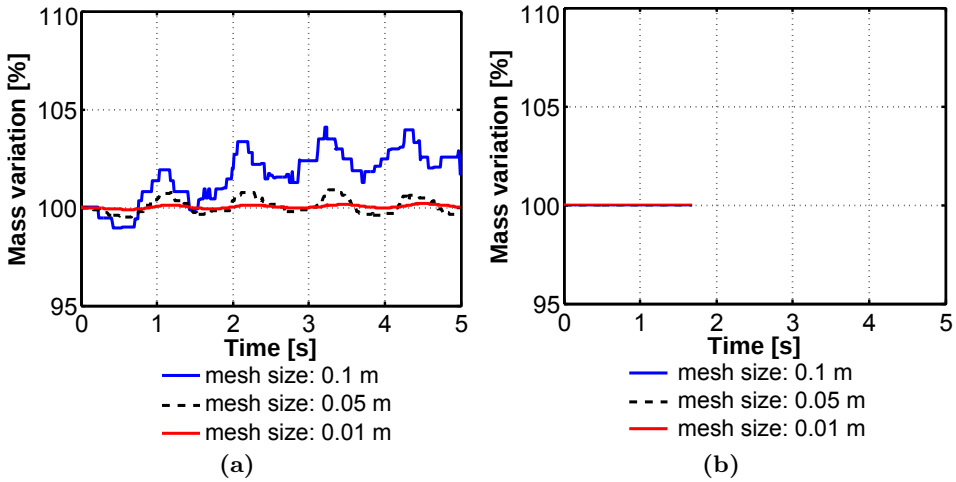


Figure 3.11: Free sloshing. Time evolution of the mass variation during the analysis with different mesh sizes. (a)With remeshing. (b) Without remeshing: results until the mesh distortion stops the analyses.

Let us now consider the mass variations introduced in the present version of the PFEM by the remeshing procedure. Figure 3.11-a represents the time evolution of the total fluid mass variation with respect to the initial mass for the different employed meshes. As described in Section 3.2.3, one can observe oscillations in the total fluid mass that are related to the free surface oscillations and the consequent variations in the number of wet boundary nodes. When the crests of the wave are at the tank sides, the number of wet boundary ghost nodes included in the triangulation increases leading to an increase in the fluid mass. These mass variations are intrinsic in the PFEM method. They are acceptable because they can be controlled and reduced refining the mesh, as it can be observed in Figure 3.11-a, where on the finest mesh the maximum variation is less than 0.5%. Finally, Figure 3.11-b shows the mass variation lines for analyses performed using the same meshes and disabling the remeshing procedure. After approximately $1.8s$ the meshes get too distorted and the analyses stop. However, it is interesting to underline that the mass curves for all the three considered meshes are perfectly aligned showing no mass variation, representing a proof that the only source of mass variation in the present method is given by the remeshing procedure.

3.4.2 Drop fall

In this example, the free fall of a disk of water and its impact onto a water basin at rest is considered [24]. The geometry of the problem is de-

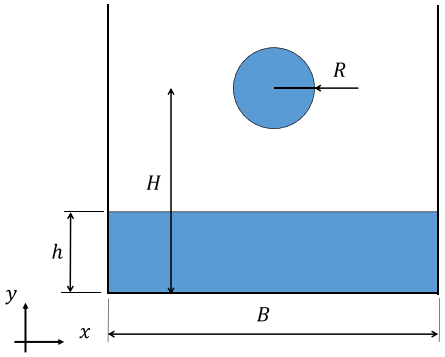
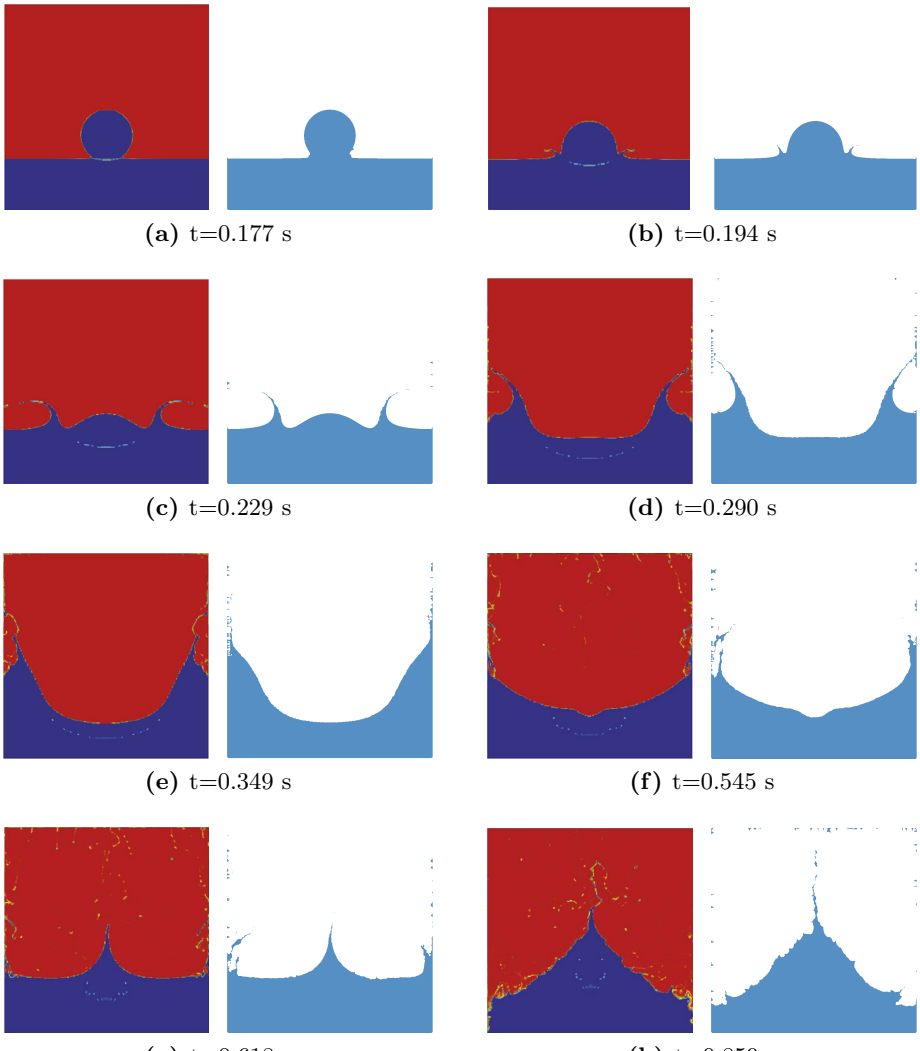


Figure 3.12: Fall of a water drop. Geometry of the problem

picted in Figure 3.12. The Newtonian fluid is characterized by density $\rho_f = 1000kg/m^3$ and viscosity $\mu_f = 10^{-3}Pa\ s$, while the adoption of the reduced value of the dilational wave speed $c_f = 150m/s$ leads to the bulk modulus $K_f = 2.25\ 10^7 Pa$. The fluid domain is discretized with a mesh of average element size $h = 0.003m$ leading to approximately $27k$ elements. The average stable time step during the analysis is then equal to $\Delta t = 4.63\ 10^{-6}s$.

The impact of the sphere with the water at rest leads to the formation of two jets on either side of the initial disk. Then the gap generated by the fall of the disk is progressively filled up with the formation of a new water jet at the middle of the box. Figure 3.13 shows the snapshots of the numerical results obtained with the present method compared with the corresponding ones presented in [24], where an ALE finite volume scheme is employed solving the equations in a Lagrangian form and then mapping the solutions back on the original mesh. The comparison shows a good agreement between the two approaches. Moreover, one can observe that the expected symmetry is well-conserved throughout the analysis. For a more quantitative comparison, Figure 3.14 superimposes the free surface profiles of the results at two different instants of the analysis, confirming the good matching of the results obtained with the present method and the ones presented in [24]. In Figure 3.15 the time evolution of the total fluid mass variation is plotted for different meshes. Also in this example the mass oscillations can be related to the variation in the number of the wet boundary nodes due to the oscillations of the water level at the two lateral walls. One can note that the mass variations are small and decrease with the mesh refinement, leading to a maximum variation of 1.81% on the finer mesh.



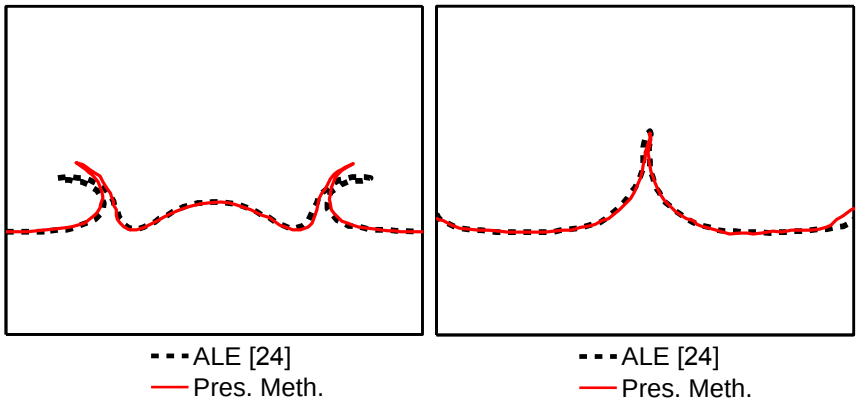
(a) t=0.177 s **(b)** t=0.194 s

(c) t=0.229 s **(d)** t=0.290 s

(e) t=0.349 s **(f)** t=0.545 s

(g) t=0.618 s **(h)** t=0.859 s

Figure 3.13: Fall of a water drop. Snapshots at different time instants. Comparison of the results presented in [24] (left) and the ones obtained with the present method (right)



(a) $t = 0.229s$ **(b)** $t = 0.618s$

Figure 3.14: Fall of a water drop. Free surface profiles comparison at two time instants.

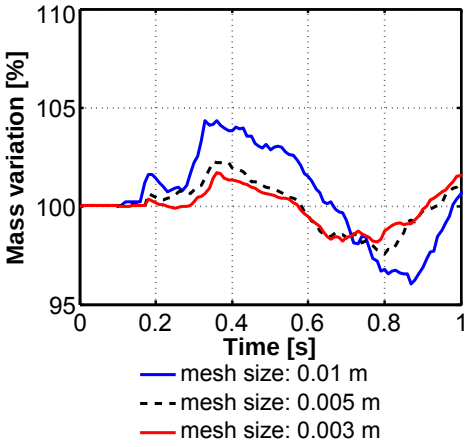


Figure 3.15: Fall of a water drop. Time evolution of the mass variation during the analysis with different mesh sizes.

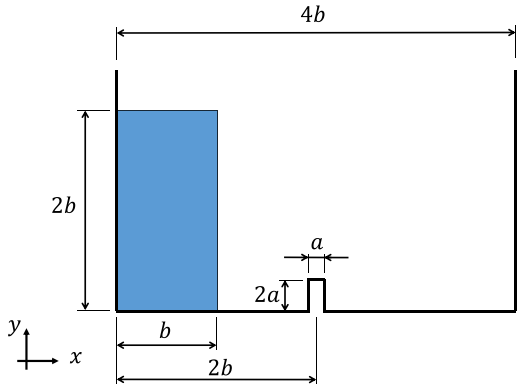
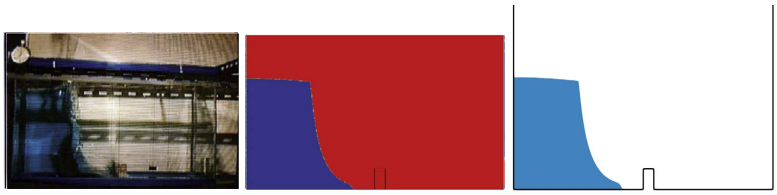


Figure 3.16: Dam collapse on rigid obstacle. Geometry of the problem

3.4.3 Dam collapse with rigid obstacle

The simulation of the experimental dam break presented in [109] is here addressed. The experimental setting is depicted in Figure 3.16: on the left side of the domain, a column of water is sustained by a rigid wall which simulates the presence of a dam. As the experiment starts, the wall is quickly removed, leaving the water flow in the tank under the effect of gravity. The flow impinges against a rigid obstacle located in the middle of the box, with the consequent formation of a long wave hitting the right wall. The constitutive parameters of the water used in the simulation are: density $\rho_f = 1000\ kg/m^3$, viscosity $\mu_f = 10^{-3} Pa\ s$ and bulk modulus $K_f = 2.25\ 10^7 Pa$ (i.e. a reduced sound speed of $c_f = 150 m/s$ is employed). The problem is solved with a mesh of approximately $65k$ elements and an average time step size of $6.25 \cdot 10^{-7}$ s.

Figure 3.17 shows some snapshots of the simulation at synchronized instants with the experimental ones presented in [109] and numerical ones in [24], showing a substantial agreement with the results presented in the literature. For a more quantitative comparison, Figure 3.18 compares the free surface profiles of the present method with the experimental ones of [109] and the numerical ones of [24] at two different instants. One can note a higher elevation of the experimental water column with respect to the numerical results at the first stage of the analysis (Figure 3.18-a). Such difference has been reported also by other numerical results in the literature [110, 111] and can be explained by the initial raise of the gate at the beginning of the problem (as noted in [24]). This procedure is modeled as a boundary condition on the fluid velocity that is instantaneously released, while in the experiment it takes a short but finite interval of time, delay-



(a) t=0.129 s

(b) t=0.258 s

(c) t=0.387 s

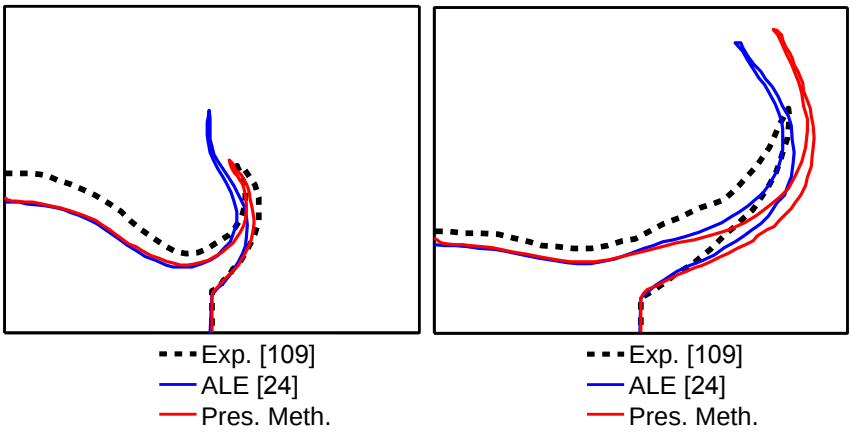
(d) t=0.516 s

Figure 3.17: Dam collapse on a rigid obstacle. Snapshots at different instants. Comparison of the experimental results presented in [109] (left), the numerical ones presented in [24] (center) and the ones obtained with the present method (right)

ing the fall of the water column. Consequently, this difference affects the first stages of the simulation and all the numerical approaches show a common lower profile with respect to the experimental one. Furthermore, in Figure 3.18-a one can note a good matching between the numerical results and a better representation of the experimental wave crest obtained with the present method. Conversely, in Figure 3.18-b the wave profile of the present method is slightly lower than the experimental one. This may be related to the missing effect of the air entrapped below the water wave. Indeed, no air phase is included in the present approach, while it is modelled in [24].

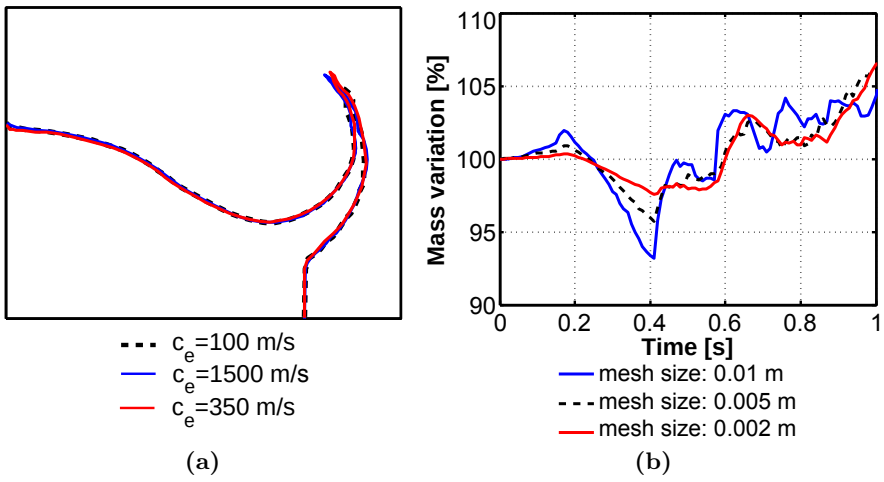
Due to its complex dynamics, the present test is a relevant case for testing the possible influence of the lower dilational wave speed c_f which is usually employed in weakly compressible approaches (Section 2.4). Let us consider the maximum fluid velocity reported in the simulation, which can be conservatively assumed equal to: $\max |\mathbf{v}_f| = 10m/s$. Then, the choice in the reference solution of a reduced dilational wave speed $c_f = 150m/s$ (i.e. a reduced bulk modulus of $K_f = 2.25\ 10^7 Pa$) leads to a Mach number of $Ma = 0.0667$ (computed through Equation (2.9)). Two additional analyses have been performed, employing a dilational wave speed of $c_f = 100m/s$ and $c_f = 1500m/s$, respectively. More in particular, the former leads to a Mach number of $Ma = 0.1$ which is usually considered as the limit case for the representation of an incompressible flow; while the latter is the real value for the dilational wave speed in water (i.e. a bulk modulus of $2.25\ 10^9 Pa$). The results of the three analyses are basically overlapping throughout the simulation, confirming that a Mach number below the value of $Ma = 0.1$ ensures a good approximation of an incompressible flow by a weakly compressible one. For the sake of brevity, the free surface profiles at one time instant ($t = 0.258s$) for the three analyses are compared in Figure 3.19-a, showing that the results are perfectly matching.

Finally, Figure 3.19-b represents the time evolution of the total fluid mass variation for different meshes. Larger variations with respect to the previous examples are observed, especially after $t = 0.5s$. However, this was expected and it is related to the impact of the long wave with the right tank wall, with a large variation of the wet ghost boundary nodes and the involved complex phenomena of fluid jets and splashes. Furthermore, once again such mass variations can be reduced refining the mesh size, leading to the maximum mass variation obtained on the finer mesh of 6.01% at the end of the analysis (while before the impact with the tank wall the maximum variation is less than 2.44%). Consequently, one can observe that also in this case with a complex dynamics and fast and strong free surface changes, the method provides accurate results and acceptable mass variations.



(a) $t = 0.258s$ **(b)** $t = 0.387s$

Figure 3.18: Dam collapse with rigid obstacle. Free surface profiles comparison at two time instants.



(a) **(b)**

Figure 3.19: Dam collapse with rigid obstacle. (a) Free surface profiles comparison varying the bulk modulus of the fluid K_f through the dilational wave speed c_e. (b) Time evolution of the mass variation during the analysis with different mesh sizes.

4 Partitioned PFEM-FEM Fluid Structure Interaction

4.1 Numerical methods for Fluid-Structure Interaction (FSI) problems

A FSI problem is defined by the fluid governing equations, the structure governing equations and the kinematic and static conditions trasmitted across the fluid-structure interface. Many different approaches have been presented to tackle the challenges related to the multidisciplinary nature of these problems. The main classifications of these methods are based on the way of enforcing the interface condition between the fluid and the structure, the way of solving the coupled system of governing equations and the kinematics setting adopted to write such equations.

Weak and Strong Coupling A first classification is based on how the static and kinematic coupling conditions are exchanged at the interface between the fluid and the structure [112]. An algorithm is defined as weakly (or loosely) coupled if the static and kinematic continuities are not exactly satisfied at each time step, otherwise a strongly coupled algorithm is obtained. The asyncronous imposition of static and kinematic conditions of weakly coupled schemes may lead to lack of accuracy and numerical energy generation at the fluid-structure interface, which has to be controlled in order to guarantee stability. Despite the weak coupling has been successfully used in aeroelasticity and in many other complex FSI applications [112], it is widely reported in the literature [113–119] that such algorithms provide unstable solutions under certain problem settings. Typically, this happens in FSI problems involving slender structures and incompressible fluid, with the structure to fluid density ratio close to one, irrespectively of the time-step size. Such conditions lead to numerical issues related to the so called

added mass effect [95, 113, 116, 120] and are generally treated with strongly coupled schemes. Nonetheless, it is important to underline that the added mass effect can lead to numerical problems such as a slow or complete lack of convergence also in strongly coupled schemes [95, 112].

Monolithic and partitioned Secondly, solution techniques to FSI problems can be divided into two categories depending on the way of solving the system of governing equations [117]: the monolithic approaches [115, 121, 122] and the partitioned ones [123, 124]. In the former, the whole problem is treated as a unique entity and the fluid and solid equations are solved simultaneously. The main advantage of this approach is the preservation of accuracy and stability because of the natural strong coupling: the exchange of interface conditions is implicitly included in the solution of the global system of equations. On the other hand this approach requires the development of a specific software, not allowing for the reuse of existing software for the fluid and structural domains. Moreover, for real engineering problems, it must be underlined that the unified solution of a large system of equations coming from two different physical domains leads to numerical difficulties that can limit its applications. By contrast, in partitioned approaches, the fluid and structural equations are solved independently and coupled through the exchange of kinematic and static interface conditions. Partitioned methods can provide strong coupling, for example through subiterations or predictor/corrector techniques, or weak coupling otherwise. Among the advantages of the partitioned approach, let us remark the possibility to reuse of existing codes for the physical subdomains (code modularity) and the possibility to solve the resulting systems of equations of smaller dimensions and better conditioned than in the case of monolithic schemes. On the other hand, the possible rise of numerical issues requires a proper choice of the interface coupling algorithm, depending on the targeted applications, to ensure accuracy, stability and efficiency of the coupled solver.

Eulerian and Lagrangian Finally, as it has been remarked for fluid numerical methods in Section 2.1.1, also the FSI algorithms can be classified according to the Eulerian or Lagrangian kinematic framework used for the governing equations both in the structural and fluid domains. As it happens in free surface flows, in complex FSI problems with large structural displacements the fluid structure interface needs to be tracked in Eulerian approaches or it is directly defined by the position of the boundary particles in Lagrangian ones. When a heterogeneous domain constituted by a fluid and a structural part is analyzed, one can apply fully Eulerian [125, 126]

or fully Lagrangian approaches [1, 89, 124, 127], where the whole domain is analysed within a unified framework. On the other hand, there are approaches which combine two different formulations for the two subdomains. For example, the Coupled Eulerian-Lagrangian [128, 129] and the Immersed Boundary Method [130, 131] employ two overlapping grids with a Lagrangian mesh for the structure moving on a fixed Eulerian mesh for the fluid. The former, computes the fluid volume fraction for each eulerian cell and then evaluates the fluid surface which interacts with the structure with standard contact algorithms. The latter instead treats the interaction interpolating the variables at the fluid structure interface from the two non conforming discretizations. Conversely, in the Arbitrary Lagrangian-Eulerian Method [132, 133] one can use the two kinematic settings in the fluid and structure domains. One of the most common strategy is to adopt a standard Lagrangian formulation for the structure and an ALE formulation for the fluid, whose mesh is Lagrangian at fluid structure interface to follow the structural motion and deforms in a standard ALE fashion elsewhere [134].

According to these classifications, the PFEM-FEM FSI approach here presented is partitioned with strong coupling and fully Lagrangian. In this partitioned scheme, the interface coupling is provided by the Gravouil and Combescure (GC) algorithm [135]. The technique has its roots in the Domain Decomposition Methods [136] and it ensures a strong coupling and the stability of the interaction algorithm. It allows to use different time steps sizes and non-conforming meshes in the different subdomains. When conforming meshes across the interface are used, the global system of governing equation is fully decoupled, resulting in a fully explicit coupled solver. Conversely, if non-conforming meshes are employed, a small linear system of equations has to be solved on the interface. However, it must be underlined that such system is small (it involves only the interface nodes) and narrow banded.

Moreover, the coupled FSI solver is fully Lagrangian, based on the PFEM for the fluid domain, and the standard Finite Element Method (FEM) provided by the commercial software Abaqus/Explicit for the structural domain. Consequently, the fluid-structure interface is automatically and accurately tracked by the mesh nodes, with no need for *ad hoc* interface capturing algorithms.

4.2 FSI problem setting and structural governing equations

Let us consider a continuum domain Ω^t evolving in the time interval $[0, T]$. The domain is constituted by two non-overlapping subdomains: a fluid

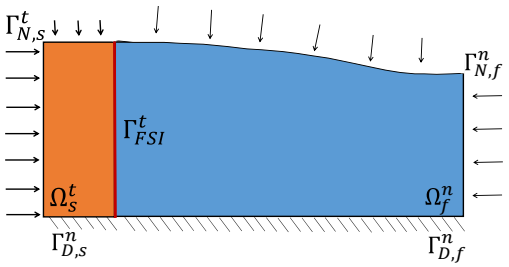


Figure 4.1: FSI Problem setting

one, Ω^t_f and a structural one, Ω^t_s. Let us define the subdomains boundaries $\Gamma^t_i = \partial\Omega^t_i$, with $i = f, s$. The fluid-structure interface is given by $\Gamma^t_{FSI} = \partial\Omega^t_f \cap \partial\Omega^t_s$. For each subdomain boundary, one can introduce two additional partitions $\Gamma^t_{D,i}$ and $\Gamma^t_{N,i}$ referring to Dirichlet and Neumann boundary conditions, respectively, such that (see Figure 4.1):

$$\begin{aligned}
&\Gamma^t_{D,i} \cup \Gamma^t_{N,i} \cup \Gamma^t_{FSI} = \Gamma^t_i \\
&\Gamma^t_{D,i} \cap \Gamma^t_{N,i} = \emptyset \\
&\Gamma^t_{D,i} \cap \Gamma^t_{FSI} = \emptyset \\
&\Gamma^t_{N,i} \cap \Gamma^t_{FSI} = \emptyset
\end{aligned}$$

In the following, we will denote with $\mathbf{u}_i = \mathbf{u}_i(\mathbf{x}, t)$, $\mathbf{v}_i = \mathbf{v}_i(\mathbf{x}, t)$, $\mathbf{a}_i = \mathbf{a}_i(\mathbf{x}, t)$, $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}_i(\mathbf{x}, t)$ and $\mathbf{b}_i = \mathbf{b}_i(\mathbf{x}, t)$ the displacement, velocity, acceleration, Cauchy stress and body force fields (being $i = f, s$) respectively. The fluid subdomain is modelled with the Explicit PFEM approach described in Chapter 3. On the other hand, the solid subdomain is analyzed using the off-the-shelf commercial code Abaqus/Explicit [137]. This allows to exploit all the advanced features of Abaqus/Explicit (e.g. the wide library of material constitutive relations and the possibility to include in the model crack propagation and contact interactions) and therefore it opens the way to the simulation of FSI problems with advanced structural modeling features. In this section, the general formulation of the initial-boundary value problem for the dynamic analysis of structures is addressed, in order to introduce all the variables that are used in the algorithm for FSI problems described in Section 4.3.

4.2.1 Structural Governing equations

The strong form of the governing equations for the structural domain consists of the momentum conservation, together with proper initial and bound-

ary conditions:

$$\rho_s \frac{d\mathbf{v}_s}{dt} = \nabla_{\mathbf{x}} \cdot \boldsymbol{\sigma}_s + \rho_s \mathbf{b}_s \quad \text{in } \Omega_s^t \tag{4.1}$$

$$\mathbf{v}_s(\mathbf{x}, t=0) = \mathbf{v}_{0,s} \quad \text{in } \Omega_s^0 \tag{4.2}$$

$$\mathbf{u}_s(\mathbf{x}, t) = \tilde{\mathbf{u}} \quad \text{on } \Gamma_{D,s}^t \tag{4.3}$$

$$\boldsymbol{\sigma}_s \cdot \mathbf{n}_s = \mathbf{h}_{\mathbf{s}}(\mathbf{x}, t) \quad \text{on } \Gamma_{N,s}^t \tag{4.4}$$

where $\mathbf{u}_{0,s}$, $\mathbf{v}_{0,s}$, $\bar{\mathbf{u}}_s$, $\mathbf{h}_s$ are assigned functions and $\mathbf{n}_s$ is the outward normal to the boundary Γ_s^t.

4.2.2 Space and time discretization

Following the same Galerkin approach used for the discretization of the fluid equations in Chapter 2, let us introduce W^u the space of admissible functions for the displacement field together with the corresponding space W_0^u of of functions satisfying homogeneous boundary conditions:

$$W^u = \{\mathbf{u}_s \in \mathbf{H}^1(\Omega_s^t) \,|\, \mathbf{u}_s = \tilde{\mathbf{u}}_s \quad on \quad \Gamma_{D,s}^t\}$$
$$W_0^u = \{\mathbf{u}_s \in \mathbf{H}^1(\Omega_s^t) \,|\, \mathbf{u}_s = \mathbf{0} \quad on \quad \Gamma_{D,s}^t\}$$

The weak form of equation (4.1) is obtained using a vector test function $\mathbf{w}_s \in W_0^u$ and integrating over the domain Ω_s^t:

$$\int_{\Omega_s^t} \mathbf{w}_s \cdot \rho_s \frac{d\mathbf{v}_s}{dt}\, d\Omega = -\int_{\Omega_s^t} \boldsymbol{\sigma}_s : \nabla_{\mathbf{x}} \mathbf{w}_s \, d\Omega +$$
$$+ \int_{\Omega_s^t} \mathbf{w}_s \cdot \rho_s \mathbf{b}_{\mathbf{s}} d\Omega + \int_{\Gamma_{N,s}^t} \mathbf{w}_s \cdot \mathbf{h}_s \, d\Gamma \quad \forall\, \mathbf{w}_s \in W_0^u \tag{4.5}$$

Furthermore, let us replace the continuous spaces W^u and W_0^u with the discrete spaces $W_h^u \in W^u$, $W_{h,0}^u \in W_0^u$ containing piecewise polynomial functions defined on the individual elements. The unknown fields can be replaced by the standard discretizations:

$$\mathbf{u}_s(\mathbf{x}, t) = \mathbf{N}_s^u(\mathbf{x})\mathbf{U}_s(t) \tag{4.6}$$

$$\mathbf{v}_s(\mathbf{x}, t) = \mathbf{N}_s^u(\mathbf{x})\mathbf{V}_s(t) \tag{4.7}$$

$$\mathbf{a}_s(\mathbf{x}, t) = \mathbf{N}_s^u(\mathbf{x})\mathbf{A}_s(t) \tag{4.8}$$

$$\tag{4.9}$$

where $\mathbf{N}_s^u$ denotes the matrix of the shape functions and $\mathbf{U}_s(t)$, $\mathbf{V}_s(t)$ and $\mathbf{A}_s(t)$ indicate the vectors of nodal displacement, velocities and accelerations, varying in time. Substituting into the weak form (4.5), one obtains

the semi-discretized system of governing equations:

$$\mathbf{M}_s \frac{d\mathbf{V}_s}{dt} = \mathbf{F}_{ext,s} - \mathbf{F}_{int,s} = \mathbf{F}_s \tag{4.10}$$

where $\mathbf{M}_s$ is the structural mass matrix, $\mathbf{F}_{ext,s}$, $\mathbf{F}_{int,s}$, $\mathbf{F}_s$ are the vectors of external and internal forces and of their resultant, respectively.
The time integration scheme implemented in Abaqus/Explicit is the Central Difference Scheme already discussed in Section 2.4. Focusing on a time step $\Delta t^{n+1} = t^{n+1} - t^n$, the solution is computed through the steps that are summarized in Algorithm 2.

Algorithm 2 Explicit FEM: Central Difference Solution Scheme

for $n = 0, ..., N-1$ **do**
estimate the new stable Δt^{n+1}
Mid-step velocity update $\mathbf{V}_s^{n+\frac{1}{2}} = \mathbf{V}_s^n + \mathbf{A}_s^n \frac{\Delta t}{2}^{n+1}$
Displacement update $\mathbf{U}_s^{n+1} = \mathbf{U}_s^n + \mathbf{V}_s^{n+\frac{1}{2}} \Delta t^{n+1}$
Internal forces update: material integration at Gauss Points
Compute acceleration $\mathbf{A}_s^{n+1} = \mathbf{M}_s^{-1} \mathbf{F}_s^{n+1}$
End-step velocity update $\mathbf{V}_s^{n+1} = \mathbf{V}_s^{n+\frac{1}{2}} + \mathbf{A}_s^{n+1} \frac{\Delta t}{2}^{n+1}$
end for

4.3 Coupling with the GC domain decomposition approach

The so-called GC method (Gravouil-Combescure [135,138]) is chosen in the present work to couple the fluid and structural domains, since it allows the implementation of different integration time steps in the different subdomains, as well as nonconforming meshes at the fluid-structure interface. The GC algorithm is a Domain Decomposition Method for non overlapping domains which was initially introduced for structural dynamics by Gravouil and Combescure [135], starting from the general frame of FETI techniques [136]; later it has been extended to FSI problems, for example in [124] and [139]. According to the GC algorithm, the fluid and structural domain are solved independently, as if there was no interaction between them. The two separated analyses are then synchronized by considering a system of constraint equations at the fluid-structure interface. This can be done imposing the continuity of the velocities through the definition of interface tractions. The tractions play the role of Lagrange multipliers and are then applied as natural boundary conditions to correct the solutions of the two subdomains. Therefore, the GC method can be defined as a predictor/corrector algorithm which ensures the strong coupling of the partitioned approach. The SIMULIA Co-Simulation Services (CSS) is

a software component for run-time coupling of simulation tools, including Dassault Systèmes simulation solvers and third-party simulation software for multi- and multi-scale simulations [137]. Specialized coupling schemes such as the GC coupling scheme can be implement using the CSS, which go beyond the standard explicit (e.g. Gauss-Seidel) and implicit-iterative coupling schemes. In the present work, the CSS have been used to couple Abaqus/Explicit with the explicit PFEM code presented in Chapter 3 through the GC algorithm. For the sake of clarity, the method is first presented in the case of equal time integration step size in the two subdomains.

The key idea behind the algorithm is to split the kinematic solution of each subdomain into two terms, respectively denoted as *free* and *link* solutions. The former is related to the free motion of each subdomain as if they were not interacting with each other; the latter evaluates the correction of the first solution, applying boundary tractions that play the role of Lagrange multipliers for the imposition of a kinematic constraint at the fluid-solid interface. The weak form of the governing equations of the coupled system is given by:

- the weak form of the momentum equation for the fluid subdomain

$$\int_{\Omega_f^t} \mathbf{w}_f \cdot \rho_f \frac{d\mathbf{v}_f}{dt} d\Omega = -\int_{\Omega_f^t} \boldsymbol{\sigma}_f : \nabla_{\mathbf{x}} \mathbf{w}_f d\Omega + \int_{\Omega_f^t} \mathbf{w}_f \cdot \rho_f \mathbf{b}_f d\Omega$$
$$- \int_{\Gamma_{FSI}^t} \boldsymbol{\lambda} \cdot (\mathbf{g}_f - \mathbf{g}_s)\, d\Gamma \quad \forall\, \mathbf{w}_f \in S_0^v \tag{4.11}$$

- the weak form of the momentum equation for the structure subdomain:

$$\int_{\Omega_s^t} \mathbf{w}_s \cdot \rho_s \frac{d\mathbf{v}_s}{dt} d\Omega = -\int_{\Omega_s^t} \boldsymbol{\sigma}_s : \nabla_{\mathbf{x}} \mathbf{w}_s d\Omega + \int_{\Omega_s^t} \mathbf{w}_s \cdot \rho_s \mathbf{b}_s d\Omega +$$
$$+ \int_{\Gamma_{N,s}^t} \mathbf{w}_s \cdot \mathbf{h}_s\, d\Gamma + \int_{\Gamma_{FSI}^t} \boldsymbol{\lambda} \cdot (\mathbf{g}_f - \mathbf{g}_s)\, d\Gamma \quad \forall\, \mathbf{w}_s \in W_0^u \tag{4.12}$$

- the additional kinematic constraint at the fluid-structure interface:

$$\int_{\Gamma_{FSI}^t} \boldsymbol{\lambda} \cdot (\mathbf{g}_f - \mathbf{g}_s) d\,\Omega = 0 \quad \forall \boldsymbol{\lambda} \tag{4.13}$$

In the previous expressions $\mathbf{g}_i$ is the kinematic variable to be constrained at the interface (displacement, velocity or acceleration), while $\boldsymbol{\lambda}$ is the corresponding Lagrange multipliers vector, having the dimensions of a traction. As it will be empahsized in Section 4.3.4, the optimal choice for the interaction problem here considered is to enforce the constraint in terms of velocities, i.e. $\mathbf{g}_i = \mathbf{v}_i$.

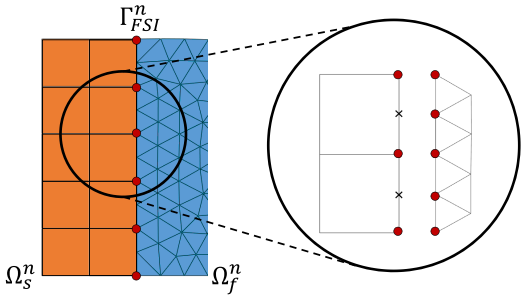


Figure 4.2: Dissimilar nested meshes at the fluid-structure interface. The kinematic constraint is enforced at the interface nodes of the finer mesh (red dots), thus structural velocities are interpolated on the intermediate nodes along the element sides (black crosses)

4.3.1 Spatial discretization

Since the requirements for mesh resolution can be very different in the fluid and structure subdomains, the possibility of adopting non conforming meshes at the fluid-structure interface is being taken advantage of. Following the approach in [140], the kinematic constraint is imposed at all the interface nodes. In other words, focusing on the more general case of dissimilar nested meshes, the constraint equations are enforced at all the nodes of the finer mesh on the fluid-structure interface (Figure 4.2). This choice represents the application, for nested meshes, of the strategy proposed in [140, 141]. Those works remark that this is the optimal strategy for first-order finite elements. It avoids locking problems and provides a perfect bonding, ensuring no additional numerical variation of energy at the interface. For rhe sake of simplicity, let us now consider the case of a smaller mesh size on the fluid side. Recalling equations (2.21)-(2.23), one can introduce the standard finite element spatial discretization also for the Lagrange multipliers:

$$\boldsymbol{\lambda}^h(\mathbf{x},t) = \sum_l \mathbf{N}^\lambda_l(\mathbf{x})\boldsymbol{\Lambda}_l(t) \tag{4.14}$$

being l the number of the finer mesh nodes at the fluid-structure interface where the multipliers are defined. Substituting (4.14) in the continuous form of the kinematic constraint (4.13), one gets:

$$\int_{\Gamma^t_{FSI}} \left(\sum_l \mathbf{N}^\lambda_l \boldsymbol{\Lambda}_l\right)^T \left(\sum_k \mathbf{N}^v_k \mathbf{V}_{f,k} - \sum_j \mathbf{N}^u_j \mathbf{V}_{s,j}\right) d\Gamma = 0 \tag{4.15}$$

which leads to the discretized equation [140]:

$$\mathbf{C}_f \mathbf{V}_f - \mathbf{C}_s \mathbf{V}_s = 0 \tag{4.16}$$

In the case of conforming meshes, both $\mathbf{C}_i$ matrices are boolean and extract the interface nodal variables from the global vector of variables, in other words:

$$\mathbf{C}_{i\ (dim\Gamma_{FSI} \times dim\Omega_i)} : \Omega_i \rightarrow \Gamma_{FSI}$$
$$\mathbf{V}_{i,\Gamma_{FSI}} = \mathbf{C}_i \mathbf{V}_i \quad i = f, s$$

In the case of non conforming meshes, as depicted in Figure 4.2, the $\mathbf{C}_i$ matrices have an equal number of rows. More specifically, in the considered case of a smaller mesh size on the fluid side, the matrix $\mathbf{C}_f$ is a boolean matrix, while $\mathbf{C}_s$ has coefficients that not only extract the interface nodal velocities, but also interpolate velocities on the intermediate points along the structural element sides (denoted with black crosses in Figure 4.2).
More specifically, $\mathbf{C}_i$ contains terms of the form $C_{i,jk} = \int_{\Gamma_{FSI}} N_j^{\lambda} N_{i,k}^{v} \, d\Gamma$ that extract/interpolate the contribution of the k-th degree of freedom of domain i to the j-th continuity equation [140].
Although not necessary in the context of this work, it is worth underlying that this procedure can be easily extended to the case of completely dissimilar meshes and multiple interfaces between fluid and structure, i.e. considering the case of $\Gamma_{FSI} = \bigcup_{j=1}^{n_i} \Gamma_{FSI,j}$, where n_i is the number of different interfaces.

4.3.2 Time discretization

The integration schemes introduced in Sections 2.4 and 4.2.2 are used for the governing equations of the fluid and solid subdomains, whilst the interface constraint conditions are enforced at the time t^{n+1} for each time step. The system of fully discretized equations for the coupled FSI problem in the case of same time step is then given by:

$$\mathbf{M}_f \mathbf{A}_f^{n+1} + \mathbf{C}_f^T \mathbf{\Lambda}^{n+1} = \mathbf{F}_f^{n+1} \quad \text{in } \Omega_f^{n+1} \tag{4.17}$$
$$\mathbf{M}_\rho^{n+1} \mathbf{R}_f^{n+1} = \tilde{\mathbf{F}}_\rho^0 \quad \text{in } \Omega_f^{n+1} \tag{4.18}$$
$$\mathbf{P}_f^{n+1} = K_f / K'_{0,f} ((\mathbf{R}_f^{n+1} / \mathbf{R}_{0,f})^{K'_{0,f}} - \mathbf{I}) \quad \text{in } \Omega_f^{n+1} \tag{4.19}$$
$$\mathbf{M}_s \mathbf{A}_s^{n+1} - \mathbf{C}_s^T \mathbf{\Lambda}^{n+1} = \mathbf{F}_s^{n+1} \quad \text{in } \Omega_s^{n+1} \tag{4.20}$$
$$\mathbf{C}_f \mathbf{V}_f^{n+1} = \mathbf{C}_s \mathbf{V}_s^{n+1} \quad \text{on } \Gamma_{FSI}^{n+1} \tag{4.21}$$

Applying the aforementioned kinematic splitting on the system (4.17)-(4.21), one can subdivide the set of governing equations in the following steps.

After the solution of equations (4.18)-(4.19) necessary to compute vector $\mathbf{F}_f^{n+1}$, the *free* solution on the two subdomains leads to:

$$\mathbf{M}_f \mathbf{A}_{free,f}^{n+1} = \mathbf{F}_f^{n+1} \quad \text{in } \Omega_f^{n+1} \tag{4.22}$$

$$\mathbf{M}_s \mathbf{A}_{free,s}^{n+1} = \mathbf{F}_s^{n+1} \quad \text{in } \Omega_s^{n+1} \tag{4.23}$$

while the *link* solution is a straightforward efficient solve of a decoupled system of equations of size equal to the number of degrees of freedom on the fluid structure interface:

$$\mathbf{M}_f \mathbf{A}_{link,f}^{n+1} = -\mathbf{C}_f^T \mathbf{\Lambda}^{n+1} \quad \text{on } \Gamma_{FSI}^{n+1} \tag{4.24}$$

$$\mathbf{M}_s \mathbf{A}_{link,s}^{n+1} = \mathbf{C}_s^T \mathbf{\Lambda}^{n+1} \quad \text{on } \Gamma_{FSI}^{n+1} \tag{4.25}$$

The overall acceleration is finally obtained as $A_i^{n+1} = A_{free,i}^{n+1} + A_{link,i}^{n+1}$.
Let us now focus on the interface constraint which is necessary to compute the Lagrange multipliers. Applying a condensation procedure on the *link* problem (for a detailed description see [135, 140]) one can get the following system of interface constrain equations:

$$\mathbf{H}\mathbf{\Lambda}^{n+1} = \mathbf{C}_f \mathbf{V}_{f,free}^{n+1} - \mathbf{C}_s \mathbf{V}_{s,free}^{n+1} \quad \text{on } \Gamma_{FSI}^{n+1} \tag{4.26}$$

where the so-called condensation matrix $\mathbf{H}$ is defined as:

$$\mathbf{H} = \mathbf{C}_f \mathbf{M}_f^{-1} \mathbf{C}_f^T \frac{\Delta t}{2}^{n+1} + \mathbf{C}_s \mathbf{M}_s^{-1} \mathbf{C}_s^T \frac{\Delta t}{2}^{n+1} \tag{4.27}$$

It is worth noticing that in the approach here presented, which couples two explicit codes, and in the case of conforming meshes at the interface, the condensation matrix $\mathbf{H}$ is a square diagonal matrix. Therefore also the kinematic constraint equations become a system of decoupled equations, leading to a fully explicit coupled solver. On the other hand, in the case of non conforming meshes, matrix $\mathbf{H}$ is not diagonal. Consequently, in this latter case, the computation of Lagrange multipliers requires the solution of a linear system. However, this system has small dimensions, since its size is proportional only to the number of degrees of freedom of the fluid structure interface, and a very narrow band of the order of the ratio between the sizes of the element on the two interface sides.

4.3.3 Different time step in the fluid and solid phase (subcycling)

As previously mentioned in Sections 2.4 and 4.2.2, one of the main issues when dealing with explicit integration schemes is represented by the stable time step size which can compromise the efficiency of the solver. Recalling

that the stable time step depends on the mesh size and the material properties (Equation (2.48)), it is crucial for the efficiency of a method which involves different material domains, to have the possibility of adopting different integration time steps on each domain (i.e. performing the so-called subcycling).
The subcycling in the context of the GC method has been introduced in [135]. Let us consider the case in which the fluid domain Ω_f^t has a smaller stable time step $\Delta t_f^{m+1} = t^{m+1} - t^m$, while the solid domain Ω_s^t has a larger time step $\Delta t_s^{n+1} = t^{n+1} - t^n = \sum_{m=0}^{M-1} \Delta t_f^{m+1}$, M being the number of smaller time steps included in the larger one. Following the approach of [135], the interface constraint equations are solved at each time step of the finer time discretization, introducing a linear time interpolation to compute the value of the variables coming from the solid subdomain, which has a larger time step. The linear time interpolation for a generic variable c from two known instants t^n, t^{n+1} to the instant t^m is given by:

$$c^m = c^n \left(\frac{t^{n+1} - t^m}{t^{n+1} - t^n} \right) + c^{n+1} \left(\frac{t^m - t^n}{t^{n+1} - t^n} \right) \tag{4.28}$$

Performing this linear interpolation, let us consider the governing equations of the coupled problem in the case of subcycling (see also Algorithm 3).
The *free* problem reads $\forall n \in \{0, .., N-1\}$:

$$\mathbf{M}_s^{n+1} \mathbf{A}_{free,s}^{n+1} = \mathbf{F}_s^{n+1} \quad \text{in } \Omega_s^{n+1} \tag{4.29}$$

$$\mathbf{M}_\rho^{m+1} \mathbf{R}_f^{m+1} = \tilde{\mathbf{F}}_\rho^0 \quad \text{in } \Omega_f^{m+1}, \quad \forall m \in \{0, .., M-1\} \tag{4.30}$$

$$\mathbf{P}_f^{m+1} = \frac{K_f}{K'_{0,f}} \left[\left(\frac{\mathbf{R}_f^{m+1}}{\mathbf{R}_{0,f}} \right)^{K'_{0,f}} - \mathbf{I} \right] \quad \text{in } \Omega_f^{m+1}, \forall m \in \{0, .., M-1\} \tag{4.31}$$

$$\mathbf{M}_f \mathbf{A}_{free,f}^{m+1} = \mathbf{F}_f^{m+1} \quad \text{in } \Omega_f^{m+1}, \quad \forall m \in \{0, .., M-1\} \tag{4.32}$$

The interface constraint equations become $\forall n \in \{0, .., N-1\}$:

$$\left(\mathbf{C}_f \mathbf{M}_f^{-1} \mathbf{C}_f^T \frac{\Delta t}{2}^{m+1} + \mathbf{C}_s \mathbf{M}_s^{-1} \mathbf{C}_s^T \frac{\Delta t}{2}^{n+1} \right) \mathbf{\Lambda}^{m+1} = \mathbf{C}_f \mathbf{V}_{f,free}^{m+1} - \mathbf{C}_s \mathbf{V}_{s,free}^{m+1}$$
$$\text{on } \Gamma_{FSI}^{m+1}, \quad \forall m \in \{0, .., M-1\} \tag{4.33}$$

The *link* problem is $\forall n \in \{0, .., N-1\}$:

$$\mathbf{M}_s \mathbf{A}_{link,s}^{n+1} = \mathbf{\Lambda}^{n+1} \quad \text{on } \Gamma_{FSI}^{n+1} \tag{4.34}$$

$$\mathbf{M}_f \mathbf{A}_{link,f}^{m+1} = -\mathbf{\Lambda}^{m+1} \quad \text{on } \Gamma_{FSI}^{m+1}, \quad \forall m \in \{0, .., M-1\} \tag{4.35}$$

Finally, the total accelerations are obtained as sum between *free* and *link* solutions $\forall n \in \{0, .., N-1\}$:

$$\mathbf{A}_s^{n+1} = \mathbf{A}_{free,s}^{n+1} + \mathbf{A}_{link,s}^{n+1} \quad \text{in } \Omega_s^{n+1} \tag{4.36}$$

$$\mathbf{A}_f^{m+1} = \mathbf{A}_{free,f}^{m+1} + \mathbf{A}_{link,f}^{m+1} \quad \text{in } \Omega_f^{m+1}, \ \forall m \in \{0, .., M-1\} \tag{4.37}$$

4.3.4 Coupling algorithm stability

When partitioned schemes are employed for FSI problems, it is essential to check possible spurious numerical energy variations at the interface due to the coupling algorithm, as numerical energy generation at the coupling interface can be a source of instability. In [135], it has been proven that the optimal choice of kinematic constraint to be enforced in the GC method is the velocity constraint. In fact, imposing the velocity continuity at the fluid-structure interface, one obtains zero numerical energy variation at the interface when the same time step size is used in the two subdomains, and a numerical energy variation which is strictly dissipative in the case of subcycling. Therefore one can state that the GC is a stable algorithm because no energy is generated at the interface and, consequently, the coupled problem preserves the stability of each individual subdomain.

4.3.5 Fluid-Structure coupled solution scheme

To summarize the main steps of the PFEM-FEM FSI solution scheme, let us consider the general case of FSI problems where the structure is not completely surrounded by the fluid and the interaction interface may change (or vanish) during the fluid flow. In such cases an algorithm for contact detection is required. Nevertheless, the PFEM Alpha-shape (Section 3.2.1.2) combined with its strategy to represent the boundaries (Section 3.2.2) allow for a natural inclusion of such feature in the simulations. A set of ghost boundary nodes is superimposed to the structure surfaces. Unlike the fixed rigid boundary walls case addressed in Section 3.2.2, in FSI problems the ghost nodes are constrained to follow the structural surfaces along their movements/deformations. With this strategy to define the computational domain, let us consider two generic time steps of the analysis (Figure 4.3), that are referred to as CASE-A and CASE-B. The solution scheme of the present method involves the following steps:

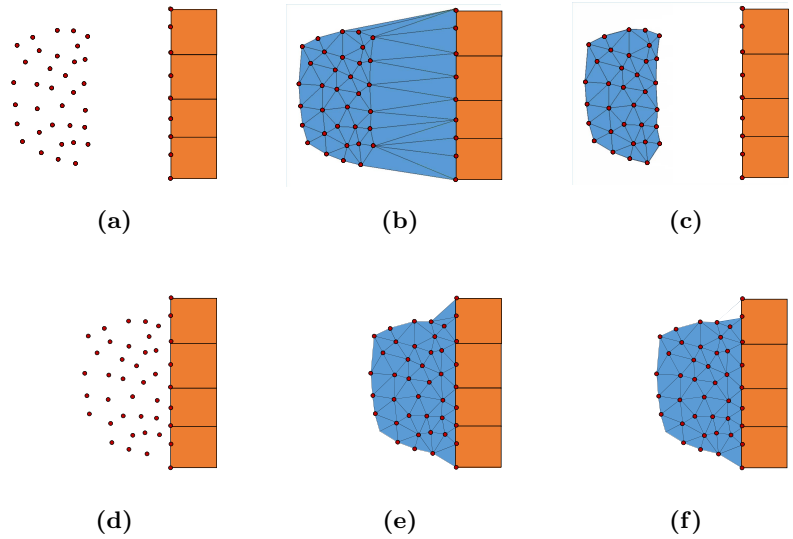


Figure 4.3: Scheme of PFEM-FEM FSI scheme [1]. CASE-A: a)Fluid as a set of particles, structure as a mesh of quadrilateral elements with ghost fluid nodes at its surfaces. (b) Delaunay Triangulation. (c) Application of Alpha-shape method which leads to a decoupled analysis. (d)-(f) Same as (a)-(c). (f) Delaunay Triangulation and Alpha-shape method lead to a coupled analysis.

- Starting from the cloud of fluid nodes (CASE-A: Figure 4.3-a; CASE-B: Figure 4.3-d) the Delaunay Triangulation of the fluid domain is performed (CASE-A: Figure 4.3-b; CASE-B: Figure 4.3-e).
- The Alpha-shape method removes the non-physical elements, if any (CASE-A: Figure 4.3-c; CASE-B: Figure 4.3-e).
- Check if the fluid is in contact with the structure, i.e. if at least one of the ghost particles belongs to the fluid mesh:
  - CASE-A: if the fluid and the structure are not in contact, a decoupled analysis is performed following Algorithms 1 and 2 separately;
  - CASE-B: if the fluid and structure are in contact, the coupled analysis described in this Section and summarized in Algorithm 3 is performed.

Remark: *The case of finer space-time discretization on the fluid side has been presented. Depending on the application, the other choice could be*

Algorithm 3 Fluid-Structure coupled problem:
GC Solution Scheme

for $n = 0, ..., N-1$ **do**
  structure mid-step velocity update
  structure displacement update
  free structure solution: (4.29)
  for $m = 0, ..., M-1$ **do**
    fluid mid-step velocity update
    fluid displacement update
    fluid mass conservation eq. solution: (4.30)
    fluid eq. of state solution: (4.31)
    free fluid momentum eq.s solution: (4.32)
    interpolation of structural nodal velocities: (4.28)
    interface constraint: (4.33)
    link fluid solution: (4.35)
    fluid total acceleration computation: (4.37)
    update $m = m + 1$
  end for
  link structure solution: (4.34)
  structure total acceleration computation: (4.36)
  update $n = n + 1$
end for

preferable but the generalization of the formulation to the other case is straightforward.

4.4 Numerical examples

4.4.1 1D shock pressure wave propagation

A first validation of the present method is obtained considering the 1D problem depicted in Figure 4.4-a. The results of the proposed method are compared with analytical and numerical results presented in [118] using a staggered ALE SPH-FEM coupling. A linear elastic beam clamped at its left end is coupled with a closed rigid tube containing water and fixed at its right end. The initial lengths of the beam and the water tube are $L_s = L_f = 1m$, their height is $h = 0.1m$. The water parameters chosen for this example are: reference density $\rho_{f,0} = 1000kg/m^3$, viscosity $\mu_f = 0Pa \cdot s$ and wave propagation speed $c_f = 1500m/s$. As far as the solid beam is concerned, two different analyses will be performed: a first one with a linear elastic material, and a second one with a bilinear elasto-plastic behaviour. The corresponding parameters are: reference density $\rho_s = 2700kg/m^3$, elastic stiffness modulus $E_{el} = 67.5GPa$, yield stress $\sigma_0 = 6.75MPa$ and tangent

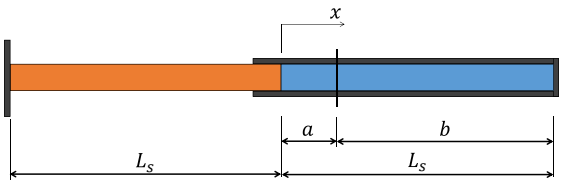


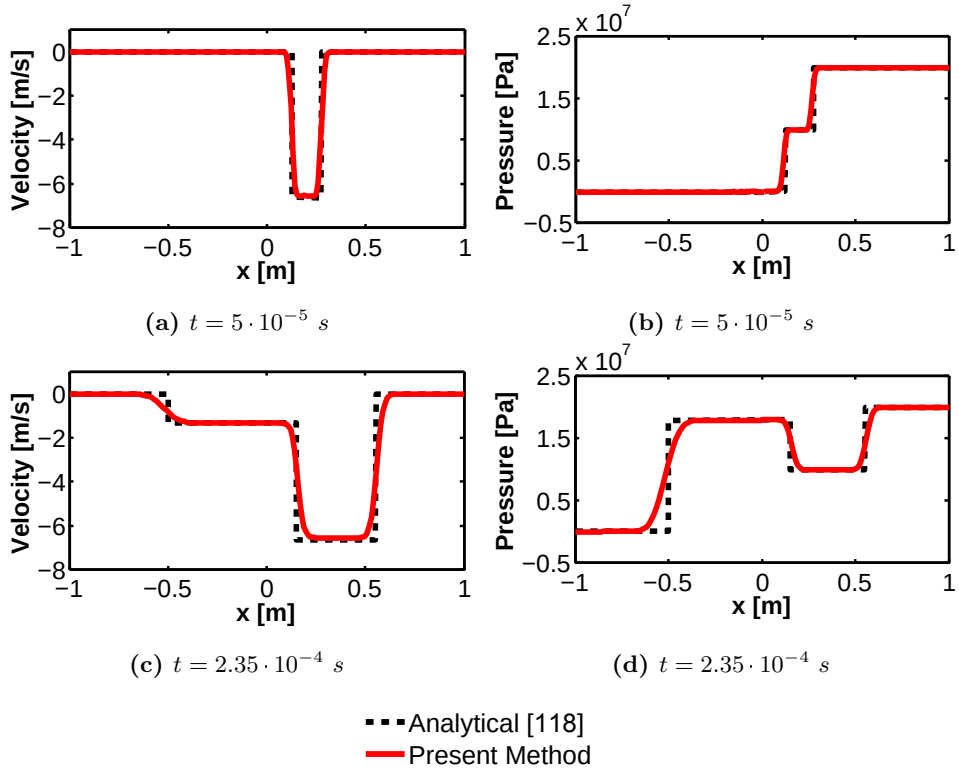
Figure 4.4: 1D shock pressure wave propagation. Geometry of the problem.

modulus $E_{pl} = 6.75GPa$ after the yielding point. Conforming meshes of $60k$ linear triangles and $25k$ 4-nodes quadrilateral elements are used for the fluid and the beam, respectively, leading to a fluid-structure interface discretized by 51 matching nodes on both sides. Consequently, the average stable time steps are $\Delta t_f = 1.91\ 10^{-7}s$ for the fluid domain and $\Delta t_s = 3.83\ 10^{-7}s$ for the structural one.

The initial condition consists of the imposition of a pressure jump on the fluid: $p_0 = 0Pa$ for $x < 0.2L_f$ and $p_0 = 2\ 10^7 Pa$ for $x \geq 0.2L_f$. As the analysis starts the wave propagation in the two subdomains can be observed. The results for the elastic material are presented in terms of velocity and pressure fields profiles at a fixed instant. In Figure 4.5, the profiles are compared with the analytical solution provided in [118] at the instants $t = 5\ 10^{-5}s$ and $t = 2.35\ 10^{-4}s$, showing a very good agreement and proving the capability of the model to reproduce shock pressure waves propagation through the fluid-structure interface.
Figure 4.6 shows the same profiles at $t = 2.35\ 10^{-4}s$ for the case of a bi-linear elasto-plastic beam. One can observe that, as the pressure wave impacts the beam right edge, two distinct waves propagate through the solid: one elastic and one plastic, being the former faster than the latter. The results are compared with the numerical solution presented in [118] and also in this case a good agreement is obtained.

4.4.2 Breaking dam flow through an elastic gate

The example presented in [127] is considered in order to compare the proposed method with experimental and numerical results obtained with a monolithic Lagrangian SPH method. The geometry is shown in Figure 4.7-a: a column of water is initially confined between two vertical walls. The right wall is rigid, while the other has an upper rigid part and a lower deformable one made of rubber. The water column is $0.1m$ wide (B) and $0.14m$ high (H), while the gate is $0.005m$ thick (s) and $0.079m$ high(L). The rubber gate is clamped at its upper side and initially kept fixed by an external rigid support, so that the water is initially in hydrostatic equi-



(a) $t = 5 \cdot 10^{-5}\ s$ **(b)** $t = 5 \cdot 10^{-5}\ s$

(c) $t = 2.35 \cdot 10^{-4}\ s$ **(d)** $t = 2.35 \cdot 10^{-4}\ s$

Figure 4.5: 1D shock pressure wave propagation. Velocity and Pressure fields in the elastic material case at different instants.

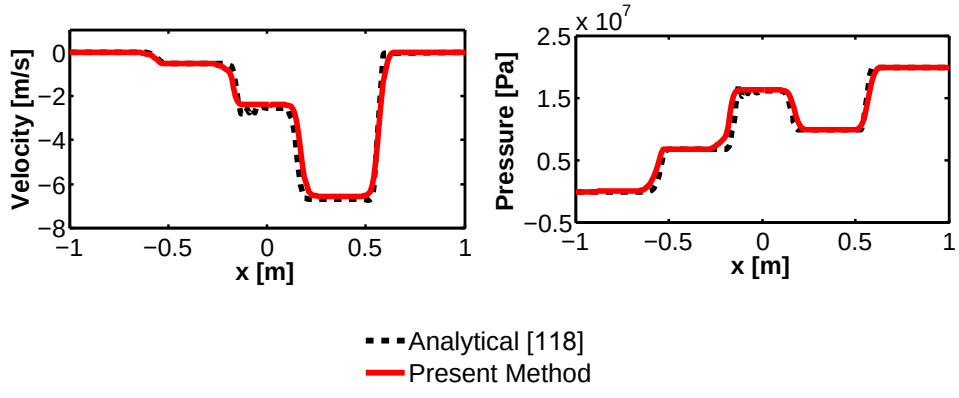
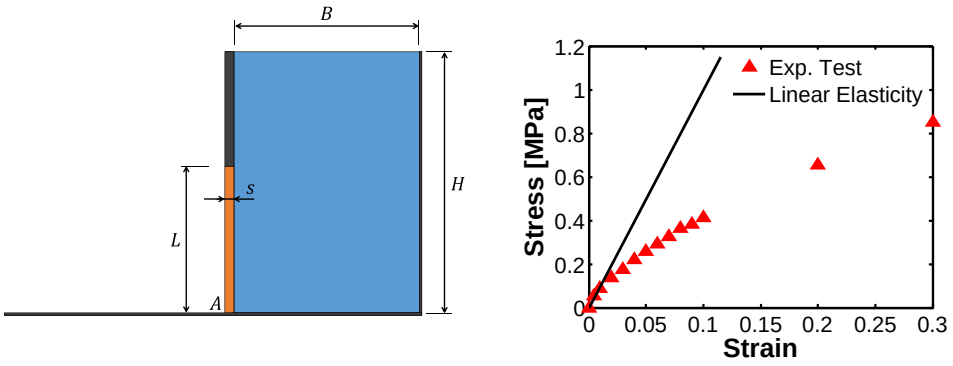


Figure 4.6: 1D shock pressure wave propagation. Velocity and Pressure fields in the elasto-plastic bilinear case material case at $t = 2.35 \cdot\ 10^{-4}\ s$.



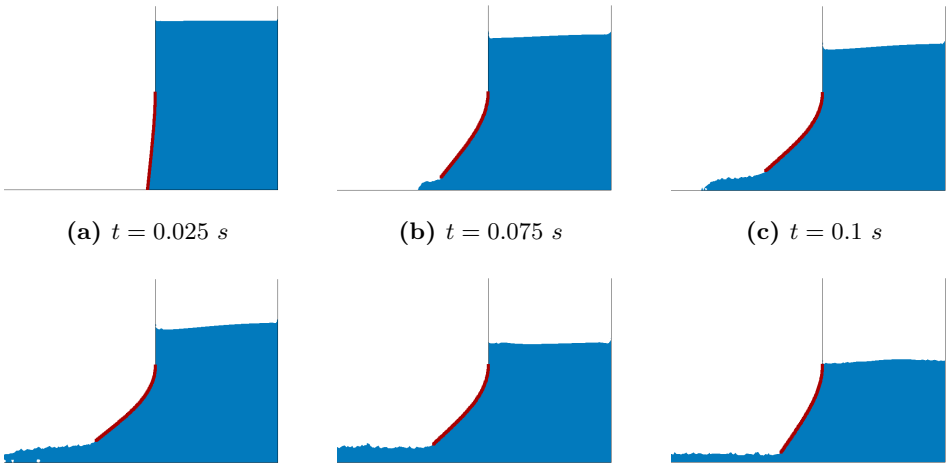
(a) (b)

Figure 4.7: Breaking dam flow through an elastic gate. (a) Geometry of the problem. (b) Tensile experimental test on rubber gate material and linear elastic approximation [2].

librium. At the beginning of the analysis, the support is instantaneously removed and the gate starts deforming under the pressure of water flowing underneath. In [127], an SPH method has been used to model both the structural and fluid parts and the results have been compared with the experimental ones.

In this example, it has been noticed that an accurate description of the fluid flow below the deformable gate requires a smaller mesh size than the one necessary for the structure. Consequently, a mesh of $47k$ linear triangles of average size of 8.0 $10^{-4}m$ is used to model the fluid side, while a mesh of 138 4-nodes plane strain quadrilateral elements of average size of 1.6 $10^{-3}m$ is used to model the deformable gate. The interface is thus discretized by 47 nodes on the structure side and 93 nodes on the fluid one (each structural element side is overlapped to 2 fluid element sides). The material parameters for the water are: reference density $\rho_{f,0} = 1000kg/m^3$ and viscosity $\mu_f = 0.001Pa \cdot s$, while for the wave propagation speed the reduced value of $c_f = 350\ m/s$ is used. Consequently, the average stable time step on the fluid domain is $\Delta t_f = 1\ 10^{-7}s$ while on the structural side it is $\Delta t_s = 1\ 10^{-5}s$.

Figure 4.7-b shows the experimental results of the tensile tests carried out on the rubber material and presented in [2]. One can observe that the behavior is highly non linear. As stated in [124] and [2], a linear elastic analysis using the tangent to the stress-strain curve at the origin (Figure 4.7-b) would underestimate the gate deflection, since it would predict a larger stiffness in the large deformation regime. Therefore the Mooney-Rivlin hyperelastic model implemented in Abaqus/Explicit, together with



(a) $t = 0.025\ s$ **(b)** $t = 0.075\ s$ **(c)** $t = 0.1\ s$

(d) $t = 0.12\ s$ **(e)** $t = 0.2\ s$ **(f)** $t = 0.4\ s$

Figure 4.8: Breaking dam flow through an elastic gate. Snapshots of the simulation at different time instants.

the geometrical non-linearity option, are used for the rubber gate, using as input the set of experimental tensile test data. The remaining rubber material parameters are taken from [127]: density $\rho_s = 1100 kg/m^3$ and Poisson coefficient $\nu_s = 0.4$. Figure 4.8 shows different snapshots of the simulation, while in Figure 4.9 the horizontal and vertical displacement of the lower tip of the elastic gate (point A in Figure 4.7-a) are compared with the experimental and numerical ones presented in [127]. The results depicted in Figure 4.9 show good agreement with the experimental ones, both in terms of the peak value of the displacements and of their final values.

4.4.3 Breaking dam flow over an elastic obstacle

The following example was originally proposed by [110, 142] and then it became a common FSI test case presented in literature with various approaches, for example in [122, 124, 143]. The example is similar to the one presented in Section 3.4.3, but the rigid obstacle is replaced by an elastic one clamped on its lower edge at the middle of the tank. The geometry of the problem is showed in Figure 4.10 and the following geometrical parameters have been used: $L = 0.146m$, $h = 0.08m$ and $s = 0.012m$. The material parameters used for the water are the same as in the previous example, while the ones for the linear elastic solid material are taken from [142]: density

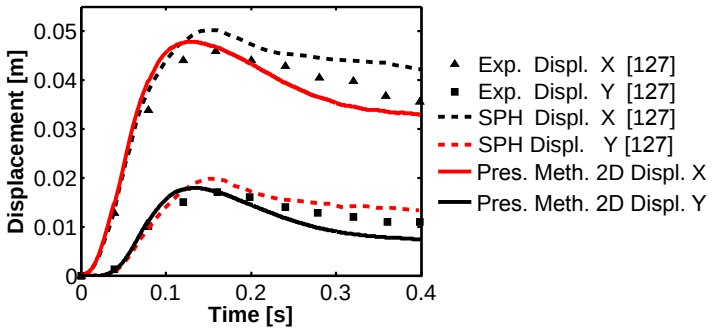


Figure 4.9: Breaking dam flow through an elastic gate. Evolution in time of the horizontal and vertical displacements of the gate tip.

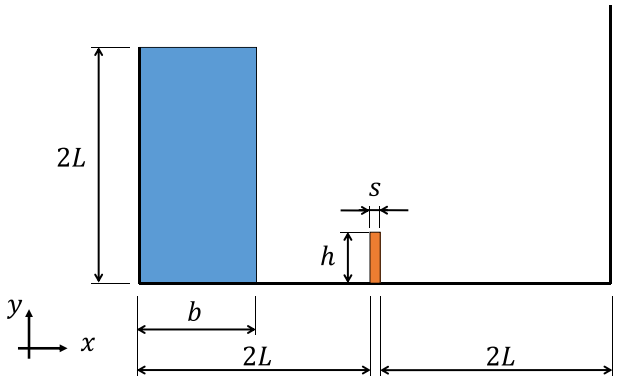
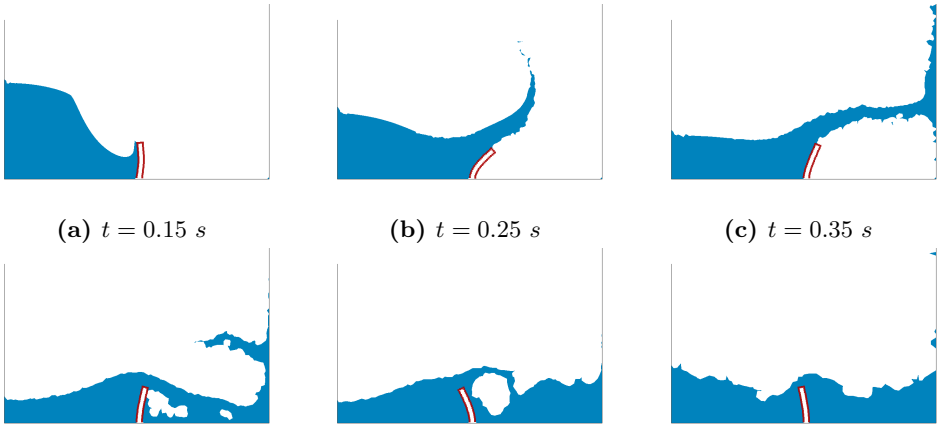


Figure 4.10: Breaking dam flow over an elastic obstacle. Geometry of the problem.



(a) $t = 0.15\ s$ **(b)** $t = 0.25\ s$ **(c)** $t = 0.35\ s$

(d) $t = 0.57\ s$ **(e)** $t = 0.67\ s$ **(f)** $t = 1.0\ s$

Figure 4.11: Breaking dam flow over an elastic obstacle. Snapshots of the simulation at different time instants.

$\rho_s = 2700 kg/m^3$, Young modulus $E_s = 1 MPa$ and Poisson ratio $\nu_s = 0$. The fluid domain is discretized using $6.5k$ linear triangular elements of average size of $0.002m$, while the structural domain has 60 4-nodes quadrilateral elements of average size of $0.004m$. The interface is thus discretized by 44 nodes on the structure side and 87 nodes on the fluid one (each structural element side is overlapped to 2 fluid element sides).

Figure 4.11 represents some snapshots of the results at different time steps. At the beginning of the analysis, the rigid wall sustaining the column of water in hydrostatic equilibrium is instantaneously removed. The water flows in the tank due to gravity and it impacts the elastic obstacle (Figure

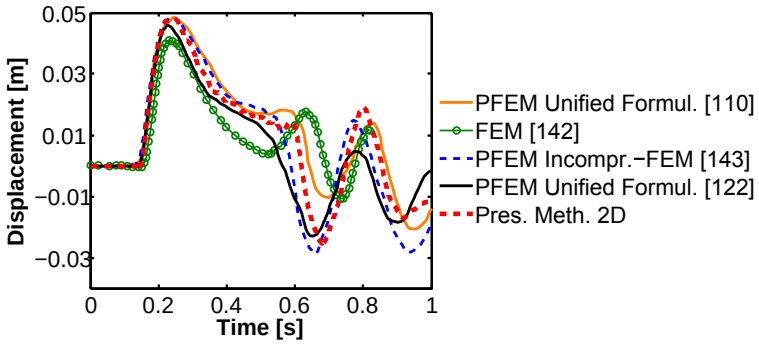
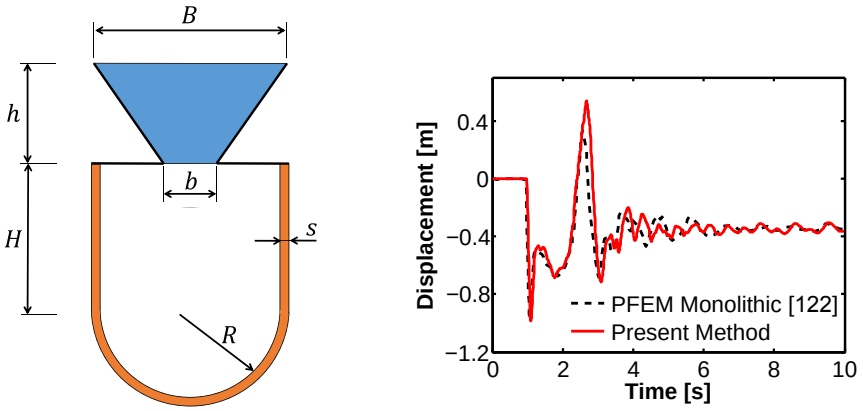


Figure 4.12: Breaking dam flow over an elastic obstacle. Evolution in time of the horizontal displacement of the tip of the obstacle.



(a) (b)

Figure 4.13: Filling of an elastic container. (a) Geometry of the problem. (b) Evolution in time of the vertical displacement of the bottom point of the elastic container.

4.11-a), with the creation of a long wave (Figure 4.11-b). After the impact with the right rigid wall of the tank (Figure 4.11-c), the flow goes back along the bottom and impacts again the elastic obstacle on its right side while the obstacle oscillates (Figure 4.11-d,e,f). The time history of the horizontal displacement of the left upper corner of the elastic obstacle is plotted in Figure 4.12, together with the results presented in [110, 122, 142, 143]. Up to 0.5 s the displacement is qualitatively and quantitatively in good agreement with the reference solutions. More specifically, it is important to note that the first displacement peak is accurately reproduced, which proves the capability of the model to represent the impact of a mass of fluid with a flexible structure and its deformation. After that, in the following oscillations some discrepancies arise and accumulate during time because of the complex dynamics involving breaking waves and splashes.

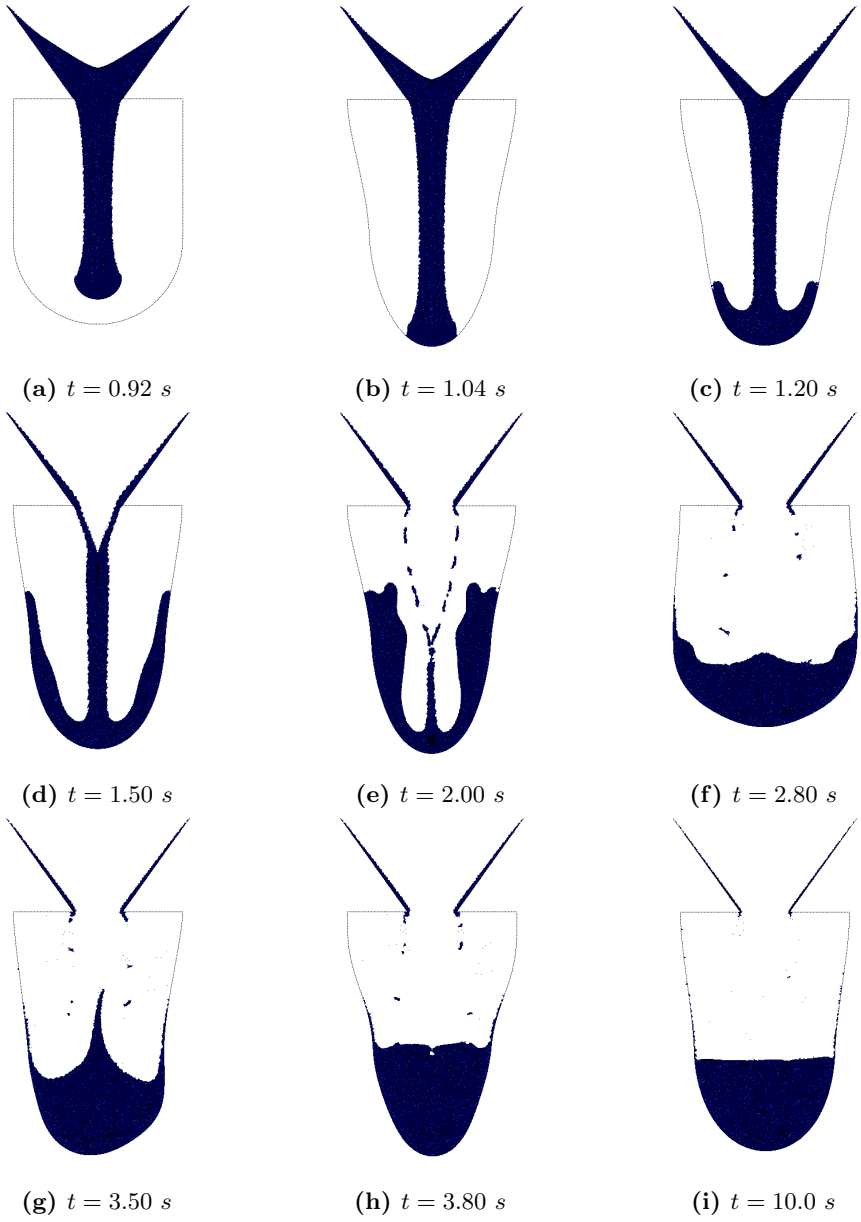
4.4.4 Filling of an elastic container

The example here addressed is taken from [122] where a monolithic unified Lagrangian PFEM is employed. This problem is a further test for the present method on simulations of complex fluid-structure coupling and large displacements of the structure. The geometry is depicted in Figure 4.13-a, while table 4.1 shows the problem data.
A viscous fluid is initially in hydrostatic equilibrium in a funnel-shape rigid container. As the analysis starts, the bottom wall of the container is instantaneously removed, leaving the fluid drop under the gravity load onto

a thin and highly deformable elastic membrane. The snapshots in Figure 4.14 show the initial stretch of the elastic membrane due to the impact, and the following oscillations.
For a quantitative comparison with the numerical results presented in [122], where a monolithic version of the PFEM has been used for both the fluid and the membrane, Figure 4.13-b shows the evolution in time of the vertical displacement of the lowest point of the elastic structure: once again, the agreement between the two curves can be considered very satisfactory, also in view of the high level of complexity of the oscillations after the fluid drop.

Geometry	
h	2.5 m
H	3.75 m
R	2.25 m
b	1.3 m
B	4.87 m
s	0.2 m
Fluid	
Density	1000 kg/m^3
Viscosity	50 $Pa \cdot s$
Bulk Modulus	$1.75 \cdot 10^7$ Pa
# Linear Triangular Elements	28165
# Ghost Nodes at Fluid-Structure Interface	581
Average stable Δt	$4 \cdot 10^{-6}$ s
Structure	
Density	20 kg/m^3
Young Modulus	$2.1 \cdot 10^7$ Pa
Poisson ratio	0.3
# Linear Triangular Elements	1308
# Ghost Nodes at Fluid-Structure Interface	291
Average Stable Δt	$8.6 \cdot 10^{-6}$ s

Table 4.1: Filling of an elastic container. Geometry and materials parameters.



(a) $t = 0.92\ s$ **(b)** $t = 1.04\ s$ **(c)** $t = 1.20\ s$

(d) $t = 1.50\ s$ **(e)** $t = 2.00\ s$ **(f)** $t = 2.80\ s$

(g) $t = 3.50\ s$ **(h)** $t = 3.80\ s$ **(i)** $t = 10.0\ s$

Figure 4.14: Filling of an elastic container. Snapshots of the simulation at different time instants.

5 Lagrangian-Eulerian enforcement of non-homogeneous BC

5.1 Introduction

In the fluid mechanics framework, a wide range of different boundary conditions can be employed in the solution of the Navier-Stokes equations, including the no-slip and free surface conditions already addressed in Chapter 3, but also inflow/outflow conditions, slip and symmetry. Following the general definition introduced in [144], in fluid mechanics the fluid domain boundaries can be divided into real or fictitious boundaries. The former is represented by physical limits of the fluid domain where velocities or tractions can be defined (e.g. boundary walls, free surfaces). The latter is related to the fact that very frequently in fluid mechanics there are problems involving open domains (e.g. the flow past an aircraft wing or a bridge pile) or closed channel flows (e.g. flows in pipes). In such problems the boundaries are simply limits of the computational domain and they are therefore fictitious. Among them, relevant cases are represented by the inflow and outflow boundaries, which are fictitious sections of the computational domain where a velocity or traction profiles can be imposed describing the state of the fluid particles entering/exiting the computational domain [144].

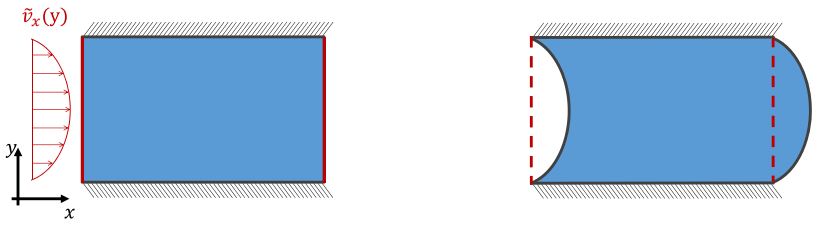
Among the family of real boundaries, a particular case is represented by the slip boundary conditions. More in details, while the no-slip boundary condition between the fluid and a basal surface or a confining wall is macroscopically accepted in most cases of fluid dynamics [145,146], there are several applications involving fluids or fluid-like flows on solid surfaces where this condition is not realistic [147, 148]. Relative fluid-wall slip can be observed in many industrial applications such as polymer extrusion [149–151], or in applications involving granular flows, such as debris flows or silos discharge [152–154]. Navier slip boundary conditions define a linear correlation

between the slip velocity and the basal tangential stress, through a parameter summarizing the interaction between the fluid and the wall surface. This parameter controls the amount of slip at the wall surface, which can range from the no-slip condition with maximum tangential stress, to free slip with zero tangential stress. More realistic slip laws take into account the fact that, in general, slip occurs only when a critical threshold on the tangential stress is exceeded (see e.g. [94]).

Another example of fictitious boundaries is represented by the symmetry boundary conditions. In three-dimensional engineering applications, the size of the numerical model often creates difficulties due to the computational cost. It is then of primary importance to exploit possible symmetries in the problem geometry. The boundary conditions on symmetry planes can be viewed as a special case of the slip conditions discussed above, where penetration of the material particles through the symmetry plane is prevented, while free slip is allowed in directions tangent to the symmetry plane.

As previously underlined, the PFEM and more in general the Lagrangian methods are particularly effective in problems characterized by the presence of moving boundaries, free surfaces and moving fluid-structure interfaces, since their evolution is naturally tracked by the motion of the fluid nodes with no need for *ad hoc* algorithms (Section 3.1). On the contrary, the PFEM strategy to model the fluid boundary walls, i.e. the ghost nodes described in Section 3.2.2, requires complex implementations in many other typical CFD problems where non-homogeneous boundary conditions are involved, such as the previously mentioned inflow, outflow, slip and symmetry conditions [155, 156]. Let us consider a generic example of inflow condition represented in Figure 5.1. When a velocity profile is imposed on the ghost nodes lying on a boundary (see Figure 5.1-a), their Lagrangian motion leads to the loss of the boundary definition (see Figure 5.1-b). The same issue arises in all the non-homogeneous boundary conditions e.g. slip and symmetry.

This problem arises in all the fully Lagrangian methods. In SPH methods, the inflow issue is usually dealt with by attaching an inflow zone upstream of the computational domain. All the particles in the inlet zone maintain their characteristics to meet the imposed boundary condition at the entrance [157, 158]. This technique should include the possibility to insert new particles in the inflow region and to remove particles at an outflow boundary [157,158]. The drawbacks of this approach are the difficult a-priori definition of the inflow region dimension and the increased computing time due the larger computational domain. Furthermore, the interpolation procedure, which is at the basis of the SPH scheme, makes the implementation of this kind of boundary conditions rather difficult [158] and instability of



(a) time t^n **(b)** time t^{n+1}

Figure 5.1: Example of the inflow and outflow boundary conditions in a Lagrangian framework.

the inflow particles can occur [159]. A detailed discussion of inflow boundary conditions in SPH and ISPH (Incompressible Smoothed Particle Hydrodynamics) methods can be found in [159].

Also in the Material Point Method (MPM) inflow and outflow are usually modeled with a large reservoir to supply the domain of interest with material points [160]. Very recently, a technique to use only one layer of elements in the inflow/outflow region has been presented in [161]. On the contrary, the non-homogeneous boundary conditions are easily treated with Eulerian approaches, because the nodes are kept fixed and they do not move according to the fluid velocity.

To combine the complementary capabilities of Lagrangian and Eulerian approaches to model the wide range of boundary condition in fluid mechanics, in this chapter a novel technique to impose non-homogeneous boundary conditions in a PFEM Lagrangian framework is proposed, based on a mixed Lagrangian-Eulerian (LE) description of the mesh nodes kinematics. All the internal and free surface mesh nodes are treated as Lagrangian while the mesh nodes on non-free boundaries are treated as Eulerian. The fluid velocity at the Eulerian nodes can be different from zero without altering the geometric definition of the boundary because of their Lagrangian motion. Furthermore, the presence of Eulerian boundary nodes greatly simplifies the implementation of slip laws, even for models which are more realistic than the linear Navier condition here considered.

5.2 LE Balance equations: strong and weak form

Let us now recall the Navier-Stokes Equations (2.1)-(2.2) to cast them in a more general Lagrangian-Eulerian framework. Applying the Eulerian

derivation rule (2.7) to the total time derivative, one gets to:

$$\rho_f \left[\frac{d\mathbf{v}_f}{t} + (\mathbf{c}_f \cdot \nabla_{\mathbf{x}})\, \mathbf{v}_f \right] = \nabla_{\mathbf{x}} \cdot \boldsymbol{\sigma}_f + \rho_f \mathbf{b}_{\mathbf{f}} \quad \text{in } \Omega_f^t \times [0,T] \tag{5.1}$$

$$\left[\frac{dp_f}{dt} + (\mathbf{c}_f \cdot \nabla_{\mathbf{x}})\, p_f \right] + K_f (\nabla_{\mathbf{x}} \cdot \mathbf{v}_f) = 0 \quad \text{in } \Omega_f^t \times [0,T] \tag{5.2}$$

where $\mathbf{c}_f = \mathbf{c}_f(\mathbf{x}, t)$ is the convective velocity defined as:

$$\mathbf{c}_f(\mathbf{x}, t) = \mathbf{v}_f(\mathbf{x}, t) - \mathbf{r}_f(\mathbf{x}, t) \tag{5.3}$$

$\mathbf{r}_f(\mathbf{x}, t)$ denoting the mesh velocity. These equations are the same as in the Arbitrary Lagrangian Eulerian (ALE) formulations [31], in which the Eulerian and Lagrangian ones are obtained as special cases setting $\mathbf{c}_f = \mathbf{v}_f$ or $\mathbf{c}_f = \mathbf{0}$, respectively.

The weak forms of Equations (5.1) and (5.2) can be obtained adopting the same approach described in Section 2.3, namely introducing a vector test function $\mathbf{w}_f \in S_0^v$, a scalar test function $q_f \in S^p$ and integrating over the domain Ω_f^t:

$$\begin{aligned} &\int_{\Omega_f^t} \mathbf{w}_f \cdot \rho_f \left[\frac{d\mathbf{v}_f}{dt} + (\mathbf{c}_f \cdot \nabla_{\mathbf{x}})\, \mathbf{v}_f \right] d\Omega = \int_{\Omega_f^t} p_f\, (\nabla_{\mathbf{x}} \cdot \mathbf{w}_f)\, d\Omega + \\ &- \int_{\Omega_f^t} 2\mu_f \nabla_{\mathbf{x}} \mathbf{w}_f : \dot{\boldsymbol{\varepsilon}}_f\, d\Omega + \\ &+ \int_{\Omega_f^t} \mathbf{w}_f \cdot \rho_f \mathbf{b}_f d\Omega + \int_{\Gamma_{N,f}^t} \mathbf{w}_f \cdot \mathbf{h}_f\, d\Gamma \quad \forall\, \mathbf{w}_f \in S_0^v \end{aligned} \tag{5.4}$$

$$\int_{\Omega_f^t} q_f \left[\frac{dp_f}{dt} + (\mathbf{c}_f \cdot \nabla_{\mathbf{x}})\, p_f \right] d\Omega + \int_{\Omega_f^t} K_f q_f\, (\nabla_{\mathbf{x}} \cdot \mathbf{v}_f)\, d\Omega = 0 \quad in\ \Omega_f^t \times [0,T] \tag{5.5}$$

5.3 LE Initial conditions and homogeneous essential boundary conditions

Let us recall the set of initial and boundary condition already introduced in Section 2.1. Initial conditions specify the values of velocity and pressure at the beginning of the analysis:

$$\mathbf{v}_f(\mathbf{X}, t = 0) = \mathbf{v}_f^0 \quad \text{in } \Omega_f^0 \tag{5.6}$$

$$p_f(\mathbf{X}, t = 0) = p_f^0 \quad \text{in } \Omega_f^0 \tag{5.7}$$

where $\mathbf{v}_f^0$ and p_f^0 are assigned functions.

In the case of homogeneous essential boundary conditions, the boundary $\Gamma_f^t = \partial\Omega_f^t$ is divided into two non-overlapping subsets $\Gamma_{D,f}^t$ and $\Gamma_{N,f}^t$, such that $\Gamma_{D,f}^t \cup \Gamma_{N,f}^t = \Gamma_f^t$ and $\Gamma_{D,f}^t \cap \Gamma_{N,f}^t = \emptyset$ and the following conditions can be imposed:

$$\mathbf{v}_f(\mathbf{x},t) = \mathbf{0} \quad \text{on } \Gamma_{D,f}^t \times [0,T] \tag{5.8}$$

$$\boldsymbol{\sigma}_f \cdot \mathbf{n}_f = \mathbf{h}_f(\mathbf{x},t) \quad \text{on } \Gamma_{N,f}^t \times [0,T] \tag{5.9}$$

where $\mathbf{h}_f(\mathbf{x},t)$ is an assigned function. Let us underline that in this more general LE framework, the symbol t at exponent of the boundary Γ_f, is intended to emphasize the Lagrangian nature of the boundary, evolving in time together with the particles motion. As a consequence of this, the natural boundary condition (5.9) has also a Lagrangian nature, meaning that the tractions $\mathbf{h}_f(\mathbf{x},t)$ are applied to the evolving free surface $\Gamma_{N,f}^t$.

The boundary conditions in (5.8) are treated without difficulties in the PFEM, since the particles on the boundary $\Gamma_{D,f}^t$ cannot move and the boundary $\Gamma_{D,f}^t$ remains defined throughout the history of motion. On the other hand, natural boundary conditions of the type in (5.9) when $\Gamma_{N,f}^t$ represents a free surface are straightforwardly treated in Lagrangian methods as the PFEM.

5.4 LE Non-homogeneous boundary conditions

5.4.1 Inflow and outflow Eulerian boundary conditions

A relevant class of problems in CFD is related to flow in pipes, channels or open domains. To reduce the computational effort, these problems are typically analysed considering only the portion of interest of the domain, with proper inflow/outflow conditions at the inlet/outlet cross-sections. In these cases, the boundary conditions are written as:

$$\mathbf{v}_f(\mathbf{x},t) = \tilde{\mathbf{v}}_f(\mathbf{x},t) \quad \text{on } \bar{\Gamma}_{D,f} \tag{5.10}$$

$$\boldsymbol{\sigma}_f \cdot \mathbf{n}_f = \mathbf{h}_f(\mathbf{x},t) \quad \text{on } \bar{\Gamma}_{N,f} \tag{5.11}$$

where $\tilde{\mathbf{v}}_f(\mathbf{x},t)$, $\mathbf{h}_f(\mathbf{x},t)$ are prescribed functions assuming non-zero values at the inlet/outlet sections and $\bar{\Gamma}_{D,f}$, $\bar{\Gamma}_{N,f}$ are boundaries fixed in time, typically coinciding with inlet or outlet sections of the computational domain.

A purely Lagrangian description is not suitable for an efficient modeling of these boundary conditions, since one would need to consider a greatly enlarged computational domain with inflow/outflow auxiliary regions. These

regions have to include all the fluid nodes that, during the simulated time interval, will flow through the considered pipe portion, enlarging enormously the computational domain. Another difficulty is related to the accurate imposition of the generic and possibly time-dependent boundary condition at the inlet cross-section. The particles which flow through the inlet cross-section are not perfectly aligned on the inlet line (in 2D problems) or plane (in 3D problems). Consequently, *ad hoc* interpolation algorithms, which can introduce inaccuracies and numerical oscillations, are needed.

To overcome this problem, a new strategy is proposed which exploits the flexibility of the Lagrangian-Eulerian formulation by assuming Eulerian ghost boundary nodes on $\bar{\Gamma}_{D,f}$ with convective velocity $\mathbf{c}_f = \mathbf{v}_f$ and, hence, zero mesh velocity $\mathbf{r}_f$. On the contrary, all the internal nodes are treated as Lagrangian, i.e. with a zero convective velocity $\mathbf{c}_f$. The way in which the non-homogeneous Eulerian velocity conditions are enforced on the Eulerian ghost boundary nodes is straightforward and is qualitatively illustrated in Figure 5.2. A set of Eulerian ghost nodes is positioned at the inlet cross-

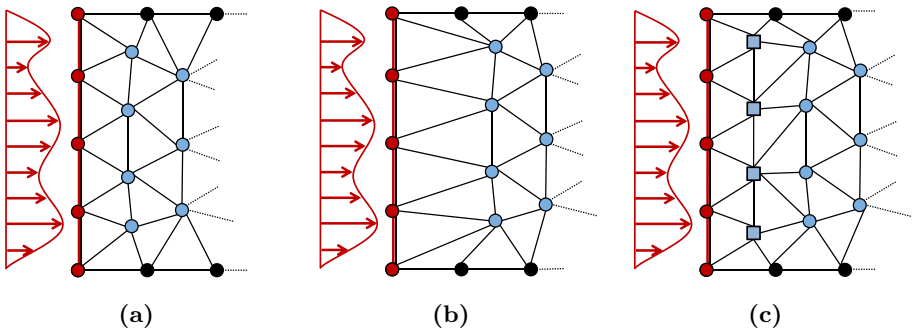


Figure 5.2: Use of mixed Lagrangian-Eulerian description to model the inflow condition. (a) A set of Eulerian ghost nodes (red dots) is placed on the inflow where any time dependent velocity or pressure profile can be imposed. (b) Consequent motion of neighbor nodes (blue dots) with enlargement of an element layer. (c) Insertion of nodes (squares) on the centroids of the large elements.

section (Figure 5.2-a). At these nodes, any time dependent velocity profile (Dirichlet conditions) or time dependent pressure distribution (Neumann conditions) can be imposed, with no need for an enlarged computational domain to define an inflow region. The non-zero velocities or pressures at the inlet nodes activate the motion of the internal neighbor nodes, causing the enlargement of the first layer of boundary elements (Figure 5.2-b). When these elements become too large, a new node is inserted at their centroid

(Figure 5.2-c). The nodes addition/reallocation is indeed a natural procedure in the PFEM framework (see Section 3.2.1.3). Similarly, regarding the outflow cross-section, a set of fixed Eulerian nodes can be used to impose the required conditions on velocity or pressure. When the Lagrangian nodes during the motion get too close to the Eulerian ones, they are removed from the mesh. Moreover, in many relevant situations where no particular condition are imposed at the outlet cross-section (i.e. outflow as a standard free-surface boundary), it not necessary to use Eulerian nodes, but it is sufficient to remove from the mesh the Lagrangian nodes which have passed the outflow condition, i.e. which are outside the computational domain. Remarkably, in both cases one can use the removed nodes at the outflow condition and directly insert them in the first layer of elements at the inlet cross-section. This creates an efficient management of the mesh, keeping the number of nodes constant and limiting the computational domain to the considered channel section.

Remark 2. *In the ALE framework, an additional equation is typically needed to govern the mesh velocity evolution [31]. In the present approach such equation is not required because the convective velocity will be directly computed from the boundary conditions on the constrained part of the boundary and will be zero elsewhere.*

5.4.2 Slip boundary conditions

Different types of slip conditions can be described by choosing the so called slip length as an interface parameter [147]. The geometrical interpretation of the slip length can be appreciated in Figure 5.3, showing a generic distribution of tangential velocity in the region close to a horizontal wall with relative slip. The slip length h_{slip} is defined as the distance from the plate to the point with zero tangential velocity, obtained with a linear extrapolation of the tangential velocity profile, enforcing the continuity at the wall of the derivative $\partial v_x/\partial y$. The two previously mentioned limit cases of no-slip and free slip are recovered for $h_{slip} = 0$ and $h_{slip} \to \infty$, respectively. Moreover, also the case of negative h_{slip} may happen [147] if the no-slip condition propagates through a small layer of fluid close to the wall.

A new partition of the fluid boundary si introduced, considering three non overlapping subsets $\Gamma^t_{D,f}$, $\Gamma^t_{N,f}$, $\Gamma^t_{S,f}$ such that $\Gamma^t_{D,f} \cup \Gamma^t_{N,f} \cup \Gamma^t_{S,f} = \Gamma^t_f$, $\Gamma^t_{S,f}$ being the part of the boundary where slip conditions are imposed. On $\Gamma^t_{D,f}$ and $\Gamma^t_{N,f}$ the standard Dirichlet and Neumann boundary conditions (5.10)-(5.11) or (5.8)-(5.9) are imposed. Note that even in the cases where the slip surface is fixed in time, the slip boundary may evolve, as its wet part may be not constant in time. On the slip boundary $\Gamma^t_{S,f}$, the relative

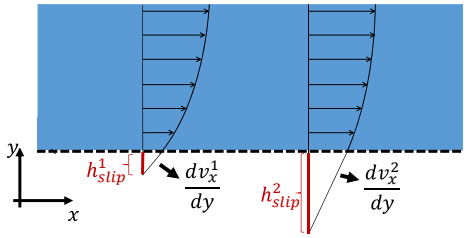


Figure 5.3: Linear Navier slip condition. Different velocity profiles according to the h_{slip} parameter. $h^2_{slip} > h^1_{slip}$.

velocity $\mathbf{v}_{slip}$ between the fluid and the wall and the tangential interface tractions vector $\mathbf{t}_f$ are defined as:

$$\mathbf{v}_{slip} = (\mathbf{I} - \mathbf{n}_f \otimes \mathbf{n}_f)\, \mathbf{v}_f \qquad \mathbf{t}_f = (\mathbf{I} - \mathbf{n}_f \otimes \mathbf{n}_f)\, (\boldsymbol{\sigma}_f \mathbf{n}_f) \tag{5.12}$$

Then, the linear Navier slip boundary condition prescribes:

$$\mathbf{v}_{slip} = -\beta_{slip} \mathbf{t}_f \quad on\ \Gamma^t_{S,f} \tag{5.13}$$

together with the through-the-wall no-flux condition:

$$\mathbf{v}_f \cdot \mathbf{n}_f = 0 \quad on\ \Gamma^t_{S,f} \tag{5.14}$$

In the previous expressions, the parameter β_{slip} is defined as $\beta_{slip} = h_{slip}/\mu_f$, where μ_f is the fluid dynamic viscosity.

To include the Navier Slip condition in the variational statement (5.4)-(5.5), the new boundary integral computed on $\Gamma^t_{S,f}$ is added on the right-hand side of the weak form of the momentum equation (5.4):

$$\int_{\Gamma^t_{S,f}} \mathbf{w} \cdot \left(\boldsymbol{\sigma}_f \mathbf{n}_f - \mathbf{t}_f - \frac{1}{\beta_{slip}} \mathbf{v}_{slip} \right) d\Gamma \tag{5.15}$$

where the test functions $\mathbf{w}$ are now required to satisfy the additional condition $\mathbf{w} \cdot \mathbf{n}_f = 0$ on $\Gamma^t_{S,f}$. The first term in the brackets stems from the Green formula applied to Equation (5.4) and the other terms represent the imposition of the slip condition (5.13) [94]. Furthermore, substituting the definition (5.12) of the tangential tractions $\mathbf{t}_f$, one can obtain:

$$\begin{aligned} \int_{\Gamma^t_{S,f}} \mathbf{w} \cdot (\boldsymbol{\sigma}_f \mathbf{n}_f - \mathbf{t}_f)\, d\Gamma &= \int_{\Gamma^t_{S,f}} \mathbf{w} \cdot (\mathbf{n}_f \otimes \mathbf{n}_f)\, (\boldsymbol{\sigma}_f \mathbf{n}_f)\, d\Gamma = \\ &= \int_{\Gamma^t_{S,f}} (\mathbf{w} \cdot \mathbf{n}_f)\, \mathbf{n}_f \cdot \boldsymbol{\sigma}_f \mathbf{n}_f\, d\Gamma = 0 \end{aligned} \tag{5.16}$$

so that the additional integral to be added on the right-hand side of the weak form of the momentum equation (5.4) becomes:

$$\int_{\Gamma^t_{S,f}} \mathbf{w} \cdot \left(-\frac{1}{\beta_{slip}} \mathbf{v}_{slip} \right) d\Gamma \tag{5.17}$$

together with the no-flux constraint $\mathbf{v}_f \cdot \mathbf{n}_f = 0$ on $\Gamma^t_{S,f}$.

As in the previous case, a set of Eulerian ghost nodes is placed on the slip surface allowing the imposition of the non-homogeneus velocity profile given by the slip law.

Remark 3. *For the sake of clarity, the simple linear Navier condition is here considered. A more realistic Navier slip law is presented in [94], taking into account the fact that usually slip activates only when a critical threshold of the tangential stress is exceeded. Also in this case, the implementation of other types of slip laws is facilitated in the mixed Lagrangian-Eulerian method here considered, thanks to the presence of Eulerian boundary nodes.*

5.4.3 Symmetry boundary conditions

The symmetry boundary condition is characterized by zero normal velocity and zero value of its gradient trough the symmetry plane. For the considered finite element model, based on linear triangles or tetrahedra, this is equivalent to enforce a zero velocity component in the direction $\mathbf{n}_f$ normal to the symmetry plane $\Gamma^t_{D,Sym}$:

$$\mathbf{v}_f \cdot \mathbf{n}_f = 0 \quad \text{on } \Gamma^t_{D,Sym} \tag{5.18}$$

which is identical to condition (5.14) on the slip boundary. As a matter of fact, the symmetry condition can be seen as equivalent to a slip condition with infinite slip length, which may lead to the same type of difficulties already discussed for a purely Lagrangian treatment. In particular, the non-zero tangential velocities very soon lead to a non-uniform distribution of ghost boundary nodes on $\Gamma^t_{D,Sym}$, compromising the geometric definition of the symmetry plane, or generating overly distorted elements. In contrast, with the proposed technique, in which the Eulerian ghost boundary nodes remain fixed, an accurate definition of the symmetry plane is preserved throughout the analysis. In practice, the symmetry condition consists of enforcing the constraint (5.18) in strong form, as in the slip case, while Equation (5.13) reduces to $\mathbf{t}_f = \mathbf{0}$ on $\Gamma^t_{D,Sym}$, which implies that the integral (5.17) does not appear in the weak form.

5.5 LE Space and time discretization

Adopting the same approach of Section 2.3, let us introduce the finite element discretization (2.22)-(2.23) and the additional discretization for the convective velocity:

$$\mathbf{c}_f^h(\mathbf{x},t) = \mathbf{N}_f^v(\mathbf{x})\mathbf{C}_f(t) \tag{5.19}$$

where $\mathbf{C}$ is the vector of convective nodal velocities. The semi-discretized momentum and mass equations can be written as:

$$\mathbf{M}_f \frac{d\mathbf{V}_f}{dt} = -\left(\mathbf{K}_{\mu,f} + \mathbf{K}_{c,f}^v\right)\mathbf{V}_f + \mathbf{D}_f^T\mathbf{P}_f + \mathbf{F}_{ext,f} \tag{5.20}$$

$$\mathbf{M}_\rho \frac{d\mathbf{P}_f}{dt} = -\mathbf{K}_{c,f}^p\mathbf{P}_f - K_f\ \mathbf{D}_f\mathbf{V}_f \tag{5.21}$$

where $\mathbf{K}_c^v$ and $\mathbf{K}_c^p$ contain the convective terms related to the time derivatives of velocity and pressure, while the other matrices are defined in Section 2.3. More in particular, the following definitions hold:

$$\mathbf{K}_{c,f}^v = \int_{\Omega_f^t} \mathbf{N}_f^{vT}\mathbf{G}^v\, d\Omega$$

$$\mathbf{K}_{c,f}^p = \int_{\Omega_f^t} \mathbf{N}_f^{pT}\mathbf{G}^p\, d\Omega$$

Let us focus on the 3D case: for every e element of the mesh, the elemental matrices $\mathbf{G}_e^v$ and $\mathbf{G}_e^p$ can be defined in terms of nodal blocks:

$$\mathbf{G}_e^v = \left[\mathbf{G}_{e,1}^v, \ldots, \mathbf{G}_{e,n_n}^v\right] \tag{5.22}$$

$$\mathbf{G}_e^p = \left[\mathbf{G}_{e,1}^p, \ldots, \mathbf{G}_{e,n_n}^p\right] \tag{5.23}$$

being n_n the number of elemental nodes (i.e. $n_n = 4$ for tetrahedra). Each $\mathbf{G}_{e,i}^v$ block is a diagonal (3×3) matrix:

$$\mathbf{G}_{e,i}^v = diag\left[\left(\mathbf{C}^T\mathbf{N}^{vT}\mathbf{L}_i\right)_1, \left(\mathbf{C}^T\mathbf{N}^{vT}\mathbf{L}_i\right)_2, \left(\mathbf{C}^T\mathbf{N}^{vT}\mathbf{L}_i\right)_3\right] \tag{5.24}$$

where the following matrix of shape function derivatives is defined:

$$\mathbf{L}_i = \begin{bmatrix} \frac{\partial N_{i,1}^v}{\partial x} & \frac{\partial N_{i,2}^v}{\partial x} & \frac{\partial N_{i,3}^v}{\partial x} \\ \frac{\partial N_{i,1}^v}{\partial y} & \frac{\partial N_{i,2}^v}{\partial y} & \frac{\partial N_{i,3}^v}{\partial y} \\ \frac{\partial N_{i,1}^v}{\partial z} & \frac{\partial N_{i,2}^v}{\partial z} & \frac{\partial N_{i,3}^v}{\partial z} \end{bmatrix} \tag{5.25}$$

Finally, each $\mathbf{G}^v_{e,i}$ block is given by a single component:

$$\mathbf{G}^p_{e,i} = \mathbf{C}^T \mathbf{N}^{vT} \mathbf{L}^p_i \tag{5.26}$$

where $\mathbf{L}^p_i$ is introduced:

$$\mathbf{L}^p_i = \left[\frac{\partial N^p_i}{\partial x}, \frac{\partial N^p_i}{\partial y}, \frac{\partial N^p_i}{\partial z} \right]^T \tag{5.27}$$

When present, the Navier slip conditions lead to an additional contribution to the matrix $\mathbf{K}_f$, namely $\mathbf{K}_f = \mathbf{K}_{\mu,f} + \mathbf{K}^v_{c,f} + \mathbf{K}_{S,f}$, with $\mathbf{K}_{S,f}$ defined as:

$$\mathbf{K}_{S,f} = \int_{\Gamma^t_{S,f}} \mathbf{Q}^T \mathbf{N}^{vT}_f \mathbf{S}_f \mathbf{N}^v_f \mathbf{Q}^v \, d\Gamma \tag{5.28}$$

In the previous expression, $\mathbf{Q}$ is the orthogonal matrix rotating the velocity components from the global reference system to the one locally normal to $\Gamma^t_{S,f}$ and the diagonal matrix $\mathbf{S}_f$ is defined as $\mathbf{S}_f = diag\,[0, 1/\beta_{slip}, 1/\beta_{slip}]$. In the present explicit solver the no-flux condition (5.14) is strongly imposed at the node level.

Let us now introduce the Central Difference Scheme already described in Section 2.4: for every time step Δt^{n+1} of the analysis, the following systems of fully decoupled equations are solved:

- Update the mid-step velocity: $\mathbf{V}_f^{n+\frac{1}{2}} = \mathbf{V}_f^n + \frac{1}{2}\mathbf{A}_f^n \Delta t^{n+1}$
- Update the displacement $\mathbf{U}_f^{n+1} = \mathbf{U}_f^n + \mathbf{V}_f^{n+\frac{1}{2}} \Delta t^{n+1}$
- Compute pressure field - Eq. (5.21):
  $\mathbf{M}^l_\rho \mathbf{P}_f^{n+1} = \mathbf{M}^c_\rho \mathbf{P}_f^n - \Delta t^{n+1} \left[\mathbf{C}_f \mathbf{P}_f^n + K_f \, \mathbf{D}_f \mathbf{V}_f^{n+\frac{1}{2}} \right]$
- Compute the acceleration - Eq. (5.20): $\mathbf{A}_f^{n+1} = (\mathbf{M}^l_f)^{-1} \mathbf{F}_f^{n+1}$
- Update the velocity (if necessary): $\mathbf{V}_f^{n+1} = \mathbf{V}_f^{n+\frac{1}{2}} + \frac{1}{2}\mathbf{A}_f^{n+1} \Delta t^{n+1}$

where $\mathbf{F}_f^{n+1}$ is now the right-hand-side of equation (5.20) discretized in time. The system of fully discretized governing equations is solved using the Particle-Finite Element Method described in Section 3. Instead of its classical purely Lagrangian formulation, in the present approach the described mixed Lagrangian-Eulerian description of the mesh nodes kinematics is adopted. All the internal fluid nodes are Lagrangian, i.e. characterized

by $\mathbf{C}_f^n = \mathbf{0}$, while the ghost nodes employed to define the boundary Γ_f^t are Eulerian, i.e. characterized by $\mathbf{C}_f^n = \mathbf{V}_f^n$. The velocity of the Eulerian ghost nodes is different from zero without altering the geometric definition of the boundary, as it would be in the case of a Lagrangian motion.

Depending on the considered Dirichlet $\Gamma_{D,f}^t$ or Neumann $\Gamma_{N,f}^t$ conditions, on the ghost nodes one can easily impose complex and possibly time dependent distributions of velocity $\tilde{\mathbf{v}}_f(\mathbf{x}, t)$ defined in Equation (5.10), or tractions $\mathbf{h}_f(\mathbf{x}, t)$ defined in Equation (5.11). In the particular case of a no-slip condition prescribed on the boundary $\Gamma_{D,f}^t$, the standard Lagrangian description of the ghost node kinematics is recovered.

From the point of view of the computational effort, it must be underlined that this choice requires the additional assembly of matrices $\mathbf{K}_{c,f}^v$ and $\mathbf{K}_{c,f}^p$, with standard elemental finite element procedures, only on the elements which include Eulerian nodes, thus on a small portion of the overall domain. In all the elements including only internal nodes, or boundary nodes with homogeneous Dirichlet conditions, these matrices vanish.

Remark 4. *In principle, the LE description could be extended to all the nodes of the fluid domain, with a mesh which moves independent of the fluid velocity, as in the case of a standard ALE formulation [162]. However, the ALE mesh movement may not be sufficient to accommodate the complex evolution of free surface and fluid-structure interfaces typically addressed by the PFEM approach. To this purpose, the remeshing strategy based on Lagrangian nodes seems more effective. Furthermore, the choice to introduce Eulerian nodes on the boundary only where needed, avoids the introduction of the convective terms on the whole mesh and the need for the additional ALE equations to govern the velocity of the mesh nodes.*

5.6 Numerical examples

5.6.1 Lid-driven cavity flow

Let us consider the Lid-driven cavity flow presented in [3]. Figure 5.4-a represents the geometry of the problem: a 2D square of side $L = 1$ m contains a Newtonian fluid initially at rest, characterized by density $\rho_{f,0} = 1000$ kg/m^3 and dynamic viscosity $\mu_f = 1$ Pa s. The fluid is discretized with a mesh of approximately 23 k elements. On the three lower sides (blue sides in Figure 5.4-a) a no-slip condition is imposed, while on the upper side (red side in Figure 5.4-a) a unit, purely tangential velocity is imposed, i.e. $v_x = \bar{V} = 1$ m/s and $v_y = 0$ m/s. The resulting Reynolds number is $Re = 1000$.

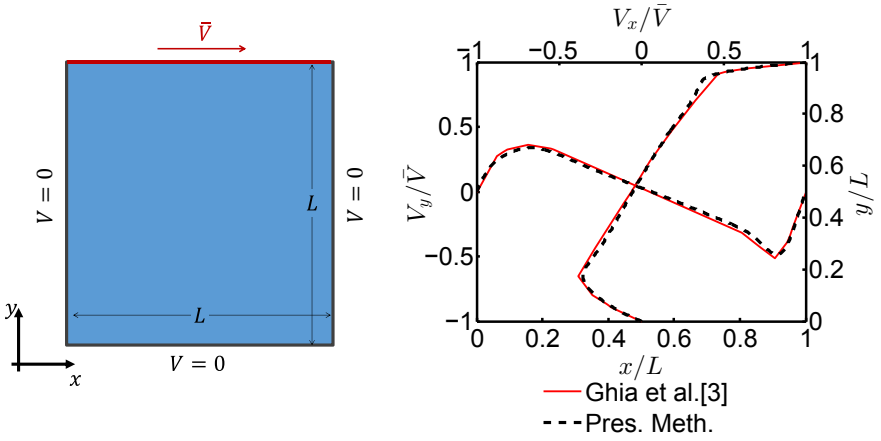


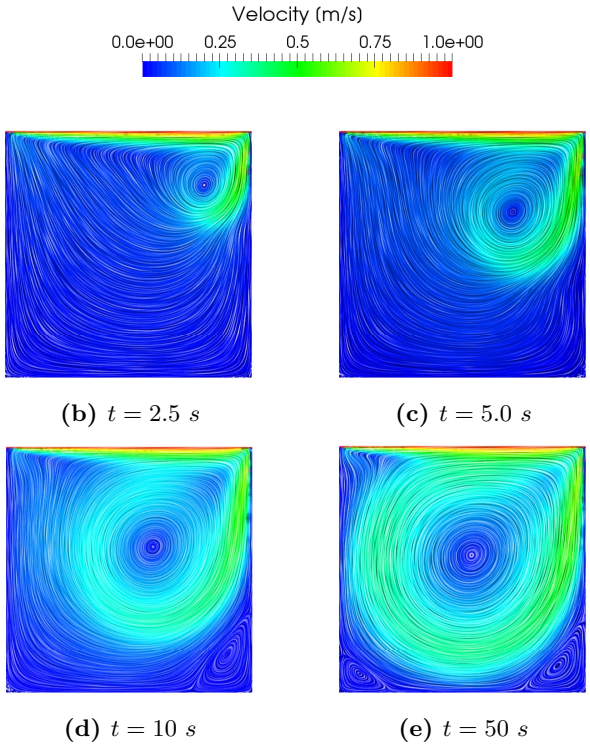
Figure 5.4: Lid-driven cavity flow. (a) Geometry of the problem. (b) Horizontal and vertical velocity profiles at $x/L = 0.5$ and $y/L = 0.5$, respectively. Comparison with the results in [3]

It should be noted that modeling this boundary condition in a purely Lagrangian framework is not straightforward: assuming a horizontal Lagrangian motion on the upper boundary, the particles would approach the right wall creating very distorted elements that can slow down the analysis and introduce numerical issues in the solver. With the present approach a line of fixed Eulerian nodes is used to model the upper boundary, on which the two velocity components can be easily imposed.

For a quantitative validation, let us consider the velocity profile at the steady state. Figure 5.4-b plots the horizontal and vertical velocity profiles at two lines passing through the center of the cavity: $y/L = 0.5$ and $x/L = 0.5$, respectively. The comparison with the results presented in [3] shows a good agreement. The gradual development of the vortex inside the cavity due to the tangential upper velocity can be observed in Figure 5.5, showing the velocity profile and the related streamlines at different instants of the analysis.

5.6.2 3D Symmetric Poiseuille flow in a rectangular pipe

The geometry depicted in Figure 5.6 is considered to assess the computational gain that can be obtained exploiting the problem symmetry in a 3D case. Let us consider a pipe with rectangular cross section filled with a Newtonian fluid characterized by density $\rho_{f,0} = 0.1$ kg/m^3 and dynamic viscosity $\mu_f = 0.01$ Pa s. The pipe dimensions are $l = 1$ m, $2b = 0.3$ m



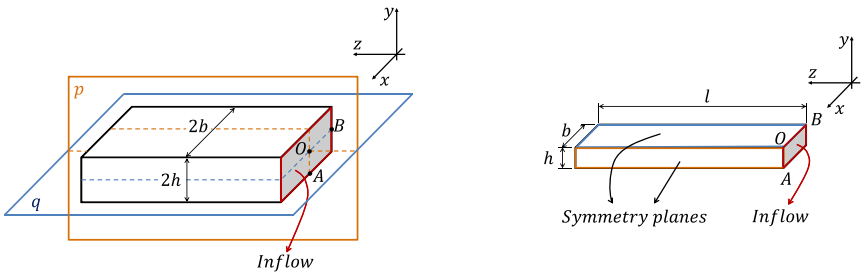
(b) $t = 2.5\ s$ **(c)** $t = 5.0\ s$

(d) $t = 10\ s$ **(e)** $t = 50\ s$

Figure 5.5: Lid-driven cavity flow. Snapshots of velocity streamlines at different time steps.

and $2h = 0.1$ m. The two planes of symmetry p ($x = 0.15$ m) and q ($y = 0$ m) are depicted in Figure 5.6-a. No slip boundary conditions are imposed on the external walls of the pipe. A constant pressure $\tilde{p} = 8$ Pa is imposed as Neumann Boundary condition at the fixed Eulerian nodes at the inlet cross-section, namely at $z = 0$ m, generating a laminar flow in the pipe.

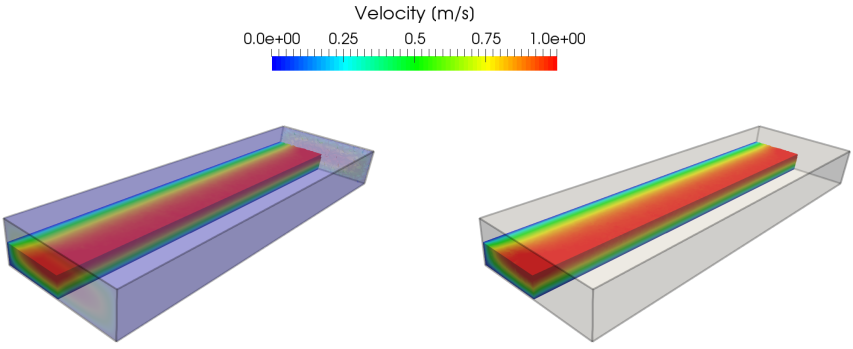
Two analyses have been performed. The first one considers the whole pipe without exploiting the symmetry planes. A mesh of approximately $320k$ tetrahedra is obtained, leading to an overall computational time of 14 hours. The second analysis instead considers the reduced computational domain depicted in Figure 5.6-b, representing one quarter of the pipe exploiting the two planes symmetry, i.e. $y = 0$ m and $x = 0.15$ m. On the symmetry planes, fixed Eulerian nodes are employed to impose the absence of flux across the plane, namely $v_y = 0$ m/s and $v_x = 0$ m/s, respectively. In particular, on the line of intersection in between the two planes representing the center of the pipe, only the z component of the velocity can be different from zero. The same average element size used for the first analysis is



(a) **(b)**

Figure 5.6: 3D Symmetric Poiseuille flow in a rectangular pipe. (a) Geometry of the pipe with two planes of symmetry. (b) Reduced computational domain exploiting the symmetry.

also employed on this reduced computational domain, leading to a mesh of $78k$ elements. The analysis duration is 5 hours and 14 minutes. Figure 5.7 shows a first qualitative comparison between the two analyses: the contour plot obtained with a double cut of the full-pipe analysis (Figure 5.7-a) is quantitatively the same to the one obtained with the quarter-pipe analysis (Figure 5.7-b).



(a) **(b)**

Figure 5.7: 3D Symmetric Poiseuille flow. Velocity contour plot at the steady state. (a) Results of the whole pipe analysis: double cut to highlight the velocity field. (b) Results of the quarter pipe analysis.

The numerical results are validated against the analytical solution considering the z component of the velocity profile on the mid plane $x = 0.15$ m at the steady state. More in details, on a line in the y direction sufficiently far from the inflow cross-section, the analytical parabolic velocity profile of

the Poiseuille flow, i.e.

$$v_z(y) = \frac{1}{2\mu_f}\left(\frac{\partial p_f}{\partial z}\right)\left(h^2 - y^2\right) = 400\left(0.05^2 - y^2\right)\frac{m}{s}$$

is recovered.

Figure 5.8 compares the analytical velocity profile with the results obtained with the two numerical analyses. One can note a good agreement of both analysis with the theoretical solution, and a substantial overlapping of the numerical curves. However, the second analysis is 2.67 times faster, a good example of how the symmetry could be exploited for a significant reduction of the computational cost, which can be crucial, especially in 3D applications. While this has been easily obtained employing the Eulerian boundary nodes, a Lagrangian tangential motion of the particles on the symmetry planes would lead to a non-uniform distribution of the moving particles on the symmetry planes. This can compromise the boundary geometric definition or lead to distorted elements that can introduce numerical issues in the solver.

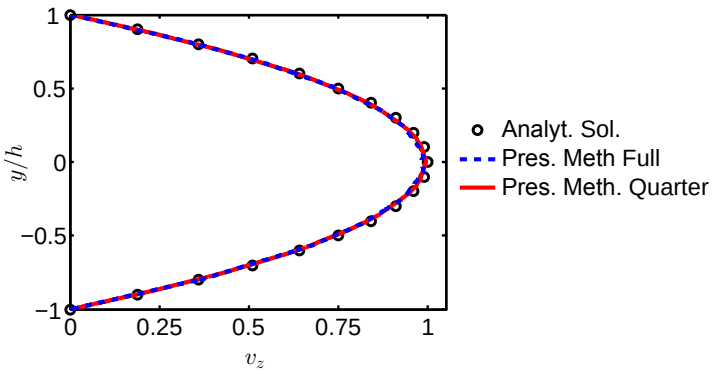


Figure 5.8: 3D Symmetric Poiseuille flow. Velocity $v_z(y)$ profile on the plane $x = 0.15m$.

5.6.3 Poiseuille flow with Navier slip at the boundary walls

In this test, a Poiseuille flow with linear Navier slip on the pipe walls is considered. The results are validated through the comparison with the analytical solution for the present case provided in [163]. Figure 5.9 illustrates the problem geometry. In a 2D pipe of length $L = 0.1m$ and height $2H = 0.01$ m, a Newtonian fluid characterized by density $\rho_{f,0} = 1$ kg/m^3 and dynamic viscosity $\mu_f = 0.001$ Pa s is considered.

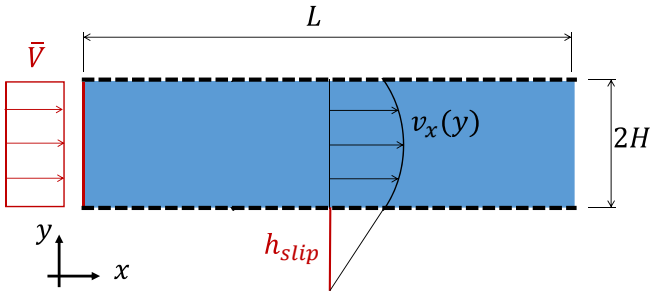


Figure 5.9: Poiseuille flow with Navier slip. Geometry of the problem.

A constant uniform velocity $\bar{V} = 1$ m/s is imposed at the inlet section, modeled by means of Eulerian boundary nodes with non-homogeneous Dirichlet conditions. The pipe walls are modeled by means of Eulerian boundary nodes at which the Navier slip law described in Section 5.4.2 is imposed. In particular three different values of slip length are tested: $h^1_{slip} = 1 \cdot 10^{-4}$ m, $h^2_{slip} = 5 \cdot 10^{-3}$ m and $h^3_{slip} = 5 \cdot 10^{-2}$ m. A mesh of of approximately 25 k elements has been used for this test.

The analytical solution for the parabolic velocity profile on a cross-section sufficiently far from the pipe inlet is [163] :

$$v_x(y) = \frac{p_x}{2\mu_f}\left(H^2 - y^2\right) + \frac{h_{slip}}{\mu_f} p_x H \tag{5.29}$$

where p_x is defined as:

$$p_x = \bar{V}\left(\frac{H^2}{3\mu_f} + \frac{h_{slip}}{\mu_f} H\right)^{-1} \tag{5.30}$$

Figure 5.10 shows the comparison of the analytical and numerical results in terms of normalized velocity profile for the three slip lengths considered. One can observe that in the three different slip regimes which span from an almost no-slip condition (h^1_{slip}) to an almost free-slip on the pipe walls (h^3_{slip}), the numerical results are in good agreement with the theoretical ones.

5.6.4 Circular Couette flow with slip

A circular Couette flow with Navier slip boundary conditions [4] is considered to test the validity of the proposed approach with Eulerian nodes in the case of conditions imposed on curved boundaries, since it has been reported that the no-flux condition (5.14) may lead to the so-called Babuska's

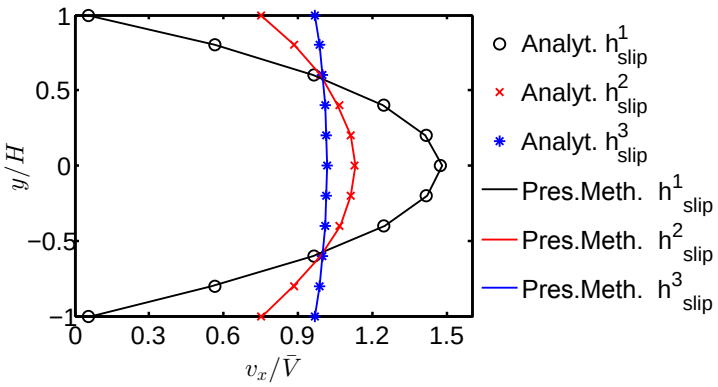


Figure 5.10: Poiseuille flow with Navier slip. Velocity profiles with different interface slip lengths h^i_{slip}.

paradox [164], with the numerical solution not converging to the real one in the presence of curved boundaries and polygonal finite elements. A weak imposition of the constraint has been suggested by [94,165] to overcome this problem. This test confirms that the explicit scheme here employed allows to implement the constraint in its strong form without any numerical issues.

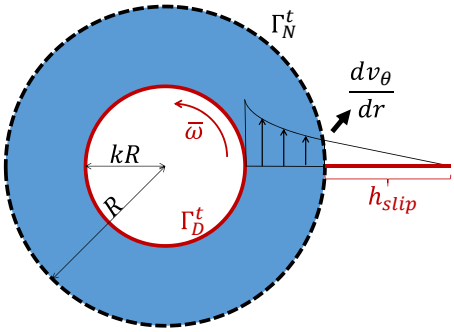


Figure 5.11: Circular Couette flow with slip. Geometry of the problem.

The problem geometry is illustrated in Figure 5.11. Inside two concentric circular boundaries of radii $R = 1$ m and $kR = 0.5$ m, respectively, a Newtonian fluid characterized by density $\rho_{f,0} = 10$ kg/m^3 and viscosity $\mu_f = 1$ Pa s is initially at rest. The inner circle $\Gamma^t_{D,f}$ is rotating at a prescribed angular velocity $\bar{\omega} = 1$ rad/s. This condition is modelled through fixed Eulerian boundary nodes with non homogeneous Dirichlet condition:

$$\mathbf{v}_f = \bar{\omega} \times \mathbf{x} \quad on\ \Gamma^t_{D,f} \tag{5.31}$$

Such condition is strongly imposed on each boundary node of the inner

circle. On the outer circle $\Gamma^t_{S,f}$ Navier slip conditions with different values of h_{slip} are imposed exploiting fixed Eulerian nodes. The rotation of the inner boundary activates a circular laminar flow inside the fluid domain. Because of the problem symmetry, the radial velocity component v_r is zero, while the analytical solution for the circumferential velocity $v_\theta(r)$ at the steady state is derived in [4] as:

$$v_\theta(r) = \frac{k^2 \bar{\omega} R}{1 - k^2 + 2B} \left(\frac{R}{r} - \frac{r}{R} \right) \tag{5.32}$$

where the factor B is defined as $B = h_{slip}/kR$. Figure 5.12 shows the

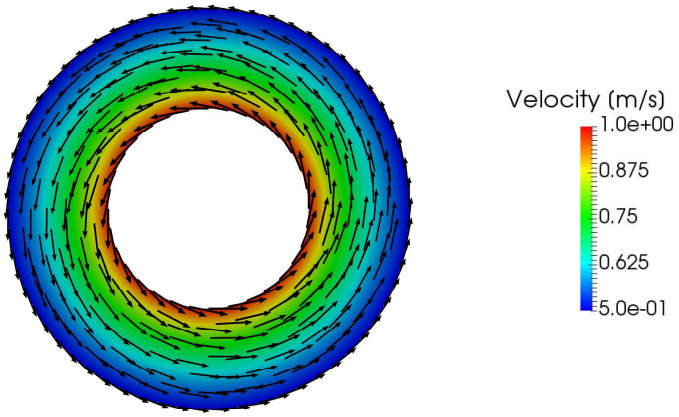


Figure 5.12: Circular Couette flow with slip. Contour plot of the velocity field at the steady state for the case of $h_{slip} = 0.5$ m.

contour plot of the velocity field at the steady state, highlighting the velocity vectors which are purely circumferential. For a quantitative validation, the numerical results in terms of v_θ for different values of h_{slip} are compared with the corresponding analytical solutions in Figure 5.13, showing a perfect match with the theoretical velocity profile also in the present case of curved boundaries.

5.6.5 Turek-Hron FSI Benchmark

The well known numerical example presented in [9] is here addressed. This problem is typically used for validation of Fluid Structure Interaction (FSI) approaches [81,134,166–168]. The geometry is depicted in Figure 5.14 while the data are listed in the table in Figure 5.15: a rigid cylinder of radius R is fixed inside a two-dimensional pipe filled with a Newtonian fluid. An elastic cantilever beam, modelled through Abaqus/Explicit is clamped to

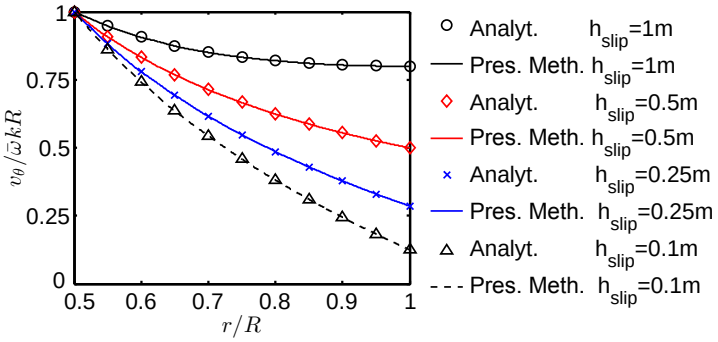


Figure 5.13: Circular Couette flow with slip. Comparison between numerical and theoretical [4] velocity profiles for different values of h_{slip}.

the cylinder. A parabolic velocity profile is imposed at the pipe inlet, with a gradual amplification to avoid numerical artifacts in the initial transient part of the analysis:

$$\tilde{v}_x(y,t) = \left[1.5\bar{V}\frac{y\,(2H-y)}{H^2}\right]\; f(t) \tag{5.33}$$

where $\bar{V}$ is a parameter depending on the considered test case and $f(t)$ is the amplification function defined for the initial velocity increase:

$$\begin{aligned} f(t) &= (1-\cos(\pi t/2))/2 \quad if \quad t \leq 2\ s \\ f(t) &= 1 \qquad\qquad\qquad\quad if \quad t > 2\ s \end{aligned} \tag{5.34}$$

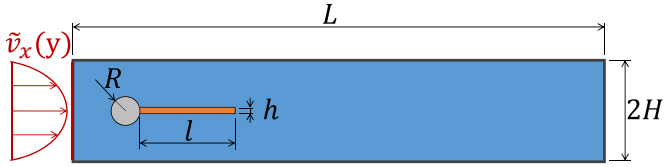
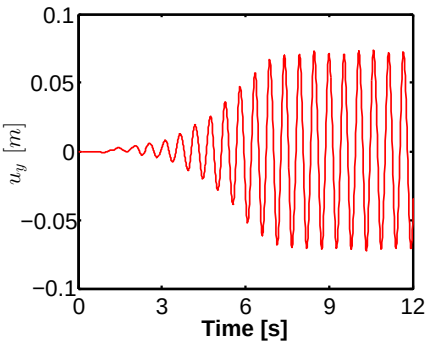


Figure 5.14: Turek-Hron FSI Benchmark. Geometry of the problem.

The laminar flow generated by the inflow condition makes the beam start oscillating, with a complex interaction involving vortex shedding and large structural displacements. The so called 'FSI2' and 'FSI3' test cases presented in [9] with the problem parameters listed in the table in Figure 5.15 are here considered. The latter case is particularly challenging for FSI solvers, because the combination of the model parameters (mainly the same value of structural and fluid densities) is responsible for the added mass

effect (Section 4.1), that can lead to lack of convergence or instabilities for FSI solvers. None of these numerical issues has been observed with the present PFEM-FEM FSI method, thanks to the strong coupling provided by the Gravouil and Combescure algorithm.

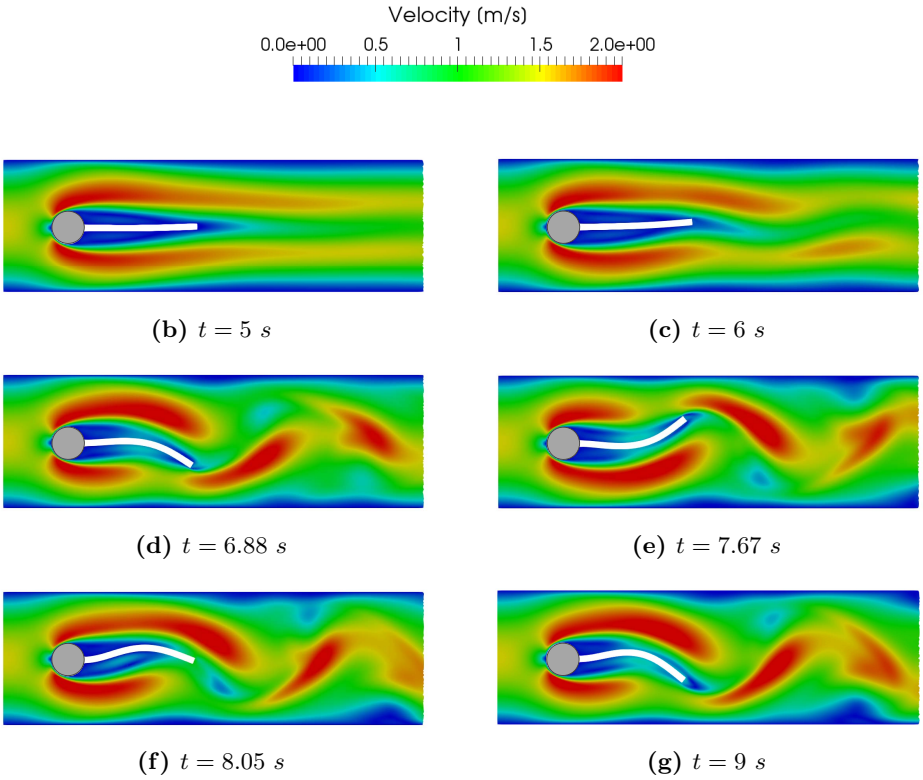


Geometry & Data		
L $[m]$	2.5	
$2H$ $[m]$	0.41	
l $[m]$	0.35	
h $[m]$	0.02	
R $[m]$	0.05	
$\rho_{f,0}$ $[kg/m^3]$	1000	
$\mu_f[Pa\ s]$	1	
	FSI2	FSI3
ρ_s $[kg/m^3]$	10000	1000
E_s $[MPa]$	1.4	5.6
ν_s	0.4	0.4
$\bar{V}$ $[m/s]$	1	2
Re	100	200

Figure 5.15: Turek-Hron FSI Benchmark. Left: time hystory of the vertical displacement of the beam tip. Right: parameters of the FSI2 and FSI3 test cases.

In this problem, a purely Lagrangian description would suffer from the inflow/outflow modelling difficulties described in Section 5.4.1. Introducing a line of Eulerian ghost nodes to model the inlet cross-section of the pipe allows to naturally impose the prescribed time-dependent parabolic velocity profile.

Let us focus on the FSI2 test case: Figure 5.16 represents some snapshots of the results of the analysis, highlighting the velocity contour plot on the fluid side. The graph in Figure 5.15 shows the time history of the vertical displacements of the tip of the elastic beam. One can observe an initial transient phase with increasing oscillations until a sort of steady state is reached, with oscillations of constant amplitude and frequency. Similar results have been obtained for the FSI3 test case. For a quantitative validation, in the Table 5.1 the amplitude (Δu_y) and frequency (f_s) of the steady state oscillations are compared with the results presented in the literature [9]. The comparisons with the reference values show a good agreement for the two test cases. The small discrepancies in the FSI2 and FSI3 test cases are smaller or of the same order of magnitude with respect to other published results [81, 166–168].



(b) $t = 5\ s$ **(c)** $t = 6\ s$

(d) $t = 6.88\ s$ **(e)** $t = 7.67\ s$

(f) $t = 8.05\ s$ **(g)** $t = 9\ s$

Figure 5.16: Turek-Hron FSI Benchmark. Snapshots of fluid velocity contours at different time steps.

Results					
		$\Delta u_y\ [m]$	err.	$f_s[1/s]$	err. [%]
FSI2	[9]	0.161	–	2.0	–
	PFEM	0.1584	1.6%	1.93	3.5%
FSI3	[9]	0.0687	–	5.3	–
	PFEM	0.0663	3.5%	5.4	1.8%

Table 5.1: Turek-Hron FSI Benchmark. Results and comparison with [9] in terms of amplitude Δu_y and frequency f_s of the steady state beam tip oscillations.

This example highlights the already commented advantages of the strategy here presented. On the one hand, the Lagrangian framework is particularly effective for the description of the fast and complex evolution of the fluid-structure interface. On the other hand, the Eulerian nodes allow to an

effective and natural enforcement of boundary conditions that are not trivial in Lagrangian approaches, such as the inflow condition. Finally, it provides a further validation on the PFEM-FEM FSI coupled approach described in Chapter 4 on a challenging test case.

6 Efficient runtime Mesh smoothing for 3D explicit PFEM

6.1 Issue of runtime remeshing in Lagrangian approaches

The PFEM method described in Section 3.2 is characterized by a Galerkin finite element approach coupled with the frequent runtime use of the Delaunay remeshing algorithm to guarantee a high quality of the mesh. Runtime remeshing in Lagrangian approaches is particularly delicate when an explicit solver is used. In implicit approaches, an unconditionally stable time integration allows to use reasonably large time step sizes also in the case of badly shaped elements. In contrast, in explicit approaches overly distorted elements would dramatically reduce the stable time step size (Equation (2.48)) and, therefore, they should be avoided in new meshes generated during the analysis. It must be underlined that the presence of only one bad quality tetrahedron is sufficient to strongly decrease the time step size. This requires a reliable mesh generation strategy, capable to produce regular meshes and fast enough to be repeatedly used during the simulation. These two requirements are particularly demanding when dealing with complex three-dimensional geometries.

Delaunay mesh generators, described in Section 3.2.1.1, have proved to be robust and versatile and particularly suited for applications involving frequent remeshing [88, 169]. In two-dimensions the Delaunay triangulation guarantees remarkable optimality properties, such as the minimization of the maximum radius of an element circumcircle and the maximization of the minimum angle among all the elements. However, in three-dimensions it loses some of these properties [170], allowing for badly shaped tetrahedra, namely the so-called sliver elements. These elements can have dramatic consequences when an explicit solver is employed, leading to almost vanishing time step sizes. Moreover, it is worthmentioning that such badly shaped

tetrahedra can be critical also for implicit solvers, leading to ill conditioned matrices and convergence issues [105, 171].

6.2 Delaunay mesh improvements algorithms

Mesh improvement techniques have been proposed in the literature to circumvent this critical issue with 3D Delaunay tessellation. Algorithms used to avoid the presence of poorly shaped elements can be classified as either *static* or *dynamic mesh improvements* [105]. *Static improvements* are conceived to act only on the initial mesh created by the Delaunay triangulation, modifying it to achieve the highest quality. This is done without limiting the required computational time, since the mesh will remain the same during the whole analysis and the time spent for its improvement is always convenient in terms of the overall accuracy and efficiency of the solution. There are however many engineering applications where the initial mesh must be modified during the analysis because the domain undergoes large changes in time, such as structures with large elastic/elastoplastic deformations, free surface fluid flows or fluid-structure interaction problems. The problem of maintaining a mesh with a good quality during the whole analysis duration can be called *dynamic* mesh management, as it may include both mesh generations and mesh improvements [172, 173]. This can be achieved by employing a constant mesh to be improved during the analysis whenever necessary, or directly generating new meshes whenever the previous one becomes too distorted. It is important to remark that dynamic mesh management has completely different requirements with respect to static mesh improvement. First of all, a key point is to limit its computational cost, as it is performed repeatedly during the analysis and not just for the initial mesh. Second, it should introduce as few mesh changes as possible both to obtain faster algorithms and, more importantly, to reduce the so called artificial diffusion, which is the numerical error accumulated in the interpolation of the physical variables whenever the mesh in modified.

Static and dynamic mesh managements usually make use of two categories of improving techniques: the ones that are based on topology altering operations and the ones that preserve the topology limiting the mesh modification to nodes re-locations, usually called “smoothing” techniques. Typical actions of the former family are edge or face swapping, local subdivision, node insertion or deletion, elements merging or splitting [174–176]. On the other hand, among the smoothing techniques, the most simple and popular one is the Laplacian smoothing [177], where a node is moved based on the average position of its neighbouring nodes. This algorithm is computationally inexpensive and easy to be implemented, but in general there is no guarantee

of a mesh improvement and, especially in three-dimensions, it can lead to mesh deterioration or even inverted elements [104–106]. Another family of smoothing techniques is given by the optimization based algorithms, which compute the new nodes positions maximizing an objective function based on a chosen quality measure [178, 179]. Optimization algorithms guarantee the final improvement of the resulting mesh in the sense of the selected quality index, but lead to a higher computational complexity and cost.

A different class of smoothing algorithms improves the mesh while it deforms during the simulation. This family is based on physical analogies, so that the mesh nodes move according to physical processes that can be modeled through numerical methods. In the linear spring analogy method [180], nodes connectivities are replaced by springs which have stiffness inversely proportional to the connectivity length. Thus, the nodal displacements can be obtained by solving a system of equilibrium equations based on Hooke's law, under prescribed displacements on the mesh boundary. In this way, starting from an initially optimized mesh undergoing large deformation during the analysis, better quality meshes can be recovered. As the possibility of elements inversion and intersection has been reported, more complex versions have been developed [181–183]. The family of methods based on linear elasticity analogy [184, 185] is conceptually similar, but instead of introducing springs, it considers a fictitious elastic problem to be solved on the mesh domain, with elastic parameters which, in its more complex versions, may depend on the element quality [186, 187]. Nodal displacements leading to an improved mesh quality can be computed solving the fictitious elasticity problem under the imposed boundary movements. Though these algorithms have been reported in the literature to provide good results in improving a deformed mesh, their application to Lagrangian explicit free-surface problems is limited, because they need to operate onto an initially optimized mesh and are not suited for the improvement of a new Delaunay mesh generated runtime.

6.3 Novel smoothing algorithm for tetrahedral mesh

Among the large variety of mesh improving techniques proposed in the literature, to the best of the authors' knowledge there is no strategy able to fulfil the following key requirements of a 3D explicit method involving frequent remeshing:

- it has to be computationally inexpensive, as it is performed very frequently during the analysis;
- it has to guarantee the removal of all sliver elements, since the presence

of just one of them causes a vanishing critical time step size in an explicit framework;

- it has to introduce as few changes as possible in the position of nodes in the mesh, in order to limit the artificial diffusion introduced by data convection to the new nodal positions.

Fulfilling these conditions is particularly demanding because the algorithms ensuring the sliver removal are usually complex and expensive and consequently they are not applicable in methods with frequent remeshing as the PFEM. For this reason, a novel efficient mesh smoothing technique has been developed to remove all the slivers and, hence, to increase the mesh critical time step. The smoothing step is composed by the application of two successive algorithms: the first is based on an elastic analogy directly designed to accomplish the specific requirements of this method, hereafter referred to as P-Smoothing (Pressure based Smoothing); the second is the implementation of the sequential version of the Geometric Element Transformation Method (GETMe) proposed in [106], to overcome few critical situations where the first algorithm is less effective.

6.4 Elastic analogy based smoothing (P-Smoothing)

The proposed P-Smoothing approach is based on a novel use of the physical analogy idea characterizing some algorithms described in Section 6.2. More in details, the mesh deformation is governed by a fictitious elastic problem, but rather than considering a fictitious body under an imposed boundary displacement, it considers a body with fixed boundaries and an imposed distribution of fictitious stresses. These stresses are properly designed to provide high values of fictitious pressures in the bad quality elements producing a movement of their nodes and an improvement of the corresponding critical time steps. Fictitious elastic problems of this type are defined only on those portions of the overall domain where the mesh needs to be improved, so that only very small scale elastic problems have to be solved.

6.4.1 Selection of the smoothing domain

Since the stable time step size is governed, through Equation (2.48), by the minimum element insphere radius r^{in}, the elements quality is measured based on this geometric property. At the beginning of the algorithm, the insphere radius r_e^{in} of each tetrahedron e is computed in terms of the current value of the element nodal coordinates $\mathbf{x}_e$. In particular, let $\{a, b, c, d\}$ denote the four tetrahedron nodes and let the (non-unit) normal vector to

the face having vertices $\{a, b, c\}$ be defined as:

$$\mathbf{n}_{abc} = (\mathbf{x}_{e,b} - \mathbf{x}_{e,a}) \times (\mathbf{x}_{e,c} - \mathbf{x}_{e,a}) \tag{6.1}$$

Then, the element insphere radius can be computed with the following formula:

$$r_e^{in} = \frac{6\ V_e}{|\mathbf{n}_{abc}|\ |\mathbf{n}_{abd}|\ |\mathbf{n}_{acd}|\ |\mathbf{n}_{bcd}|} \tag{6.2}$$

where V_e is the tetrahedron volume and $|\ \cdot\ |$ is the standard Euclidean norm.

Each element insphere is compared with a certain a priori defined target value $\bar{r}^{in}$, for example the average of the radii among the mesh elements, scaled by a tuning parameter $a \leq 1$. All the elements which satisfy the following relation:

$$r_e^{in} \leq a\ \bar{r}^{in} \tag{6.3}$$

are classified as "critical" and are selected for improvement. This quality criterion will automatically include all the sliver elements, as well as some other very distorted ones that may limit the time step size.

These critical elements, together with a suitable number of their neighbour elements, constitute the portion Ω_s^h, hereafter referred to as "smoothing domain", of the mesh on which the fictitious elastic problem will be solved . The extension of this portion can be controlled by varying the threshold for the critical elements selection through the parameter a, or through the enlargement of the set of the included neighbour element. As a limit case, one could decide to include the whole fluid domain in the smoothing step. In the present version, a is set to 3% and only a single layer of neighbour elements is considered, namely all the elements which share at least one vertex with the critical one (Figure 6.1). This choice leads to a faster algorithm with few moving nodes, thus reducing the artificial diffusion. The smoothing domain Ω_s^h obtained in this way is in general not fully connected, but consists of a series of patches $\Omega_{s,P}^h$, each one surrounding one or more than one critical elements (Figure 6.1), i.e. $\Omega_s^h = \bigcup_P \Omega_{s,P}^h$.

The external nodes defining the boundaries of the smoothing domain $\partial\Omega_s^h$ (white dots in Figure 6.1) are considered fixed, while internal ones, i.e. the vertices of the critical elements (black dots in Figure 6.1), can move. Moreover, if one of these latter vertices belongs to the boundary of the physical fluid domain or to the fluid free-surface, then it will be kept fixed, i.e. the fluid boundaries are not modified.

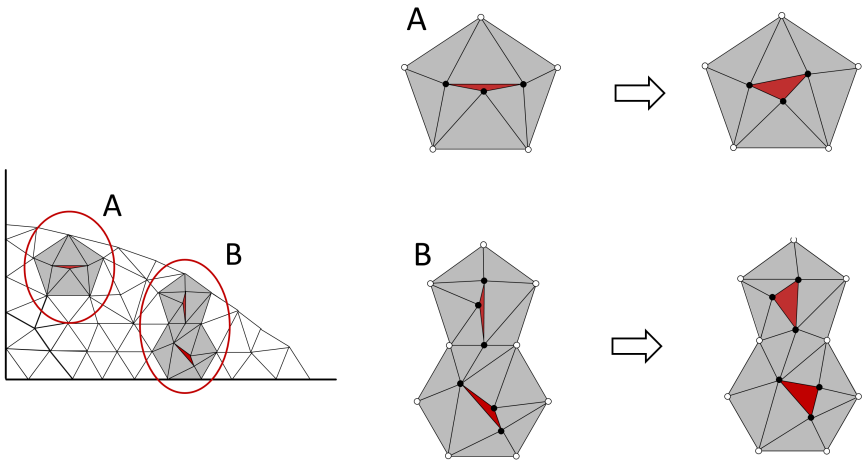


Figure 6.1: 2D representation of the action of the P-Smoothing. (a) Selection of the smoothing domain inside the fluid mesh, composed by the critical elements (red) and the neighbour ones (grey). (b) Left: examples of the patches composing the non fully-connected smoothing domain, containing one (A) or more than one (B) critical elements. The external nodes of the patch (white dots) are kept fixed while the moving nodes are those of the critical element (black dots). Right: results after the action of the P-Smooothing, with a more regular distribution of the insphere radii among the elements.

6.4.2 Definition of the P-Smoothing elastic analogy problem

After the tetrahedra in the smoothing domain Ω_s^h have been selected, a fictitious finite element elastic problem, governed by standard elasticity equations with a modified volumetric compatibility, is defined with homogeneous Dirichlet boundary conditions on $\partial\Omega_s^h$. Quantities, such as stresses, strains and time, defined in this fictitious problem, will be denoted by a star.

The equivalent internal nodal forces $\mathbf{F}_{int}^*$ for nodes in the smoothing domain Ω_s^h are given by:

$$\mathbf{F}_{int}^* = \int_{\Omega_s^h} \mathbf{B}^T[\boldsymbol{\sigma}^*]d\Omega = \int_{\Omega_s^h} \mathbf{B}^T\{2G^*[\mathbf{e}^*] + K^*\epsilon_v^*\mathbf{I}\}d\Omega \tag{6.4}$$

where $\mathbf{B}$ is the usual finite element small strain compatibility matrix, constant over each tetrahedron and assembled over the elements in the smoothing domain, $[\boldsymbol{\sigma}^*]$ and $[\mathbf{e}^*]$ are the column vectors gathering the components of the fictitious stress $\boldsymbol{\sigma}^*$ and small deviatoric strain $\mathbf{e}^*$ tensors, ϵ_v^* is a

suitably defined measure of the volumetric strain and G^* and K^* are the shear and bulk moduli of the fictitious material considered for the smoothing domain: in the examples here presented they are arbitrarily set to a unit value.

While standard linear compatibility is assumed for the deviatoric strains, i.e. $[\mathbf{e}^*] = \mathbf{B}^{dev}\mathbf{U}^*$, $\mathbf{U}^*$ being the vector of nodal displacement in Ω_s^h, the compatible small volumetric strain $\epsilon_v^*(\mathbf{x})$ is replaced in (6.4) by $\tilde{\epsilon}_v^*$, constant over each element $\Omega_{s,e}^h \in \Omega_s^h$ and defined as

$$\tilde{\epsilon}_{v,e}^* = \left(\frac{\bar{r}^{in}}{r_e^{in}}\right) \tag{6.5}$$

where r_e^{in} is the current value of the element insphere radius and $\bar{r}^{in}$ is the pre-defined target value. The definition ((6.5)) of the fictitious volumetric strain $\tilde{\epsilon}_v^*$ is motivated by the outcome of numerical experiments in Karman et al. [184] and Yang and Mavriplis [185], where elastic properties inversely proportional to the element volume are shown to lead to a faster convergence of sliver elements towards the desired shape, avoiding the problem of element inversion. In fact, looking at the evolution of $\tilde{\epsilon}_v^*$ with r_e^{in} shown in Figure 6.2, one can see that element configurations with $r_e^{in} < \bar{r}^{in}$ are highly penalized, generating a high fictitious pressure $p_e^* = K^*\tilde{\epsilon}_{v,e}^*$, tending to infinity for vanishing r_e^{in}. This prevents the possibility of invalid configurations with inverted elements, which can be a typical problem for Laplacian smoothing or other linear elasticity approaches. The fictitious pressure generated in the elements surrounding the sliver element helps avoiding that the expansion of the sliver is achieved at the cost of an excessive volume reduction of the surrounding elements. Since the smoothing is always carried out immediately after a Delaunay triangulation, it should also be noted that the starting mesh cannot contain inverted elements.

Denoting by $\tilde{\mathbf{F}}_{int}^*$ the internal force vector obtained by replacing ϵ_v^* in (6.4) with $\tilde{\epsilon}_v^*$, the discretized fictitious elastic problem for the smoothing domain Ω_s^h turns out to be governed by the following semi-discretized equations of motion

$$\begin{aligned}
&\mathbf{M}^*\mathbf{A}^* + \mathbf{C}^*\mathbf{V}^* + \tilde{\mathbf{F}}_{int}^* = \mathbf{0} \\
&\mathbf{U}^*(0) = \mathbf{0} \quad \text{in } \Omega_s^h, \qquad \mathbf{U}^*(t^*) = \mathbf{0} \quad \text{on } \partial\Omega_s^h \\
&\mathbf{V}^*(0) = \mathbf{0} \quad \text{in } \Omega_s^h, \qquad \mathbf{V}^*(t^*) = \mathbf{0} \quad \text{on } \partial\Omega_s^h
\end{aligned} \tag{6.6}$$

where $\mathbf{M}^*$ is a fictitious lumped mass matrix, $\mathbf{C}^*$ is a matrix of fictitious damping coefficients, $\mathbf{A}^*$ and $\mathbf{V}^*$ are the nodal acceleration and velocity vectors. In the initial configuration of the fictitious problem, at the fictitious time $t^* = 0$, most elements in the smoothing domain have an insphere radius

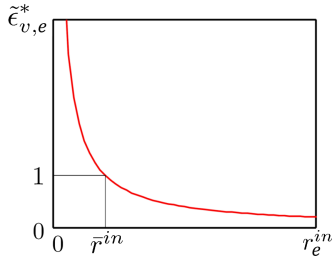


Figure 6.2: Relation between the fictitious volumetric strain $\tilde{\epsilon}^*_{v,e}$ and the insphere radius r_e^{in}.

$r_e^{in} \neq \bar{r}^{in}$. Then, from (6.4) and (6.5) one has that $\tilde{\mathbf{F}}^*_{int}(0) \neq \mathbf{0}$ and fictitious nodal accelerations $\mathbf{A}^*$ develop until a new equilibrium configuration, characterized by vanishing $\mathbf{A}^*$ and $\mathbf{V}^*$, is achieved. The nodal positions in this final configuration are recorded and used to define the improved mesh. The elements in this new mesh exhibit a more regular distribution of insphere radii lengths and, consequently, an increased value of the critical time step size (Figure 6.1).

6.4.3 Solution of the P-Smoothing problem

Problem (6.6) has to be solved on the smoothing domain Ω_s^h, whenever a mesh smoothing is required. To be computationally acceptable and compatible with the overall explicit approach, also its numerical solution should be pursued with a fully explicit and parallelizable solution scheme. To this purpose, the Adaptive Dynamic Relaxation (ADR) technique [188–191] is employed. This is an explicit pseudo-dynamic method for the static solution of structural mechanics problems, where artificial masses and damping coefficients are introduced into (6.6). The static solution of problem (6.6) is obtained as the final steady-state response, given by $\mathbf{A}^* = \mathbf{0}$, at the end of the transient dynamic analysis of the system. Mass and damping parameters are chosen so as to accelerate the convergence to the steady-state response, since the transient solution is not of interest.

The ADR technique has been successfully employed to address highly nonlinear static problems, because no linear systems have to be solved for the solution of the equivalent explicit dynamic problem and only vectorial operations need be performed. Furthermore, it has been proved that, with proper choices of the dynamic system parameters, the ADR method becomes completely equivalent to the implicit Newton-Raphson solution of the corresponding nonlinear static problem [192], but avoiding any convergence difficulties. The same explicit central difference scheme used for the

integration of the fluid motion and described in section 2.4 is chosen for the fictitious problem (6.6). The scheme is however modified in order to minimize the computing time, by defining a fictitious time step size $\Delta t^* = 1$ s [189], constant throughout the fictitious analysis. Rather than adapting the time step size, stability of the integration is achieved by modifying the fictitious mass matrix $\mathbf{M}^*$ at each time step.

Assuming a constant fictitious time step Δt^*, the integration scheme in (2.40)-(2.41) and (2.44)-(2.45) is applied to the problem (6.6) leading to:

$$\begin{aligned}\mathbf{V}^{*,n+1/2} = (2\mathbf{M}^{*,n} + \mathbf{C}^{*,n}\Delta t^*)^{-1} (2\mathbf{M}^{*,n} - \mathbf{C}^{*,n}\Delta t^*)\, \mathbf{V}^{*,n-1/2} + \\ - (2\mathbf{M}^{*,n} + \mathbf{C}^{*,n}\Delta t^*)^{-1} 2\Delta t^* \left(\tilde{\mathbf{F}}^{*,n}_{int}\right)\end{aligned} \tag{6.7}$$

$$\mathbf{U}^{*,n+1} = \mathbf{U}^{*,n} + \mathbf{V}^{*,n+1/2}\Delta t^* \tag{6.8}$$

where $\mathbf{M}^{*,n}$ is a lumped mass matrix. The analysis stops when an acceptable error in the internal forces balance is reached with vanishing inertia and viscous forces, corresponding to the achievement of a static solution. More specifically, convergence is defined as the vanishing of the following norm of nodal velocities:

$$\epsilon = \frac{\| \left(\mathbf{V}^{n+1/2} - \mathbf{V}^{n-1/2}\right) \|}{\|\mathbf{V}^{n-1/2}\|} \leq \epsilon^{ADR}_{toll} \tag{6.9}$$

To guarantee optimally fast and stable convergence to the final steady state condition, the ADR method defines the diagonal entries of the lumped mass matrix in terms of the tangent stiffness matrix $\mathbf{K}^{*,n}$ [189, 191]:

$$M^{*,n}_{ii} \geq \frac{(\Delta t^*)^2}{4} \sum_{j=1}^{3N_s} |K^{*,n}_{ij}| \quad or \quad \mathbf{M}^{*,n} \geq \frac{(\Delta t^*)^2}{4} \mathbf{S}^{*,n} \tag{6.10}$$

where N_s is the number of nodes in the smoothing domain and $\mathbf{S}^{*,n}$ is the lumped tangent stiffness matrix, obtained assembling the corresponding element matrices $\mathbf{S}^{*,n}_e$, where the index e runs over the elements in the smoothing domain. However, to avoid the expensive numerical evaluation of the tangent stiffness matrix, matrix $\mathbf{S}^{*,n}$ is replaced by an estimate of the directional tangent stiffness $\bar{\mathbf{S}}^{*,n}$ at the instant $t^{*,n}$, defined in Oakley and Knight [189] as:

$$\bar{S}^{*,n}_i = \frac{\left(\tilde{F}^*_{int}\right)^n_i - \left(\tilde{F}^*_{int}\right)^{n-1}_i}{\Delta t^* V^{*,n-1/2}_i} \tag{6.11}$$

To ensure stability of the integration, this substitution requires the introduction of a safety factor β, finally obtaining the following expression:

$$\mathbf{M}^{*,n} \geq \beta \frac{(\Delta t^*)^2}{4} \bar{\mathbf{S}}^{n,*} \quad , \quad \beta >> 1 \tag{6.12}$$

This choice compromises the optimal convergence to the equilibrium state, but the time gained avoiding the expensive computation of the full tangent stiffness matrix $\mathbf{K}^{*,n}$ compensates for the employment of a larger number of time steps. Furthermore, in view of the high level of nonlinearity in the internal forces computation, a runtime stability check has also been introduced, based on the perturbed apparent-frequency error measure [189, 191] defined as:

$$\epsilon_i^{stab} = \frac{(\Delta t^*)^2 \, |A_i^{*,n} - A_i^{*,n-1}|}{4|U_i^{*,n} - U_i^{*,n-1}|} \tag{6.13}$$

where values of $\epsilon_i^{stab} > 1$ represent possible unstable conditions that need to be eliminated with an increase of parameter β. The value of β can be defined by the user based on computational cost considerations. Considering the very fast convergence to the equilibrium state obtained with to the ADR technique, a safe value of $\beta = 10$ has been used in the presented examples without any occurrence of the unstable condition $\epsilon_i^{stab} > 1$. Finally, the ADR method makes use of a mass proportional damping [189], obtaining the following definition of the damping matrix:

$$\mathbf{C}^{*,n} = c^n \mathbf{M}^{*,n} \quad , \quad c^n = 2\sqrt{\lambda_0^n} \tag{6.14}$$

where λ_0^n is defined as:

$$\lambda_0^n = \frac{\left(\mathbf{V}^{*,n-1/2}\right)^T \bar{\mathbf{S}}^{*,n} \mathbf{V}^{*,n-1/2}}{\left(\mathbf{V}^{*,n-1/2}\right)^T \mathbf{M}^{*,n} \mathbf{V}^{*,n-1/2}}. \tag{6.15}$$

This choice of damping is intended to minimize the oscillations before reaching the steady state, without reducing too much the velocity, which would delay the reaching of the equilibrium configuration.

The main steps of the P-Smoothing algorithm are summarized in Algorithm 4, while its key features are recalled below.

- It is specifically designed to cure sliver and overly distorted elements with the goal of an increased critical time step of the explicit solver.
- It can be applied to improve any mesh generated by Delaunay tessellation because it does not need a reference target mesh as other elastic analogy based algorithms do.

- It acts only on the portion of the mesh that needs be improved. This guarantees efficiency and a small number of displaced nodes, thus reducing artificial diffusion.

- The algorithm exploits the very same structure of the explicit solver of the PFEM, drastically simplifying the overall solver architecture and the implementation complexity of the method.

- Thanks to its explicit nature and to the dynamic relaxation method, it is extremely fast and fully parallelizable, ensuring significant gains in the critical time step size of the fluid solver at an extremely low cost (the fictitious analysis of the smoothing domain usually requires only few dozens of fictitious time steps to converge to to the static solution). More in details, the fictitious elastic problem is solved in parallel on the whole smoothing domain using the same parallel procedure described in Section 3.3. Both the elemental assembly procedure and the solution of the system of decoupled nodal equations of the fictitious problem are distributed on multiple threads with shared memory.

- It acts only on the internal nodes of the fluid domain. The nodes on the boundary and on the free surfaces are not moved.

Algorithm 4 P-Smoothing

Selection of the smoothing domain Ω_s^h (Section 6.4.1)
Initialization $n = 0$; $\mathbf{U}^*(0) = \mathbf{0}$; $\mathbf{V}^*(0) = \mathbf{0}$
while $\epsilon > \epsilon_{toll}^{ADR}$ **do**
  Compute $\mathbf{F}_{int}^{*,n}$ from eq. ((6.4))
  Compute $\bar{\mathbf{S}}^{*,n}$ from eq. ((6.11))
  Compute $\mathbf{M}^{*,n}$, $\mathbf{C}^{*,n}$ matrices: eqs. ((6.12)), ((6.14))
  Solve explicit time-step: eqs. ((6.7))-((6.8))
  Compute stability parameters ϵ_i^{stab} from eq. ((6.13))
  if $\epsilon_i^{stab} > 1$ for some i **then**
    increase value of β
  end if
  Compute ϵ from ((6.9))
  $n = n + 1$
end while

6.5 Geometry based smoothing

One of the aforementioned advantages of the P-Smoothing algorithm, namely its capacity to preserve domain boundaries, may become also a limitation. If a sliver appears near the fluid free-surfaces, it may have many of its nodes constrained to remain fixed and the P-Smoothing may be ineffective. Nonetheless, this situation of constrained slivers is frequent because the complex geometry of free-surfaces forces the mesh generator to use distorted tetrahedra. Obviously, this is a crucial issue because a single bad element can frustrate all the efforts to increase the critical time step.

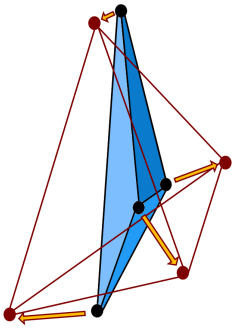


Figure 6.3: GETMe Approach: Starting from a distorted tetrahedron (in blue) and displacing all its vertices in the direction normal to their opposite faces, a more regular tetrahedron is obtained.

At the same time, one should also consider that in a pathological element such as a sliver, where the four nodes are almost coplanar and the insphere radius vanishes, even a small correction of nodal positions can lead to orders of magnitude gained on the critical time step size. Based on these considerations, an additional smoothing technique, called Sequential Geometric Element Transformation Method (GETMe) and presented in Vartziotis et. al [106], has been implemented. Its peculiarities will be now only briefly described. For a more detailed description, refer to Vartziotis et. al [106].

The GETMe approach is based on a regularizing element transformation consisting on a simple, element based construction that, applied iteratively, can transform a badly shaped element into a regular one. Let us consider Figure 6.3 showing a badly shaped tetrahedron T, with its four nodes coordinates listed in the vectors $\mathbf{x}_i$, with $i = 1, 4$. Moreover, let $\mathbf{n}_i$ represent the unit vectors that are normal to the tetrahedron faces opposite to the i-th node and pointing outwards with respect to the tetrahedron. Finally, a scaling factor σ_e, depending on the quality of the tetrahedron, is introduced. Then, the new nodes positions of the transformed tetrahedron T'

can be computed as:

$$\mathbf{x}'_i = \mathbf{x}_i + \sigma_e \mathbf{n}_i \quad i = 1, \ldots, 4 \tag{6.16}$$

where the transformed tetrahedron T' has a quality measure that is greater than the original one T. The choice of the scaling parameter σ_e becomes crucial when considering a set of elements linked together, because an improving transformation for one element may result in a quality worsening (or even in an element inversion) for the surrounding ones. As a consequence, since this is a purely geometrically driven movement, one can ensure a local quality increase for the considered element but not a global improvement for the whole mesh. In the examples here presented, σ_e is computed as one tenth of the element mean side.

From the practical point of view, the implemented algorithm consists of a certain number of iterations, within which the following steps are performed:

- the worst tetrahedron is selected among the whole mesh as the one with the smallest insphere radius;
- the outwards nodal normals $\mathbf{n}_i$ and the scaling factor σ_e are computed for that tetrahedron. If the nodes belong to the fluid free surface, the value of σ_e is reduced in order to introduce only small and acceptable displacements on the fluid free-surface (in the following examples σ_e is three times smaller on such nodes);
- the tentative new nodes positions are computed with Equation ((6.16)).
- the neighbour elements are checked to control possible element inversions or worsening in the global stable time step. If this is the case, the iteration is repeated with a reduced value of σ_e. The way of reducing σ_e can be controlled by the user. A strong reduction ensures that inversions or worsening of the surrounding elements are avoided, but it reduces the effectiveness of the GETMe iteration on the considered worst tetrahedron. In the examples here presented, in case of element inversions or worsening, the iteration is repeated with a σ_e reduced by 25% ;
- the new position of the tetrahedron nodes is confirmed and saved, namely $\mathbf{x}_i = \mathbf{x}'_i$

These iterations are repeated until a termination criterion is met. As suggested in Vartziotis et. al [106], the criterion consists in a check on the improvement of the minimum insphere radius at each iteration k, namely:

$$\frac{(r^{in}_{min})^k - (r^{in}_{min})^{k-1}}{(r^{in}_{min})^{k-1}} < \epsilon^{GETMe}_{toll} \tag{6.17}$$

where $(r_{min}^{in})^k$ is the minimum inphere radius among all the elements at the $k-th$ iteration and ϵ_{toll}^{GETMe} is a given tolerance. Additionally, a maximum number of iterations is defined. The sequential GETMe has been chosen because it can work in combination with the P-Smoothing, since it can be very effective for the critical situations previously mentioned on the constrained bad elements. Its geometry based strategy allows to fully control the magnitude of the nodal displacements, introducing only a small, acceptable variation of the position of fluid free-surface nodes, that is very effective for the improvement of the element critical time step. On the other hand, the sequential GETMe could not be globally effective if employed alone, because it would require too many iterations to act on all the elements needing an improvement, leading to an expensive algorithm, not suitable for PFEM applications. The combined action of P-Smoothing ad GETMe has proved to be really effective, fully satisfying the requirements discussed in Section 6.3 for a suitable mesh smoothing method, as it will be shown in the numerical examples.

6.6 Numerical examples

6.6.1 Cube

Let us consider a simple domain given by a cube of 1 m side. This first example only focuses on the smoothing approach, considering how the initial static mesh is modified by the combined algorithm presented in Section 6.3. The standard Delaunay triangulation generates the mesh depicted in Figure 6.4, composed by 1159 nodes and 4665 elements. The worst tetrahedra in the mesh, namely those having the smallest insphere radii, are highlighted in red. The graph in Figure 6.5(a) plots the stable time step computed on

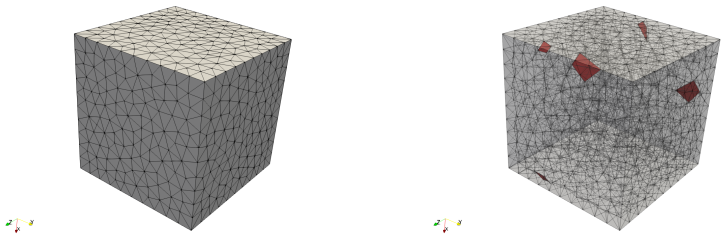
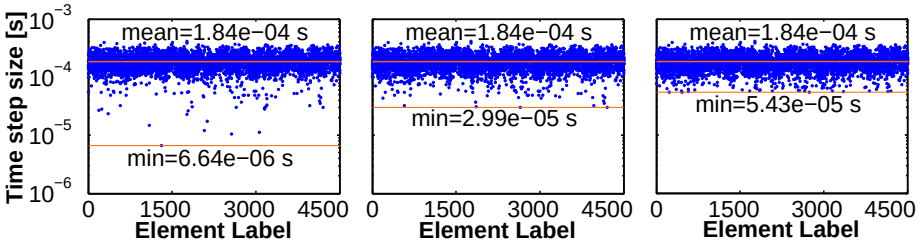


Figure 6.4: Cube. Standard Delaunay triangulation. The worst tetrahedra are highlighted in red.

each element of the mesh. One can observe that, despite a mean value of

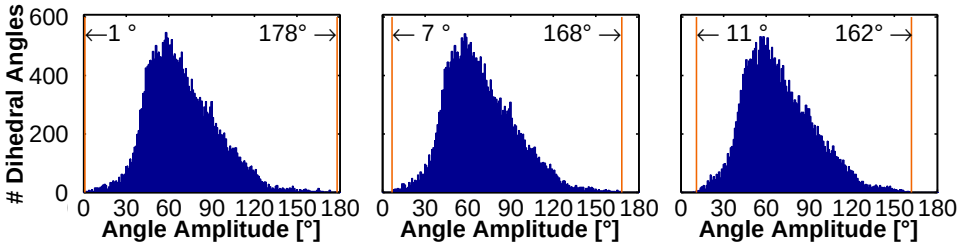


(a) After Delaunay **(b)** After P-Smoothing **(c)** After P-Smoothing and GETMe Smoothing

Figure 6.5: Cube. Stable time step computed on each mesh element after: (a) Delaunay triangulation, (b) P-Smoothing, (c) P-Smoothing and GETMe smoothing

1.84 10^{-4} s, the presence of few very bad tetrahedra leads to a minimum value of 6.6 10^{-6} s which is the actual value that would have been used for an explicit dynamic analysis. Moreover, the histogram on Figure 6.6(a) shows how many dihedral angles are present for each angle amplitude. The highlighted smallest and biggest values of 1° and 178°, respectively, prove the presence of very distorted tetrahedra. The safety parameter of Equation ((6.3)) is set to $a = 3\%$, leading to a smoothing domain involving only 22 nodes and 500 elements. The action of the P-Smoothing on those elements allows to obtain the situation represented by the graphs in Figures 6.5(b), 6.6(b), while the subsequent action of GETMe smoothing modifies less than 50 elements leading to the graphs in Figures 6.5(c), 6.6(c), respectively. The combined action of the two smoothing algorithms leads to minimum and maximum dihedral angles of 11° and 162° respectively. This is reflected in the new value of the global minimum stable time step of 5.43 10^{-5} s, i.e. more than one order of magnitude greater than the initial value. To have a comparison with the results obtained with another mesh optimization algorithm, the automatic NETGEN algorithm [193] implemented in the open source software Gmsh [194] has been considered. This software employs several techniques based on quality optimization, involving both local nodal smoothing and topology changes. The NETGEN optimization of the mesh in this test case leads to a global stable time step of 7.9 10^{-5} s.

Performing the same comparison test on a much finer grid of the same cube ($1.02M$ elements), similar results are obtained: the stable time step computed on the mesh generated by the NETGEN algorithm is 7.4 10^{-6} s, while the one computed on the mesh obtained with the proposed approach is 8.02 $10^{-6}s$. These results have been obtained in 127.1 s by the NETGEN algorithm, while 13.8 s are required by the presented smoothing procedure



(a) After Delaunay **(b)** After P-Smoothing **(c)** After P-Smoothing and GETMe smoothing

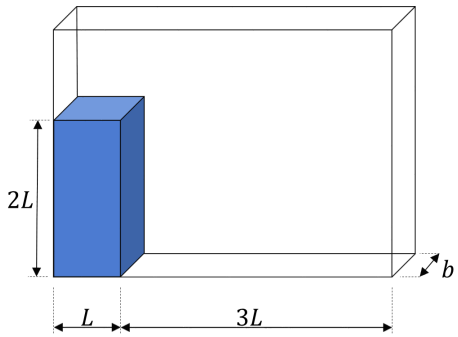
Figure 6.6: Cube. Histogram of the dihedral angles computed after: (a) Delaunay triangulation, (b) P-Smoothing, (c) P-Smoothing and GETMe smoothing

on the same computer. This is related to the fact that the NETGEN algorithm belongs to the group of static mesh improvement algorithms described in Section 6.2: they optimize based on more general definitions of quality measure, but the time required to create a single mesh is not acceptable in the frequent remeshing framework of the PFEM. In conclusion, while a more regular set of tetrahedra is obtained with the NETGEN algorithm, the quality index relevant for an explicit solver, i.e. the minimum stable time step, is comparable with respect to the one obtained with the proposed approach, which has a computational cost compatible with the PFEM applications. Finally, it must be noted that in this simple case with flat boundaries the problem of constrained critical elements is not present and the results of the P-Smoothing algorithm are already very positive. However, also in this case a non-negligible improvement is provided also by the GETMe smoothing, since it acts directly on the worst elements, thus producing a direct improvement on the stable time step size.

In the following real applications with complex free surfaces, besides the validation of the PFEM approach, the temporal evolution of the time step size obtained with the dynamic mesh management will be highlighted, showing good results and an even more visible complementarity in the action of the two methods in the combined approach.

6.6.2 Dam break

The classical dam break experimental test [5, 195, 196] is here considered. The problem has been addressed with both the 2D and 3D versions of the present PFEM approach. The model geometry is depicted in Figure 6.7, together with the table providing the geometrical parameters and other



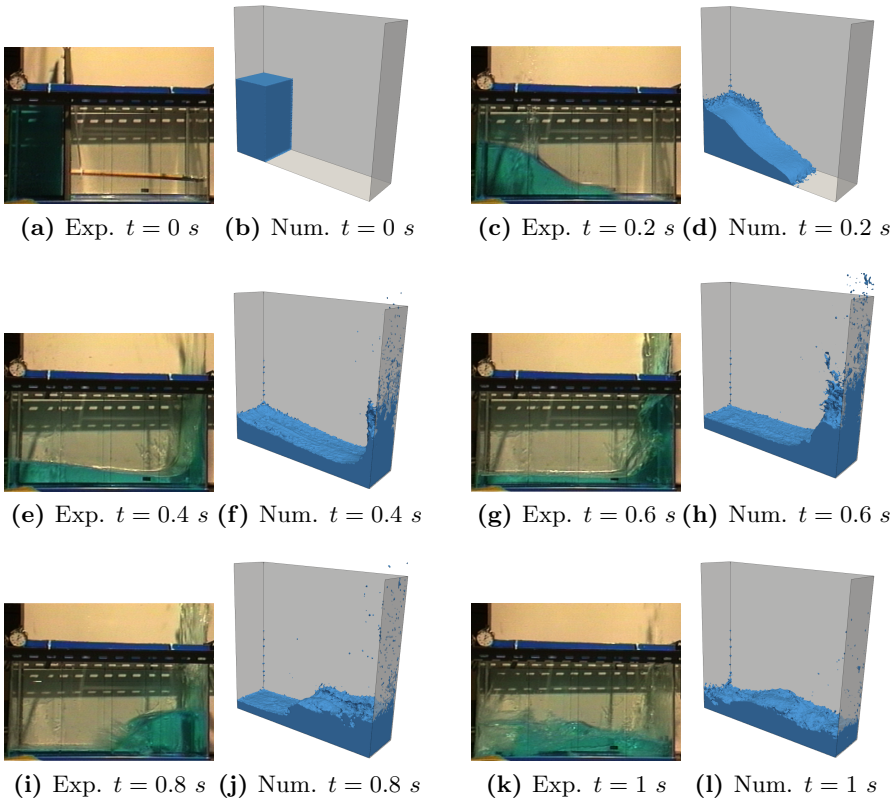
Fluid Properties	
Density ρ_f	1000 kg/m^3
Viscosity μ_f	10^{-3} $Pa \cdot s$
Geometry	
L	0.146 m
b	0.175 m
Mesh	
n. nodes	$\sim$ 200 k
n. elem.	$\sim$ 750 k
Analysis	
Time simulated	1 s
Total analysis	50 h
% Solver	84 %
% Remeshing	6 %
% Smoothing	9 %
n. threads	8

Figure 6.7: Dam Break. Geometry and parameters of the analysis.

data of the analysis. At the left side of the experimental domain, a column of water is supported by a rigid vertical barrier. At the beginning of the experiment, the barrier is suddenly removed, leaving the column of water to flow inside the tank due to gravity. The generated wave impinges against the opposite wall of the tank and rises up above the initial water column height, then it falls down creating a second wave which returns towards the left wall.

Figure 6.8 shows some snapshots of the simulation at synchronised instants with the experimental ones presented in Koshizuka and Oka [5], showing a qualitative good capability of the method to represent this complex phenomenon involving breaking waves, splashes and strong impacts with solid boundaries. For a quantitative validation, Figure 6.9 compares the front position during time of the water wave until the impact with the right wall. One can observe an overall very good agreement of the results obtained with the explicit 2D and 3D PFEM approaches with the experimental and numerical results presented in the literature. However, all the numerical simulations show a slightly faster front advancement compared to the experimental one. As it has been commented in previous works [196, 197], this can be related to the difficulties in the correct modelling of the initial removal of the vertical wall, which takes a finite amount of time in the experiment while it is instantaneous in the simulations.

Let us now focus on the effectiveness of the smoothing procedure. Fig-



(a) Exp. $t = 0\ s$ **(b)** Num. $t = 0\ s$ **(c)** Exp. $t = 0.2\ s$ **(d)** Num. $t = 0.2\ s$

(e) Exp. $t = 0.4\ s$ **(f)** Num. $t = 0.4\ s$ **(g)** Exp. $t = 0.6\ s$ **(h)** Num. $t = 0.6\ s$

(i) Exp. $t = 0.8\ s$ **(j)** Num. $t = 0.8\ s$ **(k)** Exp. $t = 1\ s$ **(l)** Num. $t = 1\ s$

Figure 6.8: Dam Break. Snapshots of the simulation at different time steps compared with the corresponding experimental results at synchronized instants presented in [5]

ure 6.10 shows the evolution of the time step size during the analysis. Each point in the graph in Figure 6.10(a) corresponds to the stable time step computed directly on each mesh generated by Delaunay tessellation, whenever the mesh distortion control of the PFEM indicates that a new mesh is required. Figures 6.10(b)-(c) show the time steps as they are modified by the action of P-Smoothing and of P-Smoothing combined with GETMe smoothing, respectively. The comparison shows the excellent performances of the combined smoothing approach. The meshes generated by Delaunay tessellation cannot be directly used in an explicit solver, since the consequent time step would lead to an unacceptable computational time. On the other hand, the P-Smoothing produces almost every time a remarkable improvement of the time step size (its average value is 30 times bigger than the average one computed after remeshing), still however presenting

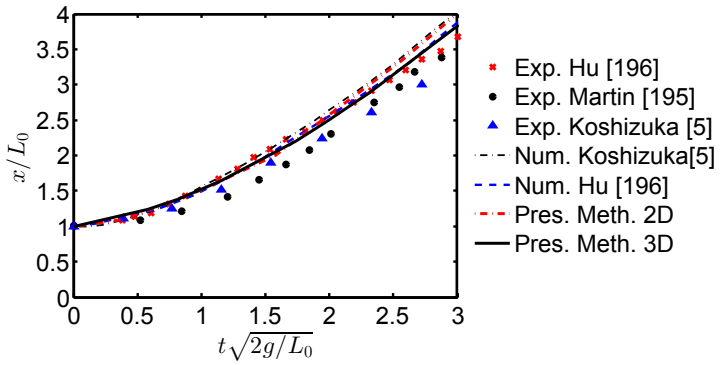
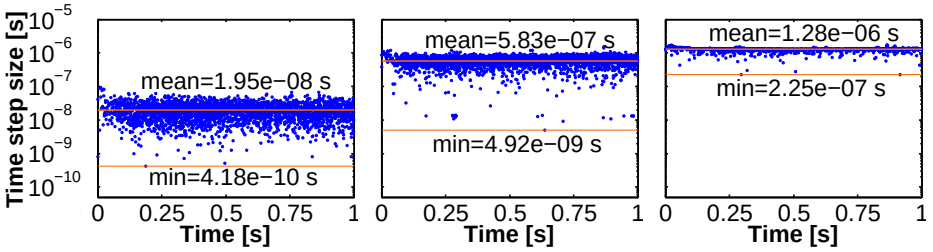


Figure 6.9: Dam Break. Time evolution of the front wave position.

some circumstances where it is not fully effective due to the aforementioned situations of constrained tetrahedra. The action of GETMe smoothing is fully complementary. It provides an additional slight improvement on the average stable time step and, most importantly, it overcomes all the situations of inefficiency of the P-Smoothing algorithm, leading to a very large minimum time step and reducing its variations during the analysis. The combined action of the two smoothing techniques leads to an average time step size during the analysis that is almost 200 times larger than the one which would have been required using the Delaunay mesh directly, while the minimum time step size is almost 500 times larger. This very large gain is obtained at a reasonable computational cost, as it can be observed in the table in Figure 6.7. The computational time spent for the remeshing and smoothing procedures is 15% of the total duration of the analysis, which is normal in the PFEM approach, especially in this type of problems with breaking waves requiring frequent remeshing. This computational efficiency is mainly related to the property of the combined smoothing algorithms to act on a small portion of the overall fluid domain and to the effectiveness of the ADR tecnique described in Section 6.4.3. The safety parameter of Equation ((6.3)) is set to $a = 3\%$, leading to a smoothing domain that ranges in between 4.6% and 8% of the total number of elements during the analysis, because it includes also elements neighbour to the sliver ones. The tolerance for the termination of the GETMe smoothing iterations in Equation (6.17) is fixed to the value $\epsilon_{toll}^{GETMe} = 10^{-4}$, which turns out to force the GETMe smoothing to stop in the average after 750 iterations. Since the number of GETMe iterations is equal to the number of modified elements, this value of the tolerance determines the algorithm to act on approximately 0.1% of the total number of elements for each smoothing step. Moreover, the ADR algorithm allows to reach the equilibrium condition of each P-Smoothing



(a) After Delaunay **(b)** After P-Smoothing **(c)** After P-Smoothing and GETMe smoothing

Figure 6.10: Dam Break: Plot of the stable time step at each remeshing computed after: (a) Delaunay triangulation, (b) P-Smoothing, (c) P-Smoothing and GETMe smoothing

procedure in less than 20 fictitious explicit steps. The same values of the safety parameter a and termination criterion tolerance ϵ_{toll}^{GETMe} are used in the following examples, and the same data in terms of size of the smoothing domains and number of ADR steps are recorded.

Finally, a check on the possible influence of the P-Smoothing on the solution accuracy is performed. A coarse mesh is considered (average of 39.2 k elements during the analysis), in order to run a benchmark analysis in a reasonable computational time without smoothing and, hence, with a very small time step. The result obtained with this coarse mesh will be used as a reference for assessing the effect of the P-Smoothing on the solution accuracy. Table 6.1 shows a comparison of the fluid front positions at different time instants, varying the parameter a defined in Equation (6.3). This parameter controls the size of the smoothing domain and consequently the influence of the nodal P-Smoothing on the results of the analysis. The GETMe smoothing has not been considered in this analysis, due to its very limited action. The results in Table 6.1 suggest that for values of $a \leq 0.05$, typically used in our numerical applications, the smoothing algorithm does not influence significantly the accuracy of the solution. It must be underlined that with such a coarse mesh, variations around 1% are acceptable because they imply a discrepancy in the front advancement of the order of the element size, which would be negligible with a finer mesh. Moreover, it must be noted that the influence on the results increases with the size of the smoothing domain. This is expected because, as commented in Section 6.2, performing nodal smoothing runtime implies the introduction of errors due to artificial diffusion. For this reason, the smoothing algorithm should act only on the critical elements in order to reduce to a minimum the data

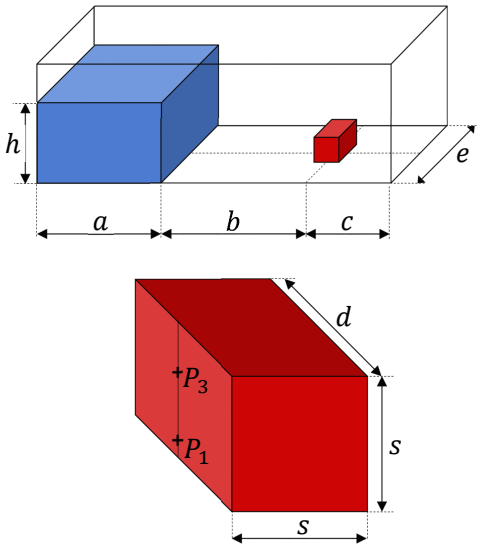
parameter a	# el. in Ω_s^t	posit. at $t = 0.2\ s$	posit. at $t = 0.24\ s$
$a = 0$	0	$0.4546\ m$	$0.554\ m$
$a = 0.005$	255	$0.462\ m$	$0.556\ m$
% error		1.63 %	0.36 %
$a = 0.01$	675	$0.447\ m$	$0.555\ m$
% error		1.67 %	0.18 %
$a = 0.03$	3230	$0.454\ m$	$0.553\ m$
% error		0.13 %	0.18 %
$a = 0.04$	5246	$0.4516\ m$	$0.548\ m$
% error		0.66 %	1.08 %
$a = 0.06$	11403	$0.446\ m$	$0.529\ m$
% error		1.89 %	4.51 %
$a = 1000$	39250	$0.43\ m$	$0.518\ m$
% error		5.41 %	6.5 %

Table 6.1: Dam Break: Comparison of the results on the fluid front position in time for different values of parameter a for the definition of the smoothing domain (Equation (6.3)). The second line of each row shows the relative difference with respect to the benchmark solution with $a = 0$, i.e without smoothing.

convection to the new nodal positions.

6.6.3 Dam break with obstacle

The experimental test performed at the Maritime Research Institute Netherlands (MARIN) and reproduced numerically with the Volume of Fluid Method in Kleefsman et al. [198] is here addressed, providing a more complete validation of the method accuracy because of the available detailed set of experimental measures. It consists of a dam break where, inside the large tank, a small box is placed to interfere with the collapse of the water column. The experimental set up geometry is depicted in Figure 6.11, together with the table providing geometrical parameters and other data of the simulation. The same values of density and viscosity of the previous dam break example have been used. The obstacle box was instrumented with several pressure sensors and, among them, sensors P1 and P3, positioned on the side of the box facing the column of water at the height of $0.021\ m$ and $0.101\ m$, are here considered (Figure 6.11). Moreover, probes measuring the fluid vertical heights were positioned inside the tank. Sensors H2 and H4, at a distance of $2.228\ m$ and $0.584\ m$ from the tank wall behind the dam are here considered.

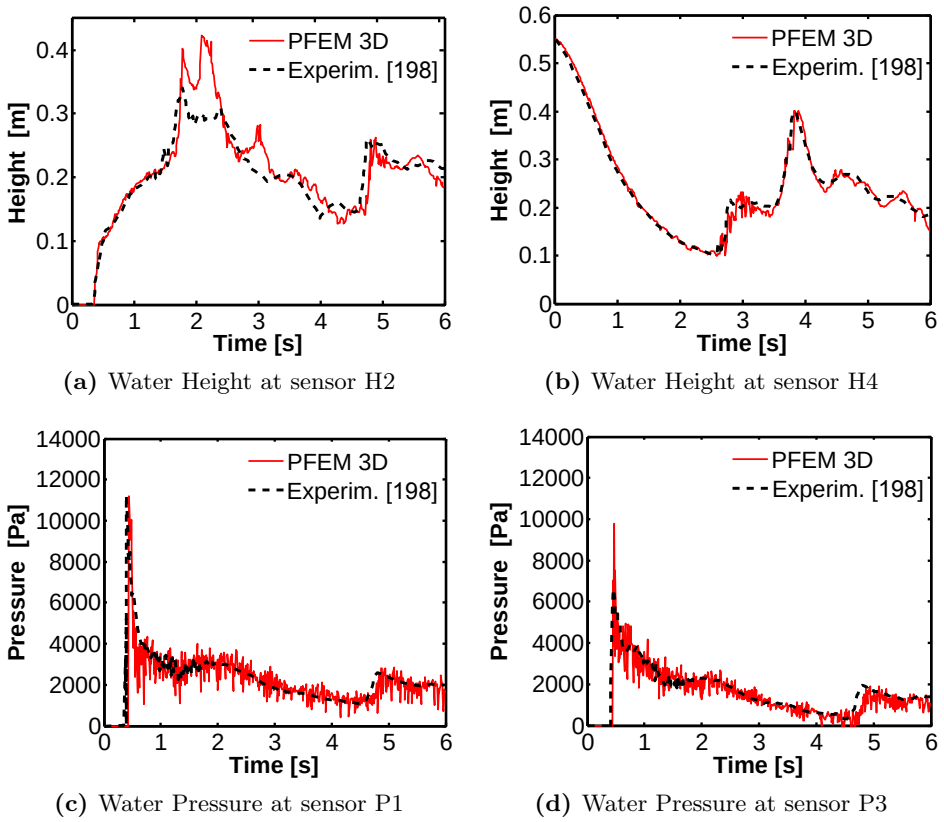


Geometry	
a	1.228 m
b	1.248 m
c	0.744 m
d	0.4 m
e	1. m
s	0.16 m
h	0.55 m
Mesh	
n. nodes	$\sim$ 200 k
n. elem.	$\sim$ 850 k
Analysis	
Time simulated	6 s
Total analysis	236 h
% Solver	95 %
% Remeshing	3 %
% Smoothing	1.2 %
n. threads	6

Figure 6.11: Dam break with obstacle. Geometry and parameters of the analysis.

Figure 6.14 contains some snapshots of the simulation at different instants, showing the first large wave impacting the obstacle and the subsequent oscillating waves. The graphs in Figure 6.12(a)-(b) show the time evolution of the water height at the points H2 and H4, compared with the experimental results presented in Kleefsman et al. [198]. One can observe an overall good agreement with the experimental results. A small overestimation of the peak is visible at H2 only in the time interval around 2 s, i.e. slightly after the first impact with the obstacle, because of the complex dynamics of the originated wave, difficult to be represented correctly at the H2 sensor, which is very close to the obstacle itself. Moreover, the graphs depicted in Figure 6.12(c)-(d) show the pressure time evolution at the probes P1 and P3. Also in this case, despite some spurious oscillations, typical of the pressure field in the explicit PFEM approach, one can observe a very good agreement with the experimental curve.

As far as the dynamic mesh management is concerned, the time step improvements provided by the two smoothing techniques are shown in Figure 6.13. Once again, the time steps computed at each remeshing, after the Delaunay triangulation, are very scattered around the average value during the analysis, showing a mean value $\Delta t_{mean} = 9.85\ 10^{-8}\ s$, but a minimum value $\Delta t_{min} = 2.09\ 10^{-10}\ s$. These values are clearly unaccept-



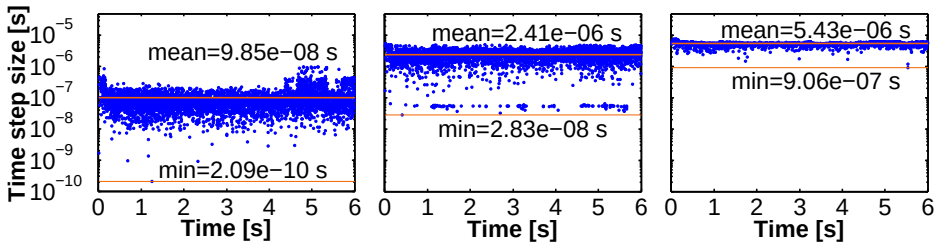
(a) Water Height at sensor H2

(b) Water Height at sensor H4

(c) Water Pressure at sensor P1

(d) Water Pressure at sensor P3

Figure 6.12: Dam Break with obstacle. Time evolution of water heights and pressures at different sensors. Comparison between experimental and numerical results presented in [198] and the numerical results obtained with the present method.



(a) After Delaunay

(b) After P-Smoothing

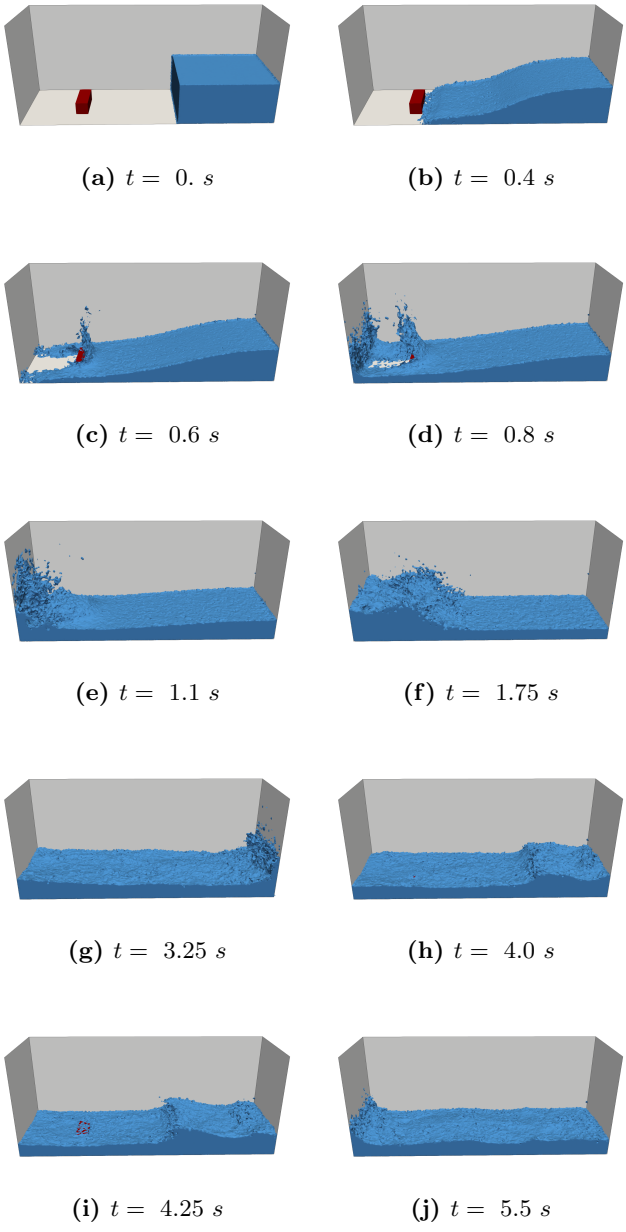
(c) After P-Smoothing and GETMe smoothing

Figure 6.13: Dam break with obstacle. Plot of the stable time step at each remeshing computed after: (a) Delaunay triangulation, (b) P-Smoothing, (c) P-Smoothing and GETMe smoothing

able because of the consequent required computational time. The action of the P-Smoothing leads to a new mean value $\Delta t_{mean} = 2.41\ 10^{-6}\ s$, still showing some situations where the improvement is less effective, so that the minimum values are around $10^{-8}\ s$. The combined action of P-Smoothing and GETMe smoothing gives an additional improvement on the mean value, leading to the final $\Delta t_{mean} = 5.43\ 10^{-6}\ s$ (almost two orders of magnitude larger than the corresponding minimum obtained with the Delaunay mesh), but more remarkably, it effectively cures the critical situations left by the P-Smoothing, obtaining a final $\Delta t_{min} = 9.06\ 10^{-7}\ s$ (almost four orders of magnitude greater than the Delaunay mesh minimum) with values tightly grouped around the mean one. These very satisfying results are obtained in a reasonable amount of time: only 1.2 % of the overall analysis duration, as summarized in the table in Figure 6.11.

6.6.4 Water drop fall into a cylindrical reservoir

A water sphere falling into a cylindrical box containing water at rest is here considered [6]. Figure 6.15 shows the problem geometry, together with the table listing geometrical and other analysis parameters. Figure 6.16 shows some snapshots of the simulation. After the impact, one can see the resulting oscillation with the creation of a central jet globally respecting the cylindrical symmetry. The comparison with the results presented in the literature [6] shows good agreement in terms of timing and free surface profile. In this analysis, performed with a fine mesh (approximately 1.7 M elements), the results of the smoothing algorithm are particularly interesting. Figure 6.17 shows how the stable time step is improved at each remeshing, starting from the one computed directly after the Delaunay triangulation in Figure 6.17(a), then the one computed after the action of P-Smoothing in Figire 6.17(b) and, finally, the results of P-Smoothing combined with GETMe smoothing in Figure 6.17(c). Once again, one can notice that the P-Smoothing provides almost every time a remarkable gain (the mean value during the analysis increases by almost two orders of magnitude from $3.5 \cdot 10^{-8}\ s$ to $1.99 \cdot 10^{-6}\ s$), but still presents some circumstances where its action is less effective, so that the minimum during the analysis is $2.58 \cdot 10^{-8}\ s$. The complementary action of GETMe smoothing fully corrects all the situations where the P-Smoothing is ineffective, leading to a much larger minimum time step ($2.32 \cdot 10^{-6}\ s$) and reducing a lot its variability around its mean value of $4.18 \cdot 10^{-6}\ s$. Furthermore, from the Table in Figure 6.15 one can see that these results have been obtained at a low computational cost, (only 3% of the overall computational time). Indeed, despite the huge number of elements in the mesh at hand, the possibility



(a) $t = 0.\ s$ **(b)** $t = 0.4\ s$

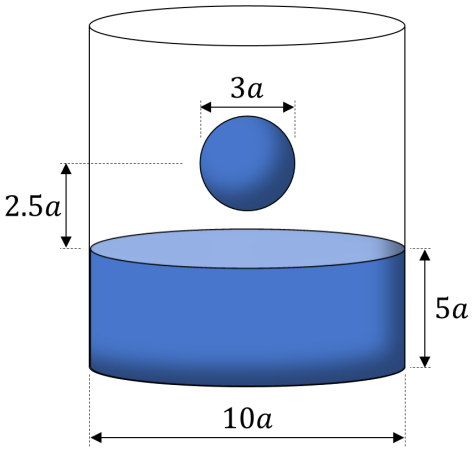
(c) $t = 0.6\ s$ **(d)** $t = 0.8\ s$

(e) $t = 1.1\ s$ **(f)** $t = 1.75\ s$

(g) $t = 3.25\ s$ **(h)** $t = 4.0\ s$

(i) $t = 4.25\ s$ **(j)** $t = 5.5\ s$

Figure 6.14: Dam break with obstacle. Snapshots of the simulation at different time steps.



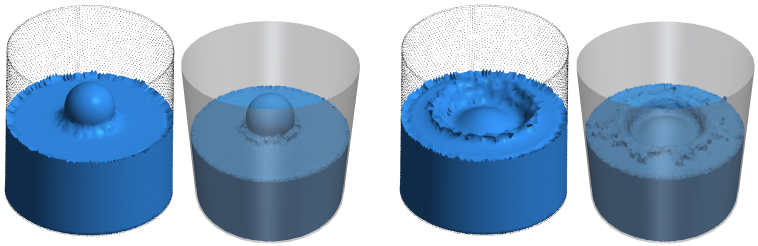
Fluid Properties	
Density ρ_f	1000 kg/m^3
Viscosity μ_f	10^{-3} $Pa \cdot s$
Geometry	
a	0.1 m
Mesh	
n. nodes	$\sim$ 300 k
n. elem.	$\sim$ 1.7 M
Analysis	
Time simulated	1 s
Total analysis	89 h
% Solver	95 %
% Remeshing	2 %
% Smoothing	3 %
n. threads	8

Figure 6.15: Water drop fall. Geometry and parameters of the analysis.

to restrict the smoothing action only to the portions containing bad quality elements allows to obtain a very efficient procedure.

6.6.5 3D Breaking dam flow over an elastic obstacle

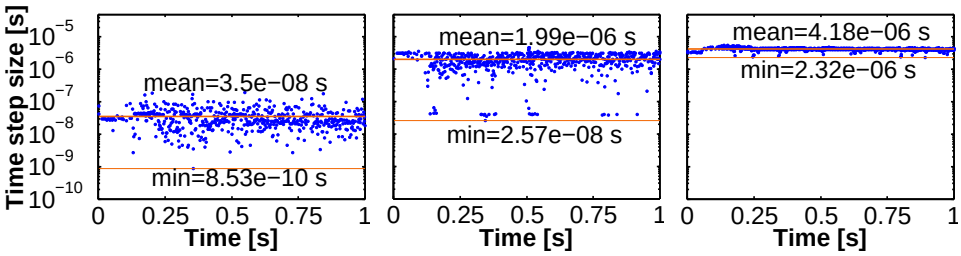
A first 3D FSI example is here considered. The problem of breaking dam flow over an elastic obstacle already addressed in Section 4.4.3 is here extended to the 3D setting. The geometry and the fluid and structural parameters are taken from Section 4.4.3 (Figure 4.10) considering an out of plane thickness of the tank of $0.15m$. The choice of an average mesh size of $4mm$ leads to a fluid mesh of $440k$ linear tetrahedra and a structural mesh of $3.8k$ 8-nodes continuum brick elements. Figure 6.18 presents some snapshots of the simulation at different time instants, showing a qualitative realistic representation of the complex dynamics of the problem. The comparison with the corresponding snapshots in Figure 4.11 shows a good agreement between the 2D and 3D results. Moreover, Figure 6.19 plots the time history of the horizontal displacement of the tip of the elastic obstacle. The numerical results of the 3D analysis are compared with the ones obtained with the same method in the 2D framework and with other methods presented in the literature. Also this quantitative comparison shows a good agreement in terms of amplitude and timing of the first peak of deflection of the elastic obstacle. As already commented in the 2D case, some discrepancies between the different numerical approaches arise and accumulate during



(a) $t = 0.175\ s$ **(b)** $t = 0.175\ s$ **(c)** $t = 0.275\ s$ **(d)** $t = 0.275\ s$

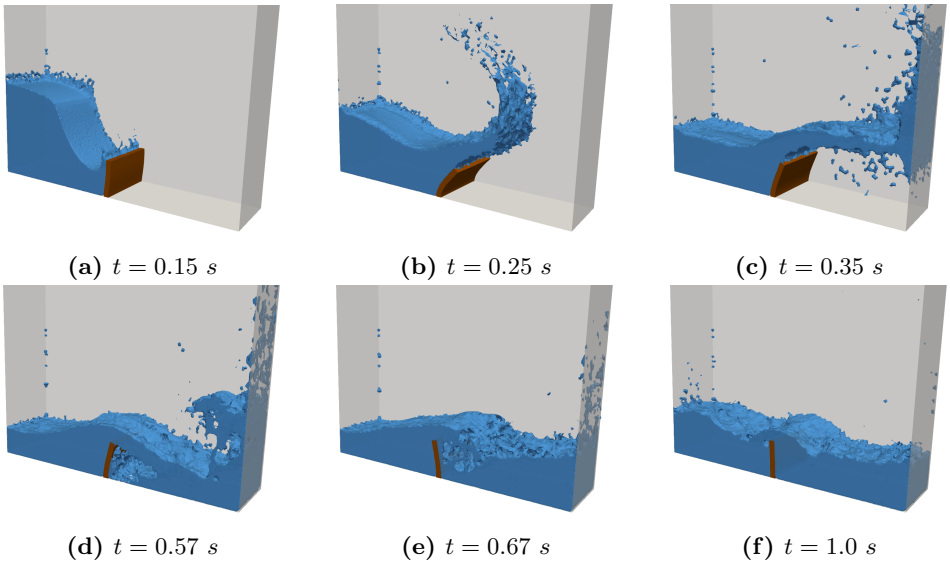
(e) $t = 0.5\ s$ **(f)** $t = 0.5\ s$ **(g)** $t = 0.9\ s$ **(h)** $t = 0.9\ s$

Figure 6.16: Water drop fall. (b)-(d)-(f)-(h) Snapshots of the simulation of the present method at different time steps compared with the corresponding ones (a)-(c)-(e)-(g) presented in [6]



(a) After Delaunay **(b)** After P-Smoothing **(c)** After GETMe smoothing

Figure 6.17: Water drop fall. Plot of the stable time step at each remeshing computed after: (a) Delaunay triangulation, (b) P-Smoothing, (c) P-Smoothing and GETMe smoothing



(a) $t = 0.15\ s$ **(b)** $t = 0.25\ s$ **(c)** $t = 0.35\ s$

(d) $t = 0.57\ s$ **(e)** $t = 0.67\ s$ **(f)** $t = 1.0\ s$

Figure 6.18: 3D Breaking dam flow over an elastic obstacle. Snapshots of the simulation at different time instants.

time in the following oscillations. However, this was expected due to the complex dynamics of the phenomenon. The results in terms of performances of the mesh smoothing algorithm are showed in Figure 6.20. Also in this case there is a remarkable gain in the values of the stable timestep size. The mean and minimum values of the time step sizes computed on the Delaunay mesh are $\Delta t_{mean} = 3.05\ 10^{-8} s$ and $\Delta t_{min} = 4.9\ 10^{-10} s$, respectively. After the action of the P-Smoothing algorithm they increase to the values of $\Delta t_{mean} = 5.99\ 10^{-7} s$ and $\Delta t_{min} = 1.44\ 10^{-8} s$, while the GETMe smoothing leads to the final employed time steps having a mean value of of $\Delta t_{mean} = 1.45\ 10^{-6} s$ and a minimum value of $\Delta t_{min} = 2.71\ 10^{-7} s$. The overall gain is of almost two orders of magnitudes on the mean value and almost three orders of magnitude on the minimum one. These results are obtained at the price of spending on the smoothing algorithms only the 8% of the total 34 h of computing time for this analysis (Table 6.2).

6.6.6 3D Breaking dam flow through an elastic gate

The 2D problem addressed in Section 4.4.2 is here simulated in a 3D framework. The geometry is represented in Figure 4.10, considering an out of plane tank thickness of $0.1m$. Let us underline that this example and the previous one 6.6.5 are typical 2D problems and a 3D simulation was not strictly necessary, but it offers a validation of the 3D FSI approach.

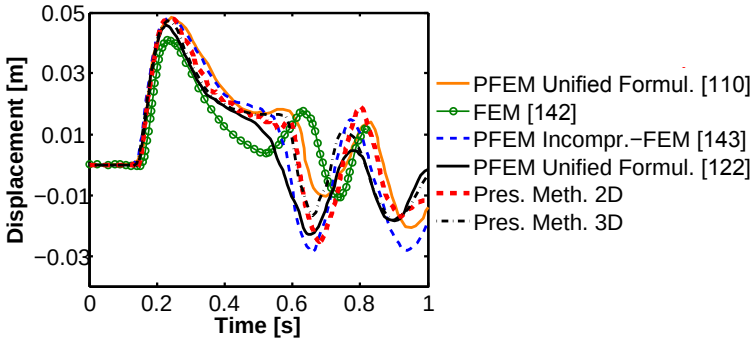
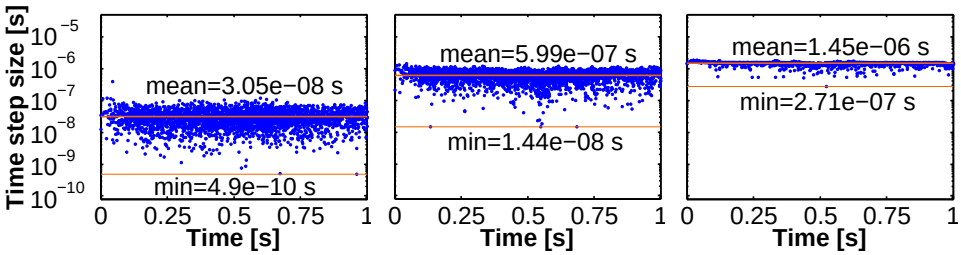


Figure 6.19: 3D Breaking dam flow over an elastic obstacle. Evolution in time of the horizontal displacement of the tip of the obstacle.

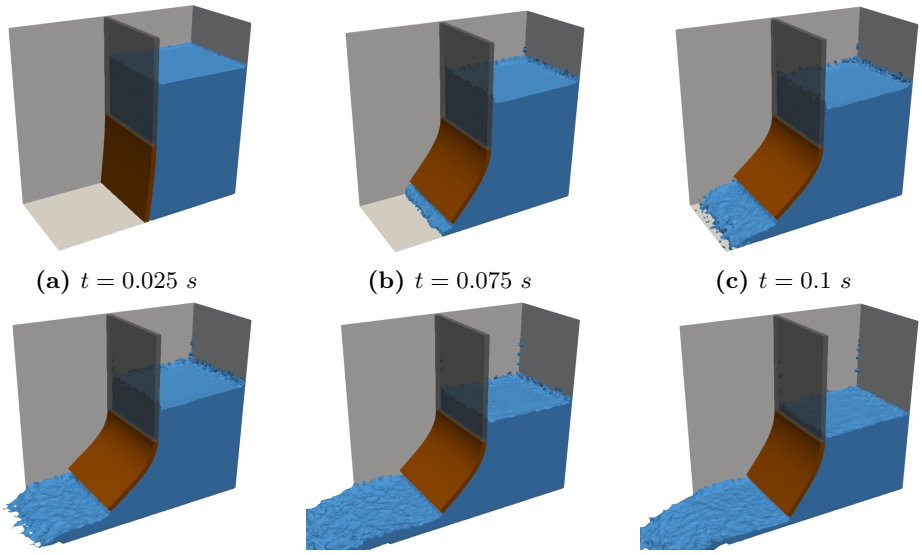


(a) After Delaunay **(b)** After P-Smoothing **(c)** After GETMe smoothing

Figure 6.20: 3D Breaking dam flow over an elastic obstacle. Plot of the stable time step at each remeshing computed after: (a) Delaunay triangulation, (b) P-Smoothing, (c) P-Smoothing and GETMe smoothing

Analysis	
Time simulated	1 s
Total analysis	31 h
% Solver	81 %
% Remeshing	11 %
% Smoothing	8 %
n. threads	8

Table 6.2: 3D Breaking dam flow over an elastic gate. Details of the computing time of the analysis.



(a) $t = 0.025\ s$ **(b)** $t = 0.075\ s$ **(c)** $t = 0.1\ s$

(d) $t = 0.12\ s$ **(e)** $t = 0.2\ s$ **(f)** $t = 0.4\ s$

Figure 6.21: 3D Breaking dam flow through an elastic gate. Snapshots of the simulation at different time instants.

The same geometry and material properties for the fluid and the structure adopted in Section 4.4.2 are employed. An average mesh size of $0.002m$ leads to a mesh of $740k$ fluid tetrahedra and $10k$ 8-nodes brick structural elements. The comparison of the snapshots of the 3D analysis in Figure 6.21 with the corresponding ones of the 2D analysis in Figure 4.8 shows a good qualitative agreement. A quantitative validation is obtained with the graph in Figure 6.22, showing how also the 3D analysis has a good prediction of the experimental time history of the displacement of the bottom edge of the elastic gate. The mesh smoothing performances are represented in Figure 6.23, while Table 6.3 summarizes the details of the computing time of the analysis. Once again, the combined action of the P-Smoothing and N-Smoothing algorithms leads to an increase of more than one order of magnitude on the mean stable time step (namely, from $\Delta t_{mean} = 1.16\ 10^{-8} s$ on the Delaunay mesh to $\Delta t_{mean} = 7.0\ 10^{-7} s$ on the smoothed mesh) and more than two orders of magnitude on the minimum time step (namely, from $\Delta t_{min} = 6.98\ 10^{-10} s$ on the Delaunay mesh to $\Delta t_{min} = 1.87\ 10^{-7} s$ on the smoothed mesh) with a much reduced variation around the mean value. This improvement is obtained spending in the smoothing algorithm only 7.% of the overall computing time of the analysis.

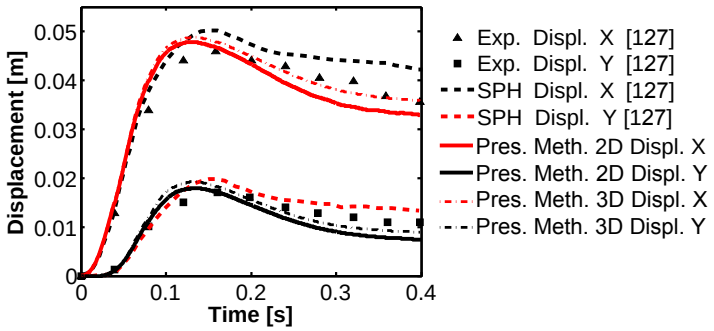
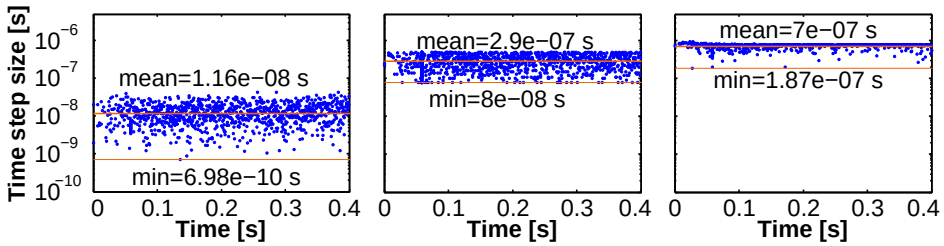


Figure 6.22: 3D Breaking dam flow through an elastic gate. Evolution in time of the horizontal and vertical displacements of the gate tip.



(a) After Delaunay **(b)** After P-Smoothing **(c)** After P-Smoothing and GETMe smoothing

Figure 6.23: 3D Breaking dam flow through an elastic gate. Plot of the stable time step at each remeshing computed after: (a) Delaunay triangulation, (b) P-Smoothing, (c) P-Smoothing and GETMe smoothing

Analysis	
Time simulated	0.4 s
Total analysis	37 h
% Solver	83 %
% Remeshing	10 %
% Smoothing	7 %
n. threads	8

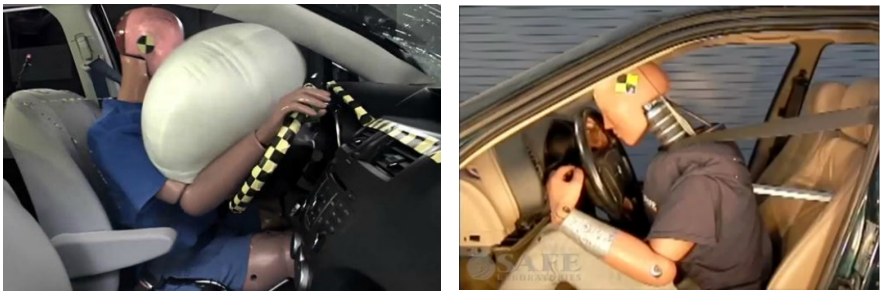
Table 6.3: 3D Breaking dam flow through an elastic gate. Details of the computing time of the analysis.

7 Industrial Application: Airbag deployment

7.1 Introduction

An airbag is a passive automotive safety system built into the steering wheel and various other strategic locations of a vehicle. In the case of accidents, these inflatable cushions restrain automobile passengers dissipating their kinetic energy and thereby reducing the risk of injury.

The design of inflatable restraint system is an engineering problem of great challenges. Such systems deploy completely in a time frame of 40-50ms or less. They must be able to handle different scenarios of impacts and the interaction with different passengers, even in the case of vulnerable ones such as children. Among these scenarios, the case of the passenger impact before the airbag is fully inflated must be carefully analyzed. In fact, despite these safety devices have strongly reduced the car accident fatalities, there is a potential risk of injury caused by the deploying airbag itself, mainly related to the case of passengers located within the deployment region before the full inflation (Figure 7.1-b) [199]. These conditions are usually referred to as Out-of-Position (OoP) conditions. On the contrary, the so-called In-Positions (IP) conditions are related to the standard interaction of the passenger with the fully inflated airbag (Figure 7.1-a) [199]. Consequently, since the early stages of development, airbag technology has been undergoing continual evolution in terms of design, materials and performances. In the designing process the numerical simulation is getting an increasing importance. Compared with the traditional experiments, the numerical simulation has remarkable advantages in terms of cost, efficiency, flexibility and amount of data which can be extracted from the analysis.



(a) **(b)**

Figure 7.1: (a) In-Position (IP) condition: the passenger impacts the fully inflated airbag. (b) Out-of-Position (OoP) condition: the passenger is located in the region of the airbag deployment during its inflation, with the high risk of severe injury.

7.2 Simulation approaches

There are mainly two categories of approaches to the airbag simulation in the automotive safety industry [200]. The first one includes the methods which adopt the simplifying assumption of neglecting the fluid dynamics inside the airbag chamber during the inflation process. The second one includes the methods which model the inner fluid dynamics, leading to a fully FSI problem. In the first category the most used approach is the Uniform Pressure Method (UPM) [201], where the inflation is approximated by a time-varying, spatially-uniform pressure field. More in details, in its simplest version, a scalar pressure can be computed at every time step of the analysis starting from the airbag chamber volume and the gas mass inflow law. Such pressure is then uniformly applied on the airbag to compute the updated configuration [200]. This simplification leads to very efficient analyses for IP conditions, because the uniform pressure distribution well approximates the situation of a fully inflated airbag. On the contrary, during the inflation process large pressure gradients are present inside the airbag, violating the UPM assumptions. Consequently, when the OoP condition is addressed to study the interaction between the passenger and the airbag during its deployment, the UPM leads to unrealistic results and a full FSI analysis is required [7,200]. Among the most used FSI numerical approaches for OoP airbag simulations in the automotive industry, let us recall the ALE method implemented in the commercial software LS-DYNA, and the CEL method implemented in Abaqus. Such methods have been briefly introduced in Section 4.1.

A clear example of the differences between the two types of approaches

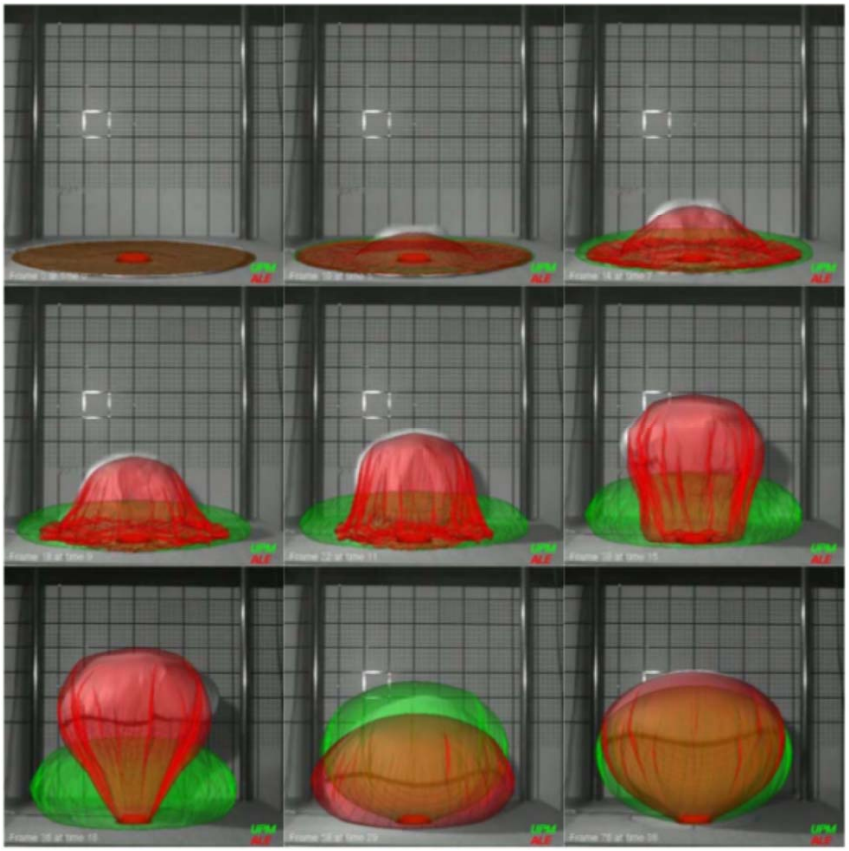


Figure 7.2: Flat airbag deployment. Comparison of the experimental results with the numerical ones obtained with a UPM approach (green) and an ALE approach (red) [7]

is represented in Figure 7.2 taken from [7]. Several snapshots at different time instants of an experimental test involving a flat airbag deployment are superimposed with the results of two numerical analyses performed with the UPM approach (green) and ALE method (red). One can observe that at the final stage of the deployment both numerical results well describe the experimental configuration. However, during the whole deployment process the UPM method is not able to reproduce correctly the real dynamics, while the inclusion of the unsteady fluid-dynamics in the ALE model leads to good representation of the entire process.

As previously commented in Chapters 1 and 2, many features of the present PFEM-FEM FSI method make it potentially very effective to address the challenging issues of a full FSI simulation of the airbag deploy-

Figure 7.3: Stages of the airbag deployment.

ment. First of all, due to the extremely fast dynamics and high degree of non-linearity, it is commonly accepted that these problems need to be addressed with an explicit solver. Secondly, this phenomenon sees a strong evolution of the interface between the fluid and the structure. The airbag is initially empty and folded inside the device housing and then the fluid flow causes the complex deployment until the completely inflated configuration (Figure 7.3). Such strong variations of the fluid structure interface can be naturally tracked by the fully Lagrangian kinematics of the present method. Finally, a partitioned approach is preferable, since the large number of complex modelling aspects both on the fluid and structural sides would lead to several issues in the implementation of a unique monolithic solver.

7.3 Inflator and simplifying assumptions

The activation of the airbag deployment occurs when an acceleration sensor in the vehicle detects a severe deceleration related to a car accident. This triggers a controlled explosion inside the airbag inflator that produces a large volume of a gas mixture (pyrotechnic inflators). The gas mixture expands and rapidly inflates the chamber of the airbag, possibly combined with a pre-compressed inert gas (in hybrid airbags) [199]. Clearly, the results of the numerical simulations are strongly influenced by the features of the gas flow, mainly the mass flow rate, the direction of the gas flow and the time history of gas temperature and composition [7]. Such features can significantly vary depending on the chosen type of inflator and the extrapolation from experimental tests of the synthetic input data required by numerical simulations is still under discussion [202, 203].

Unfortunately this amount of input data was not available for the simulations here presented, considering that the works in the literature are generally omitting relevant information that are confidential for the automotive industry. Consequently, let us remark that the aim of the present work is limited to show how the proposed PFEM-FEM FSI method may be employed in this challenging real engineering application with potentially

effective results. Only qualitative comparisons with the tests presented in the literature are perfomed. The simplifying assumptions of a normal injection of gas of constant composition and at constant temperature has been adopted [204, 205]. In the present approach, the constant temperature assumption allows to keep unaltered the fluid mathematical model described in Chapter 2 neglecting the energy conservation equation. Despite the strong influence of such an assumption, the lack of data on the temperature time history would compromise any further efforts to increase the consistency with the physical phenomenon through the introduction of the energy conservation equation.

More in details, the gas is here modelled as a light, inviscid air-like fluid, so that the fluid density is set to $\rho_{0,f} = 1.3kg/m^3$, the viscosity is set to $\mu_f = 0Pa\ s$ and the parameters for the equation of state (2.18) are: dilation waves propagation speed $c_f = 340m/s$ and specific heat ratio $K'_{0,f} = 1.4$. Let us remark that with this choice of fluid parameters and under the assumption of constant temperature, the EOS (2.18) is basically equivalent to an isentropic perfect gas law of the type (see [24]):

$$p_f(\rho_f) = p_{0,f}\left(\frac{\rho_f}{\rho_{0,f}}\right)^{K'_{0,f}} \tag{7.1}$$

Finally, the mass flow rate curve is certainly the most influencing input data to represent correctly the transient process of airbag deployment [202, 206]. Unfortunately, the curve adopted in the experimental and numerical tests that will be addressed in Section 7.5 is not provided by the reference paper [7]. However, from the review of many other works in the literature a qualitative common behaviour of such curves is observed [202, 205–207] which show an initial fast, parabolic-like increase of the mass rate with a following exponential-like decay. In the numerical examples of Section 7.5 the mass flow rate curve depicted in Figure 7.4 has been employed. The peak value and the timing of the following decay have been calibrated in order to reproduce an equivalent dynamics in the experiment of Section 7.5.2.

7.4 Fabric behaviour

Airbag fabrics are typically made of plain weave of nylon yarns, being the warp and fill yarns interlaced in a regular sequence of one-under and one-over (Figure 7.4). Woven fabrics are characterized by high flexibility allowing them to fold and conform to a variety of shapes. This structure leads to a macroscopically heterogeneous, anisotropic material, which can undergo large deformations and exhibit non-linear behavior. The non-linearity arises

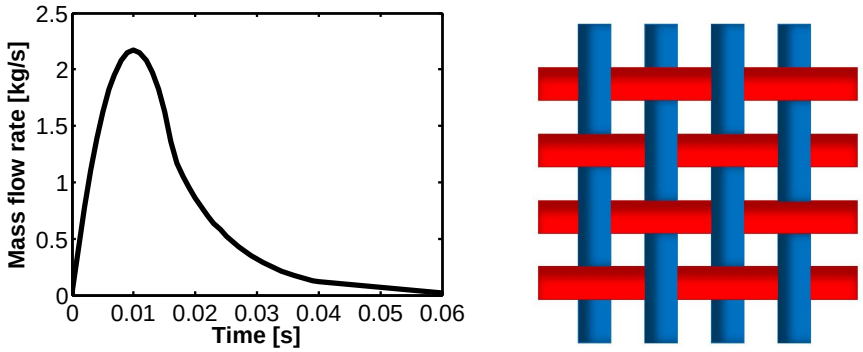


Figure 7.4: (a) Mass Flow rate curve. (b) Schematical representation of the microstructure of the fabric as a plain weave of nylon yarns.

from different sources: the non-linear response of the individual yarns, the exchange of crimp between the fill and the warp yarns as they are stretched, the contact and friction between the yarns in cross directions and between the yarns in the same direction [199]. The tensile response in the fill and warp directions may be coupled due to the aforementioned crimp exchange, while the compression stiffness is usually neglected. Under in-plane shear deformation, the fill and warp direction yarns rotate with respect to each other. The resistance increases with shear deformation as lateral contact is formed between the yarns in each direction. The bending stiffness is usually neglected, since the system of fibers are only restricted by friction [137].

Entering into the details of the airbag fabric material is beyond the scope of this work. This brief description was only aimed to highlight the complexity of the material model. In the present PFEM-FEM FSI approach, the coupling with a commercial software such as Abaqus/Explicit allows to introduce in the simulations all its features to obtain the level of complexity required by the airbag problem at hand. More in particular, the material behaviour *FABRIC, specifically conceived for these kinds of applications, is employed. Experimental data are required to characterize the material, namely the response in the two principal directions, as well as the shear response, under the assumption that they are all independent one to the others. Consequently the fabric stress response depends only on the fabric strain in that component. Other features such as rate dependent, damaged elastic or elastic-plastic behaviours may be added in the material model. Moreover, advanced algorithms to deal with the complex phenomena of folding and wrinkling are included in the contact modelling of the simulations here presented (see *contact controls assignments* in [137]).

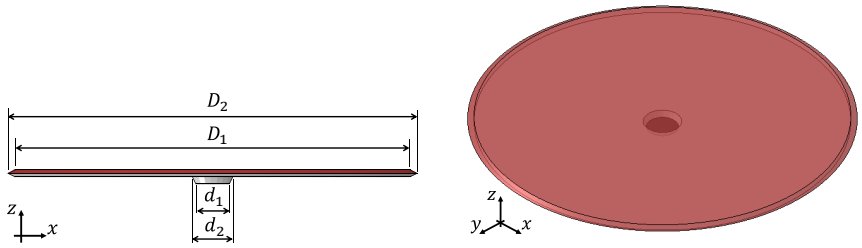
Fabric Material input data					
Density ρ_s		726 kg/m^3			
Direction 1		Direction 2		Shear	
Nom. Strain [-]	Nom. Stress $[MPa]$	Nom. Strain [-]	Nom. Stress $[MPa]$	Nom. Strain [-]	Nom. Stress $[MPa]$
-1	-22.2	-1	-28.1	0	0
-0.9	-17.76	-0.9	-22.48	0.52	4.94
-0.5	-0.0022	-0.5	-0.0028	1.57	54.37
0	0	0	0		
0.1	17.76	0.1	22.48		

Table 7.1: Input data for the Abaqus *FABRIC material model.

Nevertheless, the lack of data about the experimental tests that are here addressed led us to the choice of adopting a standard calibration of the set of parameters of the model, namely the ones provided in the *Abaqus Example Problems Guide* [137] for the "Side curtain airbag impactor test". Only the standard non-linear elastic behaviour of the *FABRIC model is considered in the applications here presented, i.e no damage, rate dependent or elastic-plastic features are included. The input data are summarized in Table 7.1. Let us remark that, despite the negligible fabric stiffness under compression, a common strategy is to specify a compressive stiffness after a certain range of strain, to prevent the collapse of wrinkled elements under compression (see *Abaqus Example Problems Guide* in [137]) .

7.5 Numerical simulations

In the present work, two numerical test cases inspired by the experimental tests presented in [7] are addressed. In both cases, the airbag has an initial flat configuration on an horizontal plane. These tests have been used in the literature for the validation of other numerical approaches [8,200]. No folded configuration is considered in the present work. The folding procedure may be introduced in the Abaqus software, as it is done in [8]. Nevertheless, it would introduce some technical issues in the remeshing procedure in the present version of the PFEM software. Indeed, the use of the Standard Delaunay triangulation and the Alpha-shape algorithm may not have the proper resolution in the small gaps between one fold and the other. In this application, however, characterized by a closed flow inside the airbag chamber, one could overcome these issues using to the Constrained Delaunay triangulation, not available in the present version of the proposed PFEM



(a) **(b)**

Figure 7.5: Flat airbag deployment. Geometry of the problem

software, where the boundary nodes can be forced and a non-convex mesh can be obtained.

7.5.1 Flat airbag deployment

The first numerical test considers the deployment of a flat horizontal airbag with no interaction with any other bodies. The geometry of the problem is depicted in Figure 7.5. The diameter of the initially circular flat configuration is $D_2 = 0.6m$. More in details, two horizontal fabric layers of diamater $D_1 = 0.58m$ are separated by an initial internal gap of $0.01m$, while the equatorial fold is schematically represented with a corner edge. The diameter of the circular area representing the inflator is $d_1 = 0.05m$, while $d_2 = 0.06m$. Finally, a rigid horizontal plate of diameter $D_3 = 1m$ is located at the level of the inflator to represent the supporting plane. The structural mesh is given by $141k$ 3-node membrane element (M3D3 in the Abaqus elements library [137]) with an out-of-plane thickness of $0.4mm$. The fluid parameters are defined in Section 7.3. At the beginning of the analysis, the fluid mesh fills the small gap inside the airbag chamber with $483k$ tetrahedral elements. Then, due to the fluid injection it rapidly increases till an average mesh of approximately $700k$ elements for the inflated configuration. The mass flow rate employed in the simulation is the one presented in Figure 7.4. Figures 7.6 shows several snapshots of the simulation at different time instants, while Figures 7.7 gives an insight on the inner fluid dynamics with a central cut which shows the contour plot of the fluid velocity field at the corresponding instants. From the velocity field one can observe the fact that the hypothesis of a uniform inner pressure adopted in the UPM approach is clearly not realistic. A real qualitative comparison with the results shown in Figure 7.2 is compromised by the aforementioned lack of input data. However, one can observe that the present FSI method seems to be able to reproduce all the peculiar stages of the inflation process.

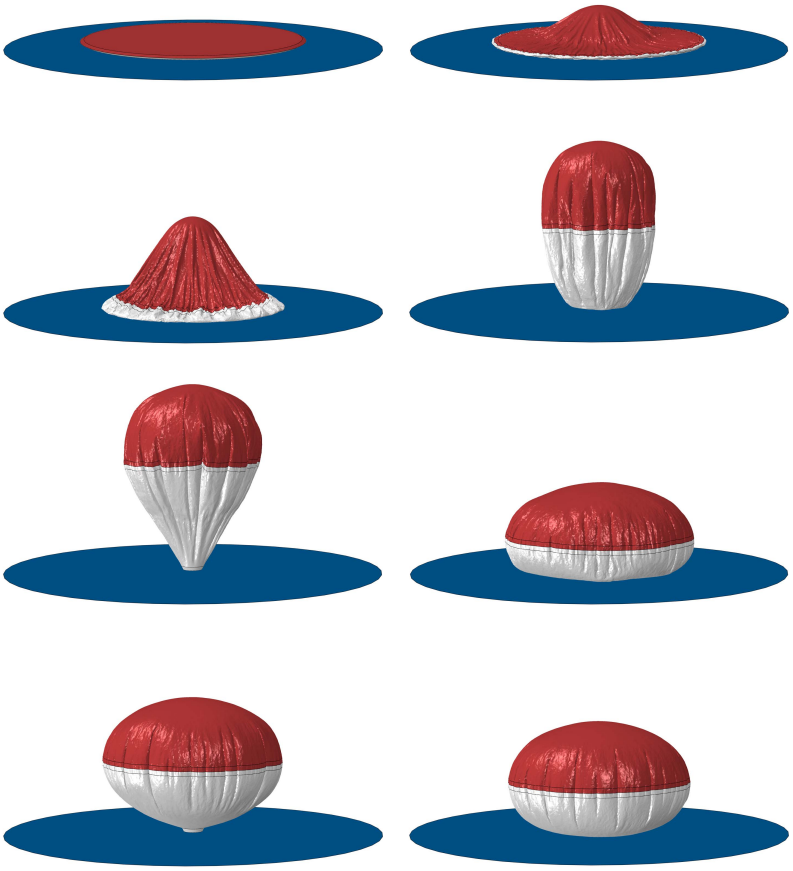


Figure 7.6: Flat airbag deployment. Snapshots of the simulation at several instants: structural output. Sequence from left to right, from top to bottom.

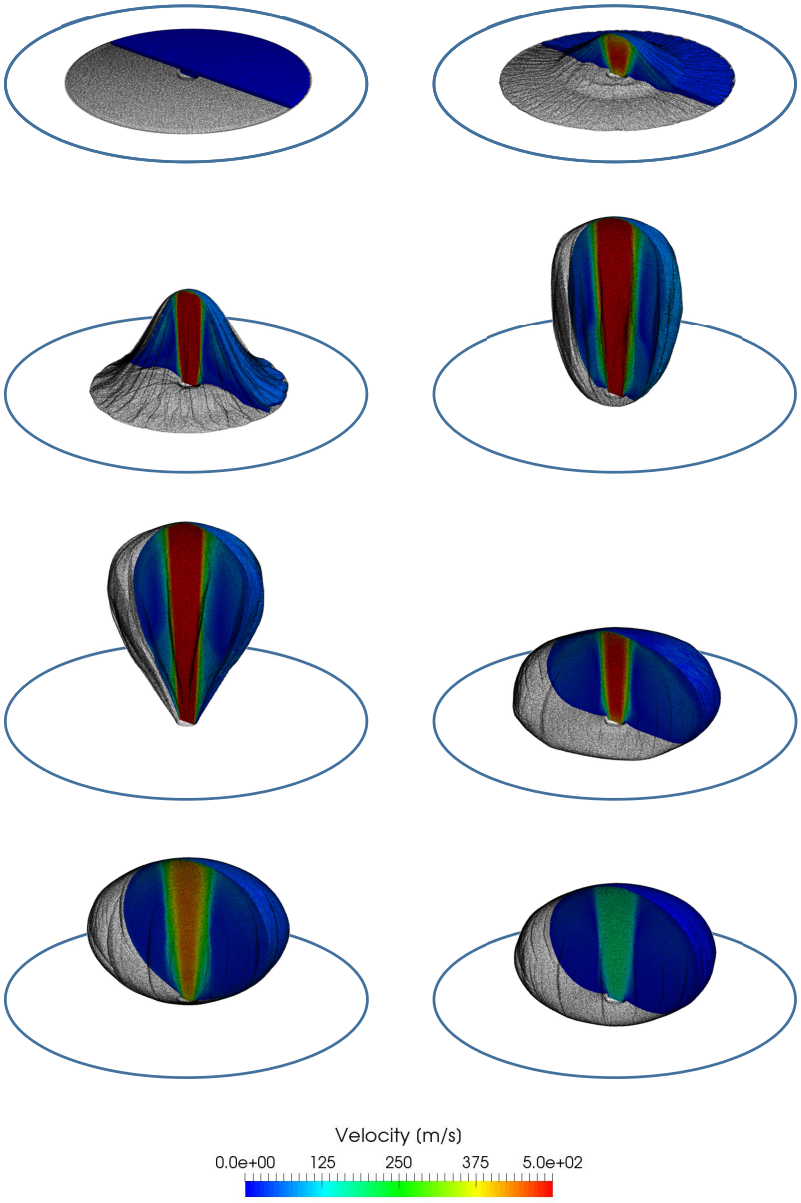


Figure 7.7: Flat airbag deployment. Snapshots of the simulation at several instants: fluid output. A middle cut shows the velocity field inside the airbag chamber. Sequence from left to right, from top to bottom.

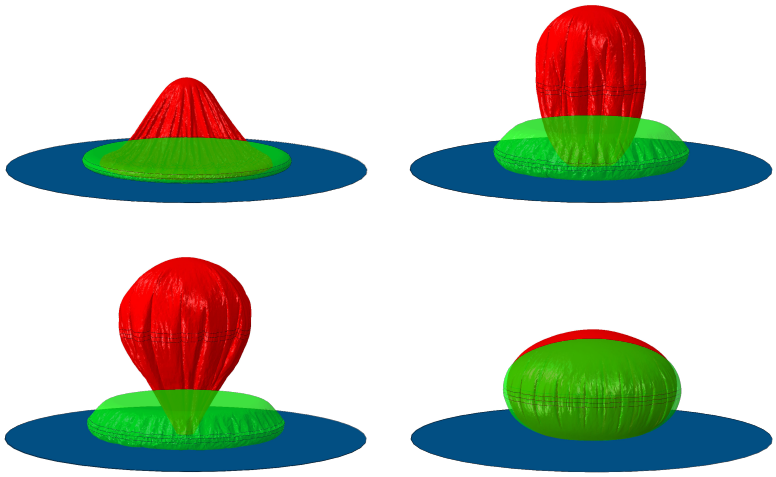


Figure 7.8: Flat airbag deployment. Comparison of numerical results obtained with the UPM approach in Abaqus (green) and the PFEM-FEM FSI method here presented (red).

The same test case has been performed in Abaqus/Explicit using the UPM approach [137]. The same geometry, material parameters and mass-flow rate curve of an equivalent fluid have been employed. Figure 7.8 shows the comparison at different time instants of these results with the ones obtained with the PFEM-FEM FSI approach here presented. Similarly to what have been commented for Figure 7.2 [7], one can see that the UPM results agrees with the ones of the FSI approach only in the fully inflated configuration. On the contrary, a completely different behaviour is reproduced during the inflation process. This provides a further confirmation that a full FSI simulation is required to represent the process of deployment of the airbag and study their interaction with passengers in OoP conditions, otherwise completely unrealistic results can be obtained.

Finally, Table 7.2 presents the computing time of the present analysis. An overall process of $0.06s$ has been simulated with analysis of 101 hours. One can observe that the computational time spent for the dynamic mesh management is more than 40% of the total computing time, which is a lot more than what has been presented in Chapter 6. This however was expected due to the peculiarities of the phenomenon. With respect to the previously addressed applications, the very fast dynamics and strong evolution of the fluid domain lead to a reduction of the relative time spent by the solver (i.e. a smaller number of time steps to cover the overall simulated time period) and an increase of the time spent for the dynamic mesh

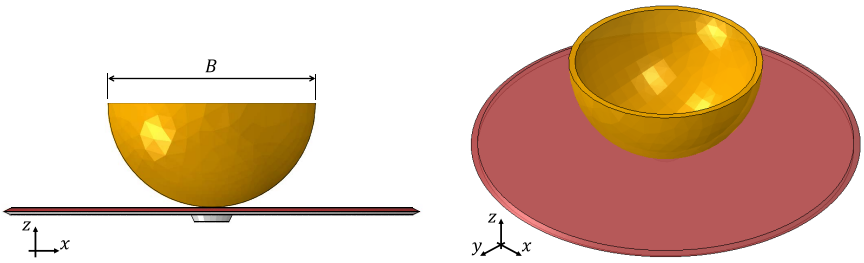
Computational time	
Time simulated	0.06 s
Total analysis	101 h
% Solver	56 %
% Remeshing	30 %
% Smoothing	14 %
n. threads	8

Table 7.2: Flat airbag deployment. Details of the computing time.

management (i.e. much more remeshing during the process). Several modification to the present version of the PFEM-FEM FSI approach could be performed to make the dynamic mesh management more effective in this type of applications. For example, the possibility of exploiting a mixed ALE kinematic description may be investigated, e.g. introducing a background mesh to use fixed Eulerian elements inside the airbag chamber and limit the remeshing procedure only to the layer close to the airbag interface. Furthermore, more efficient versions of the mesh improving tools described in Section 3.2.1.3 may be developed and optimized for this type of application, which employs them more frequently than in standard PFEM ones.

7.5.2 Flat airbag deployment against spherical mass

This second numerical example represents a simplified setting of the airbag impact with a passenger in OoP conditions. The same model of the previous test case is employed, with an additional rigid hemisphere of diameter $B = 0.3m$ and mass $m = 15kg$. The mass is initially located in front of the airbag so that it is impacted and accelerated during its deployment. Once again, the lack of input data prevents the possibility of a quantitative comparison with the results presented in [7, 8]. However, the calibration of the mass rate curve of Figure 7.4 in order to obtain a similar duration of this test allows to make at least a qualitative link with the literature results. Figure 7.10 shows several snapshots of the experimental test presented in [7], the corresponding numerical ones obtained with Abaqus CEL method in [8] and the equivalent stages of the simulation obtained with the present PFEM-FEM FSI approach. From the comparison it seems that the method is able to capture all the principal stages of the process: the initial inflation of the external part of the airbag and the consequent vertical acceleration of the mass until the full deployment. Moreover, Figure 7.11 shows the time history of the vertical mass acceleration compared with the literature ones. The good agreement is of course determined by the proper tuning of the



(a) (b)

Figure 7.9: Flat airbag deployment against spherical mass. Geometry of the problem

Computational time	
Time simulated	0.06 s
Total analysis	109 h
% Solver	53 %
% Remeshing	31 %
% Smoothing	16 %
n. threads	8

Table 7.3: Flat airbag deployment against spherical mass. Details of the computing time.

mass flow rate on this example. However, one can state that reasonable input data extracted from the literature for these type of applications led to results that are aligned with the ones presented for this specific test. Finally, Table 7.3 summarize the computing time of the present analysis, which is equivalent to the previous example presented in Section 7.5.1.

7.5.3 Concluding remarks on the numerical examples

In conclusion, let us underline that the previous numerical examples are not considered by the author as a validation of the present PFEM-FEM FSI approach on the airbag process simulation. In the absence of several input data because of their confidential nature in the automotive industry, a literature research has been performed to find realistic values to simulate equivalent phenomena. On the other hand, the results have been interpreted as a proof of the potentiality of the proposed PFEM-FEM FSI approach to represent a lot of the challenging features of this phenomenon. The main future developments for a possible quantitative validation with the required input data involve:

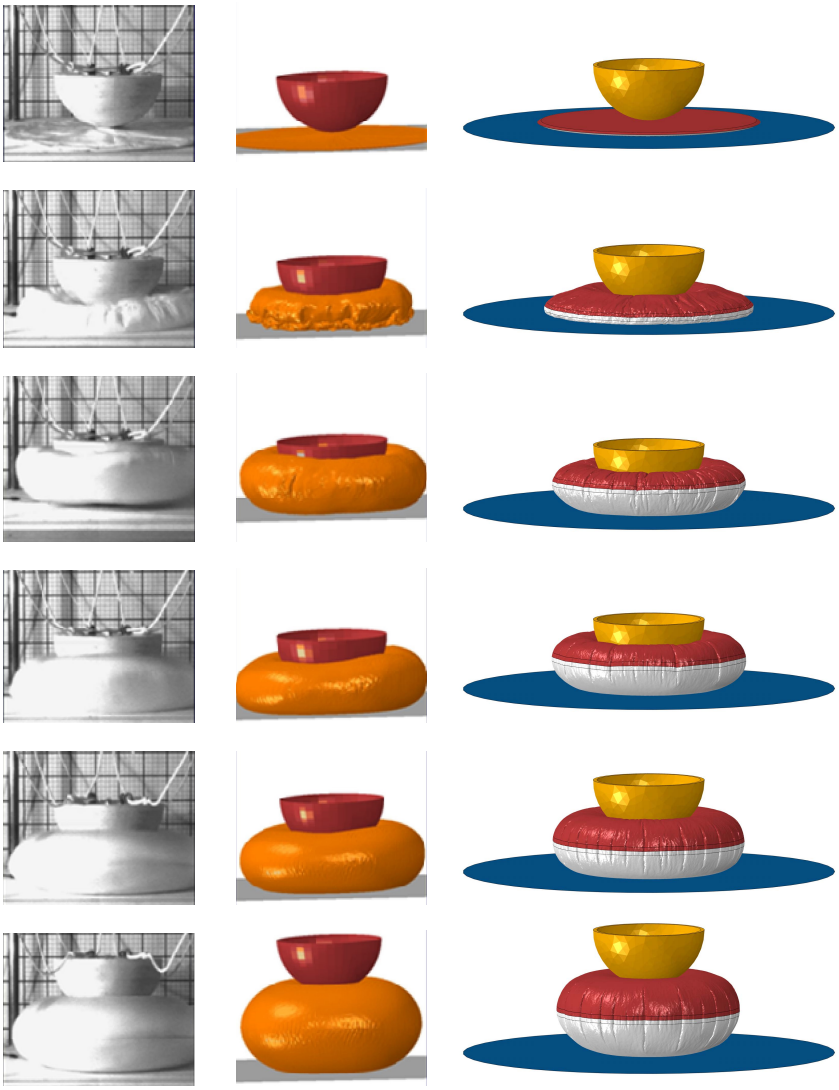


Figure 7.10: Flat airbag deployment against spherical mass. Snapshots of the experimental results presented in [7] (left), the numerical ones in [8] (center) and the numerical ones of the present PFEM-FEM FSI method (right).

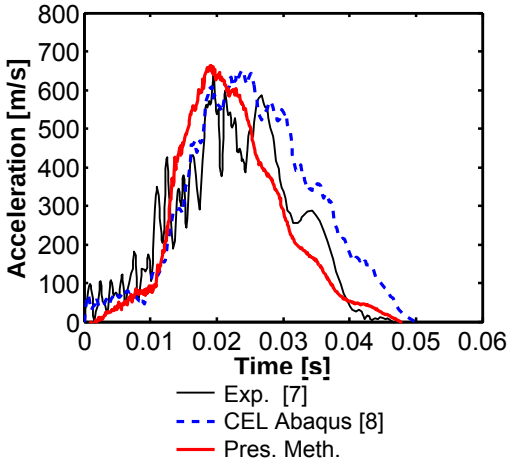


Figure 7.11: Flat airbag deployment against spherical mass. Time history of the acceleration of the impacted mass. Comparison of the experimental results presented in [7], the numerical ones in [8] and the numerical ones of the present PFEM-FEM FSI method.

- The introduction of the equation of energy conservation to reproduce the real thermodynamics of the gas inside the airbag chamber.
- The introduction of the Constrained Delaunay Triangulation, to address the initial folded configuration of the real applications.
- The reduction of the computing time spent in the dynamic mesh management process through the implementation of mesh improving tools optimized for this type of application. Moreover, the possibility to exploit mixed ALE formulation may be investigated to limit the remeshing only to the region close to the airbag interface.

8 Conclusions

This work has been devoted to the development of a numerical tool to efficiently simulate Fluid-Structure Interaction (FSI) problems involving free surface flows or structures undergoing large displacements and possibly characterized by fast dynamics and high level of non-linearity. To this purpose, a partitioned fully explicit Lagrangian Finite Element Method has been presented. The method is based on a novel explicit version of the Particle Finite Element Method (PFEM) for the fluid domain and the commercial software SIMULIA Abaqus/Explicit from Dassault Systèmes for the structural domain. The fully Lagrangian kinematic description is particularly effective for free surface flows and FSI problems with large structural displacements, i.e., when the fluid boundaries can vary significantly. The boundaries and interfaces are naturally defined by the position of the mesh nodes, with no need for *ad hoc* tracking algorithms. Because of the fully explicit nature of the coupled solver, which does not require iteration nor present convergence problems, the proposed method can be an efficient choice for real engineering applications characterized by fast dynamics and/or high degree of non-linearity, therefore where the time step size is intrinsically small.

The novel aspects and contributions of the present work are summarized in the following:

- A novel explicit version of the PFEM has been developed, based on the hypothesis of weakly compressible fluid flow. Several examples have been presented where the method has been validated against analytical, experimental and numerical results. The problem of the mass preservation of the fluid solver has been addressed. It has been verified that no mass variation is introduced by the employed stabilization method and the mass variations are only related to the PFEM remeshing procedure. These are intrinsic in the PFEM method and they can be controlled and reduced refining the mesh size.

- The explicit PFEM code has been coupled with Abaqus/Explicit through the Co-Simulation Services embedded in the SIMULIA software. This allows to perform simulations with an advanced description of the structural domain, including advanced structural material models and contact interactions. The Gravouil and Combescure method has been chosen for the structure-to-fluid coupling. This algorithm, which can be seen as a predictor/corrector scheme, synchronizes the independent solutions of the fluid and structure subdomains, ensuring the strong coupling and the stability of the partitioned approach. The possibility of different time integration steps and nonconforming meshes in the two subdomains is implemented.

- A novel technique has been proposed to introduce non-homogeneous boundary conditions in Lagrangian approaches for fluid flows. Such conditions are of practical interests in several CFD problems involving, e.g., inlet/outlet cross-sections, fluid slip at the boundary walls and symmetry surfaces. The technique is based on a mixed Lagrangian-Eulerian approach, whereby Eulerian nodes are defined only on those parts of the boundary where non-homogeneous conditions are defined. The proposed method is simple, flexible and computationally convenient, since it requires additional computations only for those nodes belonging to the interested portions of the boundary.

- A novel mesh smoothing approach has been developed to address 3D fluid problems with the explicit PFEM, where the mesh distortion management becomes a major issue. This new algorithm is able to fulfil the opposite requirements of avoiding overly distorted elements, which would dramatically reduce the time step size of an explicit solver, and of being fast enough to be applied runtime in the frequent remeshing framework of the PFEM. This is achieved exploiting an elastic analogy that allows for the use of the same explicit and parallelizable architecture of the fluid solver. The smoothing algorithm could be conveniently applied to regularize the mesh and improve the solution of other Lagrangian methods.

- To test the proposed approach on a real industrial application, the simulation of the automotive airbag deployment has been addressed. Despite the lack of available input data presented in the literature, the results seem to confirm the good potentialities of the method in this kind of applications characterized by an extremely fast dynamics and high level of non-linearity.

Future Developments

The main possible future developments of the present thesis are listed in the following:

- Developments aimed to address a wider range of fluids. The introduction in the fluid solver of the energy conservation equation and other equations of state will allow to have a more realistic representation of the fluid behaviour, as well as the possibility to address fluid problems with thermal coupling. Moreover, the introduction of non-Newtonian constitutive relations and the generalization of the solver to the case of multiple fluids can be of interest for many real engineering applications.

- The optimization of the code subroutines and improvements on the parallelization architecture can increase the solver efficiency. Furthermore, the implementation on General Purpose GPU (GPGPU) may be investigated because of the expected good speed-up of explicit solvers.

- The introduction of the Constrained Delaunay Tessellation may improve the capabilities of the remeshing algorithm to deal with geometrically challenging problems such as the simulation of the fully folded airbag deployment.

- The investigation of an ALE kinematic description in the fluid solver may be convenient to limit the remeshing frequency in challenging problems such as the airbag deployment.

- The potentialities of the proposed PFEM-FEM FSI method has to be further validated and tested on other real engineering applications.

Bibliography

[1] M. Cremonesi, A. Frangi, and U. Perego, "A Lagrangian finite element approach for the analysis of fluid-structure interaction problems," *Int. J. Numer. Meth. Engng.*, vol. 84, pp. 610–630, 2010.

[2] Q. Yang, V. Jones, and L. McCue, "Free-surface flow interactions with deformable structures using an SPH-FEM model," *Ocean Engng.*, vol. 55, pp. 136–147, 2012.

[3] U. Ghia, K. N. Ghia, and C. T. Shin, "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method," *J. Comp. Physics*, vol. 48, no. 3, pp. 387–411, 1982.

[4] M. Philippou, Y. Damianou, X. Miscouridou, and G. C. Georgiou, "Cessation of Newtonian circular and plane Couette flows with wall slip and non-zero slip yield stress," *Meccanica*, vol. 52, no. 9, pp. 2081–2099, 2017.

[5] S. Koshizuka and Y. Oka, "Moving particle semi-implicit method: Fully lagrangian analysis of incompressible flows," *Proc. Europ. Congress Comp. Meth. Appl. Sci. Engng.*, vol. 1, no. Sept., pp. 1–16, 2000.

[6] E. Oñate, A. Franci, and J. M. Carbonell, "Lagrangian formulation for finite element analysis of quasi-incompressible fluids with reduced mass losses," *Int. J. Numer. Meth. Fluids*, vol. 74, pp. 699–731, 2014.

[7] B. Beesten, A. Hirth, R. Reilink, R. Remensperger, D. Rieger, and G. Seer, "OOP-Simulation-a tool to design airbags?Current capabilities in numerical simulation," in *Airbag 2004*, (Karlsruhe, Germany), pp. 1–19, 2004.

[8] Dassault Systèmes SIMULIA, "Simulation of Airbag Deployment Using the Coupled Eulerian-Lagrangian Method in Abaqus/Explicit," Tech. Rep. June, 2011.

[9] S. Turek and J. Hron, "Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow," *Fluid-Structure Interaction*, vol. 53, pp. 371–385, 2006.

[10] M. S. Chafi, G. Karami, and M. Ziejewski, "Numerical analysis of blast-induced wave propagation using FSI and ALE multi-material formulations," *Int. J. of Impact Engng.*, vol. 36, no. 10-11, pp. 1269–1275, 2009.

[11] P. Ryzhakov, R. Rossi, A. Viña, and E. Oñate, "Modelling and simulation of the sea-landing of aerial vehicles using the Particle Finite Element Method," *Ocean Engng.*, vol. 66, pp. 92–100, 2013.

[12] C. M. Scotti, A. D. Shkolnik, S. C. Muluk, and E. Finol, "Fluid-structure interaction in abdominal aortic aneurysms: effects of asymmetry and wall thickness," *BioMedical Engng. OnLine*, vol. 4, no. 1, p. 64, 2005.

[13] K. Takizawa and T. Tezduyar, "Computational Methods for Parachute Fluid-Structure Interactions," *Arch. Comput. Meth. Engng.*, vol. 19, no. 1, pp. 125–169, 2012.

[14] A. Y. Tang and N. Amin, "Some Numerical Approaches to Solve Fluid Structure Interaction Problems in Blood Flow," *Abstract Appl. Analysis*, vol. 2014, p. 8 pages, 2014.

[15] M. Zhu and M. H. Scott, "Modeling fluid-structure interaction by the particle finite element method in OpenSees," *Comp. Struct.*, vol. 132, pp. 12–21, 2014.

[16] M. Cremonesi, S. Meduri, U. Perego, and A. Frangi, "An explicit Lagrangian finite element method for free-surface weakly compressible flows," *Comp. Particle Mech.*, vol. 4, no. 3, pp. 357–369, 2017.

[17] S. Meduri, M. Cremonesi, U. Perego, O. Bettinotti, A. Kurkchubasche, and V. Oancea, "A partitioned fully explicit Lagrangian finite element method for highly nonlinear fluid-structure interaction problems," *Int. J. Numer. Meth. Engng.*, vol. 113, no. 1, pp. 43–64, 2018.

[18] M. Cremonesi, S. Meduri, and U. Perego, "Lagrangian-Eulerian enforcement of non-homogeneous boundary conditions in the Particle Finite Element Method," *(SUBMITTED)*, 2019.

[19] S. Meduri, M. Cremonesi, and U. Perego, "An efficient runtime mesh smoothing technique for 3D explicit Lagrangian free-surface fluid flow simulations," *Int. J. Numer. Meth. Engng.*, vol. 117, no. 4, pp. 430–452, 2019.

[20] I. J. Keshtiban, F. Belblidia, and M. F. Webster, "Compressible flow solvers for low Mach number flows-a review," tech. rep., University of Wales, 2004.

[21] W. Bao and S. Jin, "Weakly compressible high-order I-stable central difference schemes for incompressible viscous flows," *Comput. Meth. Appl. Mech. Engng.*, vol. 190, no. 37-38, pp. 5009–5026, 2001.

[22] G. Scovazzi, M. Christon, T. J. R. Hughes, and J. N. Shadid, "Stabilized shock hydrodynamics: I. A Lagrangian method," *Comput. Meth. Appl. Mech. Engng.*, vol. 196, no. 4-6, pp. 923–966, 2007.

[23] V. A. Dobrev, T. E. Ellis, T. V. Kolev, and R. N. Rieben, "High-order curvilinear finite elements for axisymmetric Lagrangian hydrodynamics," *Comput. Fluids*, vol. 83, pp. 58–69, 2013.

[24] A. Bernard-Champmartin and F. De Vuyst, "A low diffusive Lagrange-remap scheme for the simulation of violent air-water free-surface flows," *J. Comp. Physics*, vol. 274, pp. 19–49, 2014.

[25] C. Farhat, A. Rallu, and S. Shankaran, "A higher-order generalized ghost fluid method for the poor for the three-dimensional two-phase flow computation of underwater implosions," *J. Comp. Physics*, vol. 227, no. 16, pp. 7674–7700, 2008.

[26] J. R. Macdonald, "Some simple isothermal equations of state," *Reviews Modern Physics*, vol. 38, no. 4, pp. 669–679, 1966.

[27] J. Fan, H. Liao, R. Ke, E. Kucukal, U. A. Gurkan, X. Shen, J. Lu, and B. Li, "A monolithic Lagrangian meshfree scheme for Fluid-Structure Interaction problems within the OTM framework," *Comput. Meth. Appl. Mech. Engng.*, vol. 337, pp. 198–219, 2018.

[28] J. J. Gilvarry, "Temperature-Dependent equations of state of solids," *J. Appl. Physics*, vol. 28, no. 11, pp. 1253–1261, 1957.

[29] F. D. Murnaghan, "The compressibility of media under extreme pressures," *Proc. National Academy Sci.*, vol. 30, no. 9, pp. 244–247, 1944.

[30] F. D. Murnaghan, "Finite Deformations of an Elastic Solid," *American J. of Math.*, vol. 59, no. 2, pp. 235–260, 1937.

[31] J. Donea and A. Huerta, *Finite element methods for flow problems*. John Wiley & Sons, Ltd., 2003.

[32] F. Brezzi and K. J. Bathe, "A discourse on the stability conditions for mixed finite element formulations," *Comput. Meth. Appl. Mech. Engng.*, vol. 82, no. 1-3, pp. 27–57, 1990.

[33] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Method*. Springer-Veralg, 1991.

[34] N. M. Newmark, "A Method of Computation for Structural Dynamics," *J. Engng. Mech.*, vol. 85, no. 7, pp. 67–94, 1959.

[35] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Publications, Dover, 1987.

[36] R. Courant, K. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics," *IBM J. R&D*, vol. 11, no. 2, pp. 215–234, 1967.

[37] P. Ryzhakov, J. Marti, and S. Idelsohn, "Fast fluid-structure interaction simulations using a displacement-based finite element model equipped with an explicit streamline integration prediction," *Comput. Meth. Appl. Mech. Engng.*, vol. 315, pp. 1080–1097, 2017.

[38] F. Brezzi and J. Pitkaranta, "On the stabilization of finite element approximations of the Stokes problem," in *Efficient Solutions Elliptic Systems*, pp. 11–19, Vieweg and Teubner Verlag, 1984.

[39] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert, "A new finite element formulation for computational fluid dynamics: VIII. The galerkin/least-squares method for advective-diffusive equations," *Comput. Meth. Appl. Mech. Engng.*, vol. 73, no. 2, pp. 173–189, 1989.

[40] E. Oñate, "A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation," *Comput. Meth. Appl. Mech. Engng.*, vol. 182, no. 3-4, pp. 355–370, 2000.

[41] E. Oñate, "Finite increment calculus (FIC): a framework for deriving enhanced computational methods in mechanics," *Advanced Model. Sim. Engng. Sci.*, vol. 3, no. 14, pp. 1–18, 2016.

[42] C. R. Dohrmann and P. B. Bochev, "A stabilized finite elements method for the Stokes problem based on polynomial pressure projections," *Int. J. Numer. Meth. Fluids*, vol. 46, no. 2, pp. 183–201, 2004.

[43] P. B. Bochev, C. R. Dohrmann, and M. D. Gunzburger, "Stabilization of low-order mixed finite elements for the stokes equations," *SIAM J. Numer. Anal.*, vol. 44, no. 1, pp. 82–101, 2006.

[44] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*. Oxford: Elsevier Butterworth-Heinemann, 6th ed., 2005.

[45] P. Ryzhakov, R. Rossi, S. Idelsohn, and E. Oñate, "A monolithic Lagrangian approach for fluid-structure interaction problems," *Comp. Mech.*, vol. 46, no. 6, pp. 883–899, 2010.

[46] F. H. Harlow and J. E. Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *Physics Fluids*, vol. 8, no. 12, pp. 2182–2189, 1965.

[47] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Berlin: Springer-Veralg, 3rd ed., 2002.

[48] A. Caboussat, "Numerical Simulation of Two-Phase Free Surface Flows," *Arch. Comput. Meth. Engng.*, vol. 12, no. 2, pp. 165–224, 2005.

[49] S. McKee, M. F. Tomé, V. G. Ferreira, J. A. Cuminato, A. Castelo, F. S. Sousa, and N. Mangiavacchi, "The MAC method," *Comput. Fluids*, vol. 37, no. 8, pp. 907–930, 2008.

[50] C. W. Hirt and B. D. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *J. Comp. Physics*, vol. 39, no. 1, pp. 201–225, 1981.

[51] R. Scardovelli and S. Zaleski, "Direct Numerical Simulation of Free-Surface and Interfacial Flow," *Annu. Rev. Fluid Mech.*, vol. 31, no. 1, pp. 567–603, 1999.

[52] J. E. Pilliod and E. G. Puckett, "Second-order accurate volume-of-fluid algorithms for tracking material interfaces," *J. Comp. Physics*, vol. 199, no. 2, pp. 465–502, 2004.

[53] A. Cervone, S. Manservisi, and R. Scardovelli, "An optimal constrained approach for divergence-free velocity interpolation and multilevel VOF method," *Comput. Fluids*, vol. 47, no. 1, pp. 101–114, 2011.

[54] M. A. Cruchaga, L. Battaglia, M. Storti, and J. D'Elía, "Numerical Modeling and Experimental Validation of Free Surface Flow Problems," *Arch. Comput. Meth. Engng.*, vol. 23, no. 1, pp. 139–169, 2016.

[55] E. Olsson, G. Kreiss, and S. Zahedi, "A conservative level set method for two phase flow II," *J. Comp. Physics*, vol. 225, no. 1, pp. 785–807, 2005.

[56] M. Sussman, K. M. Smith, M. Y. Hussaini, M. Ohta, and R. Zhi-Wei, "A sharp interface method for incompressible two-phase flows," *J. Comp. Physics*, vol. 221, no. 2, pp. 469–505, 2007.

[57] F. Losasso, R. Fedkiw, and S. Osher, "Spatially adaptive techniques for level set methods and incompressible flow," *Comput. Fluids*, vol. 35, no. 10, pp. 995–1010, 2006.

[58] J. Donea, A. Huerta, J. P. Ponthot, and A. Rodriguez-Ferran, "Arbitrary Lagrangian-Eulerian Methods," *Encyc. Comp. Mech.*, no. 1969, pp. 413–437, 2004.

[59] A. J. Barlow, "A compatible finite element multi-material ALE hydrodynamics algorithm," *Int. J. Numer. Meth. Fluids*, vol. 56, pp. 953–964, 2008.

[60] N. R. Morgan, J. I. Waltz, D. E. Burton, M. R. Charest, T. R. Canfield, and J. G. Wohlbier, "A point-centered arbitrary Lagrangian Eulerian hydrodynamic approach for tetrahedral meshes," *J. Comp. Physics*, vol. 290, pp. 239–273, 2015.

[61] Y. Crutzen, R. Boman, L. Papeleux, and J. P. Ponthot, "Lagrangian and arbitrary Lagrangian Eulerian simulations of complex roll-forming processes," *C. R. Mecanique*, vol. 344, no. 4-5, pp. 251–266, 2016.

[62] R. Boman and J. P. Ponthot, "Efficient ALE mesh management for 3D quasi-Eulerian problems," *Int. J. Numer. Meth. Engng.*, vol. 92, pp. 857–890, 2012.

[63] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," *Astronomical J.*, vol. 82, no. 12, pp. 1013–1024, 1977.

[64] J. J. Monaghan, "An introduction to SPH," *Comput. Physics Communic.*, vol. 48, no. 1, pp. 89–96, 1988.

[65] M. S. Shadloo, A. Zainali, M. Yildiz, and A. Suleman, "A robust weakly compressible SPH method and its comparison with an incompressible SPH," *Int. J. Numer. Meth. Engng.*, vol. 89, pp. 939–956, 2012.

[66] X. Yang, M. B. Liu, S. Peng, and C. Huang, "Numerical modeling of dam-break flow impacting on flexible structures using an improved SPH-EBG method," *Coastal Engng.*, vol. 108, pp. 56–64, 2016.

[67] Z. Chen, Z. Zong, M. B. Liu, and H. T. Li, "A comparative study of truly incompressible and weakly compressible SPH methods for free surface incompressible flows," *Int. J. Numer. Meth. Fluids*, vol. 73, no. 9, pp. 813–829, 2013.

[68] S. Shao and E. Y. M. Lo, "Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface," *Advances Water Resources*, vol. 26, no. 7, pp. 787–800, 2003.

[69] M. S. Shadloo, G. Oger, and D. Le Touzé, "Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges," *Comput. Fluids*, vol. 136, pp. 11–34, 2016.

[70] T. Belytschko, Y. Y. Lu, and L. Gu, "Element-free Galerkin methods," *Int. J. Numer. Meth. Engng.*, vol. 37, no. 2, pp. 229–256, 1994.

[71] S. Garg and M. Pant, "Meshfree Methods: A Comprehensive Review of Applications," *Int. J. Comp. Meth.*, vol. 15, no. 3, pp. 1–85, 2018.

[72] I. Iaconeta, A. Larese, R. Rossi, and E. Oñate, "An Implicit Material Point Method Applied to Granular Flows," *Procedia Engng.*, vol. 175, pp. 226–232, 2017.

[73] D. Sulsky, S. J. Zhou, and H. L. Schreyer, "Application of a particle-in-cell method to solid mechanics," *Comput. Physics Communic.*, vol. 87, no. 1-2, pp. 236–252, 1995.

[74] Z. Wieckowski, "The material point method in large strain engineering problems," *Comput. Meth. Appl. Mech. Engng.*, vol. 193, pp. 4417–4438, 2004.

[75] B. Ramaswamy, "Lagrangian finite element method for the analysis of two dimensional sloshing problems," *Int. J. Numer. Meth. Fluids*, vol. 6, no. 9, pp. 659–670, 1986.

[76] M. Kawahara and A. Anjyu, "Lagrangian finite element method for solitary wave propagation," *Comp. Mech.*, vol. 3, no. 1, pp. 299–307, 1988.

[77] R. Radovitzky and M. Ortiz, "Lagrangian Finite Element Analysis of Newtonian Fluid Flow," *Int. J. Numer. Meth. Engng.*, vol. 43, pp. 607–619, 1998.

[78] E. Oñate, S. Idelsohn, F. Del Pin, and R. Aubry, "The particle finite element method. An overview.," *Int. J. Comp. Meth.*, vol. 1, no. 2, pp. 267–307, 2004.

[79] S. Idelsohn, N. Nigro, A. Limache, and E. Oñate, "Large time-step explicit integration method for solving problems with dominant convection," *Comput. Meth. Appl. Mech. Engng.*, vol. 217-220, pp. 168–185, 2012.

[80] S. Idelsohn, N. Nigro, J. Gimenez, R. Rossi, and J. Marti, "A fast and accurate method to solve the incompressible Navier-Stokes equations," *Engng. Computations*, vol. 30, no. 2, pp. 197–222, 2013.

[81] P. Becker, S. Idelsohn, and E. Oñate, "A unified monolithic approach for multi-fluid flows and fluid-structure interaction using the Particle Finite Element Method with fixed mesh," *Comp. Mech.*, 2014.

[82] R. Benzi, S. Succi, and M. Vergassola, "The lattice Boltzmann equation: theory and applications," *Physics Reports*, vol. 222, no. 3, pp. 145–197, 1992.

[83] U. R. Frisch, D. D'humieres, B. Hasslacher, P. Lallemand, Y. Ves Pomeau, and J. P. R. Ivet, "Lattice Gas Hydrodynamics in Two a nd Three Dimensions," *Complex Systems*, vol. 1, pp. 649–707, 1987.

[84] X. He and L. S. Luo, "Lattice Boltzmann Model for the Incompressible Navier-Stokes Equation," *J. Statist. Physics*, vol. 88, no. 3/4, pp. 927–944, 1997.

[85] C. Janssen and M. Krafczyk, "Free surface flow simulations on GPGPUs using the LBM," *Comput. Math. Applications*, vol. 61, no. 12, pp. 3549–3563, 2011.

[86] I. Ginzburg and K. Steiner, "Lattice Boltzmann model for free-surface flow and its application to filling process in casting," *J. Comp. Physics*, vol. 185, no. 1, pp. 61–99, 2003.

[87] J. Tolke and M. Krafczyk, "TeraFLOP computing on a desktop PC with GPUs for 3D CFD," *Int. J. Comp. Fluid Dyn.*, vol. 22, no. 7, pp. 443–456, 2008.

[88] S. Idelsohn, E. Oñate, and F. Del Pin, "The particle finite element method: A powerful tool to solve incompressible flows with free-surfaces and breaking waves," *Int. J. Numer. Meth. Engng.*, vol. 61, no. 7, pp. 964–989, 2004.

[89] S. Idelsohn, E. Oñate, F. Del Pin, and N. Calvo, "Fluid-structure interaction using the particle finite element method," *Comput. Meth. Appl. Mech. Engng.*, vol. 195, pp. 2100–2123, mar 2006.

[90] M. Zhu and M. H. Scott, "Direct Differentiation of the Particle Finite-Element Method for Fluid-Structure Interaction," *J. Struct. Eng.*, vol. 142, no. 3, pp. 1–14, 2016.

[91] M. L. Cerquaglia, G. Deliége, R. Boman, L. Papeleux, and J. P. Ponthot, "The Particle Finite Element Method for the numerical simulation of bird strike," *Int. J. of Impact Engng.*, vol. 109, pp. 1–13, 2017.

[92] J. Cante, C. Davalos, J. A. Hernandez, J. Oliver, P. Jonsen, G. Gustafsson, and H. Haggblad, "PFEM-based modeling of industrial granular flows," *Comp. Particle Mech.*, vol. 1 (1), pp. 47–70, 2014.

[93] X. Zhang, K. Krabbenhoft, D. M. Pedroso, A. V. Lyamin, D. Sheng, M. Vicente, and D. Wang, "Computers and Geotechnics Particle finite element analysis of large deformation and granular flow problems," *Comput. Geotech.*, vol. 54, pp. 133–142, 2013.

[94] M. Cremonesi, F. Ferri, and U. Perego, "A basal slip model for Lagrangian finite element simulations of 3D landslides Massimiliano," *Int. J. Numer. Analyt. Meth. Geomech.*, vol. 41, no. 1, pp. 30–53, 2016.

[95] S. Idelsohn, F. Del Pin, R. Rossi, and E. Oñate, "Fluid-structure interaction problems with strong added-mass effect," *Int. J. Numer. Meth. Engng.*, vol. 80, pp. 1261–1294, 2009.

[96] E. Oñate, A. Franci, and J. M. Carbonell, "A particle finite element method for analysis of industrial forming processes," *Comp. Mech.*, vol. 54, no. 1, pp. 85–107, 2014.

[97] R. Aubry, S. Idelsohn, and E. Oñate, "Particle finite element method in fluid-mechanics including thermal convection-diffusion," *Comp. Struct.*, vol. 83, no. 17-18, pp. 1459–1475, 2005.

[98] E. Oñate, A. Franci, and J. M. Carbonell, "A Particle Finite Element Method (PFEM) for Coupled Thermal Analysis of Quasi and Fully Incompressible Flows and Fluid-Structure Interaction Problems," in *Num. Sim. Coupled Problems Engng.* (S. R. Idelsohn, ed.), pp. 129–156, Springer International Publishing, 2014.

[99] E. Oñate, M. A. Celigueta, S. Idelsohn, F. Salazar, and B. Suárez, "Possibilities of the particle finite element method for fluid-soil-structure interaction problems," *Comp. Mech.*, vol. 48, no. 3, pp. 307–318, 2011.

[100] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Transactions Graphics*, vol. 13, no. 1, pp. 43–72, 1994.

[101] S. Idelsohn, N. Calvo, and E. Oñate, "Polyhedrization of an arbitrary 3D point set," *Comput. Meth. Appl. Mech. Engng.*, vol. 192, no. 22-23, pp. 2649–2667, 2003.

[102] M. L. Cerquaglia, G. Deliége, R. Boman, L. Papeleux, V. Terrapon, and J. P. Ponthot, "On some drawbacks and possible improvements of a Lagrangian finite element approach for simulating incompressible flows," in *4th Int. Conf. Particle-Based Meth. PARTICLES 2015*, pp. 1081–1092, 2015.

[103] A. Franci and M. Cremonesi, "On the effect of standard PFEM remeshing on volume conservation in free-surface fluid flow problems," *Comp. Particle Mech.*, vol. 4, no. 3, pp. 331–343, 2017.

[104] L. A. Freitag, "On Combining Laplacian And Optimization-Based Mesh Smoothing Techniques," *Trends Unstruct. Mesh Generation*, pp. 37–43, 1997.

[105] B. M. Klingner, *Improving Tetrahedral Meshes*. PhD thesis, University of California, Berkeley, 2008.

[106] D. Vartziotis, J. Wipper, and B. Schwald, "The geometric element transformation method for tetrahedral mesh smoothing," *Comput. Meth. Appl. Mech. Engng.*, vol. 199, no. 1-4, pp. 169–182, 2009.

[107] D. Pelletier, A. Fortin, and R. Camarero, "Are FEM solutions of incompressible flows really incompressible?(or how simple flows can

cause headaches!)," *Int. J. Numer. Meth. Fluids*, vol. 9, no. 1, pp. 99–112, 1989.

[108] G. X. Wu and R. Eatock Taylor, "Finite element analysis of two-dimensional non-linear transient water waves," *Applied Ocean Research*, vol. 16, no. 6, pp. 363–372, 1994.

[109] S. Koshizuka, H. Tamako, and Y. Oka, "A particle method for incompressible viscous flow with fluid fragmentation," *J. Comp. Fluid Dyn.*, vol. 4, pp. 29–46, 1995.

[110] S. Idelsohn, J. Marti, A. Limache, and E. Oñate, "Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluid-structure interaction problems via the PFEM," *Comput. Meth. Appl. Mech. Engng.*, vol. 197, no. 19-20, pp. 1762–1776, 2008.

[111] D. M. Greaves, "Simulation of viscous water column collapse using adaptive hierarchical grids," *Int. J. Numer. Meth. Fluids*, vol. 50, pp. 693–711, 2006.

[112] C. Farhat, A. Rallu, K. Wang, and T. Belytschko, "Robust and provably second-order explicit-explicit and implicit-explicit staggered time-integrators for highly non-linear compressible fluid-structure interaction problems," *Int. J. Numer. Meth. Engng.*, vol. 84, pp. 73–107, 2010.

[113] P. Causin, J. F. Gerbeau, and F. Nobile, "Added-mass effect in the design of partitioned algorithms for fluid-structure problems," *Comput. Meth. Appl. Mech. Engng.*, vol. 194, no. 42-44, pp. 4506–4527, 2005.

[114] M. A. Fernández, "Coupling schemes for incompressible fluid-structure interaction: implicit, semi-implicit and explicit," *SeMA J.*, vol. 55, pp. 59–108, 2011.

[115] C. Minchler, S. J. Hulshoff, E. H. van Brummelen, and R. de Borst, "A monolithic approach to fluid-structure interaction," *Comput. Fluids*, vol. 33, no. 5-6, pp. 839–848, 2004.

[116] C. Forster, W. A. Wall, and E. Ramm, "Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows," *Comput. Meth. Appl. Mech. Engng.*, vol. 196, no. 7, pp. 1278–1293, 2007.

[117] C. A. Felippa, K. C. Park, and C. Farhat, "Partitioned analysis of coupled mechanical systems," *Meth. Appl. Mech. Engrg.*, vol. 190, no. 24-25, pp. 3247–3270, 2001.

[118] Z. Li, J. Leduc, A. Combescure, and F. Leboeuf, "Coupling of SPH-ALE method and finite element method for transient fluid-structure interaction," *Comput. Fluids*, vol. 103, pp. 6–17, 2014.

[119] S. Deparis, M. Discacciati, G. Fourestey, and A. Quarteroni, "Fluid-structure algorithms based on Steklov-Poincaré operators," *Comput. Meth. Appl. Mech. Engng.*, vol. 195, no. 41-43, pp. 5797–5812, 2006.

[120] E. Burman and M. A. Fernández, "Stabilization of explicit coupling in fluid-structure interaction involving fluid incompressibility," *Comput. Meth. Appl. Mech. Engng.*, vol. 198, no. 5-8, pp. 766–784, 2009.

[121] S. Potapov, B. Maurel, A. Combescure, and J. Fabis, "Modeling accidental-type fluid-structure interaction problems with the SPH method," *Comp. Struct.*, vol. 87, no. 11-12, pp. 721–734, 2009.

[122] A. Franci, E. Oñate, and J. M. Carbonell, "Unified Lagrangian formulation for solid and fluid mechanics and FSI problems," *Comput. Meth. Appl. Mech. Engng.*, vol. 298, pp. 520–547, 2016.

[123] S. Piperno, C. Farhat, and B. Larrouturou, "Partitioned procedures for the transient solution of coupled aroelastic problems Part I: Model problem, theory and two-dimensional application," *Comput. Meth. Appl. Mech. Engng.*, vol. 124, no. 1-2, pp. 79–112, 1995.

[124] Z. Li, J. Leduc, J. Nunez-Ramirez, A. Combescure, and J. C. Marongiu, "A non-intrusive partitioned approach to couple smoothed particle hydrodynamics and finite element methods for transient fluid-structure interaction problems with large interface motion," *Comp. Mech.*, vol. 55, no. 4, pp. 697–718, 2015.

[125] T. Dunne, "An Eulerian approach to fluid-structure interaction and goal-oriented mesh adaptation," *Int. J. Numer. Meth. Fluids*, vol. 51, no. 9, pp. 1017–1039, 2000.

[126] T. Richter, "A fully eulerian formulation for fluid-structure interaction problems," *J. Comp. Physics*, vol. 233, no. 1, pp. 227–240, 2013.

[127] C. Antoci, M. Gallati, and S. Sibilla, "Numerical simulation of fluid-structure interaction by SPH," *Comput. Struct.*, vol. 85, no. 11-14, pp. 879–890, 2007.

[128] W. F. Noh, "CEL: a time dependent, two space dimensional, coupled Eulerian-Lagrangian code," *Meth. Comp. Physics*, vol. 157, no. 1-2, pp. 95–114, 1964.

[129] D. J. Benson, "Computational methods in lagrangian and eulerian hydrocodes," *Comput. Meth. Appl. Mech. Engng.*, vol. 99, no. 2-3, pp. 235–394, 1992.

[130] C. S. Peskin, "The immersed boundary method," *Acta Numerica*, vol. 11, pp. 479–517, 2002.

[131] R. Mittal and G. Iaccarino, "Immersed Boundary Methods," *Annu. Rev. Fluid Mech.*, vol. 37, no. 1, pp. 239–261, 2005.

[132] T. Belytschko, D. P. Flanagan, and J. M. Kennedy, "Finite element methods with user-controlled meshes for fluid-structure interaction," *Comput. Meth. Appl. Mech. Engng.*, vol. 33, no. 1-3, pp. 669–688, 1982.

[133] J. Donea, S. Giuliani, and J. P. Halleux, "An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interaction," *Comput. Meth. Appl. Mech. Engng.*, vol. 33, pp. 689–723, 1982.

[134] T. Richter and T. Wick, "Finite elements for fluid-structure interaction in ALE and fully Eulerian coordinates," *Comput. Meth. Appl. Mech. Engng.*, vol. 199, no. 41-44, pp. 2633–2642, 2010.

[135] A. Combescure and A. Gravouil, "A time-space multi-scale algorithm for transient structural nonlinear problems," *Mecanique et Industries*, vol. 2, no. 1, pp. 43–55, 2001.

[136] C. Farhat and F. X. Roux, "A method of finite element tearing and interconnecting and its parallel solution algorithm," *Int. J. Numer. Meth. Engng.*, vol. 32, no. 6, pp. 1205–1227, 1991.

[137] Dassault Systèmes SIMULIA, *Abaqus Documentation*. 2016.

[138] A. Gravouil and A. Combescure, "Multi-time-step explicit-implicit method for non-linear structural dynamics," *Int. J. Numer. Meth. Engng.*, no. July 1999, pp. 199–225, 2001.

[139] J. Nunez-Ramirez, J. C. Marongiu, M. Brun, and A. Combescure, "A partitioned approach for the coupling of SPH and FE methods for transient non-linear FSI problems with incompatible time-steps," *Int. J. Numer. Meth. Engng.*, vol. 109, no. 10, pp. 1391–1417, 2016.

[140] V. Faucher and A. Combescure, "A time and space mortar method for coupling linear modal subdomains and non-linear subdomains in explicit structural dynamics," *Comput. Meth. Appl. Mech. Engng.*, vol. 192, no. 5-6, pp. 509–533, 2003.

[141] A. Combescure, A. Gravouil, and B. Herry, "An algorithm to solve transient structural non-linear problems for non-matching time-space domains," *Comp. Struct.*, vol. 81, no. 12, pp. 1211–1222, 2003.

[142] E. Walhorn, B. Hübner, A. Kölke, and D. Dinkler, "Fluid-structure coupling within a monolithic model involving free surface flows," *Comp. Struct.*, vol. 83, pp. 2100–2111, 2005.

[143] M. Cremonesi, *A Lagrangian finite element method for the interaction between flexible structures and free surfaces fluid flows.* Phd thesis, Politecnico di Milano, 2010.

[144] O. C. Zienkiewicz, R. L. Taylor, and P. Nithiarasu, *The Finite Element Method for fluid dynamics.* Oxford: Elsevier Butterworth-Heinemann, 6th ed., 2005.

[145] S. Richardson, "On the no-slip boundary condition," *J. Fluid Mech.*, vol. 59, no. 4, pp. 707–719, 1973.

[146] D. Y. C. Chan and R. G. Horn, "The drainage of thin liquid films between solid surfaces," *J. Chem. Physics*, vol. 83, no. 10, pp. 5311–5324, 1985.

[147] P. Thompson and S. M. Troian, "A general boundary condition for liquid flow at solid surfaces," *Nature*, vol. 389, no. September, pp. 360–362, 1997.

[148] W. R. Schowalter, "The behavior of complex fluids at solid boundaries," *J. Non-Newtonian Fluid Mech.*, vol. 29, no. C, pp. 25–36, 1988.

[149] M. M. Denn, "Extrusion instabilities and wall slip," *Annu. Rev. Fluid Mech.*, vol. 33, pp. 265–287, 2001.

[150] L. Léger, H. Hervet, G. Massey, and E. Durliat, "Wall slip in polymer melts," *J. Physics Condens. Matter*, vol. 9, no. 37, pp. 7719–7740, 1997.

[151] S. G. Hatzikiriakos, "Wall slip of molten polymers," *Progress Polymer Sci.*, vol. 37, no. 4, pp. 624–643, 2012.

[152] M. Schaefer, L. Bugnion, M. Kern, and P. Bartelt, "Position dependent velocity profiles in granular avalanches," *Granular Matter*, vol. 12, no. 3, pp. 327–336, 2010.

[153] A. Lucas, A. Mangeney, and J. P. Ampuero, "Frictional velocity-weakening in landslides on Earth and on other planetary bodies," *Nature Communic.*, vol. 5, pp. 1–9, 2014.

[154] J. Choi, A. Kudrolli, and M. Z. Bazant, "Velocity profile of granular flows inside silos and hoppers," *J. Physics Condens. Matter*, vol. 17, no. 24, pp. 2533–2548, 2005.

[155] J. Étienne, E. J. Hinch, and J. Li, "A Lagrangian-Eulerian approach for the numerical simulation of free-surface flow of a viscoelastic material," *J. Non-Newtonian Fluid Mech.*, vol. 136, no. 2, pp. 157–166, 2006.

[156] J. Basic, N. Degiuli, and A. Werner, "Simulation of water entry and exit of a circular cylinder using the isph method," *Trans. Famena*, vol. 38, no. 1, pp. 45–62, 2014.

[157] M. Lastiwka, M. Basa, and N. J. Quinlan, "Permeable and non-reflecting boundary conditions in SPH," *Int. J. Numer. Meth. Fluids*, vol. 61, no. 7, pp. 709–724, 2009.

[158] I. Federico, S. Marrone, A. Colagrossi, F. Aristodemo, and M. Antuono, "Simulating 2D open-channel flows through an SPH model," *European J. Mech. Fluids*, vol. 34, no. Supplement C, pp. 35–46, 2012.

[159] S. K. Tan, N. S. Cheng, Y. Xie, and S. Shao, "Incompressible SPH simulation of open channel flow over smooth bed," *J. Hydro-Environment Research*, vol. 9, no. 3, pp. 340–353, 2015.

[160] M. Bolognin, M. Martinelli, K. J. Bakker, and S. N. Jonkman, "Validation of material point method for soil fluidisation analysis," *Procedia Engng.*, vol. 175, pp. 233–241, 2017.

[161] X. Zhao, M. Bolognin, D. Liang, A. Rohe, and P. J. Vardon, "Development of in/outflow boundary conditions for MPM simulation of uniform and non-uniform open channel flows," *Comput. Fluids*, vol. 179, pp. 27–33, 2019.

[162] F. Del Pin, S. Idelsohn, E. Oñate, and R. Aubry, "The ALE/Lagrangian Particle Finite Element Method: A new approach

to computation of free-surface flows and fluid-object interactions," *Comput. Fluids*, vol. 36, no. 1, pp. 27–38, 2007.

[163] L. L. Ferrás, J. M. Nóbrega, and F. T. Pinho, "Analytical solutions for Newtonian and inelastic non-Newtonian flows with wall slip," *J. Non-Newtonian Fluid Mech.*, vol. 175-176, pp. 76–88, 2012.

[164] R. Verfurth, "Finite Element Approximation of Incompressible Navier-Stokes Equations with Slip Boundary Condition," *Numerische Mathematik*, vol. 721, pp. 697–721, 1987.

[165] I. Dione, C. Tibirna, and J. Urquiza, "Stokes equations with penalised slip boundary conditions," *Int. J. Comp. Fluid Dyn.*, vol. 27, no. 6-7, pp. 283–296, 2013.

[166] P. Ryzhakov, E. Oñate, R. Rossi, and S. Idelsohn, "Improving mass conservation in simulation of incompressible flows," *Int. J. Numer. Meth. Engng.*, vol. 90, pp. 1435–1451, mar 2012.

[167] T. Wick, "Fluid-structure interactions using different mesh motion techniques," *Comp. Struct.*, vol. 89, no. 13-14, pp. 1456–1467, 2011.

[168] F. Tian, H. Dai, H. Luo, J. F. Doyle, and B. Rousseau, "Fluid-structure interaction involving large deformations : 3D simulations and applications to biological systems," *J. Comp. Physics*, vol. 258, pp. 451–469, 2014.

[169] S. Idelsohn, E. Oñate, N. Nigro, P. Becker, and J. Gimenez, "Lagrangian versus Eulerian integration errors," *Comput. Meth. Appl. Mech. Engng.*, vol. 293, no. May 2015, pp. 191–206, 2015.

[170] S. W. Cheng, T. K. Dey, and J. R. Shewchuk, *Delaunay Mesh Generation*. CRC Press, 2013.

[171] I. Fried, "Condition of finite element matrices generated from nonuniform meshes.," *AIAA Journal*, vol. 10, no. 2, pp. 219–221, 1972.

[172] P. Clausen, M. Wicke, J. R. Shewchuk, and J. F. O'brien, "Simulating liquids and solid-liquid interactions with lagrangian meshes," *ACM Transactions Graphics*, vol. 32, no. 2, pp. 1–15, 2013.

[173] M. Wicke, D. Ritchie, B. M. Klingner, S. Burke, J. R. Shewchuk, and J. F. O'Brien, "Dynamic local remeshing for elastoplastic simulation," *ACM Transactions Graphics*, vol. 29, no. 4, p. 1, 2010.

[174] L. A. Freitag and C. Ollivier-Gooch, "Tetrahedral mesh improvement using swapping and smoothing," *Int. J. Numer. Meth. Engng.*, vol. 40, no. 21, pp. 3979–4002, 1997.

[175] B. M. Klingner and J. R. Shewchuk, "Agressive Tetrahedral Mesh Improvement," *Proc. 16th Int. Mesh. Roundtable*, pp. 3–23, 2007.

[176] H. Edelsbrunner and N. R. Shah, "Incremental Topological Flipping Works for Regular Triangulations," *Algorithmica*, vol. 15, no. 3, pp. 223–241, 1996.

[177] D. A. Field, "Laplacian smoothing and Delaunay triangulation," *Int. J. Numer. Meth. Biomed. Eng.*, vol. 4, no. 6, pp. 709–712, 1988.

[178] S. A. Canann, M. B. Stephenson, and T. Blacker, "Optismoothing: An optimization-driven approach to mesh smoothing," *Finite Elements Analysis Design*, vol. 13, no. 2-3, pp. 185–190, 1993.

[179] L. A. Freitag, P. M. Knupp, T. Munson, and S. Shontz, "A comparison of two optimization methods for mesh quality improvement," *Engng. Comput.*, vol. 22, no. 2, pp. 61–74, 2006.

[180] J. T. Batina, "Using Unstructured Dynamic Meshes," *AIAA Journal*, vol. 28, no. 8, pp. 1381–1388, 1990.

[181] F. J. Blom, "Considerations on the spring analogy," *Int. J. Numer. Meth. Fluids*, vol. 32, no. 6, pp. 647–668, 2000.

[182] C. L. Bottasso, D. Detomi, and R. Serra, "The ball-vertex method: A new simple spring analogy method for unstructured dynamic meshes," *Comput. Meth. Appl. Mech. Engng.*, vol. 194, no. 39-41, pp. 4244–4246, 2005.

[183] C. Farhat, M. Lesoinne, and P. Le Tallec, "Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservations, optimal discretization and application to aeroelasticity," *Comput. Meth. Appl. Mech. Engng.*, vol. 157, no. 1-2, pp. 95–114, 1998.

[184] S. L. Karman, W. K. Anderson, and M. Sahasrabudhe, "Mesh Generation Using Unstructured Computational Meshes and Elliptic Partial Differential Equation Smoothing," *AIAA Journal*, vol. 44, no. 6, pp. 1277–1286, 2006.

[185] Z. Yang and D. Mavriplis, "Unstructured Dynamic Meshes with Higher-order Time Integration Schemes for the Unsteady Navier-Stokes Equations," *43rd AIAA Aerospace Sci. Meeting and Exhibit*, vol. 1, no. January, 2005.

[186] E. J. Nielsen and W. K. Anderson, "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, vol. 40, no. 6, pp. 1155–1163, 2002.

[187] A. Johnson and T. Tezduyar, "Simulation of multiple spheres falling in a liquid-filled tube," *Comput. Meth. Appl. Mech. Engng.*, vol. 134, no. 3-4, pp. 351–373, 1996.

[188] J. R. H. Otter and A. S. Day, "Tidal flow computations," *Engineer*, vol. 209, pp. 177–182, 1960.

[189] D. R. Oakley and N. F. Knight, "Adaptive dynamic relaxation algorithm for non-linear hyperelastic structures Part I. Formulation," *Comput. Meth. Appl. Mech. Engng.*, vol. 126, pp. 91–109, 1995.

[190] D. R. Oakley and N. F. Knight, "Adaptive Dynamic Relaxation algorithm for non-linear hyperelastic structures Part II. Single-processor implementation," *Comput. Meth. Appl. Mech. Engng.*, vol. 126, pp. 67–89, 1995.

[191] P. Underwood, "Dynamic Relaxation," in *Comput. Meth. Transient Analysis*, ch. 5, pp. 245–265, North Holland, 1983.

[192] J. Rombouts, G. Lombaert, L. de Laet, and M. Schevenels, "On the equivalence of dynamic relaxation and the Newton Raphson method: application to the design and analysis of bending-active structures," *Proc. IASS Annual Symposium*, pp. 1–10, 2017.

[193] J. Schöberl, "An advancing front 2D/3D-mesh generator based on abstract rules," *Computing Visualization Sci.*, vol. 1, no. 1, pp. 41–52, 1997.

[194] C. Geuzaine and J. F. Remacle, "Gmsh: A 3D finite element mesh generator with built in pre and post processing facilities," *Int. J. Numer. Meth. Engng.*, vol. 79, pp. 1309–1331, 2009.

[195] J. C. Martin and W. J. Moyce, "Part IV. An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane," *Ph. Trans. Roy. Soc. London-Math. Physical Sci.*, vol. 244, no. 882, pp. 312–324, 1952.

[196] C. Hu and M. Sueyoshi, "Numerical simulation and experiment on dam break problem," *J. Marine Sci. Appl.*, vol. 9, no. 2, pp. 109–114, 2010.

[197] F. R. Saucedo-Zendejo and E. O. Resendiz-Flores, "A new approach for the numerical simulation of free surface incompressible flows using a meshfree method," *Comput. Meth. Appl. Mech. Engng.*, vol. 324, pp. 619–639, 2017.

[198] K. M. T. Kleefsman, G. Fekken, A. E. P. Veldman, B. Iwanowski, and B. Buchner, "A Volume-of-Fluid based simulation method for wave impact problems," *J. Comp. Physics*, vol. 206, no. 1, pp. 363–393, 2005.

[199] R. Nayak, R. Padhye, K. Sinnappoo, L. Arnold, and B. K. Behera, "Airbags," *Textile Progress*, vol. 45, no. 4, pp. 209–301, 2013.

[200] A. Hirth, A. Haufe, and L. Olovsson, "Airbag Simulation with LS-DYNA: Past - Present - Future," in *LS-DYNA User Meeting*, (Frankenthal, Germany), pp. 51–74, 2007.

[201] J. T. Wang and D. J. Nefske, "A new CAL3D airbag inflation model," in *SAE Tech. Paper 880650*, (Detroit, USA), pp. 1–10, SAE Transactions, 1988.

[202] B. Hussey and L. Rink, "Predictive techniques for airbag inflator exit properties," in *Int. Congress and Exposition*, (Detroit, USA), pp. 67–76, 1995.

[203] C. Ruff, T. Jost, and A. Eichberger, "Simulation of an airbag deployment in out-of-position situations," *Int. J. Vehicle Mech. Mobility*, vol. 45, no. 10, pp. 953–967, 2007.

[204] J. Masiá, B. Eixerés, J. F. Dols, and F. J. Colomina, "Influence of steering control devices mounted in cars for the disabled on passive safety," in *3rd Manufacturing Engng. Society Int. Conf.*, vol. 1181, pp. 611–621, 2009.

[205] X. Wang, Z. Zong, Y. Zhao, and L. Zou, "A numerical study of passenger side airbag deployment based on arbitrary Lagrangian-eulerian method," *Sci. China*, vol. 58, no. 3, pp. 397–404, 2015.

[206] D. Fokin, N. Lokhande, and L. Fredriksson, "On Airbag Simulation in LS-DYNA with the use of the Arbitrary Lagrangian-Eulerian

Method," in *4th European LS-DYNA Users Conference*, pp. 11–22, 2003.

[207] P. O. Marklund and L. Nilsson, "Optimization of airbag inflation parameters for the minimization of out of position occupant injury," *Comp. Mech.*, vol. 31, no. 6, pp. 496–504, 2003.