Particle methods in Computational Fluid Dynamics

Sergio Idelsohn
Centre Internacional de Mètodes Numèrics en Enginyeria - CIMNE
Institució Catalana de Recerca i Estudis Avançats - ICREA

Eugenio Oñate
Centre Internacional de Mètodes Numèrics en Enginyeria - CIMNE
Universitat Politécnica de Catalunya - UPC

Pablo Becker
Centre Internacional de Mètodes Numèrics en Enginyeria - CIMNE

January 18, 2017

1 Introduction

Particle methods in Computational Fluid Dynamics (CFD) are numerical tools for the solution of the equations of fluid dynamics obtained by replacing the fluid continuum with a finite set of particles. For mathematicians, particles are just points from which properties of the fluid can be interpolated. For physicists the particles are material points, which can be treated like any other particle system. Either way, particle methods have a number of attractive features. One of the key attributes is that pure advection is treated exactly. For example, if the particles are given a determined colour and the velocity is specified, the transport of colours by the particle system is exact. The convection of properties also eases the solution of multi material problems, simplifying the detection of interfaces. The use of particles also allows to bridge the gap between the continuum and fragmentation in a natural way, for example in fracture or droplets problems. Since the computation domain, the particles, matches exactly the material domain of interest, the computational resources are optimized with the corresponding reduction in storage and calculation time compared to other methods. Finally, because of the close similarity between particle methods and the physics of the problems to be solved, it is often possible to account for complex physics more easily than with other methods.

Particle methods can be roughly classified into two types: a) those based on probabilistic models, such as molecular dynamics, direct simulation Monte Carlo, and lattice gas automata procedures, and b) those based on deterministic models, such as the particle-and-force method, smooth particle hydrodynamics (SPH), particle-in-cell method (PIC), moving-particle semi-implicit method (MPS), material point method (MPM), particle finite element method (PFEM) and discrete element method (DEM), among others.

Probabilistic methods represent macroscopic properties as statistical behaviours of microscopic particles, and therefore a large number of particles should be traced for a long period to obtain accurate average values. Deterministic methods, on the other hand, require much shorter computation times and smaller storage. Only methods based on deterministic models will be presented in this chapter.

Deterministic particle methods can be subdivided in particle methods that use a mesh (or a grid) and particle methods without a mesh, named meshless methods. When a mesh is used, particles are only used to trace interfaces or to calculate the convection terms adding no numerical diffusion. On the other hand, in particle methods without a mesh, the particles are also used to evaluate the spatial derivatives of the unknown functions.

It must be noted that the term *meshless methods* is not equivalent to *particle methods*. The word meshless refers to methods in which a mesh is not necessary to evaluate the different field variables or their derivatives.

A meshless method may be used to solve a CFD problem with or without particles. In the same way, a particle method may be used with or without a mesh. Standard particle methods that do not use a mesh are SPH and MPS. On the other hand, typical particle methods that use a background mesh are MPM, PIC and PFEM.

Finally, another class of deterministic method is the DEM. Although it is a method originally designed for the study of solid particle interactions, the DEM has also been used to approximate fluids and also, in conjunction with standard CFD formulations, to model particulate fluids. In the DEM each particle has an associated volume, mass and shape and the interaction is defined the by physical contacts between the particles.

The layout of this chapter is as follows. First we present the governing equations of fluid dynamics to be solved. Then we describe the basic concepts of the SPH, MPS and MPM procedures. Next we explain in more detail the theory and computational implementation of the PFEM, both in its standard form as well as the so-called PFEM-2, recently developed by the authors. The chapter concludes with an overview of the DEM and the coupling of the DEM with the finite element method (FEM) for the analysis of particulate flows and their interactions with structures.

2 The equations to be solved

While particle methods have been used for both, compressible and incompressible fluid flows and also for both laminar and turbulent flows, in this chapter we will refer only to laminar and incompressible or quasi-incompressible fluid flows.

The momentum and mass conservation equations that define the flow of a fluid in a domain Ω read:

$$\text{Momentum: } \rho\frac{D\mathbf{V}}{Dt} = \nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{b} \quad \text{in } \Omega \tag{1a}$$

$$\text{Mass conservation: } \frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{V}) = 0 \quad \text{in } \Omega \tag{1b}$$

where $\mathbf{V}$ is the velocity vector, $\frac{DA}{Dt}$ is the material time derivative of a function A or of the component of a vector, $\boldsymbol{\sigma}$ the Cauchy stress tensor, ρ the density and $\mathbf{b}$ a source vector term (typically, the body force vector). The material time derivative is expressed in different form in Lagrangian and Eulerian frames, as explained below.

For a Newtonian fluid, the stress tensor is split as:

$$\boldsymbol{\sigma} = \boldsymbol{\tau} - p\mathbf{I} = 2\mu\boldsymbol{\epsilon}^S - p\mathbf{I} \tag{2}$$

where μ is the dynamic viscosity coefficient, $\boldsymbol{\tau}$ is the deviatoric stress tensor, p the pressure (positive in compression), $\mathbf{I}$ the second order unit tensor and $\boldsymbol{\epsilon}^S$ the symmetric strain tensor: $\epsilon^S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right)$.

Inserting Eq. (2) into Eq. (1a) and taking into account the incompressibility condition, $\frac{D\rho}{Dt} = 0$, the Navier-Stokes equations read:

$$\begin{cases} \rho\dfrac{D\mathbf{V}}{Dt} = \nabla\cdot\boldsymbol{\tau} - \nabla p + \mathbf{b} \quad \text{in } \Omega \\ \nabla\cdot\boldsymbol{V} = 0 \quad \text{in } \Omega \end{cases} \tag{3}$$

with appropriate boundary conditions:

$$\mathbf{V} = \bar{\mathbf{V}} \text{ on } \Gamma_V \quad \text{and} \quad \bar{\boldsymbol{\sigma}}\cdot\mathbf{n} = (2\mu\boldsymbol{\epsilon}^S - p\mathbf{I})\cdot\mathbf{n} \text{ on } \Gamma_{\boldsymbol{\sigma}} \tag{4}$$

where $\bar{\mathbf{V}}$ and $\bar{\boldsymbol{\sigma}}$ represent known values of the velocity and the surface stresses on the Dirichlet (Γ_V) and Neumann (Γ_σ) boundaries with $\Gamma = \Gamma_V \bigcup \Gamma_r$ being the boundary of Ω and $\mathbf{n}$ the unit normal vector to the boundary.

2.1 Lagrangian and Eulerian formulations

To solve numerically Eqs.(3) we have to approximate the unknowns in space and time and integrate the equations over the total space-time domain. Concerning the coordinate frame where the governing equations are solved,

the solution methods can be classified in Eulerian and Lagrangian formulations. Classical numerical methods to solve CFD problems typically use the Eulerian formulation. However, particle methods use in general a Lagrangian formulation.

The main difference between the two formulations lies in the way the time derivative is described. In an Eulerian formulation the spatial or partial derivative is required. On the other hand, in a Lagrangian frame, the total or material time derivative is used and, therefore, the form of Eq. (3) can be used without further modifications:

Lagrangian frame:

$$\rho \frac{D\mathbf{V}}{Dt} = \nabla \cdot \boldsymbol{\tau} - \nabla p + \rho \mathbf{b} \tag{5a}$$

$$\nabla \cdot \boldsymbol{V} = 0 \tag{5b}$$

Note that for the definition of the total derivative:

$$\frac{D\mathbf{V}}{Dt} = \lim_{\Delta t \to 0} \frac{\mathbf{V}\left(\mathbf{x}_p^{n+1}, t^{n+1}\right) - \mathbf{V}\left(\mathbf{x}_p^{n}, t^{n}\right)}{\Delta t} \tag{6}$$

the position of a particle $\mathbf{x}_p^{n+1}$ after the time integration is known. For this reason it is necessary, in case of a Lagrangian frame, to add the movement of the particle to the equation. Hence, Eqs.(5) are rewritten as

$$\rho \frac{D\mathbf{V}(\mathbf{x}_p)}{Dt} = \nabla \cdot \boldsymbol{\tau}(\mathbf{x}_p) - \nabla p(\mathbf{x}_p) + \rho \mathbf{b}(\mathbf{x}_p) \tag{7a}$$

$$\nabla \cdot \boldsymbol{V}(\mathbf{x}_p) = 0 \tag{7b}$$

$$\text{with } \mathbf{V}(\mathbf{x}_p) = \frac{D\mathbf{x}_p}{Dt} \tag{7c}$$

where the unknowns are $\mathbf{V}(\mathbf{x}_p, t)$, $p(\mathbf{x}_p, t)$ and $\mathbf{x}_p(t)$.

Eulerian frame: For the Eulerian frame, Eq. (6) yields $\frac{D\mathbf{V}}{Dt} = \frac{\partial \mathbf{V}}{\partial t} + \mathbf{V}(x,t)\boldsymbol{\nabla}\mathbf{V}(x,t)$ where $\frac{\partial \mathbf{V}}{\partial t}$ is the spatial time derivative. Substituting this expression into Eq. (5a) gives the usual form for the governing equations for a fluid in an Eulerian frame as

$$\rho \left[\frac{\partial \mathbf{V}(\mathbf{x}, t)}{\partial t} + \mathbf{V}(\mathbf{x}, t) \nabla \mathbf{V}(\mathbf{x}, t) \right] = \nabla \cdot \boldsymbol{\tau}(\mathbf{x}, t) - \nabla p(\mathbf{x}, t) + \rho \mathbf{b}(\mathbf{x}, t) \tag{8a}$$

$$\nabla \cdot \boldsymbol{V}(\mathbf{x}, t) = 0 \tag{8b}$$

and the unknowns are now only $\mathbf{V}(\mathbf{x}, t)$ and $p(\mathbf{x}, t)$.

2.2 Material interfaces and free-surface boundaries

Particles-based methods are especially well adapted to the treatment of internal interfaces for multi-fluids or free-surface boundaries when one of the external fluid may be neglected. In fact, the Lagrangian frames used in most particle-based methods facilitate the definition of the interfaces. In many cases, for instance when a meshless method or a moving mesh is used, the internal interfaces or the free-surface is directly defined by the particles. In other cases, where a background fixed mesh is used as in the MPM or PFEM-2, the interfaces are defined using a projection of a level set function. In order to improve the approximation on the internal surfaces, an enriched space can be introduced as it will be described for the PFEM-2 procedure.

3 The Smoothed Particle Hydrodynamic Method (SPH)

Smoothed-particle hydrodynamics (SPH) is a mesh-free Lagrangian method originally developed by Gingold and Monaghan (1977) and Lucy (1977). In the first publications, it was focused on the simulation of astrophysical

problems. Subsequently, it was applied to fluids (Monaghan, 1994), which is its current most popular application. Since its conception, it has also been applied to a broad range of fields, including solid mechanics (Allahdadi et al., 1993) , geomechanics, ballistics and FSI simulations (Hoover et al., 2006).

3.1 The interpolant

The basis of the SPH method lies in the employed interpolation. Any quantity $A(\mathbf{x})$ is defined using the following integral interpolant:

$$A(\mathbf{x}) = \int A(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', h)dx' \tag{9}$$

where $W(\mathbf{x}, h)$ is the kernel function. The previous interpolant reproduces $A(\mathbf{x})$ exactly if the kernel is a Dirac delta function. Since in practice only a finite number of points are available with a finite distance between each other, it is not possible to use the Dirac function. Instead, the interpolant is defined by a smooth function that is non-zero for a radius $< 2h$. A commonly used kernel is based on the cubic spline, expressed as a piecewise cubic polynomial. Defining $q = \frac{|\mathbf{x}-\mathbf{x}'|}{h}$, the spline has the form

$$M(q) = \begin{cases} \frac{1}{6}\left[(2-q)^3 - 4(1-q)^3\right] & \text{for } \ 0 \leq q \leq 1 \\ \frac{1}{6}(2-q)^3 & \text{for } \ 1 < q \leq 2 \\ 0 & \text{for } \ q > 2 \end{cases} \tag{10}$$

As noted in Monaghan (2005), the SPH kernel associated to $M(q)$ in one dimension ($D = 1$) is $W(\mathbf{x}, h) = M(q)/h$. For higher dimensions, the spline is multiplied by a different constant to normalize it and $W(\mathbf{x}, h) = M(q)/h^D$. The factor $1/6$ is replaced by $15/(14\pi)$ in 2D and by $1/(4\pi)$ in 3D. Despite higher order kernels offer better accuracy for homogeneous particle distribution, they require the cancellation of negative and positive distributions, besides losing the definite positive dissipation term since the sign of the gradient of the kernel changes sign. For this reason higher order splines will not be considered here.

The discretization process in the SPH method consists on splitting the domain into a finite set of particles with a fixed mass $m_p = \int_p \rho_p(\mathbf{x}')d\mathbf{x}'$. Then the value of A can be expressed as:

$$A(\mathbf{x}) = \int \frac{A(\mathbf{x}')}{\rho(\mathbf{x}')}\rho(\mathbf{x}')d\mathbf{x}' \tag{11}$$

Using the proposed mass discretization and the spline definition, the previous equation can be written as:

$$A(\mathbf{x}) = \sum_p \frac{A_p}{\rho_p(\mathbf{x}')}m_p W(\mathbf{x}-\mathbf{x}', h) \tag{12}$$

Since the cubic spline is given a zero value for particles that are at distance larger than $2h$, only a limited number of particles are needed to compute the value of A for a given point. The density function is defined as :

$$\rho(\mathbf{x}) = \sum_p m_p W(\mathbf{x}-\mathbf{x}', h) \tag{13}$$

If h is held constant in the whole domain, the integration of equation (13) yields the exact total mass, proving that mass is perfectly conserved, i.e.

$$\int \rho(\mathbf{x}) = \sum_p m_p \tag{14}$$

3.2 Derivative evaluations

Assuming W is differentiable, the gradient of A can be expressed as:

$$\nabla A = \sum_p m_p \frac{A_p}{\rho_p} \nabla W \tag{15}$$

Since the derivative of the kernel can be computed analytically, the gradients of the functions are exact derivatives of the approximation $A(\mathbf{x})$. Unfortunately, this approximation leads to important errors. As an example, the previous form (15) yields derivatives different from zero for constant functions, showing that it is unable of capturing the correct derivative value even for simple cases. By expanding $A(\mathbf{x})$ in Taylor series, defining the first derivative from there and subtracting the first error term (Price, 2012), more accurate gradient approximations can be obtained. In general, the enhanced derivatives will be based on the continuous form, where Φ is any differentiable function, i.e.

$$\nabla A = \frac{1}{\Phi} \left[\nabla(\Phi A) - A \nabla \Phi \right] \tag{16}$$

The discrete SPH versions of Eq. (16) evaluated in a particle a is:

$$(\nabla A)_a = \frac{1}{\Phi_a} \sum_p m_p \frac{\Phi_p}{\rho_p} (A_p - A_a) \nabla_a W(\mathbf{x}_p - \mathbf{x}_a, h) \tag{17}$$

Different choices of Φ yield all the derivatives in the SPH literature. For example, choosing $\Phi = 1$ gives

$$(\nabla A)_a = \sum_p \frac{m_p}{\rho_p} (A_p - A_a) \nabla_a W(\mathbf{x}_p - \mathbf{x}_a, h) \tag{18}$$

and choosing $\Phi = \rho$

$$(\nabla A)_a = \frac{1}{\rho} \sum_p m_p (A_p - A_a) \nabla_a W(\mathbf{x}_p - \mathbf{x}_a, h) \tag{19}$$

3.3 Second derivative evaluations

To compute the second derivatives, we proceed in the same way. Hence, deriving two times the kernel function gives

$$\nabla^2 A = \sum_p m_p \frac{A_p}{\rho_p} \nabla^2 W \tag{20}$$

Unfortunately, this simple form of the second derivatives also has its shortcomings, specially for particles that are poorly arranged or disorganized. Since $\nabla^2 W$ can change its sign, erratic behaviour is possible in diffusion problems, with fluxes in the opposite direction to the gradient. Consequently, using Eq. (20) the conservation laws are not guaranteed.

For these reasons, other approximations to the second derivatives have been proposed (Brookshaw, 1985). For instance

$$(\nabla^2 A)_a = \sum_p \frac{m_p}{\rho_p} (A_p - A_a) F_a(\mathbf{x}_p - \mathbf{x}_a, h) \tag{21}$$

where

$$F(\mathbf{x}_p - \mathbf{x}_a, h) = \frac{\nabla_a (\mathbf{x}_p - \mathbf{x}_a, h)}{|\mathbf{x}_p - \mathbf{x}_a|} \tag{22}$$

For 2D problems, similar approximated formulas for the Laplace equation take the form (Espanol and Revenga, 2003)

$$\left(\frac{\partial^2 A}{\partial x^i \partial x^j}\right)_a = \sum_p \frac{m_p}{\rho_p}\left(\frac{4\Delta x^i \Delta h^j}{\Delta x^2} - \delta^{ij}\right)(A_p - A_a)F_a(\mathbf{x}_p - \mathbf{x}_a, h) \tag{23}$$

where x^i is the ith component of $\mathbf{x}$ and $\Delta\mathbf{x} = \mathbf{x}_a - \mathbf{x}_p$.

Eqs. (18), (19) and (23) are used to approximate the first and second derivatives in Eqs.(5) to solve the Lagrangian Navier-Stokes equations. The position of the particles after each time step is simply approximated in an explicit way with a constant in time velocity as

$$\mathbf{x}_a^{n+1} = \mathbf{x}_a^n + \mathbf{V}\left[\mathbf{x}_a(t^n), t^n\right]\Delta t \tag{24}$$

3.4 Mass conservation

Since each particle has a fixed associated mass that is assigned at the onset of the simulation, the mass is guaranteed to be conserved without solving any equation. However, a normal force between the particles must be added in order to solve the compressibility equation of the fluid. The normal force is computed by comparing the density of the particle with the target density ρ_0 and multiplying this vector by the compressibility modulus, i.e.

$$(\mathbf{F})_a^{normal} \propto \sum_p \frac{\mathbf{x}_p - \mathbf{x}_a}{h}\kappa(\rho_a + \rho_p - 2\rho_0) \tag{25}$$

It must be noted that the compressibility modulus k will reduce considerably the maximum admissible time-step for a stable algorithm. For this reason, this coefficient is usually artificially lowered in order to reduce the required computational power.

3.5 Advantages and disadvantages of the SPH

The advantages of the SPH method are the same as for all particle methods. That is, pure advection is treated exactly; for multi material problems, each material is described by its own set of particles and the gap between the continuum and fragmentation is treated in a natural way. Also, in problems involving changes in the domain shape such as fragmentation and droplets, the computation is performed only where matter is located. Finally, it is often possible to include complex physics more easily than in other methods due to the explicit nature of the SHP technique.

Nevertheless, the disadvantages of the SPH method compared to other particle methods are also important. The time-steps must be small due to the explicit way to solve the mass conservation equation. Also, the particle distribution must be homogeneous or have smooth variations in order to obtain an acceptable solution in the evaluation of the forces. Failing to do so causes unreliable results which are specially unstable for the pressure values.

4 The Moving Particle Semi-implicit (MPS) method

Koshizuka and Oka (1996) developed in 1996 the moving particle semi-implicit method (MPS) for incompressible flows with free-surfaces. In the MPS method, fluids are represented by particles and the derivatives of the unknown functions are evaluated without a grid, as in SPH methods.

4.1 Particle interaction model

In the same way as for the SPH approach, the MPS technique uses a weighting function $W(\mathbf{x}-\mathbf{x}', h)$ to compute the interactions with the neighbouring particles. Defining again $q = \frac{|\mathbf{x}-\mathbf{x}'|}{h}$, W is defined as:

$$W(q) = \begin{cases} \frac{1}{q} & for \;\; 0 \le q \le 1 \\ 0 & for \;\; q > 1 \end{cases} \tag{26}$$

which implies that the interactions are limited to a radius h.

Using (26), the particle number density on the volume centered in the particle a and limited by the radius h can be obtained as:

$$(n)_a = \sum_{p \neq a} W(\mathbf{x}_a - \mathbf{x}_p, h) \tag{27}$$

In the previous equation, the number of particles in a unit volume ρ_n can be approximated as

$$(\rho_n)_a = \frac{(n)_a}{\int W(q)dr} \tag{28}$$

Using a constant radius h, the value of the integral can be computed analytically. Then the fluid density can be easily obtained with the following expression, where m are the particle masses

$$(\rho)_a = m\,(\rho_n)_a = \frac{m\,(n)_a}{\int W(q)dr} \tag{29}$$

From the previous equations it is clear that the fluid density depends directly on the particle number density. This translates into the requirement of a constant particle number density for incompressible fluids, defined as n^0. On the other hand, for problems in which several fluids with different densities interact between each other, the value of n^0 is kept constant in all the domain, varying only the value of the particle masses m.

4.2 Derivative evaluations

The gradient operator is computed in the MPS method using the contribution from all the particles within a radius h. The gradient between the point of interest a and each particle p is evaluated in the following way, where A is the quantity of interest

$$\nabla_{ap} A = (A_a - A_p)\frac{\mathbf{x}_a - \mathbf{x}_p}{|\mathbf{x}_a - \mathbf{x}_p|^2} \tag{30}$$

Using W as the weighting function for each of the contributions, the gradient reads

$$(\nabla A)_a = \frac{d}{n^0}\sum_{p \neq a}(A'_a - A_p)\frac{\mathbf{x}_p - \mathbf{x}_a}{|\mathbf{x}_a - \mathbf{x}_p|^2} W(\mathbf{x}_a - \mathbf{x}_p, h) \tag{31}$$

where d is the number of spatial dimensions and p refers to the particles within the radius h. Note that in Equation (31) the value of A_a is not used, but rather $A'_a = min(A_p)$. Since the gradient is, in principle, insensitive to the value in a, this does not affect the value of the gradient and, moreover, improves its definition for irregular particles distributions. This implies that $(A_p - A'_a)$ will always be positive, which, according to practical experience, leads to more robust algorithms (Koshizuka et al., 1998).

4.3 Second derivative evaluations

In continuum problems, the Laplacian of a scalar/vectorial field is required to solve the diffusion phenomenon, such as heat transfer or viscous forces. Due to the irregular distribution of the particles in the MPS method, this approach is not typically used. Instead, a discrete flow from two neighbour particles is computed and used for the diffusive term. The quantity given from particle p to particle a is defined as

$$\Delta_{p \to a} = \frac{2d\kappa\Delta t}{n^0 \lambda} A_p W(\mathbf{x}_a - \mathbf{x}_p, h) \tag{32}$$

where κ is the diffusivity and λ is introduced so that the change in the variable is equal to the analytical solution. Defining $r = |\mathbf{x} - \mathbf{x}'|$

$$\lambda = \frac{\int_V r^2 W(r)dr}{\int_V W(r)dr} \tag{33}$$

Taking into account the flow in the opposite direction (from particle a to p) and adding the contribution of all the particles within the radius h, the Laplacian term is approximated by the following expression

$$(\nabla^2 A)_a = \sum_{p \neq a} \frac{2d\kappa\Delta t}{n^0 \lambda}(A_p - A_a) W(\mathbf{x}_a - \mathbf{x}_p, h) \tag{34}$$

Note that the scheme is conservative since the quantity contributed by one particle is exactly the same as the one received by its neighbour.

4.4 Modeling of incompressibility

As stated in Section 4.1, the particle number density must be held constant in order to guarantee incompressibility. Starting from an initial value n^0 at the time-step n, the convection of the particles will lead to a new value n* $\neq n^0$, which differs from the target value by $n' : n^* + n' = n^0$. Using the mass conservation expression equation (1b), the value of n' can be expressed as a change in the divergence of the velocity field $\nabla \cdot \mathbf{V}'$ as

$$\frac{1}{\Delta t}\frac{n'}{n^0} = \nabla \cdot \mathbf{V}' \tag{35}$$

The change in the velocity field $\mathbf{V}'$ can be directly linked to a pressure change using the following expression

$$\mathbf{V}' = \frac{\Delta t}{\rho} \nabla p^{n+1} \tag{36}$$

Inserting Equation (36) into (35) and recalling that $n^* + n' = n^0$, the Laplacian equation for the pressure is obtained as

$$(\nabla^2 p^{n+1})_a = \frac{\rho}{\Delta t^2} \frac{(n^*)_a - n^0}{n^0} \tag{37}$$

Since the incompressibility condition implies that the speed of sound is infinite, the previous equation has to be solved implicitly. In other words, a system of equations has to be assembled and solved at each time step.

4.5 Boundary conditions

The definition of density is linked to the particle number density through Equation (29). Then it follows that if

$$(n)_a^* \leq \beta n^0 \tag{38}$$

the particle a is considered to lay on the free surface. The parameter β is adjusted so that the small deviations from n^0 inside the fluid are not mistaken with the location of the free surface. In general $\beta = 0.95$ yields good results. Once the particles fulfilling this criterion have been identified, the pressure is strongly imposed at these points as a Dirichlet boundary condition in the Laplacian equation for the pressure (equation (37)). Having done this, no additional conditions are required to capture the behaviour of the free surface. The algorithm is capable of simulating its evolution including fragmentation and coalescence of water drops.

Solid walls have to be modelled differently. Since at the fluid regions close to the wall $(n)_a = n^0$ must be held, layers of particles with a fixed velocity and $(n)_{wall} = n^0$ have to be added to define the walls. The number of layers depends on the size of h, but in general three layers yield good results.

4.6 Advantages and disadvantages of MPS

The MPS method shares the advantages of all particle methods, that is pure advection is treated exactly; for multi-materials, each material is described by its own set of particles and the gap between continuum and fragmentation is treated in a natural way. Also in problems involving fragments or drops; the computation takes place only where matter is located and it is often possible to include complex physics easier than in other methods. Furthermore, the semi implicit treatment of the pressure terms in the MPS method introduces a big advantage compared to the SPH method in the solution of incompressible flows, allowing to solve the equations without decreasing the time step to zero.

The main disadvantage of the MPS method is the way to evaluate the derivatives, which is not accurate for non-homogeneous particle distributions. In order to avoid this shortcoming, the particle distribution on the total domain must be homogeneous or have smooth variations for obtaining acceptable solutions.

5 The Material Point Method (MPM)

The MPM is a Lagrangian particle-based procedure to solve continuum mechanics problems. The MPM was developed as an extension of the Particle-in-Cell (PIC) (Brackbill and Ruppel, 1986) method in order to expand the capabilities of this procedure, limited to fluids. The main improvements of the MPM over the PIC technique were possibility to write the equations in a weak form (so that finite elements could be combined with particles), and the possibility to use history-dependant models. These additions allow the MPM to model both fluid and solid materials.

The distinct approach of the MPM consists on using a set of N^p material points with extensive properties (i.e. mass and momentum). A standard finite element method (FEM) is then used to solve the momentum equations. Contrary to the previously described particle methods, the MPM relies on a fixed mesh to evaluate de derivatives of the unknown functions.

5.1 Discretized equations in the MPM

Let us discretize the domain into a finite set of material particles with position $\mathbf{x}_p^t$ at time t. As stated previously, the material points will have associated extensive and intensive properties such as mass m_p, density ρ_p, Cauchy stress tensor $\boldsymbol{\sigma}$, strain $\mathbf{s}$, and any other variable that the problem requires. In the same way, and similarly to the MPS method, the mass will be conserved as long as the number of points remains unchanged during the analysis.

The difference between the MPM and the previously described particle methods appears at this point. Instead of using the particles themselves to create the field of unknowns and their derivatives, the MPM lies on a background FEM mesh. Then it follows that the information of the particles has to be mapped into the nodes of this fixed mesh. The nodal values are used to interpolate the information from the nodes to the material particles as in the standard FEM. Having done this, the momentum equations are written in the standard weak FEM form. Using the FEM shape functions $\mathbf{N}$ as test functions and defining the specific stress as $\boldsymbol{\sigma}^s = \boldsymbol{\sigma}/\rho$, the variational form of the momentum Equation (5a) reads

$$\int_\Omega \rho \mathbf{N} \frac{D\mathbf{V}}{Dt} d\Omega = -\int_\Omega \rho \boldsymbol{\sigma}^s : \nabla \mathbf{N} d\Omega + \int_\Omega \mathbf{N} \rho d\Omega + \int_{\Gamma_\sigma} \mathbf{N} \rho \bar{\boldsymbol{\sigma}}^s \Gamma \tag{39}$$

where Ω is the analysis domain and Γ_σ is the Neumann boundary with prescribed tractions. As in the standard FEM, $\mathbf{N}$ is chosen to take a unit value at the Dirichlet boundaries.

Unlike the MPS method, the density field is not approximated with a smooth function, but rather using a discrete distribution via the Dirac δ function. In this way the conservation of the total mass is guaranteed, but the density field itself is not defined at every point. This is computed as

$$\rho \mathbf{x} = \sum_p m_p \delta(\mathbf{x} - \mathbf{x}_p^t) \tag{40}$$

where $\delta(\cdot)$ is the Dirac delta function.

Inserting Equation (40) into (39), the integrals are transformed into their discrete MPM counterparts as

$$\sum_p m_p \mathbf{N}(\mathbf{x}_p^t) \frac{D\mathbf{V}^t(\mathbf{x}_p^t)}{Dt} = \sum_p m_p \left[-\nabla \mathbf{N}(\mathbf{x}_p^t) \boldsymbol{\sigma}^s(\mathbf{x}_p^t) + \mathbf{N}(\mathbf{x}_p^t) \bar{\boldsymbol{\sigma}}^s(\mathbf{x}_p^t) h^{-1} + \mathbf{N}(\mathbf{x}_p^t) \mathbf{b}(\mathbf{x}_p^t) \right] \tag{41}$$

where h is the thickness of the boundary layer, if present.

Since the Dirac delta function δ is not differentiable, an alternative procedure to calculate the gradients has to be found. In this sense, the background mesh provides the smooth functions defined by the FEM approximation with well-defined derivatives within the different finite element in the mesh. Using the shape functions $\mathbf{N}$, the velocity of the pth particle can be expressed as the sum of all the non-zero i functions at that point. In the same way, the Lagrangian framework allows us to express the velocity and the acceleration in the same form, namely

$$\mathbf{V}_p^t = \sum_i \mathbf{N}_i(\mathbf{x}_p^t) \bar{\mathbf{V}}_i^t \tag{42}$$

$$\frac{D\mathbf{V}_p^t}{Dt} = \mathbf{a}_p^t = \sum_i \mathbf{N}_i(\mathbf{x}_p^t)\bar{\mathbf{a}}_i^t \tag{43}$$

where $\bar{\mathbf{V}}_i$ and $\bar{\mathbf{a}}_i$ are the nodal velocity vector and the acceleration vector of node i, respectively.

Inserting Equation (42) into (41) gives

$$\begin{aligned}\sum_i \sum_j \mathbf{M}_{ij}^t \frac{D\mathbf{V}_i^t}{Dt} = &- \sum_i \sum_p m_p \mathbf{N}_i(\mathbf{x}_p^t)\boldsymbol{\sigma}^s(\mathbf{x}_p^t)\nabla \mathbf{N}_i(\mathbf{x}_p^t) \\ &+ \sum_i \sum_p m_p \bar{\boldsymbol{\sigma}}^s h^{-1} \mathbf{N}_i(\mathbf{x}_p^t) + \sum_i \sum_p m_p \boldsymbol{b}_p \mathbf{N}_i(\mathbf{x}_p^t)\end{aligned} \tag{44}$$

where $\mathbf{M}_{ij}^t$ are the components of the mass matrix

$$\mathbf{M}_{ij}^t = \sum_p m_p \mathbf{N}_i(\mathbf{x}_p^t)\mathbf{N}_j(\mathbf{x}_p^t) \tag{45}$$

The system of Equations (44) is the result of mapping the information of the particles into the mesh using the finite element shape functions. This system can be solved either with an implicit strategy by computing the corresponding stiffness matrices, or with an explicit technique by lumping the mass matrix. In general the latter is more popular due to the lower computational cost and a more natural approach for non-linear analysis. When using explicit strategies, the time step must be set below the critical value to avoid instabilities in the algorithm.

Once the nodal accelerations have been computed, the new velocities are used to update the position of the particles (the material points). On the other hand, the stresses are updated using the gradients of the FEM shape functions at the material point position. Having updated both the velocity and the position of the particles, the new step can be started. Since the background mesh does not store any information, the previous mesh can be discarded and a new one is created if the problem requires so.

5.2 Advantages and disadvantages of the MPM

The advantages of the MPM are the same as those of all particle methods, that is pure advection is treated exactly and each material is described by its own set of particles. Compared to the previously described particle methods, the background mesh can be refined in areas where there are large gradients of the unknowns, which improves considerably the efficiency of the method. The use of the FEM mesh also allows to obtain a better approximation on the variables and its gradients, leading to higher accuracy specially in the forces and stresses.

The main disadvantages of the MPM is the way to evaluate the mass conservation. The fact that each particle has its own mass is a big handicap because the number of particles must be held constant during all the process in order to preserve the total mass. This can cause an irregular distribution of particles in large deformation problems, difficulting the use variable mesh sizes (to enhance the accuracy in certain areas). The way to move the particles also restricts the efficiency of the method, giving acceptable solutions only for very small time-steps. On the other hand, the lack of a pressure equation in the MPM can induce a locking behaviour, which might require special techniques to avoid a non-physical numerical solution for the fluid flow (Mast et al., 2012).

6 The Particle Finite Element Method (PFEM)

Similarly to the MPM, the particle finite element method (PFEM) is a particle method that uses a mesh to evaluate de derivatives of the unknown functions. Many of the disadvantages or the MPM are overcome in the PFEM, making it a very versatile and efficient method for analysis of multi-fluids, free-surface fluids and fluid-structure interaction (FSI) problems with or without free-surfaces. The PFEM is particularly suited to problems with breaking waves and splashing of liquids (Idelsohn et al., 2003b; Oñate et al., 2004).

Currently there exists two PFEM strategies; the standard PFEM using moving meshes and then another one using fixed meshes (as in the MPM). The latter has been named PFEM-2 and incorporates other important

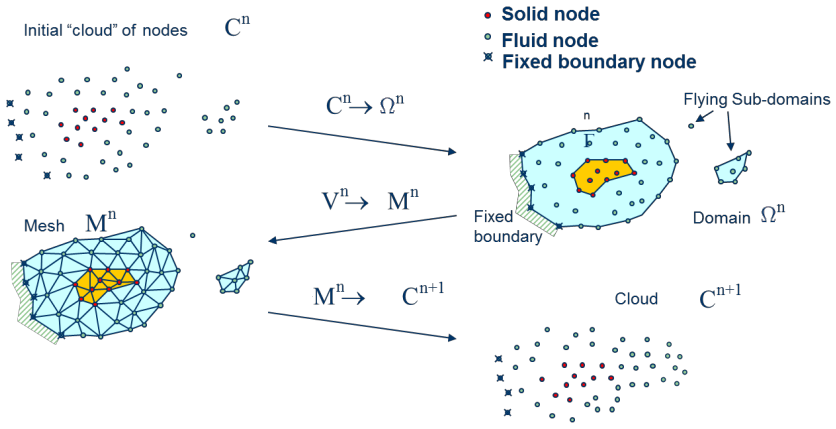


Figure 1: Sequence of steps to update a "cloud" of nodes representing a domain containing a fluid and a solid part from time step n ($t = t_n$) to time $n + 1$ ($t = t_n + 1\Delta t$)

differences with the hereafter "standard" PFEM concerning the time integration of the equations (Idelsohn et al., 2013). For this reason the PFEM-2 is explained in a separate section.

The main component of the PFEM in both versions, either with moving or fixed mesh, is a set of particles that move in a Lagrangian frame, convecting all the intrinsic and extrinsic properties and variables (e.g. density, viscosity, conductivity, velocity, pressure, etc.). These values are projected at the end of each time-step on a moving mesh (for the standard PFEM) or on a fixed mesh (for the PFEM-2). The main difference with the MPM is that particles in the PFEM do not have an associated mass, i.e. they are material points without any associated volume. In the moving mesh version (the standard PFEM) the advantage is that, as the mesh moves together with the particles, it is not necessary any projection from the particles to the mesh. Conversely, in the PFEM-2 a permanent projection from the particles to the mesh is needed. The disadvantage of the standard PFEM versus the MPM and the PFEM-2 is the need for a permanent remeshing.

6.1 The basis of the standard PFEM

Let us consider a domain containing both fluid and solid subdomains (the solid subdomain may include soil/rock materials and/or structural elements). The moving fluid particles interact with the solid boundaries, thereby inducing the deformation of the solid, which in turn affects the flow motion, leading to a fully coupled problem.

In the PFEM both the fluid and the solid domains are modelled using an *updated Lagrangian formulation* (Idelsohn et al., 2003b; Oñate et al., 2004; Oñate and Carbonell, 2014; Zienkiewicz and Taylor, 2005). That is, all variables are assumed to be known in the *current configuration* at time t. The new set of variables in both domains are sought for in the *next or updated configuration* at time $t + \Delta t$. The finite element method (FEM) is used to solve the equations of continuum mechanics for each of the subdomains. Hence a mesh discretizing these domains must be generated in order to solve the governing equations for each subdomain in the standard FEM fashion.

The quality of the numerical solution depends on the discretization chosen as in the standard FEM. Adaptive mesh refinement techniques can be used to improve the solution in zones where large motions of the fluid or the structure occur.

The solution of the problem using the PFEM involves four steps, as show in Figure 1. In the first step, the cloud of points C^n at time t_n is convected using the last known velocity. Having determined the new analysis domain (Ω^n) configuration, the boudaries between the different phases are determined and later a FEM mesh (M^n) is created using the points C^n as nodes. Using this FEM mesh, the Lagrangian system of equations is solved and then the connectivities of the mesh are erased to start the new time step.

6.2 Solution of the governing equations in the PFEM

Both equations (5) are used in PFEM. This means that not only the momentum conservation equations are solved, but also the mass conservation is implicitly solved. This is an important difference with the SPH, in which the mass conservation is solved explicitly, and also with the MPM, where the mass conservation is not solved at all because the mass is preserved by keeping constant the number of particles.

Equation (5a) is integrated implicitly in time as:

$$\rho\frac{D\mathbf{V}}{Dt} \approx \rho\frac{\mathbf{V}\left(\mathbf{x}_p^{n+1}, t^{n+1}\right) - \mathbf{V}\left(\mathbf{x}_p^{n}, t^{n}\right)}{\Delta t} = \rho\frac{\mathbf{V}^{n+1} - \mathbf{V}^{n}}{\Delta t} = \left[-\nabla p + \nabla\cdot(\mu\nabla\mathbf{V}) + \rho\mathbf{b}\right]^{n+\theta} \tag{46}$$

Only the case of $\theta = 1$ (fully implicit) will be considered next. Other values (for instance $\theta = 1/2$), may be considered without major changes. The time-integrated equations become:

$$\rho\frac{\mathbf{V}^{n+1} - \mathbf{V}^{n}}{\Delta t} = -\nabla p^{n+1} + \left[\nabla\cdot(\mu\nabla\mathbf{V}) + \rho\mathbf{b}\right]^{n+1} \tag{47}$$

The mass conservation equation is also integrated implicitly as:

$$\frac{D\rho}{Dt} \approx \frac{\rho^{n+1} - \rho^{n}}{\Delta t} = -\nabla\rho^{n+1}\cdot(\rho\boldsymbol{V}^{n+1}) \tag{48}$$

The time integration of equations (47) presents some difficulties because it is a fully-coupled equation involving four degrees of freedom per node. When the fluid is incompressible (or nearly incompressible) and the Laplacian form of the viscosity is used, advantages can be taken from the fact that in equations (47), the three components of the velocity are only coupled via the pressure. Ever though the monolithic solution of equations (47) and (48) is possible, the fractional-step method proposed in Idelsohn et al. (2003b) will be used. This basically consists in splitting each time step into two pseudo-times steps. In the first step the implicit part of the pressure is avoided in order to have a decoupled equation in each of the velocity components. The implicit part of the pressure is added during a second step. The fractional-step algorithm for equations (47) and (48) is the following:

Split of the momentum equations:

$$\frac{D\mathbf{V}}{Dt} \approx \frac{\mathbf{V}^{n+1} - \mathbf{V}^{n}}{\Delta t} = \frac{\mathbf{V}^{n+1} - \hat{\mathbf{V}} + \hat{\mathbf{V}} - \mathbf{V}^{n}}{\Delta t} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla\cdot\boldsymbol{\tau}^{n+\theta} + \mathbf{b} \tag{49}$$

where $\hat{\mathbf{V}}$ are fictitious variables termed fractional velocities defined by the split:

$$\text{A)}\quad \hat{\mathbf{V}} = \mathbf{V}^{n} - \gamma\frac{\Delta t}{\rho}\nabla p + \frac{\Delta t}{\rho}\nabla\cdot\boldsymbol{\tau}^{n+\theta} + \Delta t\ \mathbf{b} \tag{50a}$$

$$\text{C)}\quad \mathbf{V}^{n+1} = \hat{\mathbf{V}} - \frac{\Delta t}{\rho}\nabla(p^{n+1} - \gamma p^{n}) \tag{50b}$$

in which $p^n = p(\mathbf{x}, t^n)$ is the value of the pressure at time t^n but evaluated at the final position of the particles and is considered constant in time (50b).

In equation (50) γ defines the pressure splitting, and can take the value 0 or 1. For $\gamma = 1$ the algorithm can achieve a quadratic splitting error (SE2), but requires adding stabilization terms on the pressure equation (Hughes et al., 1986; Codina and Badia, 2006). In this work $\gamma = 0$ will be used, leading to only linear convergence (SE1) only but also to a stable method that does not require stabilization terms for time-steps that do not approach zero.

Taking into account (2), the last term in equation (50a) may be written as:

$$\nabla\cdot\boldsymbol{\tau}^{n+\theta} = \nabla\cdot\left[\mu(1-\theta)\nabla\mathbf{V}^{n}\right] + \nabla\cdot\left(\mu\theta\nabla\mathbf{V}^{n+1}\right) \tag{51}$$

Using the fractional velocity instead of the final velocity yields

$$\nabla\cdot\boldsymbol{\tau}^{n+\theta} \approx \nabla\cdot\left[\mu(1-\theta)\nabla\mathbf{V}^{n}\right] + \nabla\cdot\left(\mu\theta\nabla\hat{\mathbf{V}}\right) \tag{52}$$

Setting $\theta = 1$ and $\gamma = 0$, equation (50a) can be expressed as:

$$\hat{\mathbf{V}} - \frac{\Delta t}{\rho}\nabla \cdot \left(\mu \nabla \hat{\mathbf{V}}\right) = \mathbf{V}^n + \Delta t\ \mathbf{b} \tag{53}$$

Splitting also the mass conservation equation gives

$$\frac{D\rho}{Dt} \approx \frac{\rho^{n+1} - \rho^n}{\Delta t} = \frac{\rho^{n+1} - \hat{\rho} + \hat{\rho} - \rho^n}{\Delta t} = \rho \nabla \cdot \left(\mathbf{V}^{n+1} - \hat{\mathbf{V}} + \hat{\mathbf{V}}\right) \tag{54}$$

where $\hat{\rho}$ is a fictitious variable defined by the split:

$$\frac{\hat{\rho} - \rho^n}{\Delta t} = -\rho \nabla \cdot \hat{\mathbf{V}} \tag{55a}$$

$$\frac{\rho^{n+1} - \hat{\rho}}{\Delta t} = -\rho \nabla \cdot \left(\mathbf{V}^{n+1} - \hat{\mathbf{V}}\right) \tag{55b}$$

Using equation (50b) the coupled mass-momentum equation (55b) becomes:

$$\text{B)} \quad \frac{\rho^{n+1} - \hat{\rho}}{\Delta t} = \nabla \cdot \left(\nabla p^{n+1}\right) = \nabla^2 p^{n+1} \tag{56}$$

Taking into account equation (55a) the above expression can be written as:

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} + \frac{\rho}{\Delta t}\nabla \cdot \hat{\mathbf{V}} = \nabla^2 p^{n+1} \tag{57}$$

In equation (57) the incompressibility condition has not been introduced yet. The simplest way to introduce this condition in a Lagrangian formulation is to write:

$$\rho^{n+1} = \rho^n = \rho^0 = \hat{\rho} \tag{58}$$

Then, the first term of equation (57) vanishes, giving

$$\frac{\rho}{\Delta t}\nabla \cdot \hat{\mathbf{V}} = \nabla^2 p^{n+1} \tag{59}$$

As a summary, the three-step fractional method used can be expressed as:

$$\text{A)} \quad \hat{\mathbf{V}} - \frac{\Delta t}{\rho}\nabla \cdot \left(\mu \nabla \hat{\mathbf{V}}\right) = \mathbf{V}^n + \Delta t\ \mathbf{b} \quad \rightarrow \hat{\mathbf{V}} \tag{60a}$$

$$\text{B)} \quad \nabla^2 p^{n+1} = \frac{\rho}{\Delta t}\nabla \cdot \hat{\mathbf{V}} \quad \rightarrow p^{n+1} \tag{60b}$$

$$\text{C)} \quad \mathbf{V}^{n+1} = \hat{\mathbf{V}} - \frac{\Delta t}{\rho}\nabla p^{n+1} \quad \rightarrow \mathbf{V}^{n+1} \tag{60c}$$

6.3 Generation of a new mesh

One of the key points for a competitive PFEM formulation is the fast regeneration of the mesh at every time step on the basis of the position of the nodes in the space domain. An efficient way to do so is generating the mesh using the so-called extended Delaunay tesselation (EDT) presented in Idelsohn et al. (2003a). The EDT allows generating meshes of elements with arbitrary polyhedrical shapes, combining triangles, quadrilaterals and other polygons in 2D and tetrahedra, hexahedra and arbitrary polyhedra in 3D in a computing time of order n, being n the total number of nodes in the mesh.

The shape functions for arbitrary polyhedral elements can be simply obtained using the so-called non-Sibsonian interpolations. Details of the mesh generation procedure and the shape functions for arbitrary polyhedra can be found in Idelsohn et al. (2003b). Once the new mesh has been generated at each time step, the numerical solution is found using the standard finite element interpolation.

6.4 Summary of a time-step solution with the standard PFEM

Starting with the known values $\mathbf{V}^n$ and p^n for each particle, the computation of the new particle position at t_{n+1} involves the following steps:

1. Approximate $\mathbf{V}^{n+1}$ (For the first iteration $\mathbf{V}^{n+1} = \mathbf{V}^n$. For the subsequent iterations the value of $\mathbf{V}^{n+1}$ corresponding to the last iteration is taken).
2. Move the particles to the $\mathbf{x}^{n+1}$ position and generate a new mesh.
3. Evaluate the $\hat{\mathbf{V}}$ velocity from (60a). It must be noted that the mass matrix and the stiffness matrix are split in 3 blocks. Then, these equations can be solved separately for $\hat{V}_x$, $\hat{V}_y$ and $\hat{V}_z$. For $\theta \neq 0$ (implicit scheme) this involves the solution of three Laplacian equations. For $\theta = 0$ (explicit scheme) the mass matrix may be lumped and inverted directly.
4. Evaluate the pressure p^{n+1} by solving the Laplacian equation (60b).
5. Evaluate the velocity $\mathbf{V}^{n+1}$ using (60c). Go to 1 until convergence.

The Lagrangian fractional step algorithm described has two important advantages:

- Step 3. is linear and may be explicit ($\theta = 0$) or implicit ($\theta \neq 0$). The nonlinearity remains only due to the unknown of the final particle position.
- In all steps, the system of equations to be solved are the evaluation of the velocity components (Step 3) and the evaluation of the pressure (Step 4). Those systems are scalar (only one degree of freedom per node), symmetric and positive definite. Then, it is very easy to solve them using a symmetric iterative scheme (such as the conjugate gradient method).

6.5 Stabilization of the incompressibility condition

In the Eulerian form of the momentum equations, the discrete form must be stabilized in order to avoid numerical wiggles in the velocity and pressure results. This is not the case in Lagrangian formulations such as PFEM, where no stabilization term are required in equation (49). Nevertheless, the incompressibility condition must be stabilized when equal-order velocity-pressure FEM approximations are used to avoid possible pressure oscillations in some particular cases.

For instance for $\gamma = 1$ or for small time-step increments (Courant number much less than one) it is well known that the fractional-step method does not stabilize pressure waves (Codina and Badia, 2006; Oñate et al., 2014a). In those cases, a stabilization term must be introduced in equation (60b) in order to eliminate pressure oscillations. A simple and effective procedure to derive a stabilized formulation for incompressible flows is based in the Finite Increment Calculus (FIC)(also called Finite Calculus) formulation (Oñate, 2004). Applications of the FIC approach to the PFEM can be found in Oñate et al. (2006).

In all the examples presented in this chapter, the γ parameter was always set to zero and the time increments were fixed to a given value so that the Courant number ≈ 1, avoiding in this way all the stability problems.

6.6 Boundary surface recognition

One of the main problems in the mesh generation stage is the correct definition of the boundary domain. Sometimes, boundary nodes are explicitly defined as special nodes, which are different from internal nodes. In other cases, the total set of nodes is the only information available and the algorithm must recognize the boundary nodes. Such is the case in particle methods such as PFEM in which, at each time step, the new coordinates of the particles are obtained and the boundary-surface must be recognized using the new particle positions. The use of the PFEM with the Extended Delaunay partition makes it easier to recognize boundary nodes.

Considering that the particles follow a variable $h(\mathbf{x})$ distribution, where $h(\mathbf{x})$ is the minimum distance between two particles, the following criterion has been used:

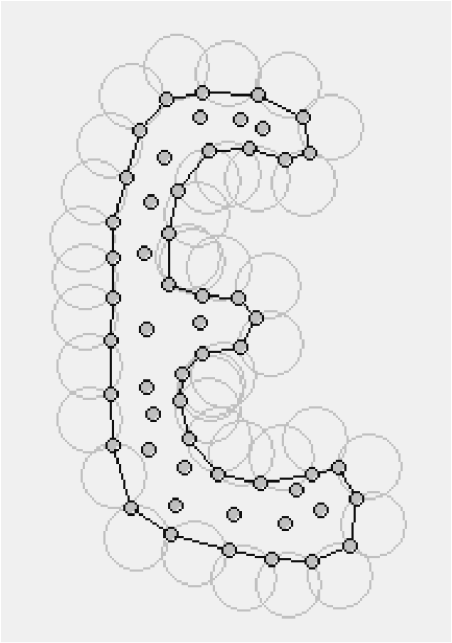


Figure 2: Contour recognition: Empty circles with radius $\alpha h(\mathbf{x})$ define the boundary particles.

All the particles on an empty sphere with a radius $r(\mathbf{x})$ bigger than $\alpha h(\mathbf{x})$, are considered as boundary particle (Figure 2). Thus, α is a parameter close to, but greater than one. This criterion is coincident with the Alpha Shape concept (Edelsbrunner and Mücke, 1994).

Once a decision has been made concerning which of the particles are on the boundaries, the boundary surface must be defined. It is well known that for 3D problems the surface fitting a number of particles is not unique. For instance, four boundary particles on the same sphere may define two different boundary surfaces, a concave one and a convex one. To overcome this problem, the boundary surface is defined with all the polyhedral surfaces having all their particles on the boundary and belonging to just one polyhedron.

The correct boundary surface is important for computing the normal vectors to the interfaces in problems including surface tension. Furthermore; in Galerkin FEM a correct evaluation of the volume domain is also important. Nevertheless, it must be noted that in the criterion proposed above, the error in the boundary surface definition is proportional to h. This is the error order accepted in a numerical method for a given nodal distribution. The only way to obtain a more accurate boundary surface definition is by reducing the distance between the particles.

6.7 Treatment of contact conditions in the standard PFEM

6.7.1 Contact between the fluid and a fixed boundary

The condition of prescribed velocities at the fixed boundaries in the PFEM is applied in strong form to the boundary nodes. These nodes might belong to fixed external boundaries or to moving boundaries linked to the interacting solids. Contact between the fluid particles and the fixed boundaries is accounted for by the incompressibility condition which *naturally prevents the fluid nodes to penetrate into the solid boundaries*, providing the time step increment is adjusted accordingly (Oñate et al., 2004, 2008).

6.7.2 Contact between solid-solid interfaces

The contact between two solid interfaces is simply treated in the PFEM by introducing a layer of *contact elements* between the two interacting solid interfaces. This layer is *automatically created during the mesh generation step* by prescribing a minimum distance (h_c) between two solid boundaries. If the distance exceeds the minimum value (h_c) then the generated elements are treated as fluid elements. Otherwise the elements are

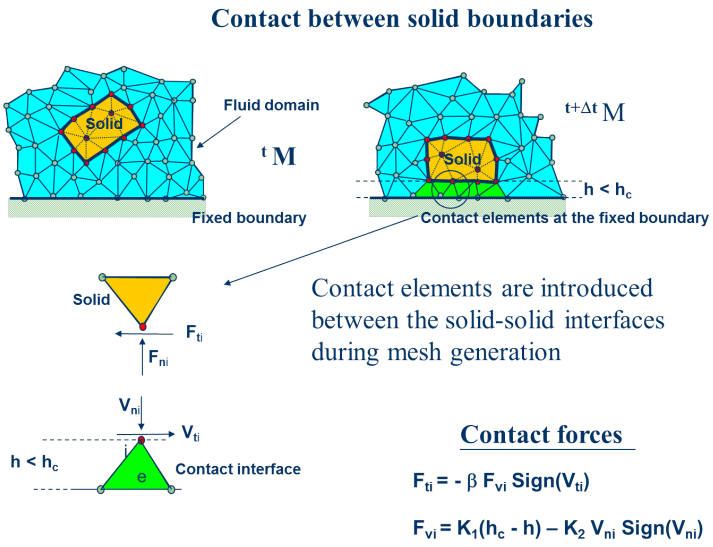


Figure 3: Modelling of contact conditions at a solid-solid interface with the PFEM

treated as contact elements where a relationship between the tangential and normal forces and the corresponding displacement is introduced (Figure 3).

This algorithm has proven to be very effective and it allows to identifying and modeling complex frictional contact conditions between two or more interacting bodies moving in water in an extremely simple manner.

This algorithm can also be used effectively to model frictional contact conditions between rigid or elastic solids in structural mechanics applications (Carbonell et al., 2010; Oñate et al., 2008). The PFEM contact algorithm has been successfully extended to model bed erosion in rivers and the sea bottom under the frictional forces induced by the water motion and the dragging of bed material. Details can be found in (Oñate et al., 2008).

6.8 Some outstanding results using the standard PFEM

The examples chosen show the applicability of the standard PFEM to solve problems involving large motions of the free surface, fluid-multi-body and multi-fluid interactions, and burning, melting and dripping of polymers.

The examples presented were solved using the PFEM and the formulation described above implemented in the Kratos software platform (Dadvand et al., 2010) and the PFLOW software developed at CIMNE.

6.8.1 Rigid objects falling into water

Figure 4 shows the penetration and evolution of a square and a circle of rigid shape in a container with water. The colours denote the different sizes of the elements at several times. In order to increase the accuracy of the FSI problem, smaller size elements have been generated in the vicinity of the moving bodies during their motion (Figure 5). The possibility to generate refined meshes in some particular domains is one of the main advantages of the PFEM.

The examples shown in Figure 6 and Figure 7 evidence the potential of the PFEM to solve 3D problems involving complex interactions between water and moving solid objects. Figure 6 shows the simulation of two tetrapods falling in a water container. Figure 7 shows the motion of a collection of ten tetrapods placed in a slope under an incident wave. The example shows the complex 3D interactions between the water particles and the tetrapods and between the tetrapods themselves, which can be easily modeled with the PFEM.

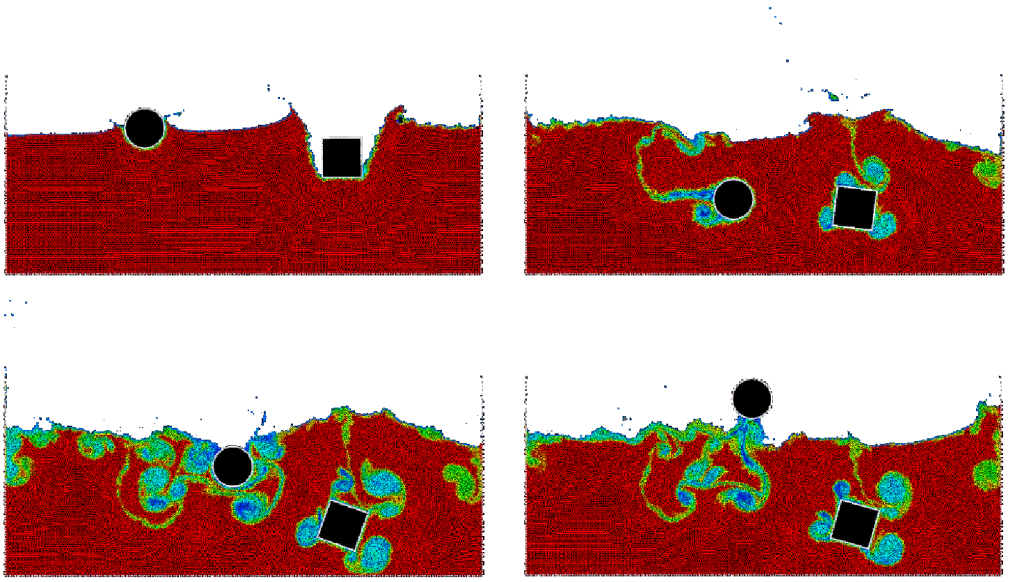


Figure 4: 2-D simulation of the penetration and evolution of a square and a circle in a water container. The colors denote the different sizes of the elements at several time-steps.

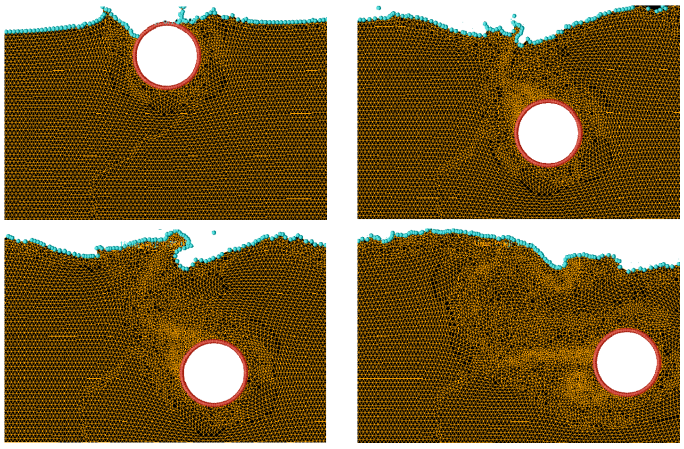


Figure 5: Detail of element sizes during the motion of a rigid circle within a water container.

6.8.2 Impact of sea waves on piers and breakwaters

Figures 8 and 9 shows the analysis of the effect of breaking waves on two different sites of a breakwater containing reinforced concrete blocks (each one of 4 m). The figures correspond to the study of Langosteira harbour in A Coruña, Spain using the PFEM.

6.8.3 The landslide in Lituya Bay

A case of much interest is when landslides occur in the vicinity of a reservoir (Fritz, 2001). The fall of debris material into the reservoir typically induces large waves that can overtop the dam, originating an unexpected flooding that can cause severe damage to the constructions and population in the downstream area.

In this example, results of the 3D analysis of the landslide produced in Lituya Bay (Alaska) on July 9, 1958, are presented (Figure 10). The landslide was originated by an earthquake and mobilized 90 millions tons of rocks that fell on the bay, originating a large wave that travelled to the opposite coast, reaching a height of $524m$.

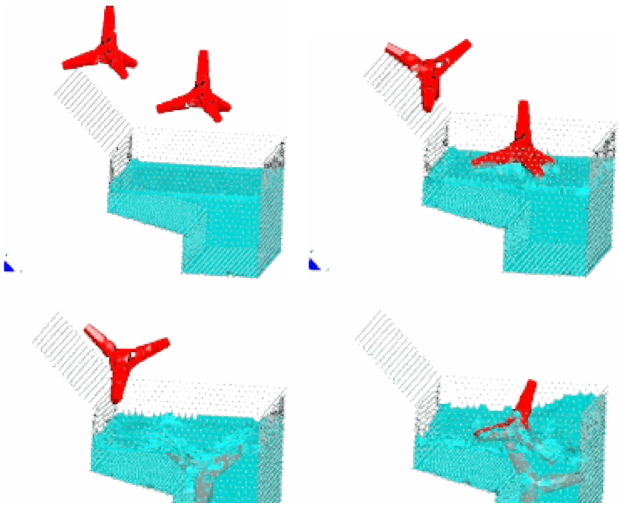


Figure 6: Motion of two tetrapods falling in a water container.

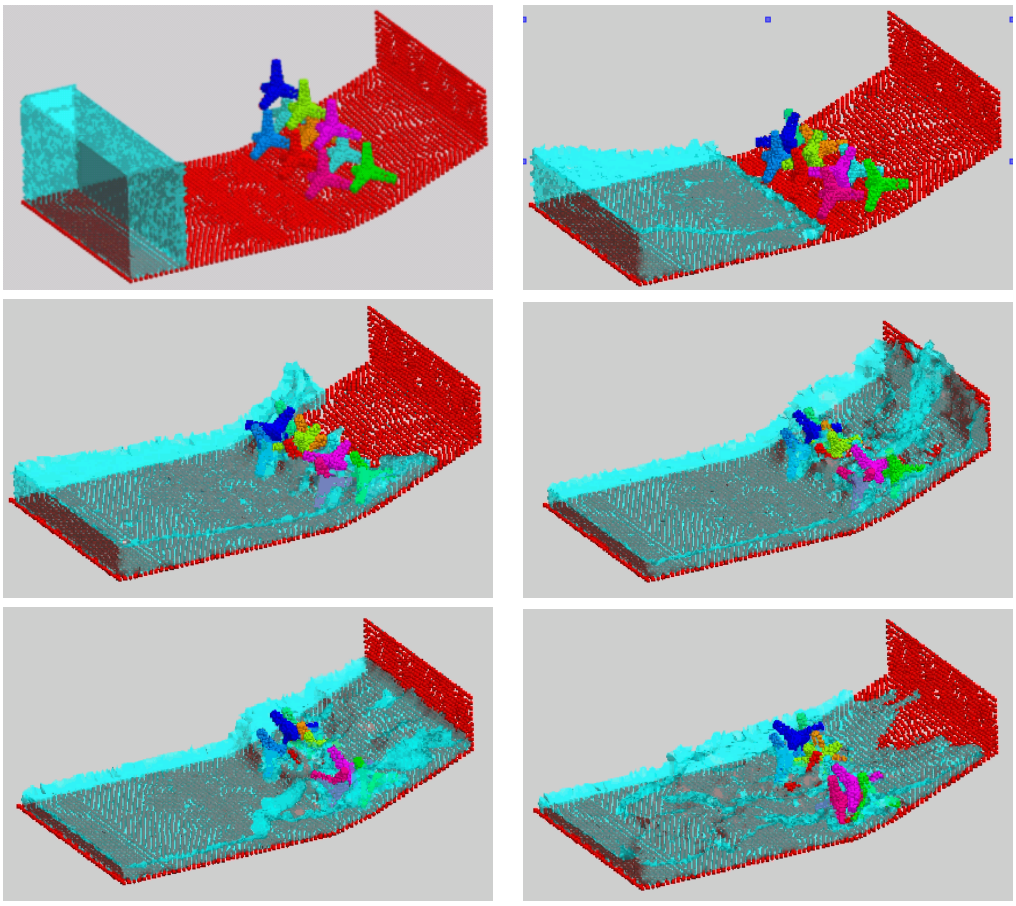


Figure 7: Motion of ten tetrapods on a slope under an incident wave.

Figure 11 show snapshots of the simulation of the landslide with the PFEM. The sliding mass has been modelled as a quasi-incompressible continuum with a prescribed shear modulus. No frictional effect between the sliding mass and the underneath soil has been considered. Also the analysis has not taken into account the

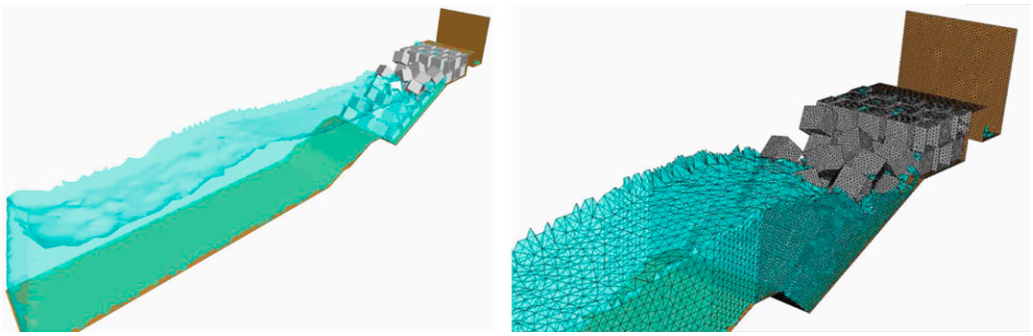


Figure 8: Breaking waves on breakwater slope containing reinforced concrete blocks. Mesh of 4-noded tetrahedra near the slope.

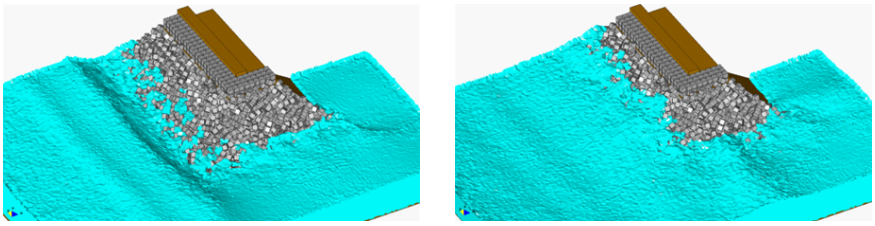


Figure 9: Study of breaking waves on the edge of a breakwater structure formed by reinforced concrete blocks

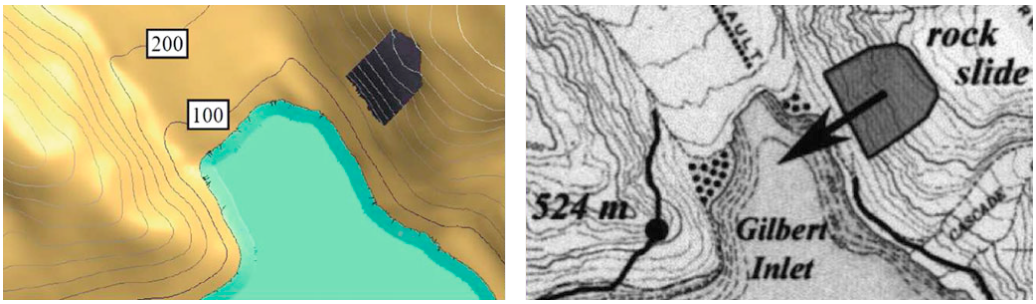


Figure 10: Lituya Bay landslide. Left geometry for the simulation. Right landslide direction and maximum wave level.

erosion and dragging of soil material induced by the landslide mass during motion.

The PFEM results have been compared to the observed values of the maximum water level in the north hill adjacent to the reservoir. The water height reached in this hill obtained with the PFEM was $551m$, which is 5% higher than the value of $524m$ observed by Fritz (2001). The maximum height location differs in $300m$ from the observed value. On the other hand, in the south slope the maximum water height observed was $208m$, while the PFEM result (not shown here) was $195m$ (6% error). More information on the PFEM solutions of this example are reported in Salazar et al. (2012, 2016).

6.8.4 Melting flow and flame spread of thermoplastics

In the next examples we show the possibilities of the PFEM to simulate the melting and spread of polymer objects under the effect of a heat source induced by a fire. The simulation reproduces a laboratory experiment in which a chair of polymeric material is exposed to an initial heat source. Degradation of the polymer decreases its viscosity by several orders of magnitude and produces combustible gases that increase the heat source when ignited. The problem was solved as a multi-fluid problem, simulating air as a Newtonian fluid and the chair as a high-viscosity fluid. The changing viscous polymer included gasification effects and a pseudo combustion between the air and the polymer when the temperature reaches the ignition temperature. Figure 12a shows

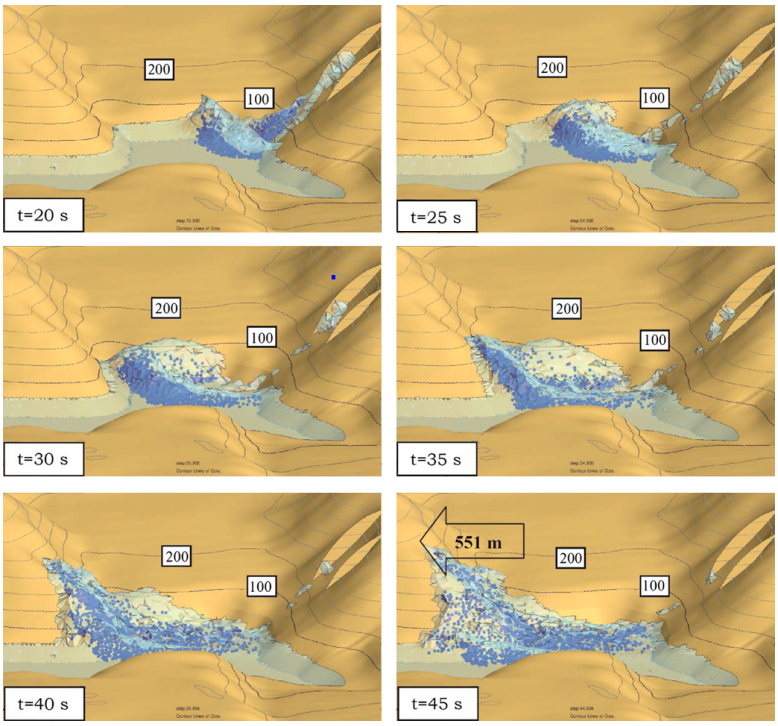


Figure 11: Lituya Bay landslide. Evolution of the landslide into the reservoir obtained with the PFEM. Maximum level of generated wave (551m) in the north slope.

the temperature distribution and the mesh used in one time step. A similar PFEM analysis of the burning of a rectangular polymer is displayed in Figure 12b.

Figure 13 shows another application of the PFEM to the melting and dripping of a polimer coating protecting three cables.

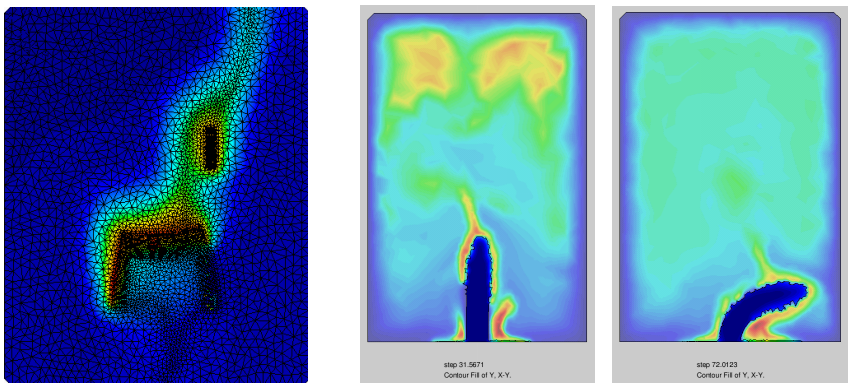


Figure 12: (a) Temperature distribution and instant mesh during the burning of a polymer chair. (b) Evolution of the burning, melting and dripping of a rectangular thermoplastic object in a close cavity.

6.9 Advantages and disadvantages of the standard PFEM

The PFEM shares the same advantages of all the particles methods; pure advection is treated exactly and when several materials are present, each is described by its own set of particles. The PFEM also shares the advantage using a finite element mesh with the MPM. Hence, reducing the mesh size in certain areas allows to improve the resolution limiting the increase in the computational cost. The FEM mesh also allows a better approximation of

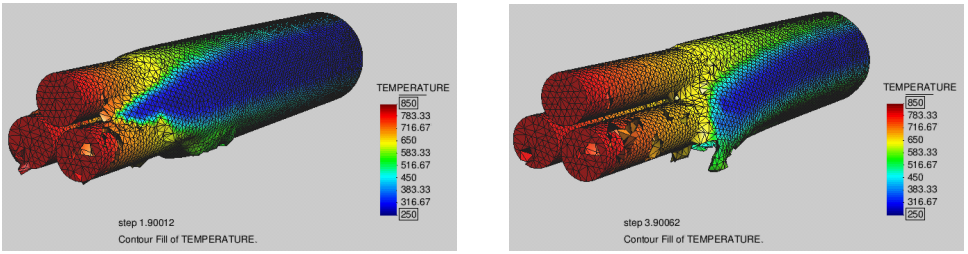


Figure 13: PFEM simulation of the melting and dripping of a polymer coating protecting three cables. Evolution of the 4-noded tetrahedra mesh discretizing the polymer domain for two instants of time

the unknowns, with greater precision in the definition of the stresses and the forces. In many meshless particle methods the pressure results are useless due to the magnitude of the oscillations in the solution.

Compared to the MPM, the main advantage of the PFEM is the way to evaluate the mass conservation with an implicit time integration. The fact that each particle is not assigned with a fixed mass is an important advantage when the number of particles changes due to mesh refinement. In the PFEM, it is possible to add or to eliminate as many particles as necessary during the time integration without any modification in the total mass of the domain.

The main disadvantage of the PFEM is the permanent need for a mesh generation stage, which must be called in almost all the time steps. The mesh generation process does not only imply the computer time to generate the mesh itself, but also the creation of a new data structure, with possible new degrees of freedom, new matrices, new boundary conditions, etc. Furthermore, the other constraint of PFEM is the same as observed in all the other previously described particle methods: the need of small time-steps to obtain reasonable results.

7 The Particle Finite Element Method second generation (PFEM-2)

7.1 General formulation

The particle finite element method second generation (PFEM-2) is a particle-based strategy combining the characteristics of moving and fixed mesh methods. The PFEM-2 is based in a Lagrangian framework, making use of moving particles to convect the material and physical properties of the fluid, but also utilizing a fixed finite element mesh to solve the equations, in a similar fashion to the MPM. In order to derive the PFEM-2, let us start with the momentum equations written in the Lagrangian form:

$$\rho \frac{D\mathbf{V}}{Dt} = \nabla . \boldsymbol{\sigma} + \rho \mathbf{b} \qquad \text{in } \Omega \tag{61}$$

Equation (61) is solved by evaluating the terms in a limited number of spatial configurations. For first order time integrators, the equation is only evaluated in a single configuration, either in the previous state or in the new configuration (forward Euler and backward Euler respectively). On the other hand, the Crank-Nicholson scheme uses information from both configurations for a better accuracy. The problem with equation (61) is that the intermediate configurations are not used. While this is not a limitation when the time steps are small and, therefore, the forces do not change significantly, it can lead to severe errors when the configuration changes significantly. The goal of the PFEM-2 is to improve the approximation in the intermediate configurations. To do so, the equations are written in an integral form in time for a material particle with position $\mathbf{x}_p$, as

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^t(\mathbf{x}_p^t) dt \tag{62a}$$

$$\mathbf{V}_p^{n+1} = \mathbf{V}_p^n + \int_n^{n+1} \mathbf{a}^t(\mathbf{x}_p^t) dt \tag{62b}$$

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

The advantage of the form (62) is that no time integration errors are introduced, as long as the acceleration and velocity vectors are known exactly. Unfortunately equations (62) cannot be solved directly in the way they are presented. On one hand they are linked together since one is the derivative of the other. The convection of the particles, defined by (62a) depends on the velocity at every instant, while the forces (accelerations) of the momentum equation in (62b) depend on the position of each particle. On the other hand, the need for a time discretization to solve the equations implies that the variables are not known for any arbitrary time t, but only at discrete time steps $n-1$, n, $n+1$, etc.

Following this line, the convection of the material points in Equation (62a) can be approximated using information from the current and previous time steps. Defining the parameter $\theta = 0$ for a fully explicit scheme and $\theta = 1$ for a fully implicit one, the convection equation (62a) reads

$$\mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + (1-\theta)\int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t)dt + \theta \int_n^{n+1} \mathbf{V}^{n+1}(\mathbf{x}_p^t)dt \tag{63}$$

The problem with Equation (63) is that the new velocity depends on the new configuration (i.e. on the new position of the particle), and, hence, this equation is strongly coupled to the momentum equations. By using an explicit strategy ($\theta = 0$) and Lagrangian particles, the convection stage reduces to simply moving the particles across the domain with a known velocity. Note that in the following equation the position of the particle ($\mathbf{x}_p^t$) is computed approximately, by using explicit information as

$$\mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t)dt \tag{64}$$

Idelsohn et al. (2013) applied the same explicit strategy for the momentum equation (62b). The resulting method was termed PFEM-2 without implicit viscous correction, which showed good accuracy for low viscosity problems. Unfortunately, this strategy cannot be used for all problems. When the acting forces are large compared to the inertia of the material points, explicit algorithms yield unstable schemes. The stability of such schemes can be evaluated by the Fourier number. For example, for a diffusion problem with a mesh size h, a time step Δt and a diffusion coefficient D, the limit condition for stable explicit schemes is

$$Fo = \frac{D\Delta t}{h^2} < \frac{1}{2} \tag{65}$$

Failing to fulfill this condition leads to unstable algorithms. On the other hand, implicit or semi-implicit algorithms can be unconditionally stable. Using again $\theta = 0$ for a fully explicit scheme and $\theta = 1$ for a fully implicit one, the momentum equation (62b) is expressed as

$$\mathbf{V}_p^{n+1} \approx \mathbf{V}_p^n + (1-\theta)\int_n^{n+1} \mathbf{a}^n(\mathbf{x}_p^t)dt + \theta \int_n^{n+1} \mathbf{a}^{n+1}(\mathbf{x}_p^t)dt \tag{66}$$

On the other hand, if the streamline integration is not used, the velocity in the new time step can be expressed as

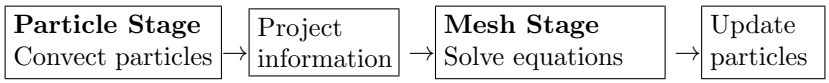
$$\mathbf{V}_p^{n+1} \approx \mathbf{V}_p^n + (1-\theta)\mathbf{a}^n(\mathbf{x}_p^n)\Delta t + \theta\ \mathbf{a}^{n+1}(\mathbf{x}_p^{n+1})\Delta t \tag{67}$$

Combining the explicit contribution of Equation (66) with the implicit term of equation (67), the general PFEM-2 algorithm with implicit correction is obtained. This simplifies the implementation of the method, as the implicit terms are added directly in the final configuration. Then the final set of equations including the convection terms reads:

$$\mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t)dt \tag{68a}$$

$$\mathbf{V}_p^{n+1} \approx \mathbf{V}_p^n + (1-\theta)\int_n^{n+1} \mathbf{a}^n(\mathbf{x}_p^t)dt + \theta\ \mathbf{a}^{n+1}(\mathbf{x}_p^{n+1})dt \tag{68b}$$

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



The particles used in this scheme do not represent a fixed amount of mass, but rather material points with certain properties and velocity. This allows us using a variable number of particles per element depending on the zone, to ensure a better accuracy on selected areas. Figure 14 shows the velocity streamline integration for a single particle.

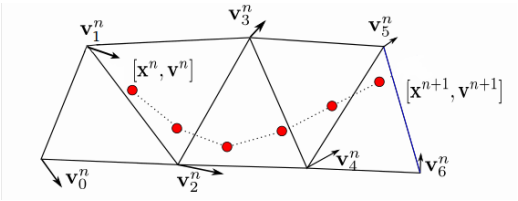


Figure 14: PFEM-2 streamline integration

As for any Lagrangian method, the equations to solve in the PFEM-2 do not include the convective term. This fact, combined with the streamline integration, allows us to use larger time steps compared to alternative Eulerian formulations. Idelsohn et al. (2012) presented accurate PFEM-2 results with Courant numbers higher than 10, that is, a particle moves across more than ten elements in a single time step.

7.2 Interface description

The linear shape functions used in the PFEM-2 are well suited to represent variables whose field is smooth. In other words, there can be no discontinuities on the variables or in their gradients. However, in multi-material simulations there can be sharp interfaces which translate into abrupt changes in the problems unknowns.

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

7.2.1 Interface capturing between two materials

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

Each particle has an associated material and the interface should be located where the material properties change. The distribution of particles inside each element can define complicated curves that become impossible to manage with simple shape functions due to the large number of particles. In Figure 15 some particles are in the "wrong" side of the interface. For this reason the instantaneous local interface inside each element is simply defined by a line (or a plane in 3D) taking into account a weighted average using the shape functions. Assuming that only two materials are involved (e.g. air and water), the lighter one is given a positive sign and the denser one a negative sign. In order to compute the location of the interface, a pseudo φ level set function is used, which defines the interface at the isosurface $\varphi = 0$. To calculate the value of φ at each j node, all the i particles in the neighbour elements to j are used. Next, the weighting function is defined as the standard shape function of the element for the node j in the position of each i particle. Once the contribution of all the particles has been added, finding the interface location is trivial using the following expression

$$\varphi_j = \frac{\sum_i^n \text{sign}_i N_i^j}{\sum_i^n N_i^j} \tag{69}$$

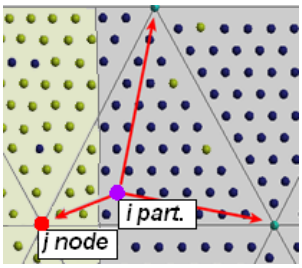


Figure 15: Contribution of i particle to j node

7.3 Improving the fields of unknowns: Enrichments

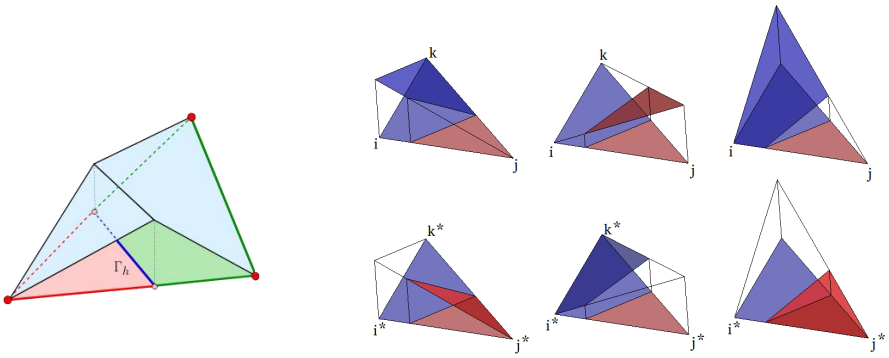
Standard FE formulations only allow for continuous interpolations of the variables inside the elements. However discontinuities in the material properties can cause jumps in the gradients of the variables and even complete discontinuities. The chosen strategy must take into account these special cases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

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

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

Gradient Discontinuities. For cases in which the material discontinuities only produce a kink in the gradients of the unknowns, it is necessary to add new shape functions that are continuous in the variables but discontinuous in the gradient. The most simple function that can fulfil this criteria is the one proposed in Coppola-Owen and Codina (2005) to improve the pressure field in two-fluid problems with low viscosities, such as mould casting simulations. In this case, the pressure is still continuous but with a steep change in the gradient, meaning that no discontinuities in this variable are needed. The addition of this function improves considerably mass conservation and the convergence of the solution. Figure 16(a) shows the gradient enrichment for the two-dimensional (2D) case. The concept is easily extended to 3D problems; the function is constructed by creating a continuous function whose gradient must be constant on each side of the interface.

Variables Discontinuities. When the variables can have sharp discontinuities at the interface, a gradient enrichment is not enough to capture the solution. In these cases more freedom must be given to the unknowns to obtain an accurate result (Ausas et al., 2012). Following this line, three enrichments shape functions plus special shape functions to replace the standard pressure field are used. Figure 16(b) shows the six functions used for triangles, with the goal of fully uncoupling the pressure from both sides. For 3D problems the concept is easily extrapolated, using a total of eight shape functions in split tetrahedra. To create this function, first the standard shape functions in the interface elements are replaced by their discontinuous counterparts. These functions are basically the same as the original ones, but the integrals are calculated only with the contribution of the partitions whose sign matches the sign of the node. On the other hand, when partitions and nodes have different signs, the contribution is added to the enrichments functions i^*, j^* or k^*. In other words, the original shape functions are split into two independent functions across the interface, allowing more freedom in the pressure field.



(a) Discontinuous gradient enrichment

(b) Discontinous Functions. Top: replacement functions. Bottom: enrichments

Figure 16: Enriched shape functions for the pressure field over a 3-noded triangle

7.3.1 Condensing the new degrees of freedom

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

Despite the number of unknowns is larger than in the original system, the associated computational cost to solve the sparse system would not be significantly larger. As stated previously, the main time increase would be the result of resizing the matrix and changing the graph of connectivities in this matrix. Condensing DOFs allows to circumvent this resizing. This process consists on eliminating the enrichments from the global system, by performing matrix operations and condensing the new DoFs. The reduced algebraic system is solved and then the condensed unknowns can be recovered.

7.4 PFEM-2 using a Fractional Step strategy

Recalling the fractional step (FS) strategy of Section 6.2 we can write the standard three solution steps for arbitrary values of θ as:

$$\text{A)} \quad \hat{\mathbf{V}} - \frac{\Delta t}{\rho}\nabla\cdot\left(\mu\theta\nabla\hat{\mathbf{V}}\right) = \mathbf{V}^n + \Delta t\ \mathbf{b} + \frac{\Delta t}{\rho}(1-\theta)\nabla\cdot(\mu\nabla\mathbf{V}^n) - \gamma\nabla p^n \tag{70a}$$

$$\text{B)} \quad \nabla^2(p^{n+1} - \gamma p^n) = \frac{\rho}{\Delta t}\nabla\cdot\hat{\mathbf{V}} \tag{70b}$$

$$\text{C)} \quad \mathbf{V}^{n+1} = \hat{\mathbf{V}} - \frac{\Delta t}{\rho}\nabla(p^{n+1} - \gamma p^n) \tag{70c}$$

Note that all the terms in the RHS of (70a) are explicit, which makes them good candidates for the explicit streamline integration. We define next:

$$\hat{\hat{\mathbf{V}}} = \mathbf{V}^n + \Delta t\ \mathbf{b} + \frac{\Delta t}{\rho}(1-\theta)\nabla\cdot(\mu\nabla\mathbf{V}^n) - \gamma\nabla p^n \tag{71}$$

This equation is solved by integrating along the streamlines. Hence, the FS strategy for the PFEM-2 becomes

$$0) \quad \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t) dt$$

$$\hat{\hat{\mathbf{V}}} = \mathbf{V}^n + \int_n^{n+1} \left(\mathbf{b} + \frac{1}{\rho}(1-\theta)\nabla \cdot (\mu \nabla \mathbf{V}^n) - \gamma \frac{1}{\rho} \nabla p^n \right) dt \tag{72a}$$

$$\text{A)} \quad \hat{\mathbf{V}} - \frac{\Delta t}{\rho} \nabla \cdot \left(\mu \theta \nabla \hat{\mathbf{V}} \right) = \hat{\hat{\mathbf{V}}} \tag{72b}$$

$$\text{B)} \quad \nabla^2 (p^{n+1} - \gamma p^n) = \frac{\rho}{\Delta t} \nabla \cdot \hat{\mathbf{V}} \tag{72c}$$

$$\text{C)} \quad \mathbf{V}^{n+1} = \hat{\mathbf{V}} - \frac{\Delta t}{\rho} \nabla (p^{n+1} - \gamma p^n) \tag{72d}$$

Finite Element discretization. Equations (72b) (72c) and (72d) are solved on a fixed Galerkin FE mesh. Using linear shape functions $\mathbf{N}$ for the velocity and pressure variables, steps A-B-C can be expressed in the following way, where the bars $(\bar{\cdot})$ denote the vectors of nodal unknowns:

$$\text{A)} \quad \mathbf{M}(\rho)\bar{\hat{\mathbf{V}}} + \mathbf{K}(\mu)\bar{\hat{\mathbf{V}}} = \mathbf{M}(\rho)\bar{\hat{\hat{\mathbf{V}}}} \tag{73a}$$

$$\text{B)} \quad \mathbf{L}(\frac{1}{\rho})\bar{\mathbf{p}}^{n+1} = \mathbf{L}(\frac{1}{\rho})\bar{\mathbf{p}}^n + \frac{1}{\Delta t}\mathbf{D}\bar{\hat{\mathbf{V}}} \tag{73b}$$

$$\text{C)} \quad \mathbf{M}(\rho)\bar{\mathbf{V}}^{n+1} = \mathbf{M}(\rho)\bar{\hat{\mathbf{V}}} - \Delta t \mathbf{G}(\bar{\mathbf{p}}^{n+1} - \gamma \bar{\mathbf{p}}^n) \tag{73c}$$

where the matrix components are:

$$\mathbf{M}(\rho)_{ij} = \int_\Omega \mathbf{N}_i \ \rho \ \mathbf{N}_j d\Omega \qquad \mathbf{K}(\mu)_{ij} = \int_\Omega \nabla \mathbf{N}_i \ \mu \ \nabla \mathbf{N}_j d\Omega$$

$$\mathbf{G}_{ij} = \int_\Omega \nabla \mathbf{N}_i \ \mathbf{N}_j d\Omega \qquad \mathbf{D}_{ij} = \int_\Omega \mathbf{N}_i \ \nabla \mathbf{N}_j d\Omega$$

Despite the momentum equations do not need any spatial stabilization in the Lagrangian formulation, Equation (73b) must be stabilized in space for equal order velocity - pressure formulations when $\gamma = 1$ is used. Here the Orthogonal Sub-grid Scale (OSS) approach has been chosen (Codina, 2002), which consists on using the projection on the pressure as a stabilization term. For the pressure stabilization and using ∇q as a weighting function, the OSS approach can be written in the following way, where π is the standard L^2 orthogonal projection:

$$0 = \tau \left(\nabla N, \boldsymbol{\pi}(\nabla p^{n+1}) - \nabla p^{n+1} \right)_\Omega \tag{74}$$

In matrix form, and using the explicit projection we can obtain

$$\tau \mathbf{L} \bar{\mathbf{p}}^{n+1} = -\tau \mathbf{G} \bar{\boldsymbol{\pi}}^n \tag{75}$$

Adding the terms of Equation (75) to the second step in Equation (73b), the stabilized form of the pressure equation becomes

$$\left[\mathbf{L}(\frac{1}{\rho}) + \tau \mathbf{L} \right] \bar{\mathbf{p}}^{n+1} = \mathbf{L}(\frac{1}{\rho})\bar{\mathbf{p}}^n + \frac{1}{\Delta t}\mathbf{D}\bar{\hat{\mathbf{V}}} - \tau \mathbf{G} \bar{\boldsymbol{\pi}}^n \tag{76}$$

Indeed any other stabilization method such as FIC procedure (Oñate, 2004; Oñate et al., 2014a) could have been used.

Computation of the explicit contributions. Starting with Step 0 of the FS method (equation (72a)), the terms $(1-\theta)\nabla \cdot (\mu \nabla \mathbf{V}^n)$ and $\gamma \nabla p^n$ in (72a) are approximated by a continuous field. Each ith component will be called g_i^n and π_i^n respectively, such that:

$$\mathbf{g}^n(\mathbf{x}) = \mathbf{N}(\mathbf{x})\bar{\mathbf{g}}^n \qquad \text{and} \qquad \boldsymbol{\pi}^n(\mathbf{x}) = \mathbf{N}(\mathbf{x})\bar{\boldsymbol{\pi}}^n \tag{77}$$

To evaluate $\bar{\mathbf{g}}^n$ and $\bar{\boldsymbol{\pi}}^n$ we use the same approximation as for the standard FEM, that is:

$$\int_\Omega \left[\mathbf{N}\ (1-\theta)\nabla\cdot(\mu\nabla\mathbf{V}^n) - \mathbf{N}\ \mathbf{g}^n\right] d\Omega = 0 \tag{78}$$

and

$$\int_\Omega \left[\mathbf{N}\ \gamma\nabla p^n - \mathbf{N}\ \boldsymbol{\pi}^n\right] = 0 \tag{79}$$

Integrating by parts the first term in both Equations (78) (79) and adding both equations, we obtain

$$\hat{\mathbf{g}} - \hat{\boldsymbol{\pi}} = \mathbf{M}^{-1}\left[(1-\theta)\mathbf{K}(\mu)\bar{\mathbf{V}}^n - \gamma\mathbf{G}\bar{\mathbf{p}}^n\right] \tag{80}$$

PFEM-2 algorithm. Having obtained the expression for the explicit terms and the three steps to be solved in the mesh, the PFEM-2 algorithm can be expressed as follows:

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

$$\hat{\bar{\mathbf{V}}}_p = \mathbf{V}^n + \frac{1}{\rho_p}\int_n^{n+1}\left[\mathbf{b} + \mathbf{N}(\mathbf{x}_p^t)(\bar{\mathbf{g}}^n - \bar{\boldsymbol{\pi}}^n)\right] dt \tag{81a}$$

1) Project information into mesh

2) A) $$\mathbf{M}(\rho)\bar{\hat{\mathbf{V}}} + \mathbf{K}(\mu)\bar{\hat{\mathbf{V}}} = \mathbf{M}(\rho)\bar{\hat{\bar{\mathbf{V}}}} \tag{81b}$$

B) $$\left[\mathbf{L}(\frac{1}{\rho}) + \tau\mathbf{L}\right]\bar{\mathbf{p}}^{n+1} = \mathbf{L}(\frac{1}{\rho})\bar{\mathbf{p}}^n + \frac{1}{\Delta t}\mathbf{D}\bar{\hat{\mathbf{V}}} - \tau\mathbf{G}\bar{\boldsymbol{\pi}}^n \tag{81c}$$

C) $$\mathbf{M}(\rho)\bar{\mathbf{V}}^{n+1} = \mathbf{M}(\rho)\bar{\hat{\mathbf{V}}} - \Delta t\mathbf{G}(\bar{\mathbf{p}}^{n+1} - \gamma\bar{\mathbf{p}}^n) \tag{81d}$$

3) Correct the particles' velocity with the new mesh velocity

$$\mathbf{V}_p^{n+1} = \hat{\bar{\mathbf{V}}}_p + \mathbf{N}(\mathbf{x}_p)(\bar{\mathbf{V}}^{n+1} - \bar{\hat{\bar{\mathbf{V}}}}) \tag{81e}$$

The above PFEM-2 algorithm with explicit contributions from both the pressure and viscous forces ($\gamma = 1$ and $\theta \neq 1$) has been successfully implemented for single fluid flows, with excellent accuracy even for very large time steps. Unfortunately, the explicit terms bear some limitations on the type of problems that can be solved. Besides the time step restriction to bound the Fourier number, the drawbacks associated to the explicit force integration become more evident in multi-fluid flows. In these cases the interface location changes constantly and since the densities of the phases are different, the integral (81a) can lead to extremely large non-physical velocities for low density particles that get into higher density/viscosity zones.

For these reasons, the streamline integration for the forces cannot be used in multi-fluid problems. Alternatively, a first order FS is used in these cases as the pressure field can be completely different from one step to the next. Unfortunately, the first order FS is only first order accurate (Codina and Badia, 2006). This can lead to unacceptable results in multi-fluids since severe mass losses can appear. To overcome this problem, it is possible to change the strategy in order to improve the accuracy towards a second order FS scheme. The idea consists on, instead of finishing the solution in the third step (81d), treat this velocity as the first step of an iterative process involving steps B and C, via a predictor-corrector scheme (Codina and Badia, 2006). This means taking at the end of each iteration $\hat{V}^{n+i+1} = V^{n+i}$ and repeat again steps B and C until convergence is found using $\delta p^{n+i+1} = p^{n+i+1} - p^{n+i}$.

Multi-fluids solved with the Fractional Step and PFEM-2. When there is a sharp change in the density, the hydrostatic condition under the gravitational force leads to two different values for the gradient of the pressure. This is specially critical when the fluids have very different densities (e.g. air-water). In these cases, the term $\nabla p \approx \rho g$ changes abruptly on the interface. Figure 17 shows the exact pressure distribution for the hydrostatic case of water and air and the one obtained by three linear elements. It is clear that the solution is very poor and, in practice, it would lead to an incorrect behaviour of the interface and mass losses (Coppola, 2009).

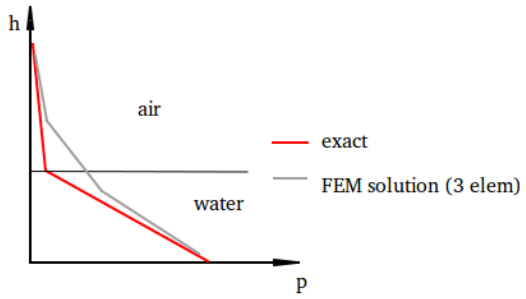


Figure 17: Pressure distribution for the hydrostatic case

To account for the gradient discontinuity, the shape function proposed in Coppola (2009) is used. Since the FS scheme is specially well suited for low viscosity problems and in these cases there can be no discontinuities in the pressure, the discontinuous shape functions are not used in this scheme. Then the new full enriched pressure system of for interface elements can be written in the following way

$$\begin{bmatrix} \mathbf{L}_{NN} & \mathbf{L}_{N*} \\ \mathbf{L}_{*N} & \mathbf{L}_{**} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{p}}_N \\ \mathbf{p}_* \end{bmatrix} = \begin{bmatrix} \mathbf{F}_N \\ \mathbf{F}_* \end{bmatrix} \tag{82}$$

The RHS is denoted as $\mathbf{F}$ and has contributions from the divergence of the velocity and the previous pressure field. To distinguish the different shape functions, sub-index N is used for the standard FE space and * for the enriched DoFs. Expanding the system and rearranging the equations, the enriched DoFs can be obtained. Note that the submatrix L_{**} must be invertible. Now, for a given iteration $i+1$, the enriched pressure is:

$$\mathbf{p}_*^{i+1} = {\mathbf{L}_{**}}^{-1}(\mathbf{F}_* - \mathbf{L}_{*i}\ \bar{\mathbf{p}}_N^{i+1}) \tag{83}$$

Finally, replacing equation (83) into (82), the new unknowns are condensed and the system recovers the same number of unknowns it originally had. It must be noted that after the pressure system has been solved, the enriched pressure field $p*$ should be recovered using (83) in order to calculate the gradient of the pressure correctly in the last step of the segregated strategy. The system to be solved with the condensed DOFs becomes:

$$\tilde{\mathbf{L}}\bar{\mathbf{p}}^{i+1} = \tilde{\mathbf{F}} \tag{84}$$

where

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

Writing all the terms that account for $\mathbf{F}^*$, the updated enriched pressure field becomes.

$$\mathbf{p}_*^{i+1} = \mathbf{L}_{**}^{-1}\left[\left(\mathbf{L}_{**}p_n^* + \mathbf{G}_*\hat{\mathbf{V}}^{i+1}\right) - \mathbf{L}_{N,*}\left(\delta\bar{\mathbf{p}}^{i+1}\right)\right] \tag{85}$$

Note that the formula includes the enriched pressures from the previous iteration.

Finally, having calculated the updated standard pressures with (84) and the updated enriched pressures with (85), the correction step (step 3) of the interface element becomes

$$\mathbf{M}\ \mathbf{V}^{i+1} \ = \ \mathbf{M}\ \hat{\mathbf{V}}^{i+1} - \Delta t\left(\mathbf{G}_N\delta\bar{\mathbf{p}}_N^{i+1} + \mathbf{G}_*\delta\mathbf{p}_*^{i+1}\right) \tag{86}$$

Then, the complete PFEM-2 algorithm for multi-fluids and no explicit streamline integration becomes

$$0) \quad \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t)dt \tag{87a}$$

1) Project information into mesh

$$2)\ \text{A}) \quad \mathbf{M}(\rho)\bar{\tilde{\mathbf{V}}} + \mathbf{K}(\mu)\bar{\tilde{\mathbf{V}}} = \mathbf{M}(\rho)\bar{\hat{\tilde{\mathbf{V}}}} \tag{87b}$$

$$\text{B}) \quad \tilde{\mathbf{L}}\left(\frac{1}{\rho}\right)\bar{\mathbf{p}}^{n+i+1} = \mathbf{G}\bar{\tilde{\mathbf{V}}}^{n+1} + \tilde{\mathbf{L}}\left(\frac{1}{\rho}\right)\bar{\mathbf{p}}^{n+i} \tag{87c}$$

$$\text{C}) \quad \mathbf{p}_*^{i+1} = \mathbf{L}_{**}^{-1}\left[\left(\mathbf{L}_{**}\mathbf{p}_n^* + \mathbf{G}_*\bar{\tilde{\mathbf{V}}}^{i+1}\right) - \mathbf{L}_{N,*}\left(\delta\bar{\mathbf{p}}^{i+1}\right)\right] \tag{87d}$$

$$\text{D}) \quad \mathbf{M}(\rho)\bar{\mathbf{V}}^{n+1} = \mathbf{M}(\rho)\bar{\tilde{\mathbf{V}}} - \Delta t[\mathbf{G}\delta\bar{\mathbf{p}}^{n+i+1} + \mathbf{G}_*\delta\mathbf{p}_*^{i+1}] \tag{87e}$$

set $\bar{\tilde{\mathbf{V}}}^{n+1} = \bar{\mathbf{V}}^{n+1+i}$ and go to step 2B) until convergence

3) Correct particles' velocity with $\mathbf{N}(\mathbf{x}_p)(\bar{\mathbf{V}}^{n+1} - \bar{\hat{\tilde{\mathbf{V}}}})$

7.5 PFEM-2 using a Monolithic strategy

When the viscosity of the fluids plays a dominant role in the simulation, the FS scheme is no longer a good alternative. The assumption used to develop the strategy do not hold and, therefore, the results are not correct using a single fractional velocity iteration. Despite it would be possible to achieve convergence by iterating all the steps of the strategy, it would no longer provide a fast algorithm, which will jeopardize the original reason why the fractional step PFEM-2 was developed for. In these high viscosity cases, the monolithic scheme is a better alternative. The method consists on solving the pressure and velocity equations together, meaning that no simplifications are made to solve the system. The solution steps are the same as in (87), replacing only the FS step strategy (equations (87b)–(87e)) with the monolithic method presented below.

Setting $\theta = 1$ and using the Galerkin FEM discretization, the system in matrix form (5) becomes (88). The monolithic problem (pressure and velocities together as unknowns) is the set of equation that has to be assembled in the fixed FE mesh. The gradient of the pressure has been integrated by parts in order to obtain a symmetric system as

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{\mathbf{M}(\rho)}{\Delta t} & \mathbf{D}^T \\ \mathbf{D} & 0 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{V}}^{n+1} \\ \bar{\mathbf{p}}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}(\bar{\mathbf{V}}^n, \mathbf{g}) \\ 0 \end{bmatrix} \tag{88}$$

where the matrix components are:

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

A FEM discretization directly implemented in the form (88) using the same shape functions for the velocity and the pressure would be unstable in the pressure since it does not fullfil the inf-sup condition (Codina and Badia, 2006).

Stabilization of the pressure equation via a PSPG method. A simple way to stabilize the pressure equation while modifying as little as possible the solution is the Pressure-Stabilizing/Petrov-Galerkin (PSPG) method (Tezduyar, 1992). It consists on adding a perturbation term multiplied by the residual of the momentum equation (first equation in (5)) as:

$$0 = \tau\left(\nabla q, \rho\frac{(\mathbf{V}^{n+1} - \mathbf{V}^n)}{\Delta t} - \nabla\left[2\mu\left(\nabla^s\mathbf{V}^{n+1}\right)\right] + \nabla p^{n+1} - \rho\mathbf{g}\right)_\Omega \tag{89}$$

Since the elements used are linear it is not possible to integrate by parts Equation (89) because the gradient of the test function is being used. Hence, the viscosity term vanishes due to the second derivatives. Now adding

the perturbation to the unstabilized original weak form, dividing by the density to obtain consistent units and assuming $\frac{|V^{n+1}-V^n|}{\Delta t} \ll g$, the stabilized system becomes:

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{1}{\Delta t}\mathbf{M}(\rho) & \mathbf{D}^T \\ \mathbf{D} & \tau L(\frac{1}{\rho}) \end{bmatrix} \begin{bmatrix} \bar{\mathbf{V}}^{n+1} \\ \bar{\mathbf{p}}^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}\bar{\mathbf{V}}^n + \mathbf{M}\mathbf{g} \\ \tau \mathbf{D}^T \mathbf{g} \end{bmatrix} \tag{90}$$

An alternative stabilization approach can be derived using the Orthogonal sub-grid scale (OSS) technique (Codina, 2002) or the FIC procedure (Oñate, 2004; Oñate et al., 2014a). Details can be found in these publications.

Pressure enrichments on the interface. Discontinuities in the pressure can appear when there is a jump in the viscosity. For this reason the full set of discontinuous shape functions will be used for the pressure field, as shown in Figure 16(b). Recalling the notation N for the standard shape functions and $*$ for the enrichments, the extended system has the form shown in equation (91). The equations are written with the PSPG stabilization, but the procedure is exactly the same for any other stabilization method (OSS, FIC, etc.)

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{1}{\Delta t}\mathbf{M}(\rho) & \mathbf{D}^T & \mathbf{D}_*^T \\ \mathbf{D}_N & \tau \mathbf{L}(\frac{1}{\rho})_{NN} & \tau \mathbf{L}(\frac{1}{\rho})_{N*} \\ \mathbf{D}_* & \tau \mathbf{L}(\frac{1}{\rho})_{*N} & \tau \mathbf{L}(\frac{1}{\rho})_{**} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{V}}^{n+1} \\ \bar{\mathbf{p}}_N^{n+1} \\ \mathbf{p}_*^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}\bar{\mathbf{V}}^n + \mathbf{M}\mathbf{g} \\ \tau {\mathbf{D}_N}^T \mathbf{g} \\ \tau {\mathbf{D}_*}^T \mathbf{g} \end{bmatrix} \tag{91}$$

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

$$\begin{bmatrix} \mathbf{K}(\mu) + \frac{1}{\Delta t}\mathbf{M}(\rho) - \frac{1}{\tau}\mathbf{D}_*\mathbf{L}_{**}^{-1}{\mathbf{D}_*}^T & {\mathbf{D}_N}^T - \mathbf{L}_{*N}\mathbf{L}_{**}^{-1}{\mathbf{D}_*}^T \\ \mathbf{D}_N - \mathbf{L}_{N*}\mathbf{L}_{**}^{-1}\mathbf{D}_* & \tau\left(\mathbf{L}_{NN} - \mathbf{L}_{*N}\mathbf{L}_{**}^{-1}\mathbf{L}_{N*}\right) \end{bmatrix} \begin{bmatrix} \bar{\mathbf{V}}^{n+1} \\ \bar{\mathbf{p}}_N^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}\bar{\mathbf{V}}^n + \mathbf{M}\mathbf{g} - \mathbf{D}_*\mathbf{L}_{**}^{-1}{\mathbf{D}_*}^T\mathbf{g} \\ \tau\left({\mathbf{D}_N}^T\mathbf{g} - \mathbf{L}_{*N}\mathbf{L}_{**}^{-1}{\mathbf{D}_*}^T\mathbf{g}\right) \end{bmatrix} \tag{92}$$

Unlike the FS scheme, the system (92) has the advantage of not requiring an additional step to recover the pressure, thus simplifying the algorithm. Implemented in this way, accurate results are possible by solving the system only once, unlike the segregated scheme described in Section 7.4, which requires at least two or three iterations of the pressure system to avoid severe mass losses.

7.6 Some results using the PFEM-2

7.6.1 Flow around an airfoil

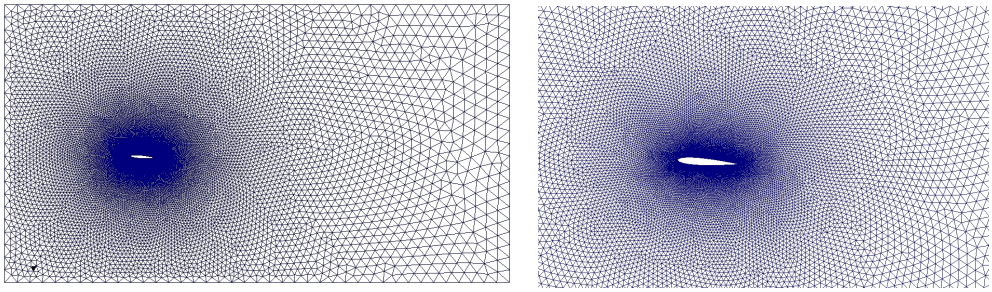
The flow around a NACA 0012 airfoil is a classical benchmark example due to the large amount of available experimental data. The angle of attach is set to 4° and a Reynolds number of 10,000 is used. No turbulence model was included in the simulation. The airfoil chord is of unit length and the computational domain is large in order not to interfere the boundary conditions with the airfoil. The computational domain and a detail of the mesh for the latter case are shown in Figure 18.

In Figure 19(a) the velocity contours are shown for the PFEM-2 using explicit integration on the streamlines for both the pressure and viscous forces. In Figure 19(b) the pressure coefficient for the lower and upper surface is plotted, with good matching with the results of Srinath and Mittal (2010).

7.6.2 Dam break with obstacle

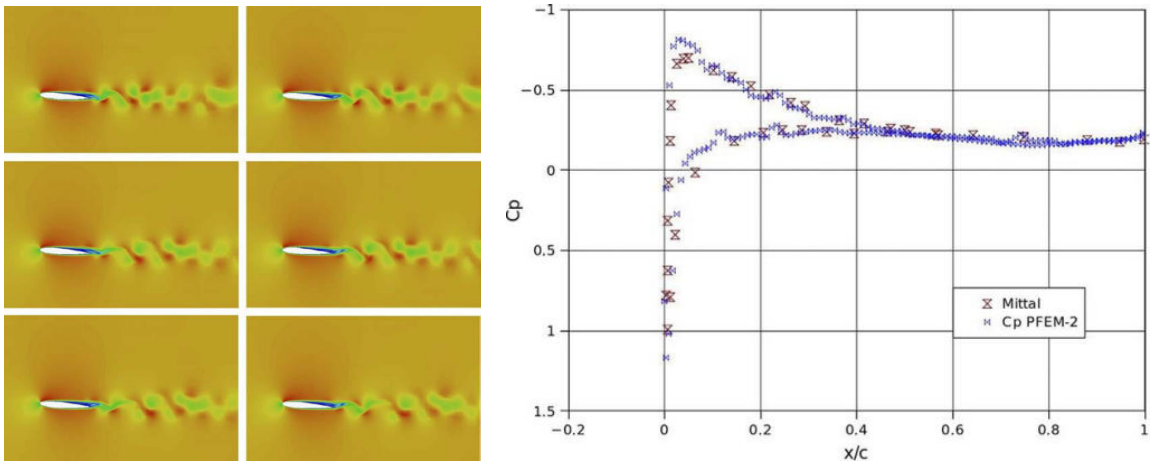
Following the experiment proposed by Koshizuka et al. (1995), numerical simulations were carried out to test the accuracy of the strategy. The domain is a closed box. The water column is held by a wall that is suddenly removed and the flow begins. The initial configuration can be seen in Figure 20.

Figure 21(a) shows the comparison between the real experiment and the simulation. The results are qualitatively accurate with small computation times. It is interesting to note that, despite not being visible, the air



(a) Computational domain (b) Mesh detail

Figure 18: NACA 0012 geometry



(a) Velocity contour snapshots (b) Pressure coefficient distribution

Figure 19: NACA 0012 results

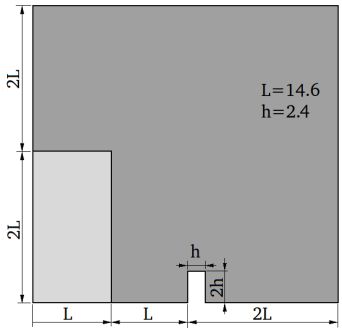
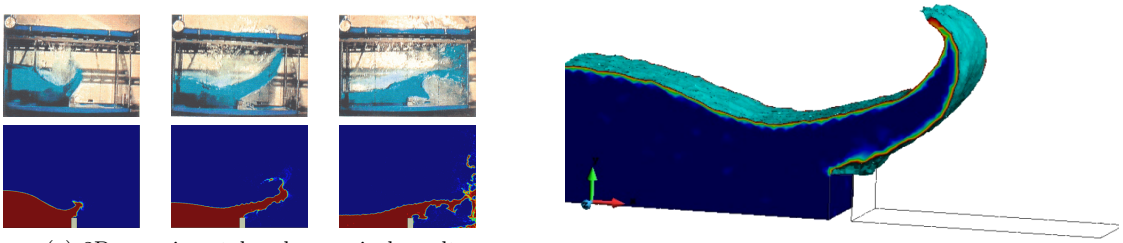


Figure 20: Dam Break experiment configuration

domain is also calculated with the aid of the enrichment functions for the air-water interface. This allows for the simulation of the interaction between the two fluids, and allows us to reproduce the trapped air that holds the water jet in the last frame of Figure 21(a). The same example was also run in 3D, as seen in figure 21(b).

7.7 Catamaran with spray and mesh deformation

The use of numerical methods to simulate ships allows to refine hull geometries and predict possible design problems, without the costs and time associated with experimental modelling (Oñate et al.). The problem of a powerboat cruising over waves is presented to test the code in a non academic example. The geometry of the



(a) 2D experimental and numerical results

(b) 3D simulation snapshot

Figure 21: Dam break problem results

boat is shown in Figure 22. It consists on a flexible twin hull design (catamaran) joined by a link, based on the WAM-V concept (WAM-V, 2016; Nadukandi et al., (accepted) . The goal of the simulation was assessing the visibility at high velocities. The dimensions are shown in the same figure. The use of the PFEM-2 strategy allows us to represent water volumes that are below the mesh scale, by just plotting individual water particles moving through the air. To improve the behaviour of these particles, a parabolic trajectory is implemented to replace the streamline integration. The convection stage is performed this way until the particles are back on the water surface, when the streamline integration is recovered. On the other hand, the mesh is deformed at each time step to track the imposed movement of the boat. Since the PFEM-2 is a Lagrangian method, there is no need to modify the set of equations as an Arbitrary Lagrangian-Eulerian solver would require. Snapshots of the simulation of half of the geometry are shown in Figure 23.

A similar example solved with the PFEM-2 can be found in Oñate et al. (2015)

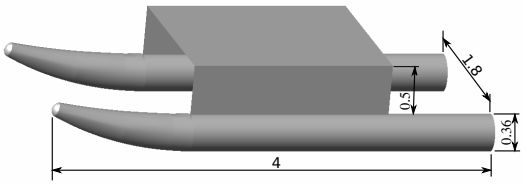


Figure 22: Catamaran Geometry

7.8 Advantages and disadvantages of the PFEM-2

In the same way as for the standard PFEM, the PFEM-2 has all the desirable properties of FEM strategies, such as a well defined field of variables and their derivatives. In the same way, the presence of a mass equation allows one to obtain the pressure as a primary variable, thus improving the accuracy on its value.

The main differences between the PFEM and PFEM-2 are the use of streamline integration and a fixed mesh in the latter. The streamline integration allows us to increase considerably the maximum time step for a given accuracy. This is a key feature of the PFEM-2 since, compared to explicit particle methods, that requires very small time steps to obtain accurate results. The use of a parallel solver for the implicit set of equations in the PFEM-2 will make this method competitive in terms of computing times versus explicit particle techniques. On the other hand, the fixed mesh overcomes the main limitation of the standard PFEM. Since the Delaunay tesselation is no longer required, the algorithm becomes trivially parallelizable with excellent scalability.

However, the computation of the momentum equations in the fixed mesh also leads to the main drawback of the PFEM-2 algorithm: the need for a projection stage. There is not a unique projection method and the different alternatives have associated errors, meaning the the field in the mesh will not exactly reproduce the Lagrangian particles field. In other words, every time the information is projected, the field is slightly degraded, adding a new error source compared to projection-free methods, such as the PFEM (Idelsohn et al.,

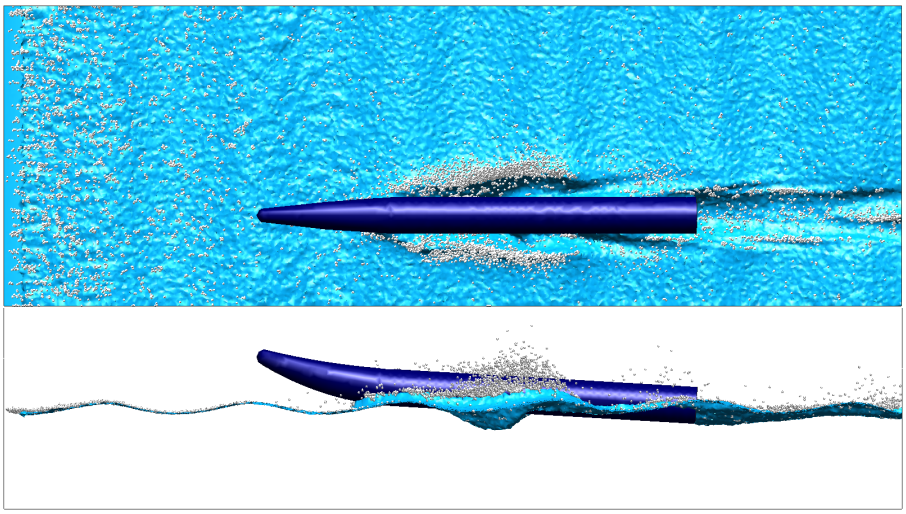


Figure 23: Catamaran Simulation snapshots at t=0.4s. Top and side views. Free surface and water particles.

2015). This projection error can be reduced using solution recovery techniques. Finally, the use of particles without associated mass also has disadvantages. Since the mass conservation depends only on the correct solution of the incompressibility equation accumulated errors can lead to mass losses when the time step is increased excessively.

8 Analysis of particulate flows using a DEM-FEM technique

8.1 Introduction

The study of fluid flows containing particles of different sizes (hereafter called *particulate flows*) is relevant to many areas of engineering and applied sciences. This type of flows is typical in slurry flows originated by natural hazards such as floods, tsunamis and landslides, as well as in many processes of the bio-medical and pharmaceutical industries, in the manufacturing industry and in the oil and gas industry (i.e. cuttings transport in boreholes (Celigueta et al., 2016)), among other applications (Anderson and Jackson, 1967; Avci and Wriggers, 2012; Clift et al., 1978; Gidaspow, 1994; Healy and Young, 2005; Hilton and Cleary, 2013; Jajcevic et al., 2013; Jackson, 2000; Kafui et al., 2002; Liu and Sun, 2002; Patankar and Joseph, 2001; El Shamy and Zeghal, 2005; Sommerfeld et al., 2008; Li et al., 2007; Zohdi, 2007b,a).

In this section we describe the modelling and simulation of free surface particulate quasi-incompressible flows containing particles of different sizes using either an Eulerian or a Lagrangian formulation. In the later case we will use the standard PFEM described in Section 6.1.

8.2 Modelling of micro, macro and large particles

Figure 24 shows a domain containing fluid and particles of different sizes. Particles are defined as *microscopic*, *macroscopic* and *large* according to their dimensions. Microscopic and macroscopic particles are assumed to have a cylindrical (in 2D) or spherical (in 3D) shape. These particles are modelled as rigid objects that undergo interaction forces computed using the physical contacts between a particle and its neighbours, as in the standard discrete element method (DEM) (Avci and Wriggers, 2012; Hilton and Cleary, 2013; Oñate and Rojek, 2004). Larger particles are discretized using a Lagrangian FEM mesh.

Extensive research on the DEM has been carried out in the last decades since the first ideas were presented by Cundall and Strack (1979). Much of the research efforts have been focused on the development of adequate DEM models for accurately reproducing the correct behaviour of non cohesive and cohesive granular assemblies and granular flows (Cleary, 2009; Oñate and Rojek, 2004; Tavarez and Plesha, 2007; Wellmann et al., 2008; Williams and O'Connor, 1999). In recent years the DEM has also been effectively applied to the study of multi-fracture and failure of geomaterials (soils and rocks), concrete, masonry and ceramic materials, among

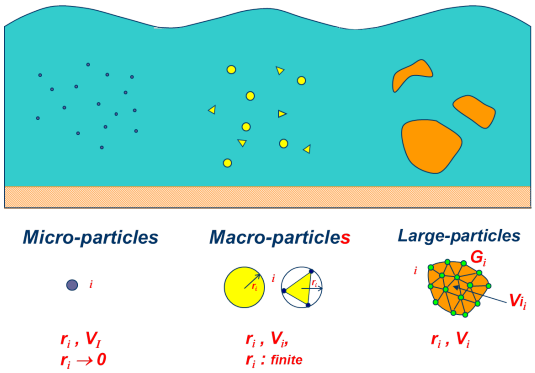
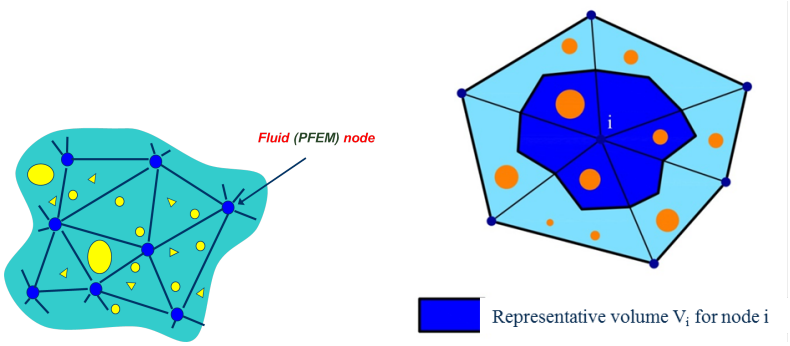


Figure 24: Microscopic, macroscopic and large particles within a fluid domain



(a) (b)

Figure 25: (a) Particles of different sizes surrounding the nodes in a FEM mesh. (b) Representative volume for a node (in shadowed darker colour)

others (Donzé et al., 2009; Fakhimi and Villegas, 2007; Hentz et al., 2004; Huang, 1999; Labra et al., 2008; Labra and Oñate, 2009; Mustoe, 1992; Oñate and Rojek, 2004; Oñate et al., 2015; Potyondy and Cundall, 2004; Rojek et al., 2012, 2001, 2008; Tavarez and Plesha, 2007; Tran et al., 2011).

Within the analysis of solids with the DEM, the material is typically represented as a collection of rigid particles interacting between each other at the contact interfaces in the normal and tangential directions. Material deformation is assumed to be concentrated at the contact points. Appropriate contact laws are defined to obtain the desired macroscopic material properties. The contact law can be seen as the formulation of the material model of the underlying continuum at the microscopic level. For frictional cohesive materials, the contact law takes into account the cohesive bonds between rigid particles. Cohesive bonds can be broken, thus allowing to simulate fracture of the material and its propagation.

A challenge in the failure analysis of solid materials with the DEM is the definition of the limit strengths in the normal and shear directions at the contact interfaces, and the characterization of the non linear relationship between forces and displacements at these interfaces beyond the onset of fracture, accounting for frictional effects, damage and plasticity. The non cohesive model considered in this section is a particular case of the more sophisticated cohesive model discussed in Oñate et al. (2015).

Microscopic and macroscopic particles are assumed to be spherical and submerged in the fluid (Figure 25). Fluid-to-particle forces are transferred to the particles via appropriate drag and buoyancy functions. Particle-to-fluid forces have equal magnitude and opposite direction as the fluid-to-particle ones and are transferred to the fluid points as an additional body force vector in the momentum equation (Figure 26). These equations, as well as the mass balance equation, account for the volume fraction of particles in the fluid, similarly as in standard immersed approaches for particulate flows (van Wachen, 2010; Wang et al., 2009; Zhang et al., 2004).

Large particles can have any arbitrary shape and they can be treated as rigid or deformable bodies. In the latter case, they are discretized with the standard FEM. The interface forces between the fluid and large

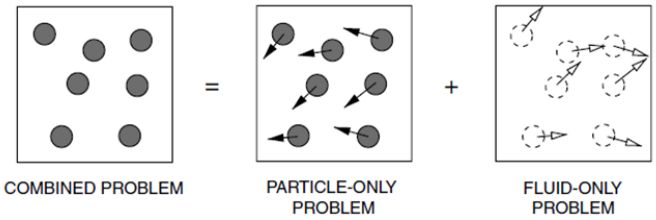


Figure 26: Immersed approach for treating the motion of physical particles in a fluid (Zohdi, 2007b)

particles and viceversa are computed using fluid-structure interaction (FSI) procedures (Oñate and García-Espinosa, 2001; Zienkiewicz et al., 2005b).

The following sections describe the basic governing equations for a Lagrangian particulate fluid and the computation of the forces for microscopic, macroscopic and large particles.

8.3 Basic governing equations for a particulate fluid

Conservation of linear momentum

The momentum equation expressed in residual form can be obtained by coupling the standard momentum balance equation for fluids (equation (3)) with the additional forces caused by the interaction with the particles. These equations can be written as

$$\mathbf{r}_m = \mathbf{0} \qquad \text{in } \Omega \quad \text{with} \quad \mathbf{r}_m := \rho_f \frac{D\mathbf{V}}{Dt} - \nabla \cdot \boldsymbol{\tau} + \nabla p - \left(\rho_f \mathbf{b} - \frac{1}{n_f} \mathbf{f}^{pf} \right) \tag{93}$$

In the expression of $\mathbf{r}_m$ equation(93), the new term $\mathbf{f}^{pf}$ is the averaged particle-to-fluid interaction force, for which closure relations must be provided and n_f is the fluid volume fraction at a point defined for each node j as

$$n_{f_j} = 1 - \frac{1}{V_j} \sum_{i=1}^{n_j} V_j^i \tag{94}$$

where V_j is the volume of the representative domain associated to a fluid node j, V_j^i is the volume of the ith particle belonging to V_j and n_j is the number of particles contained in V_j (Figure 24).

The fluid volume fraction n_f in equation(93) is a continuous function that is interpolated from the nodal values in the finite element fashion (Oñate, 2009; Zienkiewicz et al., 2005b,a).

The boundary conditions at the Dirichlet and Neumann boundaries coincide with equations(4).

Mass conservation equation

The residual form of the mass conservation equation for a compressible particulate flow is written as

$$r_v = 0 \qquad \text{in } \Omega \quad \text{with} \quad r_v := \frac{D(n_f \rho_f)}{Dt} + n_f \rho_f \varepsilon_v \tag{95}$$

Expanding the term in r_v and dividing by $n_f \rho_f$, the expression of r_v can be redefined as

$$r_v := -\frac{1}{\kappa} \frac{Dp}{Dt} + \frac{1}{n_f} \frac{Dn_f}{Dt} + \varepsilon_v \tag{96}$$

where $\kappa = \rho_f c^2$ is the bulk compressibility parameter with $c^2 = -\frac{\partial p}{\partial \rho}$, c being the speed of sound in the fluid.

Remark 1. For $n_f = 1$, no particles are contained in the fluid. Consequently, $f_i^{\rho f} = 0$ and the standard momentum and mass conservation equations for a viscous fluid are recovered (Donea and Huerta, 2003; Zienkiewicz et al., 2005b).

Remark 2. The time derivative term in equations (93) and (96) have different forms in Eulerian and Lagrangian frameworks (Section 2.1).

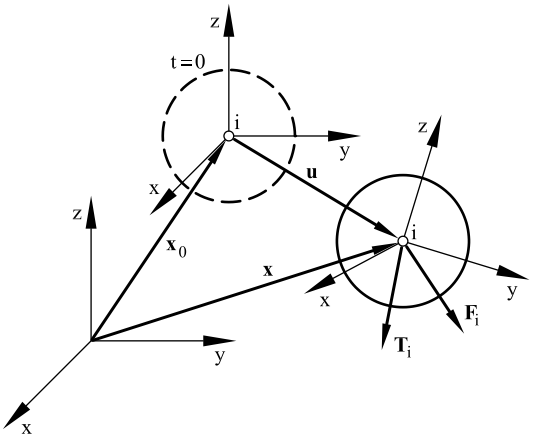


Figure 27: Motion of a rigid particle

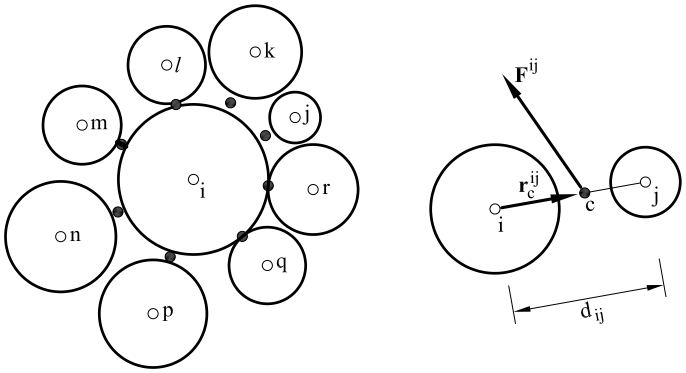


Figure 28: Force $\mathbf{F}^{ij}$ at the contact interface between particles i and j

8.4 Motion of micro and macro particles

The translational and rotational motion of rigid spherical or cylindrical particles is described by means of the standard equations of rigid body dynamics. For the i-th particle we have (Figure 27)

$$m_i \ddot{\mathbf{u}}_i = \mathbf{F}_i \,, \tag{97}$$

$$I_i \dot{\boldsymbol{\omega}}_i = \mathbf{T}_i \,, \tag{98}$$

where $\mathbf{u}_i$ is the particle centroid displacement in a fixed (inertial) coordinate frame $\mathbf{X}$, $\boldsymbol{\omega_i}$ – the angular velocity, m – the particle mass, I_i – the moment of inertia, $\mathbf{F}_i$ – the resultant force, and $\mathbf{T}_i$ – the resultant moment about the central axes (Figure 28).

Equations (97) and (98) are integrated in time using a standard central difference scheme (Oñate and Rojek, 2004; Zienkiewicz and Taylor, 2005). The time integration operator for the translational motion at the n-th time step is as follows:

$$\ddot{\mathbf{u}}_i^n = \frac{\mathbf{F}_i^n}{m_i} \,, \tag{99}$$

$$\dot{\mathbf{u}}_i^{n+1/2} = \dot{\mathbf{u}}_i^{n-1/2} + \ddot{\mathbf{u}}_i^n \Delta t \tag{100}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \Delta u_i \quad \text{with} \quad \Delta \mathbf{u}_i = \dot{\mathbf{u}}_i^{n+1/2} \Delta t \tag{101}$$

The first two steps in the integration scheme for the rotational motion are identical to equations(99) and (100):

$$\dot{\boldsymbol{\omega}}_i^n = \frac{\mathbf{T}_i^n}{I_i} \,, \tag{102}$$

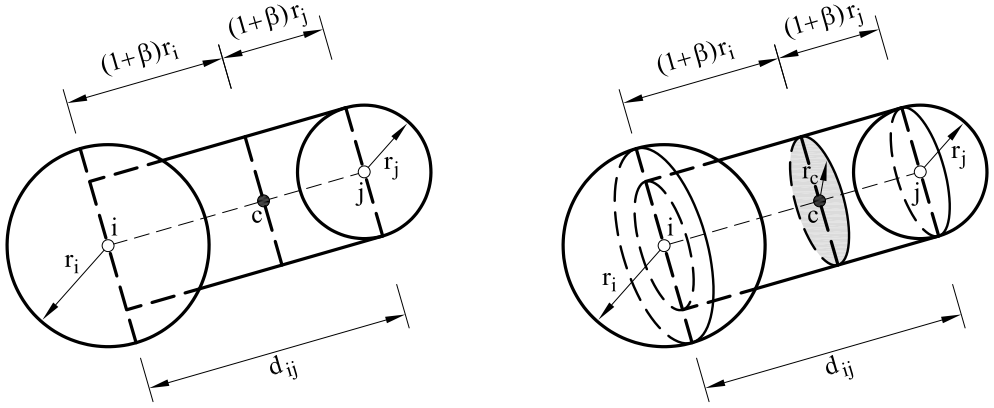


Figure 29: Definition of the contact interface.

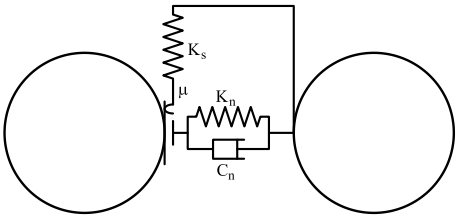


Figure 30: Model of the contact interface in the DEM

$$\boldsymbol{\omega}_i^{n+1/2} = \boldsymbol{\omega}_i^{n-1/2} + \dot{\boldsymbol{\omega}}_i^n \Delta t \tag{103}$$

The vector of incremental rotation $\Delta\boldsymbol{\theta} = [\Delta\theta_x, \Delta\theta_y, \Delta\theta_z]^{\mathrm{T}}$ is calculated for the ith particle as

$$\Delta\boldsymbol{\theta}_i = \boldsymbol{\omega}_i^{n+1/2} \Delta t \tag{104}$$

Knowledge of the incremental rotation suffices to update the tangential contact forces. It is also possible to track the rotational position of particles, if necessary. Then the rotation matrices between the moving frames embedded in the particles and the fixed global frame must be updated incrementally using an adequate multiplicative scheme (Oñate and Rojek, 2004; Zienkiewicz and Taylor, 2005).

Explicit integration in time yields high computational efficiency and it enables the solution of large models. The disadvantage of the explicit integration scheme is its conditional numerical stability imposing the limitation on the time step Δt (Zienkiewicz and Taylor, 2005).

8.4.1 Computation of forces on the particles

The forces $\mathbf{F}_i$ acting on the ith particle are computed as

$$\mathbf{F}_i = \mathbf{F}_i^w + \mathbf{F}_i^c + \mathbf{F}_i^{fp} \tag{105}$$

$\mathbf{F}_i^w$, $\mathbf{F}_i^c$ and $\mathbf{F}_i^{fp}$ are the forces on the particle due to self-weight, contact interactions between particles and fluid effects. These forces are computed as follows.

Self-weight forces

$$\mathbf{F}_i^w = -\rho_i \Omega_i \mathbf{g} \tag{106}$$

where ρ_i and Ω_i are the density and the volume of the ith particle, respectively and $\mathbf{g}$ is the gravity acceleration vector.

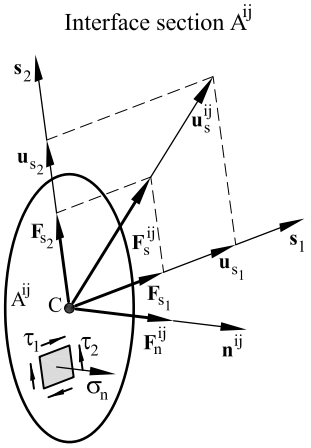


Figure 31: Acting forces and stresses

Contact forces

$$\mathbf{F}_i^c = \sum_{j=1}^{n_i} \mathbf{F}_{ij}^c \tag{107}$$

where n_i is the number of contact interfaces for the ith particle, and

$$\mathbf{F}_{ij}^c = \mathbf{F}_n^{ij} + \mathbf{F}_s^{ij} = F_n^{ij} n_i + \mathbf{F}_s^{ij} \tag{108}$$

where $\mathbf{F}_n^{ij}$ and $\mathbf{F}_s^{ij}$ are the normal and tangential forces acting at the ith interface connecting particles i and j. These forces are computed in terms of the relative motion of the interacting particles as in the standard DEM. The procedure to compute the normal and tangential forces is discussed below.

The normal force F_n at the contact interface between particles i and j (Figures 28 and 31) is obtained as

$$F_n^{ij} = \sigma_n \bar{A}^{ij} \tag{109}$$

where σ_n is the normal stress at the contact interface and $\bar{A}^{ij}$ is the *effective area* at the interface computed as

$$\bar{A}^{ij} = \alpha_{ij} A^{ij} \quad \text{with} \quad A^{ij} = \pi r_c^2 \tag{110}$$

In equation(110) α_{ij} is a parameter that accounts for the fact that the number of contacts and the packaging of particles are not optimal. Clearly $\alpha \simeq 1$ for quasi-optimal packaging distributions (Oñate et al., 2015).

Following the procedure described in Oñate et al. (2015) the relationship between the normal force and normal relative motion at the interface between particles i and j is found as

$$F_n^{ij} = K_n u_n + C_n \dot{u}_n \tag{111}$$

where u_n and $\dot{u}_n$ are the relative normal displacement and the relative normal velocity between the centroids of the interacting particles. equation(111) is assumed to hold in the elastic regime for both the normal tensile force F_{n_t} and the normal compressive force F_{n_c}. For non cohesive materials, the relationship (111) only applies for the normal compressive force. The normal stiffness (K_n) and damping (C_n) coefficients at the contact interface (Figure 30) are

$$K_n = \frac{\alpha_{ij} \pi r_c^2}{d_{ij}} E \ , \quad C_n = \frac{2\pi r_c \alpha_{ij} \xi}{d_{ij}} \sqrt{m_{ij} K_n} \tag{112}$$

The shear forces in the tangential directions s_1 and s_2 (Figure 31) are given by

$$F_{s_1} = \tau_1 \bar{A}^{ij} \quad , \quad F_{s_2} = \tau_2 \bar{A}^{ij} \tag{113}$$

where τ_1 and τ_2 are the shear stresses at the contact interface.

Following the procedure described in Oñate et al. (2015) we can find the relationship between the shear forces F_{s_1} and F_{s_2} and the relative tangential displacements u_{s_1} and u_{s_2} as

$$F_{s_1} = K_{s_1} u_{s_1} \qquad \text{and} \qquad F_{s_2} = K_{s_2} u_{s_2} \tag{114}$$

with

$$K_{s_1} = K_{s_2} = K_s = \frac{K_n}{2(1+\nu)} \tag{115}$$

where ν is the Poisson ratio.

A damping term is typically added to the contact forces in order to damp out the oscillations induced by the explicit dynamic scheme. For details see Oñate et al. (2015).

Fluid-to-particle forces: $\mathbf{F}_i^{fp} = \mathbf{F}_i^d + \mathbf{F}_i^b$, where $\mathbf{F}_i^b$ and $\mathbf{F}_i^d$ are, respectively, the buoyancy and drag forces on the ith particle. These forces are computed as:

Buoyancy forces

$$\mathbf{F}_i^b = -\Omega_i \boldsymbol{\nabla} p \tag{116}$$

Drag forces

The drag force is defined as

$$\mathbf{F}_i^d = -F_i^d \hat{\mathbf{v}}_i^r \tag{117}$$

where

$$\hat{\mathbf{v}}_i^r = \frac{\mathbf{v}_i^r}{|\mathbf{v}_i^r|} \quad \text{with} \quad \mathbf{v}_i^r = \mathbf{u}_i - \mathbf{v}_i \tag{118}$$

is the relative velocity of the particle with respect to the fluid, with $\mathbf{v}_i$ being the velocity vector of the fluid point coinciding with the ith particle.

The computation of the drag force F_i^d is explained in the next section.

Computation of $\mathbf{f}^{pf}$

The force $\mathbf{f}^{pf}$ in the r.h.s. of the momentum equations (equation(93)) is computed for each particle as $\mathbf{f}^{pf} = -\mathbf{f}^{fp}$, where $\mathbf{f}^{fp}$ computed *at each node* in the fluid mesh from the particle drag forces $\mathbf{F}_i^d$ as

$$\mathbf{f}_j^{fp} = \frac{1}{V_j} \sum_{i=1}^{n_j} N_j(x_i) \mathbf{F}_i^d \quad , \quad j = 1, N \tag{119}$$

where $N_j(x_i)$ is the value of the shape function of node j at the position of the ith particle, using the contribution of the n_j particles in the neighbour elements of node j. A continuum distribution of $\mathbf{f}^{fp}$ is obtained by interpolating its nodal values over each element in the FEM fashion. The forces on the particles due to lift effects have been neglected in this work, although they can be accounted for as explained in Jackson (2000).

8.5 Motion of large particles

Large particles may be considered as rigid or deformable bodies. In the first case the motion follows the rules for rigid Lagrangian particles. The contact forces at the particle surface due to the adjacent interacting particles are computed using a frictional contact interface layer between particles, as in the standard PFEM (Section 6.7.2) (Idelsohn et al., 2004; Oñate et al., 2008, 2004).

The fluid forces on the particles are computed by integrating the tangential (viscous) and normal (pressure) forces at the edges of the fluid elements surrounding the particles. Large deformable particles behave as deformable bodies immersed in the fluid which are discretized via the standard FEM. Their motion can be followed using a staggered FSI scheme, or else by treating the deformable bodies and the fluid as a single continuum with different material properties (Idelsohn et al., 2008; Oñate et al., 2014b,a).

8.6 Discretization of the fluid equations

We discretize the analysis domain containing N_p particles into finite elements with n_e nodes in the standard manner leading to a mesh with a total number of N_e elements and N nodes. In our work we will choose simple 3-noded linear triangles ($n_e = 3$) for 2D problems and 4-noded tetrahedra ($n_e = 4$) for 3D problems with local linear shape functions N_i^e defined for each node i of element e. The velocity components, the weighting functions and the pressure are interpolated over the mesh in terms of their nodal values in the same manner using the linear shape functions (Belytschko et al., 2013; Oñate, 2009; Zienkiewicz et al., 2005a).

Remark 3. For Eulerian flow formulations or when equal order inter polation for the velocities and the pressure are used stabilization terms must be added to the governing balance equations. In our work we use a stabilized form of the balance equations obtained via the FIC technique (Oñate, 2004; Oñate et al., 2014b, 2006; Oñate and Rojek, 2004).

Using the standard FEM procedures and omitting the stabilization terms, the final matrix system has the following form

$$\mathbf{M}_0\dot{\bar{\mathbf{v}}} + \mathbf{K}\bar{\mathbf{v}} + \mathbf{D}\bar{\mathbf{p}} - \mathbf{f}_v = \mathbf{0} \tag{120a}$$

$$\mathbf{M}_1\dot{\bar{\mathbf{p}}} - \mathbf{D}^T\bar{\mathbf{v}} + (\mathbf{L} + \mathbf{M}_b)\bar{\mathbf{p}} - \mathbf{f}_p = \mathbf{0} \tag{120b}$$

The matrices and vectors in equations(120) can be found in Oñate et al. (2014a, 2006).

Incremental solution of the discretized equations for a particulate flow

We summarize the steps for the analysis of a particulate fluid containing micro/macro particles modelled as a rigid spheres interacting "a la DEM" as described in the previous sections.

Equations(120) are solved in time with an implicit Newton-Raphson type iterative scheme (Belytschko et al., 2013; Donea and Huerta, 2003; Zienkiewicz et al., 2005a,b). The basic steps within a time interval $[n, n+1]$ are:

- Generate a new mesh using the position of the fluid nodes at t_n. This step is only required for formulations using a moving mesh (such as PFEM).
- Initialize variables: $({}^{n+1}\mathbf{x}^1, {}^{n+1}\bar{\mathbf{v}}^1, {}^{n+1}\bar{\mathbf{p}}^1, {}^{n+1}n_f^i, {}^{n+1}\bar{\mathbf{r}}_m^1) \equiv \{{}^n\mathbf{x}, {}^n\bar{\mathbf{v}}, {}^n\bar{\mathbf{p}}, {}^n n_f, {}^n\bar{\mathbf{r}}_m\}$.

  Note that ${}^n(\cdot)^j$ denote values at time n for the jth iteration.
- Iteration loop: $k = 1, \cdots, NITER$.
  For each iteration.

Step 1. Compute the nodal velocity increments in the fluid $\Delta\bar{\mathbf{v}}$

From equation(120)a, we deduce

$${}^{n+1}\mathbf{H}_v^i\Delta\bar{\mathbf{v}} = -{}^{n+1}\bar{\mathbf{r}}_m^k \rightarrow \Delta\bar{\mathbf{v}} \tag{121a}$$

with the momentum residual $\bar{\mathbf{r}}_m$ and the iteration matrix $\mathbf{H}_v$ given by

$$\bar{\mathbf{r}}_m = \mathbf{M}_0\dot{\bar{\mathbf{v}}} + \mathbf{K}\bar{\mathbf{v}} + \mathbf{D}\bar{\mathbf{p}} - \mathbf{f}_v \quad , \quad \mathbf{H}_v = \frac{1}{\Delta t}\mathbf{M}_0 + \mathbf{K} + \mathbf{K}_v \tag{121b}$$

where $\mathbf{K}_v$ is a "bulk" stiffness matrix that accounts for the changes of the pressure due to the velocity. This matrix is important for the fast convergence, mass preservation and accuracy of the iterative solution (Franci et al., 2015; Oñate et al., 2014a).

Step 2. Update the velocities

$$\text{Fluid nodes: } {}^{n+1}\bar{\mathbf{v}}^{k+1} = {}^{n+1}\bar{\mathbf{v}}^k + \Delta\bar{\mathbf{v}} \tag{122a}$$

$$\text{Rigid particles: } \begin{cases} {}^{n+1/2}\dot{\mathbf{u}}_j = {}^{n-1/2}\dot{\mathbf{u}}_j + {}^n\ddot{\mathbf{u}}_j^{k+1}\Delta t \\ \dot{\mathbf{u}}_j = \frac{1}{m_j}{}^n\mathbf{F}_j^{k+1} \quad , \quad j = 1, N_p \end{cases} \tag{122b}$$

Step 3. Compute the nodal pressures in the fluid ${}^{n+1}\bar{\mathbf{p}}^{k+1}$

From equation(120b) we obtain

$$ {}^{n+1}\mathbf{H}_p^{i\,n+1}\bar{\mathbf{p}}^{k+1} = \frac{1}{\Delta t}\mathbf{M}_1{}^{n+1}\bar{\mathbf{p}}^{i} + \mathbf{D}^{T\,n+1}\bar{\mathbf{v}}^{k+1} + {}^{n+1}\bar{\mathbf{f}}_p^{i} \rightarrow {}^{n+1}\bar{\mathbf{p}}^{k+1} \tag{123a} $$

with

$$ \mathbf{H}_p = \frac{1}{\Delta t}\mathbf{M}_1 + \mathbf{L} + \mathbf{M}_b \tag{123b} $$

Step 4. Update the coordinates of the particles

$$ \text{Rigid particles: } \begin{cases} {}^{n+1}\mathbf{u}_i^{k+1} = {}^{n}\mathbf{u}_i^{k+1} + {}^{n+1/2}\dot{\mathbf{u}}_i^{k+1}\Delta t \\ {}^{n+1}\mathbf{x}_i^{k+1} = {}^{n}\mathbf{x}_i + {}^{n+1}\mathbf{u}_i^{k+1} \quad , \quad i = 1, N_p \end{cases} \tag{124a} $$

Step 5. Update the coordinate of the fluid nodes (for moving mesh algorithms only)

$$ \text{Fluid nodes: } {}^{n+1}\mathbf{x}_i^{k+1} = {}^{n+1}\mathbf{x}_i^{k} + \frac{1}{2}({}^{n+1}\bar{\mathbf{v}}_i^{k+1} + {}^{n}\bar{\mathbf{v}}_i)\Delta t \quad , \quad i = 1, N \tag{124b} $$

Step 6. Compute the fluid volume fraction for each node ${}^{n+1}n_{f_i}^{k+1}$ via equation(94)

Step 7. Compute forces and torques on particles: ${}^{n+1}\mathbf{F}_i^{k+1}, {}^{n+1}\mathbf{T}_i^{k+1}$, $i = 1, N_p$ (equations (97) and (98))

Step 8. Compute particle-to-fluid nodes: $({}^{n+1}\mathbf{f}_i^{pf})^{k+1} = -({}^{n+1}\mathbf{f}_i^{fp})^{k+1}$, $i = 1, N$ *with* $\mathbf{f}_i^{fp}$ (equation(119))

Step 9. Check convergence

Verify the following conditions:

$$ \begin{aligned} \|{}^{n+1}\bar{\mathbf{v}}^{k+1} - {}^{n+1}\bar{\mathbf{v}}^{k}\| \leq e_v \|{}^{n}\bar{\mathbf{v}}\| \\ \|{}^{n+1}\bar{\mathbf{p}}^{k+1} - {}^{n+1}\bar{\mathbf{p}}^{k}\| \leq e_p \|{}^{n}\bar{\mathbf{p}}\| \end{aligned} \tag{125} $$

where e_v and e_p are prescribed error norms for the nodal velocities and the nodal pressures, respectively. In the examples solved in this work we have set $e_v = e_p = 10^{-3}$.

If both conditions (125) are satisfied then make $n \leftarrow n + 1$ and proceed to the next time step.

Otherwise, make the iteration counter $k \leftarrow k + 1$ and repeat Steps 1–8.

Remark 4. The time step within a time interval $[n, n + 1]$ has been chosen as $\Delta t = \min\left(\frac{{}^{n}l^{e}_{\min}}{|{}^{n}\mathbf{v}|_{\max}}, \Delta t_b\right)$ where ${}^{n}l^{e}_{\min}$ is the minimum characteristic distance of all elements in the mesh, $|{}^{n}\mathbf{v}|_{\max}$ is the maximum value of the modulus of the velocity of all nodes in the mesh and Δt_b is the critical time step of all nodes adjacent to a solid boundary Oñate et al. (2014a).

Remark 5. The Eulerian and Lagrangian versions of the particulate flow formulation presented have been implemented in the open-source Kratos software platform (Kratos, 2015). The generation of the analysis data and the visualization of the results have been carried out using the GiD pre/postprocessing system (CIMNE, 2015).

8.7 Examples

8.7.1 Water pipe with particles sediment

Figure 32 shows an example of flow of a non-cohesive granular material modeled using spherical DEM particles. The water inside the pipe is simulated with an Eulerian FEM code.

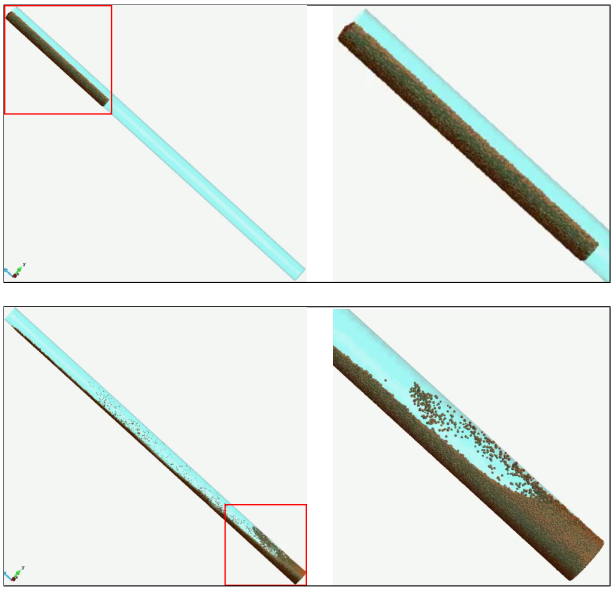


Figure 32: Interaction between particles and water in a pipe. Eulerian fluid solver coupled with DEM. Snapshots at two timesteps. Detail on the right images.

8.7.2 Filling of a water container with particles

Figure 33 shows a 3D example of the filling of a cylindrical container with water containing macroscopic spherical particles. Water is allowed to exit the cylinder by a lateral hole while particles enter from two other holes at different height and fall down by gravity until they progressively fill the cylinder. The figures show different instants of the filling process.

8.7.3 Transport of spherical particles in a tube filled with water

The example in Figure 34 models the vertical transport of 120.000 spherical particles to the surface of a tube filled with water and the subsequent deposition of the particles on the free water surface at the top of the tube. Particles move upwards within the tube due to a vertical fluid velocity of 1 m/s. The average particle radius is 2 mm and their density is 2300 Kg/m^3. Particles move vertically until they reach the top of the fluid domain and accumulate there due to the combined effect of their weight and the interaction force from the fluid. Figure 34 shows two instants of the particles ascending process.

Figure 35 shows the interaction of eight jets of ascending air bubbles with 200.000 spherical particles that fall down in a cylinder filled with water.

8.7.4 Dragging of large objects and small particles in a tsunami type flow

Figure 36 shows the dragging of a car by the impact of a water stream containing rocks.

The next example represents an schematic study of the dragging of some objects over a vertical wall, related to the tsunami in Fukushima (Japan) on March 2011 (Figure 37). Figures 38 show two snapshots of the PFEM solution of this complex problem using a simplified geometry including barrels and debris (modelled as macroscopic particles).

Figure 39 shows a similar analysis with the PFEM of the progressive erosion and failure of a soil retaining slope under the action of a water stream dragging cars and barrels.

9 Acknowledgements

This research was partially supported by the following Advanced Grants and Proof of Concepts projects of the European Research Council: REALTIME, SAFECON, FORECAST and FLOODSAFE. The People Programme

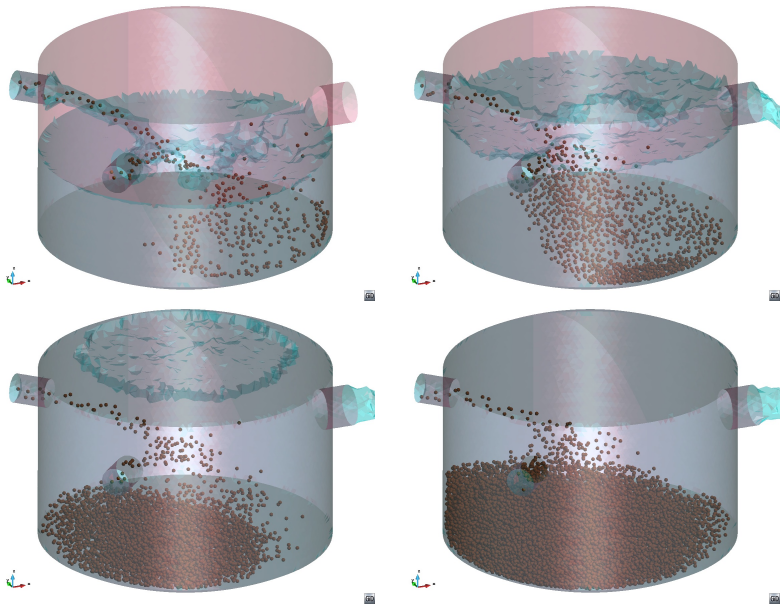


Figure 33: Filling of a container by injecting water containing macroscopic particles from two holes. Water is allowed to exit through a third hole at the upper right hand side of the cylinder. 3D PFEM results at four instants

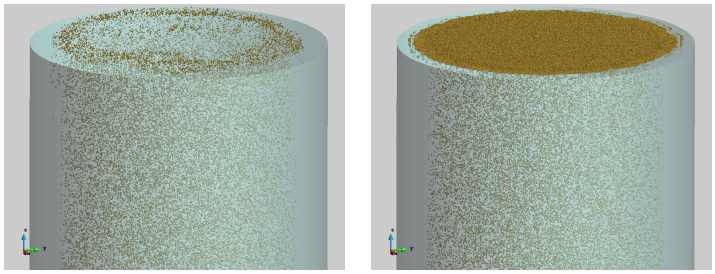


Figure 34: Transport of spherical macroparticles up to the free surface of a tube filled with water. Particles move up due to the upward water current until they accumulate on the free surface. Results obtained with a coupled DEM-Eulerian CFD code (Kratos, 2015)

(Marie Curie Actions) of the European Union's FP7/2007-2013/ under REA grant agreement n° 289911 also provided financial support.

Bibliography

F. A. Allahdadi, T. C. Carney, J. R. Hipp, L. D. Libersky, and A. G. Petschek. High strain lagrangian hydrodynamics: a three dimensional sph code for dynamic material response. Technical report, DTIC Document, 1993.

T. B. Anderson and R. Jackson. Fluid Mechanical Description of Fluidized Beds. Equations of Motion. *Industrial & Engineering Chemistry Fundamentals*, 6(4):527–539, nov 1967. ISSN 0196-4313. doi: 10.1021/i160024a007.

R. F. Ausas, G. C. Buscaglia, and S. R. Idelsohn. A new enrichment space for the treatment of discontinuous pressures in multi-fluid flows. *International Journal for Numerical Methods in Fluids*, 70(7):829–850, 2012.

B. Avci and P. Wriggers. A DEM-FEM Coupling Approach for the Direct Numerical Simulation of 3D Particulate Flows. *Journal of Applied Mechanics*, 79(1):010901, 2012. ISSN 00218936. doi: 10.1115/1.4005093.

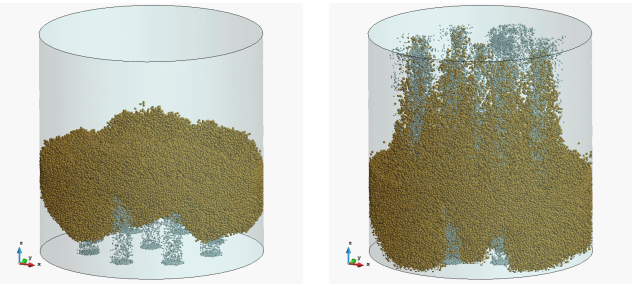


Figure 35: Interaction of eight jets of ascending air bubbles with a thick layer of 200.000 spherical particles falling within a cylinder filled with water. Results obtained with a coupled DEM-Eulerian CFD code (Kratos, 2015)

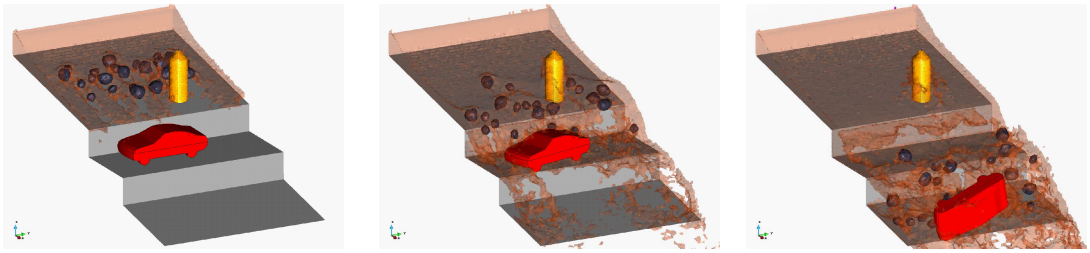


Figure 36: Dragging of a car by impact of a fluid flow containing rock particles. 3D PFEM results

T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary. *Nonlinear finite elements for continua and structures*. John Wiley & Sons, 2nd editio edition, 2013. ISBN 978-1-118-63270-3.

J. Brackbill and H. Ruppel. Flip: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics*, 65(2):314–343, 1986.

L. Brookshaw. A method of calculating radiative heat diffusion in particle simulations. In *Proceedings of the Astronomical Society of Australia*, volume 6, pages 207–210, 1985.

J. M. Carbonell, E. Oñate, and B. Suárez. Modeling of Ground Excavation with the Particle Finite-Element Method. *Journal of Engineering Mechanics*, 136(4):455–463, apr 2010. ISSN 0733-9399. doi: 10.1061/(ASCE)EM.1943-7889.0000086.

M. A. Celigueta, K. M. Deshpande, S. Latorre, and E. Oñate. A FEM-DEM technique for studying the motion of particles in non-Newtonian fluids. Application to the transport of drill cuttings in wellbores. *Computational Particle Mechanics*, 3(2):263–276, apr 2016. ISSN 2196-4378. doi: 10.1007/s40571-015-0090-3.

CIMNE. GiD. The personal pre/postprocessor. www.gidhome.com, 2015.

P. W. Cleary. Industrial particle flow modelling using discrete element method. *Engineering Computations*, 26 (6):698–743, aug 2009. ISSN 0264-4401. doi: 10.1108/02644400910975487.

R. Clift, J. R. Grace, and M. E. Weber, editors. *Bubbles, drops and particles*. Academic Press, New York, 1978.

R. Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering*, 191(39):4295–4321, 2002.

R. Codina and S. Badia. On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 195(23):2900–2918, 2006.

H. Coppola. *A finite element model for free surface and two fluid flows on fixed meshes*. PhD thesis, Universitat Politècnica de Catalunya, 2009.

Figure 37: Dragging of cars and large and small bodies over a wall in the Fukushima tsunami (Japan)

H. Coppola-Owen and R. Codina. Improving eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions. *International journal for numerical methods in fluids*, 49(12): 1287–1304, 2005.

P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29(1): 47–65, jan 1979. ISSN 0016-8505. doi: 10.1680/geot.1979.29.1.47.

P. Dadvand, R. Rossi, and E. Oñate. An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Archives of computational methods in engineering*, 17(3):253–297, 2010.

J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Wiley, Chichester, UK, apr 2003. ISBN 9780470013823. doi: 10.1002/0470013826.

F. Donzé, F. Richefeu, and S. Magnier. Advances in discrete element method applied to soil, rock and concrete mechanics. *Electronic Journal of Geotechology Engineering*, Bouquet08:1–44, 2009.

H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1): 43–72, jan 1994. ISSN 07300301. doi: 10.1145/174462.156635.

U. El Shamy and M. Zeghal. Coupled Continuum-Discrete Model for Saturated Granular Soils. *Journal of Engineering Mechanics*, 131(4):413–426, apr 2005. ISSN 0733-9399. doi: 10.1061/(ASCE)0733-9399(2005)131:4(413).

P. Espanol and M. Revenga. Smoothed dissipative particle dynamics. *Physical Review E*, 67(2):026705, 2003.

A. Fakhimi and T. Villegas. Application of Dimensional Analysis in Calibration of a Discrete Element Model for Rock Deformation and Fracture. *Rock Mechanics and Rock Engineering*, 40(2):193–211, apr 2007. ISSN 0723-2632. doi: 10.1007/s00603-006-0095-6.

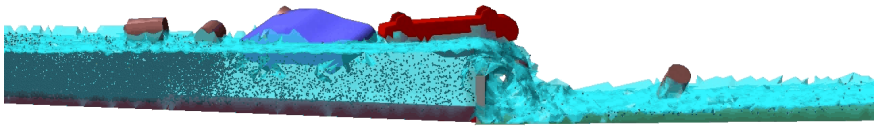
C. A. Felippa. Introduction to finite element methods. *University of Colorado, Boulder, http://www. colorado. edu/engineering/CAS/courses. d/IFEM. d*, 2004.

A. Franci, E. Oñate, and J. M. Carbonell. On the effect of the bulk tangent matrix in partitioned solution schemes for nearly incompressible fluids. *International Journal for Numerical Methods in Engineering*, 102 (3-4):257–277, dec 2015. ISSN 00295981. doi: 10.1002/nme.4839.

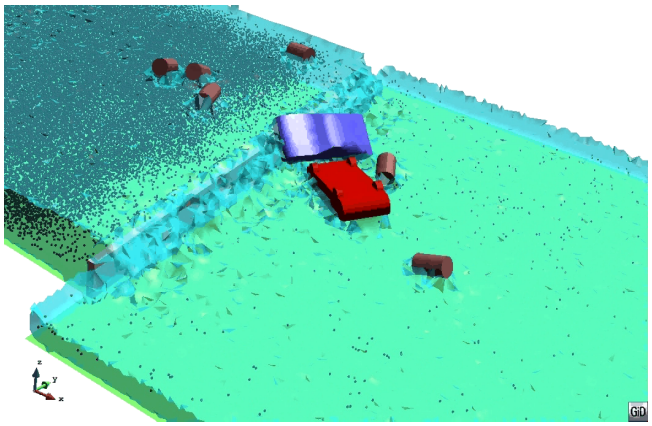
H. M. Fritz. Lituya bay case rockslide impact and wave run-up. *Science of Tsunami Hazards*, 19(1):3–22, 2001.

D. Gidaspow. *Multiphase flow and fluidization. Continuum and Kinetic Theory Description*. Academic Press, 1994.

R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.



(a)



(b)

Figure 38: Dragging of cars and barrels and macroscopic particles in a tsunami flow passing over a vertical wall. 3D PFEM results

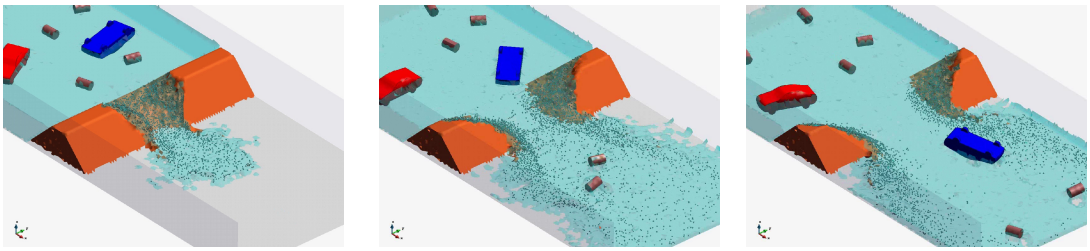


Figure 39: Motion of cars, barrels and eroded particles dragged in a fluid flow through a soil retaining slope. 3D PFEM results

D. Healy and J. Young. Full Lagrangian methods for calculating particle concentration fields in dilute gas-particle flows. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 461 (2059):2197–2225, jul 2005. ISSN 1364-5021. doi: 10.1098/rspa.2004.1413.

S. Hentz, L. Daudeville, and F. V. Donze. Identification and validation of a discrete element model for concrete. *Journal of Engineering Mechanics*, 130(6):709–719, 2004. doi: 10.1061/(ASCE)0733-9399(2004)130:6(709).

J. E. Hilton and P. W. Cleary. Dust modelling using a combined CFD and discrete element formulation. *International Journal for Numerical Methods in Fluids*, 72(5):528–549, jun 2013. ISSN 02712091. doi: 10.1002/fld.3750.

W. G. Hoover, C. Hoover, O. Kum, V. Castillo, H. Posch, and S. Hess. *Smooth particle applied mechanics.* World Scientific, 2006.

H. Huang. *Discrete element modeling of tool-rock interaction.* PhD thesis, University of Minnesota, 1999.

T. J. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. circumventing the babuška-brezzi condition: A stable petrov-galerkin formulation of the stokes problem

accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59(1): 85–99, 1986.

S. Idelsohn, J. Marti, A. Limache, E. Oñate, S. R. Idelshon, J. Marti, A. Limache, and E. Oñate. Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluidstructure interaction problems via the PFEM. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1762–1776, mar 2008. ISSN 00457825. doi: 10.1016/j.cma.2007.06.004.

S. Idelsohn, N. Nigro, A. Limache, and E. Oñate. Large time-step explicit integration method for solving problems with dominant convection. *Computer Methods in Applied Mechanics and Engineering*, 217:168–185, 2012.

S. Idelsohn, N. Nigro, J. Gimenez, R. Rossi, and J. Marti. A fast and accurate method to solve the incompressible navier-stokes equations. *Engineering Computations*, 30(2):2–2, 2013.

S. Idelsohn, E. Oñate, N. Nigro, P. Becker, and J. Gimenez. Lagrangian versus eulerian integration errors. *Computer Methods in Applied Mechanics and Engineering*, 293:191–206, 2015.

S. R. Idelsohn, N. Calvo, and E. Oñate. Polyhedrization of an arbitrary 3D point set. *Computer Methods in Applied Mechanics and Engineering*, 192(22-24):2649–2667, jun 2003a. ISSN 00457825. doi: 10.1016/S0045-7825(03)00298-6.

S. R. Idelsohn, E. Oñate, N. Calvo, and F. Del Pin. The meshless finite element method. *International Journal for Numerical Methods in Engineering*, 58(6):893–912, oct 2003b. ISSN 0029-5981. doi: 10.1002/nme.798.

S. R. Idelsohn, E. Oñate, and F. D. Pin. The particle finite element method a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *Int. J. Num. Meth. Engng.*, 61:964–989, 2004.

R. Jackson. *The dynamics of fluidized particles. Cambridge Monographs on Mechanics.* Cambridge University Press, cambridge edition, 2000. ISBN 9780521781220.

D. Jajcevic, E. Siegmann, C. Radeke, and J. G. Khinast. Large-scale CFDDEM simulations of fluidized granular systems. *Chemical Engineering Science*, 98:298–310, jul 2013. ISSN 00092509. doi: 10.1016/j.ces.2013.05.014.

K. Kafui, C. Thornton, and M. Adams. Discrete particle-continuum fluid modelling of gassolid fluidised beds. *Chemical Engineering Science*, 57(13):2395–2410, jul 2002. ISSN 00092509. doi: 10.1016/S0009-2509(02)00140-9.

S. Koshizuka and Y. Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear science and engineering*, 123(3):421–434, 1996.

S. Koshizuka, H. Tamako, and Y. Oka. A particle method for incompressible viscous flow with fluid fragmentation. *Computational Fluid Dynamics Journal*, 4(1):2946, 1995.

S. Koshizuka, A. Nobe, and Y. Oka. Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids*, 26(7):751–769, 1998.

Kratos. Open source software platform for multiphysics computations. CIMNE, www.cimne.com/kratos, 2015.

C. Labra and E. Oñate. High-density sphere packing for discrete element method simulations. *Communications in Numerical Methods in Engineering*, 25(7):837–849, 2009. doi: 10.1002/cnm.1193.

C. Labra, J. Rojek, E. Oñate, and F. Zarate. Advances in discrete element modelling of underground excavations. *Acta Geotechnica*, 3(4):317–322, dec 2008. ISSN 1861-1125. doi: 10.1007/s11440-008-0071-2.

X. Li, X. Chu, and D. C. Sheng. A saturated discrete particle model and characteristic-based SPH method in granular materials. *International Journal for Numerical Methods in Engineering*, 72(7):858–882, nov 2007. ISSN 00295981. doi: 10.1002/nme.2037.

S. H. Liu and D. A. Sun. Simulating the collapse of unsaturated soil by DEM. *International Journal for Numerical and Analytical Methods in Geomechanics*, 26(6):633–646, may 2002. ISSN 0363-9061. doi: 10.1002/nag.215.

L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82: 1013–1024, 1977.

C. Mast, P. Mackenzie-Helnwein, P. Arduino, G. Miller, and W. Shin. Mitigating kinematic locking in the material point method. *Journal of Computational Physics*, 231(16):5351–5373, 2012.

J. J. Monaghan. Simulating free surface flows with sph. *Journal of computational physics*, 110(2):399–406, 1994.

J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.

G. G. Mustoe. A generalized formulation of the discrete element method. *Engineering Computations*, 9(2): 181–190, feb 1992. ISSN 0264-4401. doi: 10.1108/eb023857.

P. Nadukandi, B. Servan-Camas, P. A. Becker, and J. Garcia-Espinosa. On seakeeping simulations with the semi-lagrangian particle finite element method. *Comp. Part. Mech.*, (accepted) 2016.

E. Oñate. Possibilities of finite calculus in computational mechanics. *Int. J. Num. Meth. Engng.*, 60(1):255–281, 2004.

E. Oñate. *Structural Analysis with the Finite Element Method. Linear Statics. Vol 1: Basis and Solids*. CIMNE-Springer, 2009.

E. Oñate and J. Carbonell. Updated lagrangian mixed finite element formulation for quasi and fully incompressible fluids. *Computational Mechanics*, 54(6):1583–1596, oct 2014. ISSN 0178-7675. doi: 10.1007/s00466-014-1078-1.

E. Oñate and J. García-Espinosa. A finite element method for fluidstructure interaction with surface waves using a finite calculus formulation. *Computer Methods in Applied Mechanics and Engineering*, 191(6-7):635–660, dec 2001. ISSN 00457825. doi: 10.1016/S0045-7825(01)00306-1.

E. Oñate and J. Rojek. Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 193(27–29):3087–3128, 2004. doi: 10.1016/j.cma.2003.12.056.

E. Oñate, J. García-Espinosa, S. R. Idelsohn, and B. Serván-Camas. Ship Hydrodynamics. *Encyclopedia of Computational Mechanics*, Vol. 3:E. Stein, R. de Borst and T.J.R. Hughes (Eds.).

E. Oñate, S. R. Idelsohn, F. Del Pin, and R. Aubry. The particle finite element method. An overview. *International Journal of Computational Methods*, 01(02):267–307, sep 2004. ISSN 0219-8762. doi: 10.1142/S0219876204000204.

E. Oñate, J. García-Espinosa, S. Idelsohn, and F. D. Pin. Finite calculus formulations for finite element analysis of incompressible flows. Eulerian, ALE and Lagrangian approaches. *Computer Methods in Applied Mechanics and Engineering*, 195(23-24):3001–3037, apr 2006. ISSN 00457825. doi: 10.1016/j.cma.2004.10.016.

E. Oñate, S. Idelsohn, M. Celigueta, and R. Rossi. Advances in the particle finite element method for the analysis of fluid-multibody interaction and bed erosion in free surface flows. *Comput. Meth. Appl. Mech. Engng.*, 197(19-20):1777–1800, 2008. doi: 10.1016/j.cma.2007.06.005.

E. Oñate, A. Franci, and J. Carbonell. Lagrangian formulation for finite element analysis of quasi-incompressible fluids with reduced mass losses. *International Journal for Numerical Methods in Fluids*, 74(10):699–731, apr 2014a. ISSN 02712091. doi: 10.1002/fld.3870.

E. Oñate, A. Franci, and J. Carbonell. A particle finite element method for analysis of industrial forming processes. *Computational Mechanics*, 54(1):85–107, jul 2014b. ISSN 0178-7675. doi: 10.1007/s00466-014-1016-2.

E. Oñate, F. Zárate, J. Miquel, M. Santasusana, M. Celigueta, F. Arrufat, R. Gandikota, K. Valiullin, and L. Ring. A local constitutive model for the discrete element method. Application to geomaterials and concrete. *Computational Particle Mechanics*, 2(2):139–160, jun 2015. ISSN 2196-4378. doi: 10.1007/s40571-015-0044-9.

N. Patankar and D. Joseph. Lagrangian numerical simulation of particulate flows. *International Journal of Multiphase Flow*, 27(10):1685–1706, oct 2001. ISSN 03019322. doi: 10.1016/S0301-9322(01)00025-8.

D. O. Potyondy and P. A. Cundall. A bonded-particle model for rock. *International Journal of Rock Mechanics and Mining Sciences*, 41(8):1329–1364, 2004. ISSN 1365-1609. doi: 10.1016/j.ijrmms.2004.09.011.

D. J. Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):759–794, 2012.

J. Rojek, E. Oñate, F. Zarate, and J. Miquel. Modelling of rock, soil and granular materials using spherical elements. In *2nd European Conference on Computational Mechanics ECCM-2001*, Cracow, Poland, 2001.

J. Rojek, E. Oñate, and E. Onate. Multiscale analysis using a coupled discrete/finite element model. *Interaction and multiscale mechanics*, 1(1):1–31, mar 2008. ISSN 1976-0426. doi: 10.12989/imm.2008.1.1.001.

J. Rojek, C. Labra, O. Su, and E. Oñate. Comparative study of different discrete element models and evaluation of equivalent micromechanical parameters. *International Journal of Solids and Structures*, 49(13):1497–1517, jun 2012. ISSN 00207683. doi: 10.1016/j.ijsolstr.2012.02.032.

F. Salazar, E. Oñate, and R. Morán. Modelación numérica de deslizamientos de ladera en embalses mediante el método de partículas y elementos finitos (pfem). *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 28(2):112–123, 2012.

F. Salazar, J. Irazábal, A. Larese, and E. Oñate. Numerical modelling of landslide-generated waves with the particle finite element method (PFEM) and a non-Newtonian flow model. *International Journal for Numerical and Analytical Methods in Geomechanics*, 40(6):809–826, apr 2016. ISSN 03639061. doi: 10.1002/nag.2428.

M. Sommerfeld, B. van Wachen, and R. Oliemans. *Best Practice Guidelines for Computational Fluid Dynamics of Dispersed Multiphase Flows*. ERCOFTAC, 2008. ISBN 9789163335648.

D. Srinath and S. Mittal. Optimal aerodynamic design of airfoils in unsteady viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 199(29):1976–1991, 2010.

F. A. Tavarez and M. E. Plesha. Discrete element method for modelling solid and particulate materials. *Int. J. Numer. Meth. Engng.*, 70(4):379–404, 2007. doi: 10.1002/nme.1881.

T. E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Adv. Appl. Mech.*, 28:1–44, 1992.

V. Tran, F.-V. Donzé, and P. Marin. A discrete element model of concrete under high triaxial loading. *Cement and Concrete Composites*, 33(9):936–948, oct 2011. ISSN 09589465. doi: 10.1016/j.cemconcomp.2011.01.003.

B. van Wachen. An immersed boundary method for interacting particles. *Bulletin 82*, 2010.

WAM-V. http://www.wam-v.com, 2016.

X. S. Wang, L. Zhang, and W. K. Liu. On computational issues of immersed finite element methods. *Journal of Computational Physics*, 228(7):2535–2551, apr 2009. ISSN 00219991. doi: 10.1016/j.jcp.2008.12.012.

C. Wellmann, C. Lillie, and P. Wriggers. Comparison of the macroscopic behavior of granular materials modeled by different constitutive equations on the microscale. *Finite Elements in Analysis and Design*, 44(5):259–271, mar 2008. ISSN 0168874X. doi: 10.1016/j.finel.2007.11.007.

J. R. Williams and R. O'Connor. Discrete element simulation and the contact problem. *Archives Comp. Meth. Engng*, 6(4):279–304, 1999. doi: 10.1007/BF02818917.

L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193(21-22):2051–2067, may 2004. ISSN 00457825. doi: 10.1016/j.cma.2003.12.044.

O. Zienkiewicz, R. Taylor, and J. Zhu. *The finite element method. Its basis and fundamentals*. Elsevier, 6th editio edition, 2005a.

O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method for Solid and Structural Mechanics*. Elsiever Butterworth-Heinemann, Oxford, 6th editio edition, 2005. ISBN 9780080455587.

O. C. Zienkiewicz, R. L. Taylor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics, Volume 3 (6th Ed.)*. Elsiever Butterworth-Heinemann, Oxford, 2005b.

T. Zohdi. Computation of strongly coupled multifield interaction in particlefluid systems. *Computer Methods in Applied Mechanics and Engineering*, 196(37-40):3927–3950, aug 2007a. ISSN 00457825. doi: 10.1016/j.cma.2006.10.040.

T. I. Zohdi. *An Introduction to Modeling and Simulation of Particulate Flows*. Society for Industrial and Applied Mathematics, jan 2007b. ISBN 978-0-89871-627-6. doi: 10.1137/1.9780898718928.