UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

A particle finite element method for fluid-related problems in civil engineering

Miguel Angel Celigueta Jordana

A PARTICLE FINITE ELEMENT METHOD FOR FLUID-RELATED PROBLEMS IN CIVIL ENGINEERING

Miguel Angel Celigueta Jordana

Advisor:

Eugenio Oñate Ibáñez de Navarra

PhD program:

Structural Analysis

September 2019

Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Acknowledgements

I would like to express my gratitude to my PhD advisor, Prof. Eugenio Oñate, who invited me to join CIMNE with the promise of working on spectacular simulations when I was still a student. Working on the PFEM in CIMNE was indeed a very motivating experience.

Thanks to Facundo Del Pin, who had the patience to teach me a bit of C++ and the internals of his code when I just started working on the PFEM.

My gratitude also to Prof. Sergio Idelsohn, who gave me some of the best lessons about Computational Fluid Dynamics when I needed them.

Thanks to Salva Latorre, my good friend, who helped me a lot running the largest cases, and had always an eye on them, within the research projects we faced together.

Thank you, GiD guys, Miguel and Kike, and Abel, for all the support you gave me, not only about GiD, but also about C++ and how to code better.

Thanks, Fernando, for reporting many bugs and requesting improvements from Madrid. Maria Jesús, for your help with the latex documentation and friendly conversations. Riccardo, for you advice in CFD, and Santi, for diagnosing some problems at first sight. Pooyan, the best programmer I have met in person, from whom I learnt a lot. Antonia, who used my code patiently when it was not finished and used to crash. In general, I would like to thank the CIMNE community for being the place where I felt like my own home for several years.

And thanks, Lucia, for showing me from close how someone can finish a PhD really fast!

Abstract

The work presented in this Thesis is a set of developments focused on the Particle Finite Element Method (PFEM) and its applicability in several fields in Civil Engineering. The PFEM had already been proven to be a powerful tool for the free surface flows with large deformation and domain separation, but the application to actual engineering problems requires many more advances. The interaction between the fluid and many solids contacting with each other, the erosion of soils and the transport of small particles are some of these advances, which are main topics addressed in this document. Apart from them, other developments related with the fluid solution are included, which are intended to get deeper than ever before in the practical use of PFEM.

Contents

List of Figures

1 Introduction

The present document and all the work conducing to it is a consequence of a real necessity. Around 2004, fluid dynamics simulations needed extra development in order to face free surface problems with highly unsteady flows and fluid-structure interaction. At that time, the state of the art of Computational Fluid Dynamics (CFD), mostly used in the form of Eulerian methods, showed some inconveniences when tracking free surface and when interacting with structures, namely dissipative tracking of the free surface or strong need of mapping methods between the fluid and the structure. The popularity of the so-called meshless methods [14] was growing up due to a good performance in these areas and the Particle Finite Element Method (PFEM) [17] appeared as a hybrid between particles methods (Lagrangian approach) and the Finite Element Method (FEM). A team of researchers, including Professors Sergio Idelsohn and Eugenio Oñate, has been developing the PFEM since then, having reached a level of applicability and validation that proves the reliability and versatility of this method [80]. The present work collects the part of this team development that was carried out by the author at CIMNE (Centre Internacional de Mètodes Numèrics a l'Enginyeria or International Center for Numerical Methods in Engineering) since 2004, most of which was used in competitive research projects, published in several papers and used as a basis for the work of other PFEM researchers.

1.1 Motivation and general research objectives

The motivation of the work included in the present thesis was to develop a numerical method based on PFEM, which will be hereafter called *PFlow*, capable of solving some specific computational challenges involving fluids, structures and soils. The compendium of the numerical developments in this Thesis, *PFlow,* should be understood as a set of formulations, algorithms and numerical techniques useful for solving those challenges, which are listed in the following lines:

- Accurate tracking of free surface flows
- Efficient way to deal with drastic changes of the domain shape (for highly non-steady problems)
- Natural approach to face complex Fluid Structure Interaction (FSI) problems with contact
- Modeling the erosion of solids under the action of fluids

- Particles transport in fluids

These aspects of the computational CFD are briefly analyzed below, where the importance of using PFEM instead of standard Eulerian CFD is emphasized.

Free surface tracking

It is well known that standard Eulerian CFD codes tend to smooth any corner present in any shape of an interface that is advected over the fixed mesh [28]. A Lagrangian method is a possible solution to this problem, since no projection, intersection or mapping must be made. The nodes forming the free surface are advected with their own velocity and any sharp vertex of the geometry is maintained, no matter the amount of displacement or the mesh size. The mesh size still defines the minimum size that can be represented, like in Eulerian CFD. However, while in Eulerian approaches the fluid is lost below the mesh size, in PFEM is either represented by isolated nodes (that can recover mass afterwards) or by too big elements (as sliding nodes on walls, see Section 2.4.2). This feature can be advantageous in many situations.

This Thesis describes the work carried out by the author in the direction of maximizing these advantages (Section 2.4).

Drastic domain changes

A good feature of Lagrangian methods is that no initial mesh is required, so for large displacements of the nodes the mesh follows the shape of the domain naturally. Eulerian methods require an initial estimation of what parts of the space 'could eventually be filled' by the fluid. This advantage of Lagrangian methods is also good for any phenomenon that does not include deformation but does include large displacements, like a falling drop of water, in which case only the drop must be discretized.

Some of the developments included in Section 2.4 improved the PFEM in this direction.

Fluid Structure Interaction with contact

Structures have been traditionally calculated with a Lagrangian approach. Using the same approach for the fluid makes the FSI coupling easier in terms of geometry. If the fluid mesh and the solid mesh share nodes at the interface, the projection of variables between them is trivial, and the equations can be solved either monolithically or by a staggered scheme. In the latter case the boundary

conditions that each part represents on the other are communicated through the shared nodes.

Erosion of solids

The erosion of solids is a complex phenomenon which depends on very local effects of the fluid dynamics. It is a problem which engineers have usually faced and tackled by means of global parameters. The details given by the CFD methods can bring in the necessary locality to assess the erosion at a given point. The present Thesis is an opportunity to evaluate the capabilities of the PFEM in this field.

Particles transport

Particles are often found suspended in fluids, either coming from erosion or from other sources. The transport of particles with the PFEM is also studied in the present Thesis.

All these aspects of the fluid dynamics are considered crucial for many engineering problems. The new PFEM approach was meant to open new possibilities of calculation, very useful for some research fields, like the Coastal Engineering, Industrial Processes, River Dynamics, Dams design, particulate flows, etc.

Despite the initial motivation focused on the first three items, the practical use of *PFlow* broadened the work to some other developments like modeling of soil erosion and modeling particulate flows, which deserve full Sections of this document as well.

The developments of *PFlow* soon happened to be a promising tool for solving engineering problems in the field of civil and environmental engineering. With these challenges in mind the work of this Thesis was focused in the objectives described in Section 1.2.

1.2 Computational challenges addressed

In order to be a numerical tool for practical use in engineering, it has to be user-friendly enough to be used by inexperienced engineers, it has to be robust, it has to be versatile and applicable to many different fields, it has to yield all the necessary results an engineer needs, and these results have to be accurate enough for a research stage. In this Section, the main research works carried

out in order to address the computational challenges presented are listed, with brief descriptions.

Multi-physics capabilities

The *PFlow* developments address the typical problems found in the civil engineering world. Typically, fluids interacting with solids, solids being eroded by the action of the fluid and contact between solids. These are common fields of study in harbor engineering, river dynamics, safety assessment in roads, seaside or rivers.

Versatility

Commercial codes must be specific and as simple as possible. A research code like *PFlow* must be versatile in terms of calculation options, accuracy, velocity and geometry of the case. It must not be limited to waves problems or inlet problems, but it must let the user access every feature available in the code, no matter what the specific problem is.

Robustness

The developments in *PFlow* must be robust. This means that the code must not be problem dependent but general enough to solve families of problems, regardless of the number of dimensions (2D/3D), the number of solids involved, the density or viscosity of the fluid, etc...

Apart from the conceptual robustness, the actual code used for testing *PFlow* must not crash when unexpected situations happen during the numerical tests but send a message to the user informing about the situation. The very common user mistakes must be avoided by warning messages or avoiding the mistake from the user interface (not letting choose some options that lead to error). Most of the codes for research under development lack robustness. Having many users accelerates this process.

Accuracy

A validation of the code implemented is mandatory to prove that the results are useful for the engineer. *PFlow* was tested in a number of situations, several of them published in [8, 9]. In some applications of *PFlow*, accuracy has proven to be against saving computational time due to the need of small time steps in some cases.

User-friendliness

Both the pre-processing and post-processing stages can take a lot of time from the user if they are not clear enough. GiD was chosen to be the framework to interact with the code running *PFlow*. The objective from the beginning was to maintain an interface easy to use, letting the user choose just the minimum number of options. Thus, after drawing the geometry, the user has just to choose what material (solid, fluid) the entities belong to, not requiring extra boundary conditions from the user. As a tool under development, *PFlow* is changing constantly. However, both the pre-process and the post-process must be developed accordingly. This means that for every feature added to *PFlow*, the option must added to the code interface immediately. The post-process is also an important aspect, since having the correct information and well organized makes life easier for the user. *PFlow* results must be visualized with a post-processor based on multiple layers that can be switched off, or whose color can be changed.

Results completeness

The results written when running *PFlow* must be useful for the developer or an engineer, separately. The developer must be able to activate lots of hidden variables for the engineer. The engineer must have access to all those results typically useful. Free surface sensors, internal pressure sensors, integral of pressure around a body or drag forces are typically demanded results by the engineer.

1.3 Summary of research and technical developments in the thesis

The main research topics developed in this thesis are summarized below and explained thoroughly in the following chapters:

- Lagrangian fluid solver
- Fluid-multibody interaction
- Erosion of solids
- Particles transport
- Fluid-solid-soil interaction

2 Developing the PFEM

This Section describes the PFEM, including the base theory and all the developments carried out during the work for the present thesis, until the creation of *PFlow* as a useful tool for solving engineering problems.

2.1 Background

The basis of the Computational Fluid Dynamics are the Navier-Stokes equations. Literature about the first attempts to solve them can be found since the late 1950's [31], but the first appearance of a paper treating the tracking of an interface is from the early 1980's [32]. It took more than twenty years to start computing the evolution of a free surface or an interface between two different fluids. However, at that time all methods were computed on a fixed mesh by means of an Eulerian formulation of the Navier-Stokes equations, and this introduced an artificial distortion on the tracked interface in the form of diffusion. These problems are explained in detail in [28]. Probably the need of a Lagrangian description for the interface triggered the development of alternative methods based on fully Lagrangian approaches. The meshless methods (SPH and others, [29]) and other hybrid approaches like the Material Point Method (MPM, with a fixed mesh with moving particles, [30]) or the Meshless Finite Element Method [16] appeared for similar reasons.

The movement of solids in fluids is usually analyzed with the finite element method (FEM) [24] using the so called arbitrary Lagrangian-Eulerian (ALE) formulation [12]. In the ALE approach the movement of the fluid particles is decoupled from that of the mesh nodes. Hence the relative velocity between mesh nodes and particles is used as the convective velocity in the momentum equations. The ALE formulation has been used in conjunction with stabilized finite element method to derive a number of numerical procedures for fluid-structure interaction (FSI) analysis. For instance, the deforming-spatial-domain/stabilized space time (DSD/SST) [21, 22] formulation has been used for computation of fluid-structure interaction and free-surface flow problems. The Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) [23] was proposed for computation of problems that involve both fluid-structure interactions and free surfaces. The MITICT can in general be used for classes of problems that involve both interfaces that can be tracked with a moving-mesh method and interfaces that are too complex or unsteady to be tracked and therefore require an interface-capturing technique. Typical difficulties of FSI analysis using the FEM with both the Eulerian and ALE formulation include the treat-

ment of the convective terms and the incompressibility constraint in the fluid equations, the modeling and tracking of the free surface in the fluid, the transfer of information between the fluid and solid domains via the contact interfaces, the modeling of wave splashing, the possibility to deal with large rigid body motions of the structure within the fluid domain, the efficient updating of the finite element meshes for both the structure and the fluid, etc. Most of these problems disappear if a Lagrangian description is used to formulate the governing equations of both the solid and the fluid domain. In the Lagrangian formulation the motion of the individual particles are followed and, consequently, nodes in a finite element mesh can be viewed as moving 'particles'. Hence, the motion of the mesh discretizing the total domain (including both the fluid and solid parts) is followed during the transient solution.

2.2 PFEM concepts

Let us consider a continuum domain containing both fluid and solid subdomains. The fluid particles interact with the solid boundaries thereby inducing the motion of the solid which in turn affects the flow motion and, therefore, the problem is two-way coupled. The solid particles always have a velocity which is imposed externally or is a consequence of the motion of the solid.

In the PFEM approach, both the fluid and the solid domains are modeled using an *updated Lagrangian formulation*. That is, all variables in the fluid and solid domains 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 $t + \triangle t$. The finite element method (FEM) is used to solve the continuum equations in the fluid domain. Hence a mesh discretizing this domain is generated in order to solve the governing equations in the standard FEM fashion. The nodes discretizing the fluid and solid domains are viewed as material particles whose motion is tracked during the transient solution. This is useful to model the separation of fluid particles from the main fluid domain, and to follow their subsequent motion as individual particles with a known density, an initial acceleration and velocity and subject to gravity forces.

It is important to recall that each fluid particle is treated as a material point characterized by the density of the fluid domain to which it belongs. The mass of a given domain is obtained by integrating the density over the volume of the domain.

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 gradients of the fluid or the structure

variables occur.

Basic steps of the PFEM

For clarity purposes I will define the *collection* or *cloud of nodes (C)* as the set of all nodes, pertaining to either the fluid and the solid subdomains, the *volume (V)* defining the analysis domain for the fluid and the solid, and the *mesh (M)* discretizing both domains.

A typical solution with the PFEM involves the following steps (see Fig.1 for clarification).

1. The starting point at each time step is the cloud of points in the fluid and solid subdomains. For instance ${}^{n}C$ denotes the cloud at time $t = t_n$ (up-left drawing of Fig.1).

2. Discretize the fluid and solid subdomains with a finite element mesh ${}^{n}M$. In this work a mesh generation scheme based on the Delaunay tesselation [16] is used.

3. Identify the boundaries for both the fluid and solid domains defining the analysis domain ${}^{n}V$ in the fluid and the solid. This is an essential step as some boundaries, such as the free surface in fluids or the bed surface, may be severely distorted during the solution process including separation and re-entering of nodes. The Alpha Shape method [15] is used for the boundary definition. Central drawing of Fig.1 represents the mesh after Steps 2 and 3.

4. Solve the coupled Lagrangian equations of motion for the fluid and the solid domains. Compute the relevant state variables in both domains at the next (updated) configuration for $t+\Delta t$: velocities, pressure and viscous stresses in the fluid and displacements. Note that the boundary conditions can be applied to the boundaries identified in Step 3. Bottom-left arrows of Fig.1 represent this step.

5. Move the mesh nodes to a new position ${}^{n+1}C$ where $n + 1$ denotes the time $t_n + \Delta t$, in terms of the time increment size. This step is typically a consequence of the solution process of step 4. Recall that a node identifies a material point in either the fluid or solid subdomains.

6. Go back to step 1 and repeat the solution process for the next time step.

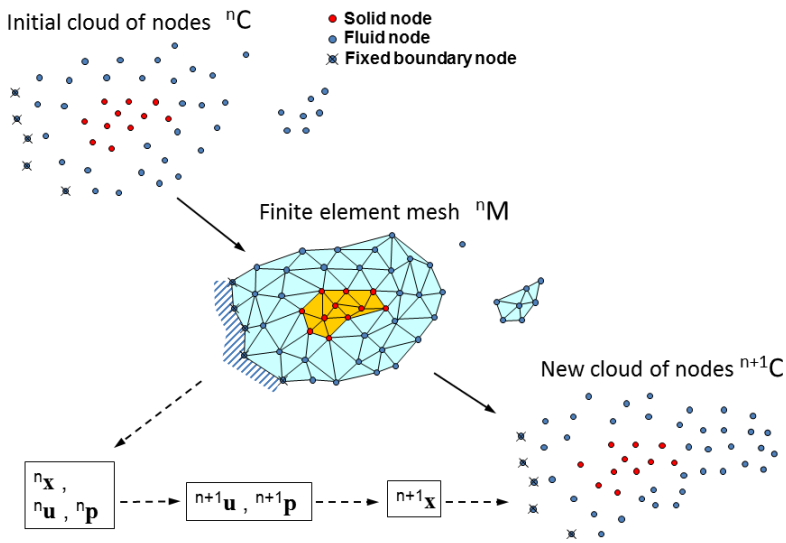


Figure 1: Steps of the PFEM in a time increment

After Step 2, the resulting mesh is a convex hull, with all the elements generated by the Delaunay tesselation (see Fig. 2). This mesh includes big elements and some of them with high aspect ratio, this means that some nodes are connected with elements even if they are very distant from each other. Step 3 removes all the elements that introduce these connections and after applying the *Alpha-shape method* the mesh can contain isolated nodes, holed domains and separated parts.

2.3 Previously existing tools

The code inherited by the author in CIMNE in 2004 was roughly a fluid solver with a minimal user interface. This Section describes the approach followed by that code, which was the base of all the developments made until getting the *PFlow* package. Many corrections and new lines were added to that code in order to reach the marked objectives (Section 1.2).

The following lines provide details about each of the features of the preexisting code in 2004:

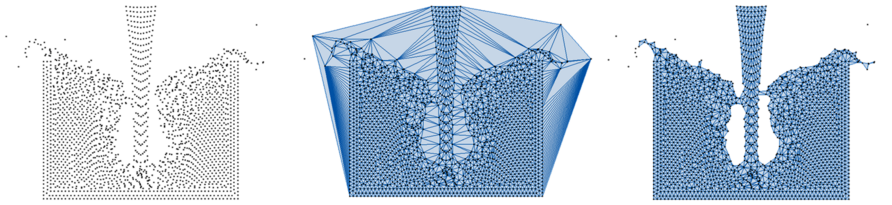


Figure 2: From left to right: steps 1, 2 and 3 (figures obtained from [26])

Fluid solver kernel

The Fluid Dynamics module already coded was a Fractional Step scheme, with a switch to choose between a first order Fractional Step or a second order Fractional Step with a Predictor Multi-Corrector algorithm. The code, written by Facundo Del Pin, was described in detail in [17]. It was intended to work both in 2D and 3D, however, the 2D version was much more robust.

Solid motion

The Solid Solver module was in charge of computing of the motion of a solid. It was a very simple one, only valid for 2D. It included a Newmark integration scheme.

Fluid Structure Interaction

The Fluid Structure Interaction module was capable of solving the interaction between several 2D solids and a fluid. It used an iterative scheme for the coupling with the fluid based on previously published works [18].

Contact

No contact was designed or coded for the interaction between solids.

User interface

The pre-process user interface was quite basic, but very intuitive. It was a *problemtype* for GiD ([25], www.gidhome.com) based on layers. That means, the geometry had to be prepared in layers, and every layer was an object with

different conditions. Every object was considered as a different entity in the calculation module and they could be either solids, fluids or fluid inlets.

2.4 Research developments carried out in the thesis

Many new features were required in order to achieve the objectives stated in Section 1.2. The main changes in the fluid solver were done to achieve the needed robustness or to improve the trajectories integration functions, so the work focused primarily in repairing the mesh after obtaining the Delaunay Tesselation, sending working elements to the assembler, and modifying the integration of the trajectories of the nodes, which usually presented problems of penetration in walls. Many other changes related to solid motion, solid-solid contact, user interface, free surface and pressure sensors, general bug fixing, robustness and post process were done. In this Section, many features added to *PFlow* by the author are described. Some of them will be explained with a short description in this very Section, while the most complex ones are explained in detail in the following chapters. The level of detail and implementation hints exposed in this thesis should be enough for any researcher to implement the same features, in case they develop their own PFEM code.

2.4.1 Main research developments

The main research developments are those which can not be summarized in short Subsections because they require a deeper explanation. Therefore, full chapters have been dedicated to them. They are nevertheless summarized here.

Fluid Multi-body interaction The multiple bodies that can be floating, sinking, or generally speaking interacting with the fluid, with each other or with the fixed walls, must be subject to a contact algorithm that prevents any penetration. In the PFEM, the detection of the contact was found to be very cheap (minor extra cost from a computational cost view) thanks to the existence of the re-meshing step. All those elements whose connectivity include nodes from different solids are considered 'contact elements', having a linear repulsion law and a drag that simulates the contact. As stated previously, the goal of the algorithm is to prevent penetration. An error of the element size is assumed in exchange for an instantaneous detection. The multi-body contact is explained in detail in Chapter 3.

Erosion An erosion module was added to the Solid Solver module of the *PFlow-Code* (the software used, see Section 2.5). This module calculates the drag produced by a fluid on a solid surface and is able to detach particles of the solid. This module also includes a stability assessment function that detaches any part of the solid which is not stable in terms of internal friction angle. This procedure is explained in detail in Chapter 4.

Particles transport in fluids An algorithm for tracking the particles motion in a fluid was developed, using empirical formulae for drag valid for Newtonian and non-Newtonian fluids and for spherical and non-spherical particles. This feature of *PFlow* is explained in detail in Chapter 5.

2.4.2 Other research developments

Many other developments were necessary in order to run practical engineering problems. In this Section some of them are explained briefly.

Euler equations for body motion The rotation of a body in 3D requires solving the Euler equations, expressed in local axes attached to the solid as

$$I_i\dot{\omega}_i + (I_k - I_j)\omega_j\omega_k = M_i \tag{1}$$

where sub indices i,j and k represent the cyclic permutations of the principal directions of the Inertia tensor, I_i are the principal moments of inertia of the rigid body, M_i are the values of the external torques in the same directions and ω is the angular velocity.

PFlow solves these equations using the following Newmark scheme:

$$\omega_{i,predicted}^{n+1} = \delta t(1-\gamma)\dot{\omega}_i^n \tag{2}$$

$$\dot{\omega}_i^{n+1} = \frac{1}{I_i}(M_i^{n+1} - (I_k - I_j)\omega_{j,predicted}^{n+1}\omega_{k,predicted}^{n+1}) \tag{3}$$

$$\omega_i^{n+1} = \delta t\gamma\dot{\omega}_i^{n+1} \tag{4}$$

Where δt is the time step, the super indices indicate the step, the sub index $_{predicted}$ stands for a temporary estimation of the value and γ can have values between 0 and 1. It is usually taken as 0.5.

As the Inertia tensor is used in its diagonal form, these equations must be solved in local, principal axes. This limitation requires to express the external

torques and the angular velocities in local coordinates, and rotate the final value of the angular velocities to global coordinates for plotting.

From this angular velocity, the rotation (θ) of the solid can obtained with a simple Crank-Nicholson scheme as

$$\theta_i^{n+1} = \delta t(1-\gamma)\omega_i^n + \delta t\gamma\omega_i^{n+1} \tag{5}$$

A similar process can be followed in order to solve the translational motion of the rigid body.

Note: In the first implementations of this feature, the trajectory of all the nodes of the solid was integrated. The differential cumulative errors over the nodes generated unwanted distortion of the solids for high angular velocities and long computations. A better approach was developed later on, based on translating and rotating the local axes of the solid and reposition the nodes using relative coordinates to these axes.

Fluid boundary conditions In the first version of the code, before any changes made by the author, the boundary conditions for the fluid were:

- Dirichlet condition for all the components of the velocity at the boundaries defined by solids, both moving or fixed solids
- Dirichlet condition for the pressure at the free surface, with a null value

Later on, some other conditions had to be added, in order to achieve better results in terms of accuracy and robustness. They are described briefly in this Section.

Pressure at the free surface

For very viscous fluids, prescribing a zero value of pressure at the free surface is inaccurate. If we take into account that the null normal stress condition at the free surface is the one written in equation 6:

$$\sigma{\cdot}\mathbf{n}{\cdot}\mathbf{n} = \mathbf{t}{\cdot}\mathbf{n} = \mathrm{t_n} = 0 \tag{6}$$

and the expression of the stress tensor as (Eq. 7):

$$\sigma = 2\mu\nabla^s\mathbf{v} + p\mathbf{I} \tag{7}$$

Considering that $\nabla^s\mathbf{v} \simeq \nabla\mathbf{v}$ under certain conditions (if the viscosity is constant and has a low value, as studied in [57]), we finally obtain that the

actual value of the pressure to be imposed at the free surface of viscous fluids is (Eq. 8):

$$p = 2\mu \frac{\partial \mathbf{v}_n}{\partial \mathbf{n}} \tag{8}$$

where σ is the stress tensor, $\mathbf{v}$ is the fluid velocity, p is the fluid pressure, $\mathbf{n}$ is the unit vector normal to the free surface (outwards), $\mathbf{v}_n$ is the projection of the velocity vector into $\mathbf{n}$, $\mathbf{t}$ is the vector of tractions at the free surface and μ is the dynamic viscosity of the fluid.

Nodes approaching walls

In terms of robustness, an extra condition had to be added to the fluid nodes approaching a wall. There is no condition in the equations that prevents the fluid nodes to have a certain velocity towards the wall. It is actually very common that they have some component of the velocity perpendicular to the walls, while the parallel component is still a lot larger. For this reason, it occurs sometimes that a fluid node approaches progressively a wall, until it crosses it. This happens because the Alpha-shape method detects this node as a free surface node (Fig. 3), and it has therefore a very low pressure compared to its neighbours. Then, this node is pushed trough the walls and a constant leakage starts at that location.

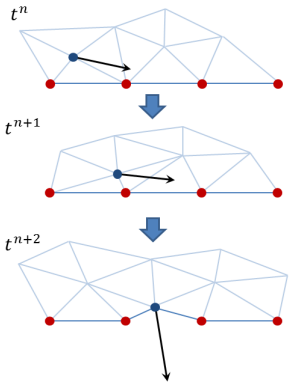


Figure 3: Fluid node (in blue) approaching a wall (red nodes) and starting leakage

The solution adopted to avoid this serious problem was to detect the fluid

nodes that are candidates to be problematic and relocate them to a position far from the wall, interpolating the fluid and the pressure at the new position using the last known information of the mesh (see Fig. 4).

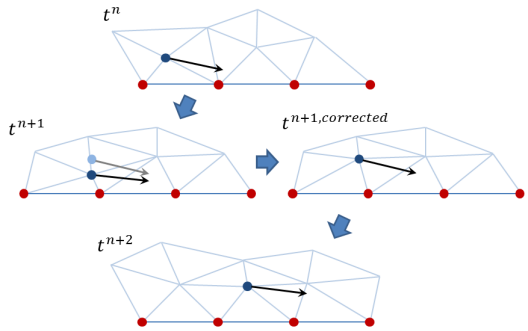


Figure 4: Position of fluid node (in blue) is corrected to avoid leakage through wall (red nodes)

This correction of the position of these problematic nodes is mandatory for any PFEM code, in the opinion of the author.

Mesh quality control With the standard formulation described in Section 2.2, the nodes move according with the velocity of the fluid. This means that, even if the initial mesh has very good shaped elements, on one hand, some groups of nodes get really close, generating very distorted elements with short and long edges, while on the other hand some nodes get very far from each other, generating big elements that can be removed by the Alpha-Shape method. Thus, it is necessary to work on the distorted meshes in order to recover the quality of the mesh. The implemented operations were of two types: node deletions and node insertions. These operations are described below, however, it is important to clarify that they respect the following 'symmetry criterion': *the mesh operations related to the Alpha-shape method should be done in such a way that, reversing the velocity field (or going backwards in time), the same elements should be recovered.* See Fig. 5 for clarification. This 'symmetry criterion' is a necessary condition for the conservation of volume across time steps, without which the creation of new elements could be larger than the deletion, or the other way around.

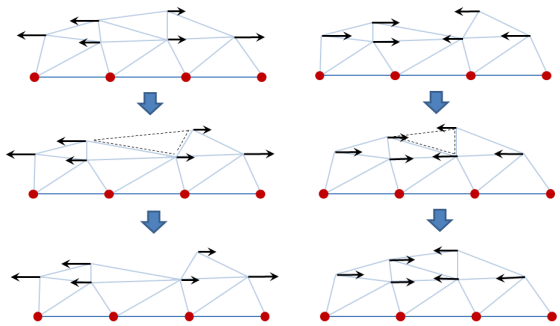


Figure 5: Expanding mass of water (left) vs. contracting mass of water (right). The same element is deleted (left) or created (right)

Node deletion

The distance between a certain node and all their neighbours is checked. If at least one of the distances is less than a certain portion of h (εh) the node is marked to be removed, unless the small distance is measured against a node which is already marked to be removed. h is the desired size of element around that node and ε is a value between 0 and 1, calibrated by the experienced user. If two nodes are too close, but only one of them belongs to the free surface, the other node is marked to be removed to avoid spurious changes in the shape of the free surface.

The nodes marked to be removed will not be added to the mesh during the next re-meshing step.

Node insertion

When an element gets big, that is, the sphere passing through all its nodes has a radius bigger than a certain threshold (βh), it might approach the range where Alpha-Shape can remove the element itself. Unless it is an element close to the free surface, it must be understood as a mesh distortion and should be fixed by adding nodes, which the Delaunay Tesselation will later use to generate a better quality mesh. The simplest and most effective way to achieve such increase of the resolution is to insert a node in the baricenter of the element. If the element is close to the free surface, having at least one face with all its nodes

marked as free surface, it should not be enriched with a new node, or the mesh would gain volume. Note that this is consistent with the 'symmetry criterion' for Alpha-Shape: an element is formed when nodes on the free surface get close, so this element must be deleted when the nodes separate, not enriched with new nodes (Fig. 5).

Many combinations of ε and β can ensure the conservation of volume in terms of re-meshing operations (solver losses must be considered as a different source), if ε is set bigger many nodes are deleted, so β must be chosen smaller to create many of them as well. However, the conservation of the number of nodes does not ensure the conservation of volume, as the density of nodes is always something relative to the imposed size h. For example, an initial mesh with a separation between nodes of $h/2$ will soon trigger many deletions of nodes, which is something expected due to a too dense distribution, but this does not mean that volume is lost.

Mesh size control Once the mesh quality control is added to a PFEM code, it is possible to control the mesh size in any part of the domain. The mesh size control is done by setting new values of h in the nodes. This simple procedure triggers automatically the necessary deletion or creation of nodes as described in Section 2.4.2. Thus, one can refine the domain progressively by decreasing the value of h in all nodes. If needed, the user can establish zones in space with different prescribed values of h, in such a way that when a node enters those zones its value of h is overwritten to the prescribed values. These zones can be prescribed in the form of analytic functions or as a function of the distance to an object, for example. Care must be taken with discontinuous jumps of h: if the values at both sides of the jump differ by more than a factor of 2 times, voids might be created on the border during the computation.

In Fig. 6 a floating caisson can be seen. Note that two different levels were used to increase the accuracy next to the caisson. The zones were defined by user mathematical functions based on spatial coordinates. The rigid walls containing the fluid were refined with the same sizes during the pre-process stage.

Fluid inlets A fluid inlet was coded in such a way that successive layers of nodes are inserted in a previously set surface (in 3D) or line (in 2D). Fig. 7 depicts a scheme of the process in 2D. Every new layer of nodes must be considered exactly in the same way as a solid wall, with a Dirichlet velocity condition (step 2). For several time steps (step 3) the boundary condition is

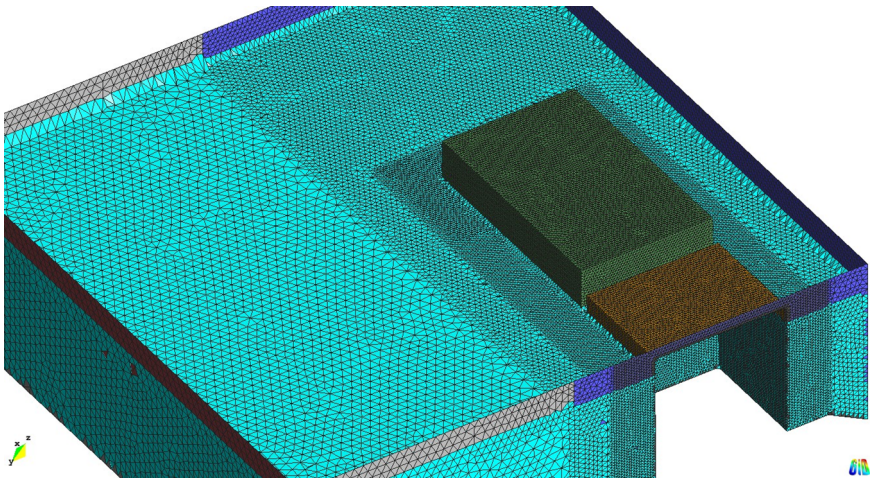


Figure 6: 2D scheme of the fluid inlet process

maintained until a certain distance from the inlet is reached and a new layer of nodes is inserted (step 4). This distance must be similar to the size of the elements. The old nodes are freed and the new ones are fixed to the same velocity as before. At this stage, both layers get connected by elements that are created automatically by the re-meshing process. After this, the newest nodes move with the imposed velocity (step 5) until they are replaced by a new layer of nodes in the same way it was done before (step 6). Parabolic laws and logarithmic laws can also be imposed at the inlet.

Surface tension For cases at small scales, or for fluids different from water, sometimes the surface tension can be necessary. This feature was added to *PFlow* (in 2D only) as an extra pressure imposed on the free surface. Following expression 9:

$$p_{fluid} = p_{out} + \sigma_{fs} \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \tag{9}$$

the nodes at the free surface should have a pressure p_{fluid}(enforced) which is equal to the external pressure p_{out} (often zero) plus a term that depends on the material surface tension σ_{fs} and the curvatures of the free surface, defined by

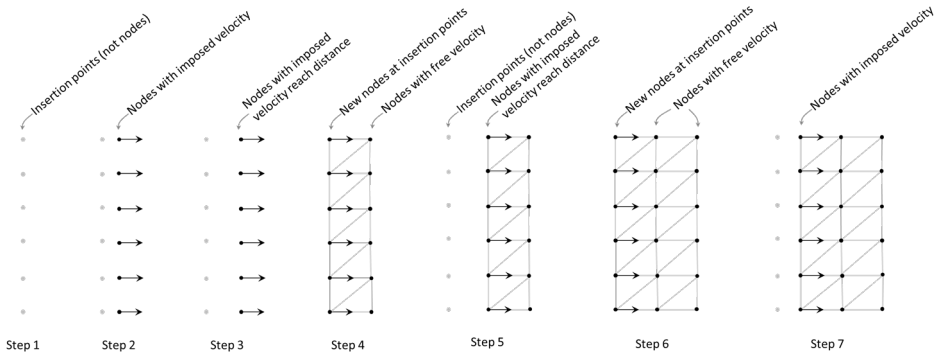


Figure 7: 2D scheme of the fluid inlet process

two radii, R_1 and R_2. In 2D one of the radii is infinite, so the Eq. 9 is simplified to

$$p_{fluid} = p_{out} + \sigma_{fs}\frac{2}{R} \tag{10}$$

The value of σ_{fs} for the water is $0.0728\,\mathrm{N/m}$. The value of p_{fluid} is computed considering the shape of the free surface at the previous time step, so this method can be considered *explicit*.

A simple experiment was carried out to test the implementation of this approach: a random shaped mass of water with no gravity was calculated to see its evolution with time. Initial and final shapes are depicted in Fig.8

For the typical problems in civil engineering, this term can be considered negligible. However, for small scale problems, this term is very important. In these cases, the explicit approach can limit the time step considerably, and it is published in the literature [63] that an implicit approach is worth in terms of time consumption.

Non-Newtonian rheology A Bingham Plastic model was added to the Fluid Solver module of the code using *PFlow*. The technology used had been published beforehand [10] and consists on using an *apparent viscosity* which is updated according to the local, current value of the *strain rate*. A user defined smoothing of the rheogram is applied in the zone close to the origin to avoid discontinuities.

Figure 8: Initial shape (left) and final shape (right) of a mass of water

User interface The pre-process Graphical User Interface (GUI) was coded with all the features that were necessary by the author and members of the research team. Also third party users, not developers, that were using the *PFlow-Code* (Section 2.5) asked for extra features that were coded and integrated in the GUI along the years. The aspect of some dialog menus of the GUI is depicted in Figures 9, 10 and 11 and all the support for graphics is brought in by GiD [25].

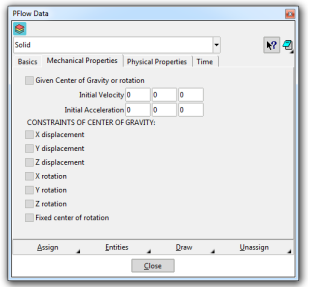


Figure 9: Screenshot of the GUI. Setting the properties of a solid.

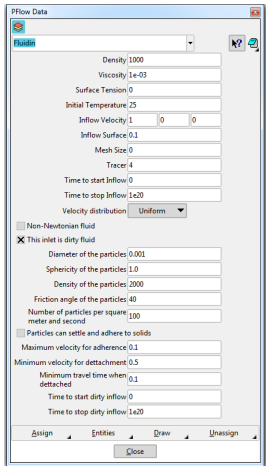


Figure 10: Screenshot of the GUI. Setting the properties of a fluid inlet.

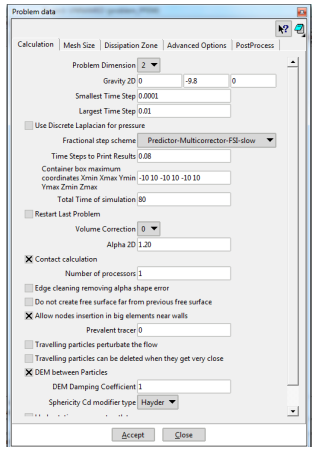


Figure 11: Screenshot of the GUI. Setting general options

Free surface sensors and pressure sensors This option is essential for engineers that use the *PFlow-Code* (Section 2.5). The user can activate the output of as many sensors as wanted and a file is written than can be easily

plotted. An example can be seen in Fig. 13. The graph represents the water level at a position marked with a line in the 2D geometry of the flume of Fig. 12

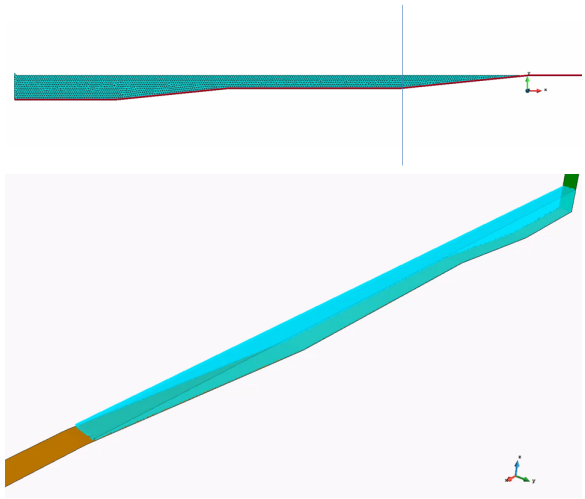


Figure 12: 2D and 3D geometry of a computed flume, originally an experimental facility

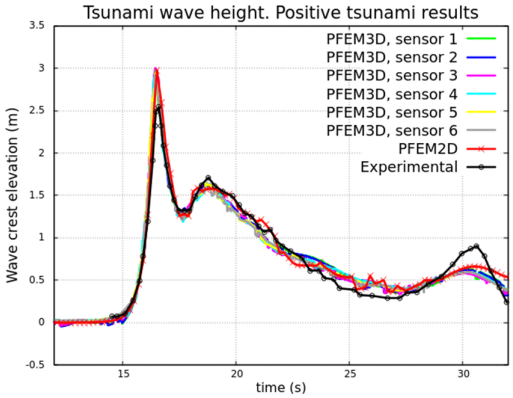


Figure 13: Water level graph for a single wave

Code performance monitoring A function was included to print the memory consumption, number of nodes and elements, time elapsed from the beginning of the calculation and time consumed by every stage of each time step (meshing, assembling, solving and other operations). Thanks to this feature, graphs of CPU consumption like the one in Fig. 14 were possible, where every point is a time step of a computation with an increasing number of nodes. It can be observed that the cost of re-meshing the domain at every time step is around 16% for around 800,000 nodes, and the tendency is to decrease even more for larger meshes.

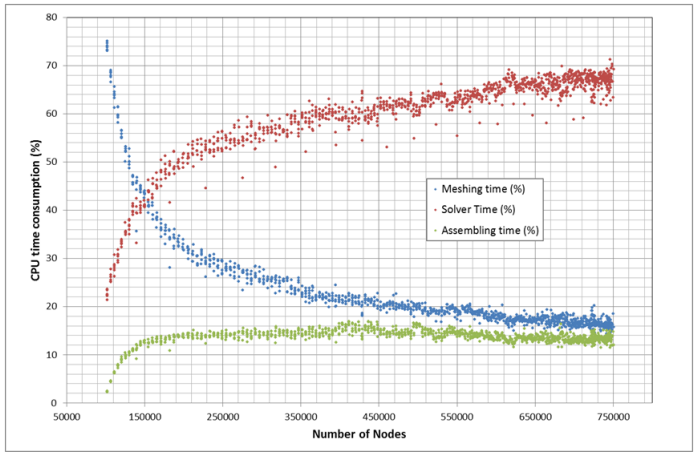


Figure 14: CPU consumption (%) of the main functions

Calculation resumption With this function, a backup of all the data is stored in the hard disk periodically, so the calculation can be resumed from and advanced point on. This feature demonstrated to be essential when code was used for research projects or in calculations subject to tight deadlines.

2.5 The translation of PFlow developments into software

The *PFlow* is the compilation of all the research work described in Sections 2.3 and 2.4. Part of this work was published in journal papers ([1, 2, 3, 4, 5, 6, 7]). However, the most tangible proof of the advances done in the field is the *PFlow-Code,* the software programmed by the author during the evolution of this thesis.

The *PFlow-Code* consists on two different components:

- The computing kernel: containing all the techniques and methods explained in this Section
- The Graphical User Interface: a customization of GiD [25] which allows the user to create and run examples

A few snapshots of the results, which are included in the referenced publications, are included in this Section. In Fig. 15 a breakwater is under the effect of big waves. All the blocks can move freely, they can be pulled away by the waves, and the sea water penetrates in the voids between the blocks.

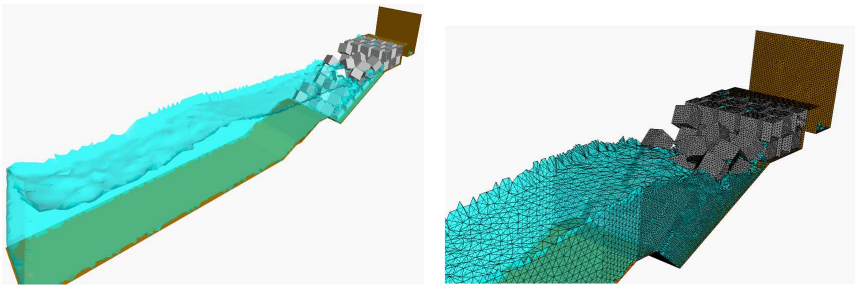


Figure 15: Waves against breakwater, extracted from [6]

In Fig. 16 a structure made of a soft material is eroded by the action of incoming waves. The object on the erodible structure falls down when the base where it stands is pulled away by the water.

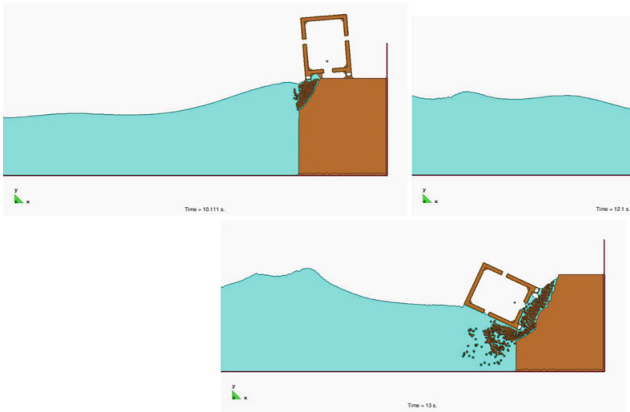


Figure 16: Waves erode soft material and object on top falls down , extracted from [7]

Many other application examples have been run by the author and his col-

laborators using the *PFlow-Code*, either for the publications cited above or the solution of numerous research projects (see Section 6).

All the post-processing work was carried out using GiD [25].

3 Fluid Multi-body Interaction

In a general problem, multiple bodies can be interacting with the fluid, with each other and with the fixed walls. For this reason, a fluid-structure-interaction (FSI) algorithm and a contact algorithm must be used in the computations. Even when the focus is just on the fluid or the FSI, the computations usually require the contact between solids to be computed, at least to avoid obvious non-physical behavior of the solids, which might not affect the fluid results in the region of interest, but would be hard to explain.

The Fluid-Multi-body Interaction problem must be solved in two separate ways: on one hand the single body FSI, on the other hand the contact between bodies. The Fluid-Structure Interaction with a single rigid body was already coded before the work of the author, as stated in Section 2.3 and followed the algorithm proposed in [18], so it will not be detailed here. The contact between rigid bodies has been developed and coded from scratch by the author, and some details were published in [4]. The ideas of the implementation are summarized in this Section.

3.1 Developments

3.1.1 Contact between solids

The contact between two rigid bodies is simply treated by introducing a layer of *contact elements* between the two interacting solid interfaces. This layer is automatically created during the mesh generation step by detecting which element is formed by solid nodes that belong to different solids (Fig. 17).

A reference distance h_c must be prescribed between the two solid boundaries. This distance can be a fraction of the size of the element. If the distance between the solids is bigger than h_c, the generated elements are treated as non-active contact elements, otherwise a frictional contact law is activated. This frictional contact includes a normal elastic force, a normal damping force and a tangential Coulomb frictional force defined as

$$F_{ni} = K_1(h_c - h) - K_2 V_{ni} Sign(V_{ni}) \tag{11}$$

$$F_{ti} = -\beta K_1(h_c - h) Sign(V_{ti}) \tag{12}$$

Where K_1 is an elastic constant that can be calculated as a function of the mass of the solids and their relative approaching velocity, K_2 is a damping

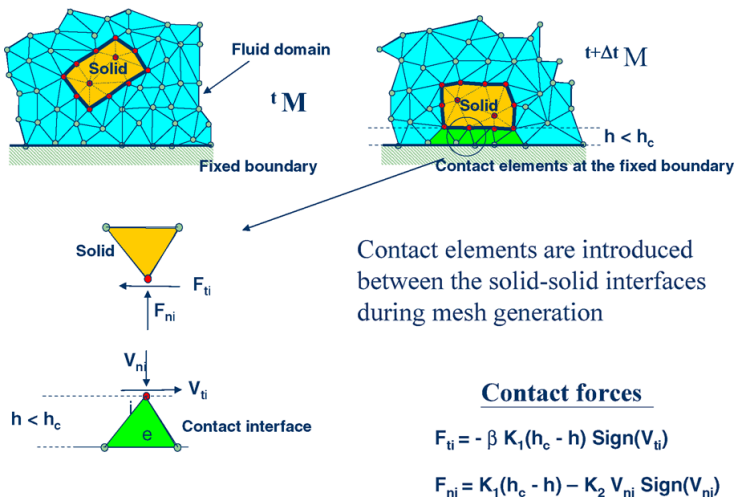


Figure 17: Contact of a moving solid with a fixed wall (extracted from [4])

coefficient that can be calculated as a function of the mass of the solids and K_1, V_{ni} is the relative normal velocity between the two solids at that contact element, V_{ti} is the relative tangential velocity between the two solids at that contact element and β is the standard Coulomb friction coefficient. h is the current distance between a node which belongs to a solid and the face which belongs to the other solid. Note that this h is not the same h used in Chapter 2, but the same notation has been used to preserve the compatibility with Fig. 17 extracted from reference [4]. Part of this implementation was reported in this publication. See Annex 9.4.

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. Of course the accuracy of this contact model depends on the critical distance above mentioned. Therefore, finer meshes imply thinner contact layers, which are more accurate but more demanding in terms of time step size (thinner contact layer require higher values of K_1 for stiffer behavior, so smaller time steps) .

This contact algorithm can also be used effectively to model frictional contact conditions between rigid or elastic solids in standard structural mechanics applications. Figures 18, 19 and 20 show examples of application of the contact

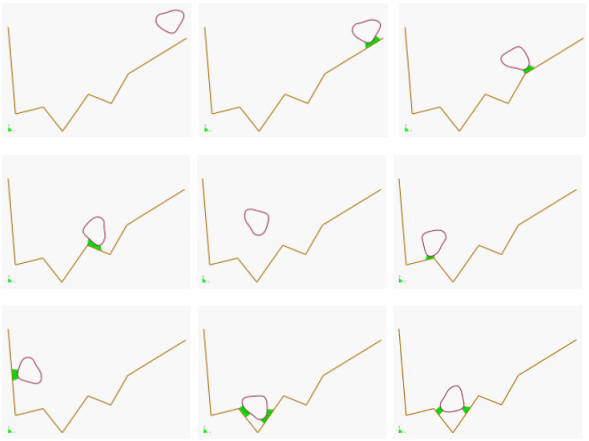


Figure 18: Bumping of a ball within a container. The layer of contact elements is shown at each contact instant

algorithm to the bumping of a ball falling in a container, the failure of an arch formed by a collection of stone blocks under a seismic loading and the motion of five tetrapods as they fall and slip over an inclined plane, respectively. The images in Figures 18, 19and 21 show explicitly the layer of contact elements which are in charge of preventing the penetration.

3.1.2 The contact force computation in the FSI-Multibody Contact scheme

The contact forces are computed according to the displacement of the solids, however, an iterative scheme was developed in *PFlow* in order to achieve a better accuracy. Several levels of nested loops are necessary for solving the couplings. The pseudocode in Algorithm 1 describes all the loops necessary for the three-way coupling.

This algorithm was the only one used for solving all the problems in Sections 3, 5, 5.3 and 6. The proven robustness, regardless of the number of solids or contacts between them, makes *PFlow* a powerful tool to solve a wide range of problems in the field of Civil Engineering and many others. Some drawbacks appear as well, which are listed in Section 9.2.

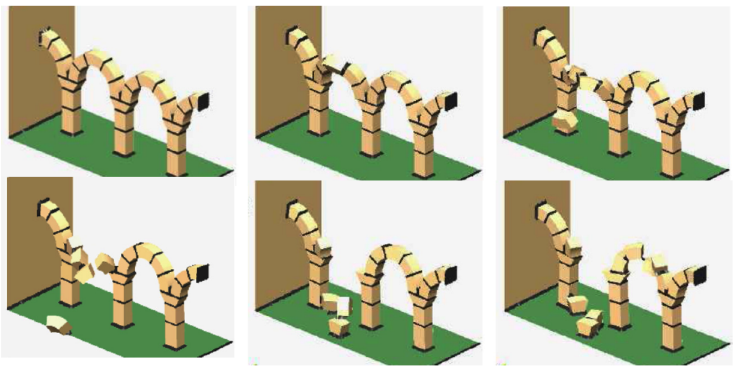


Figure 19: Failure of an arch formed by a collection of stone blocks under a seismic loading

3.1.3 Correction of the contact force

The Newmark algorithm presented in equations 13 and 14 is used to solve the translation of solids:

$$\boldsymbol{v} = \boldsymbol{v}_0 + dt[(1-\gamma)\boldsymbol{a}_0 + \gamma\boldsymbol{a}] \tag{13}$$

$$\boldsymbol{x} = \boldsymbol{x}_0 + dt[(1-\theta)\boldsymbol{v}_0 + \theta\boldsymbol{v}] \tag{14}$$

Where $\boldsymbol{v_0}$ and $\boldsymbol{v}$ are the velocities of the particle at the beginning and at the end of the time step, respectively, $\boldsymbol{x}_0$ and $\boldsymbol{x}$ are the positions, and $\boldsymbol{a}_0$ and $\boldsymbol{a}$ are the accelerations of the particle due to external forces, including the gravity (if considered) and the contact forces with the wall (spring). The values of γ and θ control how explicit or implicit the method is.

In order to solve equations 13 and 14, knowing the value of $\boldsymbol{a}$ is enough. However, $\boldsymbol{a}$ depends on the contact forces, which are a function of the unknown value of $\boldsymbol{x}$. This means that equations 13 and 14 are implicit equations. As stated in Section 3.1.1, the contact forces are solved iteratively for solids, and the same approach is used here for a single particle. $\gamma = \theta = 0.5$ are the values which yield better energy conservation properties.

The beginning and the end of a contact between two solids is a typical source of energy gain. The fact that the solids start or end the contact in the middle of a time step introduces an error when using the values of the

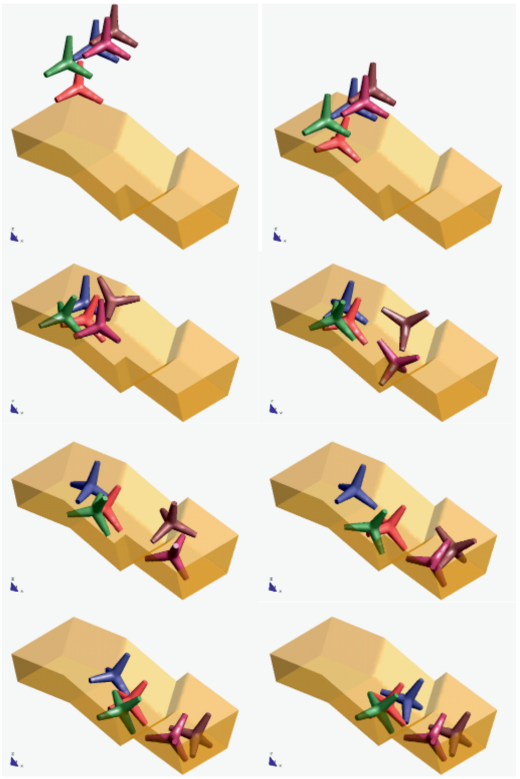


Figure 20: Motion of five tetrapods on an inclined plane

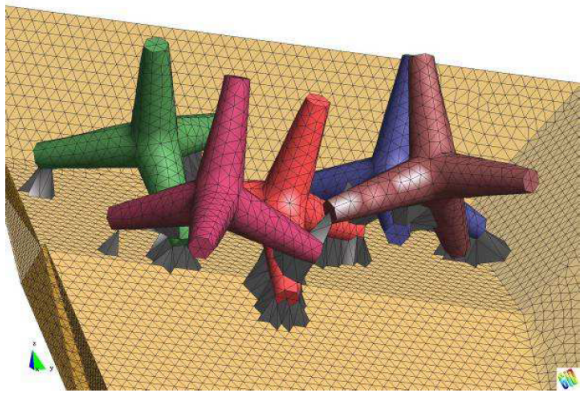


Figure 21: Detail of five tetrapods on an inclined plane. The layer of elements modeling the frictional contact conditions is shown

external forces (therefore accelerations), because they evolve actually far from linearly and present a slope discontinuity at the moment of starting or finishing a contact (see Figures 22 and 23 for clarification). These slope discontinuities were addressed by Mohammadi in [43].

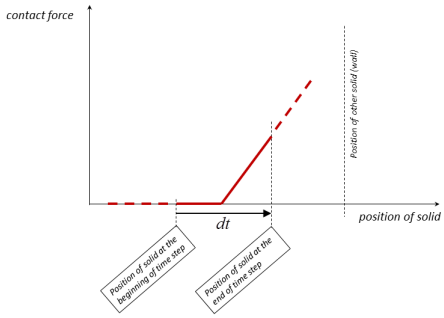


Figure 22: Contact initiation in the middle of a time step

Algorithm 1 General algorithm for the FSI with multibody contact

```
1:  while Fluid time is not ended do
2:      while convergence in Fluid velocities and pressures is not achieved do
3:          Solve Fluid equations
4:          for all solids do
5:              Integrate pressure on the skin of the Solid
6:          end for
7:          for all substeps of Solids in a Fluid time step do
8:              while convergence in contact forces is not achieved do
9:                  for all solids do
10:                     while convergence in forces is not achieve  do
11:                         Apply Solid displacement
12:                         Compute contact forces
13:                     end while
14:                 end for
15:             end while
16:         end for
17:     end while
18: end while
```

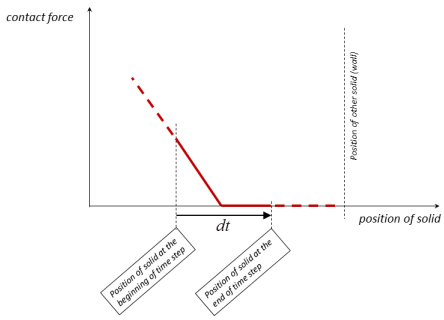


Figure 23: Contact ending in the middle of a time step

A numerical model for the motion of a particle has been studied in one

dimension, considering the gravity and an elastic contact with a horizontal rigid wall. The collision with the wall is computed like a spring exerting a force against the particle when it approaches the wall. Fig. 24 shows how the energy increases drastically at some specific moments. The input parameters for the computation were:

- Particle mass: 1 kg
- Drop height: 5 m
- Length of relaxed spring: 0.1 m
- Elastic constant for the spring: 6000 N/m
- Time step: 0.02 s

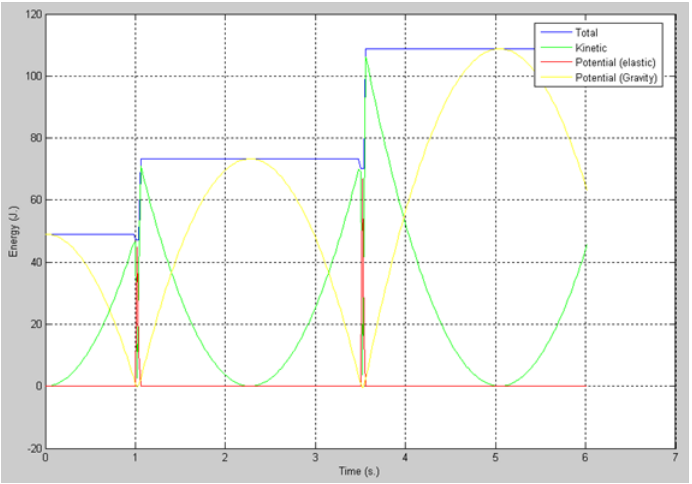


Figure 24: Energy plots for a single particle bouncing on a horizontal wall (single spring)

One way to eliminate this source of energy gain for a single particle is to split the time step dt in two parts: αdt (from the beginning of the time step until the discontinuity in forces) and $(1-\alpha)dt$ (the rest of the time step). Then the Newmark equation for the velocity (Eq. 13) can be re-written as

$$\boldsymbol{v} = \boldsymbol{v}_0 + \alpha dt[(1-\gamma)\boldsymbol{a}_0 + \gamma\tilde{\boldsymbol{a}}] + (1-\alpha)dt[(1-\gamma)\tilde{\boldsymbol{a}} + \gamma\boldsymbol{a}] \tag{15}$$

which consists on a splitting of the time step in two parts and $\tilde{\boldsymbol{a}}$ is the acceleration of the particle at the moment when the contact force appears or disappears, called 'intermediate acceleration'.

Note that $\boldsymbol{a_0}$, $\tilde{\boldsymbol{a}}$ and $\boldsymbol{a}$ contain the contribution of both gravity forces and contact forces, so they can be split as follows (equations 16 to 18):

$$\boldsymbol{a}_0 = \boldsymbol{a}_{0,gravity} + \boldsymbol{a}_{0,contact} = \frac{\boldsymbol{F}_{0,gravity} + \boldsymbol{F}_{0,contact}}{m} \tag{16}$$

$$\tilde{\boldsymbol{a}} = \tilde{\boldsymbol{a}}_{gravity} + \tilde{\boldsymbol{a}}_{contact} = \frac{\tilde{\boldsymbol{F}}_{gravity} + \tilde{\boldsymbol{F}}_{contact}}{m} \tag{17}$$

$$\boldsymbol{a} = \boldsymbol{a}_{gravity} + \boldsymbol{a}_{contact} = \frac{\boldsymbol{F}_{gravity} + \boldsymbol{F}_{contact}}{m} \tag{18}$$

Where the sub-index *gravity* means refers to gravity and the sub-index *contact* refers to contact forces. m is the mass of the particle and letter $\boldsymbol{F}$ is reserved for forces. Note that very often $\boldsymbol{F}_{0,gravity} = \tilde{\boldsymbol{F}}_{gravity} = \boldsymbol{F}_{gravity}$ and that $\tilde{\boldsymbol{F}}_{contact}$ is null in those time steps where a contact is starting or ending.

With this correction the energy gain is reduced to unnoticeable values, as shown in Fig. 25.

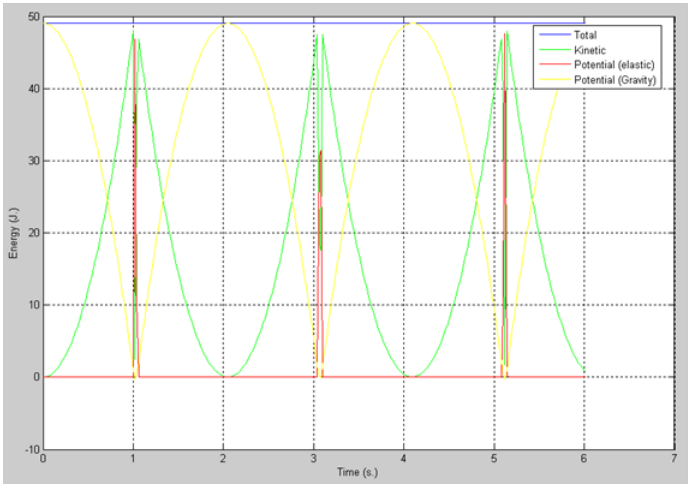


Figure 25: Energy plots for a single particle bouncing on a horizontal wall (single spring) with correction of the contact forces

This method can be extended to cases where more than one contact force is applied to the particle in the form of several springs, with different stiffness and lengths. This is exactly the case in *PFlow*, where a single solid object can have many forces coming from different contact elements, or even from other solid objects. Equation 15 can be re-written again as

$$\begin{aligned}\boldsymbol{v} = \boldsymbol{v}_0 + dt \Bigg[& (1-\gamma)\boldsymbol{a}_{0,gravity} + \gamma \boldsymbol{a}_{gravity} + \\ & + (1-\gamma)\left[\sum_i \alpha_i \boldsymbol{a}_{0,contact,i} + \sum_i (1-\alpha_i)\tilde{\boldsymbol{a}}_{contact,i}\right] + \\ & + \gamma\left[\sum_i \alpha_i \tilde{\boldsymbol{a}}_{contact,i} + \sum_i (1-\alpha_i)\boldsymbol{a}_{contact,i}\right]\Bigg] \end{aligned} \tag{19}$$

Where the sub-index i refers to the ith contact force. Equation 19 is clearer when written in the form of forces instead of accelerations (Eq. 20):

$$\begin{aligned}\boldsymbol{v} = \boldsymbol{v}_0 + dt/m \Bigg[& (1-\gamma)\boldsymbol{F}_{0,gravity} + \gamma \boldsymbol{F}_{gravity} + \\ & + (1-\gamma)\left[\sum_i \alpha_i \boldsymbol{F}_{0,contact,i} + \sum_i (1-\alpha_i)\tilde{\boldsymbol{F}}_{contact,i}\right] + \\ & + \gamma\left[\sum_i \alpha_i \tilde{\boldsymbol{F}}_{contact,i} + \sum_i (1-\alpha_i)\boldsymbol{F}_{contact,i}\right]\Bigg] \end{aligned} \tag{20}$$

Again a numerical model for the motion of a particle has been studied in one dimension, considering the gravity and an elastic contact with a horizontal rigid wall. In this case, the collision with the wall is computed by means of two different springs exerting forces against the particle when it approaches the wall. Fig. 26 shows how the energy increases drastically at some specific moments when the corrections are not used (Eq. 13). The input parameters for the computation were:

- Particle mass: 1 kg
- Drop height: 5 m
- Length of relaxed spring 1: 0.2 m

- Elastic constant for the spring 1: 2000 N/m
- Length of relaxed spring 2: 0.4 m
- Elastic constant for the spring 2: 1000 N/m
- Time step: 0.03 s

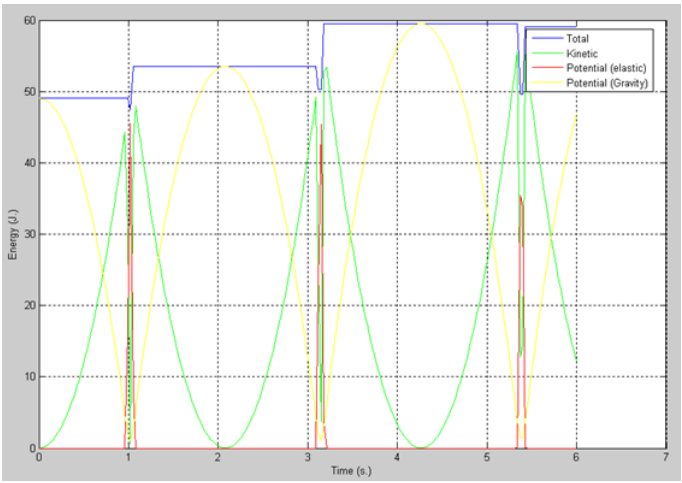


Figure 26: Energy plots for a single particle bouncing on a horizontal wall (single spring)

With the corrections introduced in Eq. 20 the energy gain is reduced to unnoticeable values again, as shown in Fig. 27.

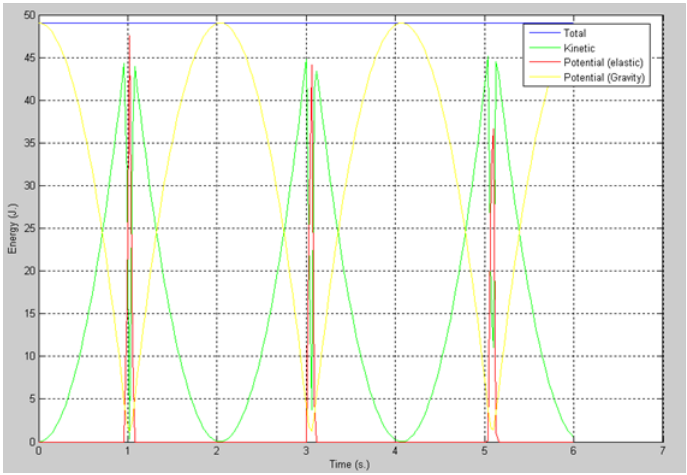


Figure 27: Energy plots for a single particle bouncing on a horizontal wall (single spring) with correction of the contact forces

3.2 Examples

Figures 28-30 in the next Subsections are snapshots of computations involving solids, fluid and contact between solids.

3.2.1 Example 1. Dragged box (2D)

Fig. 28 depicts a 2D square object dragged by a collapsed water column that creates a stream. The mesh used was certainly coarse, which allows us to see the layer of contact elements.

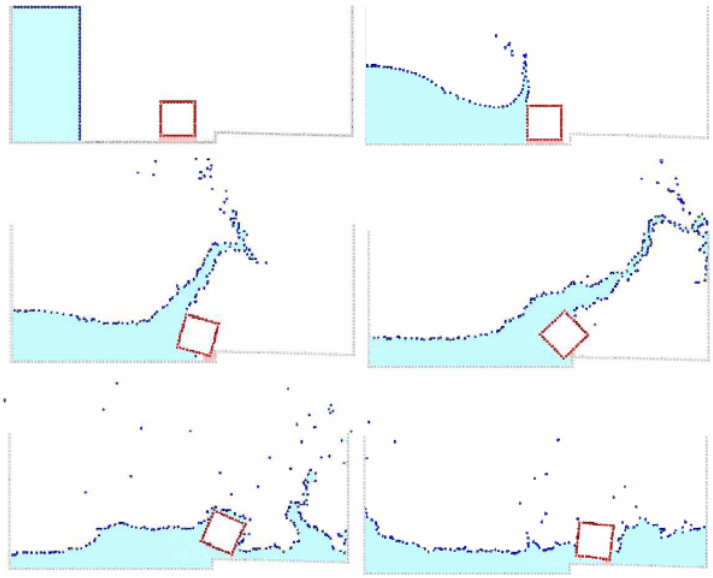


Figure 28: Square object dragged by a water stream (coarse mesh)

Fig. 29 shows a close detail of a computation similar to the previous one, in which a much finer mesh was used. The layer of contact elements is much thinner now an difficult to distinguish.

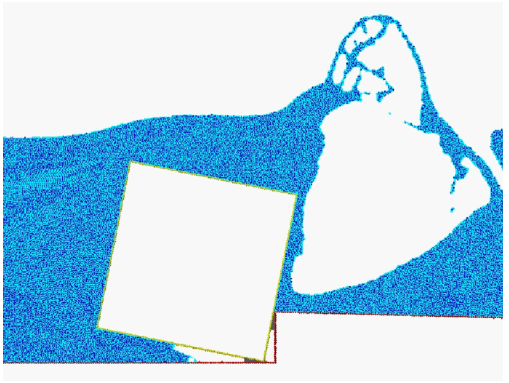


Figure 29: Detail of square object dragged by a water stream (fine mesh)

3.2.2 Example 2. Tetrapods falling into water (3D)

Fig. 30 consists of four snapshots of two tetrapods falling into a container full of water in 3D. Note that the tetrapods are half submerged during most of the simulation, and how the PFEM treats this situation in a very natural way.

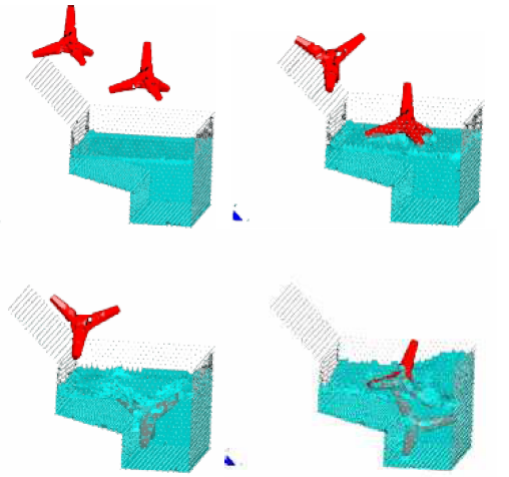


Figure 30: Two tetrapods falling into a container full of water

3.3 Conclusions

The particle finite element method (PFEM) is ideal to treat problems involving fluids with free surfaces and submerged or floating structures and bodies within a unified Lagrangian finite element framework. Problems such as fluid-structure interaction, large motion of fluid or solid particles, surface waves, water splashing, separation of water drops, frictional contact situations between fluid-solid and solid-solid interfaces, etc. can be solved with the PFEM in a very natural way. The success of the method lies in the accurate and efficient solution of the equations of an incompressible fluid and of solid dynamics using an updated Lagrangian formulation. Essential solution ingredients are the efficient regeneration of the finite element mesh using an extended Delaunay tesselation, the identification of the boundary nodes using an Alpha-Shape type technique and the simple algorithm to treat frictional contact conditions at fluid-solid and solid-solid interfaces via mesh generation. In this Thesis I have contributed to advance in several of these topics as described in the previous pages. The examples presented have shown the great potential of the PFEM for solving a wide class of practical FSI problems in engineering.

3.4 Published research

Part of the development explained in this Section and the examples were published in [4]. All the necessary coding for the innovative fluid-multibody features were coded following the *PFlow* design by the author of this Thesis.

4 Erosion

The study of the erosion process of soils and rocks under water forces has usually been developed by engineers from a global point of view, understanding 'global' as the whole section of a river, the surroundings of a bridge pile or the surf zone for a segment of beach. However, the PFEM brings a full 3D solution for the fluid velocities, which brings in a lot more information that can be used. In particular, the local vortices next to the bottom (river bed, sea bottom) can develop much lower or higher velocities than the 'average' values usually used in engineering. The Lagrangian and node-based approach of the PFEM facilitates the task of detaching nodes from the solid domain (Fig. 31).

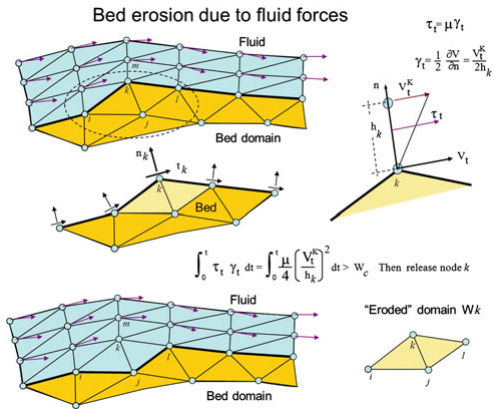


Figure 31: Detachment of a solid node due to erosion

In order to simulate the erosion processes with *PFlow*, some nodes of the solid domain, tagged as solid nodes, are converted into particles than can move within the fluid simply by changing the tag and updating the information on the node. This methodology is implemented by imposing certain criteria or conditions that must be fulfilled (Section 4.1.1). Eventually, some particles that were eroded can deposit. If they touch the bed and remain still or with very low velocities, they can be still treated as moving particles or they can be fixed to the solid bed and rejoin a the solid domain. In the first case, the fluid domain does not change, while in the second, the fluid domain is reduced because some part of it becomes solid (those elements connecting the transformed node to the rest of the solid domain become solid elements).

This Section presents the advances made by the author in the modeling of erosion processes under water forces and its consequences with respect to solids and fluids.

4.1 Developments

The PFEM can be implemented in such a way that the only difference between a solid node and a fluid node is the activation or deactivation of a couple of flags. In *PFlow*, it is extremely simple to transform one type of node into another. After the transformation, the cloud of nodes is re-meshed and the matrices to be solved are rebuilt from scratch, so the mathematical formulation for the fluid does not represent any limitation for following this procedure. Once the node type is transformed, the fluid node can be treated as a *traveler particle*, which means that it has special properties that identify it as a 'particle that was eroded from a solid'.

The general concepts about how to 'mark' nodes are simple, but subtle details on their behavior can be more complex. Most of the efforts to model the erosion process were dedicated to predict the following situations:

- the moment in time when a solid node must be transformed into a fluid node (traveler particle)
- the motion of the fluid node once it is lifted from the erodible bed
- the moment in time when a fluid node settles and must be transformed back into a solid node

These three situations are explained with more detail in the following Sections (4.1.1, 4.1.2 and 4.1.3).

4.1.1 Conditions for node detaching

The PFEM is considered a hybrid between a particles method and a continuum method, because all the information is stored in the nodes, but still the nodes form elements and the resulting mesh is actually a discretization of the continuum. Therefore, a solid node which is in contact with the fluid represents a part of the surrounding volume around the node. When it is transformed into a *traveler* node, the surrounding volume is transformed as well (see Fig. 31). The detached node actually represents the set of particles that would occupy the affected volume of solid. Considering this, the flow over the solid node under study must be capable of dragging at least one particle, but before detaching

the node it should have spent enough energy to drag the representative volume of solid associated to that node. These are the reasons why the following three Conditions were chosen to trigger the detachment of a solid node:

- Detaching Condition 1: *Activity threshold*. The fluid flow over the solid node must be strong enough to do some effect on the solid. That means that for some slow flows the solid remains unaltered. This Condition agrees with Shields' work [36] widely spread in the engineering world. Below the *Activity threshold*, the flow would never be able to erode a solid.

- Detaching Condition 2: *Accumulated wear*. While Condition 1 is fulfilled, the node will be detached if the flow acts on the solid node for long enough, taking into account that the solid node represents a certain solid volume (not just a grain), that stronger currents drag more material than the weaker ones and that a flow descending a slope can drag more than the same flow climbing the same slope.

- Detaching Condition 3: *Slope stability*. Independently from Conditions 1 and 2, any solid node that is not supported by a volume of solid which is stable enough will be detached. Slopes steeper than the angle of repose will not be stable.

Note that all three Conditions have a physical meaning. However, Conditions 2 and 3 also depend on the mesh size and shape, respectively.

Conditions 1 and 2 must be fulfilled simultaneously to detach a solid node and convert it into a *traveler particle*.

If Condition 3 alone is fulfilled, the node is also detached. More details about the implementation of these Conditions are provided in Section 4.1.4.

4.1.2 Motion of particles

The motion of a node representing a *traveler particle* follows the principles of the DEM [37] interacting with a fluid [1]. Newton's Second Law is applied to the particle, which is subject to external forces coming from the gravity and the interaction with the fluid. This is explained with more detail in Section 5.

4.1.3 Conditions for node re-attaching

Once a *traveler particle* has been moving in the fluid for some time, it might approach a solid bed. In this case, the particle might have to be settled on that

solid in order to form a new solid element. For re-attaching a node to a solid bed, the following three Conditions are proposed:

- Re-attaching Condition 1: *Low kinetic energy*. The moving particle must have a velocity relative to the bed lower than a certain threshold.
- Re-attaching Condition 2: *Mesh quality*. The elements connecting the node to the bed must be of a size similar to that of the solid domain. If those elements are considerably smaller or bigger than the mesh, the volume of solid detached by the erosion might be very different from the volume of solid recovered when re-attaching particles.
- Re-attaching Condition 3: *Minimum travel time*. The detached particle must have some time to accelerate before Re-attaching Conditions 1 and 2 are evaluated. Otherwise, they always stick the node to the solid bed immediately after detaching because the particle is moving very slowly and it is in a very good location to be re-attached.

Note that Condition 1 has physical meaning, while Conditions 2 and 3 are numerical procedures that are necessary to obtain conservative and realistic results, respectively.

All three Conditions must be fulfilled simultaneously to re-attach the traveler node.

4.1.4 Implementation

In *PFlow*, the nodes representing a solid must be aware that they belong to an erodible solid. If so, they must contain information about several parameters provided by the user, such as:

- granulometry, represented as a mean diameter or by means of a probability distribution
- density, that of a particle (not the bulk density)
- sphericity, represents how spherical are the particles forming the erodible solid
- amount of work per unit volume needed to pull a portion of solid
- minimum travel time for a traveler node (a purely numerical parameter)
- internal friction angle

With these parameters, the implementation of the Conditions presented in Sections 4.1.1 and 4.1.3 is described in the following lines.

Detaching Condition 1 In *PFlow*, the *Activity threshold* is set as a minimum parallel velocity to the erodible bed. The fluid neighbours of the candidate solid node have different velocities. The average is taken and the resultant vector is projected to a plane parallel to the solid bed ($\mathbf{v}_{parallel}$). The Condition is fulfilled if the modulus of this vector is higher than a given value ($v_{threshold}$, provided by the user).

$$\|\mathbf{v}_{parallel}\| \geq v_{threshold} \tag{21}$$

Detaching Condition 2 If the previous Condition is fulfilled (*Detaching Condition 1*), then Condition 2 is assessed, by means of the following steps.

1. Compute at every point of the bed surface the resultant tangential stress τ induced by the fluid motion. In 3D problems $\tau = (\tau_{s_n}^2 + \tau_{t_n})^2$ where s and t are the tangential stresses in the plane defined by the normal direction $\mathbf{n}$ at the bed node. The value of τ for 2D problems can be estimated as

$$\tau_t = \mu\gamma_t \tag{22}$$

with

$$\gamma_t = \frac{1}{2}\frac{\partial v_t}{\partial n} = \frac{v_t^k}{2h_k} \tag{23}$$

where v_t^k is the modulus of the tangential velocity at the node k point (i.e. $v_t^k = (v_{s_n}^2 + v_{t_n}^2)^{1/2}$) and h_k is a prescribed distance along the normal of the bed node k. Typically h_k is of the order of magnitude of the smallest fluid element adjacent to node k (Fig. 31).

2. Compute the frictional work originated by the tangential stresses at the bed surface as

$$W_f = \int_o^t \tau_t\gamma_t dt = \int_o^t \frac{\mu}{4}\left(\frac{v_t^k}{h_k}\right)^2 dt \tag{24}$$

Eq.24 is integrated in time using a simple scheme as

$${}^nW_f = {}^{n-1}W_f + \tau\gamma_t\Delta t \tag{25}$$

3. The onset of erosion at a bed point occurs when ${}^{n}W_f$ exceeds a critical threshold value W_c defined empirically according to the specific properties of the bed material.

4. If ${}^{n}W_f > W_c$ at a bed node, then Condition 2 is fulfilled and the node is detached from the bed region and it is allowed to move with the fluid flow, i.e. it becomes a traveler node.

Detaching Condition 3 The following steps define the procedure to assess the *slope stability*:

1. Nodes located on the skin of the solid, either in contact with the fluid or not, are identified.

2. For each of the identified nodes a virtual cone is created, with its vertex on this kth node and the base above the node (height is chosen as a multiple of h_k, see Fig. 32 for schematic explanation). The slope of the cone is exactly the stable slope, given by the user in the form of 'internal friction angle'.

3. If any node of the solid falls inside the volume of the cone, the Condition is fulfilled and the node is detached.

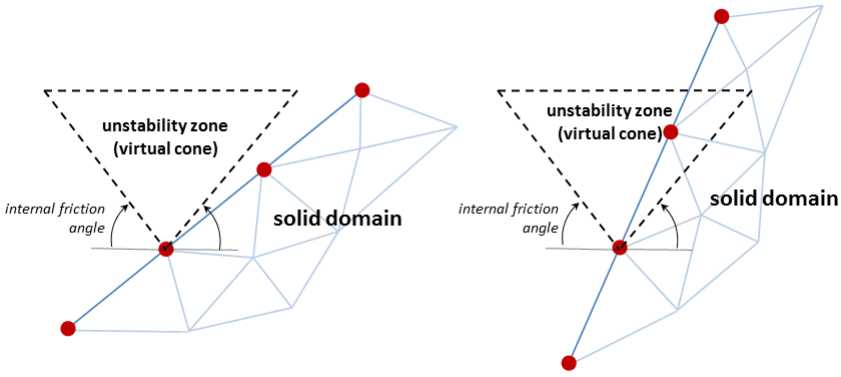


Figure 32: Evaluation of detachment Condition 3 with a 'virtual cone': a stable solid bed (left) and an unstable one (right)

Re-attaching Condition 1 The traveling node must have a low velocity. The projection of its velocity to a plane parallel to the bed, $\mathbf{v}_{parallel}$ must be smaller than a certain, small value, $v_{re-attach}$ (Eq. 26):

$$\|\mathbf{v}_{parallel}\| \leq v_{re-attach} \tag{26}$$

Re-attaching Condition 2 The traveler nodes that approach a solid and have fulfilled Condition 1 are candidates to be re-attached to the solid. Their perpendicular distance to the solid is measured, d, and compared to admissible maximum and minimum (Eq. 33):

$$d_{min} \leq d \leq d_{max} \tag{27}$$

Where d_{min} and $d_{max}$are parameters hard-coded in *PFlow*, but depend on the element size, h, at the solid nodes.

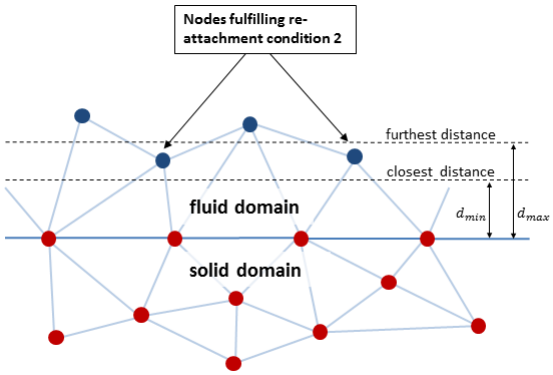


Figure 33: Scheme of 4 traveler nodes candidates to rejoin the solid, but only two fulfill the re-attaching Condition 2

Re-attaching Condition 3 As soon as a node is detached from the solid domain, an internal time counter is initiated. At each time step, the counter is updated, increasing the counter by the size of a time step.

Condition 3 is fulfilled if the time counter, $t_{counter}$ is bigger than the 'minimum travel time' provided by the user, $t_{minimum}$ (Eq. 28):

$$t_{counter} \geq t_{minimum} \tag{28}$$

4.2 Examples

Next, a collection of simple, schematic, but very illustrative 2D and 3D examples showing the potential of the PFEM formulation are presented. They are meant to demonstrate the capabilities of the *PFlow* tool to model bed erosion in complex free surface flows.

4.2.1 Example 1. 2D hill erosion

Fig. 34 shows the progressive erosion of a compacted sand domain under the action of an impacting water stream originated by a water jet. The situation is typical in sand shapes built by children in the beach and subsequently destroyed by dropping water on them. The frames in Fig. 34 show the progressive erosion of the surface of the sand domain. A kind of hydraulic jump is generated by the water jet and the sand obstacle as clearly seen in the figures. The erosion process continues until the sand domain is fully dragged by the fluid flow.

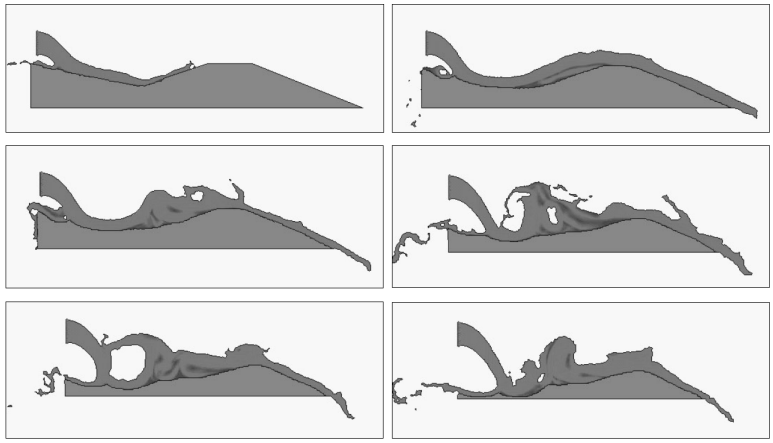


Figure 34: Erosion of a sand hill due to a water stream

4.2.2 Example 2. 3D hill erosion

The second example illustrates the erosion of an earth dam under a water stream running over the dam top. An schematic geometry of the dam has been chosen to simplify the computations. The images of Fig. 35 show the progressive erosion of the dam surface until the whole dam is wiped out by the fluid flow.

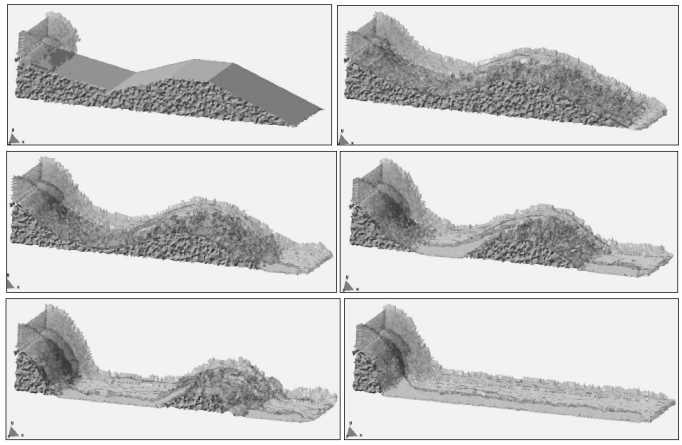


Figure 35: Erosion of a 3D earth dam due to an overspill streams

4.2.3 Example 3. Erosion around bridge pile

The next example models the progressive erosion of a river bed domain in the vicinity of a bridge pile under a water stream. Fig. 36 shows a view of the eroded bed surface at different times, until the footing can be seen. The flowing water particles are not shown in the pictures, for clarity. The erosion process continues until the bridge pile foundation is unveiled by the erosion of the adjacent bed particles. Note that the deposition of the eroded particles was not modeled in this case.

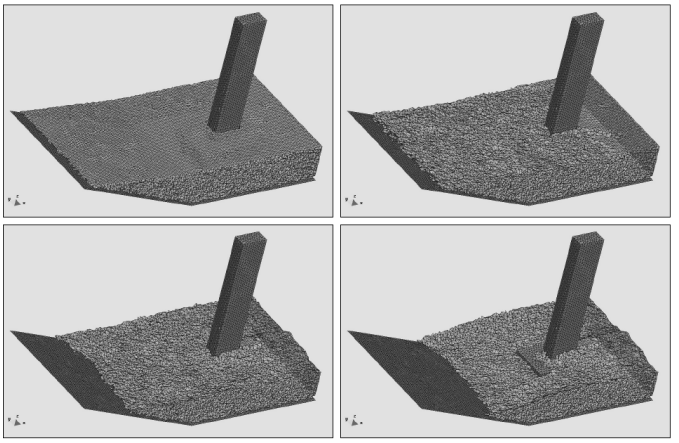


Figure 36: Evolution of the erosion of the soil in the vicinity of a bridge pile. Water particles are not shown

4.2.4 Example 4. Erosion destabilizes object

This example was chosen so as to demonstrate the effectiveness of the PFEM algorithm to combine the erosion process with the dragging of solid objects.

The pictures in Fig. 37 represent schematically a temple on the top of a mountain. The mountain is progressively eroded by a strong water stream until the temple is dragged by the fluid.

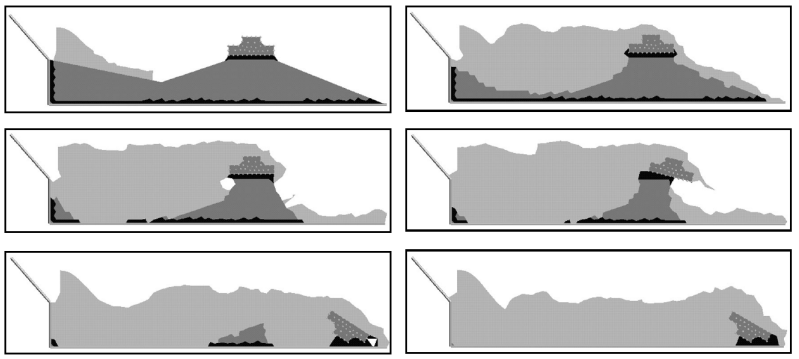


Figure 15: Erosion of a sand hill with an object on the top.

Figure 37: Erosion of a sand hill with an object on the top

4.2.5 Example 5. Erosion, transport and deposition.

This example is the only one in Section 4 in which the deposition of eroded material is modeled. The Detaching and Re-attaching Conditions (Sections 4.1.1 and 4.1.3) were active in this computation. It can be seen that the flow is capable of pulling particles from the bed, transporting them to another place where they settle down, and the solid slope collapses occasionally when it is too steep (Fig. 38). Eventually, the scouring process leads to a steady shape of the bed where the erosion rate is equivalent to the deposition rate.

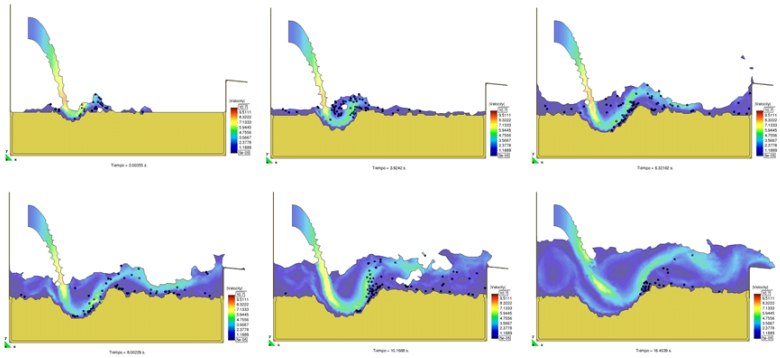


Figure 38: Erosion of a sand hill with an object on the top

4.3 Conclusions

The particle finite element method (PFEM) is a powerful technique to model bed erosion problems involving fluids with free surfaces and submerged or floating structures. Problems such as surface erosion, sediment transport and deposition, fluid-structure interaction, large motion of fluid or solid particles, surface waves, water splashing, separation of water drops, etc. can be solved with the PFEM. Essential solution ingredients are the fast regeneration of the finite element mesh using an extended Delaunay tesselation, the meshless finite element interpolation (MFEM), the identification of the boundary nodes using an Alpha Shape type technique and the simple algorithms to model onset of erosion, sediment transport and material deposition and contact conditions at the fluid-solid and solid-solid interfaces via mesh generation. In this Thesis I have contributed with new ideas and developments in these fields, as explained in the previous pages. The examples presented have shown the great potential of the PFEM for modeling bed erosion in complex free surface flows accounting for the dragging of solid objects. Applications of the sediment transport and the material deposition algorithm sketched in this Section will be expanded in Section 5.3.

4.4 Published research

The ideas about how to calculate the erosion in *PFlow* were published in one of the papers co-written by the author in [2] (see Annex: papers co-written by the author). Other developments that were published later on in [34, 35] focused on

improving the Detaching conditions and the Re-attaching conditions explained in Section 4.1. They were written under the supervision of the author of this thesis, but their content is not addressed in the present document.

5 Particles transport in fluids

The transport of particles in a fluid is a common problem of interest in engineering. Eroded parts of a solid or any other object dragged by the fluid can influence notably in the behavior of the fluid or in the force exerted by the mixture on structures. Any computation that uses the technology developed in Section 4 that includes detaching and re-attaching of nodes will also require the transport of the detached node or particle. Particles can be distinguished in two types according to their relative size to the fluid mesh:

- Big particles. The size of the particle is bigger than the size of a finite element. They can be discretized and the interaction with the fluid is modeled through a FSI strategy. This was already mentioned in Section 3.
- Small particles. The size of the particle is smaller than the size of a finite element. In terms of interaction with the fluid, the particle is represented as a point. Despite that the particle has a mass, volume and even shape, it is represented by a single point that holds all the information about it. This is the type of particles that are dealt with in this Section. This approach to the treatment of particles is also known as Discrete Element Method [37].

As a particular note, CIMNE was involved in 2012 in a research project funded by the oil drilling company 'Weatherford' about the computation of cuttings transport along a drilled well bore using *PFlow* (Section 6). In these computations, as the drill bit advances during the drilling process, the cuttings must be removed from the well bore bottom and this is done by injecting mud from inside the drilling string that exits through the drill bit nozzles and moves upwards between the drill string and the well bore walls. An intense analysis of the specific literature was done, as well as an implementation of the best methods found. The most important advances developed by the author in this field were published in a paper included in the compilation of this PhD Thesis ([1]). After the project, *PFlow* featured a new tool for particles transport using the DEM.

5.1 Developments

On one hand, the transport of disperse spherical particles in Newtonian fluids is considered as a solved problem ([38]) independently from the Reynolds number or the properties of the particle and the fluid. The usual approach is to consider

the particle as a point embedded in the fluid, where the coupling between the two phases is done through the transfer of forces. The commonly used forces are the buoyancy and the drag. Other forces that are less often considered are the Saffman lift, the Magnus lift, the virtual mass, the Basset force and the hydrodynamic torque. The *PFlow* code includes the buoyancy and the drag forces.

On the other hand, the transport of non-spherical particles in non-Newtonian fluids is a non-finished research. Actually, it is almost impossible to find drag laws for particles in non-Newtonian fluids, and when found, the literature is very specific to a type of rheology (i.e. Bingham Plastic). The definition of the Reynolds number for non-Newtonian fluids is not even unique, and it changes notably according to different authors ([48, 56]). The adaptation of some of the drag laws to a specific case of application usually leads to big errors in the estimations of the drag force and, therefore, in the terminal velocity of the particle. The errors in the results can be over two orders of magnitude. The inclusion of the non-sphericity of the particles in such approaches brings even more uncertainty to the problem.

5.1.1 Modeling of the particles

Fig. 39 shows a fluid domain containing particles of small and moderate sizes relative to the representative volume for a node. Particles are assumed to have a spherical shape in two and three-dimensions (2D/3D) and are modeled as rigid objects that undergo interaction forces due to the physical contact between a particle and its neighbors, as in the standard DEM ([37]).

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 equations (Fig. 40). These forces, as well as the mass balance equations account for the percentage of particles in the fluid, similarly as it is done in immersed approaches for particulate flows ([45, 38]). The rest of the interaction forces between fluid and particles are neglected (lift forces, virtual mass forces, drag torque, etc.) [61].

The following Sections describe the governing equations for a particulate fluid and the computation of the forces on the particles.

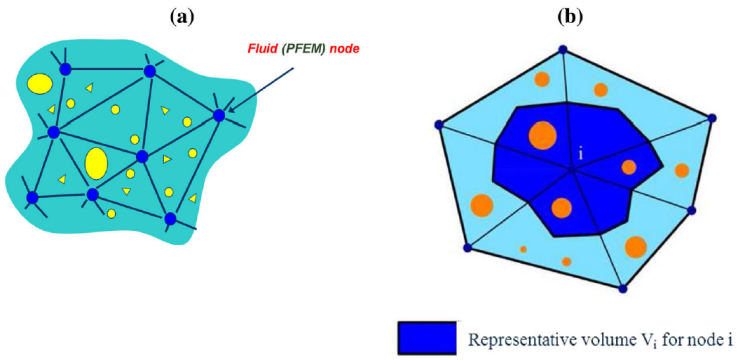


Figure 39: (a) Particles of different sizes surrounding the nodes in a FEM mesh. (b) Representative volume for a node (in shadowed darker colour)

5.1.2 Basic governing equations for a particulate fluid

Conservation of linear momentum The equation for linear momentum conservation can be expressed as

$$r_{m_i} = 0 \tag{29}$$

with

$$r_{m_i} := \rho_f \frac{Dv_i}{Dt} - \frac{\partial \sigma_{ij}}{\partial x_j} - \left(b_i - \frac{1}{n_f} f_i^{pf} \right) \quad , \quad i,j = 1, \cdots, n_s \quad in\, V \tag{30}$$

Summation of terms with repeated indices is assumed in Eqs.29 and 30 and in the following, unless otherwise specified.

In Eq. 30 V is the analysis domain, n_s is the number of space dimensions ($n_s = 3$ for 3D problems), ρ_f is the density of the fluid, v_i and b_i are the velocity and body force components along the ith Cartesian axis, respectively, σ_{ij} are the fluid Cauchy stresses, f_i^{pf} are averaged particle-to-fluid interaction forces for which closure relations must be provided and n_f is the fluid volume fraction at a point defined for each node j as

$$n_{f_j} = 1 - \frac{1}{V_j} \sum_{i=1}^{n_j} V_j^i \tag{31}$$

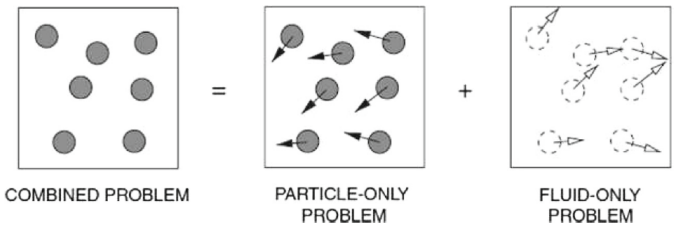


Figure 40: Immersed approach for treating the motion of physical particles in a fluid

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. Note that $n_{f_j} = 1$ for a representative fluid domain containing no particles (Fig. 39).

The fluid volume fraction n_f in Eq.30 is a continuous function that is interpolated from the nodal values in the finite element fashion.

Remark 1. The time derivative $\frac{Dv_i}{Dt}$ in Eq. 30 is computed in the Eulerian and Lagrangian frameworks as

$$Eulerian: \qquad \frac{Dv_i}{Dt} = \frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \tag{32}$$

$$Lagrangian: \qquad \frac{Dv_i}{Dt} = \frac{\partial v_i}{\partial t} = \frac{{}^{n+1}v_i - {}^{n}v_i}{\Delta t} \tag{33}$$

with

$${}^{n+1}v_i := v_i({}^{n+1}\mathbf{x}, {}^{n+1}t) \quad , \quad {}^{n}v_i := v_i({}^{n}\mathbf{x}, {}^{n}t) \tag{34}$$

In Eq. 34, ${}^{n}v_i({}^{n}\mathbf{x}, {}^{n}t)$ is the velocity of the material point that has the position ${}^{n}\mathbf{x}$ at time $t = {}^{n}t$, where $\mathbf{x} = [x_1, x_2, x_3]^T$ is the coordinates vector of a point in a fixed Cartesian system. Note that the convective term, typical of the Eulerian formulation, does not appear in the definition of the material derivative in Eq. 33 ([12]).

Constitutive equations The Cauchy stresses in the fluid, σ_{ij}, are split into the deviatoric (s_{ij}) and pressure (p) components as

$$\sigma_{ij} = s_{ij} + p\delta_{ij} \tag{35}$$

where δ_{ij} is the Kronecker delta. In this work the pressure is assumed to be positive for a tension state.

The relationship between the deviatoric stresses and the strain rates has the standard form for a Newtonian fluid ([12]),

$$s_{ij} = 2\mu\varepsilon'_{ij} \quad with \quad \varepsilon'_{ij} = \varepsilon_{ij} - \frac{1}{3}\varepsilon_v\delta_{ij} \quad and \quad \varepsilon_v = \varepsilon_{ii} \tag{36}$$

In Eq. 36, μ is the viscosity and ε'_{ij} and ε_v are the deviatoric and volumetric strain rates, respectively. The strain rates, ε_{ij}, are related to the velocities by

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \tag{37}$$

For the non-Newtonian fluids considered in this work the viscosity dependence with the strain rate is defined as

$$\mu(\gamma) = \frac{\tau(\gamma)}{\gamma} \tag{38}$$

where the expression of the shear stress, $\tau(\gamma)$, is obtained experimentally from a viscometer test. For multidimensional flows, γ is defined as the second invariant of the symmetric gradient of the velocity field, i.e.

$$(\nabla^s v)_{ij} := \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \tag{39}$$

$$\gamma = \sqrt{\frac{1}{2}\varepsilon_{ij}\varepsilon_{ij}} \tag{40}$$

Mass conservation equation The mass conservation equation for a particulate flow is written as

$$r_v = 0 \tag{41}$$

with

$$r_v := \frac{D(n_f\rho_f)}{Dt} + n_f\rho_f\varepsilon_v \tag{42}$$

Expanding the term and dividing Eq. 41 by $n_f\rho_f$, the expression of r_v can be redefined as

$$r_v := -\frac{1}{\kappa}\frac{Dp}{Dt} + \frac{1}{n_f}\frac{Dn_f}{Dt} + \varepsilon_v \tag{43}$$

where κ ($\kappa = \rho_f c^2$) is the bulk compressibility parameter, c being the speed of sound and $c^2 = -\frac{\partial p}{\partial \rho}$.

Remark 2. 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 ([12]).

Remark 3. Similarly as in Eqs. 32 and 33 the time derivative term in Eqs. 42 and 43 has different forms in Eulerian and Lagrangian frameworks, i.e.

$$Eulerian: \qquad \frac{D(\cdot)}{Dt} = \frac{\partial(\cdot)}{\partial t} + v_j \frac{\partial(\cdot)}{\partial x_j} \tag{44}$$

$$Lagrangian: \qquad \frac{D(\cdot)}{Dt} = \frac{\partial(\cdot)}{\partial t} = \frac{{}^{n+1}(\cdot) - {}^{n}(\cdot)}{\Delta t} \tag{45}$$

with ${}^{n}(\cdot)$ having the same meaning as in Eq. 33.

Boundary conditions The boundary conditions at the Dirichlet (Γ_v) and Neumann (Γ_t) boundaries with the fluid boundary $\Gamma = \Gamma_v \cup \Gamma_t$ are

$$v_i - v_i^p = 0 \qquad on\, \Gamma_v \tag{46}$$

$$\sigma_{ij} n_j - t_i^p = 0 \qquad on\, \Gamma_t \quad i,j = 1, \cdots, n \tag{47}$$

where v_i^p and t_i^p are the prescribed velocities and prescribed tractions at the Γ_v and Γ_t boundaries, respectively and n_j are the components of the unit normal vector to the boundary ([12]).

At a free surface the Neumann boundary conditions (Eq. 47) apply. These conditions are enforced at every time step.

5.1.3 Motion of particles

The motion of particles follows the standard law for Lagrangian particles. For the ith particle

$$m_i \dot{\mathbf{u}}_i = \mathbf{F}_i \quad , \quad J_i \dot{\mathbf{w}}_i = \mathbf{T}_i \tag{48}$$

where $\mathbf{u}_i$ and $\mathbf{w}_i$ are the velocity and the rotation vector of the center of gravity of the particle, m_i and J_i are the mass and rotational inertia of the particle, respectively and $\mathbf{F}_i$ and $\mathbf{T}_i$ are the vectors containing the forces and torques acting at the gravity center of the particle.

Eqs. 48 are integrated in time in order to compute the motion of the particles. An explicit Forward Euler scheme has been used with substeppping.

Substepping is necessary to avoid instabilities in the DEM solution since the fluid time steps are usually too large for the DEM solver. The information of the fluid which is projected to the particles is interpolated linearly between two steps of the fluid solution to the substep of the DEM solution.

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

$$\mathbf{F}_i = \mathbf{F}_i^w + \mathbf{F}_i^c + \mathbf{F}_i^{fp} \tag{49}$$

$\mathbf{F}_i^w$, $\mathbf{F}_i^c$ 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 The self-weight force acting on a particle can be written as

$$\mathbf{F}_i^w = -\rho_i \Omega_i \mathbf{g} \tag{50}$$

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.

Contact forces The contact forces acting on a particle emanating from other particles and walls are added up as follows:

$$\mathbf{F}_i^c = \sum_{j=1}^{n_i} \mathbf{F}_{ij}^c \tag{51}$$

where n_i is the number of contact interfaces for the ith particle.

$$\mathbf{F}_{ij}^c = \mathbf{F}_n^{ij} + \mathbf{F}_s^{ij} = F_n^{ij} n_i + \mathbf{F}_s^{ij} \tag{52}$$

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 (Fig. 41) or particle i with a wall. These forces are computed in terms of the relative motion of the interacting particles as in the standard DEM ([37]). Fig. 41 summarizes some aspects of the DEM. Fig. 41a depicts the particle i with 8 neighbor particles (j, k, l, m, n, p, q and r). Fig. 41b shows details of the contact between particles i and j: d_{ij} is the distance between centers, r_c^{ij} is the vector from the center of the particle i to the contact point between i and j, F^{ij} is the force exerted by particle j on particle i at the contact point. Fig. 41c shows the scheme of the kinematics of the contact. Both particles have a velocity ($\dot{\mathbf{u}}_i$, $\dot{\mathbf{u}}_j$) and an angular velocity (w_i,

w_j). The relative displacement of the particles at the contact point is penalized with elastic constants to avoid interpenetration between particles. Fig. 41d shows the decomposition of the relative displacement and the contact force in the normal and tangential directions at the contact point. Fig. 41e shows the linear elastic dashpot system used for modeling the mechanical behaviour at a contact point. The elastic penalty constants are K_n (normal direction) and K_s (tangential direction); C_n is a viscous parameter that provides damping to the contact; μ is the Coulomb's friction parameter. It affects the limit at which sliding between particles occurs, which follows the expression $\mathbf{F}_s^{ij} \leq \mu \mathbf{F}_n^{ij}$.

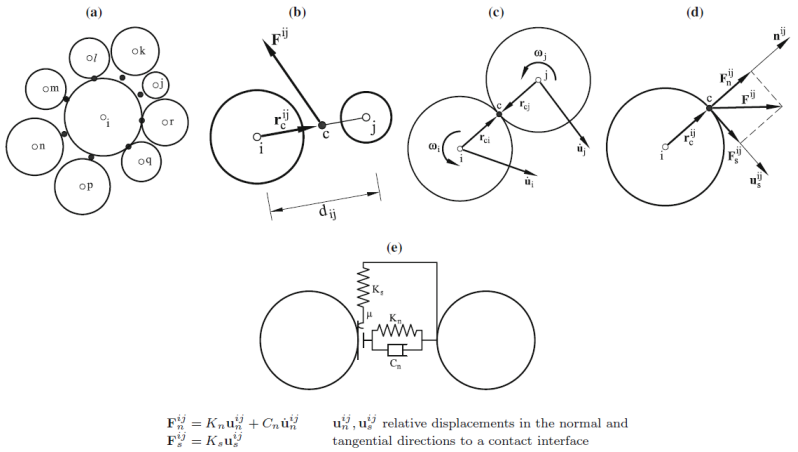


Figure 41: (a) Group of eight particles in contact with the ith particles. (b) Contact force vector for two particles interacting with a gap distance. (c) Distance vectors and velocity vector for two particle in contact. (d) Normal and tangential forces and displacement at a contact point. (e) Linear elastic dash pot system modeling the mechanical behaviour at a contact point[1, 37]

Contact radius The *contact radius* is the radius used by a particle when its contact with another particle or a wall is being computed. The value of the *contact radius,* in general, is the radius of the particle, but there is a possibility to increase or decrease its value according to other interests. In *PFlow* there is the possibility to increase the *contact radius* to a value equal to the element size. This means that, even if the particles are much smaller that the fluid mesh, the

contact between particles is established at a distance of one element size. This obviously changes the concentration of particles per element, from an accurate one to a meaningless value, but other properties are conserved correctly. For example, if a particle is detached from a solid by means of an erosion process, one of the main interests is knowing the transport of this particle. While the particulate flow remains *disperse* (particle concentration low enough to assume that the hydraulic forces do not affect other particles) the traveler particle can increase its *contact radius* to detect the walls and other particles at a distance of one element size, so the deposition generates new solid elements that have the desired size, as expressed in Section 4.1.3. This can be understood as a sort of *coarse graining [62].*

Fluid-to-particle forces The interaction force between the fluid and a particle is written as $\mathbf{F}_i^{fp} = \mathbf{F}_i^d + \mathbf{F}_i^b$, 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 follows.

Buoyancy forces

The buoyancy force depends on the volume of the particle and the gradient of pressure of the fluid:

$$\mathbf{F}_i^b = -\Omega_i \nabla p \tag{53}$$

Note that ∇p is not always aligned with gravity, but depends on the fluid dynamics.

Drag forces

The drag force is defined as

$$\mathbf{F}_i^d = -F_i^d \hat{\mathbf{v}}_i^r \tag{54}$$

where

$$\hat{\mathbf{v}}_i^r = \frac{\mathbf{v}_i^r}{|\mathbf{v}_i^r|} \quad with \quad \mathbf{v}_i^r = \mathbf{u}_i - \mathbf{v}_i \tag{55}$$

$\mathbf{v}_i^r$ 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 $\boldsymbol{F}_i^d$ is explained in the next Subsection (5.1.4).

Computation of $\boldsymbol{f}_i^{pf}$ The force term component $\boldsymbol{f}_i^{pf}$ in the right hand side of the momentum equations (Eqs. 29 and 30) is computed for each particle (in

vector form) as $\mathbf{f}^{pf} = -\mathbf{f}^{fp}$ with vector $\mathbf{f}^{fp}$ computed *at each node* in the fluid mesh from the drag forces $\mathbf{F}_i^d$ as

$$\mathbf{f}_j^{fp} = \frac{1}{V_j} \sum_{i=1}^{n_j} N_j(x_i) \mathbf{F}_i^d \quad , \quad j = 1, N \tag{56}$$

where $N_j(x_i)$ is the value of the shape function of node j at the position of the ith particle.

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 the present analysis. These forces can be accounted for as explained in [46].

Duality of the traveler nodes If the *PFlow* option of increasing the *contact radius* to the value of one element size is activated, another feature can be activated as well: the *duality of the traveler nodes*. This option forces the traveler node to have two different velocities:

- the velocity of the fluid at that point
- the velocity of the traveler particle

The second one is chosen to move the particle, as explained above in this Section. At the same time, the node representing the traveler particle is used to model the fluid and solve the FEM equations, it is, it has a fluid velocity and a pressure coming from the solution of the system of equations. The fluid velocity is ignored in terms of trajectory integration, but it is crucial for the fluid solution.

As a consequence of this duality, the PFEM mesh is distorted more than it would where there are traveler particles, but it has no effect in terms of robustness, as the mesh is rebuilt at the next time step.

5.1.4 The drag force for non-Newtonian fluids

The drag forces on particles immersed in a Newtonian fluid are well known. However, when dealing with non-Newtonian fluids different approaches for computing these forces can be followed depending on the type of fluid. Non-Newtonian fluids, for example, can be shear thickening, shear thinning or Bingham plastics, and each one of these requires a different drag law. Most drag laws require finding a suitable value for the drag coefficient (C_d) and the Reynold's number.

Drilling fluids for the oil-drilling industry usually exhibit a Herschel-Bulkley behavior ([47]). Drag laws for particles moving in drilling fluids based on a shear thickening behavior can be discarded. On the other hand, drag laws developed under the assumptions of a Bingham plastic ([48, 49, 50]) fail to predict accurately the drag force when they are applied to Herschel-Bulkley fluids. In fact, no drag laws for particles in Herschel-Bulkley fluids, or in fluids characterized by a power law rheogram are found in the literature.

Despite the lack of suitable drag laws for particles in Herschel-Bulkley fluids, some accurate estimations of the terminal velocity of the particle (i.e. the steady state velocity reached by a particle falling freely in a liquid) can be found in several papers for different fluids [53, 54]. In particular, Shah [39] proposed an estimation of the terminal velocity in power law fluids characterized by the following value of the (non linear) viscosity

$$\mu(\gamma) = K\gamma^{n-1} \tag{57}$$

where γ was defined in Eq. 39, and K and n are material parameters. Shah's method has proven to give good estimations of the terminal velocity of particles falling in drilling muds in accordance to published experiments in [50] (see Fig. 42). The terminal velocity will be used later on in this Section to generate a drag law.

Computation of the terminal velocity of the particles In the following lines the steps for estimating the terminal velocity of the particles using Shah's method are summarized (according to [39]):

1. Compute
$$A = 6.9148(n^2) - 24.838(n) + 22.642 \tag{58}$$
and
$$B = -0.5067(n^2) + 1.3234(n) - 0.1744 \tag{59}$$
where n is the power exponent in Eq. 57.

2. Obtain Shah's dimensionless parameter S_i for the ith particle as
$$S_i = (C_d^t)^{\frac{2-n}{2}} Re_t^i \tag{60}$$
where the drag coefficient for the ith particle is defined as
$$C_d^i = \frac{4}{3}\left(\frac{d_p^i g}{(v_t^i)^2}\right)\left(\frac{\rho_p^i - \rho_f}{\rho_f}\right) \tag{61}$$

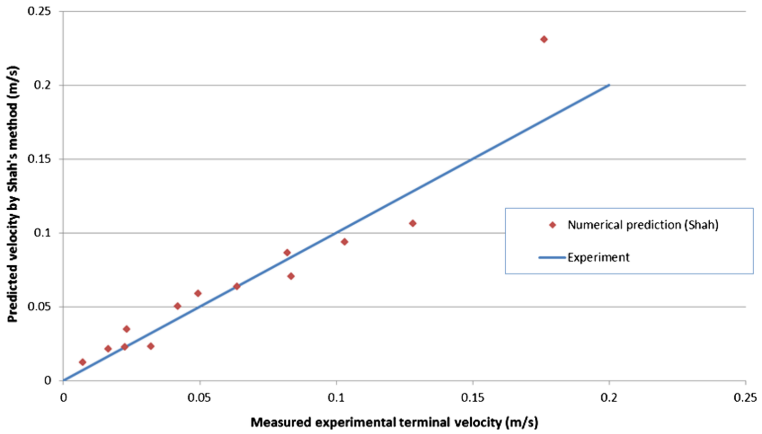


Figure 42: Relation between the experimental terminal velocity and the values predicted by Shah's method [39]. The straight line indicates the exact correlation between experimental and predicted values [50]

with

$$Re_t^i = \frac{(d_p^i)^n (v_t^i)^{2-n} \rho_f}{2^{n-1} K} \tag{62}$$

being the Reynolds number at the terminal velocity for power law fluids (other authors such as Walker *et al.* give other definition for Re in [52, 51]). In Eq. 61 g is the gravity acceleration, d_i is the particle diameter and K is the parameter in Eq. 57. Substituting Eqs. 61 and 62 into 60 Shah's dimensionless parameter can be computed as

$$S_i = \frac{1}{2^{n-1} K} \sqrt{\left(\frac{4}{3}\right)^{2-n} (d_p^i)^{n+2} \rho_f^n (\rho_p^i - \rho_f)^{2-n} g^{2-n}} \tag{63}$$

Note that Eq. 63 does not depend on v_t^i.

3. Obtain the Reynolds number for the ith particle from

$$Re_t^i = \left(\frac{S_i}{A}\right)^{\frac{1}{B}} \tag{64}$$

4. The terminal velocity for the ith particle v_t^i is computed from Eq. 62 as

$$v_t^i = \left[\frac{2^{n-1} K Re_t^i}{(d_p^i)^n \rho_f} \right]^{\frac{1}{(2-n)}} \tag{65}$$

Computation of the drag force In this Thesis a linear drag force law is proposed for cuttings moving in drilling muds of the type

$$F_i^d(v_i^r) = a_i s_i + \frac{W_i - F_i^b - a_i s_i}{v_t^i} v_i^r \tag{66}$$

where a_i is a parameter that is a function of the gel strength of the fluid and its dynamics [47], s_i is the surface area of the particle, $W_i = |\mathbf{F}_i^w|$ is the weight of the particle, F_i^b the modulus of the buoyancy force vector ($F_i^b = |\mathbf{F}_i^b|$), v_i^r is the relative velocity of the particle with respect to to the fluid ($v_i^r = |\mathbf{v}_i^r|$, see Eq. 57) and v_t^i is the relative terminal velocity obtained by Shah's method via Eq. 65. Note that for $v_i^r = v_t^i$ the equilibrium of forces for the terminal slip velocity is recovered (i.e. $F_i^d = W_i - F_i^b$). On the other hand, for $v_i^r = 0$ the initial force (gel strength) is recovered.

The gel strength is the maximum stress the fluid can withstand before showing some measurable shear rate. For those cases when $v_i^r = 0$ the drag law presents a singularity and its derivative is infinite. This means that the force can be any value between 0 and $a_i s_i$. To solve this problem, the drag force law must be regularized. In this work I have modified the drag force as

$$\bar{F}_i^d(v_i^r) = \frac{F_i^d(\hat{v}_i^r)}{\hat{v}_i^r} v_i^r \quad for \quad v_i^r \in [0, \hat{v}_i^r] \tag{67}$$

where $\hat{v}_i^r$ is a very small value.

Note that a_i is equal to the gel strength for every part of the fluid where the equivalent shear rate is zero. However, in those parts of the fluid where the shear rate is not zero, the apparent viscosity perceived by the particle decreases. In this situation, the gel strength has already been overcome and a_i must be set to zero.

Accounting for the effect of the non sphericity of the particles The sphericity coefficient (Ψ) is defined as the ratio between the skin surface of the non-spherical particle and the skin surface of a sphere with the same volume. For the same relative velocity of the particle with respect to the fluid, the drag force

grows as the sphericity decreases. Although some authors use the sphericity coefficient within a range between 0.125 and 1.0 [47] it must be pointed out that very low values of Ψ correspond actually to shapes that are far from spherical (plates, flat discs, thin sticks, ...). In those cases, the correction that Ψ brings to the equations that assume spherical shape can be very wrong. The authors do not recommended using Ψ below 0.65.

Different authors have established the dependency of the drag coefficient with the Reynold's number and the sphericity parameter for Newtonian fluids. For instance, Chien [53] proposed

$$C_d = \frac{30}{Re} + 67.289e^{(-5.03\Psi)} \tag{68}$$

while Haider and Levenspiel [55] propose:

$$C_d = \frac{24}{Re}\left[1 + \exp(2.3288 - 6.4581\Psi + 2.4486\Psi^2)Re^{(0.0964+0.5565\Psi)}\right] + \frac{73.69Re \cdot \exp(-5.0748\Psi)}{Re + 5.378\exp(6.2122\Psi)} \tag{69}$$

Both expressions and others mentioned in [50] can be used to obtain the drag coefficient for spherical particles simply making $\Psi = 1$.

In order to include the effect of the non sphericity of the particles in Shah's method, the dimensionless parameter S_i (Eq. 60) has to be modified. This has been done in this work by substituting the drag coefficient by an equivalent one which accounts for the sphericity effect. For that purpose, the following parameter is used:

$$\varphi^i = \frac{C^i_{d,non-sphere}}{C^i_{d,sphere}} \tag{70}$$

where φ^i is the ratio of drag coefficients for a particle in Newtonian fluids computed via Eqs. 68 or 69.

Taking into account that C_d in Eq. 60 corresponds to an spherical particle, Shah's dimensionless parameter is re-written using Eq. 70 as :

$$S_i = \left(\frac{C^i_{d,non-sphere}}{\varphi^i}\right)^{\frac{2-n}{2}} Re^i_t \tag{71}$$

Eq. 61 holds for both spherical and non-spherical particles. For non-spherical particles d^i_p is the diameter of the sphere with the same volume. Note that A and B were obtained for spherical particles in Eqs. 58 and 59. This

is why C_d^i must be divided by φ^i in order to compute the correct (equivalent) expression of C_d for a spherical particle.

The two-step procedure implemented in this work to compute the drag force for non-spherical particles is as follows:

1. Compute the terminal velocity for the particles using the standard Shah's method (Eq. 65) where d_i is the equivalent diameter for the non-spherical particle.

2. Compute the drag coefficient for spherical ($\varphi = 1$) and non-spherical ($\varphi \neq 1$) particles via Eqs. 68 or 69 using the Reynolds number of Eq. 64.

3. Compute the value of φ^i at the obtained Reynolds number (Eqs. 64 and 68).

4. Update S_i as $S_i^2 = S_i^1(\varphi^i)^{\frac{2}{2-n}}$, where S_i^1 is the value of Shah's parameter computed using Eq. 63.

5. With the updated Shah's parameter, perform the rest of operations in Eqs. 64 and 66.

The terminal velocity and the drag force obtained after this procedure includes the effect of the non sphericity of the particle.

5.2 Examples

5.2.1 Example 1. Vertical annulus transporting cuttings

The first problem concerned the study of the transport of cuttings in drilling muds in a vertical wellbore with a centered non-rotating drill string. Numerical results for this problem were obtained with the Lagrangian formulation presented in this work and the PFEM. The average velocity of the particles at a section of the annulus was measured and compared to the average fluid velocity. Non-spherical particles were considered. The drag force was computed as explained in Section 5.1.4using the sphericity correction of Eq. 69 [55]. Results are plotted in Fig. 43, where experimental data [40] is also shown for comparison. Table 1 shows the rheological properties of the fluids used for defining the viscosity function (Eq. 57).

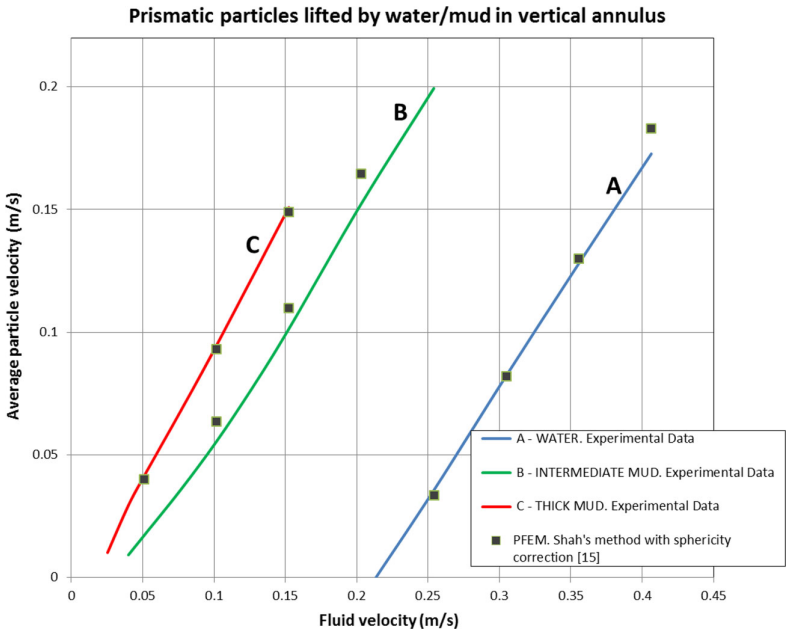


Figure 43: Average velocity of cuttings velocity vs. fluid velocity for different fluids in a vertical annulus. Lines: Experimental data

	n	K (Pa s^n)	ρ (kg/m^3)
Thick mud	0.33958	3.15275	1030
Intermediate mud	0.37826	1.7637	1030
Water	1	10^{-3}	1030

Table 1: Rheological properties of the fluids used in Fig. 43

Fig. 44 shows results of the motion of the cuttings in a centered drill string, rotating at 100 rpm, which causes the cuttings to move upwards faster. The fluid is an intermediate mud with the following rheogram characteristics $n = 0,37826$ and $K = 1,7637$ Pa s^n (Eq. 57). Results were obtained using the *PFlow-Code* developed in this Thesis. The boundary conditions used were the imposed velocity at the inlet, non-slip condition on all walls ($\bar{v} = 0$) and free surface at

the outlet ($\sigma_{ij} n_j = 0$). The free surface is kept at the same location by removing the nodes crossing the outlet.

The particles have the following characteristics: diameter (4.96 mm), sphericity (0.76766), density (2000 kg/m^3}). These characteristics correspond to particles with the shape of a brick with dimensions $1/8 \times 1/4 \times 1/8$ inches, which have been treated as spheres. It must be clarified that the experiments carried out by Sifferman *et al.* [40] were done with fluids with a density of 8.6 lb/gal (1030 kg/m^3) and simulated denser fluids with lighter particles.

The DEM contact properties were chosen to just prevent particles from penetrating each other. For this purpose the parameters used were $K_n = 3 \cdot 10^5$ N/m, $K_s = 6,5 \cdot 10^4$ N/m, a Coulomb friction coefficient of $\mu = 0.3$ and the critical damping parameter.

The method to obtain the drag force for non-Newtonian fluids is the one explained in Section 5.1.4. The example in this Section is a vertical wellbore where the slip velocity of the cuttings is close to the terminal velocity, so it is an example well suited for the method. However, the velocity profile in the annulus must be computed accurately and the distribution of the particles, affected by its horizontal velocity, must also be accurate to match the experimental results. Note that the concentration of particles in the section of the annulus is not imposed, but every single particle is injected in a random position of the inlet, and only after some time of ascending motion they reach a steady position in the section. Measurements of the average velocity of the particles were taken close to the outlet.

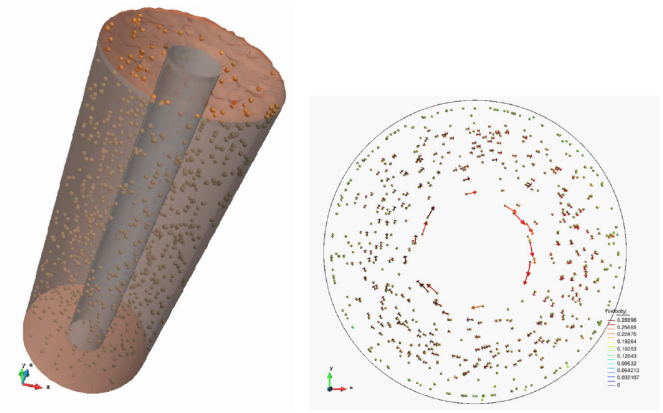


Figure 44: Global view (left) and top view (right) of particles flowing at a rate of 1500 particles/(m^{2}s) within a vertical annulus. The centered drilling pipe rotates at 100 rpm

5.2.2 Example 2. Inclined and horizontal wellbores with cuttings

PFlow has been used for the study of the transport of cuttings in inclined and horizontal wellbores that can lead to the formation of particle beds (Fig. 45). The high concentration of particles does not affect the stability of the fluid solver until the size of the cuttings equals the fluid mesh size. The fluid used is the same *intermediate mud* as in Section 5.2.1. Results in Fig. 45 were obtained using *PF*low.

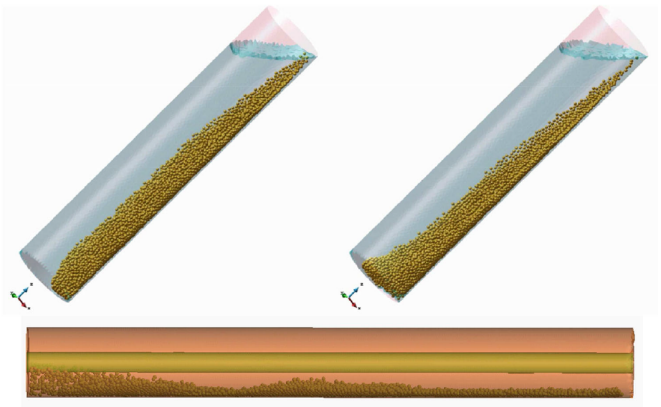


Figure 45: Global view of an inclined (above) and horizontal (below) annulus carrying mud and cuttings at a rate of 1500 particles/(m^{2}s). The horizontal annulus includes a centered non-rotating drilling pipe

5.2.3 Example 3. Tooljoints and elbow.

The formulation presented has been tested with more complex geometries, like a section of the wellbore with a tool joint (Fig. 46a) and a curved wellbore with a rotating drill string (100 rmp) (Fig. 46b). The fluid properties are the same as in Section 5.2.2. All the computations for these problems were carried out with *PFlow*.

In some of these problems, particle beds can be formed. In such conditions, the drag forces are no longer accurate, since they are based on the assumption that the particles are disperse. This method should therefore used carefully, as no estimations of the error has been obtained in dense packing conditions. However, the examples shown prove the stability of the method in those situations.

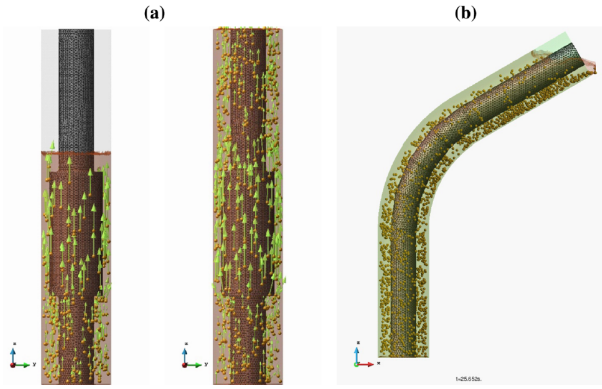


Figure 46: Simulation of drill cuttings transported in straight and curved wellbores by mud as a result of a drilling operation. Arrows denote the velocity vector of each particle

5.3 Fluid-Soil-Solid Interaction

This Section brings together all the technology, methods, algorithms and strategies mentioned in the previous Sections of this thesis. This is done just showing a test case that was run with all the features of the *PFlow* tool: the fluid solver interacting with solids that can move freely, that can collide with each other, that can be eroded, and the eroded parts travel a certain distance until they settle and form new solids.

5.3.1 Examples

The test case shows the front of a flood, which is dragging large objects (cars, barrels) and passes through a notch in a soil dyke (Fig. 47). The erosion process of the dyke (modeled with the technologies described in Section 4) modifies the shape of the boundaries. The floating objects hit the dyke and hit with each other. The eroded parts of the dyke get dragged as small particles (transport modeled with the technologies described in Section 5) until they settle and form a new, smaller dyke, which is eroded again later on.

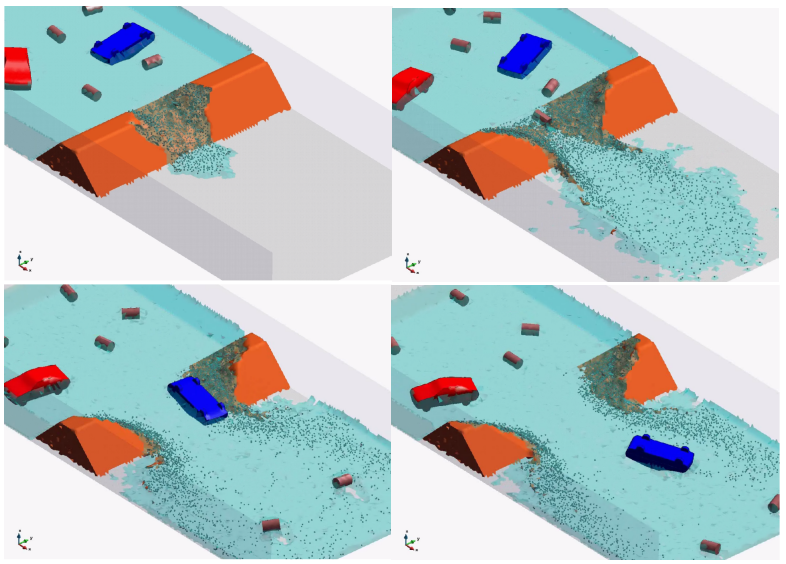


Figure 47: Dyke eroded by water current dragging particles with sizes ranging from small stones to cars.

5.4 Conclusions

A FEM-DEM technique has been developed in this Thesis to solve the transport of particles in non-Newtonian fluids which can be applied to wellbores full of circulating fluid, typical of the drilling industry. I have proposed a procedure for computing the drag force on the particles for non-Newtonian fluids using predictions of the terminal velocity available from the existing literature. The procedure has been extended to non-spherical particles, treating them as spheres in terms of contact forces but correcting the drag force according to the shape of the particles.

The usefulness of the numerical method proposed for studying the motion of the drill cuttings in vertical wellbores has been validated and the applicability and stability to other non-vertical and more complex configurations have been pointed out.

The coupling procedure is actually not dependent on the method used to solve the equations for the fluid. Therefore, it can be applied to other existing CFD methods, such as Eulerian Finite Element Method, the Finite Volume

Method (FVM) [58], the Lattice-Bolzmann Method (LBM) [59] or the Smoothed Particle Hydrodynamics (SPH) method [60]. This technique can also be combined with the erosion estimation method explained in Section 4.

Section 5.3 is the culmination of the work by the author devoted to the PFEM and the implementation of the *PFlow-code*. The examples shown involve all the technologies developed since the beginning of author's research and they are used simultaneously. The possibilities of such software are undoubtedly extense, and prove the great potential of the PFEM as an approach that allows the resolution of many engineering problems.

5.5 Published research

One of the annexed papers in the present PhD Thesis ([1]) explains with detail the advances achieved in this direction and proposes a drag law for non-Newtonian fluids and non-spherical particles. The starting point for the developments in this paper was a paper written by Shah et al. in 2007 ([39]). In this paper, a procedure is provided to estimate accurately the terminal velocity of particles falling in Power Law fluids. The author of the present work first starts validating the predictions by the approach published by Shah et al. (see Fig. 42) and then proposes a drag law based on these reliable estimations.

6 Other applications of *PFlow*

The *PFlow* developments have been used in many competitive or private research projects along this Thesis. A brief description of them is presented in this Section. The reader will observe that the applications of *PFlow* are multiple and in a wide variety of fields that have not been mentioned in the rest of this document.

6.1 Numerical computation of an unmanned underwater vehicle (NICOP)

In this project, funded by the Office of Naval Research (ONR) in the U.S., the goal was to compute the motion of self-propelled mechanical device in large domains. The need of a extremely fine mesh around the fins required a progressive refinement of the mesh. The whole domain, even with a coarse mesh far from the vehicle. The computational cost was too expensive and only the close field around the UUV was calculated (Fig. 48).

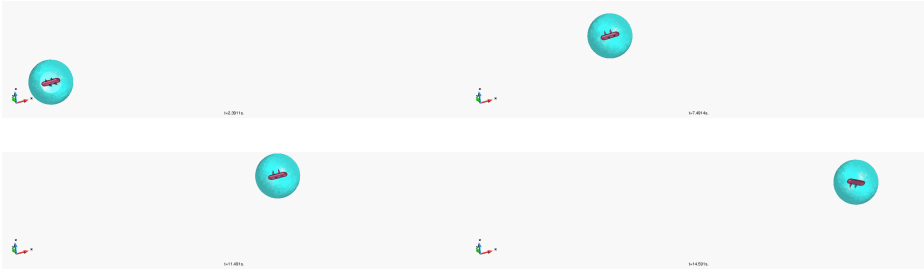


Figure 48: Unmanned underwater vehicle in large domain

Much detail was needed near the fins, due to the sharp edges of their design. Mesh refinement was used next to the body of the UUV (Fig. 49) in order to obtain engineer-useful results, like the pressure field or the velocity field (Fig. 50).

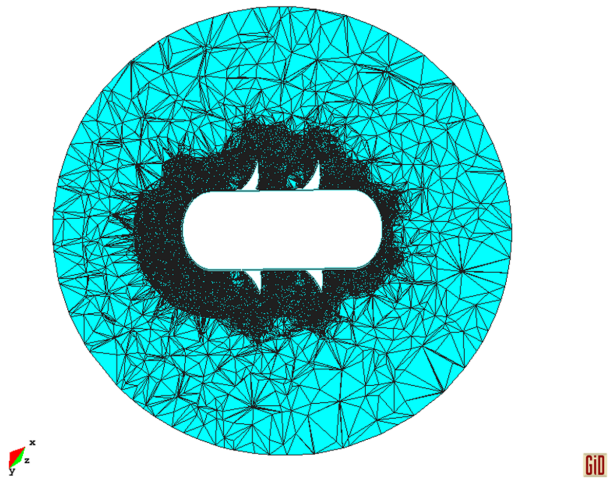


Figure 49: Mesh refinement (cut) around the UUV

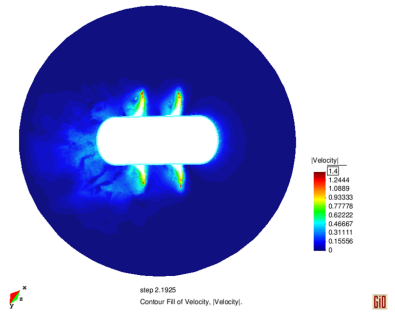


Figure 50: Velocity field around the UUV

6.2 Safety assessment in breakwaters under construction (SAYOM)

The project included numerical computation of:

- the motion of concrete blocks of a breakwater under heavy waves (Fig. 51)

- the dynamics of concrete caissons during their transport and sinking, stabilized by cables (Fig. 52)
- the effect of overtopping waves on trucks operating on a breakwater during its construction (Fig. 53)
- the erosion of an experimental model of a submerged, unprotected breakwater under a set of 12 minutes of irregular waves (Fig. 54)

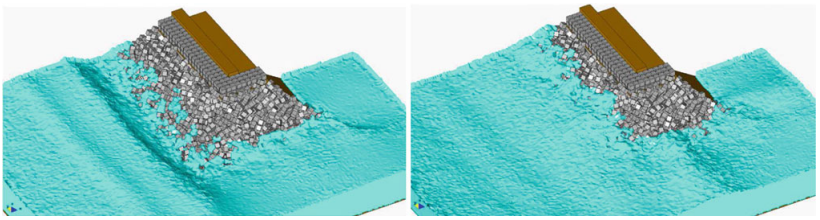


Figure 51: Waves hitting the breakwater under construction (protected int winter)

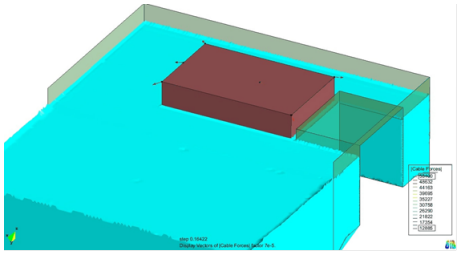


Figure 52: Concrete caisson about to be sunk. The vectors on the corners represent the forces exerted by the stabilizing cables

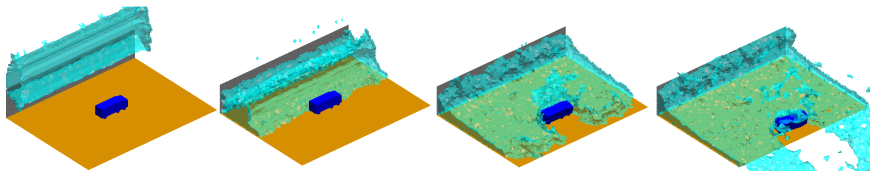


Figure 53: Effect of an overtopping wave on a truck operating on a breakwater

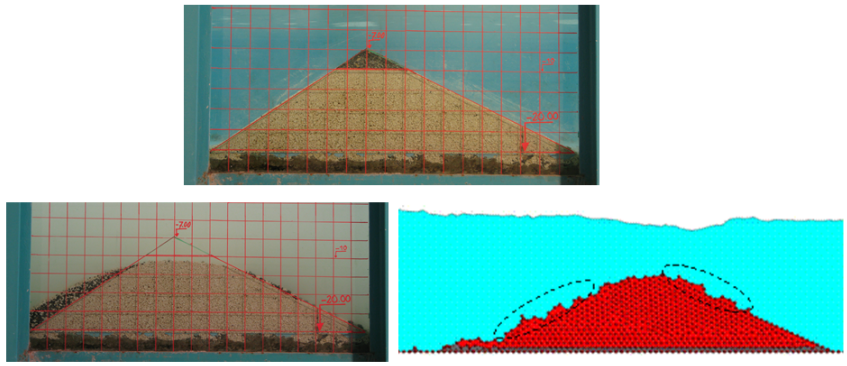


Figure 54: Calibrated computation of the erosion of a submerged, unprotected breakwater. Experiment (top, original shape, and bottom left, after erosive action) vs. computational results (right).

6.3 Numerical prediction of the energy dissipated by a pendular breakwater (DILAPE)

This project consisted on estimating the effect of a pendular wall acting as a breakwater (Fig. 55) with the goal of dissipating energy from the waves. The water level at a wall behind the oscillating wall was measured and compared with the effect of a locked wall. After several computations it was proven that the net effect of the moving wall was negligible, as no energy is extracted from the system just by using a free oscillating pendular wall. Many computations were necessary to design alternatives to the initial proposal, including virtual (coded) pistons extracting energy from the motion or inverted pendula.

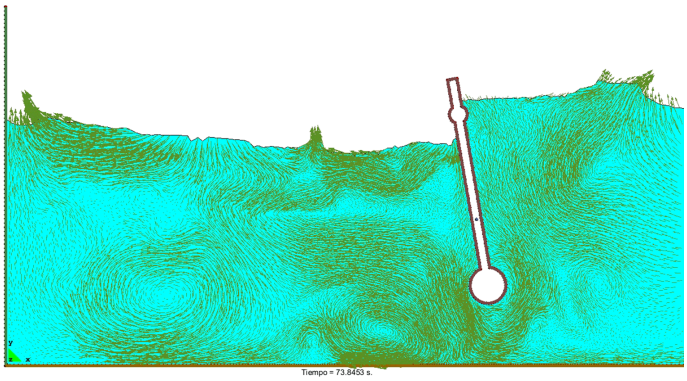


Figure 55: Pendular breakwater

6.4 Numerical estimation of the air demand in bottom outlets of dams (VADIVAP)

The project consisted on estimating the air demand after the main valve of the bottom outlet of a dam (Fig. 56). Computations with two fluids were carried out with the *PFlow*. Part of the outcomes of this project was published in a paper co-authored by the author of this Thesis ([41]).

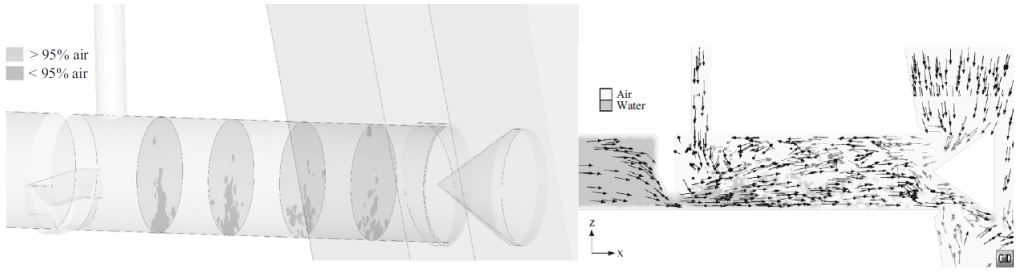


Figure 56: 3D computation of the bottom outlet (left). 2D cut of the same computation (right). Figures obtained from [41]

6.5 Numerical computation of the transport of drill cuttings in well bores with non-Newtonian fluids and non-spherical particles

The project consisted on developing a numerical method and the corresponding software capable of modeling the hole cleaning. It is, transport the cuttings from the bottom of the wellbore to the surface. Apart from the process followed to reach a satisfactory formulation and a validation of the results (Section 5), several critical situations were considered, like the tooljoints (Fig. 57) and curved annuli (Fig. 58) with rotating drill strings.

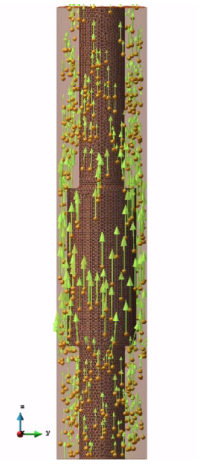


Figure 57: Particulate flow around a tooljoint in the drill string

Figure 58: Particulate flow along a curved annulus with a rotating drill string

6.6 General research on the transport of particles in fluids (SAFECON)

This project pushed the *PFlow* developments to new limits, with computations where big particles, small particles and erosion processes were taking place. Part of the outcomes of this project are the developments explained in Section 5.3.

6.7 Analysis of failure of the 'Prat' Dock in the Barcelona harbor. Numerical study of the liquefaction of the hydraulic filling.

The spontaneous liquefaction caused the failure of the whole 'Prat' Dock, moving the concrete caissons seawards over 90 m. (Fig. 59). The liquefaction was simulated by a drastic change in the viscosity of the fluid. Many computations were carried out under different assumptions about the liquefaction initiation point or the initial shape of the sea bottom. The computational results were later used by experts in the field to determine the exact source of the dock failure.

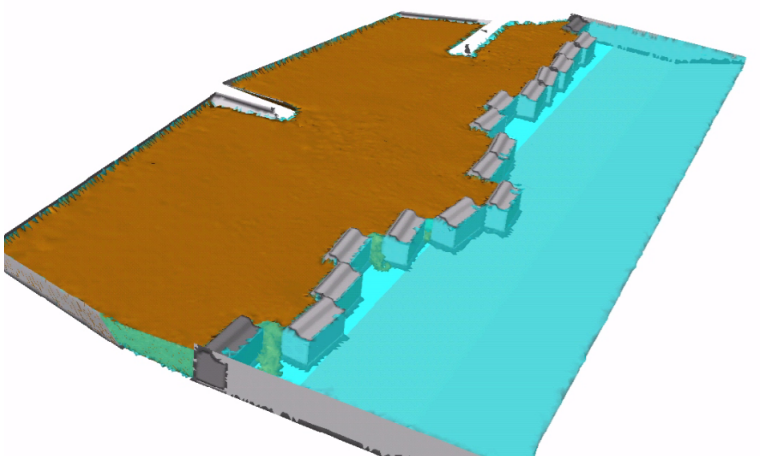


Figure 59: Failure of the Prat Dock. Situation of the caissons after the failure

7 Research works related with this thesis

The works listed in this Section contain research related to this thesis that was done in collaboration with this author, either in the form of supervision or in collaboration. Journal papers are excluded from this list, as they are included in the References Section and they have already been cited several times in the text.

Works supervised by the author in the academic context

- Güell i Pons, Maria (2007) Modelling sediment transport: short introduction and validation with PFEM. Master's Thesis at Universitat Politècnica de Catalunya.
- Calderer Elias, Ramon (2007) Aplicació del mètode dels elements finits i partícules (PFEM) al problema de transport de sediment. Tesina (Bachelor degree thesis) at Universitat Politècnica de Catalunya.
- Sancho Marín, Eliseo (2008) Validación del PFEM en problemas de cuerpos sumergidos. Tesina (Bachelor degree thesis) at Universitat Politècnica de Catalunya.
- Hospital Bravo, Raúl. (2009) Posibilidades del método de elementos finitos y partículas (PFEM) en problemas de interacción fluido-estructura en flujos incompresibles. Tesina (Bachelor degree thesis) at Universitat Politècnica de Catalunya.
- Latorre Sánchez, J. Salvador (2009) Estudio del impacto del oleaje sobre diques de bloques en puertos utilizando el método de elementos finitos y partículas (PFEM). Tesina (Bachelor degree thesis) at Universitat Politècnica de Catalunya.
- Rodríguez Prat, Francisco (2013) Posibilidades del método de elementos finitos y partículas (PFEM) en problemas de erosión de suelos granulares. Tesina (Bachelor degree thesis) at Universitat Politècnica de Catalunya.

Works where *PFlow* was used

- Larese De Tetto, Antonia (2006) On the application of particle finite element method (PFEM) to problems in civil engineering. Tesi di laurea at Università degli Studi di Padova.

8 Summary of research developments and conclusions

Section 5.3 shows examples which is almost impossible to reproduce with standard numerical tools. It involves Fluid Dynamics, Fluid-Structure-Interaction, multi-body contact, erosion, particles transport and deposition. Even with all these couplings, almost the totality of the CPU time is consumed by the fluid dynamics solver. As stated in this document, the computational cost of the rest of the tasks is almost negligible because they take advantage of all the operations the PFEM already requires (basically the re-meshing step). This is one of the most important conclusions of the work during the author's research. A PFEM code allows a large variety of problems coupled all-to-all with little extra computational time versus its simplest version with just fluid dynamics. However, a list of many other conclusions follows in the next Sections.

8.1 Fluid solution

Papers [8] and [5], among others, demonstrated the good agreement of the solution given by the fluid solver used compared to experimental results. The PFEM is useful as a CFD tool for the study of free surface flows.

8.2 Fluid-multibody-interaction

As presented in the paper [4], the Lagrangian approach used for the fluid solver allows the imposition of exact Dirichlet velocity boundary conditions at the interface with the structure. No intersection or mapping must be computed and no enrichment is needed for the fluid element formulation.

Paper [4](one of the annexed papers) details how to solve the contact between rigid bodies by means of a penalty method. The detection of the contact takes advantage of the frequent re-meshing process and a minimal cost is added to detect the contact elements.

Both the contact between solids and the contact between the fluid and the solid are treated in a very simple, natural way that actually works.

8.3 Erosion

The node-based formulation of the PFEM allows the transformation of a node representing part of a rigid body into a particle that travels inside the fluid. Details can be read in the paper [2], where a model is proposed. The PFEM is

capable of modeling the erosion processes, the transport of the eroded material and the subsequent deposition. The obtained results can be accepted as qualitatively good, while the accuracy strongly depends on the calibration stage (see Section 9.3 for more details).

8.4 Particles transport

As a result of the erosion, or coming from other sources, particles are suspended and transported by the fluid. A paper was published on this topic [1] for non-Newtonian fluids and non-spherical particles. Thus, it is possible to carry out computations of particles dragged by non-Newtonian fluids in certain ranges, like the solution proposed in Section 5, which is limited to mud-type fluids.

9 Future lines of research

Many years of work spent in the development of *PFlow* (and PFEM in general) has brought a very powerful tool for computing engineering problems related to fluids, moving bodies and particles, but they have also evidenced several issues and lacks that are listed in this Section. It could be a source of upcoming research in the PFEM world and they can also be understood as a recommendation of future developments by the author.

9.1 Fluid solution

Even though the results were very satisfactory in general, the conservation of volume proved to be extremely dependent from the time step. Bigger time steps allowed fast computations, but much fluid volume was lost. Alternative formulations can be developed in order to reduce the volume losses, as done in [33] and [13].

The *PFlow* code, based on the PFEM, has demonstrated to be an extremely versatile method and an accurate tool for those problems involving free surface flows and fluid-structure-interaction. The main concepts of

- Using a Lagrangian approach
- Re-meshing to avoid element distortion
- Alpha-shape method to detect the free surface

are enough to enable the solution of many problems that would be unapproachable (or hard to solve) by other means. However, the PFEM presents some issues that should be considered:

- In *PFlow*, the Delaunay Triangulation is a bottleneck due to the parallelization restrictions. Some authors claim to have developed parallel computing techniques for this algorithm for shared memory ([19, 20]). This is something to be studied and tested. Some other authors even claim to have developed it for hybrid (shared/distributed memory) parallelism in [42], which would be an interesting jump forward to be done in *PFlow* as well.
- The re-meshing process, even if it only consists on a diagonal swapping, does not conserve momentum or incompressibility. This can generate local problems in those parts of the domain where large strains occur and small

time steps are used. Large strains provoke frequent diagonal swapping, and when the equations try to enforce incompressibility in small time steps, some peaks of negative or positive pressure can appear at the nodes nearby. Despite the time dedicated to solve this issue, no satisfactory solution has been found by the author.

- The re-meshing process and the Alpha-Shape Method change the total volume of fluid between time steps. However, it is possible to tune the re-meshing input parameters in such a way that no losses or gains are accumulated in time. It is not an exact calibration, and a certain dependency remains, which is dependent on the specific case run.

- Mass conservation is typically a problem of the CFD solvers, and the formulations used in the PFEM are not an exception. Other researchers have been investigating additional formulations with better properties in terms of mass conservation. Examples of these publications are [33] and [13]. Considering that the mass losses are highly dependent on the time step, it is an open issue if these formulations are better in terms of computational cost: a cheap, less accurate scheme could use a smaller time step to achieve the same accuracy as an expensive, very accurate scheme.

9.2 Fluid-multibody-interaction

There are several issues that were detected in the fluid-multibody Interaction:

- Every time a fluid sub domain gets isolated from the rest of the fluid due to multiple contacts between solids (this happens commonly in 2D and much less in 3D) the pressure is not calculated properly in that isolated sub domain. This is because a Dirichlet pressure condition is needed as a pressure reference. When a sub domain has no communication with the free surface or with some point with a known imposed pressure, even though the pressure gradient is calculated correctly, the actual pressure values yielded by the solver can be very far from the values in nearby points. This makes the fluid-structure interaction completely unstable in these cases, as a consequence of the incompressibility hypothesis. A compressible formulation, or a slightly compressible formulation, would probably be a solution, but is something that has should be investigated in the PFEM.

- The contact elements between two solids that are submerged present another problem: the fluid cannot be calculated correctly inside them be-

cause there are no free degrees of freedom (the motion of all nodes is imposed by the kinematics of both solids and all nodes have a Dirichlet velocity condition). This means that the submerged solids that get in contact and form a layer of contact elements are not surrounded by a correct field of pressure. Therefore they get usually stuck and never separate from each other again unless strong forces move them apart. In order to overcome this problem, several possibilities are under study, such as slip velocity conditions in the contact elements or computing a smooth propagation of the pressure (Laplacian) inside the contact elements. An experimental version of the latter approach has been used successfully in *PFlow* recently.

- The algorithm for the contact between bodies published by the author in [4] was further developed by Carbonell in his PhD Thesis ([27]). Among other improvements, the visible gap between solids was reduced to a much smaller distance.

The layer of contact elements represents a visible error of the size of the element. This was addressed already in the paper [11], that was published after the publication included in this work ([4]).

9.3 Erosion

There are several issues that were detected in the modeling of erosion with *PFlow*:

- The solid was considered a rigid body always. However, it is well known that some erosion processes trigger landslides. For that, the solid should be computed as elastic-plastic, so that shear bands could be captured.

- The conditions established for detaching or re-attaching nodes can reproduce accurately many erosion and sedimentation processes if a previous calibration process is carried out. So far it has not been possible to get accurate results without calibration, using just parameters of the fluid or the soil (or erodible solid in general) but it would represent a big step forward if achieved.

- The second Condition for Re-attachment of eroded material is an energetic criterion based on the shear stress [44]. It can be noted that the element size h_k appears in Equation 24. However, this equation gives an accurate estimation of the work *only* if the mesh is fine enough to capture the actual

shear stresses occurring close to the walls (Dirichlet velocity condition). When the mesh is not fine enough, similar jumps in the velocity field can appear for different layers with one element in thickness, regardless of the element size. This means that the solution is strongly mesh dependent.

9.4 Particles transport

The node-based particles transport limits the CPU consumption, but can lose much detail about the behavior of the particles. An embedded approach should be developed where the particles are no longer nodes of the mesh, but they are moving freely over the fluid mesh. This would allow accumulations of particles in each element, and possibly increasing the computational cost a lot.

- The formulation detailed in the annexed paper [1] assumes that the particles are nodes of the fluid mesh. Even though their motion responds only to the dynamics of the particle, the location of the particle is used for computing the fluid as well. This means that particles cannot get as close as they should, or the fluid mesh could end up very distorted. Then, the contact between particles occurs at a distance which is bigger than it should, in the same fashion as in the *coarse graining* methods (sometimes called *upscaling*). Therefore, all the possible drawbacks of these methods are possible drawbacks of *PFlow*, too. In order to solve this issue (if it was so for specific cases), the particles could be independent from the fluid mesh, embedded in the elements and not sharing the position with the fluid mesh nodes. If this approach was used, the same interaction forces described in this Thesis (Section 5) could be used.

- Apart from the hydrodynamic interactions considered in [1], other forces and moment can be added to the fluid-particle interaction, such as the Magnus lift and the Hydrodynamic torque and the Basset forces.

- The density of particles can affect the fluid equations themselves, and they should be modified when beds are formed and the flow is more similar to a flow in porous medium rather than a free flow.

- The drag forces also change for high density of particles.

- The shape of the particles can be considered in order to change the drag force, not only as a sphericity factor but as a term that depends on the orientation of the particle with respect to the fluid velocity.

References

[1] Celigueta, M. A., Deshpande, K. M., Latorre, S., & Oñate, E. (2016). 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, 263-276.

[2] Oñate, E., Celigueta, M. A., & Idelsohn, S. R. (2006). Modeling bed erosion in free surface flows by the particle finite element method. Acta Geotechnica, 1(4), 237-252.

[3] Oliveira, T. C. A., Gonzalez-Marco, D., Sánchez-Arcilla, A., Celigueta, M. A., & Onate, E. (2007). Generation and propagation of water waves in a two-dimensional numerical flume. Computational Methods in Marine Engineering II, 241-244.

[4] Onate, E., Idelsohn, S. R., Celigueta, M. A., & Rossi, R. (2008). Advances in the particle finite element method for the analysis of fluid–multibody interaction and bed erosion in free surface flows. Computer Methods in Applied Mechanics and Engineering, 197(19), 1777-1800.

[5] Oliveira, T. C. A., Gironella, F. X., Sanchez-Arcilla, A., Sierra, J. P., & Celigueta, M. A. (2009). Nonlinear regular wave generation in numerical and physical flumes. Journal of Coastal Research, 1025-1029.

[6] Oñate, E., Celigueta, M. A., Idelsohn, S. R., Salazar, F., & Suárez, B. (2011). Possibilities of the particle finite element method for fluid–soil–structure interaction problems. Computational Mechanics, 48(3), 307-318.

[7] Oñate, E., Celigueta, M. A., Latorre, S., Casas, G., Rossi, R., & Rojek, J. (2014). Lagrangian analysis of multiscale particulate flows with the particle finite element method. Computational Particle Mechanics, 1(1), 85-102.

[8] Larese, A., Rossi, R., Oñate, E., & Idelsohn, S. R. (2008). Validation of the particle finite element method (PFEM) for simulation of free surface flows. Engineering Computations, 25(4), 385-425.

[9] Salazar, F., Oñate, E., & Morán, R. (2011). Numerical modeling of landslides in reservoirs using the Particle Finite Element Method (PFEM). Risk Analysis, Dam Safety, Dam Security and Critical Infrastructure Management, 245.

[10] Salazar, F., Irazábal, J., Larese, A., & Oñate, E. (2015). 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.

[11] Carbonell, J. M., Oñate, E., & Suárez, B. (2009). Modeling of ground excavation with the particle finite-element method. Journal of engineering mechanics, 136(4), 455-463.

[12] Donea, J., & Huerta, A. (2003). Finite element methods for flow problems. John Wiley & Sons.

[13] Oñate, E., Franci, A., & Carbonell, J. M. (2014). 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.

[14] Li, S., & Liu, W. K. (2007). Meshfree particle methods. Springer Science & Business Media.

[15] Edelsbrunner, H., & Mücke, E. P. (1994). Three-dimensional alpha shapes. ACM Transactions on Graphics (TOG), 13(1), 43-72.

[16] Idelsohn, S. R., Onate, E., Calvo, N., & Del Pin, F. (2003). The meshless finite element method. International Journal for Numerical Methods in Engineering, 58(6), 893-912.

[17] Oñate, E., Idelsohn, S. R., Del Pin, F., & Aubry, R. (2004). The particle finite element method—an overview. International Journal of Computational Methods, 1(02), 267-307.

[18] Fernández, M. A., Gerbeau, J. F., & Grandmont, C. (2007). A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. International Journal for Numerical Methods in Engineering, 69(4), 794-821.

[19] Chen, M. B. (2011, May). A parallel 3d delaunay triangulation method. In 2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications (pp. 52-56). IEEE.

[20] Foteinos, P., & Chrisochoides, N. (2011). Dynamic parallel 3D Delaunay triangulation. In Proceedings of the 20th International Meshing Roundtable (pp. 3-20). Springer Berlin Heidelberg.

[21] Tezduyar, T.E., Behr, M. and Liou, J. (1992a). A New Strategy for Finite Element Computations Involving Moving Boundaries and Interfaces - The Deforming- Spatial-Domain/Space-Time Procedure: I. The Concept and the Preliminary Numerical Tests. Computer Methods in Applied Mechanics and Engineering, 94, 339–351.

[22] Tezduyar, T.E., Behr, M., Mittal, S. and J. Liou (1992b). A New Strategy for Finite Element Computations Involving Moving Boundaries and Interfaces - The Deforming-Spatial-Domain/Space-Time Procedure: II. Computation of Freesurface Flows, Two-liquid Flows, and Flows with Drifting Cylinders. Computer Methods in Applied Mechanics and Engineering, 94, 353–371.

[23] Tezduyar, T. (2001). Finite Element Interface-Tracking and Interface-Capturing Techniques for Flows with Moving Boundaries and Interfaces ASME Paper IMECE2001/HTD-24206, Proceedings of the ASME Symposium on Fluid- Physics and Heat Transfer for Macro- and Micro-Scale Gas-Liquid and Phase- Change Flows, ASME, New York, New York, CD-ROM.

[24] Zienkiewicz, O.C. and Taylor, R.L. (2000). The finite element method. 5th Edition, 3 Volumes, Butterworth–Heinemann.

[25] Ribó, R., Pasenau, M. A. R., Escolano, E., Ronda, J. S. P., & González, L. F. (1998). GiD reference manual. CIMNE, Barcelona.

[26] Calvo, N. (2011). Generación de mallas tridimensionales por métodos duales. PhD Thesis. Universidad Nacional del Litoral, Argentina.

[27] Carbonell, J. M. (2009). Modeling of ground excavation with the particle finite element method. PhD Thesis. Universitat Politècnica de Catalunya

[28] Osher S, Fedkiw R. (2003) Level Set Methods and Dynamic Implicit Surfaces. Springer Science & Business Media.

[29] Li, S., Liu, W.K. (2004). Meshfree Particle Methods, Berlin: Springer Verlag

[30] Brackbill, J.U. and Ruppel, H.M. (1986). FLIP: A method for adaptively zoned, particle-in-cell calculations in two dimensions. Journal of Computational Physics, 65.

[31] Harlow, F.H. (1955). A Machine Calculation Method for Hydrodynamic Problems. Los Alamos Scientific Laboratory report LAMS-1956.

[32] Hirt, C.W.; Nichols, B.D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. Journal of Computational Physics.

[33] Ryzhakov, P., Oñate, E., Rossi, R., & Idelsohn, S. (2012). Improving mass conservation in simulation of incompressible flows. International Journal for Numerical Methods in Engineering, 90(12), 1435-1451.

[34] Calderer, Ramon. (2007). Aplicació del mètode dels elements finits i partícules (PFEM) al problema de transport de sediment.

[35] Rodriguez, F. (2013). Posibilidades del método de elementos finitos y partículas (PFEM) en problemas de erosión de suelos granulares.

[36] Shields, A. (1936). Application of similarity principles and turbulence research to bed-load movement. Soil Conservation Service.

[37] Cundall, P. A., & Strack, O. D. (1979). A discrete numerical model for granular assemblies. geotechnique, 29(1), 47-65.

[38] Swedish Industrial Association for Multiphase Flows (SIAMUF), & Sommerfeld, M. (2008). Best Practice Guidelines for Computational Fluid Dynamics of Dispersed Multi-Phase Flows. European Research Community on Flow, Turbulence and Combustion (ERCOFTAC).

[39] Shah, S. N., El Fadili, Y., & Chhabra, R. P. (2007). New model for single spherical particle settling velocity in power law (visco-inelastic) fluids. International journal of multiphase flow, 33(1), 51-66.

[40] Sifferman TR, Myers GM, Haden EL, Wahl HA (1974) Drillcutting transport in full-scale vertical annuli. J Pet Technol 26:1– 295

[41] Salazar, F., San-Mauro, J., Celigueta, M. Á., & Oñate, E. (2016). Air demand estimation in bottom outlets with the particle finite element method. Computational Particle Mechanics, 1-12.

[42] Peterka, T., Morozov, D., & Phillips, C. (2014, November). High-performance computation of distributed-memory parallel 3D Voronoi and Delaunay tessellation. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 997-1007). IEEE Press.

[43] Mohammadi, S. (2003). Discontinuum mechanics: Using finite and discrete elements (Vol. 1). Southampton, UK: WIT press. ISO 690

[44] Raúl Hospital Bravo. (2009). Posibilidades del método de elementos finitos y partículas (PFEM) en problemas de interacción fluido-estructura en flujos incompresibles. Bachelor thesis, Universitat Politècnica de Catalunya. Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports de Barcelona. Departament de Resistència de Materials i Estructures a l'Enginyeria, 2009

[45] Zohdi, T. I. (2007). An introduction to modeling and simulation of particulate flows. Society for Industrial and Applied Mathematics. ISO 690

[46] Jackson, R. (2000). The dynamics of fluidized particles. Cambridge University Press.

[47] Bourgoyne, A. T., Millheim, K. K., Chenevert, M. E., & Young, F. S. (1991). Applied drilling engineering (Vol. 2, pp. 137-144). Richardson, TX: Society of Petroleum Engineers.

[48] Ansley, R. W., & Smith, T. N. (1967). Motion of spherical particles in a Bingham plastic. AIChE Journal, 13(6), 1193-1196.

[49] Brookes, G. F., & Whitmore, R. L. (1969). Drag forces in Bingham plastics. Rheologica Acta, 8(4), 472-480.

[50] Kelessidis, V. C., & Mpandelis, G. (2004). Measurements and prediction of terminal velocity of solid spheres falling through stagnant pseudoplastic liquids. Powder Technology, 147(1), 117-125.

[51] Walker, R. E., & Mayes, T. M. (1975). Design of muds for carrying capacity. Journal of Petroleum Technology, 27(07), 893-900.

[52] Walker, R. E. (1976). Hydraulics limits are set by flow restrictions. Oil Gas J, 74, 86-90.

[53] Chien, S. F. (1994). Settling velocity of irregularly shaped particles. SPE Drilling & Completion, 9(04), 281-289.

[54] Clift, R., Grace, J. R., & Weber, M. E. (2005). Bubbles, drops, and particles. Courier Corporation.

[55] Haider, A., & Levenspiel, O. (1989). Drag coefficient and terminal velocity of spherical and nonspherical particles. Powder technology, 58(1), 63-70.

[56] Machac, I., Ulbrichova, I., Elson, T.P., Cheesman, D.J. (1995). Fall of spherical particles through non-Newtonian suspensions. Chemical Engineering Science, 50 (20). 3323-3327.

[57] Limache, A., Idelsohn, S., Rossi, R., & Oñate, E. (2007). The violation of objectivity in Laplace formulations of the Navier–Stokes equations. International journal for numerical methods in fluids, 54(6-8), 639-664.

[58] Breuer, M., & Rodi, W. (1994). Large-eddy simulation of turbulent flow through a straight square duct and a 180 bend. In Direct and large-eddy simulation I (pp. 273-285). Springer Netherlands.

[59] He, X., & Luo, L. S. (1997). Lattice Boltzmann model for the incompressible Navier–Stokes equation. Journal of statistical Physics, 88(3), 927-944.

[60] Monaghan, J. J. (1992). Smoothed particle hydrodynamics. Annual review of astronomy and astrophysics, 30(1), 543-574.

[61] Casas, G. (2018). Numerical analysis of particle-laden flows with the finite element method. PhD Thesis at Universitat Politècnica de Catalunya

[62] Lu, L.; Yoo, K.; Benyahia, S. (2016). Coarse-Grained-Particle Method for Simulation of Liquid–Solids Reacting Flows. Industrial & Engineering Chemistry Research. 55: 10477–10491.

[63] Jarauta, A., Ryzhakov, P., Pons-Prats, J., & Secanell, M. (2018). An implicit surface tension model for the analysis of droplet dynamics. Journal of Computational Physics, 374, 1196-1218.

[80] www.cimne.com/pfem

Annex: papers co-written by the author

This Annex contains a compilation of papers co-authored by the author, in which many developments explained here are introduced as well. They are sorted in chronological order of publication, independently from the order in which they are referenced in the text.

The list of references is the following ([1] to [7]):

1. Celigueta, M. A., Deshpande, K. M., Latorre, S., & Oñate, E. (2016). 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.

2. Oñate, E., Celigueta, M. A., & Idelsohn, S. R. (2006). Modeling bed erosion in free surface flows by the particle finite element method. Acta Geotechnica, 1(4), 237-252.

3. Oliveira, T. C. A., Gonzalez-Marco, D., Sánchez-Arcilla, A., Celigueta, M. A., & Oñate, E. (2007). Generation and propagation of water waves in a two-dimensional numerical flume. Computational Methods in Marine Engineering II, 241-244.

4. Onate, E., Idelsohn, S. R., Celigueta, M. A., & Rossi, R. (2008). Advances in the particle finite element method for the analysis of fluid–multibody interaction and bed erosion in free surface flows. Computer methods in applied mechanics and engineering, 197(19-20), 1777-1800.

5. Oliveira, T. C. A., Gironella, F. X., Sanchez-Arcilla, A., Sierra, J. P., & Celigueta, M. A. (2009). Nonlinear regular wave generation in numerical and physical flumes. Journal of Coastal Research, 1025-1029.

6. Oñate, E., Celigueta, M. A., Idelsohn, S. R., Salazar, F., & Suárez, B. (2011). Possibilities of the particle finite element method for fluid–soil–structure interaction problems. Computational Mechanics, 48(3), 307.

7. Oñate, E., Celigueta, M. A., Latorre, S., Casas, G., Rossi, R., & Rojek, J. (2014). Lagrangian analysis of multiscale particulate flows with the particle finite element method. Computational Particle Mechanics, 1(1), 85-102.

Paper 1

This paper, published in 2006, explains with detail the approach explained in Section 4. The *PFlow* code was used for all the computations.

MODELING BED EROSION IN FREE SURFACE FLOWS BY THE PARTICLE FINITE ELEMENT METHOD

Eugenio Oñate[1], **Miguel A. Celigueta**[2] and **Sergio R. Idelsohn**[3]
International Center for Numerical Methods in Engineering (CIMNE)
Universidad Politecnica de Cataluña, 08034 Barcelona, Spain
Email: onate@cimne.upc.edu
Web page: http://www.cimne.com/pfem

[1] Director of CIMNE; [2] Ph.D. Student; [3] ICREA Research Professor at CIMNE

Abstract. We present a general formulation for modeling bed erosion in free surface flows using the particle finite element method (PFEM). The key feature of the PFEM is the use of an updated Lagrangian description to model the motion of nodes (particles) in domains containing fluid and solid subdomains. Nodes are viewed as material points (called particles) which can freely move and even separate from the fluid and solid subdomains representing, for instance, the effect of water drops or soil/rock particles. A mesh connects the nodes defining the discretized domain in the fluid and solid regions where the governing equations, expressed in an integral form, are solved as in the standard FEM. The necessary stabilization for dealing with the incompressibility of the fluid is introduced via the finite calculus (FIC) method. An incremental iterative scheme for the solution of the non linear transient coupled fluid-structure problem is described. The erosion mechanism is modeled by releasing the material adjacent to the bed surfarce accordingly to the frictional work generated by the fluid shear stresses. The released bed material is subsequently transported by the fluid flow. Examples of application of the PFEM to solve a number of bed erosion problems involving large motions of the free surface and splashing of waves are presented.

Key words: bed erosion, free surface flows, particle finite element method

1 INTRODUCTION

The calculation of bed erosion and sediment transport in open channel flows are extremely important tasks in many areas of river and environmental engineering and related areas. For example, bed erosion can lead to instabilities of the river basin slopes. It can also undermine the foundation of bridge piles thereby favouring structural failure. Modeling of bed erosion is also relevant for predicting the evolution of surface material dragged in earth dams in overspill situations. Bed erosion is one of

the main causes of environmental damage in floods.

Prediction of erosion of soil/rock particles in bed surfaces due to water streams is very difficult due to the complexity of accurately predicting the tangential stresses at the fluid-bed interface which are mainly responsible for the detachment of bed particles. The definition of the erosion onset conditions for different bed geomaterials is also an important and difficult task. The modeling of sediment transport phenomena is also very complex.

The erosion and transport of sediment particles in environmental flows can be analyzed by solving the Navier-Stokes equations for the water flow, either in the fully 3D version, or via a simpler 2D depth average model in combination with an assumed vertical velocity profile (typically of logarithmic type). The flow field variables computed at each time step, or at every iteration within a time step in a strongly coupled scheme, are used as input data for solving the sediment transport equations and other relevant information such as the suspended load concentration, the bed load transport rate and the bed deformation. Numerical solutions for these type of problems have been reported using mainly finite difference and finite volume schemes in Eulerian and arbitrary Lagrangian-Eulerian (ALE) grids for solving both the fluid flow and the sediment transport equations [van Rijn (1984,85); Struiksma *et al.* (1985); Phillips and Sutherland (1989); Rahuel *et al.* (1989); Kovacs and Parker (1994); Darby and Thorne (1996); Wu *et al.* (1997); Fell *et al.* (2003); Wan and Fell (2004); Parker *et al.* (2005)].

In addition to the intrinsic complexities of the multiphysics flow-erosion-transport problem, the numerical solution of the equations for the fluid-structure interaction (FSI) problem in free surface flows is faced with the treatment of the convective terms and the incompressibility constraint in the fluid equations, the modeling of the free surface accounting for wave splashing, the transfer of information between the fluid and solid domains via the contact interfaces and the tracking of solid elements within the fluid domain. Indeed most of these problems are extremely difficult to model using the Eulerian and ALE formulations.

An alternative approach which simplifies many of above difficulties is to use a *Lagrangian description* to formulate the governing equations of both the solid and the fluid domains. In the Lagrangian formulation the motion of the individual particles are followed and, consequently, nodes in a finite element mesh can be viewed as moving material points (hereforth called "particles"). Hence, the motion of the mesh discretizing the total domain (including both the fluid and solid regions) is followed during the transient solution.

In this paper we present a particular class of Lagrangian formulation developed by the authors to solve bed erosion problems in free surface flows. The method is an extension of the so-called *particle finite element method* (PFEM). The PFEM treats the mesh nodes in the fluid and solid domains as *particles* which can freely move and even separate from each domain representing, for instance, the effect of water drops or solid particles. A finite element mesh connects the nodes defining the discretized domain where the governing equations are solved in the standard FEM fashion. The particular application of the PFEM to model bed erosion problems here described is the natural evolution of recent work of the authors for the solution

of FSI problems using Lagrangian finite element and meshless methods [Oñate *et al.* (2003; 2004a,b); Idelsohn *et al.* (2003a; 2003b; 2004); Aubry *et al.* (2005)].

An obvious advantage of the Lagrangian formulation is that the convective terms disappear from the fluid equations. The difficulty is however transferred to the problem of adequately (and efficiently) moving the mesh nodes. Indeed in the PFEM approach remeshing is a frequent necessity along the time solution. We use an innovative mesh regeneration procedure blending elements of different shapes using an extended Delaunay tesselation with adequate C° continuous shape functions [Idelsohn *et al.* (2003a; 2003c)].

The need to properly treat the incompressibility condition in the fluid still remains in the Lagrangian formulation. The use of standard finite element interpolations may lead to a volumetric locking defect unless some precautions are taken [Donea and Huerta (2003), Zienkiewicz *et al.* (2006)]. In our work volumetric locking is avoided via a finite calculus (FIC) procedure [Oñate (2004)].

The layout of the paper is the following. In the next section the basic ideas of the PFEM are outlined. Next the basic equation for an incompressible flow using a Lagrangian description and the FIC formulation are presented. Then a fractional step scheme for the transient solution via standard finite element procedures is described. Details of the treatment of the coupled FSI problem are given. The procedures for mesh generation, for identification of the free surface nodes, for treating frictional contact situations and for modeling bed erosion are described. Finally, the efficiency of the PFEM is shown in its application to a number of bed erosion problems involving surface waves.

2 THE BASIS OF THE PARTICLE FINITE ELEMENT METHOD

Let us consider a continuum domain containing both fluid and solid subdomains. The fluid particles interact with the solid boundaries thereby inducing the deformation of the solid which in turn affects the flow motion and, therefore, the problem is fully coupled.

In the PFEM approach, both the fluid and the solid domains are modeled using an *updated Lagrangian formulation*. That is, all variables in the fluid and solid domains 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 $t + \Delta t$ (Figure 1). The finite element method (FEM) is used to solve the continuum equations in both domains. Hence a mesh discretizing these domains is generated in order to solve the governing equations for both the fluid and solid problems in the standard FEM fashion. We note again that the nodes discretizing the fluid and solid domains are viewed as material particles whose motion is tracked during the transient solution. This is useful to model the separation of fluid particles from the main fluid domain, *or solid particles from the bed surface*, and to follow their subsequent motion as individual particles with a known density, an initial acceleration and velocity and subject to gravity forces.

It is important to recall that each particle is treated as a material point characterized by the density of the solid or fluid domain to which it belongs. The mass of a given domain is obtained by integrating the density at the different material

points over the domain.

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 gradients of the fluid or the structure variables occur.

2.1 Basic steps of the PFEM

For clarity purposes we will define the *collection or cloud of nodes (C)* pertaining to either the fluid and the solid for bed erosion problems subdomains, the *volume (V)* defining the analysis domain for the fluid and the solid, and the *mesh (M)* discretizing both domains.

A typical solution with the PFEM involves the following steps.

1. The starting point at each time step is the cloud of points in the fluid and solid subdomains. For instance ${}^nC$ denotes the cloud at time $t = t_n$ (Figure 2).

2. Identify the boundaries for both the fluid and solid domains defining the analysis domain ${}^nV$ in the fluid and the solid. This is an essential step as some boundaries, such as the free surface in fluids or the bed surface, may be severely distorted during the solution process including separation and re-entering of nodes. The Alpha Shape method [Edelsbrunner and Mucke (1999)] is used for the boundary definition (Section 7).

3. Discretize the fluid and solid subdomains with a finite element mesh ${}^nM$. In our work we use an innovative mesh generation scheme based on the extended Delaunay tesselation (Section 6) [Idelsohn *et al.* (2003a; 2003b; 2004)].

4. Solve the coupled Lagrangian equations of motion for the fluid and the solid domains. Compute the relevant state variables in both domains at the next (updated) configuration for $t + \Delta t$: velocities, pressure and viscous stresses in the fluid and displacements, stresses and strains in the solid. An overview of the coupled FSI algorithm is given in the next section.

5. Compute the frictional work (W_f) performed by the tangential stresses at the bed surface. Bed erosion initiates if W_f exceeds a critical value W_c. Bed surface points where $W_f > W_c$ are released from the bed domain and are subsequently transported by the fluid velocity.

6. Move the mesh nodes to a new position ${}^{n+1}C$ where $n + 1$ denotes the time $t_n + \Delta t$, in terms of the time increment size. This step is typically a consequence of the solution process of step 4. Recall that a node identifies a material point in either the fluid or solid subdomains.

7. Go back to step 1 and repeat the solution process for the next time step.

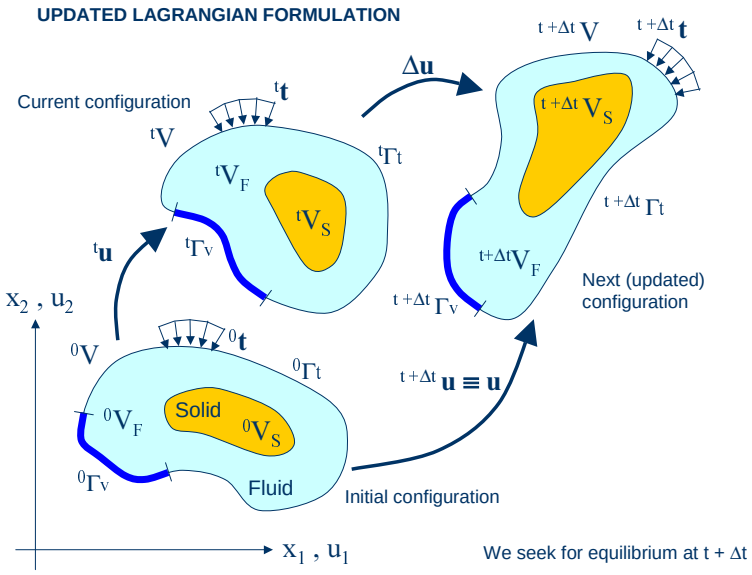


Figure 1: Updated Lagrangian description for a continuum containing a fluid and a solid domain

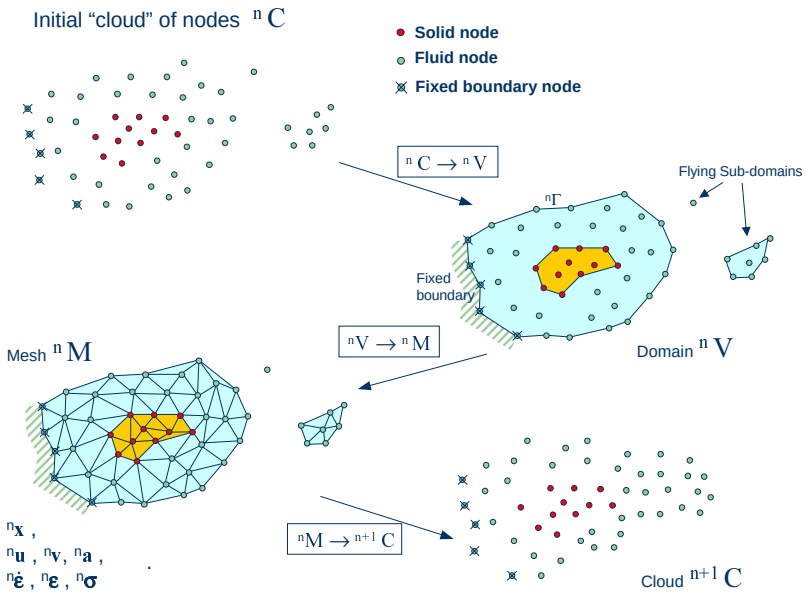


Figure 2: Sequence of steps to update a "cloud" of nodes from time n ($t = t_n$) to time $n+1$ ($t = t_n + \Delta t$)

2.2 Overview of the coupled FSI algoritm

Figure 3 shows a typical domain V with external boundaries Γ_V and Γ_t where the velocity and the surface tractions are prescribed, respectively. The domain V

is formed by fluid (V_F) and solid (V_S) subdomains. Both subdomains interact at a common boundary Γ_{FS} where the surface tractions and the kinematic variables (displacements, velocities and acelerations) are the same for both subdomains. Note that both sets of variables (the surface tractions and the kinematic variables) are equivalent in the equilibrium configuration.

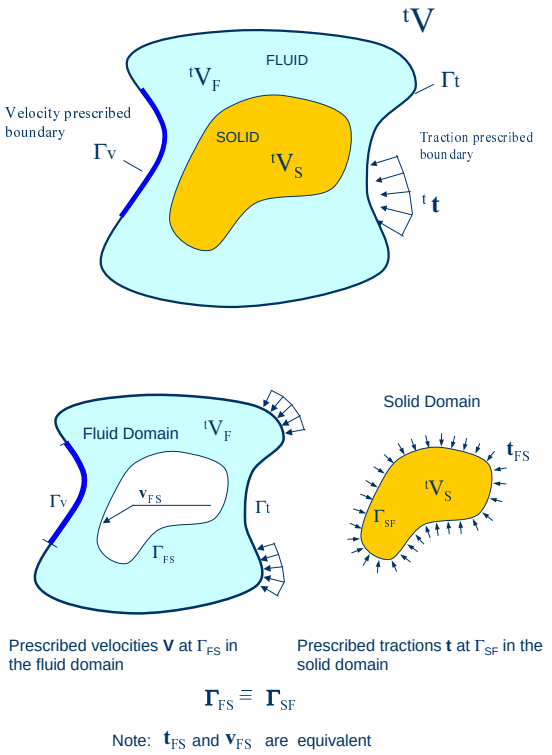


Figure 3: Split of the analysis domain V into fluid and solid subdomains. Equivalence of surface tractions and kinematic variables at the common interface

Note that the flow in an open channel is a particular case of above situations where the solid domain constitutes the bed region whose surface is eroded by the interaction with the fluid particles in motion.

Let us define ${}^tS$ and ${}^tF$ as the set of variables defining the kinematics and the stress-strain fields in the solid and fluid domains at time t, respectively, i.e.

$$ {}^tS := [{}^t\mathbf{x}_s, {}^t\mathbf{u}_s, {}^t\mathbf{v}_s, {}^t\mathbf{a}_s, {}^t\boldsymbol{\varepsilon}_s, {}^t\boldsymbol{\sigma}_s, \cdots]^T \quad (1) $$

$$ {}^tF := [{}^t\mathbf{x}_F, {}^t\mathbf{u}_F, {}^t\mathbf{v}_F, {}^t\mathbf{a}_F, {}^t\dot{\boldsymbol{\varepsilon}}_F, {}^t\boldsymbol{\sigma}_F, \cdots]^T \quad (2) $$

where $\mathbf{x}$ is the nodal coordinate vector, $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{a}$ are the vector of displacements, velocities and accelerations, respectively, $\boldsymbol{\varepsilon}, \dot{\boldsymbol{\varepsilon}}$ and $\boldsymbol{\sigma}$ are the strain vector, the strain-rate (or rate of deformation) vectors and the Cauchy stress vector, respectively and

F and S denote the variables in the fluid and solid domains, respectively. In the discretized problem, a bar over these variables will denote nodal values.

The coupled FSI problem of Figure 3 is solved using the following conceptual scheme:

0. We assume that the variables in the solid and fluid domains at time t (^tS and ^tF) are known.

1. Solve for the variables at the solid domain at time $t + \Delta t$ ($^{t+\Delta t}S$) under prescribed surface tractions at the fluid-solid boundary Γ_{FS}.

2. Solve for the variables at the fluid domain at time $t + \Delta t$ ($^{t+\Delta t}F$) under prescribed surface tractions at the external boundary Γ_t and prescribed velocities at the external and internal boundaries Γ_V and Γ_{FS}, respectively.

   Iterate between 1 and 2 until convergence.

The variables at the solid domain $^{t+\Delta t}S$ are found via the integration of the dynamic equations of motion in the solid region written as

$$\mathbf{M}_s\mathbf{a}_s + \mathbf{g}_s - \mathbf{f}_s = \mathbf{0} \tag{3}$$

where $\mathbf{M}_s, \mathbf{g}_s$ and $\mathbf{f}_s$ denote the mass matrix, the internal node force vector and the external nodal force vector in the solid domain. The time integration of Eq.(3) is performed using a standard Newmark method. An incremental iterative scheme is implemented within each time step to account for nonlinear geometrical and material effects [Zienkiewicz and Taylor (2005)].

The FEM solution of the variables in the (incompressible) fluid domain implies solving the momentum and incompressibility equations. In our work we use a stabilized FEM based on the Finite Calculus approach which allows to use a linear approximation for the velocity and pressure variables [Oñate (1998,2004)]. Details of the FEM/FIC formulation used are given in the next section.

Figure 4 shows a typical example of a PFEM solution in 2D. The pictures correspond to the analysis of the problem of breakage of a water column [Oñate *et al.* (2004); Idelsohn *et al.* (2004)]. Figure 4a shows the initial grid of four node rectangles discretizing the fluid domain and the solid walls. Boundary nodes identified with the Alpha-Shape method have been marked with a circle. Figures 4b and 4c show the mesh for the solution at two later times.

3 FIC/FEM FORMULATION FOR A LAGRANGIAN INCOMPRESSIBLE FLUID

The standard infinitesimal equations for a viscous incompressible fluid can be written in a Lagrangian frame as [Oñate (1998); Zienkiewicz *et al.* (2006)].

Momentum

$$r_{m_i} = 0 \quad \text{in } V_F \tag{4}$$

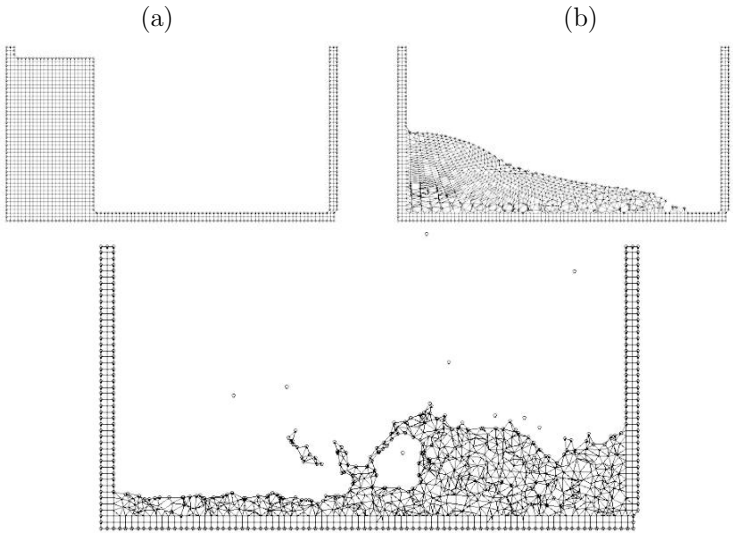


Figure 4: Breakage of a water column. (a) Discretization of the fluid domain and the solid walls. Boundary nodes are marked with circles. (b) and (c) Mesh in the fluid and solid domains at two different times.

Mass balance

$$r_d = 0 \quad \text{in } V_F \tag{5}$$

where

$$r_{m_i} = \rho \frac{\partial v_i}{\partial t} - \frac{\partial \sigma_{ij}}{\partial x_j} - b_i \quad , \quad \sigma_{ji} = \sigma_{ij} \tag{6}$$

$$r_d = \frac{\partial v_i}{\partial x_i} \qquad i,j = 1, n_d \tag{7}$$

Above n_d is the number of space dimensions, v_i is the velocity along the ith global axis ($v_i = \partial u_i / \partial t$, where u_i is the ith displacement), ρ is the (constant) density of the fluid, b_i are the body forces, σ_{ij} are the total stresses given by $\sigma_{ij} = s_{ij} - \delta_{ij} p$, p is the absolute pressure (defined positive in compression) and s_{ij} are the viscous deviatoric stresses related to the viscosity μ by the standard expression

$$s_{ij} = 2\mu \left(\dot{\varepsilon}_{ij} - \delta_{ij} \frac{1}{3} \frac{\partial v_k}{\partial x_k} \right) \tag{8}$$

where δ_{ij} is the Kronecker delta and the strain rates $\dot{\varepsilon}_{ij}$ are

$$\dot{\varepsilon}_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \tag{9}$$

In the above all variables are defined at the current time t (current configuration). The standard summation convention for repeated indexes is assumed unless otherwise specified.

In our work we will solve a *modified set of governing* equations derived using a finite calculus (FIC) formulation. The FIC governing equations are [Oñate (1998; 2000; 2004); Oñate and García (2001)].

Momentum

$$r_{m_i} - \underline{\frac{1}{2} h_j \frac{\partial r_{m_i}}{\partial x_j}} = 0 \qquad \text{in } V_F \tag{10}$$

Mass balance

$$r_d - \underline{\frac{1}{2} h_j \frac{\partial r_d}{\partial x_j}} = 0 \qquad \text{in } V_F \tag{11}$$

The problem definition is completed with the following boundary conditions

$$n_j \sigma_{ij} - t_i + \underline{\frac{1}{2} h_j n_j r_{m_i}} = 0 \quad \text{on } \Gamma_t \tag{12}$$

$$v_j - v_j^p = 0 \quad \text{on } \Gamma_v \tag{13}$$

and the initial condition is $v_j = v_j^0$ for $t = t_0$.

In Eqs.(12) and (13), t_i and v_j^p are surface tractions and prescribed velocities on the boundaries Γ_t and Γ_v, respectively, n_j are the components of the unit normal vector to the boundary.

The $h_i's$ in above equations are *characteristic lengths* of the domain where balance of momentum and mass is enforced. In Eq.(12) these lengths define the domain where equilibrium of boundary tractions is established. In our work we have taken h_i to be constant at each element and equal to a typical element dimension h^e computed as $h^e = [V^e]^m$ where V^e is the element volume and $m = 1/2$ for 2D problems and $m = 1/3$ for 3D problems. Details of the derivation of Eqs.(10)–(13) can be found in Oñate (1998,2000,2004).

Eqs.(10)–(13) are the starting points for deriving stabilized finite element methods to solve the incompressible Navier-Stokes equations in a Lagrangian frame of reference using equal order interpolation for the velocity and pressure variables [Idelsohn *et al.* (2002; 2003a; 2003b; 2004); Oñate *et al.* (2003); Aubry *et al.* (2005)]. Application of the FIC formulation to finite element and meshless analysis of fluid flow problems can be found in [García and Oñate (2003); Oñate (2000; 2004); Oñate *et al.* (2000; 2004a); Oñate and García (2001); Oñate and Idelsohn (1988)].

3.1 Transformation of the mass balance equation. Integral governing equations

The underlined term in Eq.(11) can be expressed in terms of the momentum equations. The new expression for the mass balance equation is [Oñate (2000); Oñate *et al.* (2004b)]

$$r_d - \sum_{i=1}^{n_d} \tau_i \frac{\partial r_{m_i}}{\partial x_i} = 0 \qquad \text{with} \qquad \tau_i = \frac{3h_i^2}{8\mu} \tag{14}$$

At this stage it is no longer necessary to retain the stabilization terms in the momentum equations. These terms are critical in Eulerian formulations to stabilize the numerical solution for high values of the convective terms. In the Lagrangian formulation the convective terms dissappear from the momentum equations and the FIC terms in these equations are just useful to derive the form of the mass balance equation given by Eq.(14) and can be disregarded there onwards. Consistently, the stabilization terms are also neglected in the Neumann boundary conditions (Eq.(12)).

The weighted residual expression of the final form of the momentum and mass balance equations can be written as

$$\int_{V_F} \delta v_i r_{m_i} dV + \int_{\Gamma_t} \delta v_i (n_j \sigma_{ij} - t_i) d\Gamma = 0 \tag{15}$$

$$\int_{V_F} q \left[r_d - \sum_{i=1}^{n_d} \tau_i \frac{\partial r_{m_i}}{\partial x_i} \right] dV = 0 \tag{16}$$

where δv_i and q are arbitrary weighting functions equivalent to virtual velocity and virtual pressure fields.

The computation of the residual terms in Eq.(16) is simplified if we introduce the pressure gradient projections π_i, defined as

$$\pi_i = r_{m_i} - \frac{\partial p}{\partial x_i} \tag{17}$$

We express now r_{m_i} in Eq.(16) in terms of the π_i which then become additional variables. The system of integral equations is therefore augmented in the necessary number of equations by imposing that the residual r_{m_i} vanishes within the analysis domain (in an average sense). We proceed next to integrate by parts the r_{m_i} term in Eq.(16) and the deviatoric stresses and the pressure terms within r_{m_i} in Eq.(15). The final system of governing equation is

$$\int_{V_F} \left[\delta v_i \rho \frac{\partial v_i}{\partial t} + \delta \dot{\varepsilon}_{ij} (s_{ij} - \delta_{ij} p) \right] dV - \int_{V_F} \delta v_i b_i dV - \int_{\Gamma_t} \delta v_i t_i d\Gamma = 0 \tag{18}$$

$$\int_{V_F} q \frac{\partial v_i}{\partial x_i} dV + \int_{V_F} \sum_{i=1}^{n_d} \tau_i \frac{\partial q}{\partial x_i} \left(\frac{\partial p}{\partial x_i} + \pi_i \right) dV = 0 \tag{19}$$

$$\int_{V_F} \delta \pi_i \tau_i \left(\frac{\partial p}{\partial x_i} + \pi_i \right) dV = 0 \qquad \text{no sum in } i \tag{20}$$

with $i, j, k = 1, n_d$. In Eq.(18) $\delta \dot{\varepsilon}_{ij}$ are virtual strain rates. In Eqs.(20) $\delta \pi_i$ are appropriate weighting functions and the τ_i weights are introduced for symmetry reasons. Note that the boundary term resulting from the integration by parts of r_{m_i} in Eq.(16) has been neglected as the influence of this term in the numerical solution has been found to be negligible.

3.2 Finite element discretization

We choose equal order C° continuous interpolations of the velocities, the pressure and the pressure gradient projections π_i over each element with n nodes. The interpolations are written as

$$v_i = \sum_{j=1}^{n} N_j \bar{v}_i^j \quad , \quad p = \sum_{j=1}^{n} N_j \bar{p}^j \quad , \quad \pi_i = \sum_{j=1}^{n} N_j \bar{\pi}_i^j \tag{21}$$

where $\bar{(\cdot)}^j$ denotes nodal variables and N_j are the shape functions [Zienkiewicz *et al.* (2006)].

Substituting the approximations (21) into Eqs.(19–20) and choosing a Galerkin form with $\delta v_i = q = \delta \pi_i = N_i$ leads to the following system of discretized equations

$$\mathbf{M}\dot{\bar{\mathbf{v}}} + \mathbf{K}\bar{\mathbf{v}} - \mathbf{G}\bar{\mathbf{p}} - \mathbf{f} = \mathbf{0} \tag{22a}$$

$$\mathbf{G}^T\bar{\mathbf{v}} + \mathbf{L}\bar{\mathbf{p}} + \mathbf{Q}\bar{\boldsymbol{\pi}} = \mathbf{0} \tag{22b}$$

$$\mathbf{Q}^T\bar{\mathbf{p}} + \hat{\mathbf{M}}\bar{\boldsymbol{\pi}} = \mathbf{0} \tag{22c}$$

The matrices and vectors in Eqs.(22) are assembled from the element contributions given by (for 2D problems)

$$\mathbf{M}_{ij} = \int_{V_F^e} \rho \mathbf{N}_i \mathbf{N}_j dV \quad , \quad \mathbf{K}_{ij} = \int_{V_F^e} \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j dV$$

$$\mathbf{D} = \mu \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad \mathbf{B}_i = \begin{bmatrix} \dfrac{\partial N_i}{\partial x_1} & 0 \\ 0 & \dfrac{\partial N_i}{\partial x_2} \\ \dfrac{\partial N_i}{\partial x_2} & \dfrac{\partial N_i}{\partial x_1} \end{bmatrix}$$

$$L_{ij} = \int_{V_F^e} \tau_k \frac{\partial N_i}{\partial x_k} \frac{\partial N_j}{\partial x_k} dV \quad , \quad \mathbf{Q} = [\mathbf{Q}^1, \mathbf{Q}^2] \quad , \quad Q_{ij}^k = \int_{V_F^e} \tau_k \frac{\partial N_i}{\partial x_k} N_j dV$$

$$\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{M}}^1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{M}}^2 \end{bmatrix} \quad , \quad \hat{\mathbf{M}}_{ij}^k = \int_{V_F^e} \tau_k N_i N_j dV \quad , \quad \mathbf{G}_{ij} = \int_{V_F^e} \mathbf{B}_i^T \mathbf{m} N_j dV$$

$$\mathbf{f}_i = \int_{V_F^e} N_i \mathbf{b} dV + \int_{\Gamma_t^e} N_i \mathbf{t} d\Gamma \quad , \quad \mathbf{b} = [b_1, b_2]^T \quad , \quad \mathbf{t} = [t_1, t_2]^T \tag{23}$$

with $i, j = 1, n$ and $k, l = 1, 2$.

In above $\mathbf{B}$ is the strain rate matrix [Zienkiewicz *et al.* (2006)], V_F^e and Γ_t^e are the volume and the Neumann boundary of the element and $\mathbf{m} = [1, 1, 0]^T$ for 2D problems.

3.3 Fractional step algorithm for the fluid variables

The starting point of the iterative algorithm are the variables at time n in the fluid domain (${}^nF$). The sought variables are the variables at time $n+1$ (${}^{n+1}F$). For the sake of clarity we will skip the upper left index $n+1$ for all variables, i.e.

$$ {}^{n+1}\bar{\mathbf{x}} \equiv \bar{\mathbf{x}} \; ; \; {}^{n+1}\bar{\mathbf{p}} \equiv \bar{\mathbf{p}} \; ; \; {}^{n+1}\bar{\boldsymbol{\pi}} \equiv \bar{\boldsymbol{\pi}} \; ; \; {}^{n+1}\bar{\mathbf{x}} \equiv \bar{\mathbf{x}} \; ; \text{etc.} \tag{24} $$

A simple iterative algorithm is obtained by splitting the pressure from the momentum equations as follows

$$ \bar{\mathbf{v}}^* = {}^n\bar{\mathbf{v}} - \Delta t \mathbf{M}^{-1}[\mathbf{K}\bar{\mathbf{v}}^j - \mathbf{G}^n\mathbf{p} - \mathbf{f}] \tag{25} $$

$$ \bar{\mathbf{v}}^{j+1} = \bar{\mathbf{v}}^* + \Delta t \mathbf{M}^{-1}\mathbf{G}\delta\bar{\mathbf{p}} \tag{26} $$

where $\delta\bar{\mathbf{p}}$ denotes a pressure increment. In above equations and in the following the left upper index n refers to values in the current configuration ${}^nV_F$ whereas the right index j denotes the iteration number within each time step.

The value of $\bar{\mathbf{v}}^{j+1}$ from Eq.(26) is substituted now into Eq.(22b) to give

$$ \mathbf{G}^T\bar{\mathbf{v}}^* + \Delta t \mathbf{S}\delta\bar{\mathbf{p}} + \mathbf{L}\bar{\mathbf{p}}^{j+1} + \mathbf{Q}\bar{\boldsymbol{\pi}}^j = \mathbf{0} \tag{27a} $$

where

$$ \mathbf{S} = \mathbf{G}^T\mathbf{M}^{-1}\mathbf{G} \tag{27b} $$

Typically matrix $\mathbf{S}$ is computed using a diagonal matrix $\mathbf{M} = \mathbf{M}_d$, where the subscript d denotes hereonward a diagonal matrix. Diagonalization can be performed by a lumping technique.

An alternative is to approximate matrix $\mathbf{S}$ by a Laplacian matrix. This reduces considerably the bandwith of $\mathbf{S}$. The disadvantage is that the pressure increment must be prescribed on the free surface and this reduces the accuracy in the satisfaction of the incompressibility condition in these regions.

A semi-implicit algorithm can be derived as follows. For each iteration:

Step 1 Compute $\mathbf{v}^*$ from Eq.(25) with $\mathbf{M} = \mathbf{M}_d$. For the first iteration $(\bar{\mathbf{v}}^1, \bar{\mathbf{p}}^1, \bar{\boldsymbol{\pi}}^1, \bar{\mathbf{x}}^1) \equiv ({}^n\bar{\mathbf{v}}, {}^n\bar{\mathbf{p}}, {}^n\bar{\boldsymbol{\pi}}, {}^n\bar{\mathbf{x}})$

Step 2 Compute $\delta\bar{\mathbf{p}}$ and $\bar{\mathbf{p}}^{j+1}$ from Eq.(27a) as

$$ \delta\bar{\mathbf{p}} = -(\mathbf{L} + \Delta t \mathbf{S})^{-1}[\mathbf{G}^T\bar{\mathbf{v}}^* + \mathbf{Q}\bar{\boldsymbol{\pi}}^j + \mathbf{L}\bar{\mathbf{p}}^j] \tag{28a} $$

The pressure $\bar{\mathbf{p}}^{j+1}$ is computed as follows

$$ \bar{\mathbf{p}}^{j+1} = \bar{\mathbf{p}}^j + \delta\bar{\mathbf{p}}^j \tag{28b} $$

Step 3 Compute $\bar{\mathbf{v}}^{j+1}$ from Eq.(26) with $\mathbf{M} = \mathbf{M}_d$
Step 4 Compute $\bar{\boldsymbol{\pi}}^{j+1}$ from Eq.(22c) as

$$ \bar{\boldsymbol{\pi}}^{j+1} = -\hat{\mathbf{M}}_d^{-1}\mathbf{Q}^T\bar{\mathbf{p}}^{j+1} \tag{29} $$

Step 5 Update the coordinates of the mesh nodes. From the definition of the velocity $v_i = \partial u_i / \partial t$ it is deduced.

$$\mathbf{x}_i^{j+1} = {}^n\mathbf{x}_i + \bar{\mathbf{v}}_i^{j+1} \Delta t \tag{30}$$

Step 6 Check the convergence of the velocity and pressure fields. If convergence is achieved move to the next time step, otherwise return to step 1 for the next iteration with $j \leftarrow j + 1$.

Note that solution of steps 1, 3 and 4 does not require the solution of a system of equations as a diagonal form is chosen for $\mathbf{M}$ and $\hat{\mathbf{M}}$.

In the examples presented in the paper the time increment size has been chosen as

$$\Delta t = \min(\Delta t_i) \quad \text{with} \quad \Delta t_i = \frac{h_i^{\min}}{|\mathbf{v}|} \tag{31}$$

where $h_i^{\min}$ is the distance between node i and the closest node in the mesh.

Although not explicitely mentioned all matrices and vectors in Eqs.(27)–(31) are computed at the *updated configuration* ${}^{n+1}V_F$. This means that the integration domain changes for each iteration and, hence, all the terms involving space derivatives must be updated at each iteration. An alternative is to refer the integrations domain at each time step to the current configuration ${}^nV_F$. The jacobian matrix is needed in this case to transform the space derivatives and the differencial of volume from ${}^{n+1}V_F$ to ${}^nV_F$ at each iteration.

The boundary conditions are applied as follows. No condition is applied for the computation of the fractional velocities $\bar{\mathbf{v}}^*$ in Eq.(25). The prescribed velocities at the boundary are applied when solving for $\bar{\mathbf{v}}^{j+1}$ in step 3.

4 STAGGERED SCHEME FOR THE FSI PROBLEM

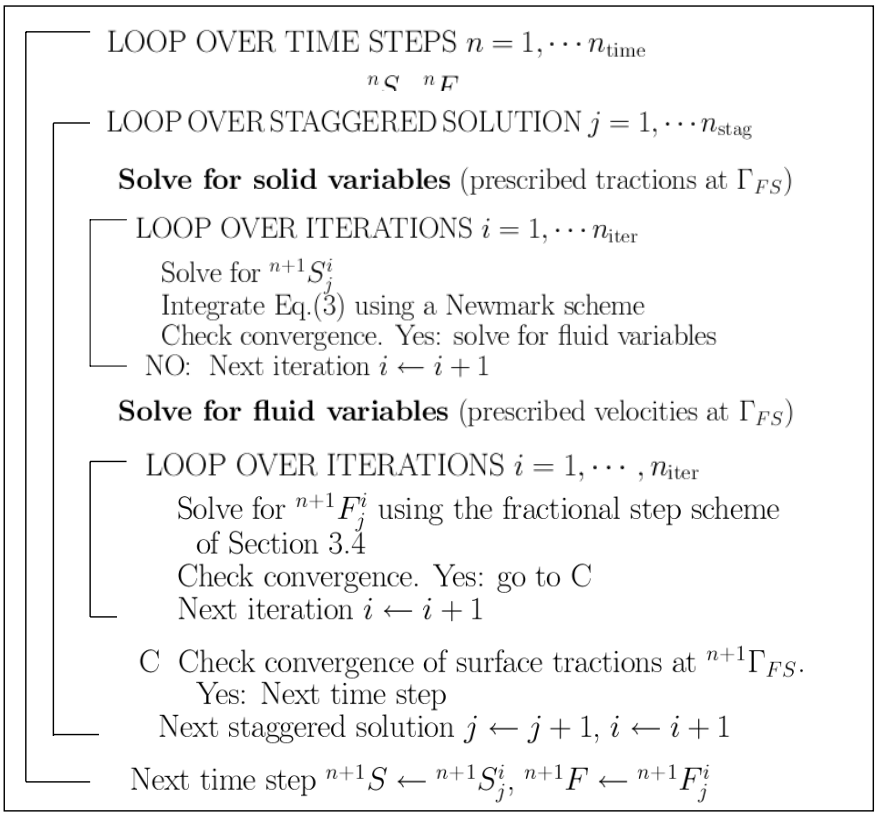
The solution for the variables in the solid and fluid domains at the updated configuration ${}^{n+1}F, {}^{n+1}S$ is found using the staggered scheme shown in Box 1.

Indeed a "weak" version of the staggered scheme can be implemented simply by eliminating the loop over the staggered solution in Box 1. The strong staggered scheme shown in Box 1 is recommended for problems with a large number of solid bodies interacting with the fluid particles. For the bed erosion problems presented in this paper we have used the weak staggered scheme.

5 TREATMENT OF CONTACT BETWEEN THE FLUID AND A FIXED BOUNDARY

The motion of the solid is governed by the action of the fluid flow forces induced by the pressure and the viscous stresses acting at the fixed boundary, as mentioned above.

The condition of prescribed velocities at the fixed boundaries in the PFEM are 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 penetration of the fluid nodes into*



Box 1. Staggered scheme for the FSI problem

the solid boundaries (Figure 5). This simple way to treat the fluid-wall contact is another distinct and attractive feature of the PFEM formulation.

6 GENERATION OF A NEW MESH

One of the key points for the success of the PFEM formulation is the fast regeneration of a mesh at every time step on the basis of the position of the nodes in the space domain. In our work the mesh is *regenerated at each time step* using the so called extended Delaunay tesselation (EDT) [Idelsohn *et al.* (2003a; 2003c; 2004)]. The EDT generates non-standard meshes combining elements of arbitrary polyhedrical shapes (triangles, quadrilaterals and other polygons in 2D and tetrahedra, hexahedra and arbitrary polyhedra in 3D) in a computing time of order n, where n is the total number of nodes in the mesh (Figure 6). The C° continuous shape functions of each element are obtained using the so-called meshless finite element interpolation (MFEM). Details of the mesh generation procedure and the derivation of the MFEM shape functions can be found in [Idelsohn *et al.* (2003a; 2003c; 2004)].

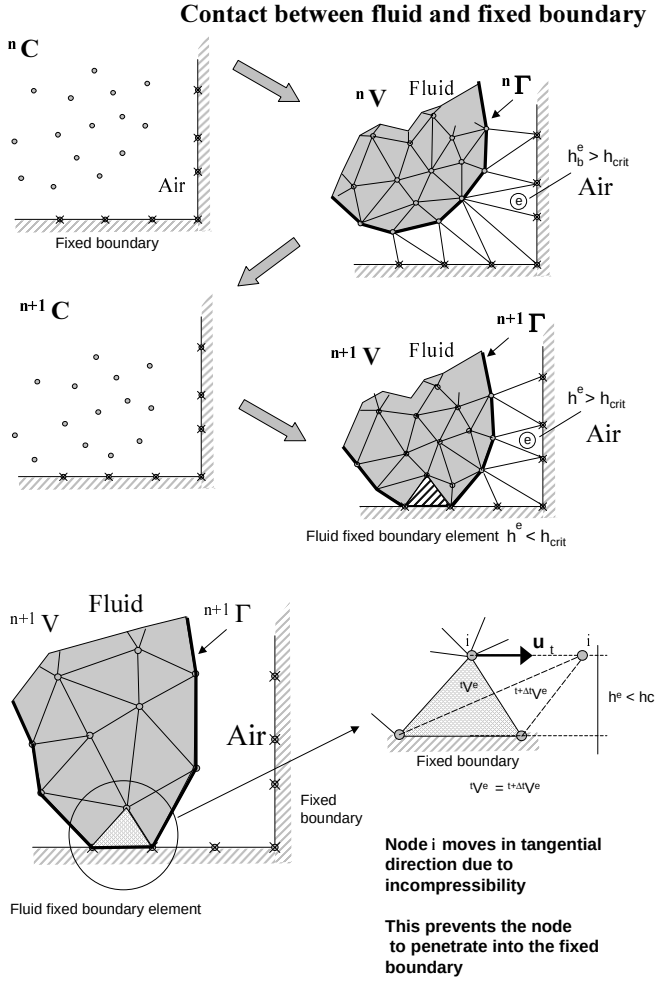


Figure 5: Automatic treatment of contact condition at the fluid-wall interface

7 IDENTIFICATION OF BOUNDARY SURFACES

One of the main tasks in the PFEM is the correct definition of the boundary domain. Sometimes, boundary nodes are explicitly identified differently from internal nodes. In other cases, the total set of nodes is the only information available and the algorithm must recognize the boundary nodes.

The extended Delaunay partition makes it easier to recognize boundary nodes. Considering that the nodes follow a variable $h(x)$ distribution, where $h(x)$ is typically the minimum distance between two nodes, the following criterion has been used. *All*

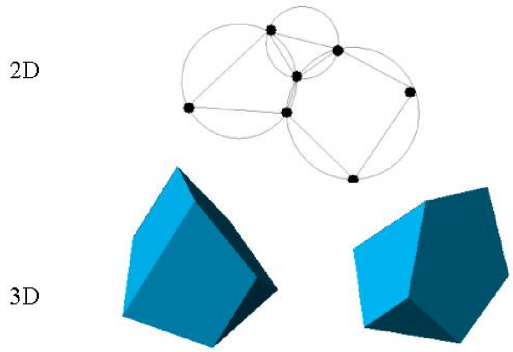


Figure 6: Generation of non standard meshes combining different polygons (in 2D) and polyhedra (in 3D) using the extended Delaunay technique.

nodes on an empty sphere with a radius greater than αh, are considered as boundary nodes. In practice α is a parameter close to, but greater than one. This criterion is coincident with the Alpha Shape concept [Edelsbrunner and Mucke (1999)]. Figure 7 shows an example of the boundary recognition using the Alpha Shape technique.

Once a decision has been made concerning which nodes are on the boundaries, the boundary surface and its normal are defined by all the polyhedral surfaces (or polygons in 2D) having all their nodes on the boundary and belonging to just one polyhedron.

The boundary definition allows us to compute the volume of each of the fluid and solid subdomains which is also an important task. In the criterion proposed above, the error in the boundary surface definition is proportional to h which is an acceptable error.

The method described also allows one to identify isolated fluid particles outside the main fluid domain. These particles are treated as part of the external boundary where the pressure is fixed to the atmospheric value (Figure 7). We recall that each particle is a material point characterized by the density of the solid or fluid domain to which it belongs. Mass is lost in the analysis domain when a boundary element is eliminated due to departure of a node (a particle) from the domain. This mass is however regained when the "flying" node falls down and a new boundary element is created by the Alpha Shape algorithm when the falling node is at a distance less than αh from the boundary. This concept is essential for modeling the splashing of surface waves and bed erosion as described in Section 8. An example of wave splashing is presented in Figure 8 where the motion of a fluid within an oscillating container is shown.

7.1 Contact between solid-solid interfaces

The contact between two solid interfaces can be modeled by introducing a layer of *contact elements* between the two interacting solid interfaces. This layer is automatically created during the mesh generation by prescribing a minimum distance between two solid boundaries. If the distance exceeds the minimum value, then the

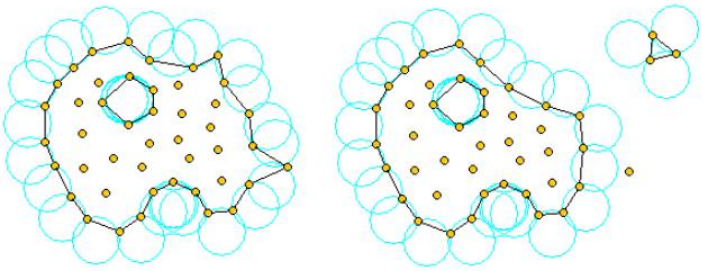


Figure 7: Identification of individual particles (or a group of particles) starting from a given collection of nodes.

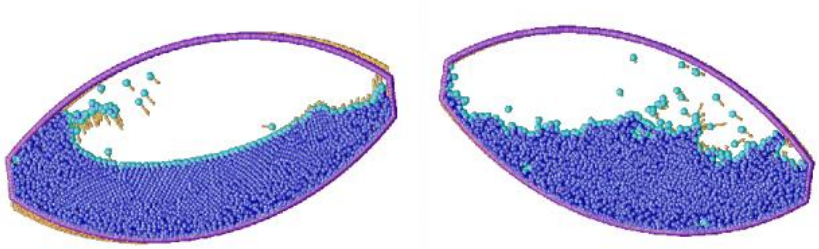


Figure 8: Motion of a liquid within an oscillating container. Position of the liquid particles at two different times. Arrows represent the velocity vector.

generated elements are treated as fluid (or air) elements. Otherwise the elements are treated as contact elements where a relationship between the tangential and normal forces and the corresponding displacement is introduced so as to model elastic and frictional contact in the normal and tangential directions, respectively (Figure 9).

This algorithm has proven to be very effective and it allows to identify and model complex frictional contact conditions between two or more interacting solids in an extremely simple manner. The accuracy on this contact model depends on the critical distance above mentioned.

Figure 10 shows an example of the contact algorithm in the analysis of the dragging of a cubic object by a water stream. The contact algorithm described above models accurately the frictional contact effects between the moving cube and the fixed bottom. Other examples of this kind can be found in Oñate *et al.* (2006).

8 MODELING OF BED EROSION

Bed erosion models are traditionally based on a relationship between the rate of erosion and the shear stress level [Kovacs and Parker (1994); Wan and Fell (2004)]. The effect of water velocity on soil erosion was studied in Parker *et al.* (1995). In our work we propose a simple erosion model based on the frictional work at the bed surface originated by the shear stresses in the fluid. The resulting erosion model

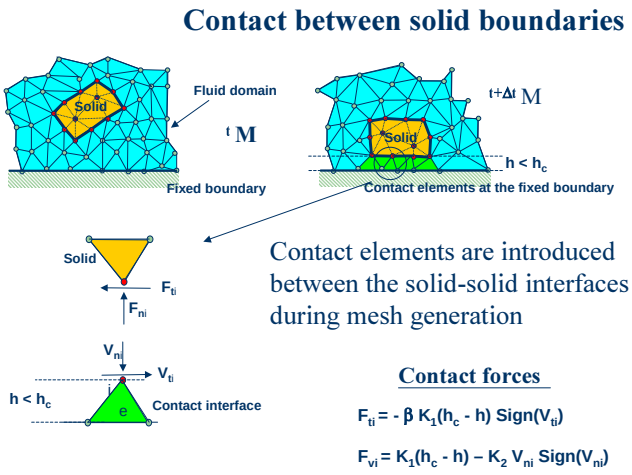


Figure 9: Contact conditions at a solid-solid interface

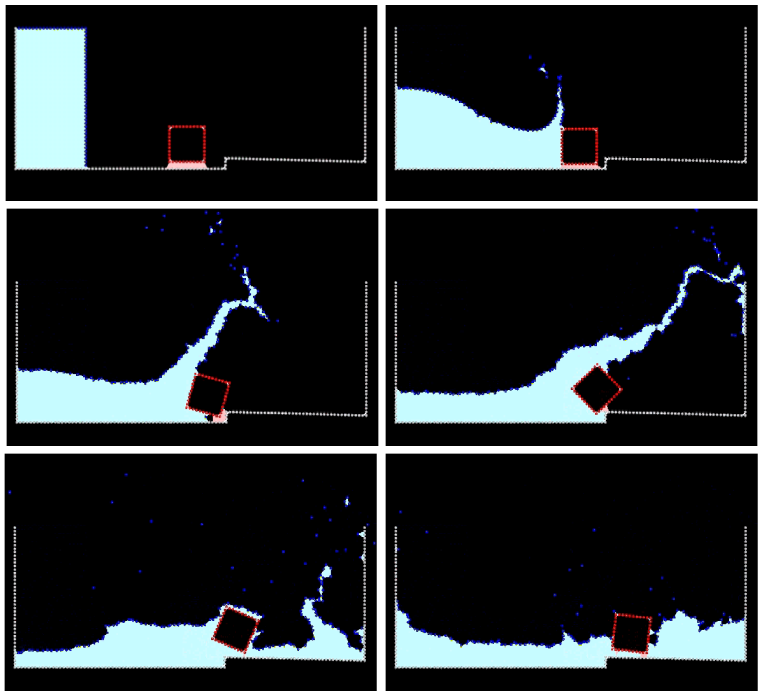


Figure 10: Dragging of a cubic object by a water stream. Note the contact elements at the cubic bottom interface.

ressembles Archard law typically used for modeling abbrasive wear in surfaces under frictional contact conditions [Archard (1953)]. An application of Archard law for

modeling surface wear in rock cutting tools can be found in Oñate and Rojek (2004).

The algorithm proposed to model the erosion of soil/rock particles at the fluid bed is the following:

1. Compute at every point of the bed surface the resultant tangential stress τ induced by the fluid motion. In 3D problems $\tau = (\tau_{s_n}^2 + \tau_{t_n})^2$ where s and t are the tangential stresses in the plane defined by the normal direction $\mathbf{n}$ at the bed node. The value of τ for 2D problems can be estimated as follows:

$$\tau_t = \mu \gamma_t \tag{32a}$$

with

$$\gamma_t = \frac{1}{2}\frac{\partial v_t}{\partial n} = \frac{v_t^k}{2h_k} \tag{32b}$$

where v_t^k is the modulus of the tangential velocity at the node k point (i.e. $v_t^k = (v_{s_n}^2 + v_{t_n}^2)^{1/2}$) and h_k is a prescribed distance along the normal of the bed node k. Typically h_k is of the order of magnitude of the smallest fluid element adjacent to node k (Figure 11).

2. Compute the frictional work originated by the tangential stresses at the bed surface as

$$W_f = \int_\circ^t \tau_t \gamma_t \, dt = \int_\circ^t \frac{\mu}{4} \left(\frac{v_t^k}{h_k} \right)^2 dt \tag{33}$$

Eq.(33) is integrated in time using a simple scheme as

$${}^nW_f = {}^{n-1}W_f + \tau \gamma_t \, \Delta t \tag{34}$$

3. The onset of erosion at a bed point occurs when ${}^nW_f$ exceeds a critical threshold value W_c defined empirically according to the specific properties of the bed material.

4. If ${}^nW_f > W_c$ at a bed node, then the node is detached from the bed region and it is allowed to move with the fluid flow, i.e. it becomes a fluid node. As a consequence, the mass of the patch of bed elements surrounding the bed node vanishes in the bed domain and it is transferred to the new fluid node. This mass is subsequently transported with the fluid. Conservation of mass of the bed particles within the fluid is guaranteed by changing the density of the new fluid node so that the mass of the suspended sediment traveling with the fluid equals the mass originally assigned to the bed node. Note that the mass assigned to a node is computed by multiplying the node density by the tributary domain of the node.

5. Sediment deposition is modeled by an inverse process to that described in the previous step. Hence, a suspended node adjacent to the bed surface with a velocity below a threshold value is assigned to the bed surface. This automatically leads to the generation of new bed elements adjacent to the boundary of the bed region. The original mass of the bed region is recovered by adjusting the density of the newly generated bed elements.

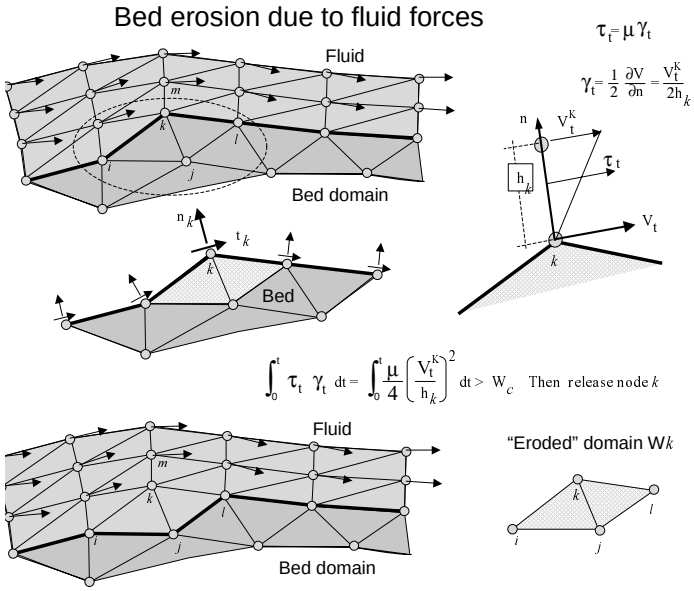


Figure 11: Modeling of bed erosion by dragging of bed material

Figure 11 shows an schematic view of the bed erosion algorithm proposed.

The examples chosen show the applicability of the PFEM to solve bed erosion problems in free surface flows.

9 EXAMPLES

We present next a collection of simple, schematic, but very illustrative two and three dimensional examples showing the potential of the PFEM formulation presented here to model bed erosion in complex free surface flows. Sediment deposition is not considered in any of the examples.

9.1 Example 1. Erosion of a sand hill under a water stream

Figure 12 shows the progressive erosion of a compacted sand domain under the action of an impacting water stream originated by a water jet. The situation is typical in sand shapes built by children in the beach and subsequently destroyed by dropping water on them. The frames in Figure 12 show the progressive erosion of the surface of the sand domain. A kind of hydraulic jump is generated by the water jet and the sand obstacle as clearly seen in the figures. The erosion process continues until the sand domain is fully dragged by the fluid flow.

9.2 Example 2. Erosion of a 3D earth dam due to an overspill stream

The second example illustrates the erosion of an earth dam under a water stream running over the dam top. An schematic geometry of the dam has been chosen to

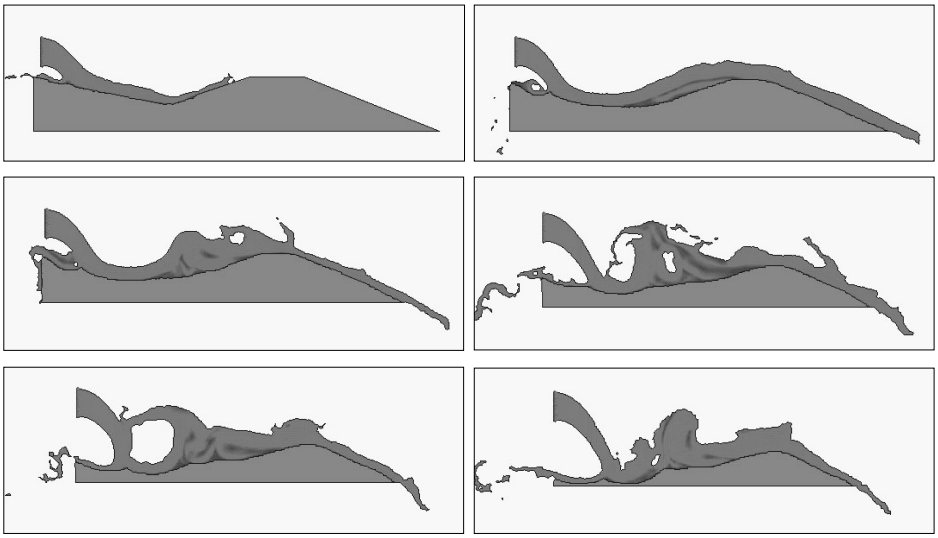


Figure 12: Erosion of a sand hill due to a water stream

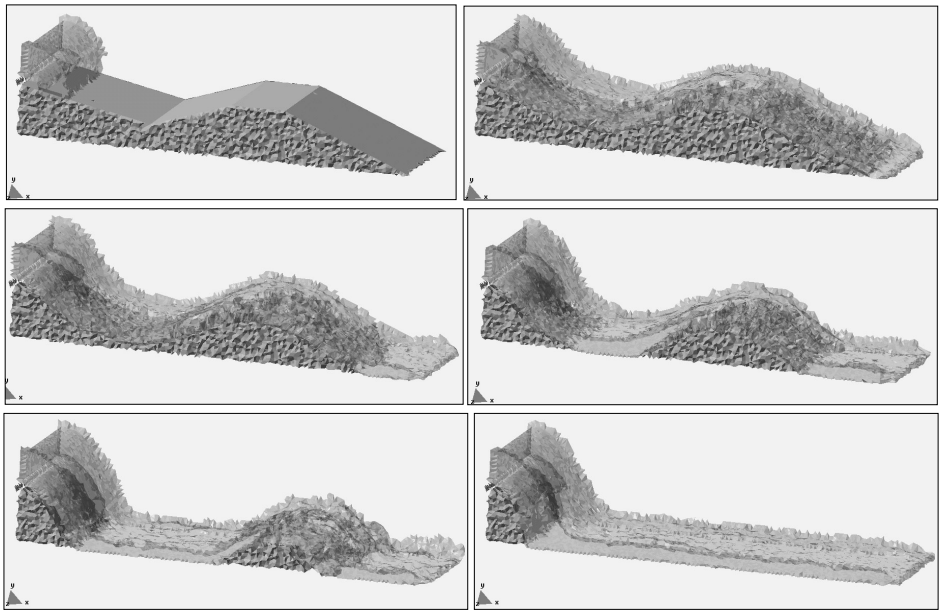


Figure 13: Erosion of a 3D earth dam due to an overspill streams.

simplify the computations. The images of Figure 13 show the progressive erosion of the dam surface until the whole dam is wiped out by the fluid flow.

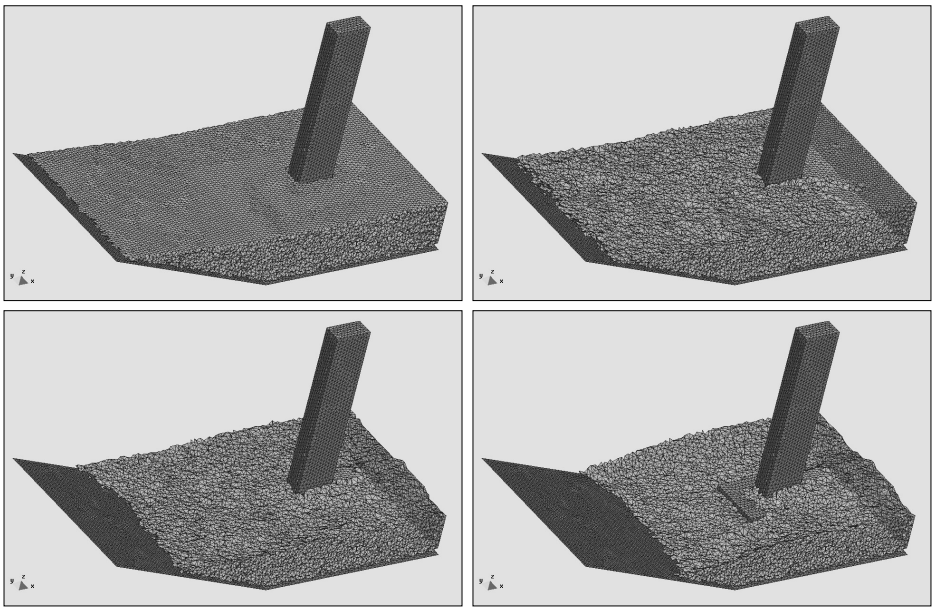


Figure 14: Evolution of the erosion of the soil in the vecinity of a bridge pile. Water particles are not shown.

9.3 Example 3. 3D erosion of a river bed adjacent to a bridge pile

The next example models the progressive erosion of a river bed domain in the vecinity of a bridge pile under a water stream. Figure 14 shows a view of the eroded bed surface at different times. The flowing water particles are not shown in the pictures, for clarity. The erosion process continues until the bridge pile foundation is unveiled by the erosion of the adjacent bed particles. We note that the deposition of the eroded particles was not modeled in this case.

9.4 Example 4. Erosion of a solid domain with an object on the top

This final example was chosen so as to demonstrate the effectiveness of the PFEM algorithm to combine the erosion process with the dragging of solid objects.

The pictures in Figure 15 represent schematically a temple on the top of a mountain. The mountain is progressively eroded by a strong water stream until the temple is dragged by the fluid.

10 CONCLUSIONS

The particle finite element method (PFEM) is a powerful technique to model bed erosion problems involving fluids with free surfaces and submerged or floating structures. Problems such as surface erosion, sediment transport and deposition, fluid-structure interaction, large motion of fluid or solid particles, surface waves, water splashing, separation of water drops, etc. can be solved with the PFEM. The

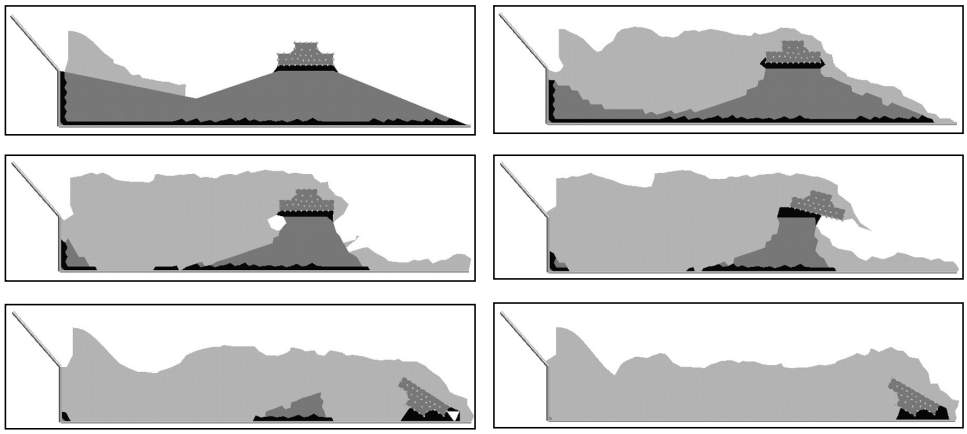


Figure 15: Erosion of a sand hill with an object on the top.

success of the method lies in the accurate and efficient solution of the coupled equations for an incompressible fluid and solid dynamics using an updated Lagrangian formulation and a stabilized finite element method. Low order elements with equal order interpolation for all the variables can be effectively used. Other essential solution ingredients are the fast regeneration of the finite element mesh using an extended Delaunay tesselation, the meshless finite element interpolation (MFEM), the identification of the boundary nodes using an Alpha Shape type technique and the simple algorithms to model onset of erosion, sediment transport and material deposition and contact conditions at the fluid-solid and solid-solid interfaces via mesh generation. The examples presented have shown the great potential of the PFEM for modeling bed erosion in complex free surface flows accounting for the dragging of solid objects. Applications of the sediment transport and the material deposition algorithm sketched in this paper will be reported in a forthcoming publication.

Acknowledgements

Thanks are given to Dr. R. Rossi and Ms M. de Mier for useful suggestions. This research was partially supported by the FP6 Programme of the European Commission on Global Change and Ecosystems through Project RAMWAS, Project no. FP6-037081.

REFERENCES

[1] Archard JF (1953) Contact and rubbing of flat surfaces. *J. Appl. Phys.* 24(8):981–988

[2] Aubry R, Idelsohn SR, Oñate E (2005) Particle finite element method in fluid mechanics including thermal convection-diffusion. *Computer & Structures* 83(17-18):1459–1475

[3] Codina R, Zienkiewicz OC (2002) CBS versus GLS stabilization of the incompressible Navier-Stokes equations and the role of the time step as stabilization parameter. *Communications in Numerical Methods in Engineering* 18(2):99–112

[4] Darby S, Thorne C (1996) Numerical simulation of widening and bed deformation of straight sand-bed rivers. *J. Hydr. Engrg. ASCE* 122(4): 184–193

[5] Donea J, Huerta A (2003) *Finite element method for flow problems*. J. Wiley

[6] Edelsbrunner H, Mucke EP (1999) Three dimensional alpha shapes. *ACM Trans. Graphics* 13:43–72

[7] Fell R, Wan CF, Cyganiewics J and Foster M (2003) Time for development of internal erosion and piping in embankment dams. *J. Geotechnical and Geoenvironmental Engineering* 129:307–314

[8] García J, Oñate E (2003) An unstructured finite element solver for ship hydrodynamic problems. *J. Appl. Mech.* 70:18–26 January

[9] Idelsohn SR, Oñate E, Del Pin F, Calvo N (2002) Lagrangian formulation: the only way to solve some free-surface fluid mechanics problems. Fith World Congress on *Computational Mechanics*, Mang HA, Rammerstorfer FG, Eberhardsteiner J (eds), July 7–12, Viena, Austria

[10] Idelsohn SR, Oñate E, Calvo N, Del Pin F (2003a) The meshless finite element method. *Int. J. Num. Meth. Engng.* 58(6):893–912

[11] Idelsohn SR, Oñate E, Del Pin F (2003b) A lagrangian meshless finite element method applied to fluid-structure interaction problems. *Computer & Structures* 81:655–671

[12] Idelsohn SR, Calvo N, Oñate E (2003c) Polyhedrization of an arbitrary point set. *Comput. Method Appl. Mech. Engng.* 192(22-24):2649–2668

[13] Idelsohn SR, Oñate E, Del Pin F (2004) 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

[14] Kovacs A, Parker G (1994) A new vectorial bedload formulation and its application to the time evolution of straight river channels. *J. Fluid Mech.* 267:153–183

[15] Oñate E (1998) Derivation of stabilized equations for advective-diffusive transport and fluid flow problems. *Comput. Meth. Appl. Mech. Engng.* 151:233–267

[16] Oñate E, (2000) A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. *Comp. Meth. Appl. Mech. Engng.* 182(1–2):355–370

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

[18] Oñate E, Idelsohn SR (1998) A mesh free finite point method for advective-diffusive transport and fluid flow problems. *Computational Mechanics* 21:283–292

[19] Oñate E, García J (2001) A finite element method for fluid-structure interaction with surface waves using a finite calculus formulation. *Comput. Meth. Appl. Mech. Engrg.* 191:635–660

[20] Oñate E, Sacco C, Idelsohn SR (2000) A finite point method for incompressible flow problems. *Comput. Visual. in Science* 2:67–75

[21] Oñate E, Idelsohn SR, Del Pin F (2003) Lagrangian formulation for incompressible fluids using finite calculus and the finite element method. *Numerical Methods for Scientific Computing Variational Problems and Applications*, Y Kuznetsov, P Neittanmaki and O Pironneau (Eds.), CIMNE, Barcelona

[22] Oñate E, García J, Idelsohn SR (2004a) Ship hydrodynamics. *Encyclopedia of Computational Mechanics*, E Stein, R de Borst and T.J.R. Hughes (Eds), J. Wiley.

[23] Oñate E, Idelsohn SR, Del Pin F, Aubry R (2004b) The particle finite element method. An overview. *Int. J. Comput. Methods* 1(2):267-307

[24] Oñate E, Rojek J (2004) Combination of discrete element and finite element method for dynamic analysis of geomechanic problems. *Comput. Meth. Appl. Mech. Engrg.* 193:3087–3128

[25] Oñate E, Idelsohn SR, Celigueta MA (2006) Lagrangian formulation for fluid-structure interaction problems using the particle finite element method. G Bugeda *et al.* (Eds.), CIMNE, Barcelona

[26] Parker DB, Michel TG, Smith JL (1995) Compaction and water velocity effects on soil erosion in shallow flow. *J. Irrigation and Drainage Engineering* 121:170–178

[27] Phillips BC, Sutherland AJ (1989) Spatial lag effects in bed load sediment transport. *J. Hydr. Res.* 24(1):45–56

[28] Rahuel JL, Holly FM, Belleudy PJ, Yang G (1989) Modeling of riverbed evolution for bedload sediment mixtures. *J. Hydr. Engrg., ASCE* 115(1):1521–1542

[29] Sekine M, Parker G (1992) Bed-load transport on transverse slope. *Int. J. Hydr. Engrg., ASCE* 118(4):513–535

[30] Struiksma N, Olesen KW, Flokstra C, Vriend HJ (1985) Bed deformation in curved alluvial channels. *J. Hydr. Res.* 23:57–79, Delft, The Netherlands

[31] van Rijn LC (1984) Sediment transport. Part III: Bed forms and alluvial roughness. *J. Hydr. Engrg., ASCE* 110(12):1733—1754

[32] van Rijn LC (1986) Mathematical modeling of suspended sediment in nonuniform flow. *J. Hydr. Engrg., ASCE* 112(6):433–455

[33] Wan CF and Fell R (2004) Investigation of erosion of soils in embankment dams. *J. Geotechnical and Geoenvironmental Engineering* 130:373–380

[34] Wu W, Rodi W, Wenka T (1997) Three-dimensional calculation of river flow. Proc. *XXVIIth IAHR Congr.*, International Association for Hydraulic Research, Delft, The Netherlands

[35] Zienkiewicz OC, Taylor RL, Nithiarasu P (2006) *The finite element method for fluid dynamics*. Elsevier

[36] Zienkiewicz OC, Taylor RL (2005) *The finite element method for solid and structural mechanics*. Elsevier

Paper 2

This paper, published in 2007, extends the research published in the previous paper ([2]) and adds the technology introduced in Section 3. The *PFlow* code was used for all the computations.

ADVANCES IN THE PARTICLE FINITE ELEMENT METHOD FOR THE ANALYSIS OF FLUID-MULTIBODY INTERACTION AND BED EROSION IN FREE SURFACE FLOWS

Eugenio Oñate, Sergio R. Idelsohn*, Miguel A. Celigueta
and **Riccardo Rossi**

International Center for Numerical Methods in Engineering (CIMNE)
Technical University of Catalonia
Campus Norte UPC, 08034 Barcelona, Spain
Email: onate@cimne.upc.edu
Web page: http://www.cimne.com
* ICREA Research Professor at CIMNE

Abstract. We present some advances in the formulation of the Particle Finite Element Method (PFEM) for solving complex fluid-structure interaction problems with free surface waves. In particular, we present extensions of the PFEM for the analysis of the interaction between a collection of bodies in water allowing for frictional contact conditions at the fluid-solid and solid-solid interfaces via mesh generation. An algorithm to treat bed erosion in free surface flows is also presented. Examples of application of the PFEM to solve a number of fluid-multibody interaction problems involving splashing of waves, large motions of floating and submerged bodies and bed erosion situations are given.

Key words: Lagrangian formulation, fluid-structure interaction, particle finite element method, bed erosion, free surface flows

1 INTRODUCTION

The analysis of problems involving the interaction of fluids and structures accounting for large motions of the fluid free surface and the existence of fully or partially submerged bodies which interact among themselves is of big relevance in many areas of engineering. Examples are common in ship hydrodynamics, off-shore and harbour structures, spill-ways in dams, free surface channel flows, environmental flows, liquid containers, stirring reactors, mould filling processes, etc.

Typical difficulties of fluid-multibody interaction analysis in free surface flows using the FEM with both the Eulerian and ALE formulation include the treatment of the convective terms and the incompressibility constraint in the fluid equations, the modelling and tracking of the free surface in the fluid, the transfer of information between the fluid and the moving solid domains via the contact interfaces, the modeling of wave splashing, the possibility to deal with large motions of the bodies

within the fluid domain, the efficient updating of the finite element meshes for both the structure and the fluid, etc. For a comprehensive list of references in FEM for fluid flow problems see [5, 34] and the references there included. A survey of recent works in fluid-structure interaction analysis can be found in [16], [25] and [32].

Most of the above problems disappear if a *Lagrangian description* is used to formulate the governing equations of both the solid and the fluid domains. In the Lagrangian formulation the motion of the individual particles are followed and, consequently, nodes in a finite element mesh can be viewed as moving material points (hereforth called "particles"). Hence, the motion of the mesh discretizing the total domain (including both the fluid and solid parts) is followed during the transient solution.

The authors have successfully developed in previous works a particular class of Lagrangian formulation for solving problems involving complex interaction between fluids and solids. The method, called the *particle finite element method* (PFEM), treats the mesh nodes in the fluid and solid domains as particles which can freely move and even separate from the main fluid domain representing, for instance, the effect of water drops. A finite element mesh connects the nodes defining the discretized domain where the governing equations are solved using a stabilized FEM based in the Finite Calculus (FIC) approach. An advantage of the Lagrangian formulation is that the convective terms disappear from the fluid equations. The difficulty is however transferred to the problem of adequately (and efficiently) moving the mesh nodes. We use a mesh regeneration procedure blending elements of different shapes using an extended Delaunay tesselation with special shape functions [9, 11]. The theory and applications of the PFEM are reported in [2, 4, 9, 10, 12, 13, 24, 25, 26, 28, 29].

The aim of this paper is to describe two recent advances of the PFEM: a) the analysis of the interaction between a collection of bodies which are floating and/or submerged in the fluid, and b) the modeling of bed erosion in open channel flows. Both problems are of great relevance in many areas of civil, marine and naval engineering, among others. It is shown in the paper that the PFEM provides a general analysis methodology for treat such a complex problems in a simple and efficient manner.

The layout of the paper is the following. In the next section the key ideas of the PFEM are outlined. Next the basic equations for an incompressible flow using a Lagrangian description and the FIC formulation are presented. Then a fractional step scheme for the transient solution is briefly described. Details of the treatment of the coupled FSI problem are given. The methods for mesh generation and for identification of the free surface nodes are outlined. The procedure for treating at mesh generation level the contact conditions at fluid-wall interfaces and the frictional contact interaction between moving solids is explained. A methodology for modeling bed erosion due to fluid forces is described. Finally, the efficiency of the PFEM is shown in its application to a number of problems involving large flow motions, surface waves, moving bodies in water and bed erosion.

2 THE BASIS OF THE PARTICLE FINITE ELEMENT METHOD

Let us consider a domain containing both fluid and solid subdomains. The moving fluid particles interact with the solid boundaries thereby inducing the deformation of the solid which in turn affects the flow motion and, therefore, the problem is fully coupled.

In the PFEM both the fluid and the solid domains are modelled using an *updated Lagrangian formulation*. That is, all variables in the fluid and solid domains 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 $t + \Delta t$ (Figure 1). The finite element method (FEM) is used to solve the continuum equations in both domains. Hence a mesh discretizing these domains must be generated in order to solve the governing equations for both the fluid and solid problems in the standard FEM fashion. Recall that the nodes discretizing the fluid and solid domains are treated as *material particles* which motion is tracked during the transient solution. This is useful to model the separation of fluid particles from the main fluid domain in a splashing wave, or soil particles in a bed erosion problem, and to follow their subsequent motion as individual particles with a known density, an initial acceleration and velocity and subject to gravity forces. The mass of a given domain is obtained by integrating the density at the different material points over the domain.

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.

2.1 Basic steps of the PFEM

For clarity purposes we will define the *collection or cloud of nodes (C)* pertaining to the fluid and solid domains, the *volume (V)* defining the analysis domain for the fluid and the solid and the *mesh (M)* discretizing both domains.

A typical solution with the PFEM involves the following steps.

1. The starting point at each time step is the cloud of points in the fluid and solid domains. For instance ${}^nC$ denotes the cloud at time $t = t_n$ (Figure 2).

2. Identify the boundaries for both the fluid and solid domains defining the analysis domain ${}^nV$ in the fluid and the solid. This is an essential step as some boundaries (such as the free surface in fluids) may be severely distorted during the solution, including separation and re-entering of nodes. The Alpha Shape method [6] is used for the boundary definition (Section 5).

3. Discretize the fluid and solid domains with a finite element mesh ${}^nM$. In our work we use an innovative mesh generation scheme based on the extended Delaunay tesselation (Section 4) [9, 10, 12].

4. Solve the coupled Lagrangian equations of motion for the fluid and the solid domains. Compute the relevant state variables in both domains at the next

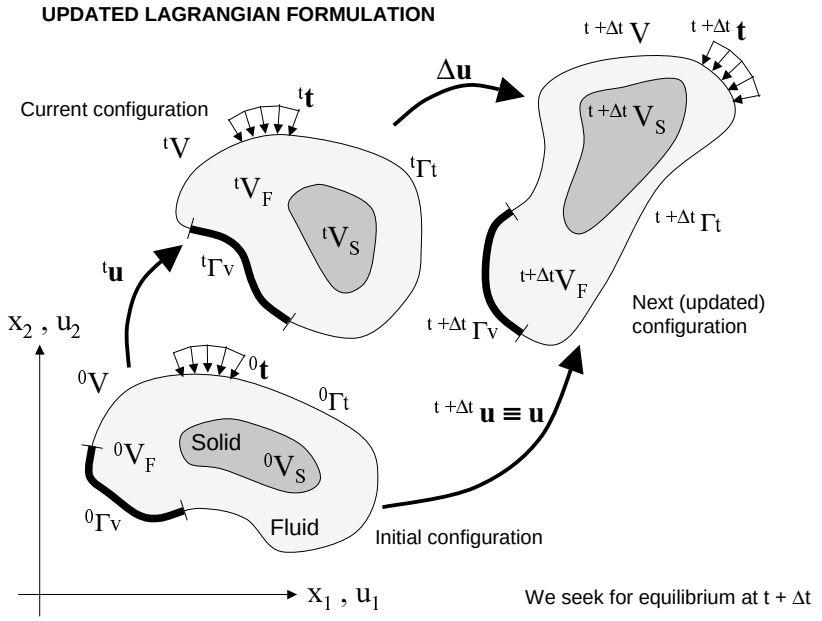


Figure 1: Updated lagrangian description for a continuum containing a fluid and a solid domain

(updated) configuration for $t + \Delta t$: velocities, pressure and viscous stresses in the fluid and displacements, stresses and strains in the solid.

5. Move the mesh nodes to a new position ${}^{n+1}C$ where $n + 1$ denotes the time $t_n + \Delta t$, in terms of the time increment size. This step is typically a consequence of the solution process of step 4.

6. Go back to step 1 and repeat the solution process for the next time step to obtain ${}^{n+2}C$. The process is shown in Figure 2.

2.2 Overview of the coupled FSI algoritm

Figure 3 shows a typical domain V with external boundaries Γ_V and Γ_t where the velocity and the surface tractions are prescribed, respectively. The domain V is formed by fluid (V_F) and solid (V_S) subdomains (i.e. $V = V_F \cup V_S$). Both subdomains interact at a common boundary Γ_{FS} where the surface tractions and the kinematic variables (displacements, velocities and acelerations) are the same for both subdomains. Note that both set of variables (the surface tractions and the kinematic variables) are equivalent in the equilibrium configuration.

Let us define ${}^tS$ and ${}^tF$ the set of variables defining the kinematics and the stress-strain fields at the solid and fluid domains at time t, respectively, i.e.

$$ {}^tS := [{}^t\mathbf{x}_s, {}^t\mathbf{u}_s, {}^t\mathbf{v}_s, {}^t\mathbf{a}_s, {}^t\boldsymbol{\varepsilon}_s, {}^t\boldsymbol{\sigma}_s, \cdots]^T \quad (1) $$

$$ {}^tF := [{}^t\mathbf{x}_F, {}^t\mathbf{u}_F, {}^t\mathbf{v}_F, {}^t\mathbf{a}_F, {}^t\dot{\boldsymbol{\varepsilon}}_F, {}^t\boldsymbol{\sigma}_F, \cdots]^T \quad (2) $$

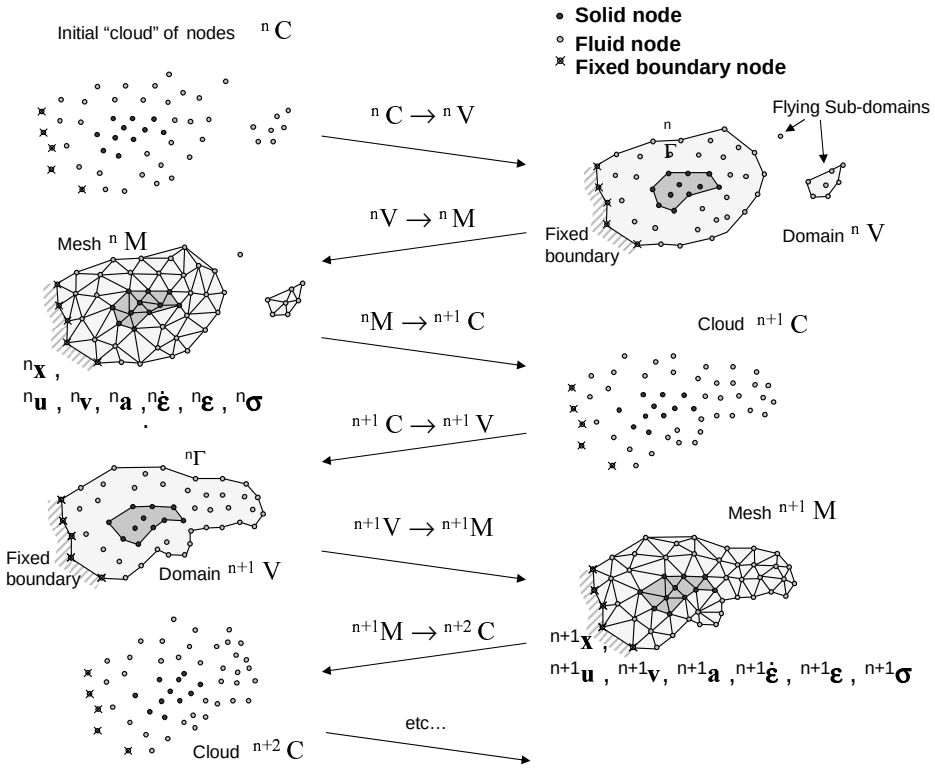


Figure 2: Sequence of steps to update a "cloud" of nodes from time n ($t = t_n$) to time $n + 2$ ($t = t_n + 2\Delta t$)

where $\mathbf{x}$ is the nodal coordinate vector, $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{a}$ are the vector of displacements, velocities and accelerations, respectively, $\boldsymbol{\varepsilon}, \dot{\boldsymbol{\varepsilon}}$ and $\boldsymbol{\sigma}$ are the strain vector, the strain-rate (or rate of deformation) vectors and the Cauchy stress vector, respectively and F and S denote the variables in the fluid and solid domains, respectively. In the discretized problem, a bar over these variables will denote nodal values.

The coupled fluid-structure interaction (FSI) problem of Figure 3 is solved, in this work, using the following *strongly coupled staggered scheme*:

0. We assume that the variables in the solid and fluid domains at time t (^{t}S and ^{t}F) are known.

1. Solve for the variables at the solid domain at time $t + \Delta t$ ($^{t+\Delta t}S$) under prescribed surface tractions at the fluid-solid boundary Γ_{FS}. The boundary conditions at the part of the external boundary intersecting the domain are the standard ones in solid mechanics.

2. Solve for the variables at the fluid domain at time $t + \Delta t$ ($^{t+\Delta t}F$) under prescribed surface tractions at the external boundary Γ_t and prescribed velocities at the external and internal boundaries Γ_V and Γ_{FS}, respectively.

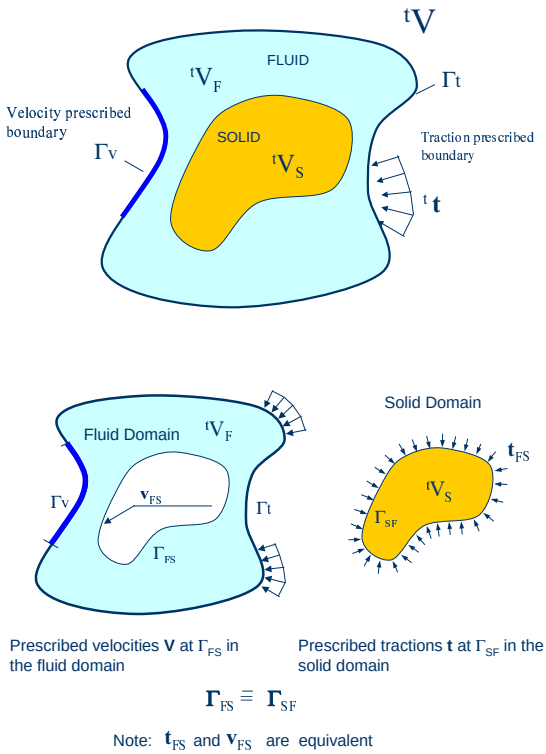


Figure 3: Split of the analysis domain V into fluid and solid subdomains. Equality of surface tractions and kinematic variables at the common interface

Iterate between 1 and 2 until convergence.

The variables at the solid domain ${}^{t+\Delta t}S$ are found via the integration of the equations of dynamic motion in the solid written as

$$\mathbf{M}_s\mathbf{a}_s + \mathbf{g}_s - \mathbf{f}_s = \mathbf{0} \tag{3}$$

where $\mathbf{M}_s, \mathbf{g}_s$ and $\mathbf{f}_s$ denote the mass matrix, the internal node force vector and the external nodal force vector at the solid domain. The time integration of Eq.(3) is performed using a standard Newmark method. An incremental iterative scheme is implemented within each time step to account for non linear geometrical and material effects.

The FEM solution of the variables in the (incompressible) fluid domain implies solving the momentum and incompressibility equations. As mentioned above this is not such as simple problem as the incompressibility condition limits the choice of the FE approximations for the velocity and pressure to overcome the well known *div*-stability condition [5, 34]. In our work we use a stabilized mixed FEM based on the Finite Calculus (FIC) approach which allows for a linear approximation for the velocity and pressure variables. Details of the FEM/FIC formulation are given in the next section.

Figure 4 shows a typical example of a PFEM solution in 2D. The pictures correspond to the analysis of the problem of breakage of a water column [12, 26]. Figure 4a shows the initial grid of four node rectangles discretizing the fluid domain and the solid walls. Figures 4b and 4c show the mesh for the solution at two later times.

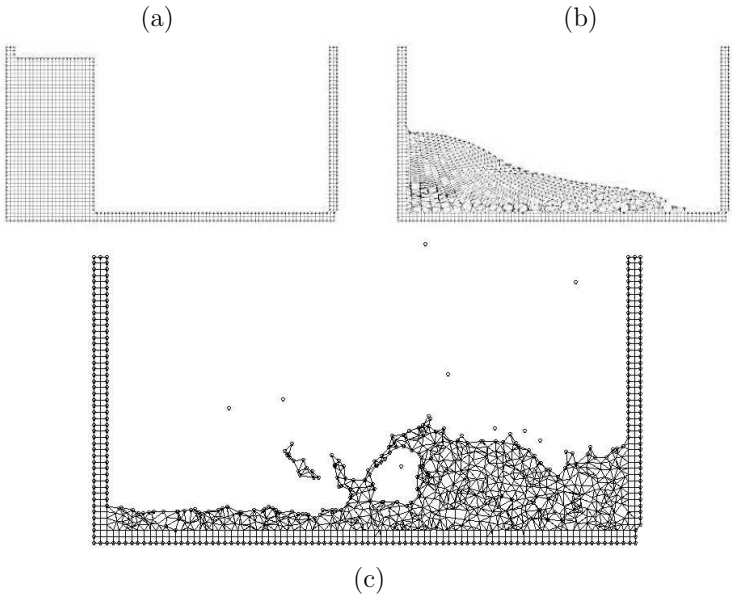


Figure 4: Breakage of a water column. (a) Discretization of the fluid domain and the solid walls. Boundary nodes are marked with circles. (b) and (c) Mesh in the fluid domain at two different times.

3 FIC/FEM FORMULATION FOR A LAGRANGIAN INCOMPRESSIBLE FLUID

The standard infinitesimal equations for a viscous incompressible fluid can be written in a Lagrangian frame as [17, 34].

Momentum

$$r_{m_i} = 0 \quad \text{in } V_F \tag{4}$$

Mass balance

$$r_d = 0 \quad \text{in } V_F \tag{5}$$

where

$$r_{m_i} = \rho \frac{\partial v_i}{\partial t} + \frac{\partial \sigma_{ij}}{\partial x_j} - b_i \quad , \quad \sigma_{ji} = \sigma_{ij} \tag{6}$$

$$r_d = \frac{\partial v_i}{\partial x_i} \qquad i,j = 1, n_d \tag{7}$$

Above n_d is the number of space dimensions, v_i is the velocity along the ith global axis ($v_i = \frac{\partial u_i}{\partial t}$, where u_i is the ith displacement), ρ is the (constant) density of the fluid, b_i are the body forces, σ_{ij} are the total stresses given by $\sigma_{ij} = s_{ij} - \delta_{ij} p$, p is the absolute pressure (defined positive in compression) and s_{ij} are the viscous deviatoric stresses related to the viscosity μ by the standard expression

$$s_{ij} = 2\mu \left(\dot{\varepsilon}_{ij} - \delta_{ij} \frac{1}{3} \frac{\partial v_k}{\partial x_k} \right) \tag{8}$$

where δ_{ij} is the Kronecker delta and the strain rates $\dot{\varepsilon}_{ij}$ are

$$\dot{\varepsilon}_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \tag{9}$$

In the above all variables are defined at the current time t (current configuration).

In our work we will solve a *modified set of governing* equations derived using a finite calculus (FIC) formulation. The FIC governing equations are [17, 18, 19, 21].

Momentum

$$r_{m_i} - \underline{\frac{1}{2} h_j \frac{\partial r_{m_i}}{\partial x_j}} = 0 \qquad \text{in } V_F \tag{10}$$

Mass balance

$$r_d - \underline{\frac{1}{2} h_j \frac{\partial r_d}{\partial x_j}} = 0 \qquad \text{in } V_F \tag{11}$$

The problem definition is completed with the following boundary conditions

$$n_j \sigma_{ij} - t_i + \underline{\frac{1}{2} h_j n_j r_{m_i}} = 0 \quad \text{on } \Gamma_t \tag{12}$$

$$v_j - v_j^p = 0 \quad \text{on } \Gamma_v \tag{13}$$

and the initial condition is $v_j = v_j^0$ for $t = t_0$. The standard summation convention for repeated indexes is assumed unless otherwise specified.

In Eqs.(12) and (13) t_i and v_j^p are surface tractions and prescribed velocities on the boundaries Γ_t and Γ_v, respectively, n_j are the components of the unit normal vector to the boundary. Recall that Γ_v includes both the external boundary and the internal boundary Γ_{F_S} (Figure 3).

The $h_i's$ in above equations are *characteristic lengths* of the domain where balance of momentum and mass is enforced. In Eq.(12) these lengths define the domain where equilibrium of boundary tractions is established. We note that at the discretized level, the $h_i's$ become of the order of a typical element or grid dimension [17, 18, 19].

Eqs.(10)–(13) are the starting point for deriving stabilized finite element methods to solve the incompressible Navier-Stokes equations in a Lagrangian frame of reference using equal order interpolation for the velocity and pressure variables [2, 8, 9, 10, 12, 24]. Application of the FIC formulation to finite element and meshless analysis of fluid flow problems can be found in [7, 18, 19, 20, 21, 23, 25, 27].

3.1 Transformation of the mass balance equation. Integral governing equations

The underlined term in Eq.(11) can be expressed in terms of the momentum equations. The new expression for the mass balance equation is [18, 26]

$$r_d - \sum_{i=1}^{n_d} \tau_i \frac{\partial r_{m_i}}{\partial x_i} = 0 \tag{14}$$

with

$$\tau_i = \frac{3h_i^2}{8\mu} \tag{15}$$

In our work we have taken the characteristic distances h_i to be constant within each element and equal to a typical element dimension computed as $h^e = [V^e]^m$ where V^e is the element domain and $m = 1/2$ for 2D problems and $m = 1/3$ for 3D problems.

At this stage it is no longer necessary to retain the stabilization terms in the momentum equations and the traction boundary conditions (Eqs.(10) and (12)). These terms are critical in Eulerian formulations to stabilize the numerical solution for high values of the convective terms [18, 21, 27, 28].

The weighted residual expression of the final form of the momentum and mass balance equations is written as

$$\int_{V_F} \delta v_i r_{m_i} dV + \int_{\Gamma_t} \delta v_i (n_j \sigma_{ij} - t_i) d\Gamma = 0 \tag{16}$$

$$\int_{V_F} q \left[r_d - \sum_{i=1}^{n_d} \tau_i \frac{\partial r_{m_i}}{\partial x_i} \right] dV = 0 \tag{17}$$

where δv_i and q are arbitrary weighting functions equivalent to virtual velocity and virtual pressure fields.

The r_{m_i} term in Eq.(17) and the deviatoric stresses and the pressure terms within r_{m_i} in Eq.(16) are integrated by parts to give

$$\int_{V_F} \left[\delta v_i \rho \frac{\partial v_i}{\partial t} + \delta \dot{\varepsilon}_{ij} (s_{ij} - \delta_{ij} p) \right] dV - \int_{V_F} \delta v_i b_i d\Omega - \int_{\Gamma_t} \delta v_i t_i d\Gamma = 0 \tag{18}$$

$$\int_{V_F} q \frac{\partial v_i}{\partial x_i} dV + \int_{V_F} \left[\sum_{i=1}^{n_d} \tau_i \frac{\partial q}{\partial x_i} r_{m_i} \right] dV = 0 \tag{19}$$

In Eq.(18) $\delta \dot{\varepsilon}_{ij}$ are virtual strain rates. Note that the boundary term resulting from the integration by parts of r_{m_i} in Eq.(19) has been neglected in this work. Retaining this term has been recently found to be advantageous for enhancing the satisfaction of the incompressibility condition in FEM predictor-corrector schemes for incompressible fluid flow analysis [30].

3.2 Pressure gradient projection

The computation of the residual terms in Eq.(19) is simplified if we introduce the pressure gradient projections π_i, defined as

$$\pi_i = r_{m_i} - \frac{\partial p}{\partial x_i} \tag{20}$$

We express r_{m_i} in Eq.(19) in terms of the π_i which then become additional variables. The system of integral equations is now augmented in the necessary number of equations by imposing that the residual r_{m_i} vanishes within the analysis domain (in an average sense). This gives the final system of governing equation as:

$$\int_{V_F} \left[\delta v_i \rho \frac{\partial v_i}{\partial t} + \delta \dot{\varepsilon}_{ij} (s_{ij} - \delta_{ij} p) \right] dV - \int_{V_F} \delta v_i b_i dV - \int_{\Gamma_t} \delta v_i t_i d\Gamma = 0 \tag{21}$$

$$\int_{V_F} q \frac{\partial v_i}{\partial x_i} dV + \int_{V_F} \sum_{i=1}^{n_d} \tau_i \frac{\partial q}{\partial x_i} \left(\frac{\partial p}{\partial x_i} + \pi_i \right) dV = 0 \tag{22}$$

$$\int_{V_F} \delta \pi_i \tau_i \left(\frac{\partial p}{\partial x_i} + \pi_i \right) dV = 0 \qquad \text{no sum in } i \tag{23}$$

with $i, j, k = 1, n_d$. In Eqs.(23) $\delta \pi_i$ are appropriate weighting functions and the τ_i weights are introduced for symmetry reasons.

3.3 Finite element discretization

We choose equal order C° continuous interpolations of the velocities, the pressure and the pressure gradient projections π_i over each element with n nodes. The interpolations are written as

$$v_i = \sum_{j=1}^{n} N_j \bar{v}_i^j \quad , \quad p = \sum_{j=1}^{n} N_j \bar{p}^j \quad , \quad \pi_i = \sum_{j=1}^{n} N_j \bar{\pi}_i^j \tag{24}$$

where $\bar{(\cdot)}^j$ denotes nodal variables and N_j are the shape functions [34]. More details of the mesh discretization process and the choice of shape functions are given in Section 4.

Substituting the approximations (24) into Eqs.(21–23) and choosing a Galerkin form with $\delta v_i = q = \delta \pi_i = N_i$ leads to the following system of discretized equations

$$\mathbf{M}\dot{\bar{\mathbf{v}}} + \mathbf{K}\bar{\mathbf{v}} - \mathbf{G}\bar{\mathbf{p}} - \mathbf{f} = \mathbf{0} \tag{25a}$$

$$\mathbf{G}^T\bar{\mathbf{v}} + \mathbf{L}\bar{\mathbf{p}} + \mathbf{Q}\bar{\boldsymbol{\pi}} = \mathbf{0} \tag{25b}$$

$$\mathbf{Q}^T\bar{\mathbf{p}} + \hat{\mathbf{M}}\bar{\boldsymbol{\pi}} = \mathbf{0} \tag{25c}$$

The form of the element matrices and vectors in Eqs.(25) can be found in [28].

3.4 Fractional step algorithm for the fluid variables

The starting point of the iterative algorithm are the variables at time n in the fluid domain (${}^nF$). The sought variables are the variables at time $n+1$ (${}^{n+1}F$). For the sake of clarity we will skip the upper left index $n+1$ for all variables, i.e.

$$ {}^{n+1}\bar{\mathbf{x}} \equiv \bar{\mathbf{x}} \; ; \; {}^{n+1}\bar{\mathbf{p}} \equiv \bar{\mathbf{p}} \; ; \; {}^{n+1}\bar{\boldsymbol{\pi}} \equiv \bar{\boldsymbol{\pi}} \; ; \; {}^{n+1}\bar{\mathbf{x}} \equiv \bar{\mathbf{x}} \; ; \text{etc.} \tag{26} $$

A simple iterative algorithm is obtained by splitting the pressure from the momentum equations as follows

$$ \bar{\mathbf{v}}^* = {}^n\bar{\mathbf{v}} - \Delta t \mathbf{M}^{-1}[\mathbf{K}\bar{\mathbf{v}}^j - \mathbf{G}{}^n\mathbf{p} - \mathbf{f}] \tag{27} $$

$$ \bar{\mathbf{v}}^{j+1} = \bar{\mathbf{v}}^* + \Delta t \mathbf{M}^{-1}\mathbf{G}\delta\bar{\mathbf{p}} \tag{28} $$

where $\delta\bar{\mathbf{p}}$ denotes the pressure increment. In above equations and in the following the left upper index n refers to values in the current configuration ${}^nV$, whereas the right upper index j denotes the iteration number within each time step.

The value of $\bar{\mathbf{v}}^{j+1}$ from Eq.(29) is substituted now into Eq.(25b) to give

$$ \mathbf{G}^T\bar{\mathbf{v}}^* + \Delta t \mathbf{S}\delta\bar{\mathbf{p}} + \mathbf{L}\bar{\mathbf{p}}^{j+1} + \mathbf{Q}\bar{\boldsymbol{\pi}}^j = \mathbf{0} \tag{29a} $$

where

$$ \mathbf{S} = \mathbf{G}^T\mathbf{M}^{-1}\mathbf{G} \tag{29b} $$

Typically matrix $\mathbf{S}$ is computed using a diagonal matrix $\mathbf{M} = \mathbf{M}_d$, where the subscript d denotes hereonward a diagonal matrix. We note that the form of $\mathbf{S}$ in Eq.(29b) avoids the need for prescribing the pressure at the boundary nodes.

An alternative is to approximate $\mathbf{S}$ by a Laplacian matrix. This reduces considerably the bandwidth of $\mathbf{S}$. The disadvantage is that the pressure increment must be then prescribed on the free surface and this reduces the accuracy in the satisfaction of the incompressibility condition in these regions. These problems are overcome however by retaining the residual term r_{m_i} in the boundary integral resulting from the integration by parts of Eq.(17) [30]. In this work however the form of matrix $\mathbf{S}$ given by Eq.(29a) has been used.

A semi-implicit algorithm can be derived as follows. For each iteration:

Step 1 Compute $\mathbf{v}^*$ from Eq.(27) with $\mathbf{M} = \mathbf{M}_d$. For the first iteration $(\bar{\mathbf{v}}^1, \bar{\mathbf{p}}^1, \bar{\boldsymbol{\pi}}^1, \bar{\mathbf{x}}^1) \equiv ({}^n\bar{\mathbf{v}}, {}^n\bar{\mathbf{p}}, {}^n\bar{\boldsymbol{\pi}}, {}^n\bar{\mathbf{x}})$

Step 2 Compute $\delta\bar{\mathbf{p}}$ and $\mathbf{p}^{j+1}$ from Eq.(29a) as

$$ \delta\bar{\mathbf{p}} = -(\mathbf{L} + \Delta t\mathbf{S})^{-1}[\mathbf{G}^T\bar{\mathbf{v}}^* + \mathbf{Q}\bar{\boldsymbol{\pi}}^j + \mathbf{L}\bar{\mathbf{p}}^j] \tag{30a} $$

The pressure $\bar{\mathbf{p}}^{n+1,j}$ is computed as

$$ \bar{\mathbf{p}}^{j+1} = \bar{\mathbf{p}}^j + \delta\bar{\mathbf{p}}^j \tag{30b} $$

Step 3 Compute $\bar{\mathbf{v}}^{j+1}$ from Eq.(28) with $\mathbf{M} = \mathbf{M}_d$

Step 4 Compute $\bar{\boldsymbol{\pi}}^{j+1}$ from Eq.(25c) as

$$\bar{\boldsymbol{\pi}}^{j+1} = -\hat{\mathbf{M}}_d^{-1}\mathbf{Q}^T\bar{\mathbf{p}}^{j+1} \tag{31}$$

Step 5 Update the coordinates of the mesh nodes as

$$\mathbf{x}_i^{j+1} = {}^n\mathbf{x}_i + \bar{\mathbf{v}}_i^{j+1}\Delta t \tag{32}$$

Step 6 Check the convergence of the velocity and pressure fields. If convergence is achieved move to the next time step, otherwise return to step 1 for the next iteration with $j \leftarrow j+1$.

Note that solution of steps 1, 3 and 4 does not require the solution of a system of equations as a diagonal form is chosen for $\mathbf{M}$ and $\hat{\mathbf{M}}$.

In the examples presented in the paper the time increment size has been chosen as

$$\Delta t = \min(\Delta t_i) \quad \text{with} \quad \Delta t_i = \frac{h_i^{\min}}{|\mathbf{v}|} \tag{33}$$

where $h_i^{\min}$ is the distance between node i and the closest node in the mesh.

Although not explicitely mentioned all matrices and vectors in Eqs.(25) are computed at the updated configuration ${}^{n+1}V_F$. This means that the integration domain changes for each iteration and, hence, all the terms involving space derivatives must be updated at each iteration.

The boundary conditions are applied as follows. No condition is applied for the computation of the fractional velocities $\mathbf{v}^*$ in Eq.(27). The prescribed velocities at the boundary are applied when solving for $\bar{\mathbf{v}}^{j+1}$ in step 3. As mentioned earlier there is no need for prescribing the pressure at the boundary if the form of Eq.(29b) is chosen for $\mathbf{S}$.

Box 1 shows a summary of the staggered scheme used for the solution for the variables in the solid and fluid domain at the updated configuration $({}^{n+1}F, {}^{n+1}S)$.

4 GENERATION OF A NEW MESH

One of the key points for the success of the PFEM is the fast regeneration of a mesh at every time step on the basis of the position of the nodes in the space domain. Indeed, any fast meshing algorithm can be used for this purpose. In our work the mesh is generated at each time step using the so called extended Delaunay tesselation (EDT) presented in [9, 11, 12]. The EDT allows one to generate non standard meshes combining elements of arbitrary polyhedrical shapes (triangles, quadrilaterals and other polygons in 2D and tetrahedra, hexahedra and arbitrary polyhedra in 3D) in a computing time of order n, where n is the total number of nodes in the mesh (Figure 5). The C° continuous shape functions of the elements can be simply obtained using the so called meshless finite element interpolation (MFEM). In our work the simpler linear C° interpolation has been chosen. Details of the mesh generation procedure and the derivation of the linear MFEM shape functions can be found in [9, 11, 12].

Figure 6 shows the evolution of the CPU time required for generating the mesh, for solving the system of equations and for assembling such a system in terms of the

LOOP OVER TIME STEPS $n = 1,..n_{\text{time}}$

$^{n}S, \quad ^{n}F$

LOOP OVER STAGGERED SOLUTION $j = 1,..n_{\text{stag}}$

Solve for solid variables (prescribed tractions at $^{n+1}\Gamma_{FS}$)

LOOP OVER ITERATIONS $i = 1,..n_{\text{iter}}$

Solve for $^{n+1}S_j^i$

Integrate Eq.(3) using a Newmark scheme
Check convergence. Yes: solve for fluid variables

NO: Next iteration $i \leftarrow i+1$

Solve for fluid variables (prescribed velocities at $^{n+1}\Gamma_{FS}$)

LOOP OVER ITERATIONS $i = 1,..n_{\text{iter}}$

Solve for $^{n+1}F_j^i$ using the scheme of Section 3.4

Check convergence. Yes: go to C

Next iteration $i \leftarrow i+1$

C Check convergence of surface tractions at $^{n+1}\Gamma_{FS}$

Yes: Next time step

Next staggered solution $j \leftarrow j+1,\ i \leftarrow i+1$

Next time step $^{n+1}S \leftarrow^{n+1} S_j^i,\ ^{n+1}F \leftarrow^{n+1} F_j^i$

Box 1. Staggered scheme for the FSI problem (see also Figure 3)

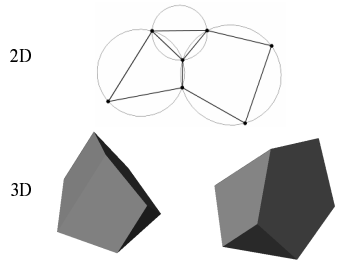


Figure 5: Generation of non standard meshes combining different polygons (in 2D) and polyhedra (in 3D) using the extended Delaunay technique.

number of nodes. the numbers correspond to the solution of a 3D flow in an open channel with the PFEM. The figure shows the CPU time in seconds for each time

step of the algorithm of Section 3.4. It is clearly seen that the CPU time required for meshing grows linearly with the number of nodes, as expected. Note also that the CPU time for solving the equations exceeds that required for meshing as the number of nodes increases. This situation has been found in all the problems solved with the PFEM. As a general rule for large 3D problems meshing consumes around 30% of the total CPU time for each time step, while the solution of the equations and the assembling of the system consume approximately 40% and 20% of the CPU time for each time step, respectively. These figures prove that the generation of the mesh has an acceptable cost in the PFEM solution. An improvement of the mesh generation process will in any case help to reducing the computational cost.

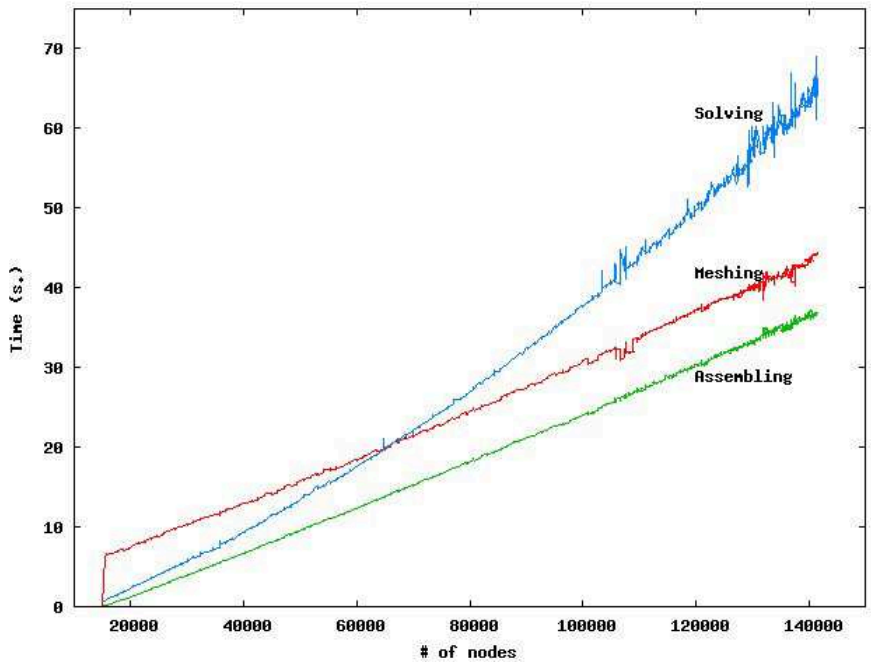


Figure 6: 3D flow problem solved with the PFEM. CPU time for meshing, assembling and solving the system of equations at each time step in terms of the number of nodes

5 IDENTIFICATION OF BOUNDARY SURFACES

One of the main tasks in the PFEM is the correct definition of the boundary domain. Boundary nodes are sometimes explicitly identified. In other cases, the total set of nodes is the only information available and the algorithm must recognize the boundary nodes.

In our work we use an extended Delaunay partition for recognizing boundary nodes. Considering that the nodes follow a variable $h(x)$ distribution, where $h(x)$ is typically the minimum distance between two nodes, the following criterion has been used. *All nodes on an empty sphere with a radius greater than αh, are considered as*

boundary nodes. In practice α is a parameter close to, but greater than one. Values of α ranging between 1.3 and 1.5 have been found to be optimal in all examples analyzed. This criterion is coincident with the Alpha Shape concept [6]. Figure 7 shows an example of the boundary recognition using the Alpha Shape technique.

Once a decision has been made concerning which nodes are on the boundaries, the boundary surface is defined by all the polyhedral surfaces (or polygons in 2D) having all their nodes on the boundary and belonging to just one polyhedron.

The method described also allows one to identify isolated fluid particles outside the main fluid domain. These particles are treated as part of the external boundary where the pressure is fixed to the atmospheric value. We recall that each particle is a material point characterized by the density of the solid or fluid domain to which it belongs. The mass which is lost when a boundary element is eliminated due to departure of a node (a particle) from the main analysis domain is again regained when the "flying" node falls down and a new boundary element is created by the Alpha Shape algorithm (Figures 2 and 7).

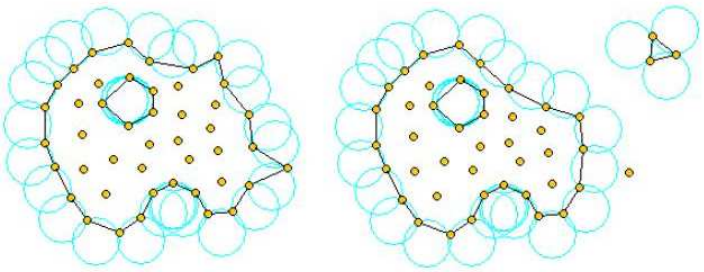


Figure 7: Identification of individual particles (or a group of particles) starting from a given collection of nodes.

The boundary recognition method above described is also useful for detecting contact conditions between the fluid domain and a fixed boundary, as well as between different solids interacting with each other. The contact detection procedure is detailed in Section 6.

In order to show the quality of the boundary recognition approach, the following simple example has been performed. A square fluid domain of 0.25m^2 is at a stationary position within a recipient (Figure 8). Then, as time evolves, the fluid falls down into the lower part of the recipient due to gravity effects. At the end of the process the total volume of the fluid within the recipient must be the same as that of the initial square domain. It must be noted that during the different time steps, the fluid has completely different free-surfaces including waves, breaking waves and fluid fragmentation zones.

The meshes used have average element sizes of 0.05m, 0.025m and 0.01m each which correspond to a total initial number of particles of 161, 552 and 3105 each. A value of $\alpha = 1.4$ for the Alpha-Shape method was used for the three analyses. Figure 8 shows the initial position of the fluid domain and one of the three meshes used for the analysis. Figure 9 shows the fluid domain at different time steps.

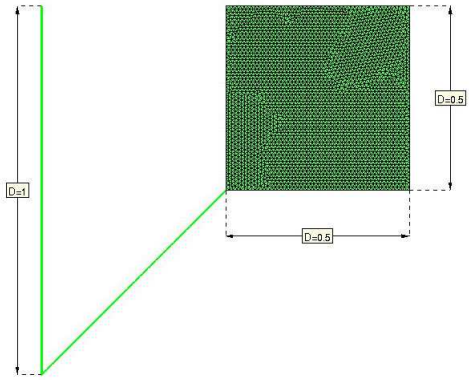
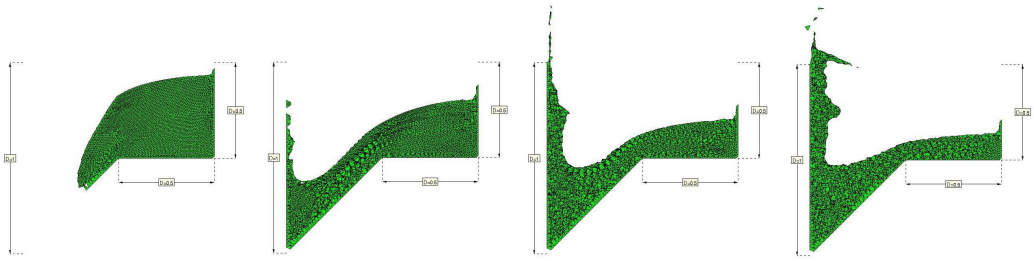
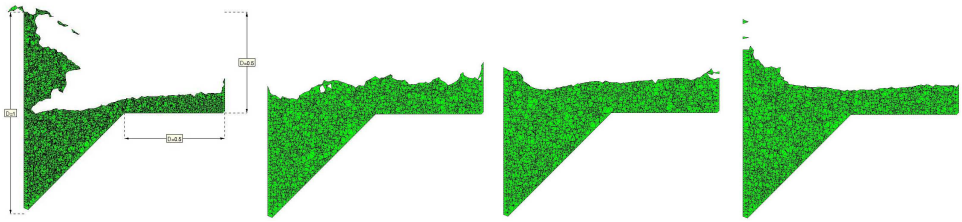


Figure 8: Fluid domain following into a recipient. Initial position. Fine mesh of 3105 nodes (element size of 0.01m)



(a) $t = 0.2$s (b) $t = 0.4$s (c) $t = 0.6$s (d) $t = 0.8$s



(e) $t = 1.0$s (f) $t = 2.0$s (g) $t = 3.0$s (h) $t = 4.0$s

Figure 9: Positions of the fluid domain at different time steps.

This simple example is interesting to show the quality of the boundary identification procedure. Another aim is to evaluate the volume variation from the incompressibility point of view, as well as the preservation of the total volume of the fluid due to possible errors in the boundary recognition using the Alpha-Shape method. Figure 10 shows the total fluid volume during the different time steps for the three different meshes. The charge of volume is insignificant for the fine mesh and becomes larger but acceptable for the coarse meshes.

It must be noted that the main difference between the PFEM and the classical FEM is just the remeshing technique and the evaluation of the boundary position at

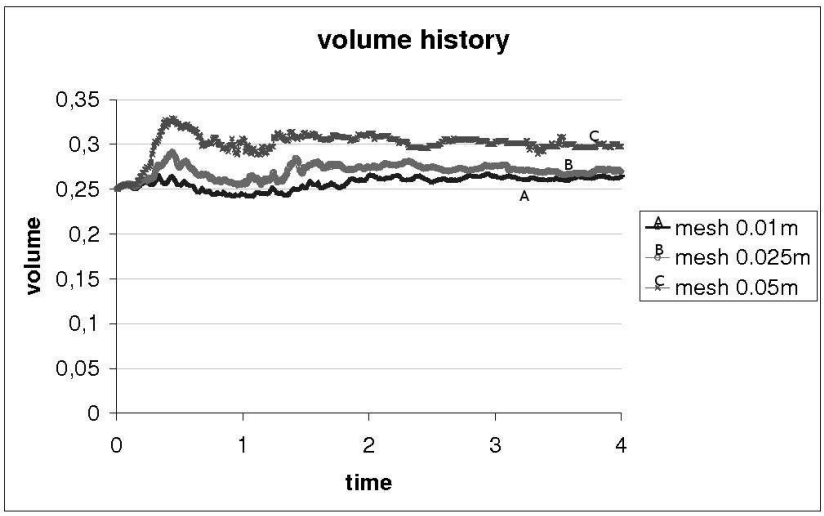


Figure 10: Total volume change as a function of time for different meshes.

each time step. The rest of the steps in the computation are coincident with those of the classical FEM. This simple example shows that in spite of the permanent remeshing and the evaluation of the boundary position via the Alpha-Shape method, the total fluid mass is preserved. We note however, that a good selection of the α parameter is essential for the good behaviour of the boundary recognition process. Examples showing the accuracy of the PFEM for fixed boundary problems can be found in [2].

6 TREATMENT OF CONTACT CONDITIONS IN THE PFEM

6.1 Contact between the fluid and a fixed boundary

The motion of the solid is governed by the action of the fluid flow forces induced by the pressure and the viscous stresses acting at the common boundary Γ_{FS}, as mentioned above.

The condition of prescribed velocities at the fixed boundaries in the PFEM are 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* (Figure 11). This simple way to treat the fluid-wall contact at mesh generation level is a distinct and attractive feature of the PFEM formulation.

6.2 Contact between solid-solid interfaces

The contact between two solid interfaces is simply treated 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

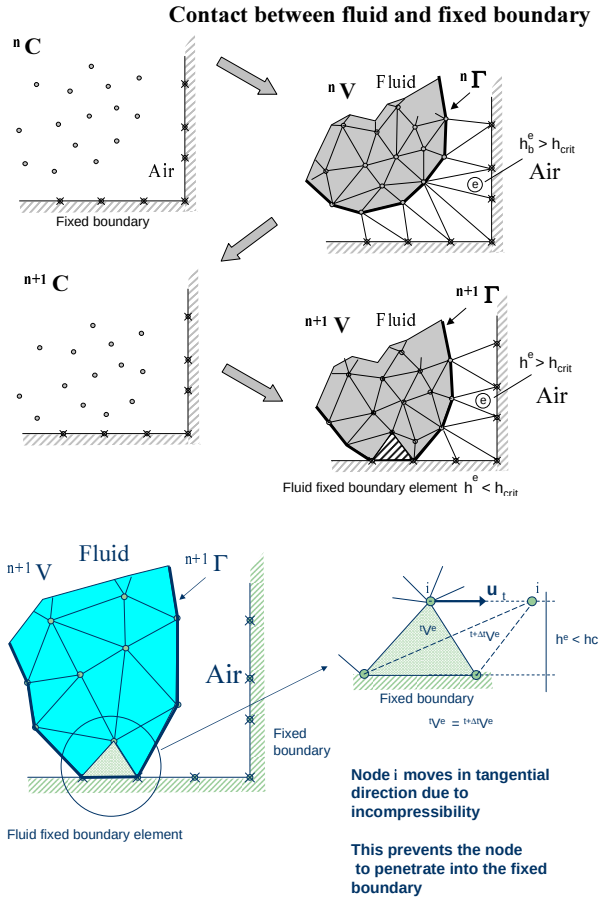


Figure 11: Automatic treatment of contact conditions at the fluid-wall interface

(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 treated as contact elements where a relationship between the tangential and normal forces and the corresponding displacement is introduced so as to model elastic and frictional contact effects in the normal and tangential directions, respectively (Figure 12).

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. Of course the accuracy of this contact model depends on the critical distance above mentioned.

This contact algorithm can also be used effectively to model frictional contact

conditions between rigid or elastic solids in standard structural mechanics applications. Figures 13–16 show examples of application of the contact algorithm to the bumping of a ball falling in a container, the failure of a domino set, the failure of an arch formed by a collection of stone blocks under a seismic loading and the motion of five tetrapods as they fall and slip over an inclined plane, respectively. The images in Figures 13 and 17 show explicitely the layer of contact elements which controls the accuracy of the contact algorithm.

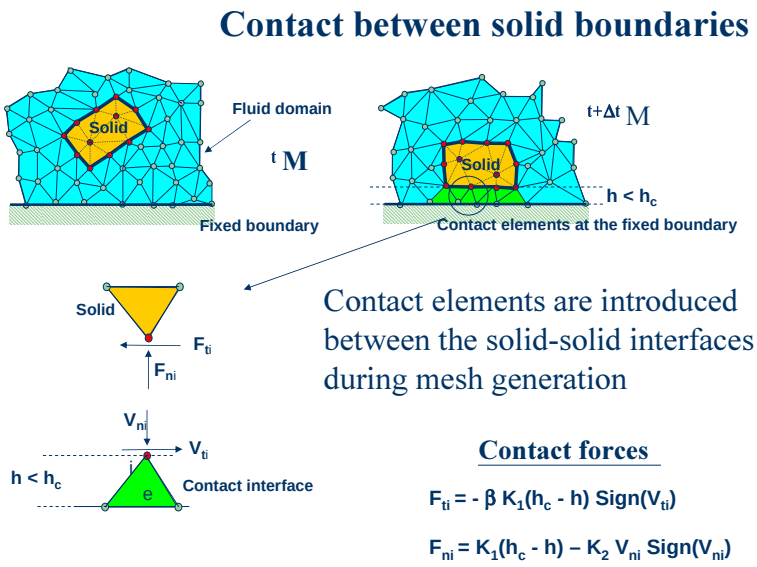


Figure 12: Contact conditions at a solid-solid interface

7 MODELING OF BED EROSION

Prediction of bed erosion and sediment transport in open channel flows are important tasks in many areas of river and environmental engineering. Bed erosion can lead to instabilities of the river basin slopes. It can also undermine the foundation of bridge piles thereby favouring structural failure. Modeling of bed erosion is also relevant for predicting the evolution of surface material dragged in earth dams in overspill situations. Bed erosion is one of the main causes of environmental damage in floods.

Bed erosion models are traditionally based on a relationship between the rate of erosion and the shear stress level [14, 33]. The effect of water velocity on soil erosion was studied in [31]. In a recent work we have proposed an extension of the PFEM to model bed erosion [29]. The erosion model is based on the frictional work at the bed surface originated by the shear stresses in the fluid. The resulting erosion model resembles Archard law typically used for modeling abrasive wear in surfaces under frictional contact conditions [1, 22].

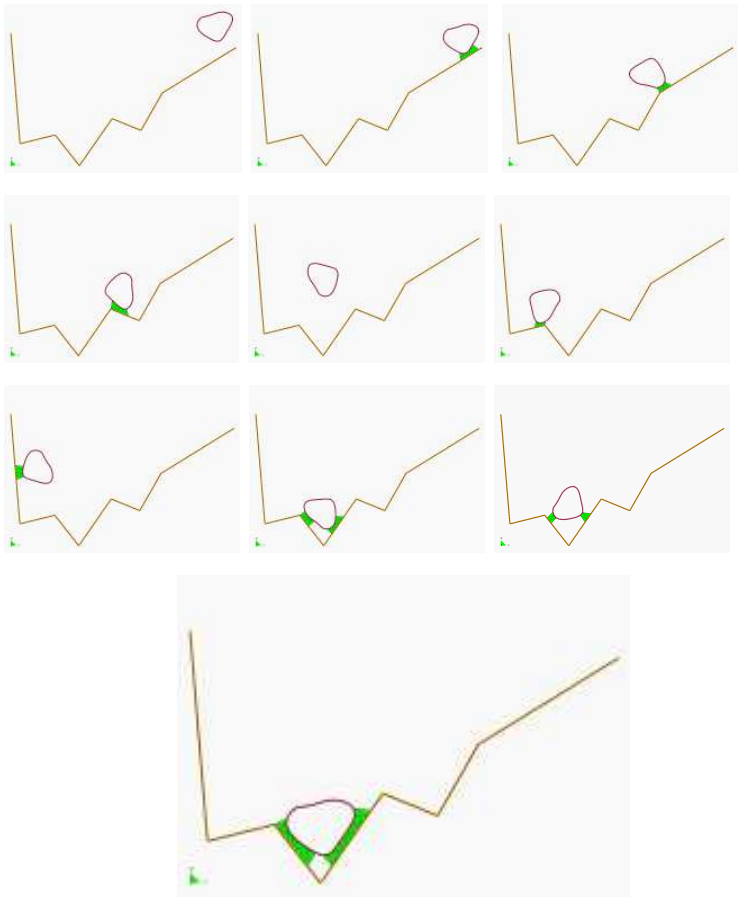


Figure 13: Bumping of a ball within a container. The layer of contact elements is shown at each contact instant

The algorithm for modeling the erosion of soil/rock particles at the fluid bed is the following:

1. Compute at every point of the bed surface the resultant tangential stress τ induced by the fluid motion. In 3D problems $\tau = (\tau_s^2 + \tau_t)^2$ where τ_s and τ_t are the tangential stresses in the plane defined by the normal direction $\mathbf{n}$ at the bed node. The value of τ for 2D problems can be estimated as follows:

$$\tau_t = \mu \gamma_t \tag{34a}$$

with

$$\gamma_t = \frac{1}{2}\frac{\partial v_t}{\partial n} = \frac{v_t^k}{2h_k} \tag{34b}$$

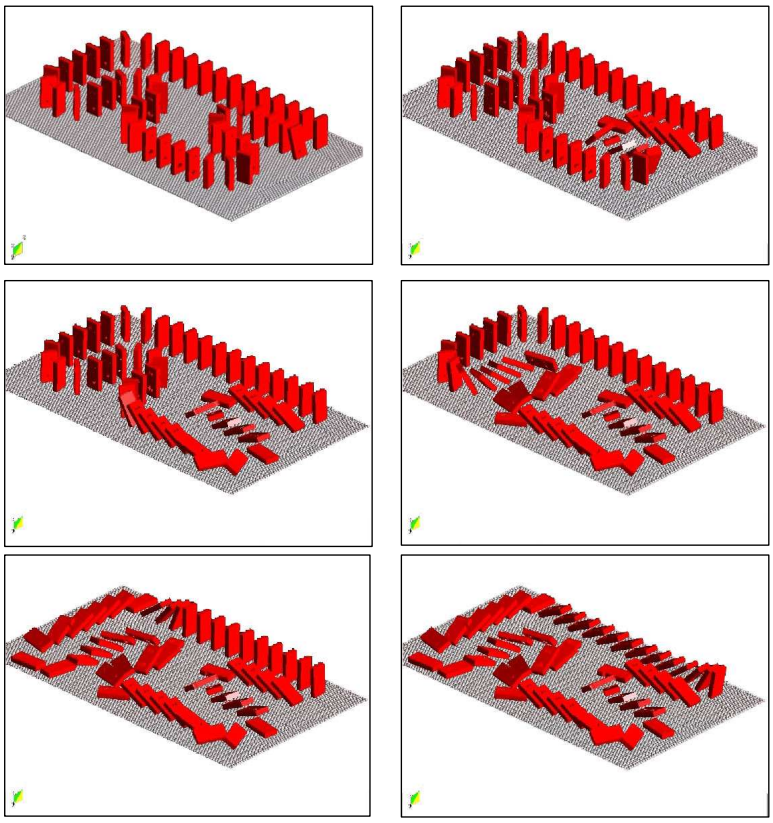


Figure 14: Failure of a domino set. The distance between the domino chips shows the contact element layer

where v_t^k is the modulus of the tangential velocity at the node k and h_k is a prescribed distance along the normal of the bed node k. Typically h_k is of the order of magnitude of the smallest fluid element adjacent to node k (Figure 18).

2. Compute the frictional work originated by the tangential stresses at the bed surface as

$$W_f = \int_\circ^t \tau_t \gamma_t \, dt = \int_\circ^t \frac{\mu}{4} \left(\frac{v_t^k}{h_k} \right)^2 dt \tag{35}$$

Eq.(35) is integrated in time using a simple scheme as

$$^nW_f = {}^{n-1}W_f + \tau_t \gamma_t \, \Delta t \tag{36}$$

3. The onset of erosion at a bed point occurs when ^nW_f exceeds a critical threshold value W_c defined empirically according to the specific properties of the bed material.

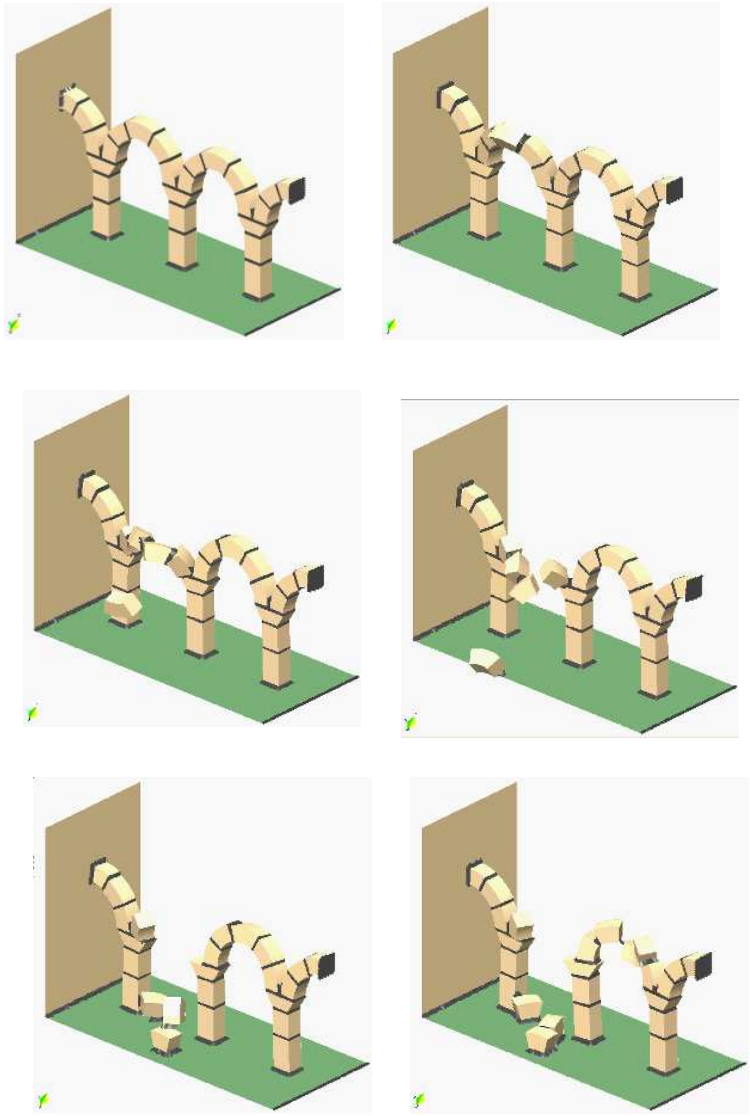


Figure 15: Failure of an arch formed by stone blocks under seismic loading

4. If ${}^{n}W_f > W_c$ at a bed node, then the node is detached from the bed region and it is allowed to move with the fluid flow, i.e. it becomes a fluid node. As a consequence, the mass of the patch of bed elements surrounding the bed node vanishes in the bed domain and it is transferred to the new fluid node. This mass is subsequently transported with the fluid. Conservation of mass of the bed particles within the fluid is guaranteed by changing the density of the new fluid node so that the mass of the suspended sediment traveling with

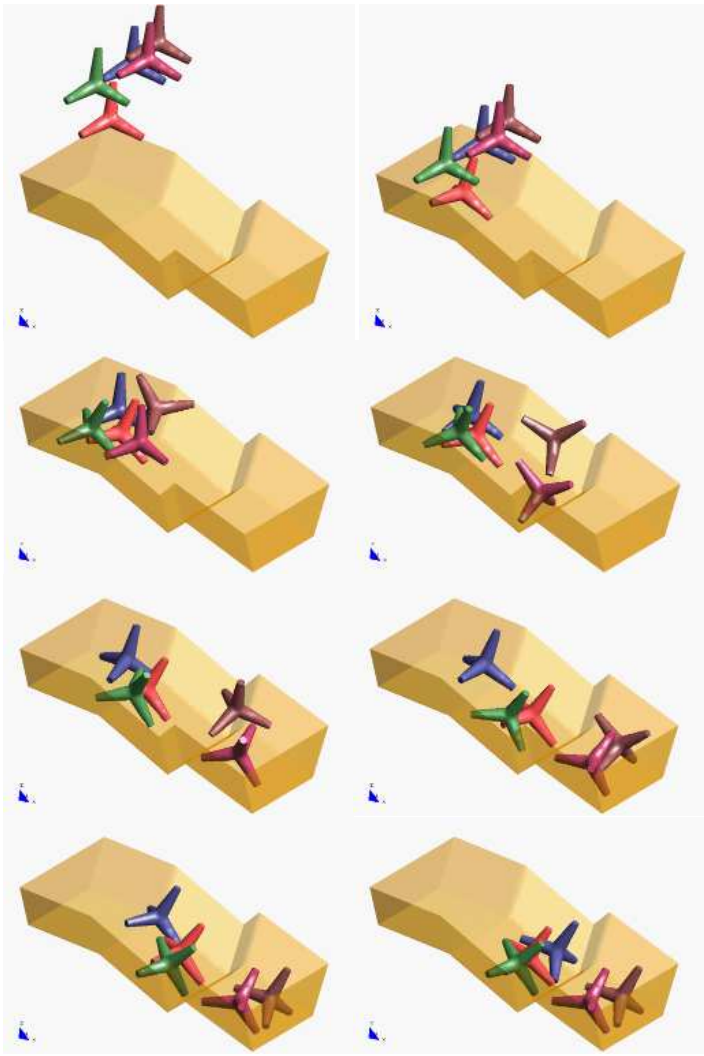


Figure 16: Motion of five tetrapods on an inclined plane

the fluid equals the mass originally assigned to the bed node. Recall that the mass assigned to a node is computed by multiplying the node density by the tributary domain of the node.

5. Sediment deposition can be modeled by an inverse process to that described in the previous step. Hence, a suspended node adjacent to the bed surface with a velocity below a threshold value is assigned to the bed surface. This automatically leads to the generation of new bed elements adjacent to the boundary of the bed region. The original mass of the bed region is recovered

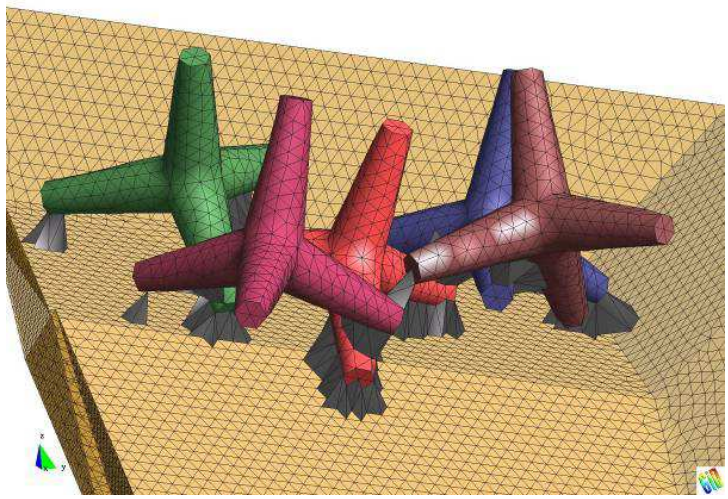


Figure 17: Detail of five tetrapods on an inclined plane. The layer of elements modeling the frictional contact conditions is shown

by adjusting the density of the newly generated bed elements.

Figure 18 shows an schematic view of the bed erosion algorithm proposed.

8 FSI EXAMPLES

The examples chosen show the applicability of the PFEM to solve problems involving large motions of the free surface, fluid-multibody interactions and bed erosion.

8.1 Rigid objects falling into water

The analysis of the motion of submerged or floating objects in water is of great interest in many areas of harbour and coastal engineering and naval architecture among others.

Figure 19 shows the penetration and evolution of a cube and a cylinder 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 20).

8.2 Impact of water streams on rigid structures

Figure 21 shows an example of a wave breaking within a prismatic container including a vertical cylinder. Figure 22 shows the impact of a wave on a vertical column sustained by four pillars. The objective of this example was to model the impact of a water stream on a bridge pier accounting for the foundation effects.

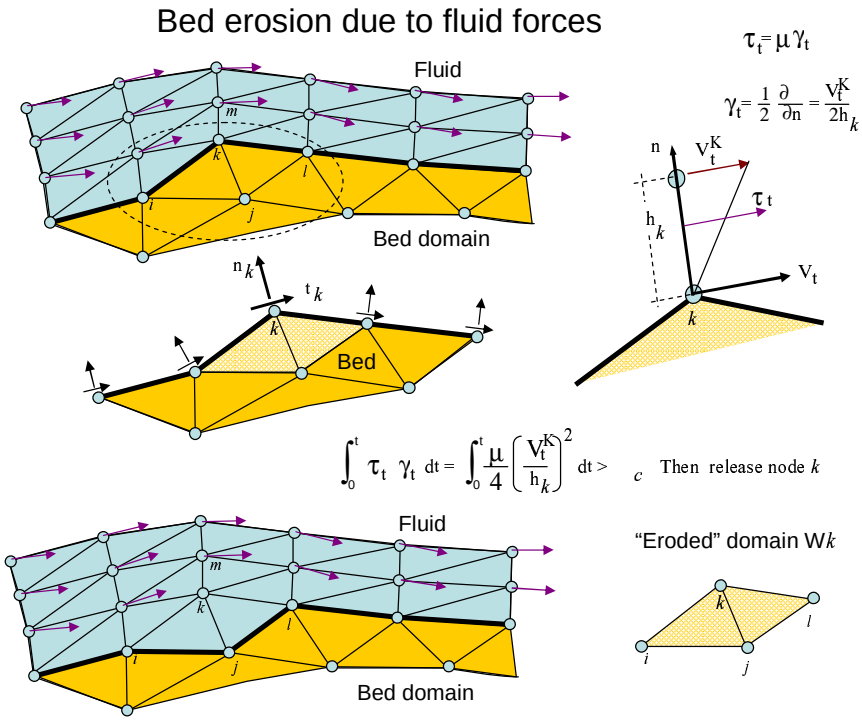


Figure 18: Modeling of bed erosion by dragging of bed material

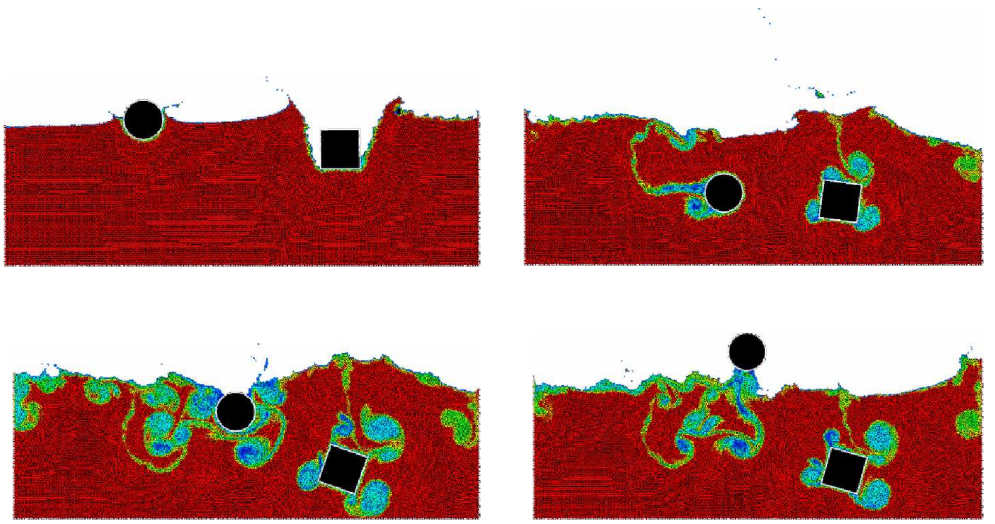


Figure 19: 2D simulation of the penetration and evolution of a cube and a cylinder in a water container. The colours denote the different sizes of the elements at several times.

8.3 Dragging of objects by water streams

Figure 23 shows the effect of a wave impacting on a rigid cube representing a vehicle. This situation is typical in flooding and Tsunami situations. Note the layer of contact elements modeling the frictional contact conditions between the cube and the bottom surface.

8.4 Impact of sea waves on breakwaters and piers

Figure 24 shows the 3D simulation of the impact of a wave generated in an experimental flume on a collection of rigid rocks representing a breakwater. Details of the water-rock interaction are shown in Figure 25.

Figure 26 shows a 3D analysis of a similar problem. Figure 27 shows the 3D simulation of the interaction of a wave with a vertical pier formed by a collection of reinforced concrete cylinders.

The examples shown in Figures 28 and 29 evidence the potential of the PFEM to solve 3D problems involving complex interactions between water and moving solid objects. Figure 28 shows the simulation of the falling of two tetrapods in a water container. Figure 29 shows the motion of a collection of ten tetrapods placed in a slope under an incident wave.

Figure 30 shows a detail of the complex three-dimensional interactions between the water particles and the tetrapods and between the tetrapods themselves, which can be easily modeled with the PFEM.

8.5 Erosion of a 3D earth dam due to an overspill stream

We present finally a simple, schematic, but very illustrative example showing the potential of the PFEM to model bed erosion in free surface flows.

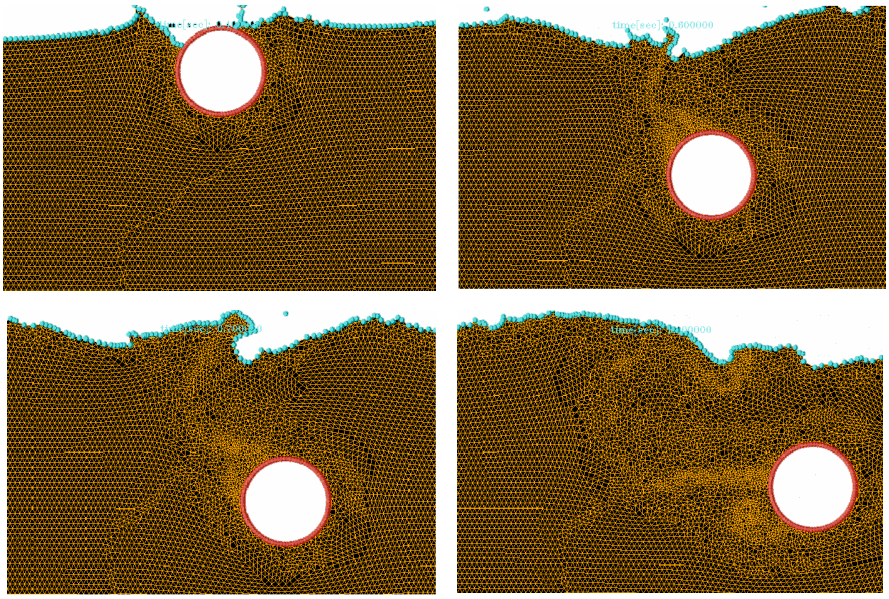


Figure 20: Detail of element sizes during the motion of a rigid cylinder within a water container.

The example represents the erosion of an earth dam under a water stream running over the dam top. A schematic geometry of the dam has been chosen to simplify the computations. Sediment deposition is not considered in the solution. The images of Figure 31 show the progressive erosion of the dam until the whole dam is dragged out by the fluid flow.

Other applications of the PFEM to bed erosion problems can be found in [29].

9 CONCLUSIONS

The particle finite element method (PFEM) is ideal to treat problems involving fluids with free surfaces and submerged or floating structures and bodies within a unified Lagrangian finite element framework. Problems such as fluid-structure interaction, large motion of fluid or solid particles, surface waves, water splashing, separation of water drops, frictional contact situations between fluid-solid and solid-solid interfaces, bed erosion, etc. can be easily solved with the PFEM. The success of the method lies in the accurate and efficient solution of the equations of an incompressible fluid and of solid dynamics using an updated Lagrangian formulation and a stabilized finite element method, allowing the use of low order elements with equal order interpolation for all the variables. Other essential solution ingredients are the efficient regeneration of the finite element mesh using an extended Delaunay tesselation, the identification of the boundary nodes using an Alpha-Shape type technique and the simple algorithm to treat frictional contact conditions at fluid-solid and solid-solid interfaces via mesh generation. The examples presented have

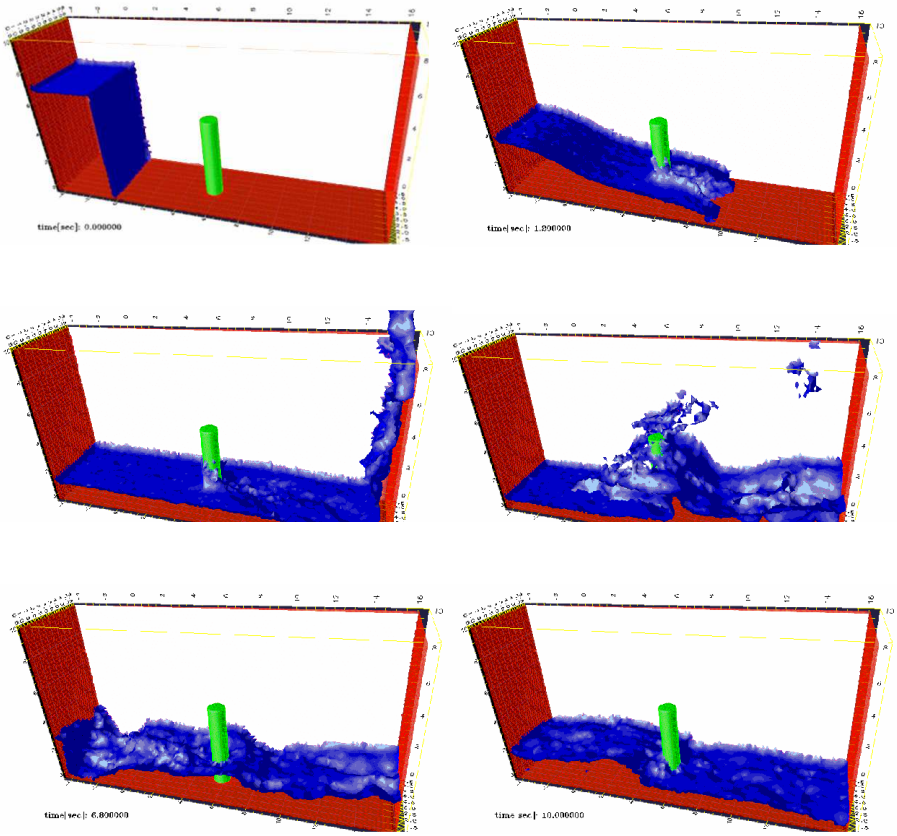


Figure 21: Evolution of a water column within a prismatic container including a vertical cylinder.

shown the great potential of the PFEM for solving a wide class of practical FSI problems in engineering. Examples of validation of the PFEM results with data from experimental tests are reported in [15].

Acknowledgements

Thanks are given to Mrs. M. de Mier for many useful suggestions. This research was partially supported by project SEDUREC of the Consolider Programme of the Ministerio de Educación y Ciencia of Spain.

REFERENCES

[1] J.F. Archard, Contact and rubbing of flat surfaces, J. Appl. Phys. 24(8) (1953) 981–988.

[2] R. Aubry, S.R. Idelsohn, E. Oñate, Particle finite element method in fluid mechanics including thermal convection-diffusion, Computer & Structures 83(17-18) (2005) 1459–1475.

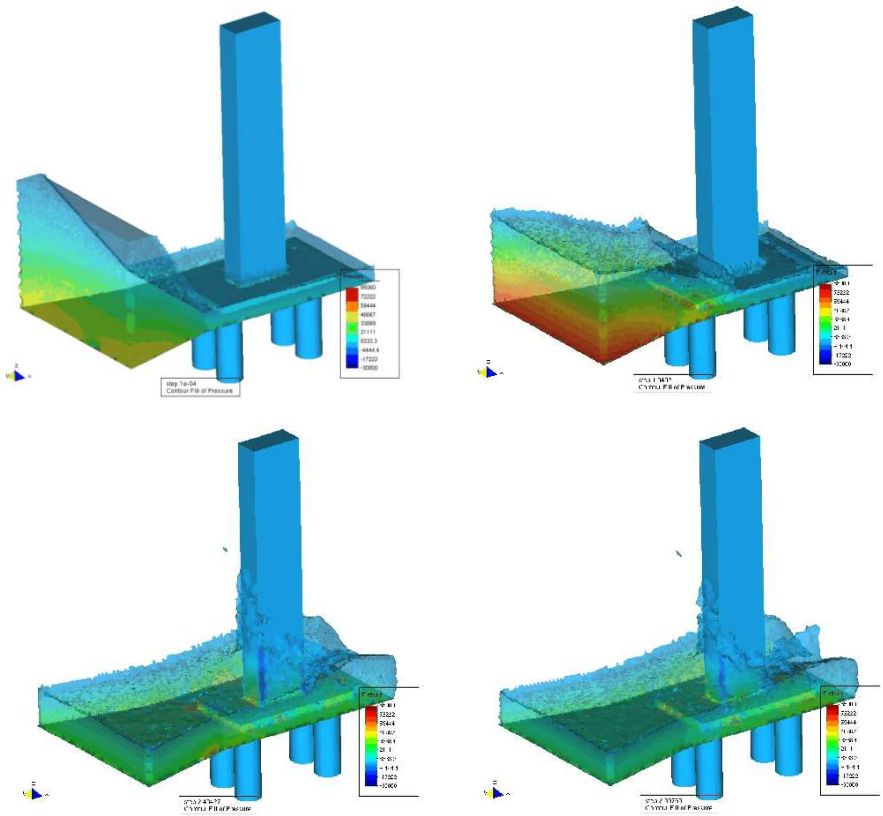


Figure 22: Impact of a wave on a prismatic column on a slab sustained by four pillars.

[3] R. Codina, O.C. Zienkiewicz, CBS versus GLS stabilization of the incompressible Navier-Stokes equations and the role of the time step as stabilization parameter, Communications in Numerical Methods in Engineering (2002) 18(2) (2002) 99–112.

[4] F. Del Pin, S.R. Idelsohn, E. Oñate, R. Aubry, The ALE/Lagrangian particle finite element method: A new approach to computation of free-surface flows and fluid-object interactions. Computers & Fluids 36 (2007) 27–38.

[5] J. Donea, A. Huerta, Finite element method for flow problems, J. Wiley, (2003).

[6] H. Edelsbrunner, E.P. Mucke, Three dimensional alpha shapes, ACM Trans. Graphics 13 (1999) 43–72.

[7] J. García, E. Oñate, An unstructured finite element solver for ship hydrodynamic problems, J. Appl. Mech. 70 (2003) 18–26.

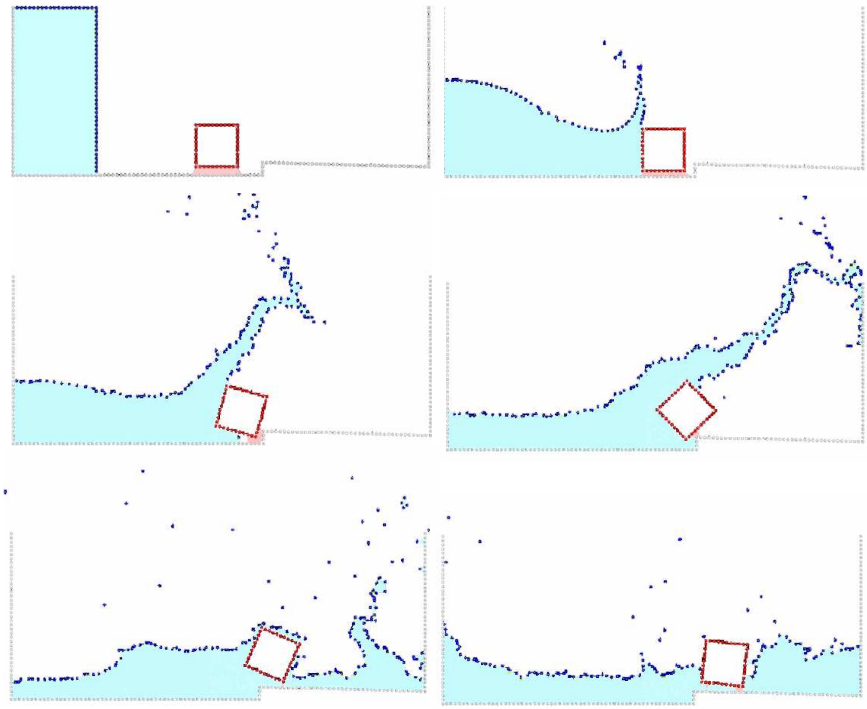


Figure 23: Dragging of a cubic object by a water stream.

[8] S.R. Idelsohn, E. Oñate, F. Del Pin, N. Calvo, Lagrangian formulation: the only way to solve some free-surface fluid mechanics problems, Fith World Congress on Computational Mechanics, H.A. Mang, F.G. Rammerstorfer, J. Eberhardsteiner (eds), July 7–12, Viena, Austria, (2002).

[9] S.R. Idelsohn, E. Oñate, N. Calvo, F. Del Pin, The meshless finite element method, Int. J. Num. Meth. Engng. 58(6) (2003a) 893–912.

[10] S.R. Idelsohn, E. Oñate, F. Del Pin, A lagrangian meshless finite element method applied to fluid-structure interaction problems, Computer and Structures 81 (2003b) 655–671.

[11] S.R. Idelsohn, N. Calvo, E. Oñate, Polyhedrization of an arbitrary point set, Comput. Method Appl. Mech. Engng. 192(22-24) (2003c) 2649–2668.

[12] S.R. Idelsohn, E. Oñate, F. Del 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 (2004) 964-989.

[13] S.R. Idelsohn, E. Oñate, F. Del Pin, N. Calvo, Fluid-structure interaction using the particle finite element method, Comput. Meth. Appl. Mech. Engng. 195 (2006) 2100-2113.

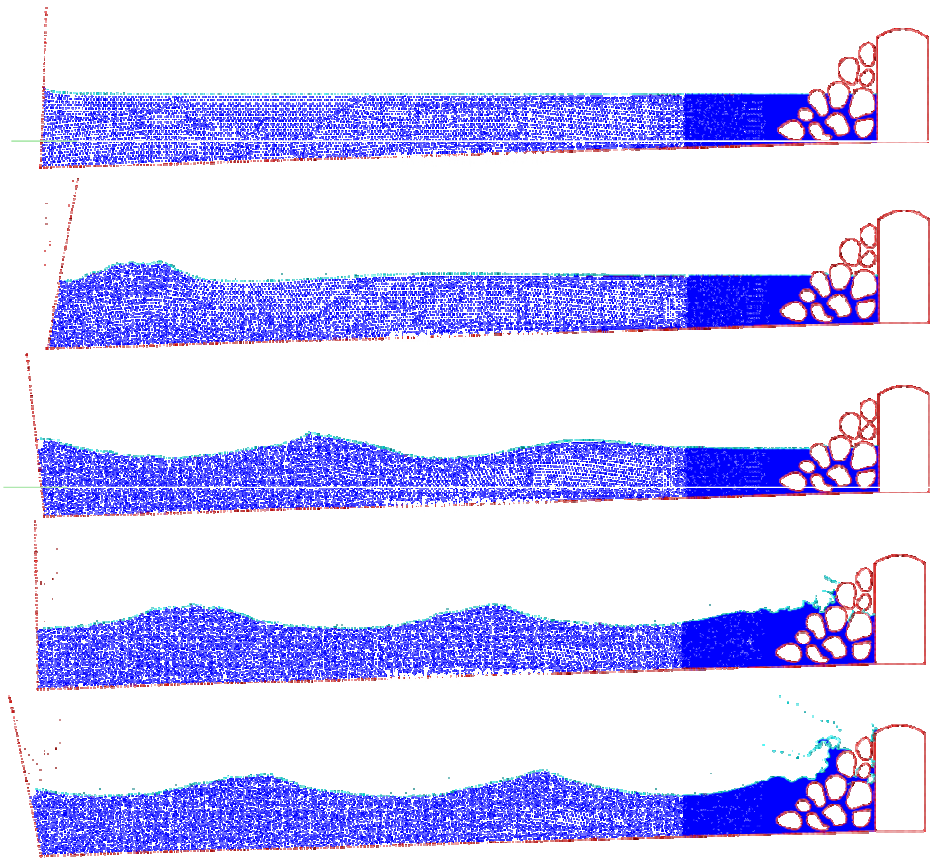


Figure 24: Generation and impact of a wave on a collection of rocks in a breakwater.

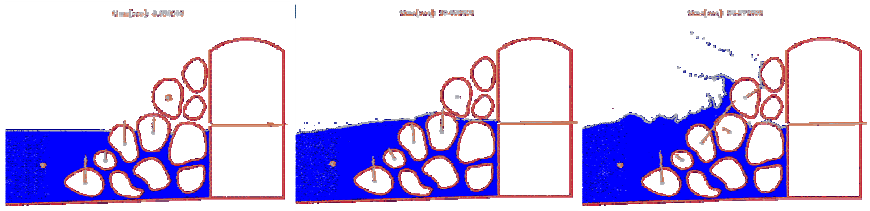


Figure 25: Detail of the impact of a wave on a breakwater. The arrows indicate the water force on the rocks at different instants.

[14] A. Kovacs, G. Parker, A new vectorial bedload formulation and its application to the time evolution of straight river channels, J. Fluid Mech. 267 (1994) 153–183.

[15] L. Larese, R. Rossi, E. Oñate, S.R. Idelsohn, Validation of the Particle Finite Element Method (PFEM) for simulation of free surface flows, submitted to Engineering Computations, March (2007).

[16] R. Ohayon, Fluid-structure interaction problem, in: E. Stein, R. de Borst, T.J.R. Hugues (Eds.), Enciclopedia of Computatinal Mechanics, Vol. 2, (J. Wiley, 2004) 683–694.

[17] E. Oñate, Derivation of stabilized equations for advective-diffusive transport and fluid flow problems, Comput. Meth. Appl. Mech. Engng. 151 (1998) 233–267.

[18] E. Oñate, A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation, Comp. Meth. Appl. Mech. Engng. 182(1–2) (2000) 355–370.

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

[20] E. Oñate, S.R. Idelsohn, A mesh free finite point method for advective-diffusive transport and fluid flow problems, Computational Mechanics 21 (1998) 283–292.

[21] E. Oñate, J. García, A finite element method for fluid-structure interaction with surface waves using a finite calculus formulation, Comput. Meth. Appl. Mech. Engrg. 191 (2001) 635–660.

[22] E. Oñate, J. Rojek, Combination of discrete element and finite element method for dynamic analysis of geomechanic problems, Comput. Meth. Appl. Mech. Engrg. 193 (2004) 3087–3128.

[23] E. Oñate, C. Sacco, S.R. Idelsohn, A finite point method for incompressible flow problems, Comput. Visual. in Science 2 (2000) 67–75.

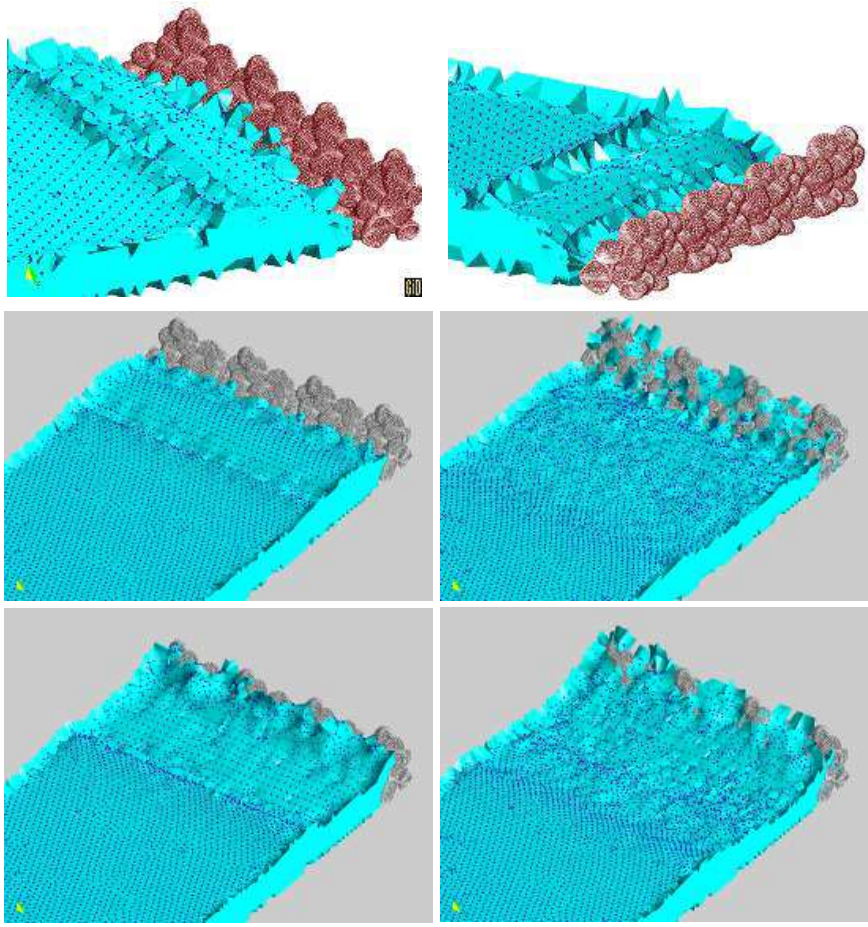


Figure 26: 3D simulation of the impact of a wave on a collection of rocks in a breakwater.

[24] E. Oñate, S.R. Idelsohn, F. Del Pin, Lagrangian formulation for incompressible fluids using finite calculus and the finite element method, Numerical Methods for Scientific Computing Variational Problems and Applications, Y Kuznetsov, P Neittanmaki, O Pironneau (Eds.), CIMNE, Barcelona (2003).

[25] E. Oñate, J. García, S.R. Idelsohn, Ship hydrodynamics, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds), Encyclopedia of Computational Mechanics, Vol 3, (J. Wiley 2004a) 579–610.

[26] E. Oñate, S.R. Idelsohn, F. Del Pin, R. Aubry, The particle finite element method. An overview, Int. J. Comput. Methods 1(2) (2004b) 267-307.

[27] E. Oñate, A. Valls, J. García, FIC/FEM formulation with matrix stabilizing terms for incompressible flows at low and high Reynold's numbers, Computa-

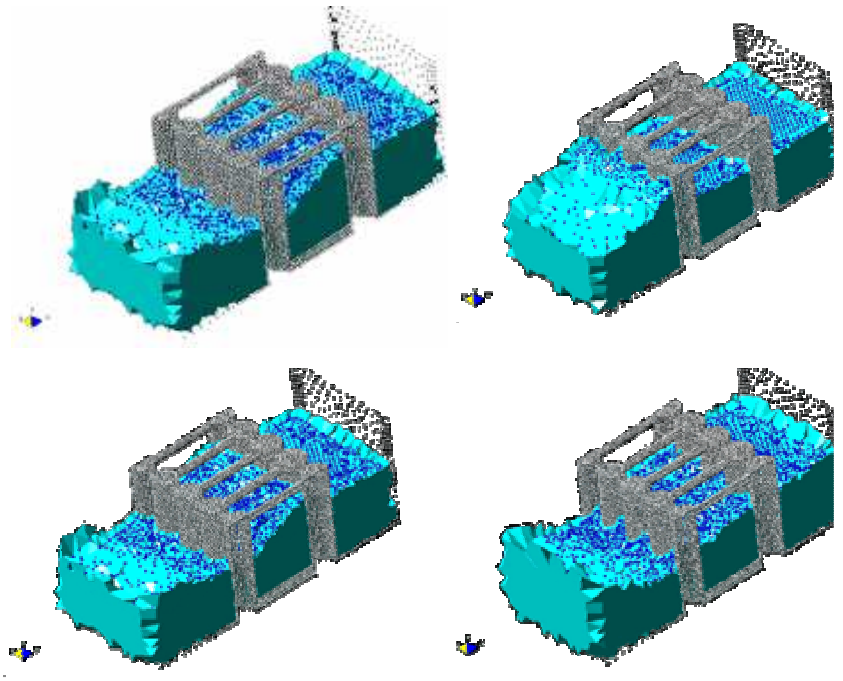


Figure 27: Interaction of a wave with a vertical pier formed by reinforced concrete cylinders.

tional Mechanics 38 (4-5) (2006a) 440-455.

[28] E. Oñate, J. García, S.R. Idelsohn, F. Del Pin, FIC formulations for finite element analysis of incompressible flows. Eulerian, ALE and Lagrangian approaches, Comput. Meth. Appl. Mech. Engng. 195 (23-24) (2006b) 3001-3037.

[29] E. Oñate, M.A. Celigueta, S.R. Idelsohn, Modeling bed erosion in free surface flows by the Particle Finite Element Method, Acta Geotechnia 1 (4) (2006c) 237-252.

[30] E. Oñate, S.R. Idelsohn, R. Rossi, Enhanced FIC-FEM formulation for incompressible flows, Research Report, CIMNE Barcelona, March 2007.

[31] D.B. Parker, T.G. Michel, J.L. Smith, Compaction and water velocity effects on soil erosion in shallow flow, J. Irrigation and Drainage Engineering 121 (1995) 170–178.

[32] T.E. Tezduyar, Finite element method for fluid dynamics with moving boundaries and interface, in: E. Stein, R. de Borst, T.J.R. Hugues (Eds.), Enciclopedia of Computatinal Mechanics, Vol. 3, (J. Wiley, 2004) 545–578.

[33] C.F. Wan, R. Fell, Investigation of erosion of soils in embankment dams, J. Geotechnical and Geoenvironmental Engineering 130 (2004) 373–380.

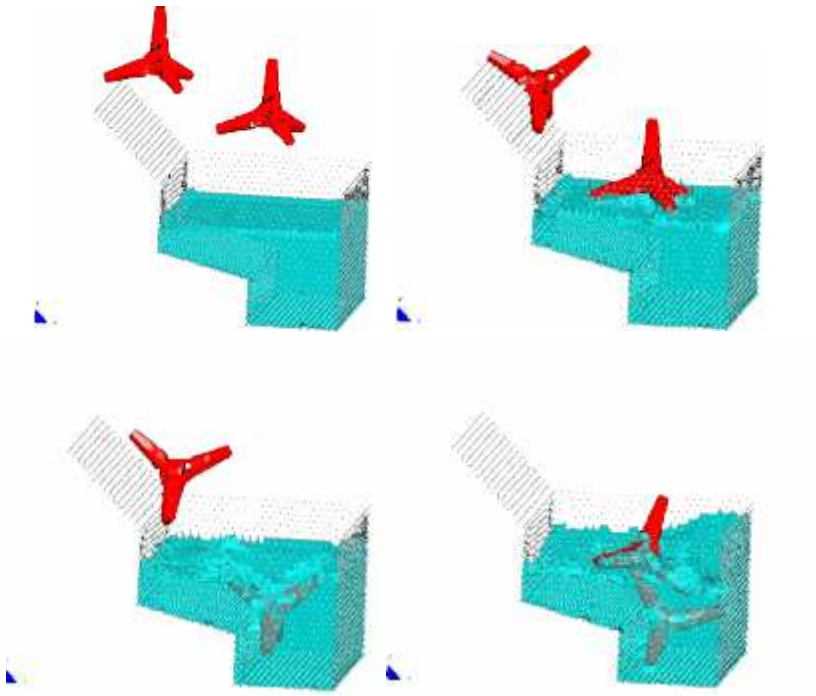


Figure 28: Motion of two tetrapods falling in a water container.

[34] O.C. Zienkiewicz, R.L. Taylor, P. Nithiarasu, The finite element method for fluid dynamics, Elsevier, (2006).

[35] O.C. Zienkiewicz, R.L. Taylor, The finite element method for solid and structural mechanics, Elsevier, (2005).

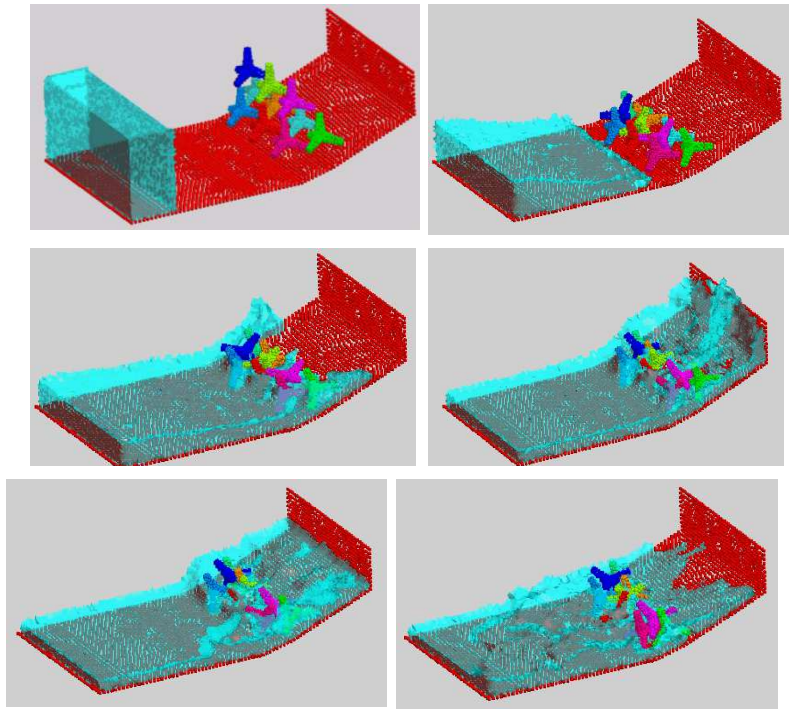


Figure 29: Motion of ten tetrapods on a slope under an incident wave.

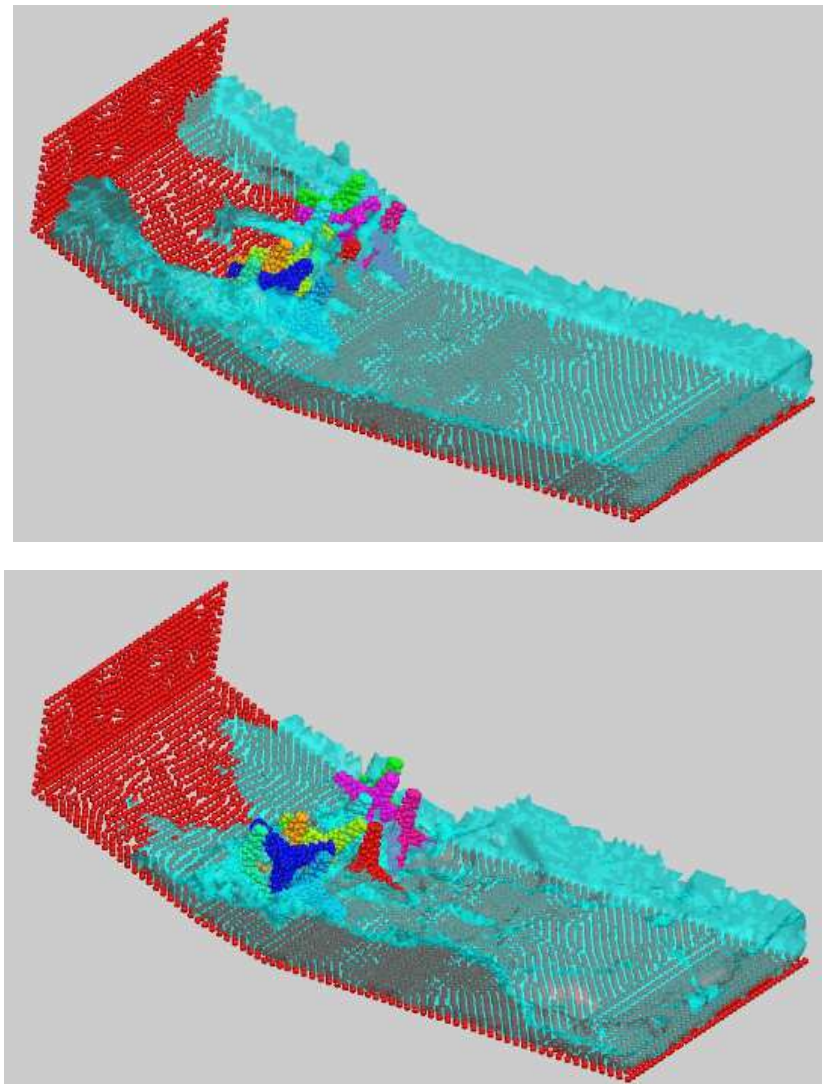


Figure 30: Detail of the motion of ten tetrapods on a slope under an incident wave. The figure shows the complex interactions between the water particles and the tetrapods.

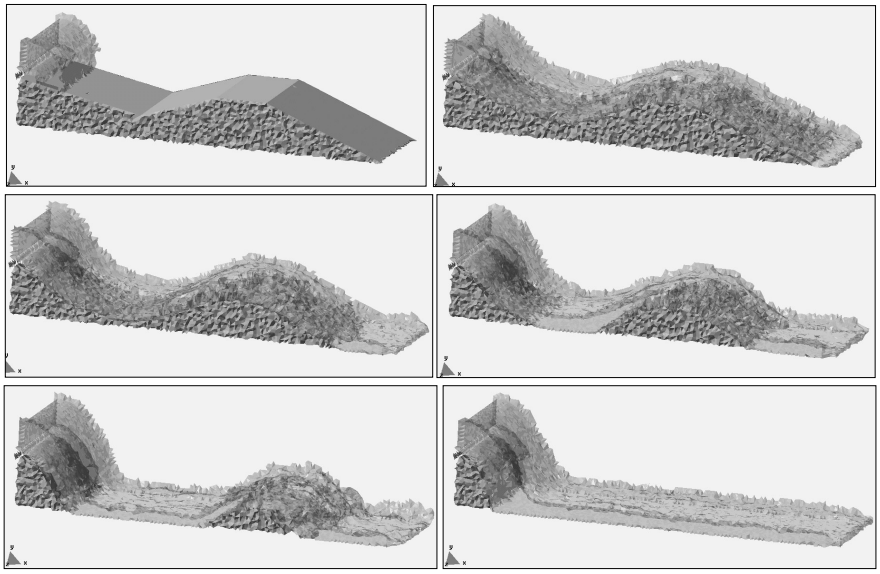


Figure 31: Erosion of a 3D earth dam due to an overspill stream.

Paper 3

This paper, published in 2008, consists on a validation of the fluid kernel of *PFlow*. The validation is based on experimental test of sloshing water in moving tanks. Special emphasis is put on the validation of the pressure values. The *PFlow* code was used for all the computations.

PRESSURE MEASUREMENT IN 2D SLOSHING SIMULATIONS WITH SPH

L. Delorme[1], M.A. Celigueta[2], M.A. Oñate[2], A. Souto-Iglesias[1]

[1]Model Basin Research Goup (CEHINAV), Naval Architecture Dpt (ETSIN), Technical University of Madrid (UPM). Avda Arco de la Victoria s/n, 28040 Madrid, SPAIN.
[2]International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Gran Capitán s/n, 08034 Barcelona, Spain

Abstract

Sloshing for low filling level resonant pitch motion is studied experimentally and numerically using SPH. Special attention is paid to the pressure fields on the tanks. Comparisons are made with experimental data and with Particle Finite Element Method (PFEM) calculations.

1 Introduction

Extensive experimental programs aimed at a better comprehension of the sloshing loads have been conducted for the last 30 years (refs. [1] and [2]). The reason for this interest lies mainly in the influence of these loads in the design and operation of LNG tankers. CFD technologies are helping in the understanding of these loads, usually tracing the free surface evolution by VOF techniques [4], but to date, it is difficult for these techniques to model fragmentation and compressibility effects, which are crucial during the impact. Meshless methods like Smoothed Particle Hydrodynamics (SPH) [5] can be especially appropriate when modelling the highly non linear free surface flows with impact and fragmentation that appear in violent sloshing flows. This short paper focuses on the assessment of these local loads, following a previous one from the same group [7], in which global loads were successfully reproduced. SPH results are compared with experiments and with monophasic Particle Finite Element Method (PFEM) results [3] for the same case.

2 Experimental Results

The case studied is a 2D longitudinal section of a tank that belongs to a 138 000 m^3 LNG membrane tanker in operation, at scale 1:50. Model dimensions are 90 x 58 x 5 cm and water depth is 9.3 cm (depth ratio ≈ 0.1). The tank is excited with a sinusoidal type motion (θ_{max} = 4º) whose period matches the first sloshing period T_0 = 1.9s.

The flow is composed by a main wave, travelling from one side of the tank to the other, forming a plunging-type breaker at half way that impacts on the structure. The dissipation due to breaking is high and the experiments demonstrate that the water motion in the tank is qualitatively periodic, including the breaking process.

Figure 1 shows the angle and pressure time series. In the following, the angle is made non-dimensional with θ_{max}, the time with T_0 and the pressure with the hydrostatic one. The pressure sensor is located at the unperturbed free-surface height. The pressure register is qualitatively repetitive at each cycle. However, the maximum value of the pressure is not equal in each cycle. These peaks result from the impact of the wave on the tank, presenting a random behavior. This can be explained by the very short duration of the impact and the extreme sensitivity of the impact pressure to the shape of the wave just before impact. Other physical parameters, such as the compressibility of the air and water mixture as well as the ullage pressure, have also a very important effect (refs. [1] and [2]) and are very difficult to model

A zoom of the time series over one impact event is shown in figure 2. Frames F1 to F6 have been located on the pressure curve representing the most interesting instants regarding the pressure history. Pressure register and videos demonstrate this process to be qualitatively repetitive

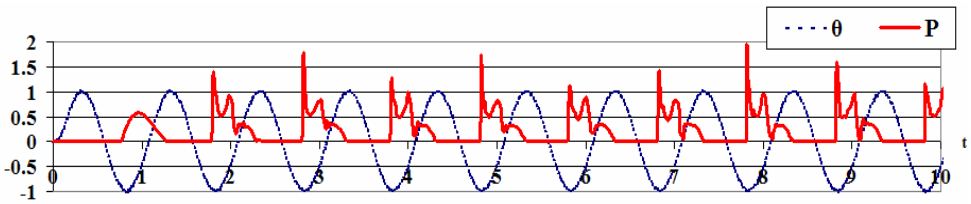


Figure 1: Experimental angle and pressure versus time (non-dimensional values)

3 Simulations

A standard SPH formulation has been used for the simulation [5]. Free slip boundary conditions have been imposed with boundary particles [5]. In order to calculate the pressure at the sensor position, the forces exerted by all the boundary particles within a distance *h* to the center of the sensor have been averaged, *h* being the smoothing length. The standard viscosity term is used with α=0.02. Numerical integration has been performed with a leap-frog scheme.

Simulations have been performed with 5 different resolutions: 3043, 4928, 8970, 12924 and 20205 fluid particles. Figure 3 presents the pressure time series for three resolutions. The graph shows that the trends in the experimental curve are qualitatively reproduced with SPH. However, numerical instabilities appear that need further study.

PFEM results for the same case are presented in figure 4. The shape of the pressure curve is qualitatively reproduced too. PFEM results present numerical instabilities of greater amplitude and frequency. Pressure maxima at the impact are greater and this can be explained by the incompressibility of the fluid, imposed when using PFEM.

The compressibility of the fluid plays an important role in the impact phenomena [2]. This has been investigated performing SPH simulations with different numerical sound speeds. Sound speed is typically chosen such that the Mach number is 0.01. SPH simulations have been performed using sound speeds 10, 20, 30 and 40 m/s but the variations found in the values of the pressure peaks were not significant.

It has been demonstrated [6] that the impulse given by a wave is a more useful information than the pressure in assessing its impact. The pressure impulse (integral of the pressure through the impact) can be calculated from the pressure time series and compared with the experiments (figure 5). After the third cycle, the variations of the impulse are small and it can be noticed that both SPH and PFEM overestimate the experimental value. The biphasic nature of the impact could explain the lower experimental values but further investigation has to be done.

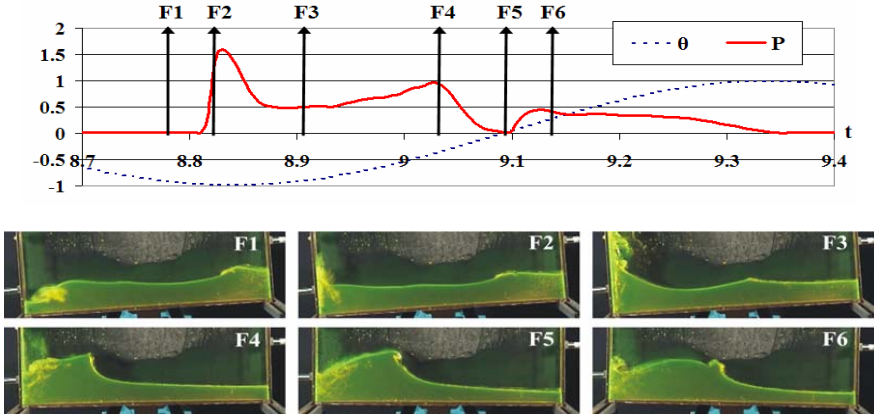


Figure 2: Experimental register above one impact event with the corresponding frame

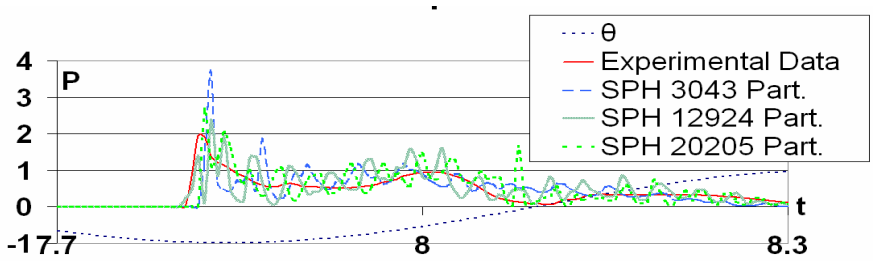


Figure 3: Non-dimensional pressure over one impact event. SPH results

The global dynamics of the flow, including breaking waves, is well reproduced by both methods. Figure 6, for instance, presents the frames of figure 2 obtained with SPH, showing good agreement, even after more than eight cycles.

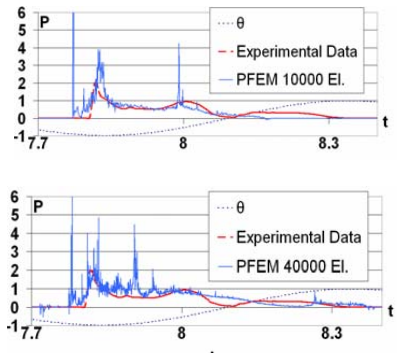


Figure 4: Non-dimensional pressure over one impact event. PFEM results.

4 Conclusions

Numerical computations of long impact pressure sequences for a 2D low filling sloshing case have been performed both with SPH and PFEM codes. Good agreement has been found in the general dynamics but unphysical oscillations in the time series of the pressure appear for both methods. Pressure impulse has been compared and reasonable but overestimated values have been found, regardless of the resolution and of the SPH numerical sound speed. So far, the influence of the gas phase on the pressure history has not been assessed with enough quality to discriminate the origin of the numerical errors. Further work has yet to be done.

Acknowledgements

This work has been partially funded by the program PROFIT 2007 of the Spanish Ministerio de Educación y Ciencia through the project STRUCT-LNG (file number CIT-370300-2007-12) leaded by the Technical University of Madrid(UPM).

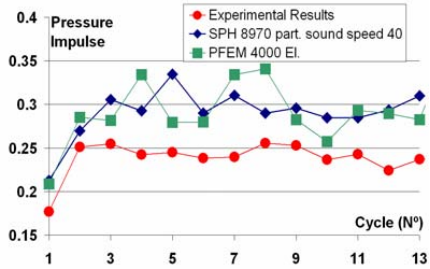


Figure 5: Pressure impulse.

References

[1] Berg, A., Scaling laws and statistical distributions of impact pressures in liquid sloshing, *Det Norske Veritas DNV*, Report no. 87-2008, 1987.

[2] Bass, R.L., Bowles, E.B., Trudell, R.W., Navickas, J., Peck, J.C., Endo, N. and Pots. B.F.M., Modeling criteria for scaled LNG sloshing experiments, *Trans. ASME*, 107, 272—280, 1985.

[3] Idelsohn,S., del Pin, F., Onate, E., Aubry, R., The ALE/Lagrangian Particle Finite Element Method: A new approach to computation of free-surface flows and fluid-object interactions. *Comput. Fluids*, 36, 1, 27-38, 2007.

[4] Kleefsman, K.M.T., Fekken, G., Veldman, A.E.P., Iwanowski, B., Buchner, B., A Volume-of-Fluid based simulation method for wave impact problems. *J.Comp.Phys*., 206, 1, 363-393, 2005.

[5] Monaghan, J.J., Smoothed Particle Hydrodynamics, *Rep. Prog. Phys.*, 37, 1703—1759, 2005.

[6] Peregrine, D.H., Water-wave impact on walls. *Ann. Rev. Fl. Mech*, 37, 2003.

[7] Souto-Iglesias, A., Delorme, L., Abril-Pérez, S. And Pérez-Rojas, L., Liquid moment amplitude assesment in sloshing type problems with SPH, *Ocean Eng.*, 33, 11—12, 2006.

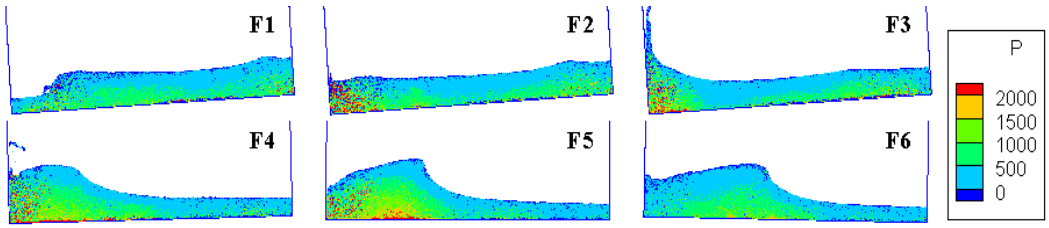


Figure 6: SPH simulation with 20205 fluid particles. F1 to F6 refer to figure 2

Paper 4

This paper, published in 2009, consists on another validation of the fluid kernel of *PFlow*. The validation is based on experimental test of water waves in a flume. In this case, the emphasis is put on the tracking of the free surface. The *PFlow* code was used for all the computations.

Journal of Coastal Research	SI 56	1025 - 1029	ICS2009 (Proceedings)	Portugal	ISSN 0749-0258

Nonlinear Regular Wave Generation in Numerical and Physical Flumes

T.C.A. Oliveira†, F.X. Gironella†, A. Sanchez-Arcilla†, J.P. Sierra† and M.A. Celigueta‡

† Maritime Engineering Laboratory
Technical University of Catalonia
08034 Barcelona, Spain
tiago.oliveira@upc.edu

‡ International Center for Numerical Methods in Engineering
Technical University of Catalonia
08034 Barcelona, Spain
maceli@cimne.upc.edu

ABSTRACT

OLIVEIRA, T.C.A., GIRONELLA, F.X., SANCHEZ-ARCILLA, A., SIERRA, J.P. and CELIGUETA, M.A., 2009. Nonlinear regular wave generation in numerical and physical flumes. Journal of Coastal Research, SI 56 (Proceedings of the 10th International Coastal Symposium), 1025 – 1029. Lisbon, Portugal, ISSN 0749-0258.

The generation of nonlinear regular waves in a numerical wave flume using first-order wavemaker theory is discussed comparing numerical results with free surface data from large scale physical tests (CIEM wave flume) and Stokes wave theories. A general formulation for the analysis of fluid-structure interaction problems is employed to simulate the numerical wave flume using the Particle Finite Element Method (PFEM). This method uses a Lagrangian description to model the motion of particles in both the fluid and the structure domains. With this work we can conclude that PFEM formulations simulate the generation of naturally-occurring nonlinear waves with different paddles types, for varied wave conditions and at different scales. Like in physical flumes if we use first-order wavemaker theory in numerical flumes unwanted nonlinearities can be found for some wave conditions.

ADITIONAL INDEX WORDS: *Particle Finite Element Method, first-order wavemaker theory, unwanted nonlinearities*

INTRODUCTION

Physical wave flumes have been widely applied in laboratory studies on hydraulic and stability behavior of coastal structures, beach profile evolution and other related costal phenomena involving waves. The wave generation is one of most important tasks in this kind of laboratory studies.

The most common way to generate waves in physical flumes is through the movement of a paddle that is generally located at one of the ends of the flume. Of the several types of paddles used, we can identify as the most frequent the flap, piston and wedge types. They differ among themselves by the kind of movement executed and consequently, the imposed water boundary condition and the necessary mechanisms to control their movements.

An analytical solution for waves generated by piston and flap wavemakers based on linear wave theory was derived by HAVELOCK (1929). The first order wavemaker theory for a piston was experimentally verified by URSELL *et al.* (1960) and FLICK and GUZA (1980). HUDSPETH *et al*. (1981) did an experimental verification for the flap first order wavemaker theory. Due to the difference in the type of movement, the velocity field in the area near the paddle changes depending on the type of paddle used. If we compare the form of the velocity profiles generated by the three most common paddles near the wavemaker with the profiles according to linear theory, we can conclude that each paddle reproduces different conditions. Thus, for a flap we have profiles that are similar to deep water wave conditions, for piston shallow water wave conditions (DEAN AND DALRYMPLE, 1992) and for wedge intermediate water wave conditions (GIRONELLA, 2004). The maximum wave heights generated by a paddle depend on the wave period, the water depth in front of the paddle, and the power of the actuator. The firsts two are related with wave breaking and the third with the maximum paddle stroke and velocity allowed (GIRONELLA, 2004).

In the last years numerical waves flumes begun to be considered as a possible tool to support the design of vertical breackwaters (OUMERACI *et al.,* 2001), the design of low crested structures (HAWKINS *et al.,* 2007) and the overtopping calculation (Overtopping Manual, 2007).

The numerical wave flumes presented in the scientific literature can be grouped in several ways. One simple way is to divide them into two groups, one based on Non Linear Shallow Water (NLSW) equations and another one based on the Navier Stokes equations.

Examples of numerical waves flumes based on NLSW can be found in VAN GENT (1994), DODD (1998) and HU *et al.* (2000).

Numerical wave flumes based on the Navier-Stokes equations are in most cases controlled by two techniques: i) Volume of Fluid (VOF) described the first time by HIRT and NICHOLS (1981) and ii) Smooth Particle Hydrodynamics (SPH) technique developed at the end of the 70s in the astrophysics community by GINGOLD and MONAGHAN (1977).

LEMOS (1990) developed a VOF numerical model for the study of the movement of two-dimensional waves. VAN GENT *et al.* (1994) presented a model that can simulate plunging waves breaking into porous structures using the VOF technique for solving (2D-V) Navier-Stokes equations for incompressible fluids. LIN and LIU (1998) described the development of the COBRA numerical model to study the evolution of groups of waves, shoaling and breaking in Swash zone. LARA *et al.* (2006) show the ability of the COBRA model to simulate the interaction of irregular waves with permeable slope structures.

The SPH application to Coastal Engineering began at the end of the 90's (MONAGHAN AND KOS, 1999). DALRYMPLE and ROGERS (2006) studied the plunging wave type breaking using a model

based on SPH method. SHAO *et al.* (2006) presented an incompressible SPH model to investigate overtopping in coastal structures.

The Moving Particle Semi-implicit (MPS) method proposed by KOSHIZUKA *et al.* (1995) and the Particle Finite Element Method (PFME) revised in OÑATE *et al.* (2004) are other two less common methods based on Navier Stokes equations and used as numerical wave flumes (KOSHIZUKA *et al.* 1998 and OLIVEIRA *et al.* 2007).

In numerical wave flumes based on non fixed mesh methods as SPH, MPS and PFEM the generation of waves by means of different wavemaker types is possible. In these cases the wavemakers can be simulated by means of solid bodies located at one end of the flume and moving according to the transfer functions, the same ones used in physical flumes to determine the paddle movement. The selection of the paddle type could be based on the wave condition required and contrary to physical flumes the channel dimensions, the stroke and actuator velocity is not a limitation in the maximum generated wave height.

Within this context, the main aim of this paper is to study and compare the nonlinear regular wave generation in a numerical wave flume based on the PFEM formulation and using the same first order wavemaker theory that is used in physical flumes to generate waves.

PARTICLE FINITE ELEMENT METHOD

The PFEM is a well Know method in literature (OÑATE *et al.,* 2004). However, some important key features of the PFEM are presented in this paper.

The PFEM is a particular class of Lagrangian flow formulation developed at the International Center for Numerical Methods in Engineering (CIMNE) in Barcelona to solve free surface flow problems involving large motions of the free surface, as well as the interaction with rigid bodies.

The PFEM treats the mesh nodes in the fluid and solid domains as particles which can freely move and even separate from the main fluid domain representing, for instance, the effect of water drops or melted zones. The finite element method (FEM) is used to solve the continuum equations in both domains. Hence a mesh discretizing these domains must be generated in order to solve the governing equations for the fluid, in the standard FEM fashion. To do this, the nodes discretizing the fluid and solid domains are treated as material particles whose motion is tracked during the transient solution.

In the PFEM the motion of the individual particles are followed using a Lagrangian description and, consequently, nodes in a finite element mesh can be viewed as moving *particles*. Hence, the motion of the mesh discretizing the total domain (including both the fluid and solid parts) is followed during the transient solution.

An obvious advantage of the Lagrangian formulation used in the PFEM is that the convective terms disappear from the fluid and energy equations. The difficulty is however transferred to the problem of adequately (and efficiently) moving the mesh nodes. Indeed for large mesh motions remeshing may be a frequent necessity along the time solution. An innovative mesh regeneration procedure is used, based on the well known Delaunay Tessellation (IDELSOHN *et al.,* 2003).

It must be noted that the information in the PFEM is typically nodal-based, i.e. the element mesh is mainly used to obtain the values of the state variables (i.e. velocities, pressure, viscosity, etc.) at the nodes. A difficulty arises in the identification of the boundary of the domain from a given collection of nodes. Indeed the *boundary* can include the free surface in the fluid and the individual particles moving outside the fluid domain. The Alpha Shape technique is used to identify the boundary nodes (IDELSOHN *et al.,* 2003).

In the PFEM both the fluid and the solid domains are modelled using an updated Lagrangian description of the motion. That is, all variables in the fluid and solid domains are assumed to be known in the current configuration at time t. The new set of variables in both domains is sought for in the next or updated configuration at the next time step.

NONLINEAR WAVE GENERATION

In the following two points the validation of naturally-occurring nonlinearities in wave generation is made comparing free surface data from a numerical flume with experimental data and with theoretical free surface profiles given by Stokes Wave theories.

Then, scale effects in wave generation are analyzed comparing data from three different scale numerical flumes.

Finally, an example of unwanted nonlinearities generated in a numerical flume using first-order wavemaker theory is presented.

Comparison with experimental data

The experiments were carried out at the Maritime, Experimental and Research Flume (CIEM, Canal d'Investigació I Experimentació Marítima) of the Maritime Engineering Laboratory (LIM, Laboratori d'Enginyeria Marítima) of the Technical University of Catalonia (UPC, Universitat Politecnica de Catalunya). The flume is 100m long, 3m wide and 5m deep. Due to its dimensions the CIEM flume is an excellent tool for scaled tests and analyses close to reality allowing a reduction of scale effects inherent to all scaled experiments. Controlled wave generation is achieved by a wedge-type wave paddle, particularly suited for intermediate depth waves.

The experimental set-up is presented in Figure 1. A rigid bottom was used across the flume and a dissipative rock beach was constructed at the right end side. A constant 2.62m water depth zone in front of the paddle has been separated from a second 1.50m constant water depth zone by a 1V:9.3H follow by a 1V:36.2H slopes. Six resistence wave gauges (WG) were used to measure the free surface elevation. Their positions are represented in Figure 1. A positional sensor was used to measure the paddle movement. Twenty two regular wave cases were tested for different wave heights ($0.1m<H<0.6m$) and wave periods ($0.5s<T<4.0s$).

The firsts 100s of experiments were simulated using PFEM for the three cases denoted in this work as case 1 (H=0.182m, T=3.0s), case 2 (H=0.351, T=4.0s) and case 3 (H=0.546m, T=3.0s). The maximum time step used during the simulations was 0.01s and the nodes distance 0.10m corresponding to 14045 initial nodes.

The free surface for case 3 at 58s of simulation is presented at Figure1. The waves were generated reproducing the paddle movement recorded by the positional sensor used during the physical experiments.

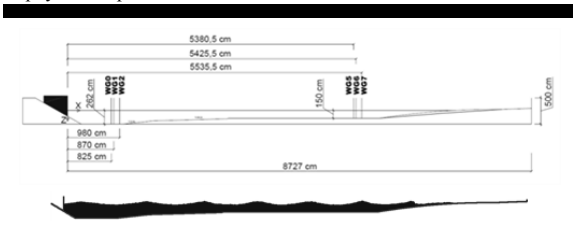


Figure 1. Experimental set-up and numerical flume for H=0.546m T=3.0s (case 3) after 58s of simulation

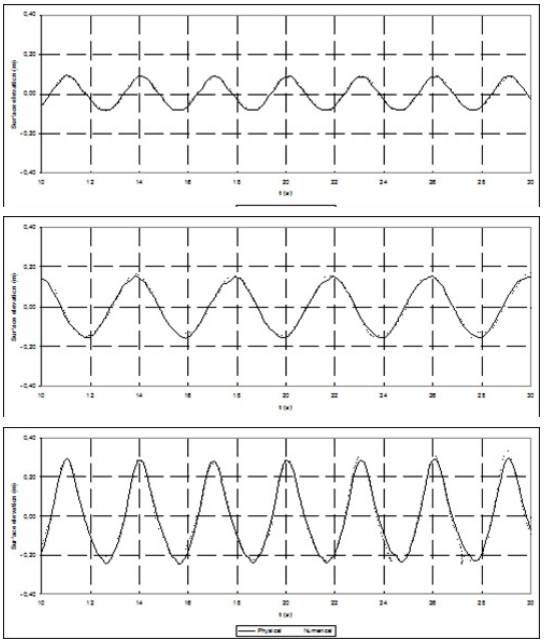


Figure 2. Comparison between physical and numerical free surface at WG0; case 1 (H=0.182m, T=3.0s), case 2 (H=0.351, T=4.0s) and case 3 (H=0.546m, T=3.0s)

The paddle was simulated as a vertical solid wall moving through a 30 degrees inclined plane according with CIEM wavemaker system layout.

The free experimental water surface obtained experimentally at WG0 (see Figure 1) is compared with numerical results for case 1, 2 and 3 in Figure 2 where the dashed line represents physical data and the dot line numerical results.

Comparison with Stokes Waves Theory

A numerical prototype scale flume with a 15m constant water depth with 250m length and with a dissipative beach of 250m length was simulated with PFEM. Four regular wave cases, called here case A (H=0.20m, T=6.93s), case B (H=1.25m, T=6.93s), case C (H=5.00m, T=6.93s) and case D (H=7.00m, T=6.93s) were generated using a numerical piston paddle. The paddle movement was determined using the corresponding first order wavemaker theory.

The maximum time step used during the simulations was 0.02s and the nodal distance 0.50m corresponding to 23078 initial nodes. A snapshot of the four cases is presented in Figure 3 after 58s of simulation.

Figure 3. Snapshot for case A, B, C and D (order from top to bottom), 58s after the beginning of simulation

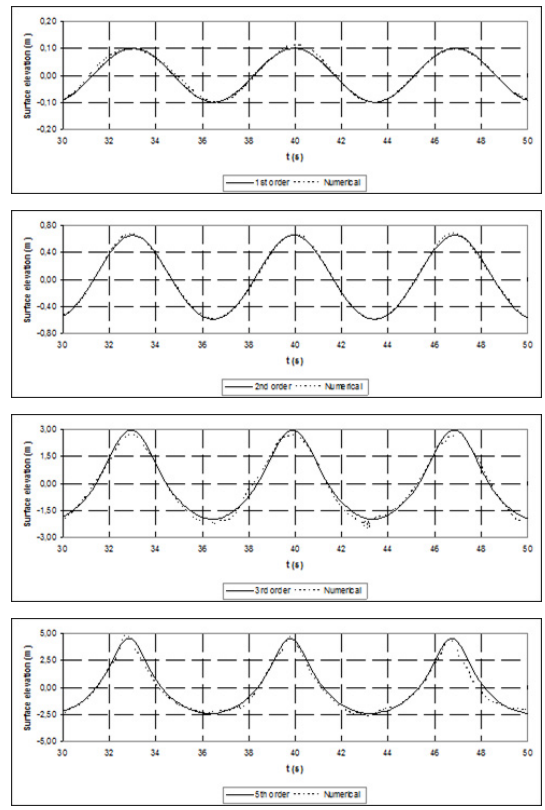


Figure 4. Comparison between numerical and Stokes wave theory for case A (H=0.20m, T=6.93s), case B (H=1.25m, T=6.93s), case C (H=5.00m, T=6.93s) and case D (H=7.00m, T=6.93s)

The comparison between the free surface given by the smallest order wave theory applicable to wave conditions tested according LE MÉHAUTÉ (1976) and the numerical results obtained at 50m far from the paddle is presented in Figure 4 for all cases. For case A the 1st order Stokes theory is applicable, for case B the 2nd order, case C with the 3rd order, and case D with 5th order.

Scale effects analysis

A regular wave of H=0.75m and T=6.93s was generated in the prototype scale numerical wave flume presented above. This situation was repeated in a 1:5 large scale numerical flume and in a 1:30 small scale numerical flume, with corresponding regular waves of H= 0.15m, T=3.10s and H=0.025m, T=1.27s.

Table 1 summarizes the three flume dimensions and the numerical features used in each scale.

Table 1: Numerical features and flume dimensions.

Scale	Mesh size (m)	Time step (s)	Depth (m)	Flume length (m)
1:1	0.50	0.020	15.0	500.0
1:5	0.10	0.009	3.0	100.0
1:30	0.02	0.004	0.5	16.6

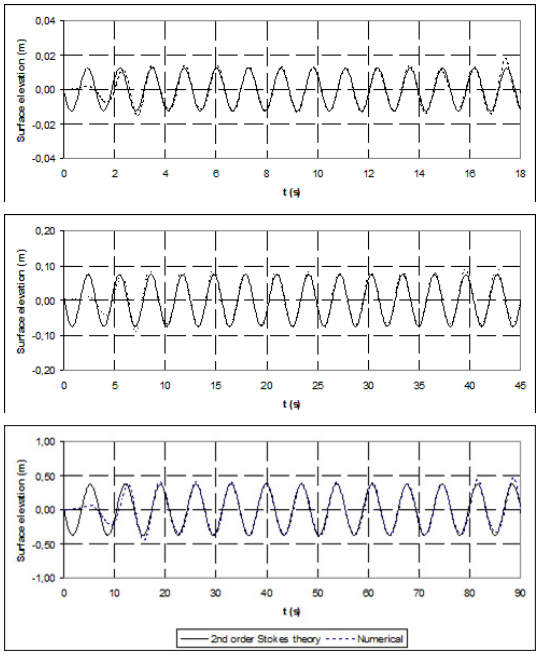


Figure 5. Comparison between numerical results and 2nd order Stokes theory for small scale (H=0.025m, T=1.27s), large scale (H=0.15m, T=3.10s) and prototype scale (H=0.75m, T=6.93s)

The paddle movement in the three scales was generated using the first order wavemaker theory for a piston paddle type. In Figure 5 the free surface obtained at small, large and prototype scale at 1.67m, 10m and 50m respectively in front of paddle is presented and compared with the free surface given by second order Stokes wave theory.

Unwanted nonlinearities generation

A regular wave of H=5.00m and T=13.0s was generated in the prototype scale numerical wave flume presented above. A piston paddle type was used to generate the wave and its movement was calculated using the first order wavemaker theory. The maximum time step used during the simulations was 0.02s and the nodes distance 0.50m corresponding to 23078 initial nodes.

The free surface obtained at 50m, 100m, and 150m far from the paddle is presented in Figure 6.

DISCUSSION

As it is seen in Figure 2 the numerical free surface results are in a good agreement with physical data. The wave height, wave period and wave shape are well reproduced by the numerical model in the three cases. However, in maximum steepness wave cases (case 3) differences up to 0.05m at the wave crest are found. Differences between the three cases in wave shape due to nonlinear effects are reproduced with numerical and physical models.

Analyzing Figure 4 it is possible to see that for case A and B the wave shapes obtained with PFEM agree well with the wave shapes proposed by the Stokes wave theories. Increasing wave height and consequently wave steepness some differences can be found (case C and D).

For case C the wave height generated is 0.20m less than the theoretical wave height being the largest differences at the wave crests. For case D at the wave crest the free surface is steeper for numerical results than in theory. As we can expect is possible to see also differences induced by nonlinear effects in wave shapes for all four cases.

Observing Figure 5 we can see that the numerical results for the three scales studied are in good agreement with 2nd Stokes wave theory. The free surface in the three scaled numerical flumes is qualitative equivalent. At the three cases the first arriving wave is smaller than expected and the second one hasn't the imposed period. This characteristic can be considered as transient waves due to wavemaker movement start effect.

As is possible to see in numerical free surface presented in Figure 6 a second crest is generated also by the paddle. The second crest position in the wave profile is not the same one at different distances from the paddle. This means that the second crest travels at a velocity different from that of the principal wave.

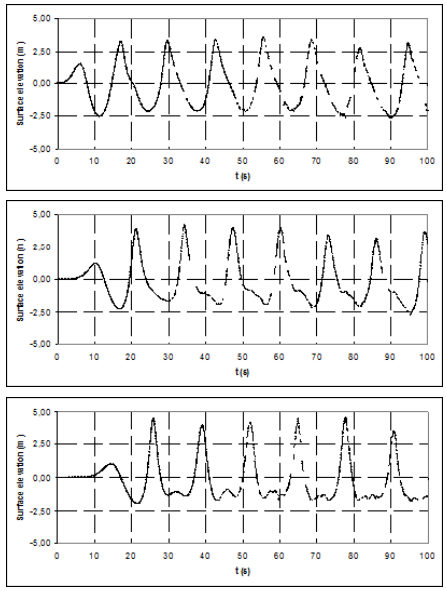


Figure 6. Numerical free surface at 50m, 100m and 150m far from the paddle (H=5m; T=13s)

CONCLUSIONS

With the capacity of PFEM to simulate solid-fluid interactions the generation of linear and nonlinear regular waves by means of different paddles types is possible.

Based on results here presented, with the PFEM formulation it is possible to simulate and to obtain the same waves generated in physical flumes reproducing the recorded physical paddle movement in the numerical paddle. By this way the real paddle performance can be imposed in the numerical paddle. This is an advantaged in front of others numerical flumes that can't add the "imperfections" in the real wavemaker systems.

The capacity of PFEM to simulate a large motion of the free surface allows the reproduction of very steep waves.

The first-order wavemaker theory used in physical flumes for different paddle types can be used in PFEM to generate a bigger range of wave conditions.

Generation of waves at different scales, including prototype dimensions can be reproduced with this numerical facility.

Like in physical flumes transient waves appear in PFEM numerical flume and should be taken into account in numerical wave studies.

The unwanted nonlinearities presents in laboratories with first-order wavemaker performance also occur in PFEM numerical flume.

PFEM numerical wave flume is a possible tool to find the range of applicability of first order wavemaker theory without the generation of unwanted nonlinearities.

Higher order wavemaker theories can be a possible solution to avoid the generation of unwanted wave nonlinearities in numerical flumes. The second-order wavemaker theories being used successfully at some physical flumes can be tested in numerical flumes based on the PFEM formulation.

LITERATURE CITED

DALRYMPLE, R.A. and ROGERS, B.D., 2006. Numerical modeling of water waves with the SPH method, *Coastal Engineering,* 53(2-3), 141-147

DEAN, R.G. and DALRYMPLE, R.A., 1992. *Water wave mechanics for Engineers and Scientists,* Prentice-Hall, New Jersey, 376p.

Dodd, N., 1998. Numerical Model of Wave Run-Up, Overtopping, and Regeneration, *Journal of Waterway, Port, Coastal, and Ocean Engineering,* 124(2), 73-81

FLICK, R.E. and GUZA, R.T., 1980. Paddle Generated Waves in Laboratory Channels, *Jornal of Waterway, Port, Coastal and Ocean Engineering, American Society of Civil Engineers,*106(1), 79-97

GINGOLD, R.A. and MONAGHAN, J.J., 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Royal Astronomical Society, Monthly Notices,* 181, 375-389.

GIRONELLA, X., 2004. Análisis experimental a gran escala de diques de baja cota de coronación., Technical University of Catalonia, Ph.D. thesis, 196p.

Havelock, T.H., 1929. Forced surface-wave on water, *Philosophical Magazine,* 8(7), 569-576.

HAWKINS, S., BURCHARTH, H.F., ZANUTTIGH, B. and LAMBERTI, A., 2007. *Environmental design guidelines for low crested coastal structures,* Elsevier, Amsterdam, 448p.

HIRT, C.W. and NICHOLS, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics,* 39(1), 201-225.

HU, K., MINGHAM, C.G. and CAUSON, D.M., 2000. Numerical simulation of wave overtopping of coastal structures using the non-linear shallow water equations, *Coastal Engineering,* 41(4), 433-465.

HUDSPETH, R.T., LEORNARD, J.W. and CHEN, M.C., 1981. Design Curves for hinged wavemakers: Experiments, *Journal of the Hydraulics Division, American Society of Civil Engineers,* 107(5), 553-574.

IDELSOHN, S.R., OÑATE, E., CALVO, N. and PIN, F.D., 2003. The meshless finite element method, *International Journal for Numerical Methods in Engineering,* 58(6), 893-912.

KOSHIZUKA, S., TAMAKO, H. and OKA, Y., 1995. A particle method for imcompressible viscous flow with fluid fragmentation, *Computational Fluid Dynamics Journal,* 4, 29-46.

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

LARA, J.L., GARCIA, N. and LOSADA, I.J., 2006. RANS modelling applied to random wave interaction with submerged permeable structures, *Coastal Engineering,* 53(5-6), 395-417.

LE MÉHAUTÉ, B., 1976. *An introduction to hydrodynamics and water waves,* Springer-Verlag, 315p.

LEMOS, C.M., 1990. Numerical modelling of shallow water waves: Application of the VOF technique and k-E turbulence model, Technical University of Catalonia, Ph.D. thesis, 300p.

LIN, P.Z. and LIU, P.L.-F., 1998. A numerical study of breaking waves in the surf zone, *Journal of fluid mechanics,* 359(1), 239-264.

MONAGHAN, J.J. and KOS, A., 1999. Solitary Waves on a Cretan Beach, *Journal of Waterway, Port, Coastal, and Ocean Engineering, American Society of Civil Engineers,*125(3),145-155.

OLIVEIRA, T.C.A., GONZALEZ-MARCO, D., SÁNCHEZ-ARCILLA, A., CELIGUETA, M.A. and OÑATE, E., 2007. Generation and propagation of water waves in a two-dimensional numerical flume. *Computational Methods in Marine Engineering II*, (Barcelona, Spain, CIMNE), pp. 241-244.

OÑATE, E., DEL PIN, F. and AUBRY, R., 2004. The Particle Finite Element Method. An overview, *International Journal of Computational Methods,* 1(2), 267-307.

Overtopping Manual, 2007. EurOtop; *Wave Overtopping of Sea Defences and Related Structures - Assessment Manual.* UK: N.W.H. Allsop, T. Pullen, T. Bruce. NL: J. W. van der Meer. DE: H. Schüttrumpf, A. Kortenhaus. http://www.overtopping-manual.com/.

OUMERACI, H., KORTENHAUS, A., ALLSOP, W., GROOT, M.D., CROUCH, R., VRIJLING, H. and VOORTMAN, H., 2001. *Probabilistic Design Tools for Vertical Breakwaters,* Aa Balkema, 366p.

SHAO, S., Ji, C., GRAHAM, D.I., REEVE, D.E., JAMES, P.W. and CHADWICK, A.J., 2006. Simulation of wave overtopping by an incompressible SPH model, *Coastal Engineering,* 53(9), 723-735.

URSELL, F., DEAN, R.G. and YU, Y.S., 1960. Forced Small Amplitude Water Waves: Comparation of Theory and Experiment, *Journal of Fluid Mechanics,* 7(1), 32-53.

VAN GENT, M.R.A., TÖNJES, P., PETIT, H.A.H. and VAN DEN BOSCH, P., 1994. Wave action on and in permeable structures, *Proc. 24th Int. Conf. on Coastal Eng (Kobe, Japan),* pp. 1739-1753.

VAN GENT, M.R.A., 1994. The modelling of wave action on and in coastal structures, *Coastal Engineering,* 22(3-4), 311-339.

ACKNOWLEDGEMENTS

The first author gratefully acknowledges the doctoral scholarship provided by "Fundação para a Ciência e Tecnologia", contract number SFRH/BD/44020/2008, funded by the European Social Fund and the Portuguese Ministry of Science and Technology.

Paper 5

This paper, published in 20011, explores the possibilities of the PFEM when mixing fluid, soil and structures. The *PFlow* code was used for all the computations.

Possibilities of the Particle Finite Element Method for fluid-soil-structure interaction problems

Eugenio Oñate · Miguel Angel Celigueta · Sergio R. Idelsohn · Fernando Salazar · Benjamín Suárez

Abstract We present some developments in the Particle Finite Element Method (PFEM) for analysis of complex coupled problems in mechanics involving fluid-soil-structure interaction (FSSI). The PFEM uses an updated Lagrangian description to model the motion of nodes (particles) in both the fluid and the solid domains (the later including soil/rock and structures). A mesh connects the particles (nodes) defining the discretized domain where the governing equations for each of the constituent materials are solved as in the standard FEM. The stabilization for dealing with an incompressibility continuum is introduced via the finite calculus (FIC) method. An incremental iterative scheme for the solution of the non linear transient coupled FSSI problem is described. The procedure to model frictional contact conditions and material erosion at fluid-solid and solid-solid interfaces is described. We present several examples of application of the PFEM to solve FSSI problems such as the motion of rocks by water streams, the erosion of a river bed adjacent to a bridge foundation, the stability of breakwaters and constructions sea waves and the study of landslides.

E. Oñate · M.A. Celigueta · F. Salazar
International Center for Numerical Methods in Engineering (CIMNE), Campus Norte UPC
Gran Capitán s/n, 08034 Barcelona, Spain
www.cimne.com
Tel.: +34-93-2057016
Fax: +34-93-4016517
E-mail: onate@cimne.upc.edu

S.R. Idelsohn
ICREA Research Professor at CIMNE

E. Oñate · M.A. Celigueta · B. Suárez
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain

1 Introduction

The analysis of problems involving the interaction of fluids, soil/rocks and structures is of relevance in many areas of engineering. Examples are common in the study of landslides and their effect on reservoirs and adjacent structures, off-shore and harbour structures under large waves, constructions hit by floods and tsunamis, soil erosion and stability of rock-fill dams in overspill situations, etc.

These studies can be considered as an extension of the so-called fluid-structure interaction (FSI) problems [46]. Typical difficulties of FSI analysis in free surface flows using the FEM both the Eulerian or ALE formulation include the treatment of the convective terms and the incompressibility constraint in the fluid equations, the modelling and tracking of the free surface in the fluid, the transfer of information between the fluid and the moving solid domains via the contact interfaces, the modeling of wave splashing, the possibility to deal with large motions of multi-bodies within the fluid domain, the efficient updating of the finite element meshes for both the structure and the fluid, etc. Examples of 3D analysis of FSI problems using ALE and space-time FEM are reported in [4,6,26,27,31,34,40].

Most of the above problems disappear if a *Lagrangian description* is used to formulate the governing equations of both the solid and the fluid domains. In the Lagrangian formulation the motion of the individual particles are followed and, consequently, nodes in a finite element mesh can be viewed as moving material points (hereforth called "particles"). Hence, the motion of the mesh discretizing the total domain (including both the fluid and solid parts) is followed during the transient solution.

A powerful Lagrangian method for FSI analysis is the so-called Soboran Grid CIP technique, which has been successfully applied to different class of 3D problems [44].

The authors have successfully developed in previous works a particular class of Lagrangian formulation for solving problems involving complex interactions between (free surface fluids) and solids. The method, called the *particle finite element method* (PFEM, www.cimne.com/pfem), treats the mesh nodes in the fluid and solid domains as particles which can freely move and even separate from the main fluid domain representing, for instance, the effect of water drops. A mesh connects the nodes discretizing the domain where the governing equations are solved using a stabilized FEM.

An advantage of the Lagrangian formulation is that the convective terms disappear from the fluid equations [11,48]. The difficulty is however transferred to the problem of adequately (and efficiently) moving the mesh nodes. We use a mesh regeneration procedure blending elements of different shapes using an extended Delaunay tesselation with special shape functions [17, 19]. The theory and applications of the PFEM are reported in [2,7,10,18,20,21,23,26,32,34–39].

The FEM solution of (incompressible) fluid flow problem implies solving the momentum and incompressibility equations. This is not a simple problem as the incompressibility condition limits the choice of the FE approximations for the velocity and pressure to overcome the well known *div*-stability condition [11,48]. In our work we use a stabilized mixed FEM based on the Finite Calculus (FIC) approach which allows for a linear approximation for the velocity and pressure variables [15, 29–31,33,34]. Among the other stabilized FEM with similar features we mention the PSPG method [41], multiscale methods [3,6,8,9] and the CBS method [9, 48].

The aim of this paper is to describe recent advances of the PFEM for fluid-soil-structure interaction (FSSI) problems. These problems are of relevance in many areas of civil, hydraulic, marine and environmental engineering, among others. It is shown that the PFEM provides a general analysis methodology for treat such complex problems in a simple and efficient manner.

The layout of the paper is the following. In the next section the key ideas of the PFEM are outlined. Next the basic equations for a compressible/incompressible continuum using a Lagrangian description and the FIC formulation are schematically presented. Then an algorithm for the transient solution is briefly described. The treatment of the coupled FSSI problem and the methods for mesh generation and for identification of the free surface nodes are outlined. The procedure for treating the frictional contact interaction between fluid, soil and structure interfaces is explained. We present several examples of application of the PFEM to solve FSSI problems such as the motion of rocks by water streams, the erosion of a river bed adjacent to a bridge foundation, the stability of breakwaters and constructions under sea waves and the study of landslides falling into reservoirs.

2 The basis of the particle finite element method

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 and, therefore, the problem is fully coupled.

In the PFEM both the fluid and the solid domains are modelled using an *updated Lagrangian formulation* [47]. 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.

2.1 Basic steps of the PFEM

For clarity purposes we will define the *collection or cloud of nodes (C)* pertaining to the fluid and solid domains, the *volume (V)* defining the analysis domain for the fluid and the solid and the *mesh (M)* discretizing both domains.

A typical solution with the PFEM involves the following steps.

1. The starting point at each time step is the cloud of points in the fluid and solid domains. For instance ${}^nC$ denotes the cloud at time $t = t_n$ (Figure 1).
2. Identify the boundaries for both the fluid and solid domains defining the analysis domain ${}^nV$ in the

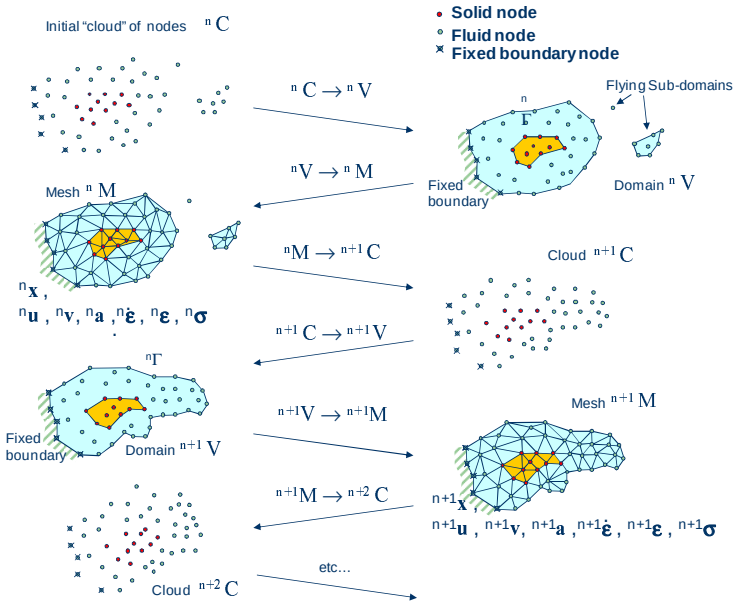


Fig. 1 Sequence of steps to update a "cloud" of nodes representing a domain containing a fluid and a solid part from time n ($t = t_n$) to time $n+2$ ($t = t_n + 2\Delta t$)

fluid and the solid. This is an essential step as some boundaries (such as the free surface in fluids) may be severely distorted during the solution, including separation and re-entering of nodes. The Alpha Shape method [12] is used for the boundary definition.

3. Discretize the fluid and solid domains with a finite element mesh ^{n}M. In our work we use an innovative mesh generation scheme based on the extended Delaunay tesselation [17,19,20].
4. Solve the coupled Lagrangian equations of motion for the fluid and the solid domains. Compute the state variables in both domains at the next (updated) configuration for $t + \Delta t$: velocities, pressure and viscous stresses in the fluid and displacements, stresses and strains in the solid.
5. Move the mesh nodes to a new position ^{n+1}C where $n+1$ denotes the time $t_n + \Delta t$, in terms of the time increment size. This step is typically a consequence of the solution process of step 4.
6. Go back to step 1 and repeat the solution process for the next time step to obtain ^{n+2}C (Figure 1).

3 FIC/FEM formulation for a Lagrangian continuum

3.1 Governing equations

The equations to be solved are the standard ones in continuum mechanics, written in the Lagrangian frame of reference:

Momentum

$$\rho \frac{\partial v_i}{\partial t} = \frac{\partial \sigma_{ij}}{\partial x_j} + b_i \qquad \text{in } V \tag{1}$$

Pressure-velocity relationship

$$\frac{1}{K}\frac{\partial p}{\partial t} - \frac{\partial v_i}{\partial x_i} = 0 \qquad \text{in } V \tag{2}$$

In above equations v_i is the velocity along the ith global (cartesian) axis, p is the pressure (assumed to be positive in compression) ρ and K are the density and bulk modulus of the material, respectively, b_i and σ_{ij} are the body forces and the (Cauchy) stresses. Eqs.(1) and (2) are completed with the constitutive relationships:

Incompressible continuum

$$^{t+1}\sigma_{ij} = 2\mu\dot{\varepsilon}_{ij} - {}^{t+1}p\delta_{ij} \tag{3}$$

Compressible/quasi-incompressible continuum

$$^{t+1}s_{ij} = {}^{t}\hat{\sigma}_{ij} + 2\mu\dot{\varepsilon}_{ij} + \lambda\dot{\varepsilon}_{ii}\delta_{ij} \tag{4a}$$

where $\hat{\sigma}_{ij}$ are the component of the stress tensor $[\hat{\sigma}]$

$$[\hat{\sigma}] = \frac{1}{J}\mathbf{F}^T\mathbf{S}\mathbf{F} \tag{4b}$$

where $\mathbf{S}$ is the second Piola-Kirchhoff stress tensor, $\mathbf{F}$ is the deformation gradient tensor and $J = \det\mathbf{F}$ [22, 47]. Parameters μ and λ take the following values for a fluid or solid material:

Fluid: μ: viscosity; $\lambda = \Delta tK - \frac{2\mu}{3}$

Solid: $\mu = \dfrac{\Delta tG}{J}$; $\lambda = \dfrac{2G\nu\Delta t}{J(1-2\nu)}$, where ν is the Poisson ration, G is the shear modulus and Δt the time increment.

In Eqs.(3) and (4), s_{ij} are the deviatoric stresses, $\dot{\varepsilon}_{ij}$ is the rate of deformation, μ is the viscosity and δ_{ij} is the Kronecker delta. ${}^{t}(\cdot)$ denotes values at time t.

Indexes in Eqs.(1)–(4) range from $i, j = 1, n_d$, where n_d is the number of space dimensions of the problem (i.e. $n_d = 2$ for 2D problems). These equations are completed with the standard boundary conditions of prescribed velocities and surface tractions in the mechanical problem [11,36,47,48].

3.2 Discretization of the equations

A key problem in the numerical solution of Eqs.(1)–(4) is the satisfaction of the mass balance condition for the incompressible case (i.e. $K = \infty$ in Eq.(2)). A number of procedures to solve his problem exist in the finite element literature [11,48]. In our approach we use a stabilized formulation based in the so-called finite calculus procedure [15,29–31,33,34]. The essence of this method is the solution of a *modified mass balance* equation which is written as

$$\frac{1}{K}\frac{\partial p}{\partial t} - \frac{\partial v_i}{\partial x_i} - \sum_{i=1}^{3}\tau\frac{\partial q}{\partial x_i}\left[\frac{\partial p}{\partial x_i} + \pi_i\right] = 0 \tag{5}$$

where q are weighting functions, τ is a stabilization parameter given by [34]

$$\tau = \left(\frac{2\rho|\mathbf{v}|}{h} + \frac{8\mu}{3h^2}\right)^{-1} \tag{6}$$

In the above, h is a characteristic length of each finite element and $|\mathbf{v}|$ is the modulus of the velocity vector. In Eq.(5) π_i are auxiliary pressure projection variables chosen so as to ensure that the second term in Eq.(5) can be interpreted as weighted sum of the residuals of the momentum equations and therefore it vanishes for the exact solution. The set of governing equations is completed by adding the following constraint equation [32,36]

$$\int_V \tau w_i\left(\frac{\partial p}{\partial x_i} + \pi_i\right)dV = 0 \quad i = 1, n_d \quad \text{(no sum in)}i \tag{7}$$

where w_i are arbitrary weighting functions.

The rest of the integral equations are obtained by applying the standard weighted residual technique to the governing equations (1), (2), (3) and (5) and the corresponding boundary conditions [11,22,48].

We interpolate next in the standard finite element fashion the set of problem variables. For 3D problems these are the three velocities v_i, the pressure p, the temperature T and the three pressure gradient projections π_i. In our work we use equal order *linear interpolation* for all variables over meshes of 3-noded triangles (in 2D) and 4-noded tetrahedra (in 3D). The resulting set of discretized equations using the standard Galerkin technique has the following form

Momentum

$$\mathbf{M}\dot{\bar{\mathbf{v}}} + \mathbf{K}\bar{\mathbf{v}} - \mathbf{G}\bar{\mathbf{p}} = \mathbf{f} \tag{8}$$

Pressure-velocity relationship

$$\bar{\mathbf{M}}\dot{\bar{\mathbf{p}}} - \mathbf{G}\bar{\mathbf{v}} - \mathbf{L}\bar{\mathbf{p}} - \mathbf{Q}\bar{\boldsymbol{\pi}} = \mathbf{0} \tag{9}$$

Pressure gradient projection

$$\hat{\mathbf{M}}\bar{\boldsymbol{\pi}} + \mathbf{Q}^T\bar{\mathbf{p}} = \mathbf{0} \tag{10}$$

In Eqs.(8)–(10) $\bar{(\cdot)}$ denotes nodal variables, $\dot{\bar{(\cdot)}} = \frac{\partial}{\partial t}\bar{(\cdot)}$. The different matrices and vectors are given in [22,34,36].

The solution in time of Eqs.(8)–(10) can be performed using any time integration scheme typical of the updated Lagrangian FEM [36,47]. A basic algorithm following the conceptual process described in Section 2 is presented in Box I.

4 Generation of a new mesh

One of the key points for the success of the PFEM is the fast regeneration of a mesh at every time step on the basis of the position of the nodes in the space domain. Indeed, any fast meshing algorithm can be used for this purpose. In our work the mesh is generated at each time step using the so called extended Delaunay tesselation (EDT) presented in [17,19].

Box I. Basic PFEM algorithm for a Lagrangian continuum

1. LOOP OVER TIME STEPS, $t = 1$, NTIME
Known values
${}^t\bar{\mathbf{x}}, {}^t\bar{\mathbf{v}}, {}^t\bar{\mathbf{p}}, {}^t\bar{\boldsymbol{\pi}}, {}^t\bar{T}, {}^t\mu, {}^t\mathbf{f}, {}^t\mathbf{q}, {}^tC, {}^tV, {}^tM$

2. LOOP OVER NUMBER OF ITERATIONS, $i = 1$, NITER

- Compute nodal velocities by solving Eq.(8)

$$\left[\frac{1}{\Delta t}\mathbf{M} + \mathbf{K}\right]{}^{t+1}\bar{\mathbf{v}}^{i+1} = {}^{t+1}\mathbf{f} + G^{t+1}\bar{\mathbf{p}}^i + \frac{1}{\Delta t}\mathbf{M}{}^t\bar{\mathbf{v}}$$

- Compute nodal pressures from Eq.(9)

$$\left[\frac{1}{\Delta t} - \mathbf{L}\bar{\mathbf{M}}\right]{}^{t+1}\bar{p}^{i+1} = \mathbf{G}^{Tt+1}\bar{v}^{i+1} + \mathbf{Q}^{t+1}\bar{\boldsymbol{\pi}}^i + \frac{1}{\Delta t}\bar{\mathbf{M}}{}^t\bar{\mathbf{p}}$$

- Compute nodal pressure gradient projections from Eq.(10)

$${}^{n+1}\bar{\boldsymbol{\pi}}^{i+1} = -\hat{\mathbf{M}}_D^{-1}\left[\mathbf{Q}^T\right]{}^{t+1}\bar{\mathbf{p}}^{i+1} \ , \ \hat{\mathbf{M}}_D = diag\left[\hat{\mathbf{M}}_D\right]$$

- Update position of analysis domain nodes:

$${}^{t+\Delta t}\bar{\mathbf{x}}^{i+1} = {}^t\mathbf{x}^i + {}^{t+\Delta t}\mathbf{v}^{i+1}\Delta t$$

Define new "cloud" of nodes ${}^{t+1}C^{i+1}$

- Update strain rate and strain values
- Update stress values

Check convergence → NO → Next iteration $i \to i+1$

↓ YES

Next time step $t \to t+1$

- Identify new analysis domain boundary: ${}^{t+1}V$
- Generate mesh:${}^{t+1}M$

Go to 1

The CPU time required for meshing grows linearly with the number of nodes. The CPU time for solving the equations exceeds that required for meshing as the number of nodes increases. This situation has been found in all the problems solved with the PFEM. As a general rule for large 3D problems meshing consumes around 15% of the total CPU time for each time step, while the solution of the equations (with typically 3 iterations to reach convergence within a time step) and the assembling of the system consume approximately 70% and 15% of the CPU time for each time step, respectively. These figures refer to solutions obtained in a standard single processor Pentium IV PC for all the computations and prove that the generation of the mesh has an acceptable cost in the PFEM. The cost of remeshing is similar to that reported in [24]. Indeed considerable speed can be gained using parallel computation techniques.

5 Identification of boundary surfaces

One of the main tasks in the PFEM is the correct definition of the boundary domain. Boundary nodes are sometimes explicitly identified. In other cases, the total set of nodes is the only information available and the algorithm must recognize the boundary nodes.

In our work we use an extended Delaunay partition for recognizing boundary nodes [19]. Considering that the nodes follow a variable $h(x)$ distribution, where $h(x)$ is typically the minimum distance between two nodes. *All nodes on an empty sphere with a radius greater than αh, are considered as boundary nodes.* In practice α is a parameter close to, but greater than one. Values of α ranging between 1.3 and 1.5 have been found to be optimal in all examples analyzed. This criterion is coincident with the Alpha Shape concept [12].

Once a decision has been made concerning which nodes are on the boundaries, the boundary surface is defined by all the polyhedral surfaces (or polygons in 2D) having all their nodes on the boundary and belonging to just one polyhedron.

The method described also allows one to identify isolated fluid particles outside the main fluid domain. These particles are treated as part of the external boundary where the pressure is fixed to the atmospheric value. We recall that each particle is a material point characterized by the density of the solid or fluid domain to which it belongs. The mass which is lost when a boundary element is eliminated due to departure of a node from the main analysis domain is again regained when the "flying" node falls down and a new boundary element is created by the Alpha Shape algorithm.

The boundary recognition method is also useful for detecting contact conditions between the fluid domain and a fixed boundary, as well as between different solids interacting with each other as detailed in the next section.

We emphasize that the main difference between the PFEM and the classical FEM is just the remeshing technique and the identification of the domain boundary at each time step.

6 Treatment of contact conditions in the PFEM

6.1 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* [32, 36].

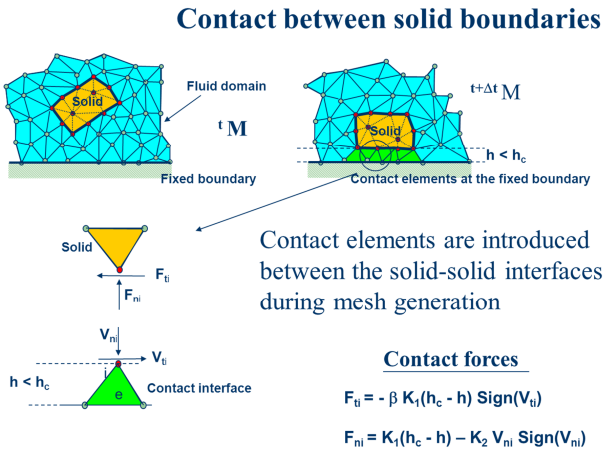


Fig. 2 Modelling of contact conditions at a solid-solid interface with the PFEM

6.2 Contact between solid-solid interfaces

The contact between two solid interfaces is simply treated 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 treated as contact elements where a relationship between the tangential and normal forces and the corresponding displacement is introduced (Figure 2).

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 [7,36].

7 Modeling of bed erosion

Prediction of bed erosion and sediment transport in open channel flows are important tasks in many areas of river and environmental engineering. Bed erosion can lead to instabilities of the river basin slopes. It can also undermine the foundation of bridge piles thereby favouring structural failure. Modeling of bed erosion is also relevant for predicting the evolution of surface material dragged in earth dams in overspill situations. Bed erosion is one of the main causes of environmental damage in floods.

Bed erosion models are traditionally based on a relationship between the rate of erosion and the shear stress level [25]. In a recent work we have proposed an extension of the PFEM to model bed erosion [35,36]. The erosion model is based on the frictional work at the bed surface originated by the shear stresses in the fluid. The resulting erosion model resembles Archard law typically used for modeling abrasive wear in surfaces under frictional contact conditions [1].

The algorithm for modeling the erosion of soil/rock particles at the fluid bed is the following:

1. Compute at every point of the bed surface the resultant tangential stress τ induced by the fluid motion. In 3D problems $\tau = (\tau_s^2 + \tau_t)^2$ where τ_s and τ_t are the tangential stresses in the plane defined by the normal direction $\mathbf{n}$ at the bed node. The value of τ for 2D problems can be estimated as follows:

$$\tau_t = \mu\gamma_t \quad \text{with} \quad \gamma_t = \frac{1}{2}\frac{\partial v_t}{\partial n} = \frac{v_t^k}{2h_k} \tag{11}$$

   where v_t^k is the modulus of the tangential velocity at the node k and h_k is a prescribed distance along the normal of the bed node k. Typically h_k is of the order of magnitude of the smallest fluid element adjacent to node k (Figure 3).
2. Compute the frictional work originated by the tangential stresses at the bed surface as

$$W_f = \int_\circ^t \tau_t\gamma_t\, dt = \int_\circ^t \frac{\mu}{4}\left(\frac{v_t^k}{h_k}\right)^2 dt \tag{12}$$

   Eq.(12) is integrated in time as

$$^nW_f = {}^{n-1}W_f + \tau_t\gamma_t\,\Delta t \tag{13}$$

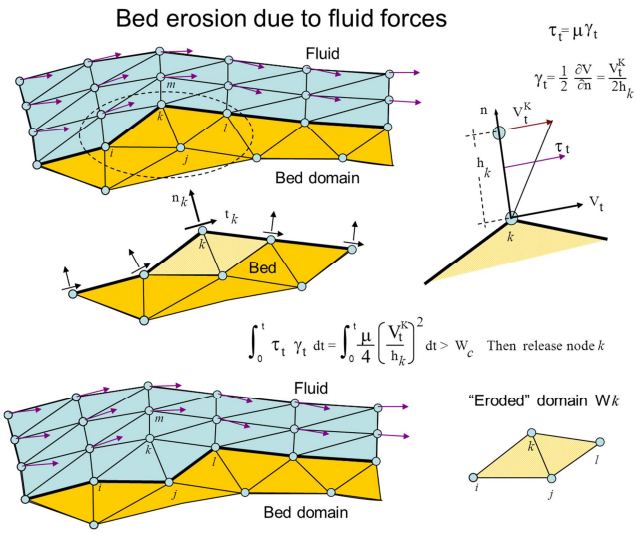


Fig. 3 Modeling of bed erosion with the PFEM by dragging of bed material

3. The onset of erosion at a bed point occurs when ${}^nW_f$ exceeds a critical threshold value W_c.
4. If ${}^nW_f > W_c$ at a bed node, then the node is detached from the bed region and it is allowed to move with the fluid flow. As a consequence, the mass of the patch of bed elements surrounding the bed node vanishes in the bed domain and it is transferred to the new fluid node. This mass is subsequently transported with the fluid.
5. Sediment deposition can be modeled by an inverse process to that described in the previous step. Hence, a suspended node adjacent to the bed surface with a velocity below a threshold value is attached to the bed surface.

Figure 3 shows an schematic view of the bed erosion algorithm described.

8 Examples

8.1 Dragging of rocks by a water stream

Predicting the critical speed at which a rock will be dragged by a water stream is of great importance in many problems in hydraulic, harbour, civil and environmental engineering.

The PFEM has been successfully applied to the study of the motion of a 1Tn quasi-spherical rock due to a water stream. The rock lays on a collection of rocks that are kept rigid. Frictional conditions between the analyzed rock and the rest of the rocks have been assumed. Figure 4a shows that a water stream of 1m/s is not able to displace the individual rock. An increase of the water speed to 2m/s induces the motion of the rock as shown in Figure 4b.

8.2 Impact of sea waves on piers and breakwaters

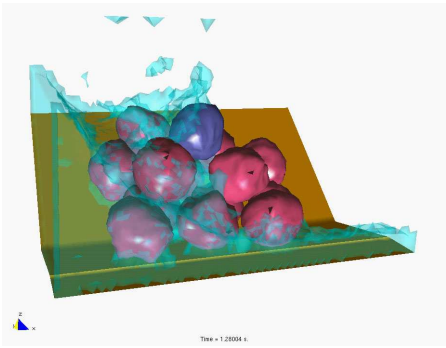
Figures 5 and 6 show the analysis of the effect of breaking waves on two different sites of a breakwater containing reinforced concrete blocks (each one of 4×4 mts). The figures correspond to the study of Langosteira harbour in A Coruña, Spain using PFEM.

8.3 Soil erosion problems

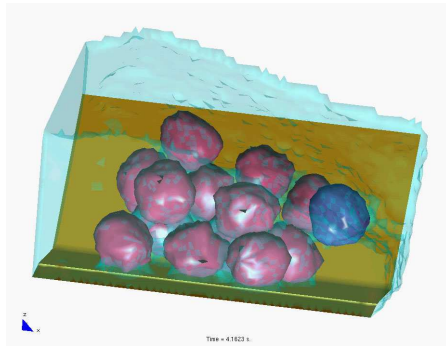
Figure 7 shows the capacity of the PFEM for modelling soil erosion, sediment transport and material deposition in a river bed. The soil particles are first detached from the bed surface under the action of the jet stream. Then they are transported by the flow and eventually fall down due to gravity forces into the bed surface at a downstream point.

Figure 8 shows the progressive erosion of the unprotected part of a breakwater slope in the Langosteira harbour in A Coruña, Spain. The non protected upper shoulder zone is progressively eroded under the sea waves.

Figure 9 displays the progressive erosion and dragging of soil particles in a river bed adjacent to the foot of bridge pile due to the water stream (water is not shown in the figure). Note the disclosure of the bridge



(a) Water speed of 1m/s. The individual rock can not be dragged by the stream



(b) Water speed of 2m/s. The individual rock is dragged by the stream

Fig. 4 Study of the drag of an individual rock of 1Tn under a water stream at speeds of (a) 1m/s and (b) 2m/s

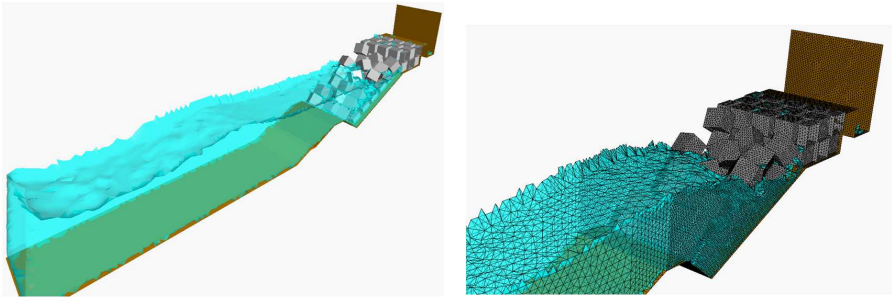


Fig. 5 Breaking waves on breakwater slope containing reinforced concrete blocks. Mesh of 4-noded tetrahedra near the slope

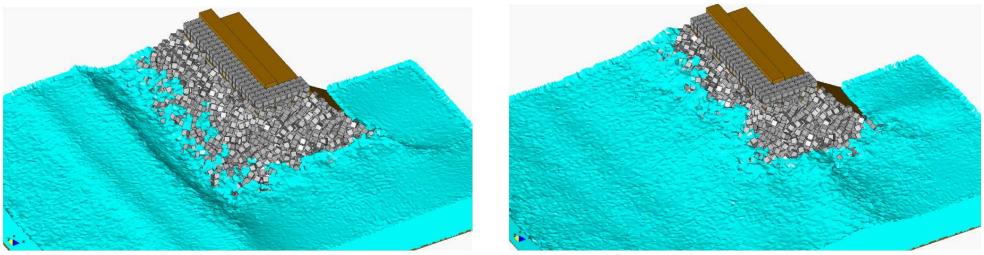


Fig. 6 Study of breaking waves on the edge of a breakwater structure formed by reinforced concrete blocks

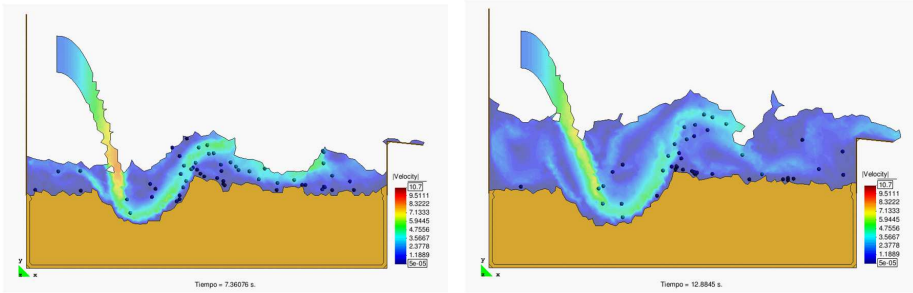


Fig. 7 Erosion, transport and deposition of soil particles at a river bed due to an impacting jet stream

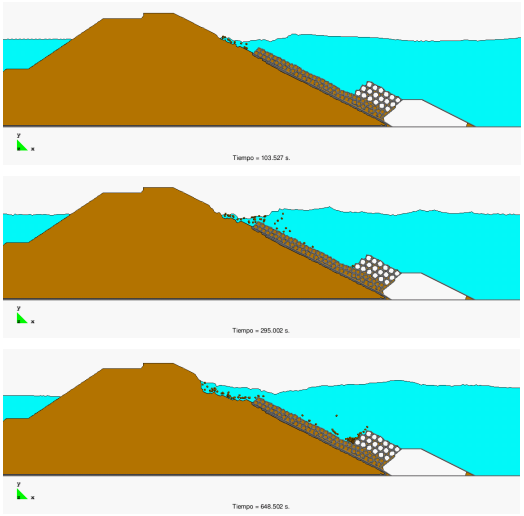


Fig. 8 Erosion of an unprotected shoulder of a breakwater due to sea waves

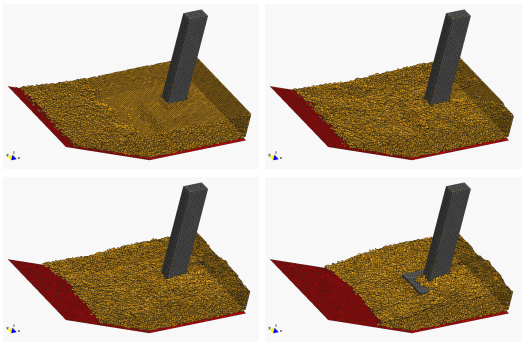


Fig. 9 Erosion and dragging of soil particles in a river bed adjacent to the foot of a bridge pile due to a water stream. Water is not shown

foundation as the adjacent soil particles are removed due to erosion.

Other applications of the PFEM to bed erosion problems can be found in [35,36].

8.4 Falling of a lorry into the sea by sea wave erosion of the road slope

Figure 10 shows a representative example of the progressive erosion of a soil mass adjacent to the shore due to sea waves and the subsequent falling into the sea of a 2D object representing the section of a lorry. The object has been modeled as a rigid solid.

This example, although still quite simple and schematic, shows the possibility of the PFEM for modeling complex FSSI problems involving soil erosion, free surface waves and rigid/deformable structures.

8.5 Simulation of landslides

The PFEM is particularly suited for modelling and simulation of landslides and their effect in the surrounding structures. Figure 11 shows an schematic 2D simulation of a landslide falling on two adjacent constructions. The landslide material has been modelled as a viscous incompressible fluid.

8.6 The landslide in Lituya Bay

A case of much interest is when the landslide occurs in the vicinity of a reservoir [43]. 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 the example, we present some results of the 3D analysis of the landslide produced in Lituya Bay (Alaska) on July 9th 1958 (Figure 12). The landslide was originated by an earthquake and movilized 90 millions tons of rocks that fell on the bay originating a large wave that reached a hight on the opposed slope of 524 mts.

Figure 13 shows images of the simulation of the landslide with 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 erosion and dragging of soil material induced by the landslide mass during motion.

PFEM results have been compared with observed values of the maximum water level in the north hill adjacent to the reservoir. The maximum water level in this hill obtained with PFEM was 551 mts. This is 5% higher than the value of 524 mts. observed experimental by [13,14]. The maximum height location differs in 300 mts from the observed value [13,14]. In the south slope the maximum water height observed was 208 mts, while the PFEM result (not shown here) was 195 mts (6% error).

More information on the PFEM solutions of this example can be found in [38,39].

9 Conclusions

The particle finite element method (PFEM) is a promising numerical technique for solving fluid-soil-structure

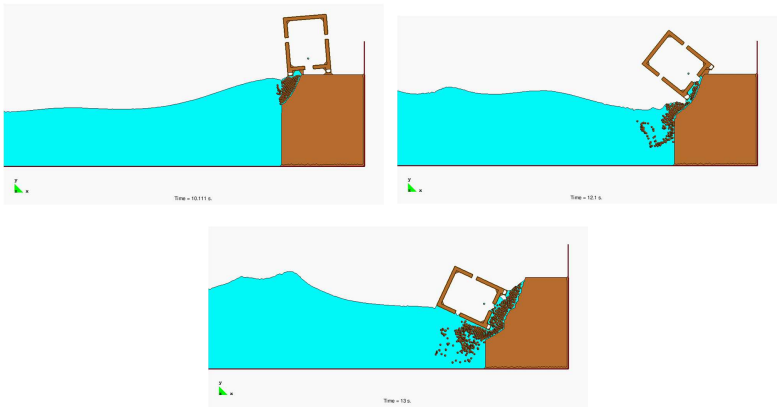


Fig. 10 Erosion of a soil mass due to sea waves and the subsequent falling into the sea of an adjacent lorry

Fig. 11 Landslide falling on two constructions 2D simulation using PFEM

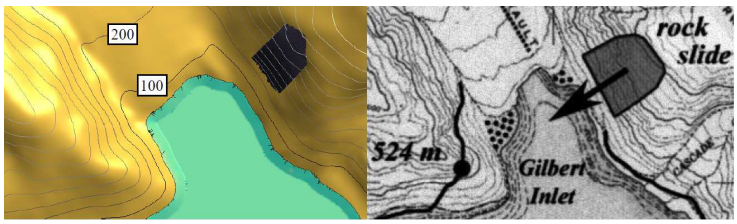


Fig. 12 Lituya Bay landslide. Left: Geometry for the simulation. Right: Landslide direction and maximum wave level [13, 14]

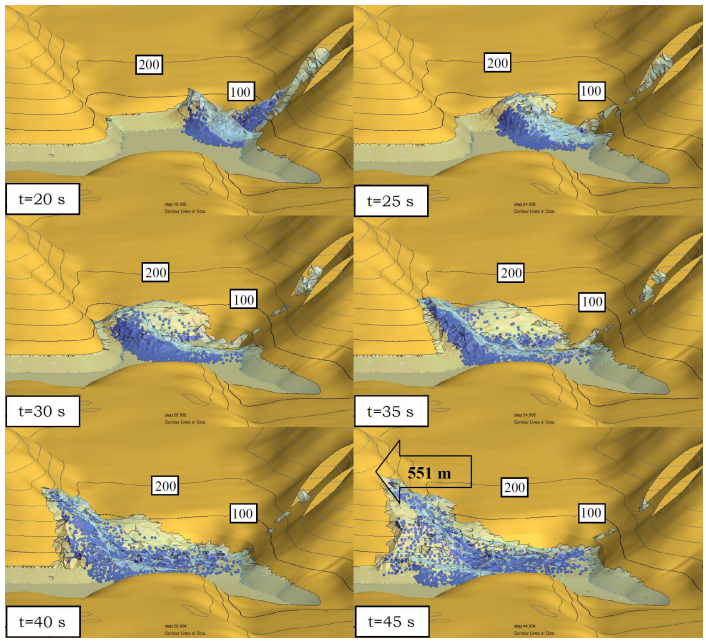


Fig. 13 Lituya Bay landslide. Evolution of the landslide into the reservoir obtained with the PFEM. Maximum level of generated wave (551 mts) in the north slope

interaction (FSSI) problems involving large motion of fluid and solid particles, surface waves, water splashing, frictional contact situations between fluid-solid and solid-solid interfaces and bed erosion, among other complex phenomena. The success of the PFEM lies in the accurate and efficient solution of the equations of an incompressible continuum using an updated Lagrangian formulation and a stabilized finite element method allowing the use of low order elements with equal order interpolation for all the variables. Other essential solution ingredients are the efficient regeneration of the finite element mesh, the identification of the boundary nodes using the Alpha-Shape technique and the simple algorithm to treat frictional contact conditions and erosion/wear at fluid-solid and solid-solid interfaces via mesh generation. The examples presented have shown the potential of the PFEM for solving a wide class of practical FSSI problems in engineering.

Acknowledgements

This research was partially supported by project SEDUREC of the Consolider Programme of the Ministerio de Educación y Ciencia (MEC) of Spain and the projects SAFECON and REALTIME of the European Research Council of the European Commission (EC). Thanks are also given to the Spanish construction company Dragados for financial support for the study of harbour engineering problems with the PFEM .

References

1. Archard JF (1953) Contact and rubbing of flat surfaces. J. Appl. Phys. 24(8):981–988
2. Aubry R, Idelsohn SR, Oñate E (2005) Particle finite element method in fluid mechanics including thermal convection-diffusion, Computer & Structures 83(17-18):1459–1475
3. Badia S, Codina R (2009) On a multiscale approach to the transient Stokes problem. Transient subscales and anisotropic space-time discretizations. Applied Mathematics and Computation 207:415–423
4. Baiges J, Codina R (2010) The fixed-mesh ALE approach applied to solid mechanics and fluid-structure interaction problems. Int. J. Num. Meth. Engrg. 81:1529–1557
5. Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G (2007) Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comp. Meth. Appl. Mech Engrg. 197:173–201
6. Bazilevs Y, Hsu M-C, Zhang Y, Wang W, Liang X, Kvamsdal T, Brekken R, Isaksen J (2010) A fully coupled fluid-structure interaction simulation of cerebral aneurysms. Computational Mechanics 46:3–16
7. Carbonell JM, Oñate E, Suárez B (2010) Modeling of ground excavation with the Particle Finite Element Method. Journal of Engineering Mechanics (ASCE) 136(4):455–463

8. Codina R (2002) Stabilized finite element approximation of transient incompressible flows using subscales. Comput. Meth. Appl. Mech. Engrg. 191: 4295–4321
9. Codina R, Coppola-Owen H, Nithiarasu P, Liu CB (2006) Numerical comparison of CBS and SGS as stabilization techniques for the incompressible Navier-Stokes equations. Int. J. Num. Meth. Engrg. 66:1672–1689
10. Del Pin F, Idelsohn SR, Oñate E, Aubry R (2007) The ALE/Lagrangian particle finite element method: A new approach to computation of free-surface flows and fluid-object interactions. Computers & Fluids 36:27–38
11. Donea J, Huerta A (2003) Finite element method for flow problems. J. Wiley
12. Edelsbrunner H, Mucke EP (1999) Three dimensional alpha shapes. ACM Trans. Graphics 13:43–72
13. Fritz, HM, Hager WH, Minor HE (2001) Lituya Bay Case: Rockslide impact and wave run-up. Science of Tsunami Hazards 19(1):3–22
14. Fritz, HM, Hager WH, Minor HE (2004) Near field characteristics of landslide generated impulse waves. Journal of Waterway, Port, Coastal and Ocean Engineering. ASCE 130(6):287–302
15. García J, Oñate E (2003) An unstructured finite element solver for ship hydrodynamic problems. J. Appl. Mech. 70:18–26
16. Idelsohn SR, Oñate E, Del Pin F, Calvo N (2002) Lagrangian formulation: the only way to solve some free-surface fluid mechanics problems. 5th World Congress on Comput. Mechanics, HA Mang, FG Rammerstorfer, J Eberhardsteiner (Eds), July 7–12, Viena, Austria
17. Idelsohn SR, Oñate E, Calvo N, Del Pin F (2003a) The meshless finite element method. Int. J. Num. Meth. Engng. 58(6):893–912
18. Idelsohn SR, Oñate E, Del Pin F (2003b) A lagrangian meshless finite element method applied to fluid-structure interaction problems. Comput. and Struct. 81:655–671
19. Idelsohn SR, Calvo N, Oñate E (2003c) Polyhedrization of an arbitrary point set. Comput. Method Appl. Mech. Engng. 192(22-24):2649–2668
20. Idelsohn SR, Oñate E, Del Pin F (2004) 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
21. Idelsohn SR, Oñate E, Del Pin F, Calvo N (2006) Fluid-structure interaction using the particle finite element method. Comput. Meth. Appl. Mech. Engng. 195:2100–2113
22. Idelsohn SR, Marti J, Limache A, Oñate E (2008) Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluid-structure interaction problems via the PFEM. Comput Methods Appl Mech Engrg. 197:1762–1776
23. Idelsohn SR, Mier-Torrecilla M, Oñate E (2009) Multi-fluid flows with the Particle Finite Element Method. Comput Methods Appl Mech Engrg. 198:2750–2767
24. Johnson AA, Tezduyar TE (1999) Advanced mesh generation and update methods for 3D flow simulations. Computational Mechanics 23:130–143
25. Kovacs A, Parker G (1994) A new vectorial bedload formulation and its application to the time evolution of straight river channels. J. Fluid Mech. 267:153–183
26. Larese A, Rossi R, Oñate E, Idelsohn SR (2008) Validation of the Particle Finite Element Method (PFEM) for simulation of free surface flows. Engineering Computations 25(4):385–425
27. Löhner R (2008) Applied CFD Techniques. J. Wiley
28. Löhner R, Yang Ch, Oñate E (2007) Simulation of flows with violent free surface motion and moving objects using unstructured grids. Int. J. Num. Meth. Fluids 153:1315–1338
29. Oñate E (1998) Derivation of stabilized equations for advective-diffusive transport and fluid flow problems. Comput. Meth. Appl. Mech. Engng. 151:233–267
30. Oñate E (2004) Possibilities of finite calculus in computational mechanics. Int. J. Num. Meth. Engng. 60(1):255–281
31. Oñate E, García J (2001) A finite element method for fluid-structure interaction with surface waves using a finite calculus formulation. Comput. Meth. Appl. Mech. Engrg. 191:635–660
32. Oñate E, Idelsohn SR, Del Pin F, Aubry R (2004b) The particle finite element method. An overview. Int. J. Comput. Methods 1(2):267–307
33. Oñate E, Valls A, García J (2006a) FIC/FEM formulation with matrix stabilizing terms for incompressible flows at low and high Reynold's numbers. Computational Mechanics 38 (4-5):440–455
34. Oñate E, García J, SR Idelsohn, F. Del Pin (2006b) FIC formulations for finite element analysis of incompressible flows. Eulerian, ALE and Lagrangian approaches. Comput. Meth. Appl. Mech. Engng. 195(23-24):3001–3037
35. Oñate E, M.A. Celigueta, Idelsohn SR (2006c) Modeling bed erosion in free surface flows by the Particle Finite Element Method, Acta Geotechnia 1(4):237–252
36. Oñate E, Idelsohn SR, Celigueta MA, Rossi R (2008) 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
37. Oñate E, Rossi R, Idelsohn SR, Butler K (2010) Melting and spread of polymers in fire with the particle finite element method. Int. J. Numerical Methods in Engineering, 81(8):1046–1072
38. Oñate, E, Salazar F, Morán R (2011) Modeling of landslides into reservoir with the Particle Finite Element Method. Research Report CIMNE No. PI355. Submitted to Int. J. Numerical Methods in Geomechanics
39. Salazar F, Oñate E, Morán R (2011) Modelación numérica de deslizamientos de ladera en embalses mediante el método de partículas y elementos finitos (PFEM). Rev. Int. Mét. Num. Cálc. Dis. Ing. Acepted for publication
40. Takizawa K, Tezduyar TE (2011) Multiscale space-time fluid-structure interaction techniques. Computational Mechanics, Published online, DOI:10.1007/s00466-011-0571-z
41. Tezduyar TE, Mittal S, Ray SE, Shih R (1992) Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. Comput. Meth. Appl. Mech. Engng. 95:221–242
42. Tezduyar TE (2007) Finite elements in fluids: special methods and enhanced solution techniques. Computers & Fluids 36:207–223
43. Wan CF, Fell R (2004) Investigation of erosion of soils in embankment dams. J. Geotechnical and Geoenvironmental Engineering 130:373–380
44. Yabe T, Takizawa K, Tezduyar TE, Im H-N (2007) Computation of fluid-solid and fluid-fluid interfaces with the CIP method based on adaptive Soroban Grids. An overview. Int. J. Num. Meth. in Fluids 54:841–853
45. Zienkiewicz OC, Jain PC, Oñate E (1978) Flow of solids during forming and extrusion: Some aspects of numerical solutions. Int. Journal of Solids and Structures 14:15–38

46. Zienkiewicz OC, Taylor RL, Zhu JZ (2005) The finite element method. Its basis and fundamentals, Elsevier
47. Zienkiewicz OC, Taylor RL (2005) The finite element method for solid and structural mechanics, Elsevier
48. Zienkiewicz OC, Taylor RL, Nithiarasu P (2005) The finite element method for fluid dynamics, Elsevier

Paper 6

This paper, published in 2014, extends the research published in the previous paper ([6]) and adds the technology of particles transport in fluids introduced in Section 5. The *PFlow* code was used for all the computations.

Lagrangian analysis of multiscale particulate flows with the particle finite element method

Eugenio Oñate · Miguel Angel Celigueta · Salvador Latorre · Guillermo Casas · Riccardo Rossi · Jerzy Rojek

Abstract We present a Lagrangian numerical technique for the analysis of flows incorporating physical particles of different sizes. The numerical approach is based on the Particle Finite Element Method (PFEM) which blends concepts from particle-based techniques and the FEM. The basis of the Lagrangian formulation for particulate flows and the procedure for modelling the motion of small and large particles that are submerged in the fluid are described in detail. The numerical technique for analysis of this type of multiscale particulate flows using a stabilized mixed velocity-pressure formulation and the PFEM is also presented. Examples of application of the PFEM to several particulate flows problems are given.

Keywords Lagrangian analysis · Multiscale particulate flows · Particle finite element method

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. In this work we are concerned with particulate flows containing small to large particles. 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), among other applications [1, 2, 6, 13, 14, 16, 21–23, 26, 47, 50, 51, 55, 61, 62].

Our interest in this work is the modelling and simulation of free surface particulate quasi-incompressible flows containing particles of different sizes using a particular class of Lagrangian FEM termed the Particle Finite Element Method (PFEM, www.cimne.com/pfem) [4, 5, 8, 11, 17–20, 25, 27, 28, 35, 36, 38, 40, 42–46]. The PFEM treats the mesh nodes in the fluid and solid domains as particles which can freely move and even separate from the main fluid domain. A mesh connects the nodes discretizing the domain where the governing equations are solved using a stabilized FEM.

In Lagrangian analysis procedures (such as the PFEM) the motion of fluid particles is tracked during the transient solution. Hence, the convective terms vanish in the momentum equations and no numerical stabilization is needed. Another source of instability, however, remains in the numerical solution of Lagrangian flows, that due to the treatment of the incompressibility constraint which requires using a stabilized numerical method.

In this work we use a stabilized Lagrangian formulation that has excellent mass preservation features. The success of the formulation relies on the consistent derivation of a residual-based stabilized expression of the mass balance equation using the Finite Calculus (FIC) method [29–33, 37–39].

E. Oñate, M.A. Celigueta, S. Latorre, G. Casas, R. Rossi
Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE)
Campus Norte UPC, 08034 Barcelona, Spain
Tel.: +34-93-2057016
Fax: +34-93-4016517
E-mail: onate,maceli,latorre,gcasas,rrossi@cimne.upc.edu

J. Rojek
Institute of Fundamental Technological Research
Polish Academy of Sciences
E-mail: jrojek@ippt.pan.pl

The lay-out of the paper is the following. In the next section we present the basic equations for conservation of linear momentum and mass for a quasi-incompressible particulate fluid in a Lagrangian framework. The treatment of the different force terms for micro, macro and large particles are explained. Next we derive the stabilized FIC form of the mass balance equation. Then, the finite element discretization using simplicial element with equal order approximation for the velocity and the pressure is presented and the relevant matrices and vectors of the discretized problem are given. Details of the implicit transient solution of the Lagrangian FEM equations for a particulate flow using a Newton-Raphson type iterative scheme are presented. The basic steps of the PFEM for solving free-surface particulate flow problems are described.

The efficiency and accuracy of the PFEM for analysis of particulate flows are verified by solving a set of free surface and confined fluid flow problems incorporating particles of small and large sizes in two (2D) and three (3D) dimensions. The problems include the study of soil erosion, landslide situations, tsunami and flood flows, soil dredging problems and particle filling of fluid containers, among others. The excellent performance of the numerical method proposed for analysis of particulate flows is highlighted.

2 Modelling of micro, macro and large particles

Figure 1 shows a domain containing a fluid and particles of different sizes. Particles will be termed *microscopic*, *macroscopic* and *large* in terms of their dimensions. Microscopic and macroscopic particles will be assumed to have a cylindrical (in 2D) or spherical (in 3D) shape. These particles are modelled as rigid objects that undergo interaction forces computed in terms of the relative distance between particles (for microscropic particles) or via the physical contact between a particle and its neighbors (for macroscopic particles), as in the standard discrete element method (DEM) [2, 16, 34].

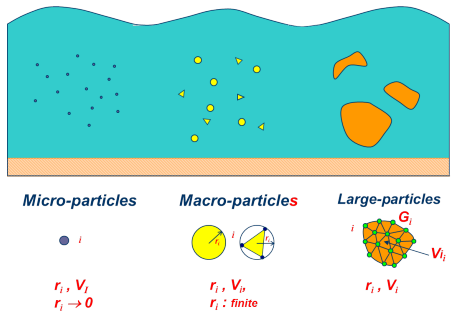


Fig. 1: Microscopic, macroscopic and large particles within a fluid domain

In this work microscopic and macroscopic particles are assumed to be spherical and submerged in the fluid (Figure 2). 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 3). These equations, as well as the mass balance equations account for the percentage of particles in the fluid, similarly at it is done in standard immersed approaches for particulate flows [53, 54, 56].

Large particles, on the other hand, can have any arbitrary shape and they can be treated as rigid or deformable bodies. In the later case, they are discretized with the standard FEM. The forces between the fluid and the particles and viceversa are computed via fluid-structure interaction (FSI) procedures [31, 60].

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.

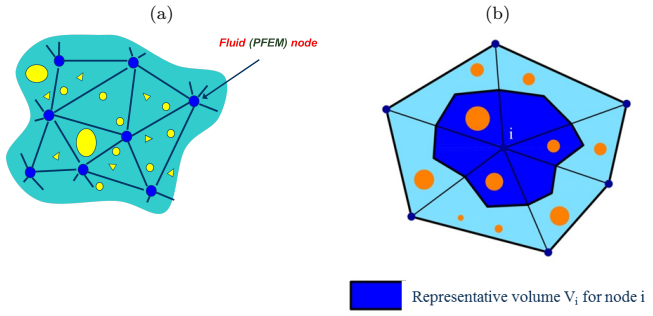


Fig. 2: (a) Particles of different sizes surrounding the nodes in a FEM mesh. (b) Representative volume for a node (in shadowed darker colour)

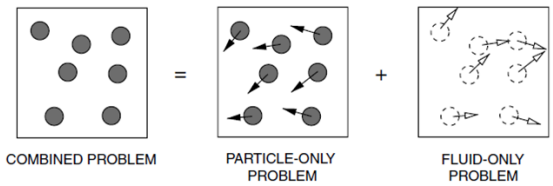


Fig. 3: Immersed approach for treating the motion of physical particles in a fluid [61]

3 Basic governing equations for a Lagrangian particulate fluid [1, 22, 23, 61]

Conservation of linear momentum

$$\rho_f \frac{Dv_i}{Dt} = \frac{\partial \sigma_{ij}}{\partial x_j} + b_i + \frac{1}{n_f} f_i^{pf} \quad , \quad i,j = 1, \cdots, n_s \quad \text{in } V \tag{1}$$

In V is the analysis domain, n_s is the number of space dimensions ($n_s = 3$ for 3D problems), ρ_f is the density of the fluid, v_i and b_i are the velocity and body force components along the ith cartesian axis, respectively, σ_{ij} are the fluid Cauchy stresses, f_i^{pf} are averaged particle-to-fluid interaction forces for which closure relations must be provided and n_f is the fluid volume fraction defined for each node j as

$$n_{f_j} = 1 - \frac{1}{V_j} \sum_{i=1}^{n_j} V_j^i \tag{2}$$

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. Note that $n_{f_j} = 1$ for a representative fluid domain containing no particles (Figure 2).

The fluid volume fraction n_f in Eq.(1) is a continuous function that is interpolated from the nodal values in the finite element fashion [22].

Summation of terms with repeated indices is assumed in Eq.(1) and in the following, unless otherwise specified.

Remark 1. The term $\frac{Dv_i}{Dt}$ in Eq.(1) is the *material derivative* of the velocity v_i. This term is typically computed in a Lagrangian framework as

$$\frac{Dv_i}{Dt} = \frac{{}^{n+1}v_i - {}^{n}v_i}{\Delta t} \tag{3a}$$

with

$${}^{n+1}v_i := v_i({}^{n+1}\mathbf{x}, {}^{n+1}t) \quad , \quad {}^{n}v_i := v_i({}^{n}\mathbf{x}, {}^{n}t) \tag{3b}$$

where ${}^{n}v_i({}^{n}\mathbf{x}, {}^{n}t)$ is the velocity of the material point that has the position ${}^{n}\mathbf{x}$ at time $t = {}^{n}t$, where $\mathbf{x} = [x_1, x_2, x_3]^T$ is the coordinates vector of a point in a fixed Cartesian system. Note that the convective term, typical of Eulerian formulations, does not appear in the definition of the material derivative [3, 9, 60].

Constitutive equations

The Cauchy stresses σ_{ij} in the fluid are split in the deviatoric (s_{ij}) and pressure (p) components as

$$\sigma_{ij} = s_{ij} + p\delta_{ij} \tag{4}$$

where δ_{ij} is the Kronecker delta. In this work the pressure is assumed to be positive for a tension state.

The relationship between the deviatoric stresses and the strain rates has the standard form for a Newtonian fluid [9],

$$s_{ij} = 2\mu\varepsilon'_{ij} \quad \text{with } \varepsilon'_{ij} = \varepsilon_{ij} - \frac{1}{3}\varepsilon_v\delta_{ij} \quad \text{and} \quad \varepsilon_v = \varepsilon_{ii} \tag{5}$$

In Eq.(5) μ is the viscosity, ε'_{ij} and ε_v are the deviatoric and volumetric strain rates, respectively. The strain rates ε_{ij} are related to the velocities by

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \tag{6}$$

Mass conservation equation

The mass conservation equation for a particulate flow is written as

$$r_v = 0 \tag{7a}$$

with

$$r_v := \frac{D(n_f\rho_f)}{Dt} + n_f\rho_f\varepsilon_v \tag{7b}$$

Expanding the material derivative and dividing Eq.(7a) by n_f the expression of r_v can be rewritten as

$$r_v := \frac{1}{\kappa}\frac{Dp}{Dt} + \frac{1}{n_f}\frac{Dn_f}{Dt} + \varepsilon_v \tag{8}$$

where $\kappa = \rho_f c^2$ and $c = -\frac{\partial p}{\partial \rho}$ is the speed of sound in the fluid.

Remark 2. For $n_f = 1$ (i.e. no particles are contained in the fluid) the standard momentum and mass conservation equations for the fluid are recovered.

Boundary conditions

The boundary conditions at the Dirichlet (Γ_v) and Neumann (Γ_t) boundaries with $\Gamma = \Gamma_v \cup \Gamma_t$ are

$$v_i - v_i^p = 0 \qquad \text{on } \Gamma_v \tag{9}$$

$$\sigma_{ij}n_j - t_i^p = 0 \qquad \text{on } \Gamma_t \quad i,j = 1,\cdots,n_s \tag{10}$$

where v_i^p and t_i^p are the prescribed velocities and prescribed tractions on the Γ_v and Γ_t boundaries, respectively and n_j are the components of the unit normal vector to the boundary [3, 9, 60].

At a free surface the Neumann boundary conditions typically apply.

4 Motion of microscopic and macroscopic particles

As mentioned early, microscopic and macroscopic particles are assumed to be rigid spheres (in 3D). Their motion follows the standard law for Lagrangian particles. For the ith spherical particle

$$m_i\dot{\mathbf{u}}_i = \mathbf{F}_i \quad , \quad J_i\dot{\mathbf{w}}_i = \mathbf{T}_i \tag{11}$$

where $\mathbf{u}_i$ and $\mathbf{w}_i$ are the velocity and rotation vector of the center of gravity of the particle, m_i and J_i are the mass and rotational inertia of the particle, respectively and $\mathbf{F}_i$ and $\mathbf{T}_i$ are the vectors containing the forces and torques acting at the gravity center of the particle [34].

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

$$\mathbf{F}_i = \mathbf{F}_i^w + \mathbf{F}_i^c + \mathbf{F}_i^{fp} \tag{12}$$

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

Self-weight forces

$$\mathbf{F}_i^w = -\rho_i \Omega_i \mathbf{g} \tag{13}$$

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.

Contact forces

$$\mathbf{F}_i^c = \sum_{j=1}^{n_i} \mathbf{F}_{ij}^c \tag{14}$$

where n_i is the number of contact interfaces for the ith particle.

$$\mathbf{F}_{ij}^c = \mathbf{F}_n^{ij} + \mathbf{F}_s^{ij} = F_n^{ij} n_i + \mathbf{F}_s^{ij} \tag{15}$$

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 (Figure 4). These forces are computed in terms of the relative motion of the interacting particles as in the standard DEM [2, 16, 34].

For microscopic particles the tangential forces $\mathbf{F}_s^{ij}$ are neglected in Eq.(15).

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

$$\mathbf{F}_i^b = \Omega_i \boldsymbol{\nabla} p \tag{16}$$

Drag forces

$$\mathbf{F}_i^d = \mathbf{f}_i^d n_f^{-(\chi+1)}$$

where $\chi = \chi(Re)$ is a coefficient that depends on the local Reynolds number for the particle Re [1, 6, 15, 22, 23] and

$$\mathbf{f}_i^d = \frac{1}{2} \rho_f A_i C_d \|\mathbf{v}_{f_i} - \mathbf{v}_i\| (\mathbf{v}_{f_i} - \mathbf{v}_i) \tag{17}$$

In Eq.(17) $\mathbf{v}_{fi}$ and $\mathbf{v}_i$ are respectively the velocity of the fluid and of the particle center, A_i is the area of the particle surface with radius r_i ($2\pi r_i$ or $4\pi r_i^2$ for a circle or a sphere, respectively) and C_d is a drag coefficient that depends on the particle geometry and the rugosity of its surface [22, 23].

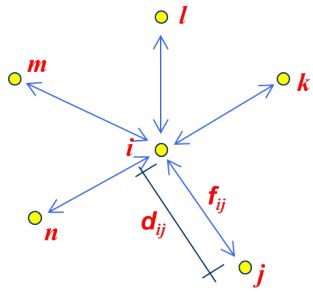
The force term f_i^{pf} in the r.h.s. of the momentum equations (Eq.(1)) is computed for each particle (in vector form) as $\mathbf{f}^{pf} = -\mathbf{f}^{fp}$ with vector $\mathbf{f}^{fp}$ computed *at each node* in the fluid mesh from the drag forces $\mathbf{F}_i^d$ as

$$\mathbf{f}_j^{fp} = \frac{1}{V_j} \sum_{i=1}^{n_j} \mathbf{F}_i^d \quad , \quad j = 1, N \tag{18}$$

A continuum distribution of $\mathbf{f}^{fp}$ is obtained by interpolating its nodal values over each element in the FEM fashion.

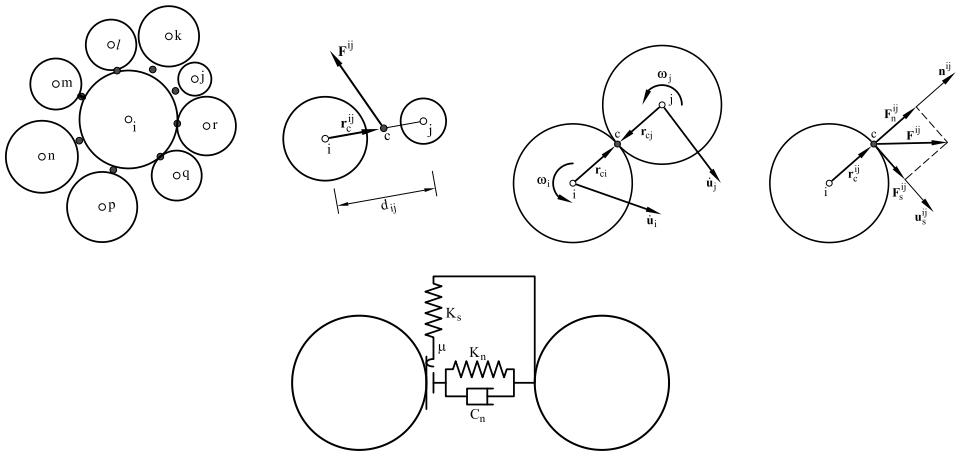
We note that the forces on the particles due to lift effects have been neglected in the present analysis. These forces can be accounted for as explained in [22].

(a) Contact between microscopic particles



Forces between two microscopic particles act in the direction of the line connecting their radii

(b) Contact "a la DEM" between macroscopic particles [34]



$\mathbf{F}_n^{ij} = K_n \mathbf{u}_n^{ij} + C_n \dot{\mathbf{u}}_n^{ij}$
$\mathbf{F}_s^{ij} = K_s \mathbf{u}_s^{ij}$

$\mathbf{u}_n^{ij}, \mathbf{u}_s^{ij}$ relative displacements in the normal and tangential directions to a contact interface

Fig. 4: Interacting forces between microscopic (a) and macroscopic (b) particles

5 Motion of large particles

As mentioned earlier, large particles may be considered as rigid or deformable bodies. In the first case the motion follows the rules of Eq.(11) 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 10.2).

The fluid forces on the particles are computing by integrating the tangential (viscous) and normal (pressure) forces at the edges of the fluid elements surrounding the particles.

Large deformable particles, on the other hand, 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. Details of this unified treatment of the interaction between fluids and deformable solids can be found in [12, 18, 46].

6 Stabilized FIC form of the mass balance equation

The modelling of incompressible fluids with a mixed finite element method using an equal order interpolation for the velocities and the pressure requires introducing a stabilized formulation for the mass balance equation.

In our work we use a stabilized form of the mass balance equation obtained via the Finite Calculus (FIC) approach [29–33, 37–39]. The FIC stabilized mass balance equation is written as

$$r_v - \tau \frac{\partial \bar{r}_{m_i}}{\partial x_i} = 0 \quad \text{in } V \tag{19}$$

where

$$\bar{r}_{m_i} = \frac{\partial \sigma_{ij}}{\partial x_j} + b_i + \frac{1}{n_f} f_i^{pf} \tag{20}$$

is a static momentum term and τ is a stabilization parameter computed as

$$\tau = \left(\frac{8\mu}{h^2} + \frac{2\rho_f}{\delta} \right)^{-1} \tag{21}$$

where h is a characteristic distance of each finite element and δ is a time parameter.

The derivation of Eq.(19) for an homogeneous quasi-incompressible fluid is presented in [45].

The stabilization parameter τ is computed in practice for each element e using $h = l^e$ and $\delta = \Delta t$ as

$$\tau = \left(\frac{8\mu}{(l^e)^2} + \frac{2\rho}{\Delta t} \right)^{-1} \tag{22}$$

where Δt is the time step used for the transient solution and l^e is a characteristic element length computed as $l^e = 2(V^e)^{1/n_s}$ where V^e is the element area (for 3-noded triangles) or volume (for 4-noded tetrahedra). For fluids with heterogeneous material properties the values of μ and ρ in Eq.(22) are computed at the element center.

7 Variational equations for the fluid

The variational form of the momentum and mass balance equations is obtained via the standard weighted residual approach [9, 60]. The resulting integral expressions after integration by parts and some algebra are:

Momentum equations

$$\int_V w_i \rho \frac{Dv_i}{Dt} dV + \int_V \left[\delta\varepsilon_{ij} 2\mu \varepsilon'_{ij} + \delta\varepsilon_v p \right] dV - \int_V w_i \left(b_i + \frac{1}{n_f} f_i^{pf} \right) dV - \int_{\Gamma_t} w_i t_i^p d\Gamma = 0 \tag{23}$$

Mass balance equation

$$\begin{aligned} \int_V \frac{q}{\kappa} \frac{Dp}{Dt} dV - \int_V q \left(\frac{1}{n_f} \frac{Dn_f}{Dt} + \varepsilon_v \right) dV + \int_V \tau \frac{\partial q}{\partial x_i} \left(\frac{\partial}{\partial x_i}(2\mu\varepsilon_{ij}) + \frac{\partial p}{\partial x_i} + b_i \right) dV \\ - \int_{\Gamma_t} q\tau \left[\rho \frac{Dv_n}{Dt} - \frac{2}{h_n}(2\mu \frac{\partial v_n}{\partial n} + p - t_n) \right] d\Gamma = 0 \end{aligned} \tag{24}$$

The derivation of Eqs.(23) and (24) for homogeneous fluids can be found in [45].

8 FEM discretization

We discretize the analysis domain containing N_p microscopic and macroscopic particles and a number of large particles into finite elements with n 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 ($n = 3$) for 2D problems and 4-noded tetrahedra ($n = 4$) for 3D problems with local linear shape functions N_i^e defined for each node i of element e [41, 58]. 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 global linear shape functions N_j spanning over the elements sharing node j ($j = 1, N$) [41, 58].

The finite element interpolation over the fluid domain can be written in matrix form as

$$\mathbf{v} = \mathbf{N}_v \bar{\mathbf{v}} \ , \ \mathbf{w} = \mathbf{N}_v \bar{\mathbf{w}} \ , \ p = \mathbf{N}_p \bar{\mathbf{p}} \ , \ q = \mathbf{N}_p \bar{\mathbf{q}} \tag{25}$$

where

$$\bar{\mathbf{v}} = \begin{Bmatrix} \bar{\mathbf{v}}^1 \\ \bar{\mathbf{v}}^2 \\ \vdots \\ \bar{\mathbf{v}}^N \end{Bmatrix} \quad \text{with } \bar{\mathbf{v}}^i = \begin{Bmatrix} \bar{v}_1^i \\ \bar{v}_2^i \\ \bar{v}_3^i \end{Bmatrix} \ , \ \bar{\mathbf{w}} = \begin{Bmatrix} \bar{\mathbf{w}}^1 \\ \bar{\mathbf{w}}^2 \\ \vdots \\ \bar{\mathbf{w}}^N \end{Bmatrix} \quad \text{with } \bar{\mathbf{w}}^i = \begin{Bmatrix} \bar{w}_1^i \\ \bar{w}_2^i \\ \bar{w}_3^i \end{Bmatrix} \ , \ \bar{\mathbf{p}} = \begin{Bmatrix} \bar{p}^1 \\ \bar{p}^2 \\ \vdots \\ \bar{p}^N \end{Bmatrix} \quad \text{and } \bar{\mathbf{q}} = \begin{Bmatrix} \bar{q}^1 \\ \bar{q}^2 \\ \vdots \\ \bar{q}^N \end{Bmatrix} \tag{26}$$

$$\mathbf{N}_v = [\mathbf{N}_1, \mathbf{N}_2, \cdots, \mathbf{N}_N]^T \ , \ \mathbf{N}_p = [N_1, N_2, \cdots, N_N]$$

with $\mathbf{N}_j = N_j \mathbf{I}_{n_s}$ where $\mathbf{I}_{n_s}$ is the $n_s \times n_s$ unit matrix.

In Eq.(26) vectors $\bar{\mathbf{v}}$, $(\bar{\mathbf{w}}, \bar{\mathbf{q}})$ and $\bar{\mathbf{p}}$ contain the nodal velocities, the nodal weighting functions and the nodal pressures for the whole mesh, respectively and the upperindex denotes the nodal value for each vector or scalar magnitude.

Substituting the approximation (25) into the variational forms (23) and (24) gives the system of algebraic equations for the particulate fluid in matrix form as

$$\mathbf{M}_0 \dot{\bar{\mathbf{v}}} + \mathbf{K}\bar{\mathbf{v}} + \mathbf{Q}\bar{\mathbf{p}} - \mathbf{f}_v = \mathbf{0} \tag{27a}$$

$$\mathbf{M}_1 \dot{\bar{\mathbf{p}}} - \mathbf{Q}^T \bar{\mathbf{v}} + (\mathbf{L} + \mathbf{M}_b)\bar{\mathbf{p}} - \mathbf{f}_p = \mathbf{0} \tag{27b}$$

The different matrices and vectors in Eqs.(27) are shown in Box 1 for 2D problems.

Remark 3. The boundary terms of vector $\mathbf{f}_p$ can be incorporated in the matrices of Eq.(27b). This, however, leads to a non symmetrical set of equations. For this reason we have chosen to compute these boundary terms iteratively within the incremental solution scheme.

Remark 4. Matrix $\mathbf{M}_b$ in Eq.(27b) allows us to compute the pressure without the need of prescribing its value at the free surface. This eliminates the error introduced when the pressure is prescribed to zero in free boundaries, which may lead to considerable mass losses [20, 45].

9 Incremental solution of the discretized equations

Eqs.(27) are solved in time with an implicit Newton-Raphson type iterative scheme [3, 9, 58, 60]. The basic steps within a time interval $[n, n+1]$ are:

- Initialize variables: $({}^{n+1}\mathbf{x}^1, {}^{n+1}\bar{\mathbf{v}}^1, {}^{n+1}\bar{\mathbf{p}}^1, {}^{n+1}n_f^i, {}^{n+1}\bar{\mathbf{r}}_m^1) \equiv \{{}^n\mathbf{x}, {}^n\bar{\mathbf{v}}, {}^n\bar{\mathbf{p}}, {}^n n_f, {}^n\bar{\mathbf{r}}_m\}$.
- Iteration loop: $k = 1, \cdots, NITER$.
  For each iteration.

$$\mathbf{M}^e_{0_{ij}} = \int_{V^e} \rho N^e_i N^e_j \mathbf{I}_2 dV \ , \ \mathbf{K}^e_{ij} = \int_{V^e} \mathbf{B}^{eT}_i \mathbf{D} \mathbf{B}^e_j dV \ , \ \mathbf{Q}^e_{ij} = \int_{V^e} \mathbf{B}^{eT}_i \mathbf{m} N^e_j dV$$

$$M^e_{1_{ij}} = \int_{V^e} \frac{1}{\kappa} N^e_i N^e_j dV \ , \ M^e_{b_{ij}} = \int_{\Gamma^e_t} \frac{2\tau}{h_n} N^e_i N^e_j d\Gamma$$

$$L^e_{ij} = \int_{V^e} \tau(\boldsymbol{\nabla}^T N^e_i)\boldsymbol{\nabla} N^e_j dV \ , \ \mathbf{f}^e_{v_i} = \int_{V^e} \mathbf{N}^e_i \left(\mathbf{b} + \frac{1}{n_f}\mathbf{f}^{pf}\right) dV + \int_{\Gamma^e_t} \mathbf{N}^e_i \mathbf{t}^p d\Gamma$$

$$f^e_{p_i} = \int_{\Gamma^e_t} \tau N^e_i \left[\rho \frac{Dv_n}{Dt} - \frac{2}{h_n}(2\mu \frac{\partial v_n}{\partial n} - t_n)\right] d\Gamma - \int_{V^e} \left(\tau \boldsymbol{\nabla}^T N^e_i \mathbf{b} - N^e_i \frac{1}{n_f}\frac{Dn_f}{Dt}\right) dV$$

with $i, j = 1, n$.

$$\mathbf{D} = 2\mu \begin{bmatrix} 2/3 & -1/3 & 0 \\ -1/3 & 2/3 & 0 \\ 0 & 0 & 1/2 \end{bmatrix} \ , \ \mathbf{B}^e_i = \begin{bmatrix} \frac{\partial N^e_i}{\partial x_1} & 0 \\ 0 & \frac{\partial N^e_i}{\partial x_2} \\ 0 & 0 \\ \frac{\partial N^e_i}{\partial x_2} & \frac{\partial N^e_i}{\partial x_1} \end{bmatrix}$$

$$\mathbf{N}^e_i = N^e_i \mathbf{I}_2 \quad \text{and} \quad \boldsymbol{\nabla} = \begin{Bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{Bmatrix} \ , \ \mathbf{m} = \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix}$$

N^e_i: Local shape function of node i of element e [41, 45]

Box 1. Element form of the matrices and vectors in Eqs.(27) for 2D problems

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

From Eq.(27a), we deduce

$$^{n+1}\mathbf{H}^i_v \Delta\bar{\mathbf{v}} = -^{n+1}\bar{\mathbf{r}}^k_m \rightarrow \Delta\bar{\mathbf{v}} \tag{28a}$$

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

$$\bar{\mathbf{r}}_m = \mathbf{M}_0\dot{\bar{\mathbf{v}}} + \mathbf{K}\bar{\mathbf{v}} + \mathbf{Q}\bar{\mathbf{p}} - \mathbf{f}_v \quad , \quad \mathbf{H}_v = \frac{1}{\Delta t}\mathbf{M}_0 + \mathbf{K} + \mathbf{K}_v \tag{28b}$$

with

$$\mathbf{K}^e_v = \int_{^nV^e} \mathbf{B}^T \mathbf{m} \theta \Delta t \kappa \mathbf{m}^T \mathbf{B} dV \tag{28c}$$

Step 2. Update the velocities

$$\text{Fluid nodes:} \ ^{n+1}\bar{\mathbf{v}}^{k+1} = {}^{n+1}\bar{\mathbf{v}}^k + \Delta\bar{\mathbf{v}} \tag{29a}$$

$$\text{Rigid particles:} \ \begin{cases} ^{n+1/2}\dot{\mathbf{u}}_j = {}^{n-1/2}\dot{\mathbf{u}}_j + {}^n\ddot{\mathbf{u}}^{k+1}_j \Delta t \\ \dot{\mathbf{u}}_j = \frac{1}{m_j} {}^n\mathbf{F}^{k+1}_j \quad , \quad j = 1, N_p \end{cases} \tag{29b}$$

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

From Eq.(27b) we obtain

$$^{n+1}\mathbf{H}^{i\,n+1}_p \bar{\mathbf{p}}^{k+1} = \frac{1}{\Delta t}\mathbf{M}_1 {}^{n+1}\bar{\mathbf{p}}^i + \mathbf{Q}^{T\,n+1}\bar{\mathbf{v}}^{k+1} + {}^{n+1}\bar{\mathbf{f}}^i_p \rightarrow {}^{n+1}\bar{\mathbf{p}}^{k+1} \tag{30a}$$

with

$$\mathbf{H}_p = \frac{1}{\Delta t}\mathbf{M}_1 + \mathbf{L} + \mathbf{M}_b \tag{30b}$$

Step 4. Update the coordinates of the fluid nodes and particles

$$\text{Fluid nodes: } {}^{n+1}\mathbf{x}_i^{k+1} = {}^{n+1}\mathbf{x}_i^{k} + \frac{1}{2}({}^{n+1}\bar{\mathbf{v}}_i^{k+1} + {}^{n}\bar{\mathbf{v}}_i)\Delta t \quad , \quad i = 1, N \tag{31a}$$

$$\text{Rigid particles: } \begin{cases} {}^{n+1}\mathbf{u}_i^{k+1} = {}^{n}\mathbf{u}_i^{k+1} + {}^{n+1/2}\dot{\mathbf{u}}_i^{k+1}\Delta t \\ {}^{n+1}\mathbf{x}_i^{k+1} = {}^{n}\mathbf{x}_i + {}^{n+1}\mathbf{u}_i^{k+1} \quad , \quad i = 1, N_p \end{cases} \tag{31b}$$

Step 5. Compute the fluid volume fractions for each node ${}^{n+1}n_{fi}^{k+1}$ *via Eq.(2)*

Step 6. Compute forces and torques on particles: ${}^{n+1}\mathbf{F}_i^{k+1}, {}^{n+1}\mathbf{T}_i^{k+1}$, $i = 1, N_p$

Step 7. 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 $\mathbf{f}_i^{fp}$ computed by Eq.(18)

Step 8. Check convergence

Verify the following conditions:

$$\begin{array}{l} \|{}^{n+1}\bar{\mathbf{v}}^{k+1} - {}^{n+1}\bar{\mathbf{v}}^{k}\| \leq e_v \|{}^{n}\bar{\mathbf{v}}\| \\ \|{}^{n+1}\bar{\mathbf{p}}^{k+1} - {}^{n+1}\bar{\mathbf{p}}^{k}\| \leq e_p \|{}^{n}\bar{\mathbf{p}}\| \end{array} \tag{32}$$

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 (32) 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 5. In Eqs.(28)–(32) ${}^{n+1}(\cdot)$ denotes the values of a matrix or a vector computed using the nodal unknowns at time $n+1$. In our work the derivatives and integrals in the iteration matrices $\mathbf{H}_v$ and $\mathbf{H}_p$ and the residual vector $\bar{\mathbf{r}}_m$ are computed on the *discretized geometry at time n* (i.e. $V^e = {}^{n}V^e$) while the nodal force vectors $\mathbf{f}_v$ and $\mathbf{f}_p$ are computed on the current configuration at time $n+1$. This is equivalent to using an *updated Lagrangian formulation* [3, 12, 44, 59].

Remark 6. The tangent "bulk" stiffness matrix $\mathbf{K}_v$ in the iteration matrix $\mathbf{H}_v$ of Eq.(28b) accounts for the changes of the pressure due to the velocity. Including matrix $\mathbf{K}_v$ in $\mathbf{H}_v$ has proven to be essential for the fast convergence, mass preservation and overall accuracy of the iterative solution [11, 45].

Remark 7. The parameter θ in $\mathbf{K}_v$ ($0 < \theta \leq 1$) has the role of preventing the ill-conditioning of the iteration matrix $\mathbf{H}_v$ for very large values of the speed of sound in the fluid that lead to a dominant role of the terms of the tangent bulk stiffness matrix $\mathbf{K}_v$. An adequate selection of θ also improves the overall accuracy of the numerical solution and the preservation of mass for large time steps. Details of the derivation of Eq.(28c) can be found in [45].

Remark 8. The iteration matrix $\mathbf{H}_v$ in Eq.(28a) is an *approximation of the exact tangent matrix* in the updated Lagrangian formulation for a quasi/fully incompressible fluid [44]. The simplified form of $\mathbf{H}_v$ used in this work has yielded very good results with convergence achieved for the nodal velocities and pressure in 3–4 iterations in all the problems analyzed.

Remark 9. 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, with l^e computed as explained in Section 6, $|{}^{n}\mathbf{v}|_{\max}$ 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 approaching a solid boundary [45].

10 About the particle finite element method (PFEM)

10.1 The basis of the PFEM

Let us consider a domain V containing fluid and solid subdomains. Each subdomain is characterized by a set of points, hereafter termed *virtual particles*. The virtual particles contain all the information for defining the geometry and the material and mechanical properties of the underlying subdomain. In the PFEM both subdomains are modelled using an *updated Lagrangian formulation* [3, 44, 59].

The solution steps within a time step in the PFEM are as follows:

1. The starting point at each time step is the cloud of points C in the fluid and solid domains. For instance ${}^{n}C$ denotes the cloud at time $t = {}^{n}t$ (Figure 5).
2. Identify the boundaries defining the analysis domain ${}^{n}V$, as well as the subdomains in the fluid and the solid. This is an essential step as some boundaries (such as the free surface in fluids) may be severely distorted during the solution, including separation and re-entering of nodes. The Alpha Shape method [10] is used for the boundary definition. Clearly, the accuracy in the reconstruction of the boundaries depends on the number of points in the vicinity of each boundary and on the Alpha Shape parameter. In the problems solved in this work the Alpha Shape method has been implementation as described in [17, 35].
3. Discretize the the analysis domain ${}^{n}V$ with a finite element mesh ${}^{n}M$.We use an efficient mesh generation scheme based on an enhanced Delaunay tesselation [17, 35].
4. Solve the Lagrangian equations of motion for the overall continuum using the standard FEM. Compute the state variables in at the next (updated) configuration for ${}^{n}t + \Delta t$: velocities, pressure and viscous stresses in the fluid and displacements, stresses and strains in the solid.
5. Move the mesh nodes to a new position ${}^{n+1}C$ where n+1 denotes the time ${}^{n}t + \Delta t$, in terms of the time increment size.
6. Go back to step 1 and repeat the solution for the next time step to obtain ${}^{n+2}C$.

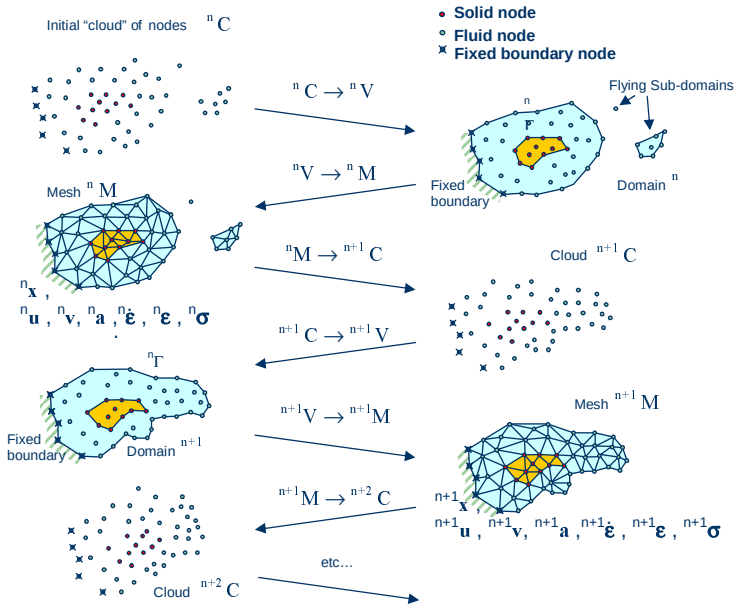


Fig. 5: Sequence of steps in the PFEM to update a "cloud" of nodes representing a domain containing a fluid and a solid part (in darker shadow) from time n ($t =^{n} t$) to time $n + 2$ ($t =^{n} t + 2\Delta t$)

Note that the key differences between the PFEM and the classical FEM are the remeshing technique and the identification of the domain boundary at each time step.

The CPU time required for meshing grows linearly with the number of nodes. As a general rule, meshing consumes for 3D problems around 15% of the total CPU time per time step [43].

Application of the PFEM in fluid and solid mechanics and in fluid-structure interaction problems can be found in [4, 5, 8, 11, 17–20, 25, 27, 28, 35, 36, 38, 40, 42–46], as well in www.cimne.com/pfem.

10.2 Treatment of contact conditions

Known velocities at boundaries in the PFEM are prescribed in strong form to the boundary nodes. These nodes might belong to fixed external boundaries or to moving boundaries linked to the interacting solids. Surface tractions are applied to the Neumann part of the boundary, as usual in the FEM.

Contact between fluid particles and fixed boundaries is accounted for by adjusting the time step so that fluid nodes do not penetrate into the solid boundaries [45].

The contact between two large particles (and between two bodies, in general) is treated by introducing a layer of *contact elements* between the two interacting particles. This contact interface layer is *automatically created during the mesh generation step* by prescribing a minimum distance (h_c) between two interacting particles. If the distance exceeds the minimum value (h_c) then the generated elements are treated as fluid elements. Otherwise the elements are treated as contact elements where a relationship between the tangential and normal forces and the corresponding displacement is introduced [35, 40, 43] (Figure 6).

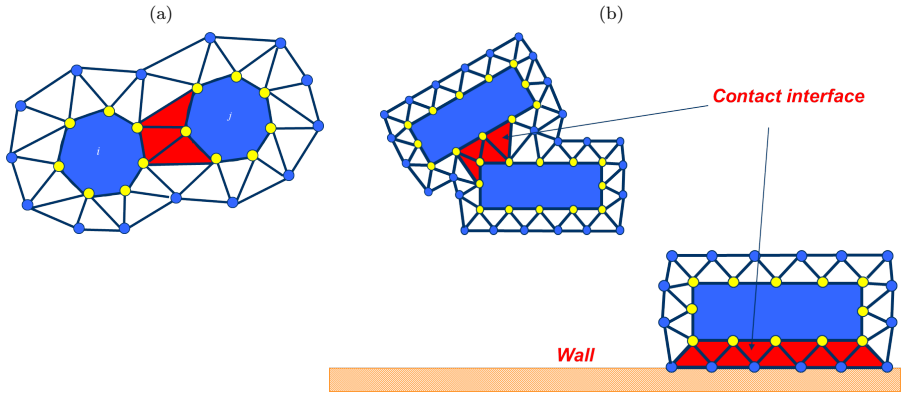


Fig. 6: (a) Large particles (in dark shadow) surrounded by a finite element mesh. The contact interface is shown in lighter shadow. (b) Contact interface between two objects treated as large particles and between an object and a wall

This algorithm allows us to model complex frictional contact conditions between two or more interacting bodies moving in water in an a simple manner. The algorithm has been used to model frictional contact situations between rigid and elastic solids in structural mechanics applications, such as soil/rock excavation problems [4, 5]. The frictional contact algorithm described above has been extended by Oliver *et al.* [27, 28] for analysis of metal cutting and machining problems.

Figure 7 shows an example of the analysis with the PFEM of a collection of large particles submerged in a tank containing water in sloshing motion.

10.3 Treatment of surface erosion

Prediction of bed erosion and sediment transport in open channel flows are important tasks in many areas of river and environmental engineering. Bed erosion can lead to instabilities of the river basin slopes. It can also undermine the foundation of bridge piles thereby favouring structural failure. Modeling of bed erosion is also

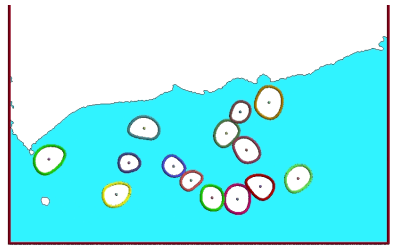


Fig. 7: PFEM results for the motion of large particles submerged in a tank containing water in sloshing motion

relevant for predicting the evolution of surface material dragged in earth dams in overspill situations. Bed erosion is one of the main causes of environmental damage in floods.

Oñate et al. [36] have proposed an extension of the PFEM to model bed erosion. The erosion model is based on the frictional work at the bed surface originated by the shear stresses in the fluid.

The algorithm for modeling the erosion of soil/rock particles at the fluid bed is briefly the following:

1. Compute at every point of the bed surface the tangential stress τ induced by the fluid motion.
2. Compute the frictional work W_f originated by the tangential stress at the bed surface.
3. The onset of erosion at a bed point occurs when ${}^nW_f$ exceeds a critical threshold value W_c.
4. If ${}^nW_f > W_c$ at a bed node, then the node is detached from the bed region and it is allowed to move with the fluid flow. As a consequence, the mass of the patch of bed elements surrounding the bed node vanishes in the bed domain and it is transferred to the adjacent fluid node. This mass is subsequently transported with the fluid as an immersed macroscopic particle.
5. Sediment deposition can be modeled by an inverse process to that described in the previous step. Hence, a suspended node adjacent to the bed surface with a velocity below a threshold value is attached to the bed surface.

Figure 8 shows an schematic view of the bed erosion algorithm described. Details of the algorithm can be found in [36].

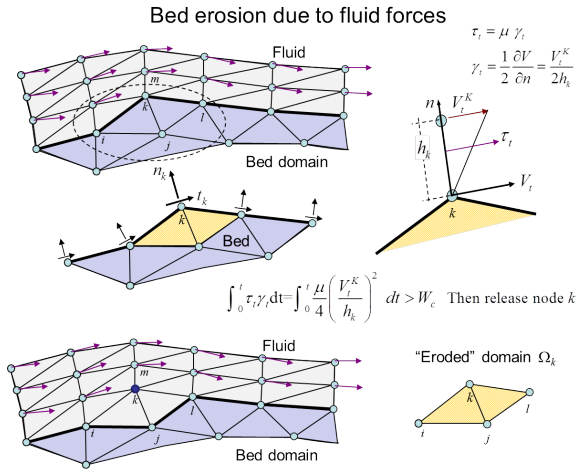


Fig. 8: Modeling of bed erosion with the PFEM. The mass of the eroded domain is assigned to the fluid node k

11 A nodal algorithm for transporting microscopic and macroscopic particles within a finite element mesh

The fact that in the PFEM a new mesh is regenerated at each time step allows us to make microscopic and macroscopic particles to be coincident with fluid nodes. An advantage of this procedure is that it provides an accurate definition of the particles at each time step which is useful for FSI problems.

The algorithm to compute the position of the particles using the nodal algorithm is as follows.

At each time step ${}^{n}t$:

1. Compute the converged value of the position of the fluid nodes (${}^{n+1}\mathbf{x}_i,\ i = 1, \cdots, N$) and the particles (${}^{n+1}\mathbf{x}_j,\ j = 1, \cdots, N_p$) using the algorithm of Section 9. The N_p particles coinciding with N_p fluid nodes ($N_p \leq N$) will typically move to a different position than that of the corresponding fluid nodes (Figure 9).
2. Regenerate the mesh discretizing the fluid domain treating the position of the N_p particles as fluid nodes and ignore the fluid nodes originally coinciding with the N_p particles at ${}^{n+1}t$.
3. Interpolate the velocity of the fluid nodes at the position of the N_p particles surrounding the fluid nodes.

The algorithm is schematically described in Figure 9.

Figures 10 show an example of the application of the nodal algorithm to the study of the motion of an individual particle within a rectangular domain filled with water. The correct end velocity for the individual particle is obtained as shown in Figure 10c. Figures 11–13 show other examples of application of the nodal algorithm to the motion of macro-particles in water containers.

Other examples of application of this algorithm are shown in the next section.

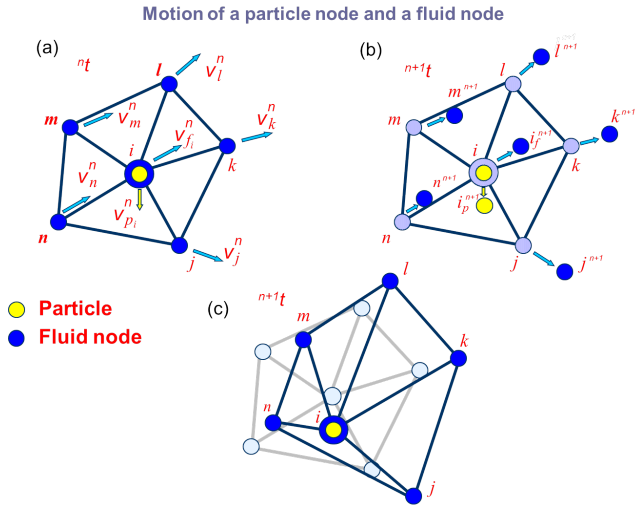


Fig. 9: Nodal algorithm for tracking the motion of particles submerged in a fluid. (a) Particle i is coincident with a fluid node. (b) Update the position of the particle and the adjacent nodes. (c) Regeneration of the fluid mesh consistent with the new particle position

12 Examples

We present the study of a several problems involving the transport of macroscopic and large particles in fluid flows. The problems are schematic representations of particulate flows occurring in practical problems of civil and environmental engineering and industrial problems.

Most problems studied have been solved with the PFEM using the nodal algorithm for the transport of macroscopic particles described in the previous section. An exception are the problems in Section 12.6 dealing

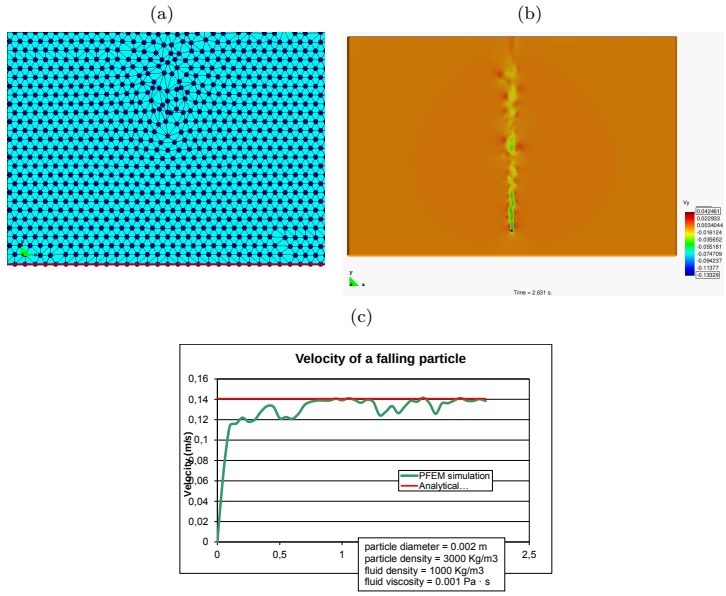


Fig. 10: Cylindrical particles falling in a water container. 2D PFEM solution using the nodal algorithm for tracking the particle motion. (a) Mesh and particle at a certain instant. (b) Contours of the vertical velocity module. (c) Evolution of the vertical velocity of the particle until a steady state solution is found [6, 15]

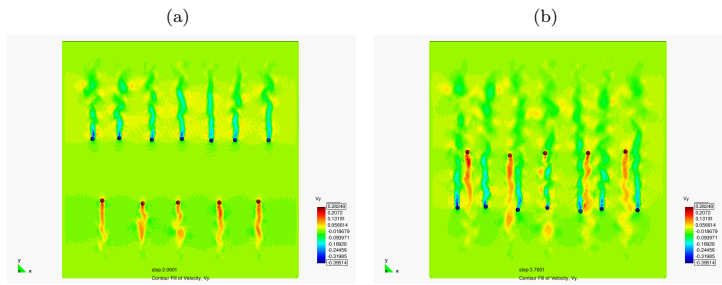


Fig. 11: Motion of ascending and descending particles of different density in a fluid domain. PFEM results using the nodal algorithm for tracking the particles motion

with the vertical transport of spherical particles in a cylinder where the standard immersed approach for the transport macroparticles described in Sections 1–4 was used and the fluid equations were solved with an Eulerian flow solver implemented in the Kratos open-source software platform of CIMNE [24].

12.1 Erosion of a slope adjacent to the shore due to sea waves

Figure 14 shows a representative example of the progressive erosion of a soil mass adjacent to the shore due to sea waves and the subsequent falling into the sea of a 2D object representing the section of a lorry. The object has been modeled as a rigid solid. Note that the eroded soil particles accumulate at the sea bottom.

This example, although still quite simple and schematic, evidences the possibility of the PFEM for modeling FSSI problems involving soil erosion, free surface waves and rigid/deformable structures.

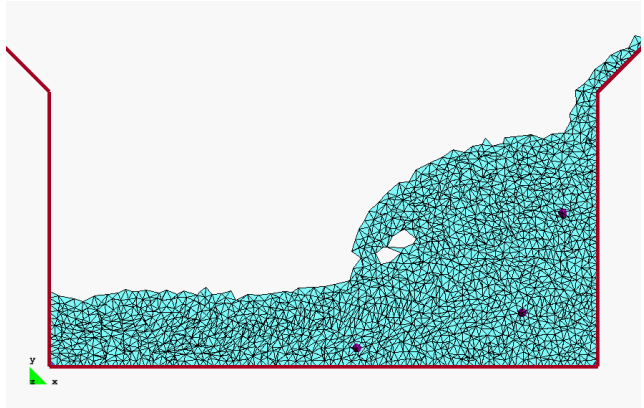


Fig. 12: Motion of three macroscopic particles in a water sloshing problem within a tank. PFEM results obtained using the nodal algorithm for particle tracking

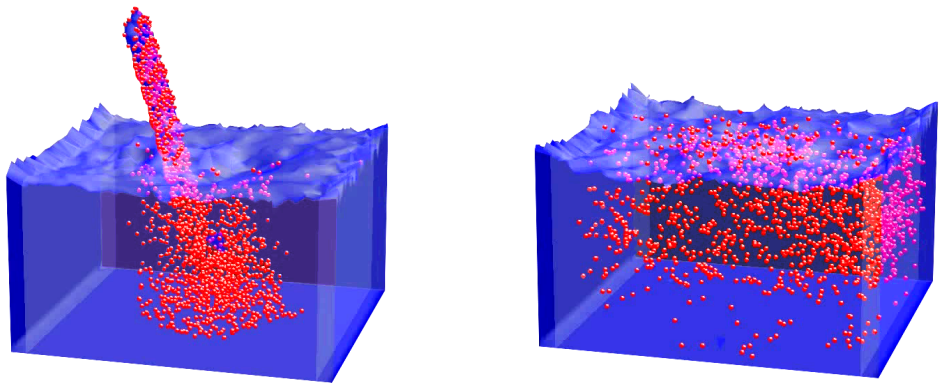


Fig. 13: PFEM analysis of the penetration of a collection of spherical (macroscopic) particles into a water container

12.2 Landslide falling on houses

Figure 15 shows two instants of the 2D simulation with PFEM of the motion of a collection of macroscopic particles as they slide over an inclined wall and fall into a water container.

The PFEM is particularly suited for the modelling and simulation of landslides and their effect in the surrounding structures. Figure 16 shows an schematic 2D simulation of a landslide falling on two adjacent constructions. The landslide material has been assumed to behave as a pure viscoplastic material modelled as a non-Newtonian viscous incompressible fluid [57]. Other applications of the PFEM to the modelling of landslides can be found in [8, 49].

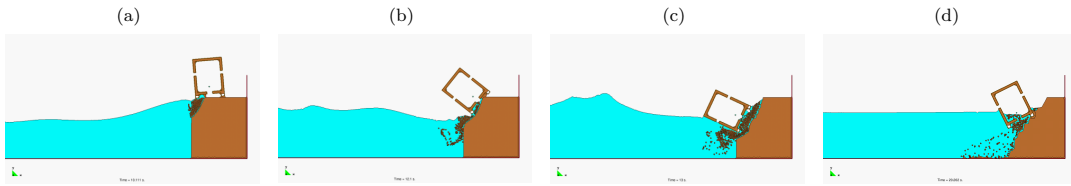


Fig. 14: Falling of a lorry into the sea by erosion of the underlying soil mass due to the action of waves

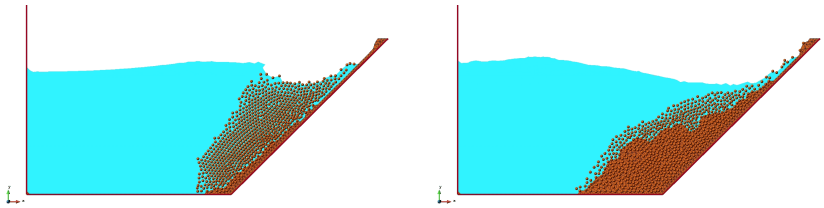


Fig. 15: Sliding of macroscopic particles over an inclined wall entering a container with water

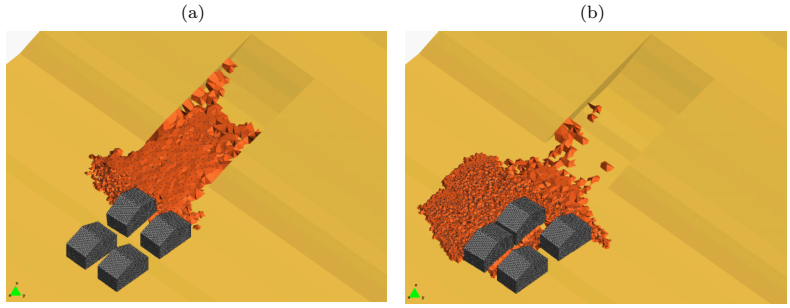


Fig. 16: 3D PFEM simulation of a landslide falling on four houses

12.3 Dragging of rocks by a water stream

Figure 17 shows the dragging of a collection of rocks modelled as large rigid particles of arbitrary shape by the action of a water stream. The particles move due to the action of the water forces on the particles computed by integrating the pressure and the viscous stresses in the elements surrounding each particle.

12.4 Suction of cohesive material submerged in water

Figures 18 and 19 show two examples of the detachment, suction and transport of particles of a cohesive material submerged on water. Figure 18 shows how the particles detatch from the cohesive soil bed and are transported within the suctioning tube (modelled as a 2D problem). The last picture shows the erosion in the soil as the mixture of water and eroded particles falls down from within the tube and hits the soil surface due to a stop in the suction pressure.

Figure 19 shows a similar 3D problem. The suction in the tube erodes the surface of the soil bed in the right hand container. The mixture of water and eroded particles is transported to the adjacent containers.

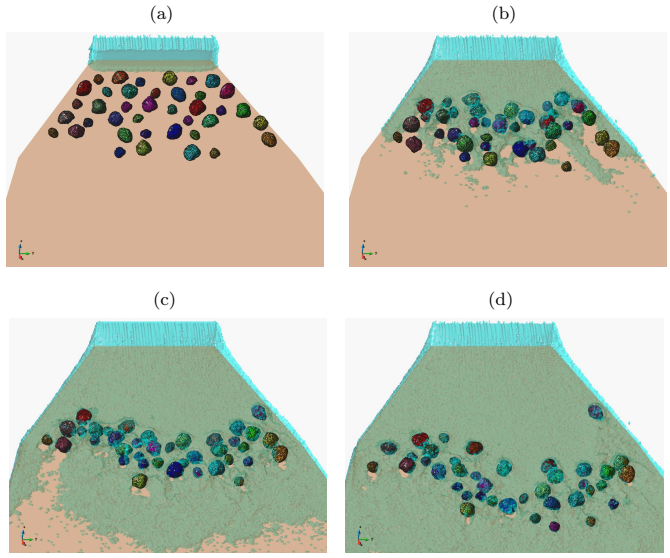


Fig. 17: 3D PFEM results for the dragging of a collection of large rocks by a water stream

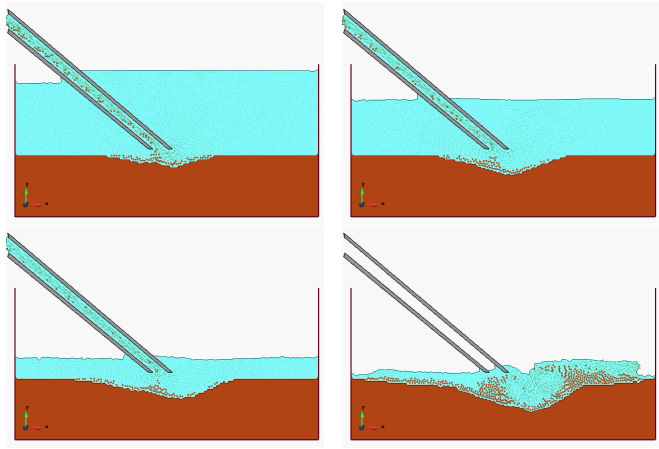


Fig. 18: 2D PFEM analysis of the detachment and suction of cohesive material submerged in water. The last picture shows the erosion of the bed material after the impact of the mixture of water and eroded particles falling from within the tube

12.5 Filling of a water container with particles

Figure 20 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.

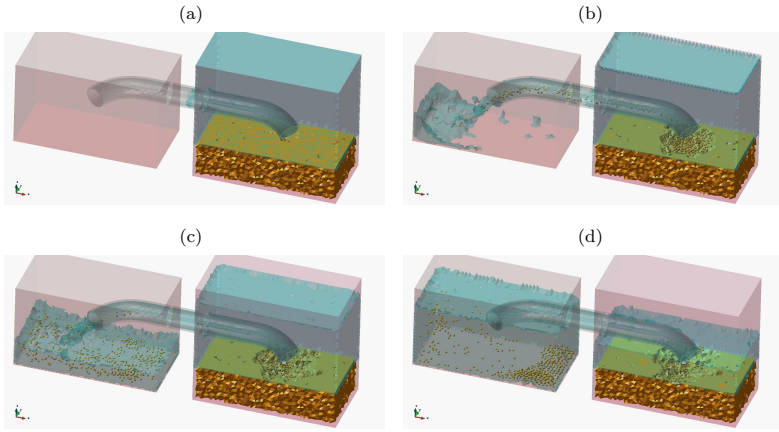


Fig. 19: 3D PFEM simulation of the detachment, suction and transport of submerged cohesive material from one recipient to another

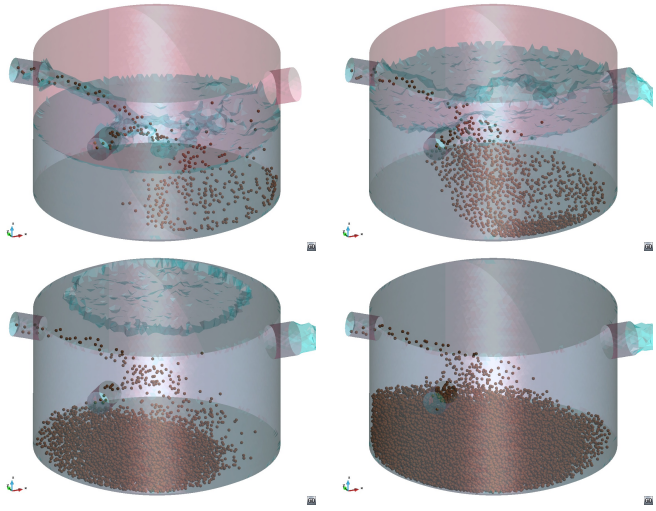


Fig. 20: Filling of a container by injecting water containing macroscopic particles from two holes. Water is allowed to exit through a third hole at the upper right hand side of the cylinder. 3D PFEM results at four instants

12.6 Transport of spherical particles in a tube filled with water

The example in Figure 21 models the vertical transport of some 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 fluid velocity of 1 m/s. The average size of the particles 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 effect of the interaction force from the fluid. Figure 21 shows two instants of the particles ascending process. The accumulation of particles in the water free surface at the top of the tube is clearly seen.

Figure 22 shows the interaction of eigth jets of ascending air bubbles with 200.000 spherical solid particles that fall down within a cylinder filled with water.

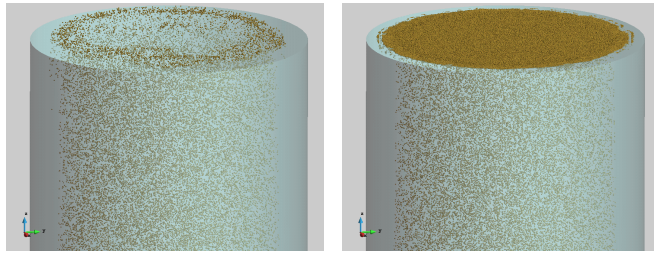


Fig. 21: Transport of spherical macroparticles up to the free surface of a tube filled with water. Particles move up with a prescribed velocity until they accumulate on the free surface. Results obtained with a coupled DEM-Eulerian CFD code [24]

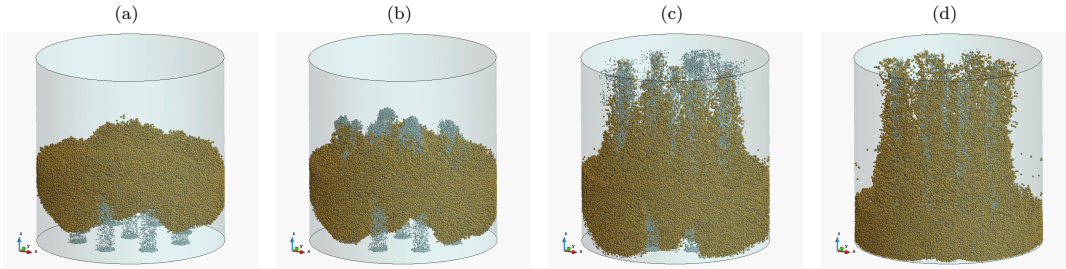


Fig. 22: Interaction of eight jets of ascending air bubbles with a thick layer of 200.000 spherical particles that fall down within a cylinder filled with water. Numerical results obtained with a coupled DEM-Eulerian CFD code [24]

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

The last example is the dragging of cars, barrels and debris (modelled as macroscopic particles) by a water stream that flows over a vertical wall. The problem represents an schematic study of a real situation corresponding to the tsunami in Fukushima (Japan) on March 2011 (Figure 23). Figures 24 show two snapshots of the PFEM solution of this complex problem.

Fig. 23: Dragging of cars and large and small objects in the Fukushima tsunami (Japan)

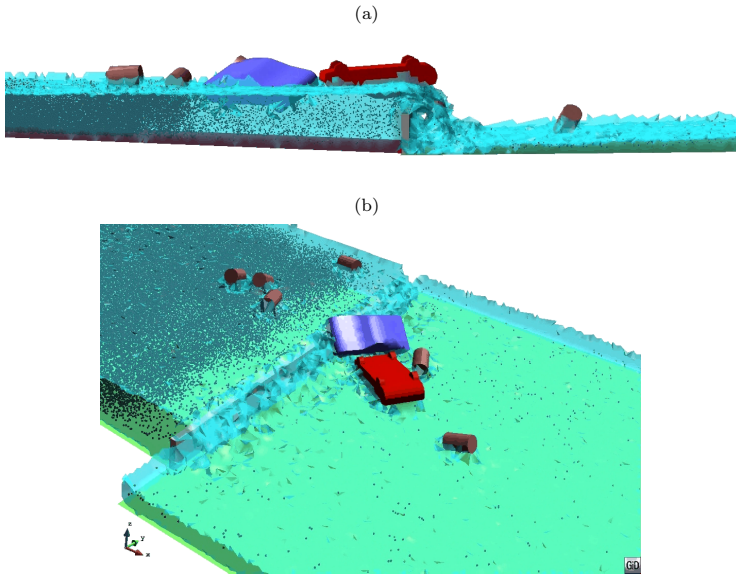


Fig. 24: Dragging of large objects and macroscopic particles in a tsunami type flow passing over a vertical wall. 3D PFEM results

13 Concluding remarks

We have presented a Lagrangian numerical technique for analysis of flows incorporating physical particles of different sizes. The numerical approach is based on the Particle Finite Element Method (PFEM) and a stabilized Lagrangian mixed velocity-pressure formulation. The examples presented in the paper evidence the possibilities of the PFEM for analysis of practical multiscale particulate flows in industrial and environmental problems.

Acknowledgements

This research was supported by the Advanced Grant project SAFECON of the European Research Council.

References

1. Anderson T, Jackson R. Fluid mechanical description of fluidized beds: equation of motion. Industrial & Engineering Chemical Fundamentals, 6(4):527–539
2. Avci B, Wriggers P, (2012) A DEM-FEM coupling approach for the direct numerical simulation of 3D particulate flows. Journal of Applied Mechanics, 79(1), 7 pages
3. T. Belytschko, W.K. Liu, B. Moran, Non linear finite element for continua and structures, 2d Edition, Wiley, 2013.
4. Carbonell JM, Oñate E, Suárez B (2010) Modeling of ground excavation with the Particle Finite Element Method. Journal of Engineering Mechanics (ASCE) 136(4):455–463
5. Carbonell JM, Oñate E, Suárez B (2013) Modelling of tunnelling processes and cutting tool wear with the Particle Finite Element Method (PFEM). Accepted in Comput. Mech. (2013) DOI:10.1007/s00466-013-0835-x
6. Clift R, Grace JR, Weber ME (1978) Bubbles, drops and particles. Academic Press, New York
7. Coussy O (2004) Poromechanics. Wiley
8. Cremonesi M, Frangi A, Perego U (2011) A Lagrangian finite element approach for the simulation of water-waves induced by landslides. Computers & Structures 89:1086–1093
9. Donea J, Huerta A (2003) Finite element method for flow problems. J. Wiley

10. Edelsbrunner H, Mucke EP (1999) Three dimensional alpha shapes. ACM Trans. Graphics 13:43–72
11. Franci A, Oñate E, Carbonell JM (2013) On the effect of the tangent bulk stiffness matrix in the analysis of free surface Lagrangian flows using PFEM. Research Report CIMNE PI402
12. Franci A, Oñate E, Carbonell JM (2014b) Unified updated Lagrangian formulation for fluid-structure interaction problems. Research Report CIMNE PI404
13. Gidaspow D (1994) Multiphase flow and fluidization. Continuum and Kinetic Theory Description, Academic Press, 467 pages
14. Healy DP, Young DB (2005) Full Lagrangian method for calculating particle concentration field in dilute gas-particle flows. Proc. Roy. Soc., London A: Mathematical, Physical and Engineering Sciences, 461(205a):2197–2225
15. Heider A, Levespiel O (1989) Drag coefficient and terminal velocity of spherical and nonspherical particles. Powder Technol. 58:63–70
16. Hilton J, Cleary P (2013) Dust modelling using a combined CFD and discrete element formulation. Int. J. Num. Meth. Fluids, 72(5):528-549
17. Idelsohn SR, Oñate E, Del Pin F (2004) The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. Int. Journal for Numerical Methods in Engineering, 61(7):964–989
18. Idelsohn SR, Marti J, Limache A, Oñate E (2008) Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluid-structure interaction problems via the PFEM. Comput Methods Appl Mech Engrg. 197:1762–1776
19. Idelsohn SR, Mier-Torrecilla M, Oñate E (2009) Multi-fluid flows with the Particle Finite Element Method. Comput Methods Appl Mech Engrg. 198:2750–2767
20. Idelsohn SR, Oñate E (2010) The challenge of mass conservation in the solution of free-surface flows with the fractional-step method: Problems and solutions. Int. J. Numer. Meth. Biomed. Engng. 26:1313-1330
21. Jajcevic D, Siegmann E, Radeke C, Khinast JG (2013) Large-scale CFD-DEM simulations of fluidized granular systems. Chemical Engineering Science, 98:298–310
22. Jackson R (2000), The dynamics of fluidized particles. Cambridge Monographs on Mechanics, Cambridge Univ. Press
23. Kafui DK, Thornton C, Adams MJ (2002) Discrete particle-continuum fluid modelling of gas-solid fluidised beds. Chemical Engng. Science, 57(13):2395–2410
24. Kratos (2014) Open source software platform for multiphysics computations. CIMNE, Barcelona, www.cimne.com/kratos
25. Larese A, Rossi R, Oñate E, Idelsohn SR (2008) Validation of the Particle Finite Element Method (PFEM) for simulation of free surface flows. Engineering Computations 25(4):385–425
26. Liu SH, Sun DA (2002). Simulating the collapse of unsaturated soil by DEM. Int. J. Num. Anal. Meth. in Geomechanics, 26:633646
27. Oliver X, Cante JC, Weyler R, González C, Hernández J (2007) Particle finite element methods in solid mechanics problems. In: Oñate E, Owen R (Eds) Computational Plasticity. Springer, Berlin, pp 87–103
28. Oliver X, Hartmann S, Cante JC, Wyler R, Hernández J (2009) A contact domain method for large deformation frictional contact problems. Part 1: theoretical basis. Comput Methods Appl Mech Eng 198:2591–2606
29. Oñate E (1998) Derivation of stabilized equations for advective-diffusive transport and fluid flow problems. Comput. Meth. Appl. Mech. Engng. 151:233–267
30. Oñate E (2000) A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. Comput Methods Appl Mech Engrg. 182(1–2):355–370
31. Oñate E, García J (2001) A finite element method for fluid-structure interaction with surface waves using a finite calculus formulation. Comput. Meth. Appl. Mech. Engrg. 191:635–660
32. Oñate E (2003) Multiscale computational analysis in mechanics using finite calculus: an introduction. Comput. Meth. Appl. Mech. Engrg. 192(28-30):3043–3059
33. Oñate E (2004) Possibilities of finite calculus in computational mechanics. Int. J. Num. Meth. Engng. 60(1):255–281
34. Oñate E, Rojek J, (2004b) Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems. Comput. Meth. Appl. Mech. Engrg. 193:3087–3128
35. Oñate E, Idelsohn SR, Del Pin F, Aubry R (2004c) The particle finite element method. An overview. Int. J. Comput. Methods 1(2):267–307

36. Oñate E, Celigueta MA, Idelsohn SR (2006a) Modeling bed erosion in free surface flows by the Particle Finite Element Method, Acta Geotechnia 1(4):237–252
37. Oñate E, Valls A, García J (2006b) FIC/FEM formulation with matrix stabilizing terms for incompressible flows at low and high Reynold's numbers. Computational Mechanics 38 (4-5):440–455
38. Oñate E, García J, Idelsohn SR, Del Pin F (2006c) FIC formulations for finite element analysis of incompressible flows. Eulerian, ALE and Lagrangian approaches. Comput. Meth. Appl. Mech. Engng. 195(23-24):3001–3037
39. Oñate E, Valls A, García J (2007) Computation of turbulent flows using a finite calculus-finite element formulation. Int. J. Numer. Meth. Engng. 54:609–637
40. Oñate E, Idelsohn SR, Celigueta MA, Rossi R (2008) 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
41. Oñate E (2009), Structural analysis with the finite element method. Linear statics. Volume 1. Basis and Solids. CIMNE-Springer
42. Oñate E, Rossi R, Idelsohn SR, Butler K (2010) Melting and spread of polymers in fire with the particle finite element method. Int. J. Numerical Methods in Engineering, 81(8):1046–1072
43. Oñate E, Celigueta MA, Idelsohn SR, Salazar F, Suárez B (2011) Possibilities of the particle finite element method for fluid-soil-structure interaction problems. Computational Mechanics, 48(3):307–318.
44. Oñate E, Carbonell JM (2013) Updated Lagrangian finite element formulation for quasi and fully incompressible fluids. Research Report PI393, CIMNE. Submitted to Comput. Mechanics
45. Oñate E, Franci A, Carbonell JM (2014) Lagrangian formulation for finite element analysis of quasi-incompressible fluids with reduced mass losses. Accepted for publication in Int. J. Num. Meth. in Fluids
46. Oñate E, Franci A, Carbonell JM (2014b) A Particle Finite Element Method (PFEM) for analysis of industrial forming processes. Accepted for publication in Comput. Mechanics
47. Patankar NA, Joseph DD (2001) Lagrangian numerical simulation of particulate flows. Int. J. Multiphase Flow, 27:1685–1706
48. Ryzhakov P, Oñate E, Rossi R, Idelsohn SR (2012) Improving mass conservation in simulation of incompressible flows. Int. J. Numer. Meth. Engng. 90(12):1435–1451
49. Salazar F, Oñate E, Morán R (2012) Modelación numérica del deslizamiento de ladeu en embalses mediante el método de partículos y elementos finitos (PFEM). Rev. Int. Meth. Num. Cal. Dis. Ing., 28(2):112–123
50. Shamy UE, Zeghal M (2005) Coupled continuum–discrete model for saturated granular soils. J. Engineering Mechanics (ASCE), 131(4):413-426
51. Sommerfeld M, van Wachen B, Oliemans R (Eds) (2008) Best practice guideliens for computational fluid dynamics of dispersed multiphase flows. European Research Community on Flow, Turbulence and Combustion (ERCOFTAC), Imperial College London.
52. Tang B, Li JF, Wang TS (2009) Some improvements on free surface simulation by the particle finite element method. Int. J. Num. Methods in Fluids, 60(9):1032-1054
53. van Wachen B, Oliveira ES (2010) An immersed boundary method for interacting particles. ERCOFTAC Bulletin 82, 17–22 March 2010
54. Wang X, Zhang LT, Liu WK (2009) On computational issues of the immersed finite element method. J. Comp. Physics, 228:2535–2551
55. Li X, Chu X, Sheng DC (2007) A saturated discrete particle model and characteristic-based SPH method in granular materials. Int. J. Numer. Meth. Engng., 72:858–882
56. Zhang LT, Gerstenberg A, Wang X, Liu WK (2004) Immersed finite element method, 193:2051–2067
57. Zienkiewicz OC, Jain PC, Oñate E (1978) Flow of solids during forming and extrusion: some aspects of numerical solutions. *Int. J. Solids Struct.*, 14:15–38
58. Zienkiewicz OC, Taylor RL, Zhu JZ (2005) The finite element method. The basis. 6th Ed., Elsevier
59. Zienkiewicz OC, Taylor RL (2005) The finite element method for solid and structural mechanics. 6th Ed., Elsevier
60. Zienkiewicz OC, Taylor RL, Nithiarasu P (2005) The finite element method for fluid dynamics. 6th Ed., Elsevier
61. Zohdi T (2007) An introduction to modelling and simulation of particulate flows. SIAM, Computational Science and Engineering
62. Zohdi T, Wriggers P (2007) Computation of strongly coupled multifield interaction in particle-fluid systems. Comput. Meth. Appl. Mech. Engng., 196:3927–3950

Paper 7

This paper, published in 2015, explains the details of the research about the transport of non-spherical particles in non-Newtonian fluids. The *PFlow* code was used for all the computations.

A FEM-DEM technique for studying the motion of particles in non-Newtonian fluids. Application to the transport of drill cuttings in wellbores

Miguel Angel Celigueta · Kedar M. Deshpande · Salvador Latorre · Eugenio Oñate

Abstract We present a procedure for coupling the finite element method (FEM) and the discrete element method (DEM) for analysis of the motion of particles in non-Newtonian fluids. Particles are assumed to be spherical and immersed in the fluid mesh. A new method for computing the drag force on the particles in a non-Newtonian fluid is presented. A drag force correction for non-spherical particles is proposed. The FEM-DEM coupling procedure is explained for Eulerian and Lagrangian flows and the basic expressions of the discretized solution algorithm are given. The usefulness of the FEM-DEM technique is demonstrated in its application to the transport of drill cuttings in wellbores.

Keywords FEM-DEM procedure · Motion of particles · Drill cuttings · Wellbores

1 INTRODUCTION

The cuttings transport process (hole-cleaning) is one challenging aspect associated with the efficiency of the wellbore drilling operations [1]. This process involves complex interactions between cuttings, drill pipe, wellbore and drilling mud [2]. The investigation of cuttings transport in a wellbore using advanced computational techniques for analysis of particulate non-Newtonian fluids can provide valuable insight for scientific and practical purposes. The focus of this work is to develop a procedure for coupling the finite element method (FEM) and the discrete element method (DEM) for analysis of the motion of particles in non-Newtonian incompressible fluids such as mud. The goal is to understand the cuttings behavior locally in critical sections of the wellbore during their transport by action of the drilling mud. Methods such as that presented here can provide useful information to estimate hole cleaning duration, find zones in danger of clogging or estimate the hole pressure gradient.

Particles within the fluid are modelled with the DEM. The coupling effects between the particles and the fluid are introduced via an immersed technique [3–5]. The fluid motion is modelled either with an Eulerian stabilized FEM formulation using a fixed mesh, or using a Lagrangian formulation using the Particle Finite Element Method (PFEM) [4, 6–16] for which the mesh evolves in time. For both the Eulerian and the Lagrangian formulations we use a mixed finite element formulation with an equal order linear interpolation for the velocities and the pressure variables.

In the Lagrangian PFEM the convective terms vanish in the momentum equations and no numerical stabilization is needed for these equations. Another source of instability, however, remains in the numerical solution of Lagrangian flows such as PFEM, that due to the treatment of the incompressibility constraint which still requires using a stabilized numerical method. In this work we use a PFEM formulation based on a residual-based stabilized expression of the mass balance equation [10–16]. The excellent mass preservation feature of this formulation has been demonstrated previously [7, 16].

M.A. Celigueta, S. Latorre, E. Oñate
Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE)
Campus Norte UPC, 08034 Barcelona, Spain
Tel.: +34-93-2057016
Fax: +34-93-4016517
E-mail: maceli,latorre,onate@cimne.upc.edu

K.M. Deshpande
Weatherford International Ltd.
E-mail: Kedar.Deshpande@Weatherford.com

The layout of the paper is the following. In the next section we present the basic equations for the conservation of linear momentum and mass for a quasi-incompressible particulate fluid in Eulerian and Lagrangian frameworks. The different force terms acting on the particles are explained. Details of the computation of the drag force for spherical and non spherical particles in non-Newtonian fluids are given. The finite element discretization is presented and the relevant matrices and vectors of the discretized problem are given. The transient solution of the FEM-DEM equations for a particulate flow using a Newton-Raphson type iterative scheme for solving the fluid equations is presented.

The efficiency and accuracy of the FEM-DEM procedure for analysis of particulate flows in non-Newtonian fluids are verified by solving a number of drilling transport problems in wellbores.

2 DESCRIPTION OF THE FEM-DEM ALGORITHM

2.1 Modelling of the particles

Figure 1 shows a fluid domain containing particles of small and moderate sizes relative to the representative volume for a node. Particles are assumed to have a spherical shape in two and three-dimensions (2D/3D) and are modelled as rigid objects that undergo interaction forces due to the physical contact between a particle and its neighbors, as in the standard DEM [17–20].

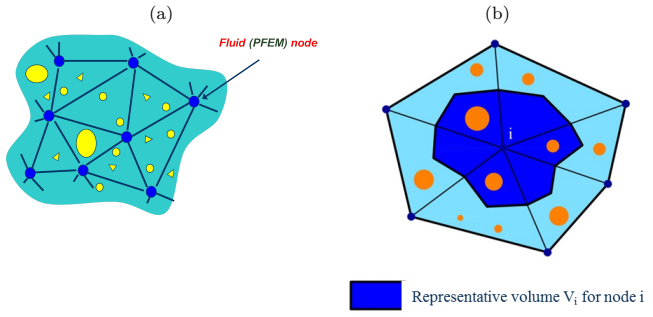


Fig. 1: (a) Particles of different sizes surrounding the nodes in a FEM mesh. (b) Representative volume for a node (in shadowed darker colour)

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 equations (Figure 2). These forces, as well as the mass balance equations account for the percentage of particles in the fluid, similarly as it is done in immersed approaches for particulate flows [4, 5, 17]. The rest of the interaction forces between fluid and particles are neglected (lift forces, virtual mass forces, drag torque, etc.) [21, 22].

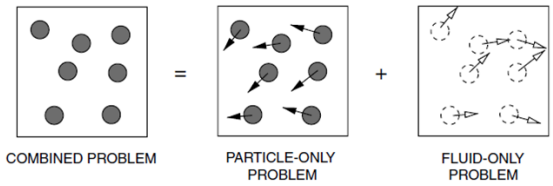


Fig. 2: Immersed approach for treating the motion of physical particles in a fluid [5]

The following sections describe the governing equations for a particulate fluid and the computation of the forces on the particles.

2.2 Basic governing equations for a particulate fluid

2.2.1 Conservation of linear momentum

The equation for linear momentum conservation can be expressed as

$$r_{m_i} = 0 \tag{1a}$$

with

$$r_{m_i} := \rho_f \frac{Dv_i}{Dt} - \frac{\partial \sigma_{ij}}{\partial x_j} - \left(b_i - \frac{1}{n_f} f_i^{pf} \right) \quad , \quad i,j = 1, \cdots, n_s \quad \text{in } V \tag{1b}$$

Summation of terms with repeated indices is assumed in Eq.(1) and in the following, unless otherwise specified.

In Eq.(1) V is the analysis domain, n_s is the number of space dimensions ($n_s = 3$ for 3D problems), ρ_f is the density of the fluid, v_i and b_i are the velocity and body force components along the ith Cartesian axis, respectively, σ_{ij} are the fluid Cauchy stresses, f_i^{pf} are averaged particle-to-fluid interaction forces for which closure relations must be provided and n_f is the fluid volume fraction at a point defined for each node j as

$$n_{f_j} = 1 - \frac{1}{V_j} \sum_{i=1}^{n_j} V_j^i \tag{2}$$

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. Note that $n_{f_j} = 1$ for a representative fluid domain containing no particles (Figure 1).

The fluid volume fraction n_f in Eq.(1) is a continuous function that is interpolated from the nodal values in the finite element fashion [14, 23, 24].

Remark 1. The time derivative $\frac{Dv_i}{Dt}$ in Eq.(1) is computed in the Eulerian and Lagrangian frameworks as

$$\textit{Eulerian}: \qquad \frac{Dv_i}{Dt} = \frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \tag{3a}$$

$$\textit{Lagrangian}: \qquad \frac{Dv_i}{Dt} = \frac{\partial v_i}{\partial t} = \frac{{}^{n+1}v_i - {}^{n}v_i}{\Delta t} \tag{3b}$$

with

$${}^{n+1}v_i := v_i({}^{n+1}\mathbf{x}, {}^{n+1}t) \quad , \quad {}^{n}v_i := v_i({}^{n}\mathbf{x}, {}^{n}t) \tag{3c}$$

In Eq.(3c) ${}^{n}v_i({}^{n}\mathbf{x}, {}^{n}t)$ is the velocity of the material point that has the position ${}^{n}\mathbf{x}$ at time $t = {}^{n}t$, where $\mathbf{x} = [x_1, x_2, x_3]^T$ is the coordinates vector of a point in a fixed Cartesian system. Note that the convective term, typical of the Eulerian formulation, does not appear in the definition of the material derivative in Eq.(3b) [24–26].

2.2.2 Constitutive equations

The Cauchy stresses in the fluid, σ_{ij}, are split into the deviatoric (s_{ij}) and pressure (p) components as

$$\sigma_{ij} = s_{ij} + p\delta_{ij} \tag{4}$$

where δ_{ij} is the Kronecker delta. In this work the pressure is assumed to be positive for a tension state.

The relationship between the deviatoric stresses and the strain rates has the standard form for a Newtonian fluid [24, 26],

$$s_{ij} = 2\mu\varepsilon'_{ij} \quad \text{with } \varepsilon'_{ij} = \varepsilon_{ij} - \frac{1}{3}\varepsilon_v \delta_{ij} \quad \text{and} \quad \varepsilon_v = \varepsilon_{ii} \tag{5}$$

In Eq.(5), μ is the viscosity and ε'_{ij} and ε_v are the deviatoric and volumetric strain rates, respectively. The strain rates, ε_{ij}, are related to the velocities by

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \tag{6}$$

For the non-Newtonian fluids considered in this work the viscosity dependence with the strain rate is defined as

$$\mu(\gamma) = \frac{\tau(\gamma)}{\gamma} \tag{7}$$

where the expression of the shear stress, $\tau(\gamma)$, is obtained experimentally from a viscometer test. For multidimensional flows, γ is defined as the second invariant of the symmetric gradient of the velocity field, i.e.

$$\gamma = \sqrt{\frac{1}{2}\varepsilon_{ij}\varepsilon_{ij}} \tag{8}$$

2.2.3 Mass conservation equation

The mass conservation equation for a particulate flow is written as

$$r_v = 0 \tag{9a}$$

with

$$r_v := \frac{D(n_f \rho_f)}{Dt} + n_f \rho_f \varepsilon_v \tag{9b}$$

Expanding the term and dividing Eq.(9a) by $n_f \rho_f$, the expression of r_v can be redefined as

$$r_v := -\frac{1}{\kappa}\frac{Dp}{Dt} + \frac{1}{n_f}\frac{Dn_f}{Dt} + \varepsilon_v \tag{10}$$

where κ ($\kappa = \rho_f c^2$) is the bulk compressibility parameter, c being the speed of sound and $c^2 = -\frac{\partial p}{\partial \rho}$.

Remark 2. 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 [24, 26].

Remark 3. Similarly as in Eqs.(2.2.1) the time derivative term in Eqs.(9b) and (10) has different forms in Eulerian and Lagrangian frameworks, i.e.

$$\textit{Eulerian}: \qquad \frac{D(\cdot)}{Dt} = \frac{\partial(\cdot)}{\partial t} + v_j \frac{\partial(\cdot)}{\partial x_j} \tag{11}$$

$$\textit{Lagrangian}: \qquad \frac{D(\cdot)}{Dt} = \frac{\partial(\cdot)}{\partial t} = \frac{{}^{n+1}(\cdot) - {}^{n}(\cdot)}{\Delta t} \tag{12}$$

with ${}^{n}(\cdot)$ having the same meaning as in Eq.(3b).

2.2.4 Boundary conditions

The boundary conditions at the Dirichlet (Γ_v) and Neumann (Γ_t) boundaries with the fluid boundary $\Gamma = \Gamma_v \cup \Gamma_t$ are

$$v_i - v_i^p = 0 \qquad \text{on } \Gamma_v \tag{13a}$$

$$\sigma_{ij} n_j - t_i^p = 0 \qquad \text{on } \Gamma_t \quad i,j = 1, \cdots, n_s \tag{13b}$$

where v_i^p and t_i^p are the prescribed velocities and prescribed tractions at the Γ_v and Γ_t boundaries, respectively and n_j are the components of the unit normal vector to the boundary [24–26].

At a free surface the Neumann boundary conditions (Eq.(13b)) apply. These conditions are enforced at every time step.

2.3 Motion of particles

The motion of particles follows the standard law for Lagrangian particles. For the ith particle

$$m_i \dot{\mathbf{u}}_i = \mathbf{F}_i \quad , \quad J_i \dot{\mathbf{w}}_i = \mathbf{T}_i \tag{14}$$

where $\mathbf{u}_i$ and $\mathbf{w}_i$ are the velocity and the rotation vector of the center of gravity of the particle, m_i and J_i are the mass and rotational inertia of the particle, respectively and $\mathbf{F}_i$ and $\mathbf{T}_i$ are the vectors containing the forces and torques acting at the gravity center of the particle [19, 20].

Eqs.(14) are integrated in time in order to compute the motion of the particles. An explicit Forward Euler scheme has been used with substeppping. The substepping is necessary to avoid instabilities in the DEM solution since the fluid time steps are usually too large for the DEM solver. The information of the fluid which is projected to the particles is interpolated linearly between two steps of the fluid solution to the substep of the DEM solution.

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

$$\mathbf{F}_i = \mathbf{F}_i^w + \mathbf{F}_i^c + \mathbf{F}_i^{fp} \tag{15}$$

$\mathbf{F}_i^w$, $\mathbf{F}_i^c$ 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.

2.3.1 Self-weight forces

The self-weight force acting on a particle can be written as

$$\mathbf{F}_i^w = -\rho_i \Omega_i \mathbf{g} \tag{16}$$

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.

2.3.2 Contact forces

The contact forces acting on a particle coming from other particles and walls are summed as follows:

$$\mathbf{F}_i^c = \sum_{j=1}^{n_i} \mathbf{F}_{ij}^c \tag{17}$$

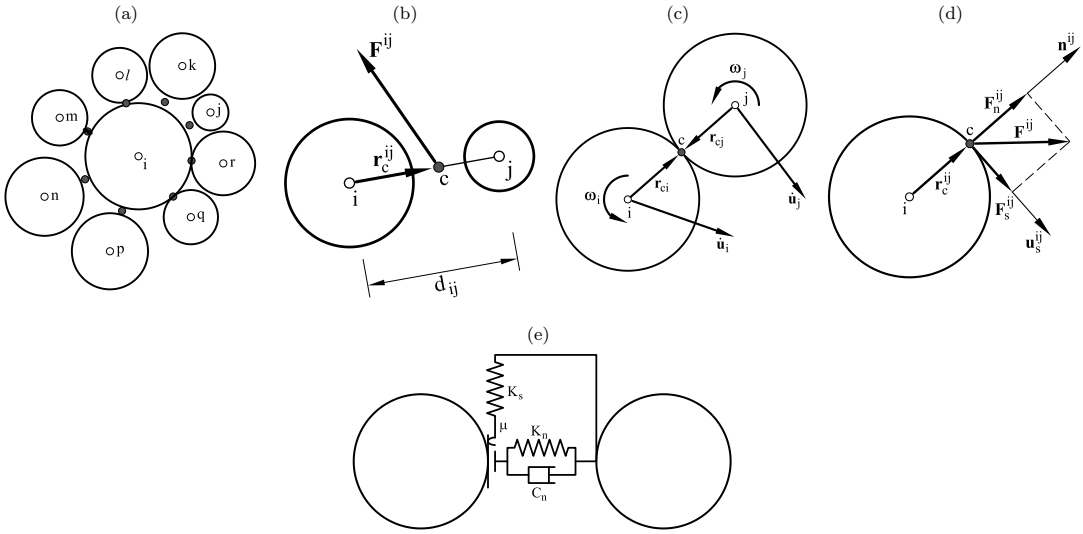
where n_i is the number of contact interfaces for the ith particle.

$$\mathbf{F}_{ij}^c = \mathbf{F}_n^{ij} + \mathbf{F}_s^{ij} = F_n^{ij} n_i + \mathbf{F}_s^{ij} \tag{18}$$

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 (Figure 3) or particle i with a wall. These forces are computed in terms of the relative motion of the interacting particles as in the standard DEM [17–20]. Figure 3 summarizes some aspects of the DEM. Figure 3a depicts the particle i with 8 neighbor particles (j, k, l, m, n, p, q and r). Figure 3b shows details of the contact between particles i and j: d_{ij} is the distance between centers, r_c^{ij} is the vector from the center of the particle i to the contact point between i and j, F^{ij} is the force exerted by particle j on particle i at the contact point. Figure 3c shows the scheme of the kinematics of the contact. Both particles have a velocity ($\dot{\mathbf{u}}_i$, $\dot{\mathbf{u}}_j$) and an angular velocity (w_i, w_j). The relative displacement of the particles at the contact point is penalized with elastic constants to avoid interpenetration between particles. Figure 3d shows the decomposition of the relative displacement and the contact force in the normal and tangential directions at the contact point. Figure 3e shows the linear elastic dashpot system used for modeling the mechanical behaviour at a contact point. The elastic penalty constants are K_n (normal direction) and K_s (tangential direction); C_n is a viscous parameter that provides damping to the contact; μ is the Coulomb's friction parameter. It affects the limit at which sliding between particles occurs, which follows the expression $\mathbf{F}_s^{ij} \leq \mu \mathbf{F}_n^{ij}$.

2.3.3 Fluid-to-particle forces

The interaction force between the fluid and a particle is written as $\mathbf{F}_i^{fp} = \mathbf{F}_i^d + \mathbf{F}_i^b$, 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:



$\mathbf{F}_n^{ij} = K_n \mathbf{u}_n^{ij} + C_n \dot{\mathbf{u}}_n^{ij}$
$\mathbf{F}_s^{ij} = K_s \mathbf{u}_s^{ij}$

$\mathbf{u}_n^{ij}, \mathbf{u}_s^{ij}$ relative displacements in the normal and tangential directions to a contact interface

Fig. 3: (a) Group of eight particles in contact with the ith particles. (b) Contact force vector for two particles interacting with a gap distance. (c) Distance vectors and velocity vector for two particle in contact. (d) Normal and tangential forces and displacement at a contact point. (e) Linear elastic dash pot system modeling the mechanical behaviour at a contact point [20]

2.3.4 Buoyancy forces

The buoyancy force depends on the volume of the particle and the gradient of pressure of the fluid:

$$\mathbf{F}_i^b = -\Omega_i \boldsymbol{\nabla} p \tag{19}$$

Note that $\boldsymbol{\nabla} p$ is not always aligned with gravity, but depends on the fluid dynamics.

2.3.5 Drag forces

The drag force is defined as

$$\mathbf{F}_i^d = -F_i^d \hat{\mathbf{v}}_i^r \tag{20}$$

where

$$\hat{\mathbf{v}}_i^r = \frac{\mathbf{v}_i^r}{|\mathbf{v}_i^r|} \quad \text{with} \quad \mathbf{v}_i^r = \mathbf{u}_i - \mathbf{v}_i \tag{21}$$

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.

2.3.6 Computation of f_i^{pf}

The force term component f_i^{pf} in the right hand side of the momentum equations (Eq.(1)) is computed for each particle (in vector form) as $\mathbf{f}^{pf} = -\mathbf{f}^{fp}$ with vector $\mathbf{f}^{fp}$ computed *at each node* in the fluid mesh from the drag forces $\mathbf{F}_i^d$ as

$$\mathbf{f}_j^{fp} = \frac{1}{V_j} \sum_{i=1}^{n_j} N_j(x_i) \mathbf{F}_i^d \quad , \quad j = 1, N \tag{22}$$

where $N_j(x_i)$ is the value of the shape function of node j at the position of the ith particle.

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 the present analysis. These forces can be accounted for as explained in [27].

2.4 Computation of the drag force $\mathbf{F}_i^d$ for non-Newtonian fluids

The drag forces on particles immersed in a Newtonian fluid [28] are well known. However, when dealing with non-Newtonian fluids different approaches for computing these forces can be followed depending on the type of fluid. Non-Newtonian fluids, for example, can be shear thickening, shear thinning or Bingham plastics, and each one of these requires a different drag law. Most drag laws require finding a suitable value for the drag coefficient (C_d) and the Reynold's number.

Drilling fluids for the oil-drilling industry usually exhibit a Herschel-Bulkley behavior [1]. Drag laws for particles moving in drilling fluids based on a shear thickening behavior can be discarded. On the other hand, drag laws developed under the assumptions of a Bingham plastic [29–31] fail to predict accurately the drag force when they are applied to Herschel-Bulkley fluids. In fact, no drag laws for particles in Herschel-Bulkley fluids, or in fluids characterized by a power law rheogram are found in the literature.

Despite the lack of suitable drag laws for particles in Herschel-Bulkley fluids, some accurate estimations of the terminal velocity of the particle (i.e. the steady state velocity reached by a particle falling freely in a liquid) can be found in several papers for different fluids [21, 32]. In particular, Shah [33] proposed an estimation of the terminal velocity in power law fluids characterized by the following value of the (non linear) viscosity

$$\mu(\gamma) = K\gamma^{n-1} \tag{23}$$

where γ was defined in Eq.(8), and K and n are material parameters. Shah's method has proven to give good estimations of the terminal velocity of particles falling in drilling muds in accordance to published experiments [31] (see Figure 4). The terminal velocity will be used later on in this section to generate a drag law.

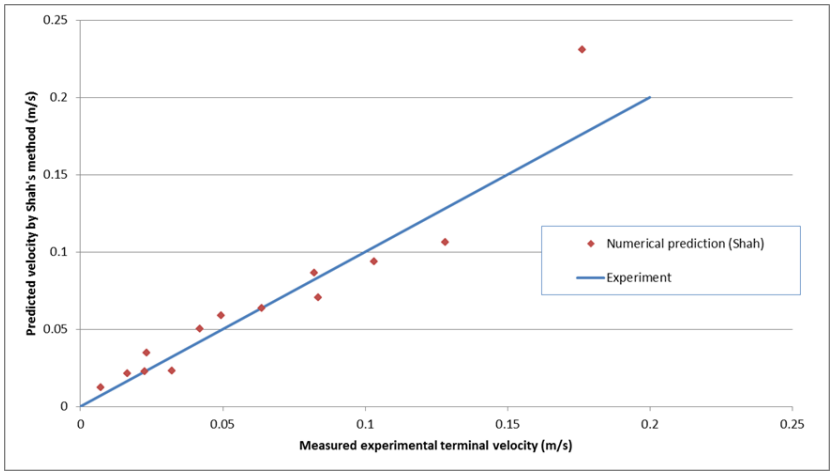


Fig. 4: Relation between the experimental terminal velocity and the values predicted by Shah's method [33]. The straight line indicates the exact correlation between experimental and predicted values [31]

2.4.1 Computation of the terminal velocity of the particles

In the following lines we summarize the steps for estimating the terminal velocity of the particles using Shah's method.

1. Compute

$$A = \ 6.9148(n^2) \ - 24.838(n) + 22.642 \ \text{ and } \ B = \ - 0.5067(n^2) + 1.3234(n) - 0.1744 \tag{24}$$

where n is the power exponent in Eq.(23).
2. Obtain Shah's dimensionless parameter S_i for the ith particle as

$$S_i = (C_d^t)^{\frac{2-n}{2}} Re_t^i \tag{25}$$

where the drag coefficient for the ith particle is defined as

$$C_d^i = \frac{4}{3}\left(\frac{d_p^i g}{(v_t^i)^2}\right)\left(\frac{\rho_p^i - \rho_f}{\rho_f}\right) \tag{26}$$

with

$$Re_t^i = \frac{(d_p^i)^n (v_t^i)^{2-n} \rho_f}{2^{n-1} K} \tag{27}$$

being the Reynolds number at the terminal velocity for power law fluids (other authors such as Walker *et al.* [34, 35] give other definition for Re). In Eq.(26) g is the gravity acceleration, d_i is the particle diameter and K is the parameter in Eq.(23).
Substituting Eqs.(26) and (27) into (25) Shah's dimensionless parameter can be computed as

$$S_i = \frac{1}{2^{n-1}K}\sqrt{\left(\frac{4}{3}\right)^{2-n} (d_p^i)^{n+2} \rho_f^n (\rho_p^i - \rho_f)^{2-n} g^{2-n}} \tag{28}$$

Note that Eq.(28) does not depend on v_t^i.
3. Obtain the Reynolds number for the ith particle from

$$Re_t^i = \left(\frac{S_i}{A}\right)^{\frac{1}{B}} \tag{29}$$

4. The terminal velocity for the ith particle v_t^i is computed from Eq.(27) as

$$v_t^i = \left[\frac{2^{n-1} K Re_t^i}{(d_p^i)^n \rho_f}\right]^{\frac{1}{(2-n)}} \tag{30}$$

2.4.2 Computation of the drag force

In this work a linear drag force law is proposed for cuttings moving in drilling muds of the type

$$F_i^d(v_i^r) = a_i s_i + \frac{W_i - F_i^b - a_i s_i}{v_t^i} v_i^r \tag{31}$$

where a_i is a parameter that is a function of the gel strength of the fluid and its dynamics [1], s_i is the surface area of the particle, $W_i = |\mathbf{F}_i^w|$ is the weight of the particle, F_i^b the modulus of the buoyancy force vector ($F_i^b = |\mathbf{F}_i^b|$), v_i^r is the relative velocity of the particle with respect to to the fluid ($v_i^r = |\mathbf{v}_i^r|$, see Eq.(23)) and v_t^i is the relative terminal velocity obtained by Shah's method via Eq.(30). Note that for $v_i^r = v_t^i$ the equilibrium of forces for the terminal slip velocity is recovered (i.e. $F_i^d = W_i - F_i^b$). On the other hand, for $v_i^r = 0$ the initial force (gel strength) is recovered.

The gel strength is the maximum stress the fluid can withstand before showing some measurable shear rate. For those cases when $v_i^r = 0$ the drag law presents a singularity and its derivative is infinite. This means that the force can be any value between 0 and $a_i\, s_i$. To solve this problem, the drag force law must be regularized. In this work we have modified the drag force as

$$\bar{F}_i^d(v_i^r) = \frac{F_i^d(\hat{v}_i^r)}{\hat{v}_i^r} v_i^r \quad \text{for } v_i^r \in [0, \hat{v}_i^r] \tag{32}$$

where $\hat{v}_i^r$ is a very small value.

Note that a_i is equal to the gel strength for every part of the fluid where the equivalent shear rate is zero. However, in those parts of the fluid where the shear rate is not zero, the apparent viscosity perceived by the particle decreases. In this situation, the gel strength has already been overcome and a_i must be set to zero.

2.4.3 Accounting for the effect of the non sphericity of the particles

The sphericity coefficient (Ψ) is defined as the ratio between the skin surface of the non-spherical particle and the skin surface of a sphere with the same volume. For the same relative velocity of the particle with respect to the fluid, the drag force grows as the sphericity decreases. Although some authors use the sphericity coefficient within a range between 0.125 and 1.0 [1] it must be pointed out that very low values of Ψ correspond actually to shapes that are far from spherical (plates, flat discs, thin sticks, ...). In those cases, the correction that Ψ brings to the equations that assume spherical shape can be very wrong. The authors do not recommended using Ψ below 0.65.

Different authors have established the dependency of the drag coefficient with the Reynolds number and the sphericity parameter for Newtonian fluids. For instance, Chien [32] proposed

$$C_d = \frac{30}{Re} + 67.289e^{(-5.03\Psi)} \tag{33}$$

while Haider and Levenspiel [36] propose:

$$C_d = \frac{24}{Re}\left[1 + \exp(2.3288 - 6.4581\Psi + 2.4486\Psi^2)Re^{(0.0964+0.5565\Psi)}\right] + \frac{73.69Re \cdot \exp(-5.0748\Psi)}{Re + 5.378\exp(6.2122\Psi)} \tag{34}$$

Both expressions and others mentioned in [31] can be used to obtain the drag coefficient for spherical particles simply making $\Psi = 1$.

In order to include the effect of the non sphericity of the particles in Shah's method, the dimensionless parameter S_i (Eq.(25)) has to be modified. This can be done by substituting the drag coefficient by an equivalent one which accounts for the sphericity effect. For that purpose, the following parameter is used:

$$\varphi^i = \frac{C^i_{d,non-sphere}}{C^i_{d,sphere}} \tag{35}$$

where φ^i is the ratio of drag coefficients for a particle in Newtonian fluids computed via Eqs.(33) or (34).

Taking into account that C_d in Eq.(25) corresponds to an spherical particle, Shah's dimensionless parameter is re-written using Eq.(35) as :

$$S_i = \left(\frac{C^i_{d,non-sphere}}{\varphi^i}\right)^{\frac{2-n}{2}} Re^i_t \tag{36}$$

Eq.(26) holds for both spherical and non-spherical particles. For non-spherical particles d^i_p is the diameter of the sphere with the same volume. Note that A and B were obtained for spherical particles in Eq.(24). This is why C^i_d must be divided by φ^i in order to compute the correct (equivalent) expression of C_d for a spherical particle.

The two-step procedure to compute the drag force for non-spherical particles is as follows:

1. Compute the terminal velocity for the particles using the standard Shah's method (Eq.(30)) where d_i is the equivalent diameter for the non-spherical particle.
2. Compute the drag coefficient for spherical ($\varphi = 1$) and non-spherical ($\varphi \neq 1$) particles via Eqs.(33) or (34) using the Reynolds number of Eq.(29).
3. Compute the value of φ^i.
4. Update S_i as $S^2_i = S^1_i(\varphi^i)^{\frac{2}{2-n}}$, where S^1_i is the value of Shah's parameter computed using Eq.(28).
5. With the updated Shah's parameter, perform the rest of operations in Eqs.(29)–(31).
   The terminal velocity and the drag force obtained after this second step includes the effect of the non sphericity of the particle.

2.5 Discretization of the fluid equations

The modelling of incompressible fluids with a mixed finite element method using an equal order interpolation for the velocities and the pressure requires introducing a stabilized formulation for the mass balance equation.

In our work we use a stabilized form of the momentum mass balance equations obtained via the Finite Calculus (FIC) technique [12, 16, 19, 37] written as

$$r_{m_i} - \underline{\frac{h_{ij}}{2}\frac{\partial r_{m_i}}{\partial x_j}} = 0 \quad \text{in } V \tag{37a}$$

$$r_v - \tau_s \frac{\partial \bar{r}_{m_i}}{\partial x_i} = 0 \quad \text{in } V \tag{37b}$$

where

$$\bar{r}_{m_i} = \frac{\partial \sigma_{ij}}{\partial x_j} + b_i + \frac{1}{n_f} f_i^{pf} \tag{38}$$

is a static momentum term and τ_s is a stabilization parameter computed as

$$\tau_s = \left(\frac{8\mu}{h^2} + \frac{2\rho_f}{\delta} \right)^{-1} \tag{39}$$

In Eqs.(37b) and (39) h_{ij} and h are characteristic length distances that are expressed in terms of the finite element dimensions and δ is a time parameter [12, 16, 19, 37].

The derivation of Eq.(37) for an homogeneous Lagrangian quasi-incompressible fluid is presented in [16].

The stabilization parameter τ is computed in practice for each element e using $h = l^e$ and $\delta = \Delta t$ as

$$\tau_s = \left(\frac{8\mu}{(l^e)^2} + \frac{2\rho}{\Delta t} \right)^{-1} \tag{40}$$

where Δt is the time step used for the transient solution and l^e is a characteristic element length computed as $l^e = 2(V^e)^{1/n_s}$ where V^e is the element area (for 3-noded triangles) or volume (for 4-noded tetrahedra). For fluids with heterogeneous material properties the values of μ and ρ in Eq.(40) are computed at the element center.

2.5.1 Variational equations for the fluid

The variational form of the momentum and mass balance equations is obtained via the standard weighted residual approach [16, 23–26]. The resulting integral expressions after integration by parts and some algebra are:

2.5.2 Momentum equations

The momentum equations can be written as

$$\int_V w_i \rho \frac{Dv_i}{Dt} dV + \int_V \left[\delta\varepsilon_{ij} 2\mu \varepsilon'_{ij} + \delta\varepsilon_v p \right] dV - \int_V w_i \left(b_i + \frac{1}{n_f} f_i^{pf} \right) dV - \int_{\Gamma_t} w_i t_i^p d\Gamma + \underline{\int_V \frac{\partial W_i}{\partial x_j} \frac{h_{ij}}{2} r_{m_i} dV} = 0 \tag{41}$$

2.5.3 Mass balance equation

The mass balance equation can be written as

$$\begin{aligned} \int_V \frac{q}{\kappa} \frac{Dp}{Dt} dV - \int_V q \left(\frac{1}{n_f} \frac{Dn_f}{Dt} + \varepsilon_v \right) dV + \int_V \tau_s \frac{\partial q}{\partial x_i} \left(\frac{\partial}{\partial x_i}(2\mu\varepsilon_{ij}) + \frac{\partial p}{\partial x_i} + b_i \right) dV \\ - \int_{\Gamma_t} q\tau \left[\rho \frac{Dv_n}{Dt} - \frac{2}{h_n}(2\mu \frac{\partial v_n}{\partial n} + p - t_n) \right] d\Gamma = 0 \end{aligned} \tag{42}$$

where h_n is a characteristic length size of an element adjacent to the boundary. For instance in our work h_n is taken as the square root of twice the element area for a triangle.

The derivation of Eqs.(41) and (42) for homogeneous Lagrangian fluids can be found in [16]. Their applications in the context of particulate Newtonian Lagrangian fluids is presented in [4].

Remark 4. For Lagrangian fluids the underlined stabilized terms in Eqs.(37) and (41) are zero.

2.5.4 FEM discretization

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 [14, 23, 25]. 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 global linear shape functions, N_j, spanning over the elements sharing node j ($j = 1, N$) [14, 23, 25].

The finite element interpolation over the fluid domain can be written in matrix form as

$$\mathbf{v} = \mathbf{N}_v \bar{\mathbf{v}} \ , \ \mathbf{w} = \mathbf{N}_v \bar{\mathbf{w}} \ , \ p = \mathbf{N}_p \bar{\mathbf{p}} \ , \ q = \mathbf{N}_p \bar{\mathbf{q}} \tag{43}$$

where

$$\bar{\mathbf{v}} = \begin{Bmatrix} \bar{\mathbf{v}}^1 \\ \bar{\mathbf{v}}^2 \\ \vdots \\ \bar{\mathbf{v}}^N \end{Bmatrix} \quad \text{with } \bar{\mathbf{v}}^i = \begin{Bmatrix} \bar{v}_1^i \\ \bar{v}_2^i \\ \bar{v}_3^i \end{Bmatrix} \ , \ \bar{\mathbf{w}} = \begin{Bmatrix} \bar{\mathbf{w}}^1 \\ \bar{\mathbf{w}}^2 \\ \vdots \\ \bar{\mathbf{w}}^N \end{Bmatrix} \quad \text{with } \bar{\mathbf{w}}^i = \begin{Bmatrix} \bar{w}_1^i \\ \bar{w}_2^i \\ \bar{w}_3^i \end{Bmatrix} \ , \ \bar{\mathbf{p}} = \begin{Bmatrix} \bar{p}^1 \\ \bar{p}^2 \\ \vdots \\ \bar{p}^N \end{Bmatrix} \quad \text{and } \bar{\mathbf{q}} = \begin{Bmatrix} \bar{q}^1 \\ \bar{q}^2 \\ \vdots \\ \bar{q}^N \end{Bmatrix} \tag{44}$$

$$\mathbf{N}_v = [\mathbf{N}_1, \mathbf{N}_2, \cdots, \mathbf{N}_N]^T \ , \ \mathbf{N}_p = [N_1, N_2, \cdots, N_N]$$

with $\mathbf{N}_j = N_j \mathbf{I}_{n_s}$, where $\mathbf{I}_{n_s}$ is the $n_s \times n_s$ unit matrix.

In Eq.(44), vectors $\bar{\mathbf{v}}$, $\bar{\mathbf{w}}$, $\bar{\mathbf{q}}$ and $\bar{\mathbf{p}}$ contain the nodal velocities, the nodal weighting functions and the nodal pressures for the whole mesh, respectively and the upper index denotes the nodal value for each vector or scalar magnitude.

Substituting the approximation (43) into the variational forms (41) and (42) gives the system of algebraic equations for the particulate fluid in matrix form as

$$\mathbf{M}_0 \dot{\bar{\mathbf{v}}} + (\mathbf{K} + \mathbf{A} + \mathbf{S})\bar{\mathbf{v}} + \mathbf{Q}\bar{\mathbf{p}} - \mathbf{f}_v = \mathbf{0} \tag{45a}$$

$$\mathbf{M}_1 \dot{\bar{\mathbf{p}}} - \mathbf{Q}^T \bar{\mathbf{v}} + (\mathbf{L} + \mathbf{M}_b)\bar{\mathbf{p}} - \mathbf{f}_p = \mathbf{0} \tag{45b}$$

where

$$\mathbf{M}^e_{0_{ij}} = \int_{\Omega^e} \rho N_i^e N_j \mathbf{I}_3 d\Omega \quad , \quad \mathbf{K}^e_{ij} = \int_{\Omega^e} \mathbf{B}_i^{eT} \mathbf{D} \mathbf{B}_j^e d\Omega \quad \text{and} \quad \mathbf{D} = \mu \begin{bmatrix} 2\,0\,0 \\ 0\,2\,0 \\ 0\,0\,1 \end{bmatrix} \quad , \quad \mathbf{B}_i = \begin{bmatrix} \dfrac{\partial N_i}{\partial x_1} & 0 \\ 0 & \dfrac{\partial N_i}{\partial x_2} \\ \dfrac{\partial N_i}{\partial x_2} & \dfrac{\partial N_i}{\partial x_1} \end{bmatrix} \quad \text{(for 2D)}$$

$$\mathbf{A}^e_{ij} = \left[\int_{\Omega^e} N_i (\rho \mathbf{u}^T \boldsymbol{\nabla}^T N_j) d\Omega \right] \mathbf{I}_3 \quad , \quad \mathbf{S}^e_{ij} = \left(\int_{\Omega^e} \boldsymbol{\nabla}^T N_i \hat{\mathbf{D}} \boldsymbol{\nabla} N_j d\Omega \right) \mathbf{I}_3$$

$$\mathbf{Q}^e_{ij} = \int_{\Omega^e} \mathbf{B}_i^{eT} \mathbf{m} N_j^e d\Omega \quad , \quad M^e_{1_{ij}} = \int_{\Omega^e} \frac{1}{\kappa} N_i^e N_j^e d\Omega \quad , \quad M^e_{b_{ij}} = \int_{\Gamma_t} \frac{2\tau_s}{h_n} N_i^e N_j^e d\Gamma$$

$$\mathbf{L}^e_{ij} = \int_{\Omega^e} \tau_s (\boldsymbol{\nabla}^T N_i^e) \boldsymbol{\nabla} N_j^e d\Omega \quad , \quad \mathbf{f}^e_{v_i} = \int_{\Omega^e} \mathbf{N}_i^e \mathbf{b} d\Omega + \int_{\Gamma_t} \mathbf{N}_i^e \mathbf{t} d\Gamma$$

$$f^e_{p_i} = \int_{\Gamma_t} \tau_s N_i^e \left[\rho \frac{Dv_n}{Dt} - \frac{2}{h_n} (2\mu \varepsilon_n - t_n) \right] d\Gamma - \int_{\Omega^e} \tau_s \boldsymbol{\nabla}^T N_i^e \mathbf{b} d\Omega \qquad \text{with} \quad i, j = 1, n_s$$

In the expressions of $\mathbf{A}^e_{ij}$ and $\mathbf{S}^e_{ij}$, $\mathbf{I}_3$ is the 3×3 unit matrix and $\hat{\mathbf{D}} = \dfrac{\rho}{2} \mathbf{h} \mathbf{u}^T$, with $\mathbf{h}$ being a characteristic length parameter. A typical definition of $\mathbf{h}$ is

$$\mathbf{h} = h_s \frac{\mathbf{u}}{u} + h_c \frac{\boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|} \tag{46}$$

where $u = |\mathbf{u}|$ and h_s and h_c are "streamline" and "cross wind" characteristic lengths given by $h_s = \max(l_j^T \mathbf{u})/u$ and $h_c = \max(l_j^T \boldsymbol{\nabla} \mathbf{u})/|\boldsymbol{\nabla} u|$, where j ranges from one to the number of element sides and l_j is the vector defining the jth element side [12].

In Eq.(45a), $\mathbf{K}$, $\mathbf{A}$ and $\mathbf{S}$ are matrices emanating from the viscosity, the advection terms and the stabilization terms in the momentum equations. Matrices $\mathbf{A}$ and $\mathbf{S}$ are zero for Lagrangian flows. The derivation of the matrices and vectors in Eqs.(45) for Eulerian and Lagrangian flows can be found in [12, 16].

2.5.5 Incremental solution of the discretized equations

Eqs.(45) are solved in time with an implicit Newton-Raphson type iterative scheme [23–26]. 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 required for a Lagrangian formulation (such as PFEM) only.
- Initialize variables: $({}^{n+1}\mathbf{x}^1, {}^{n+1}\bar{\mathbf{v}}^1, {}^{n+1}\bar{\mathbf{p}}^1, {}^{n+1}n_f^i, {}^{n+1}\bar{\mathbf{r}}_m^1) \equiv \{{}^{n}\mathbf{x}, {}^{n}\bar{\mathbf{v}}, {}^{n}\bar{\mathbf{p}}, {}^{n}n_f, {}^{n}\bar{\mathbf{r}}_m\}$.
- Iteration loop: $k = 1, \cdots, NITER$.
  For each iteration.

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

From Eq.(45a), we deduce

$$ {}^{n+1}\mathbf{H}_v^i \Delta\bar{\mathbf{v}} = -{}^{n+1}\bar{\mathbf{r}}_m^k \rightarrow \Delta\bar{\mathbf{v}} \tag{47a}$$

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

$$ \bar{\mathbf{r}}_m = \mathbf{M}_0\dot{\bar{\mathbf{v}}} + (\mathbf{K} + \mathbf{A} + \mathbf{S})\bar{\mathbf{v}} + \mathbf{Q}\bar{\mathbf{p}} - \mathbf{f}_v \quad , \quad \mathbf{H}_v = \frac{1}{\Delta t}\mathbf{M}_0 + \mathbf{K} + \mathbf{A} + \mathbf{S} + \mathbf{K}_v \tag{47b}$$

where $\mathbf{K}_v^e$ is

$$ \mathbf{K}_v^e = \int_{{}^{n}V^e} \mathbf{B}^T\mathbf{m}\theta\Delta t\kappa\mathbf{m}^T\mathbf{B}dV \tag{47c}$$

The tangent "bulk" stiffness matrix, $\mathbf{K}_v$, accounts for the changes of the pressure due to the velocity. Matrix $\mathbf{K}_v$ in $\mathbf{H}_v$ is important for the fast convergence, mass preservation and overall accuracy of the iterative solution [7, 16].

The parameter θ in $\mathbf{K}_v$ $(0 < \theta \leq 1)$ has the role of preventing the ill-conditioning of the iteration matrix $\mathbf{H}_v$ for quasi-incompressible fluids characterized by very large values of the bulk parameter κ. An adequate selection of θ improves the overall accuracy of the numerical solution and preserves the mass for large time steps [7, 16].

Step 2. Update the velocities

$$ \text{Fluid nodes: } {}^{n+1}\bar{\mathbf{v}}^{k+1} = {}^{n+1}\bar{\mathbf{v}}^{k} + \Delta\bar{\mathbf{v}} \tag{48a}$$

$$ \text{Rigid particles: } \begin{cases} {}^{n+1/2}\dot{\mathbf{u}}_j = {}^{n-1/2}\dot{\mathbf{u}}_j + {}^{n}\ddot{\mathbf{u}}_j^{k+1}\Delta t \\ \dot{\mathbf{u}}_j = \frac{1}{m_j}{}^{n}\mathbf{F}_j^{k+1} \quad , \quad j = 1, N_p \end{cases} \tag{48b}$$

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

From Eq.(45b) we obtain

$$ {}^{n+1}\mathbf{H}_p^i {}^{n+1}\bar{\mathbf{p}}^{k+1} = \frac{1}{\Delta t}\mathbf{M}_1 {}^{n+1}\bar{\mathbf{p}}^{i} + \mathbf{Q}^T {}^{n+1}\bar{\mathbf{v}}^{k+1} + {}^{n+1}\bar{\mathbf{f}}_p^i \rightarrow {}^{n+1}\bar{\mathbf{p}}^{k+1} \tag{49a}$$

with

$$ \mathbf{H}_p = \frac{1}{\Delta t}\mathbf{M}_1 + \mathbf{L} + \mathbf{M}_b \tag{49b}$$

Step 4. Update the coordinates of the particles

$$ \text{Rigid particles: } \begin{cases} {}^{n+1}\mathbf{u}_i^{k+1} = {}^{n}\mathbf{u}_i^{k+1} + {}^{n+1/2}\dot{\mathbf{u}}_i^{k+1}\Delta t \\ {}^{n+1}\mathbf{x}_i^{k+1} = {}^{n}\mathbf{x}_i + {}^{n+1}\mathbf{u}_i^{k+1} \quad , \quad i = 1, N_p \end{cases} \tag{50a}$$

Step 5. Update the coordinate of the fluid nodes (for Lagrangian flows only)

$$\text{Fluid nodes: } {}^{n+1}\mathbf{x}_i^{k+1} = {}^{n+1}\mathbf{x}_i^{k} + \frac{1}{2}({}^{n+1}\bar{\mathbf{v}}_i^{k+1} + {}^{n}\bar{\mathbf{v}}_i)\Delta t \quad , \quad i = 1, N \tag{50b}$$

Step 6. Compute the fluid volume fraction for each node ${}^{n+1}n_{f_i}^{k+1}$ *via Eq.(2)*

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

Step 8. Compute particle-to-fluid forces: $({}^{n+1}\mathbf{f}_i^{pf})^{k+1} = -({}^{n+1}\mathbf{f}_i^{fp})^{k+1}$, $i = 1, N$ with $\mathbf{f}_i^{fp}$ computed by Eq.(18)

Step 9. Check convergence

Verify the following conditions:

$$\begin{aligned} \|{}^{n+1}\bar{\mathbf{v}}^{k+1} - {}^{n+1}\bar{\mathbf{v}}^{k}\| &\leq e_v \|{}^{n}\bar{\mathbf{v}}\| \\ \|{}^{n+1}\bar{\mathbf{p}}^{k+1} - {}^{n+1}\bar{\mathbf{p}}^{k}\| &\leq e_p \|{}^{n}\bar{\mathbf{p}}\| \end{aligned} \tag{51}$$

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 (51) 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 5. In Eqs.(47)–(51), ${}^{n+1}(\cdot)$ denote values at time $n+1$. For the Lagrangian formulation using the PFEM the derivatives and integrals in the iteration matrices, $\mathbf{H}_v$ and $\mathbf{H}_p$ and the residual vector $\bar{\mathbf{r}}_m$, are computed on the *discretized geometry at time* n (i.e. $V^e = {}^{n}V^e$) while the nodal force vectors, $\mathbf{f}_v$ and $\mathbf{f}_p$, are computed on the current configuration at time $n+1$ [16].

Remark 6. 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, with l^e computed as explained in Eq.(40), $|{}^{n}\mathbf{v}|_{\max}$ 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 [16].

Remark 7. The Eulerian and Lagrangian versions of the formulation have been implemented in the open-source Kratos software platform [38]. The generation of the analysis data and the visualization of the results have been carried out using the GiD pre/postprocessing system [39].

3 EXAMPLES

3.1 Motion of cuttings in a vertical annulus for different fluids

The first problem concerned the study of the transport of cuttings in drilling muds in a vertical wellbore with a centered non-rotating drill string. Numerical results for this problem were obtained with the Lagrangian formulation presented in this work and the PFEM. The average velocity of the particles at a section of the annulus was measured and compared to the average fluid velocity. Non-spherical particles were considered. The drag force was computed as explained in Section 5.1 using the sphericity correction of Eq.(34) [36]. Results are plotted in Figure 5, where experimental data [2] is also shown for comparison. Table 1 shows the rheological properties of the fluids used for defining the viscosity function (Eq.(23)).

	n	K (Pa s^n)	ρ (kg/m^3)
Thick mud	0.33958	3.15275	1030
Intermediate mud	0.37826	1.7637	1030
Water	1	10^{-3}	1030

Table 1: Rheological properties of the fluids used in Figure 5

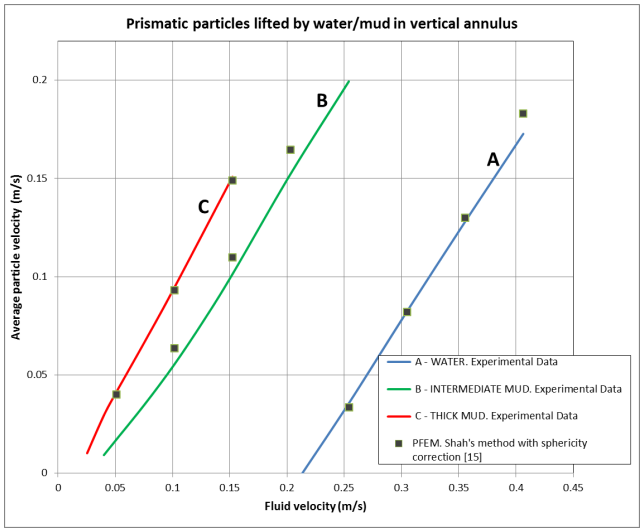


Fig. 5: Average velocity of cuttings velocity vs. fluid velocity for different fluids in a vertical annulus. Lines: Experimental data [2]. Squares: PFEM results

Figure 6 shows results of the motion of the cuttings in a centered drill string, rotating at 10 rpm, which causes the cuttings to move upwards faster. The fluid is an intermediate mud with the following rheogram characteristics $n = 0,37826$ and $K = 1,7637$ Pa s^n (Eq.(23)). Results were obtained using PFEM. The boundary conditions used were the imposed velocity at the inlet, non-slip condition on all was ($\bar{v} = 0$) and free surface at the outlet ($\sigma_{ij} n_j = 0$). The free surface is kept at the same location by removing the nodes crossing the outlet.

The particles have the following characteristics: diameter (4.96 mm), sphericity (0.76766), density (2000 $^{\text{kg}}/_{\text{m}^3}$). These characteristics correspond to particles with the shape of a brick with dimensions $1/8 \times 1/4 \times 1/8$ in, which have been treated as spheres. It must be clarified that the experiments carried out by Sifferman *et al.* [2] were done with fluids with a density of 8.6 $^{\text{le}}/_{\text{gal}}$ (1030 $^{\text{kg}}/_{\text{m}^3}$) and simulated denser fluids with lighter particles.

The DEM contact properties were chosen to just prevent particles from penetrating each other. For this purpose we have used $K_n = 3 \cdot 10^5$ N/m, $K_s = 6,5 \cdot 10^4$ N/m, a Coulomb friction coefficient of $\mu = 0.3$ and the critical damping parameter.

The method to obtain the drag force for non-Newtonian fluids is the one explained in Section 2.4. The example in this section is a vertical wellbore where the slip velocity of the cuttings is close to the terminal velocity, so it is an example well suited for the method. However, the velocity profile in the annulus must be computed accurately and the distribution of the particles, affected by its horizontal velocity, must also be accurate to match the experimental results. Note that the concentration of particles in the section of the annulus is not imposed, but every single particle is injected in a random position of the inlet, and only after some time of ascending motion they reach a steady position in the section. Measurements of the average velocity of the particles were taken close to the outlet.

3.2 Transport of cuttings in inclined and horizontal drilling annulus

The Eulerian and Lagrangian formulations have been applied to the study of the transport of cuttings in inclined and horizontal wellbores that can lead to the formation of particle beds (Figures 7 and 8). The high concentration of particles does not affect the stability of the fluid solver until the size of the cuttings equals the fluid mesh size. The fluid used is the same intermediate mud as in Section 7.1. Results in Figure 7 were obtained using PFEM, while those in Figure 8 were obtained using an Eulerian approach implemented in Kratos [12, 38].

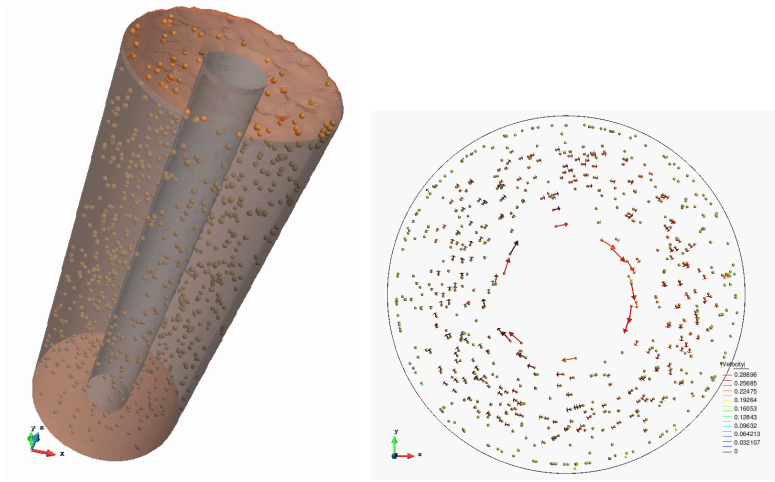


Fig. 6: Global view (left) and top view (rigth) of particles flowing at a rate of 1500 particles/(m^2s) within a vertical annulus. The centered drilling pipe rotates at 100 rpm

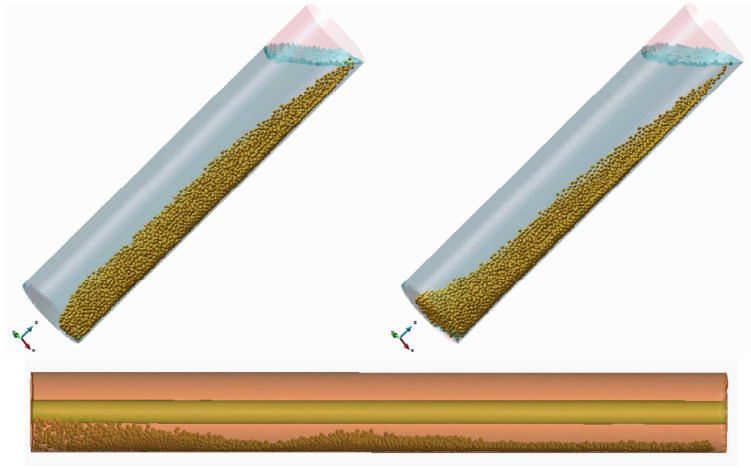


Fig. 7: Global view of an inclined (above) and horizontal (below) annulus carrying mud and cuttings at a rate of 1500 particles/(m^2s). The horizontal annulus includes a centered non-rotating drilling pipe

3.3 Cutting transport through more complex sections of the wellbore

The formulation presented has been tested with more complex geometries, like a section of the wellbore with a tool joint (Figure 9a) and a curved wellbore with a rotating drill string (100 r_{mp}) (Figure 9b). The fluid properties are the same as in Section 7.1. All the computations for these problems were carried out with PFEM.

In some of these problems beds can be formed. In such conditions, the drag forces are no longer accurate, since they are based on the assumption that the particles are disperse. This method should therefore used carefully, as no estimations of the error has been obtained in dense packing conditions. However, the examples shown prove the stability of the method in those situations.

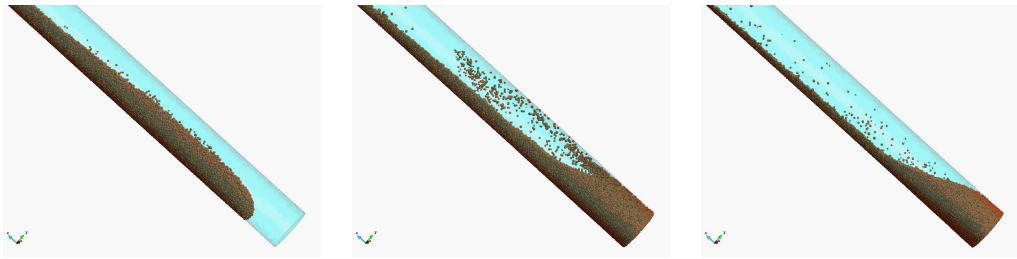


Fig. 8: Simulation of the accumulation of particles as they fall towards the bottom of a tube filled with mud

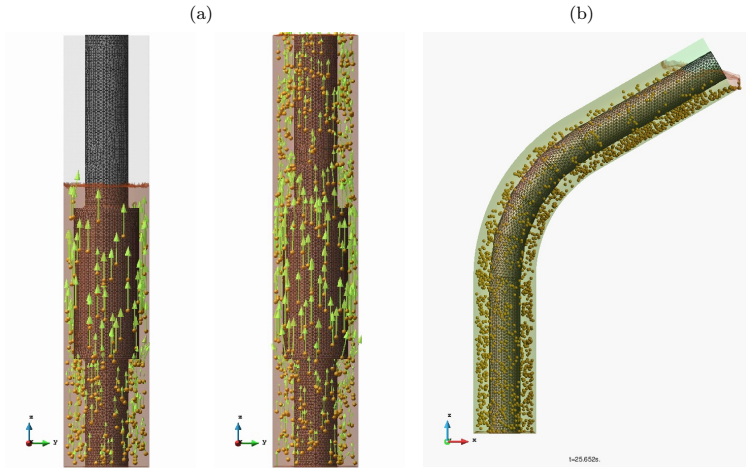


Fig. 9: Simulation of drill cuttings transported in straight and curved wellbores by mud as a result of a drilling operation. Arrows denote the velocity vector of each particle

4 CONCLUSIONS

A FEM-DEM technique has been presented to solve the transport of particles in non-Newtonian fluids which can be applied to wellbores full of circulating fluid, typical of the drilling industry. We have proposed a procedure for computing the drag force on the particles for non-Newtonian fluids using predictions of the terminal velocity available from the existing literature. The procedure has been extended to non-spherical particles, treating them as spheres in terms of contact forces but correcting the drag force according to the shape of the particles.

The usefulness of the proposed numerical method for studying the motion of the drill cuttings in vertical wellbores has been validated and the applicability and stability to other non-vertical and more complex configurations have been pointed out.

All of the developments done in terms of coupling between the non-Newtonian fluid and the DEM can be applied both to a Lagrangian PFEM approach for the fluid or to an Eulerian one. The coupling procedure is actually not dependent on the method used to solve the equations for the fluid. Therefore, it can be applied to other popular CFD methods, such as the Finite Volume Method (FVM) [40], the Lattice-Bolzmann Method (LBM) [41] or the Smoothed Particle Hydrodynamics (SPH) method [42]. This technique can also be combined with the erosion estimation methods published previously by the authors [11, 13].

Acknowledgements

This work was carried out with financial support by Weatherford. This research was also partially supported by project SAFECON of the European Research Council.

REFERENCES

1. Bourgoyne AT, Millheim KK, Chenevert ME, Young FS (1991) Applied Drilling Engineering. Society of Petroleum Engineers. Richardson, TX
2. Sifferman TR, Myers GM, Haden EL, Wahl HA (1974) Drill-Cutting Transport in Full-Scale Vertical Annuli. Journal of Petroleum Technology
3. Oñate E (1998) Derivation of stabilized equations for advective-diffusive transport and fluid flow problems. Comput. Meth. Appl. Mech. Engng. 151:233–267
4. Oñate E, Celigueta MA, Latorre S, Casas G, Rossi R, Rojek J (2014) Lagrangian analysis of multiscale particulate flows with the particle finite element method. Computational Particle Mechanics 1:85–102
5. Zohdi T (2007) An introduction to modelling and simulation of particulate flows. SIAM, Computational Science and Engineering
6. Cremonesi M, Frangi A, Perego U (2011) A Lagrangian finite element approach for the simulation of water-waves induced by landslides. Computers & Structures 89:1086–1093
7. Franci A, Oñate E, Carbonell JM (2015) On the effect of the bulk tangent matrix in partitioned solution schemes for nearly incompressible fluids. Accepted for publication in Int. J. Numer. Meth. Engng. DOI:10.1002/nme.4839
8. Idelsohn SR, Oñate E, Del Pin F (2004) The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. Int. J. Numer. Meth. Engng, 61(7):964–989
9. Idelsohn SR, Mier-Torrecilla M, Oñate E (2009) Multi-fluid flows with the Particle Finite Element Method. Comput Methods Appl Mech Engrg. 198:2750–2767
10. Oñate E, Idelsohn SR, Del Pin F, Aubry R (2004c) The particle finite element method. An overview. Int. J. Comput. Methods 1(2):267–307
11. Oñate E, Celigueta MA, Idelsohn SR (2006a) Modeling bed erosion in free surface flows by the Particle Finite Element Method, Acta Geotechnia 1(4):237–252
12. Oñate E, García J, Idelsohn SR, Del Pin F (2006c) FIC formulations for finite element analysis of incompressible flows. Eulerian, ALE and Lagrangian approaches. Comput. Meth. Appl. Mech. Engng. 195(23-24):3001–3037
13. Oñate E, Idelsohn SR, Celigueta MA, Rossi R (2008) 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
14. Oñate E (2009), Structural analysis with the finite element method. Volume 1. Basis and Solids. CIMNE-Springer
15. Oñate E, Celigueta MA, Idelsohn SR, Salazar F, Suárez B (2011) Possibilities of the particle finite element method for fluid-soil-structure interaction problems. Computational Mechanics, 48(3):307–318.
16. Oñate E, Franci A, Carbonell JM (2014) Lagrangian formulation for finite element analysis of quasi-incompressible fluids with reduced mass losses. Int. J. Num. Meth. in Fluids 74(10):699–731
17. Avci B, Wriggers P, (2012) A DEM-FEM coupling approach for the direct numerical simulation of 3D particulate flows. Journal of Applied Mechanics, 79(1), 7 pages
18. Cundall PA, Strack ODL (1979) A discrete numerical method for granular assemblies, Geotechnique 29:4765
19. Oñate E, Rojek J, (2004b) Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems. Comput. Meth. Appl. Mech. Engrg. 193:3087–3128
20. Oñate E, Zárate F, Miquel J, Santasusana M, Celigueta MA, Arrufat F, Gandikota R, Ring KL (2015) A local constitutive model for the discrete element method. Application to geomaterials and concrete. Comp. Part. Mech 2(2):139–160
21. Clift R, Grace JR, Weber ME (1978) Bubbles, drops and particles. Academic Press, New York
22. Coussy O (2004) Poromechanics. Wiley
23. Zienkiewicz OC, Taylor RL, Zhu JZ (2005) The finite element method. The basis. 6th Ed., Elsevier
24. Zienkiewicz OC, Taylor RL, Nithiarasu P (2005) The finite element method for fluid dynamics. 6th Ed., Elsevier
25. Belytschko T, Liu WK, Moran B (2013) Non linear finite element for continua and structures. 2d Edition, Wiley
26. Donea J, Huerta A (2003) Finite element method for flow problems. J. Wiley
27. Jackson R (2000), The dynamics of fluidized particles. Cambridge Monographs on Mechanics, Cambridge Univ. Press
28. Best Practice Guidelines for Computational Fluid Dynamics of Dispersed Multiphase Flows. By SIAMUF, Swedish Industrial Association for Multiphase Flows, ERCOFTAC. 2008
29. Ansley RW, Smith TN (1967) Motion of spherical particles in a Bingham plastic. AIChE Journal 13(6):11931196

30. Brookes GF, Whitmore RL (1969) Drag forces in Bingham plastics. Rheologica Acta, 8(4):472–480
31. Kelessidis VC, Mpandelis G (2004) Measurements and prediction of terminal velocity of solid spheres falling through stagnant pseudoplastic liquids. Powder Technology 147
32. Chien SF (1994), Settling velocity of irregularly shaped particles, SPE Drill. Complet. 9 281
33. Shah SN, El Fadili Y, Chhabra RP (2007) New model for single spherical particle settling ve-locity in power law (visco-inelastic) fluids. International Journal of Multiphase Flow 33:51–66
34. Walker RE, Mayes TM (1975) Design of muds for carrying capacity. Journal of Petroleum Technology 27(7)
35. Walker RE (1976) Hydraulics limits are set by flow restrictions. Oil Gas Journal 86–90
36. Haider A, Levespiel O (1989) Drag coefficient and terminal velocity of spherical and nonspherical particles. Powder Technol. 58:6370
37. Oñate E (2004) Possibilities of finite calculus in computational mechanics. Int. J. Num. Meth. Engng. 60(1):255281
38. Kratos (2015) Open source software platform for multiphysics computations. CIMNE, Barcelona, www.cimne.com/kratos
39. GiD (2015). The personal pre/postprocessor. www.gidhome.com, CIMNE, Barcelona
40. Breuer M, Rodi W (1994). Large-eddy simulation of turbulent flow through a straight square duct and a 180^o bend. In Fluid Mechanics and its Applications, Vol. 26, Voke P.R., Kleiser L. and Hollet J.P. (Eds.). Direct and Large-Eddy Simulation I. Selected Papers from the First ERCOFTAC Workshop on Direct and Large-Eddy Simulation. Guildford, Surrey, UK, 2730 March 1994. Kluwer Academic Publishers, Dordrecht, pp. 273285
41. He X, Luo LS (1997) Lattice Boltzmann model for the incompressible NavierStokes equation. J. Stat. Phys., 88(34):927944
42. Monaghan JJ (1992) Smoothed particle hydrodynamics. Annu. Rev. Astron. Astrophys., 30:543–574

Paper 8

This paper, published in 2016, explains the methodology used to assess the air demand in bottom outlets of dams. A formulation with two fluids was used. The *PFlow* code was used for all the computations.

Noname manuscript No.
(will be inserted by the editor)

Air demand estimation in bottom outlets with the particle finite element method.

Susqueda Dam case study

Fernando Salazar · Javier San-Mauro · Miguel Ángel Celigueta · Eugenio Oñate

Received: date / Accepted: date

F. Salazar · J. San-Mauro · M. Á. Celigueta · E. Oñate
Centre International de Mètodes Numèrics en Enginyeria (CIMNE). Universitat Politècnica de Catalunya (UPC), Campus Norte UPC, Gran Capitán s/n. 08034. Barcelona, Spain
Tel.: +34-934-071-495

F. Salazar
E-mail: fsalazar@cimne.upc.edu

J. San Mauro
E-mail: jsanmauro@cimne.upc.edu

M.Á. Celigueta
E-mail: maceli@cimne.upc.edu

E. Oñate
E-mail: onate@cimne.upc.edu

Noname manuscript No.
(will be inserted by the editor)

Air demand estimation in bottom outlets with the particle finite element method.

Susqueda Dam case study

Received: date / Accepted: date

Abstract Dam bottom outlets play a vital role in dam operation and safety, as they allow controlling the water surface elevation below the spillway level. For partial openings, water flows under the gate lip at high-velocity and drags the air downstream of the gate, which may cause damages due to cavitation and vibration. The convenience of installing air vents in dam bottom outlets is well known by practitioners. The design of this element depends basically on the maximum air flow through the air vent, which in turn is a function of the specific geometry and the boundary conditions. The intrinsic features of this phenomenon makes it hard to analyse either on site or in full scaled experimental facilities. As a consequence, empirical formulas are frequently employed, which offer a conservative estimate of the maximum air flow. In this work, the particle finite element method was used to model the air-water interaction in Susqueda Dam bottom outlet, with different gate openings. Specific enhancements of the formulation were developed to consider air-water interaction. The results were analysed as compared to the conventional design criteria and to information gathered on site during the gate operation tests. This analysis suggests that numerical modelling with the PFEM can be helpful for the design of this kind of hydraulic works.

Keywords Particle finite element method · Two fluids · Bottom outlets · Air demand

1 Introduction

Air-water interaction is a relevant phenomenon in multiple hydraulic works involving high-velocity free-surface flows, such as spillways and bottom outlets [37]. Under this conditions, the turbulent flow produces air entrainment, which results in flow bulking. Hence, the density of the aerated flow is given by $\rho_w (1 - V) + \rho_a (V) \approx \rho_w (1 - V)$, where ρ_w is the water density, ρ_a is the air density and V is the void fraction [6].

The practical consequences of this phenomenon are diverse. Flow bulking favours the energy dissipation and results in a water-solid friction reduction. [40].

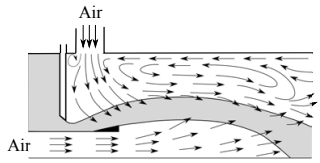


Fig. 1 Typical flow pattern in high-head bottom outlets. Adapted from [36]

In spillway chutes and open channels, aeration produces an increase of the effective depth, so that a larger section is needed to avoid wall overtopping.

In bottom outlets, there is a need to allow and even encourage air entrance to avoid the formation of negative pressures that can damage the structure [42]. To this end, a conduit is typically installed in the downstream side of the gate, which allows air supply. The design of this element is mostly based on the maximum air demand, i.e., the maximum air flow to be conducted within the gate operation range.

The occurrence of major breakdowns in Roosevelt and Pathfinder dams [9] revealed this need, and encouraged several authors to investigate the phenomenon. As a result, several systematic experimental campaigns were carried out with the aim of deriving empirical formulas [35], [4], as well as identifying the most influential factors.

Thus, it was observed that the aerated flow pattern downstream of a high-head gate essentially depends on [36]:

- The upstream and downstream boundary conditions
- The gate opening
- The presence of bottom aerators
- The geometry of the conduit

Fig. 1 depicts the typical flow pattern in a partially-opened high-head gate with bottom aerator.

Although the phenomenon was the subject of numerous studies in the past, the hydraulic behaviour of aerated flows in bottom outlets is not fully understood and depends on each particular case. Hydraulic modelling on experimental facilities at convenient scale is typically recommended [41].

As a first approximation for a preliminary design, the formulas proposed by Sharma [35] are frequently used. They are conservative, since they are based on the envelope of the maximum values obtained in laboratory for each situation. Different expressions are recommended in function of the flow regime in the conduit. The same author identified six types of aerated flow regimes in bottom outlets, as a function of the gate opening and the downstream boundary conditions (Fig. 2).

The applicability of these formulas is limited, because they are based on results of small-scale tests. Moreover, they were performed on square conduits, while many bottom outlets feature round sections.

Tullis and Larchar [38] proposed a general methodology for designing the aeration system, also based on experimental results. Nonetheless, this method is specific for small to medium-sized embankment dams with an inclined slide gate.

These methods are far from being generally applicable. As a result, it is frequent, in practice, to perform specific experimental tests (e. g. [31], [38], [11]).

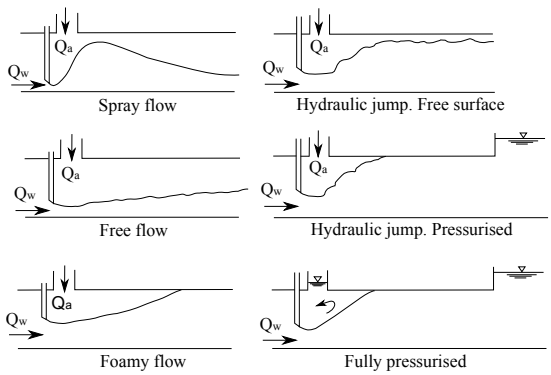


Fig. 2 Types of aerated flow regimes in bottom outlets, as defined by Sharma [35]. Q_a = air flow; Q_w = water flow;

Numerical modelling is an obvious alternative to these empirical and experimental approaches. However, the complexity of the air-water interaction phenomenon prevented practitioners from its application. The problem is three-dimensional, and requires considering two fluids which strongly interact by means of diverse processes [6]: the turbulence induces free-surface breakup which allows air entrainment into the water flow; the entrapped air may travel within the flow in bubbles and droplets of a broad range of sizes; these can collide and eventually join, or exit again if the appropriate flow conditions occur.

The direct simulation of these processes is infeasible, but they can be considered at a macroscopic scale producing useful results for practical purposes. Some formulations exist to consider the turbulent air entrainment coupled with variable density estimation, such as that implemented in the commercial software Flow3D [12]. Some examples of application to spillway hydraulics were already published [40]. In general, the objective is to calculate the macroscopic variables (air concentration, and especially air flow), which suffice to determine the appropriate geometry of the aeration system. Other commercial softwares such as ANSYS use a similar approach [2], also based on the volume of fluid (VOF) model [20]. The main drawback of this method is that convenient models need to be chosen for air entrainment and flow turbulence, whose parameters require a detailed calibration process.

The authors have implemented a particular class of Lagrangian formulation for solving problems involving complex interactions between free surface fluids. The so-called Particle Finite Element Method (PFEM) tracks the trajectory of the nodes of the mesh, including those on the free surface or in a fluid-fluid interface, and is even able to model the separation of parts of the domain, such as droplets. A mesh connects the nodes discretising the domain where the governing equations are solved using a stabilised version of the Finite Element Method (FEM). Details of the PFEM can be found in [1], [24], [27], [19], [15], [16], [3], [34].

With the PFEM, air entrainment can be naturally modelled by means of mixed elements (combining water and air nodes). This can be done because the information in the PFEM is stored in the nodes, as opposite to Eulerian approaches. The

density of the aerated flow is automatically computed depending on the proportion of air and water nodes in a certain sub-domain. The purpose of this approach is also to extract the macroscopic variables, rather than to model each volume of trapped air, which can appear in a broad range of sizes, from microscopic to centimetric. In particular, it is possible to quantitatively reproduce the air drag capacity of the flow, and therefore the air flow rate of a determined facility, which is the basis of the design of the aeration system. Additionally, other variables of interest can be analysed, as the pressure on the downstream face of the gate, or the velocity in the air vent.

The evolution of PFEM led to the so-called PFEM2 technique [14]. It shares most of the PFEM features while using large time steps, hence resulting in lower computational cost. Since the PFEM2 is also adequate to analyse multifluid flows [17], it could be an alternative to face the problem under consideration. Nonetheless, we chose PFEM instead of PFEM2 because a) the computational cost was not critical in our problem, and b) the first version was specifically validated for simulation of free surface flows in the field of dam hydraulics [18].

This paper presents some improvements implemented in the PFEM formulation to tackle the simulation of air-water flows, together with its application to verify the performance of the bottom outlet of Susqueda Dam. The results were analysed, both qualitatively and quantitatively, for different gate openings. They are consistent with the performance observed during the operation tests, as well as with existing design recommendations [39].

The rest of the paper is organised as follows: first, a brief introduction to the PFEM is presented, together with the description of some specific functionalities developed for this application. Then, the case study is introduced, and the numerical model set described. In the final sections, the results are shown and discussed, and some conclusions are drawn regarding future applications and developments.

2 Numerical model

In the PFEM, the domain is modelled using an updated Lagrangian formulation [43]. That is, all variables are assumed to be known in the current configuration at time t. The new set of variables in the domain 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. Hence a mesh discretising the domain must be generated in order to solve the governing equations in the standard FEM fashion.

The equations to be solved are the Navier-Stokes equations for incompressible fluids:

Momentum conservation

$$\rho \frac{Du_i}{Dt} = -\frac{\partial}{\partial x_i} p + \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right) + \rho f_i \tag{1}$$

for $i, j = x, y, z$

Mass conservation

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2}$$

for $i = x, y, z$
with

$$u = \bar{u} \tag{3}$$

for the solid nodes and

$$p = 0 \tag{4}$$

for the free surface fluid nodes.

In above equations, ρ and μ are the fluid density and dynamic viscosity, respectively, p is the pressure, u_i are the velocities along the ith global (cartesian) axis, f_i are the body forces, and $\bar{u}$ is the prescribed velocity.

According with the PFEM technique [27], equations 1 and 2 are discretised with a standard FEM mesh and then solved. When the finite elements get very distorted, the mesh is re-generated, but the nodes and their information are conserved. Adaptive mesh refinement techniques can be used to improve the solution in zones where large motions of the fluid or the structure occur.

The method has been employed to face a variety of problems in different fields of engineering, such as free surface flows [19], landslides [33], [34], industrial forming processes [26], ground excavation [5], fluid-structure interaction [21], among others [23], [22], [13], [28]. The details of the algorithm and our implementation was described in previous publications [21], [24], [25]. In this paper, only the basic steps of the algorithm are succinctly described, together with some enhancements specifically implemented for the present application.

2.1 Basic steps of the PFEM

In the PFEM, the mesh nodes in the fluid and solid domains are treated as particles that contain all the information as regards the geometry and the material and mechanical properties of the underlying subdomains.

A typical solution with the PFEM involves the following steps.

1. The starting point at each time step is the cloud of points C in the fluid and solid (boundary) domains. For instance, ${}^{n}C$ denotes the cloud at time $t = {}^{n}t$ (Fig. 3).
2. The domain is discretised with a finite element mesh ${}^{n}M$ using the particles as the mesh nodes. We use an efficient mesh generation scheme based on the Delaunay tesselation [16].
3. The free surface is detected by means of the Alpha Shape Method [8], which removes big and distorted elements.
4. The Lagrangian equations of motion for the overall continuum are solved using the standard FEM. The state variables in the next (updated) configuration for ${}^{n}t + \Delta t$ are computed: velocities, pressure, strain rate and viscous stresses.
5. The mesh nodes are moved to a new position ${}^{n+1}C$ where $n + 1$ denotes the time ${}^{n}t + \Delta t$, in terms of the time increment size.
6. Go back to step 1 and repeat the solution for the next time step to obtain a new ${}^{n+1}C$.

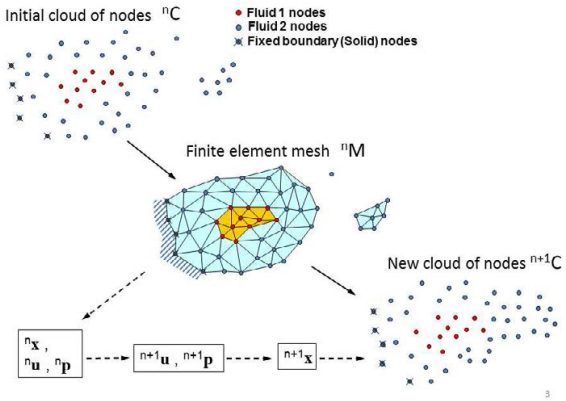


Fig. 3 Sequence of steps to update a "cloud" of particles (nodes) representing a domain containing two fluids and a solid boundary from ${}^{n}t$ to ${}^{n+1}t$ (colour figure online).

2.2 Mesh quality maintenance operations

It was previously mentioned that PFEM is suitable for modelling fluids in which the free surface suffers severe distortions during the transient solution. In the case of aeration in bottom outlets, a fluid (water) enters at high velocity into a domain initially occupied by another fluid (air) at rest. This implies a greater difficulty for maintaining a sufficient quality mesh during the calculation. To ensure this quality, some improvements in the meshing algorithm have been implemented, as described below.

Removing nodes: Usually, the computations carried out with the PFEM tend to create very distorted meshes. This means that the nodes, when following their trajectories in a Lagrangian fashion, can get very close one to another. Sometimes, two, three or more nodes join in a reduced space, generating very distorted elements with near null volume and a high aspect ratio. To avoid this problem, the authors have adopted a method consisting on removing one node of the mesh if it detects that another is present at a short distance (a fraction of h, where h is the desired/imposed mesh size). During the generation of the Delaunay Tessellation, for which the incremental insertion method is used [7], if the node to be inserted is marked with a special flag, it is not inserted. By doing this, the final connectivities of the mesh do not include the removed node, but do have elements that connect the remaining nodes according to the Delaunay Tessellation (Fig. 4).

Adding nodes: When the Delaunay Tessellation is complete, the resulting elements filling the space previously occupied by a removed node might be bigger than the desired size. If nothing is done, the Alpha-shape method might remove these elements, resulting in a void in the interior of the fluid, the free surface condition would be imposed on the nodes next to the void and the fluid pressure field would be spoiled. There must be then another mesh reparation step before the

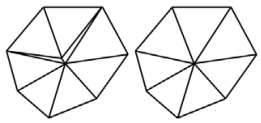


Fig. 4 Mesh quality maintenance by removing a node that forms distorted elements

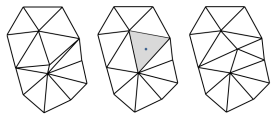


Fig. 5 Node insertion to enhance mesh quality after removing a node

Alpha-shape method is applied. Those elements bigger than a certain size (measured with the circumradius) must be refined by adding a node in the circumcenter of the element (Fig. 5). This step must be done far from the free surface detected in the last time step, or the Alpha-shape method would never remove any element. A local Delaunay Tessellation is enough for each node inserted.

2.3 Modelling water and air

For modelling water that fills a pipe (or a cavity) full of air, the whole domain was filled with air nodes and the water was injected from one of the sides. The density of an element is taken as the average of the densities of the nodes of the element. The fluid mixture is solved as a single fluid with a heterogeneous density. When two nodes of different material get too close and the mesh is too distorted, one of the nodes must be removed. The water node is considered prevalent, so the air node is removed. If one node must be added, its material will be that of the nodes which are in majority. With this approach, the mass conservation of each fluid is not enforced geometrically. However, the water behaviour is very similar to that observed in the PFEM computations with a single fluid (no air) and the free surface is modelled as a null pressure condition.

The typical velocities and pressures of this problem do not require treating the air as a compressible fluid, so the mass conservation equation imposes a divergence-free velocity field even for those parts of the domain that represent the air.

Neither the turbulence nor the surface tension between water and air were accounted for in this work, since they do not play a significant role at the mesh scale used, which is of the order of centimetres. As a result, small-scale effects, such as small bubbles that are formed and trapped by the turbulent flow of the water, cannot be modelled in full detail. Nonetheless, the numerical modelling with PFEM can be useful to obtain an estimation of the air flow demand for different situations, which is essential for designing the aeration system.

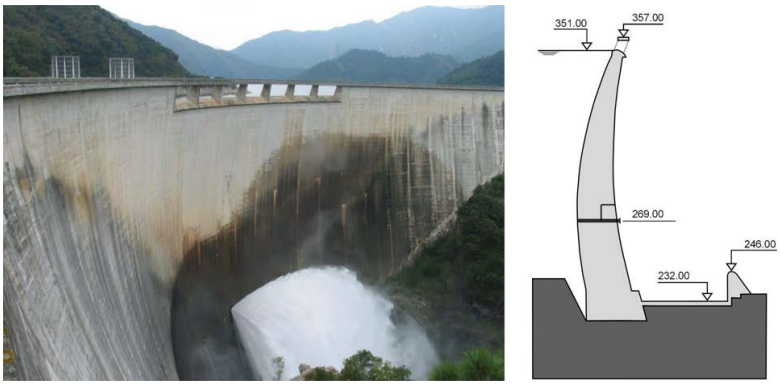


Fig. 6 Susqueda Dam. Left: view from the right abutment with one outlet in operation (courtesy of F.J. Conesa); Right: cross-section through a bottom outlet

3 Susqueda Dam case study

Susqueda Dam (Fig. 6) is located at the Ter river basin, in the north-eastern region of Spain. It is a double-curvature arch dam with a maximum height of m above foundation. The bottom outlets are situated at 37 m above the stilling basin. They comprise four round conduits of 1.5 m diameter. Each one is controlled by two identical flat-seat round-section valves [30]: that installed upstream serves as a guard gate, whereas the one in the downstream part is used for flow regulation.

This type of valves were developed for circular conduits to join the robustness of the conventional bonneted slide gates [10], [32], while avoiding the need for round-to-square upstream transition and square-to-round downstream transition [30]. Thus, the results of the conventional formulas and the design criteria for the aeration system need to be verified for its application to this typology.

The bottom outlets at Susqueda Dam feature a 4-m reach from the downstream face of the regulation gate to the downstream face of the dam (Fig. 7). A deflector was installed at the end of the conduit to improve energy dissipation by widening the flow impact area. It comprises a cone-shaped plate. The aeration system consists of a 0.4-m diameter air conduit, with a vertical 4-m long reach, a 90^{o} elbow, and a horizontal 3.5-m long reach which also ends at the downstream dam face (Fig. 7).

4 Numerical model set

4.1 Geometry and mesh

The numerical model reproduces one of the bottom outlets of Susqueda Dam. The upstream guard gate was not considered in the computation, since it remains fully open during normal operation. As a result, the model geometry comprised a 1.5 m diameter conduit, a partially-opened gate with 0.3 m thickness, and a 0.4 m

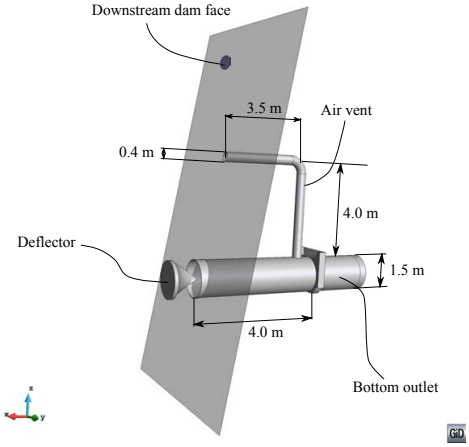
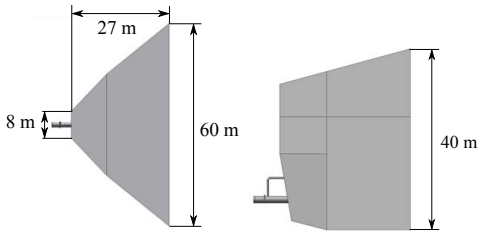


Fig. 7 Geometry of the bottom outlet in the numerical model. Perspective.

diameter air vent with a 90^{o} elbow. Fig. 7 depicts the more relevant aspects of the model geometry.

Both the outlet end section and the exit of the air vent are connected to a large volume, initially filled with air, which represents the domain at the downstream area of the dam body. Fig. 8 shows the main dimensions of the overall computational domain.



(a) Plane view (b) Side view

Fig. 8 Overall view of the computational domain, including the air surrounding the downstream face of the dam.

PFEM allows considering different mesh sizes within certain sub-domains. This is useful to optimise the computational resources, as a fine mesh can be used in the areas of interest, while other regions can be meshed with larger elements. In this implementation of the PFEM, the mesh size is defined by means of nested parallelepiped, sharing the same centre. They feature increasing mesh sizes in the inside-outside direction, from 0.07 to 1.6 m. Fig. 9 shows a detail of the different

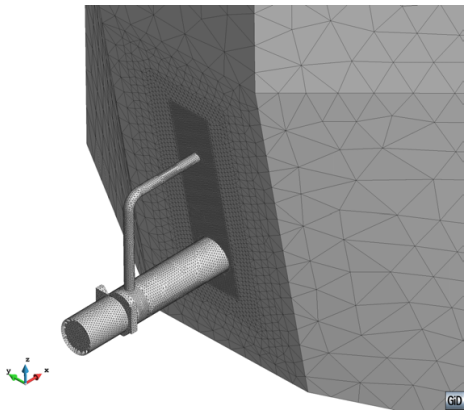


Fig. 9 Variable mesh size.

Table 1 Inflow discharge for each gate opening

Gate opening (%)	Inflow ($m^3 s^{-1}$)
25	10.94
50	20.73
75	33.84
80	36.19
90	40.59
100	44.18

mes sizes. The software GiD [29] was employed for geometry and mesh generation and results post-process.

4.2 Boundary conditions

In all cases, the upstream hydraulic head was set to 0.49 MPa, equivalent to 50 m of water. Since the study focused on the aeration system, the upstream water volume (from the upstream side of the gate to the reservoir free surface) was not considered. As an alternative, the incoming flow rate for each gate opening was computed by means of a separate numerical simulation (not described here), whose results are included in table 1.

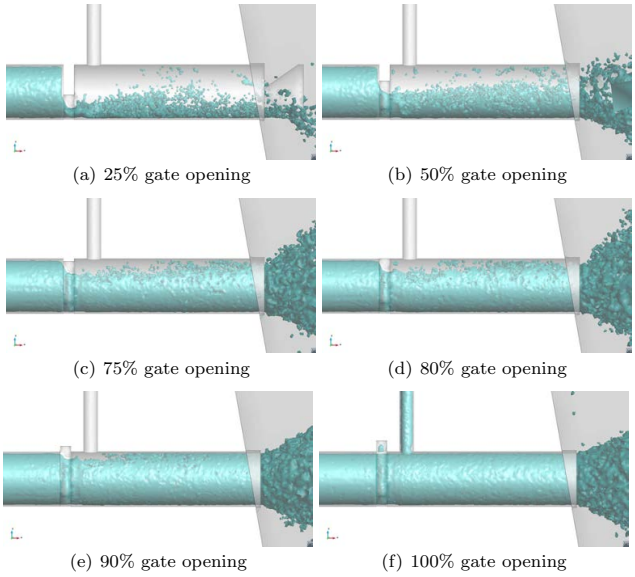
The downstream volume was limited by solid boundaries, except by a 12 x 5 m surface where zero pressure was imposed.

5 Results and discussion

5.1 Flow regime

The downstream water level is not relevant in Susqueda Dam, as the outlet exits at a sufficient height over the stilling basin so as to avoid drowning. However, the deflector introduces a localised head loss, which may modify the expected flow regime for a free discharge conduit. As a result, it was considered interesting to analyse the air-water interaction in the conduit, between the gate and the deflector. The results are presented in Fig. 10 for each gate opening.

As expected, the gate opening conditioned the air-water interaction. As it increased, a greater portion of the conduit was filled with water. The flow regime varied: it can be classified as spray flow for 25% opening; free flow for 50-80%, and foamy flow for 90%. For full gate opening, the water also flowed through the air vent, showing no air demand and a fully pressurised flow (Fig. 11). The latter behaviour was also observed on site, as depicted in Fig. 11. It should be noted that the water exits at low pressure through the air vent.



(a) 25% gate opening (b) 50% gate opening

(c) 75% gate opening (d) 80% gate opening

(e) 90% gate opening (f) 100% gate opening

Fig. 10 Flow regime in the outlet for different gate openings. Iso-surface of density = 500 kg/m^3. (Colour figure online).

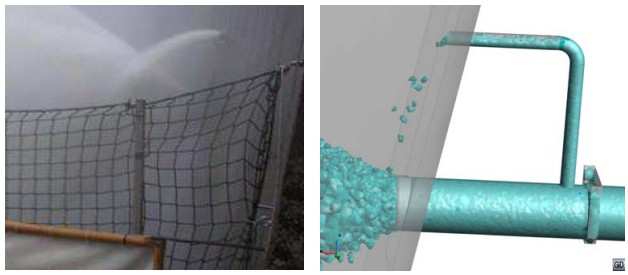


Fig. 11 Water flowing through the air vent for 100% gate opening. Left: image taken during the gate operation tests. Right: numerical model results. (Colour figure online).

5.2 Air flow pattern

The flow pattern of both fluids was analysed, both in the air vent and in the outlet. To that end, the velocity vectors were drawn on a longitudinal section of the model, as depicted in Fig. 12. The objective was to compare the velocity field for various gate openings to those defined by Sharma (Fig. 2).

It was observed that for small gate openings (up to 50%), the air demand was satisfied both from the air vent and from the downstream end of the outlet (it entered through the area around the deflector which is free from the water flow).

Between 75 and 90% gate opening, all the air was supplied by the air vent, since most of the outlet section was occupied by the water flow. In these cases, a recirculating area appeared at the top of the conduit, where part of the air flow was trapped by the high-velocity water flow at the bottom, while the mixture was evacuated around the deflector.

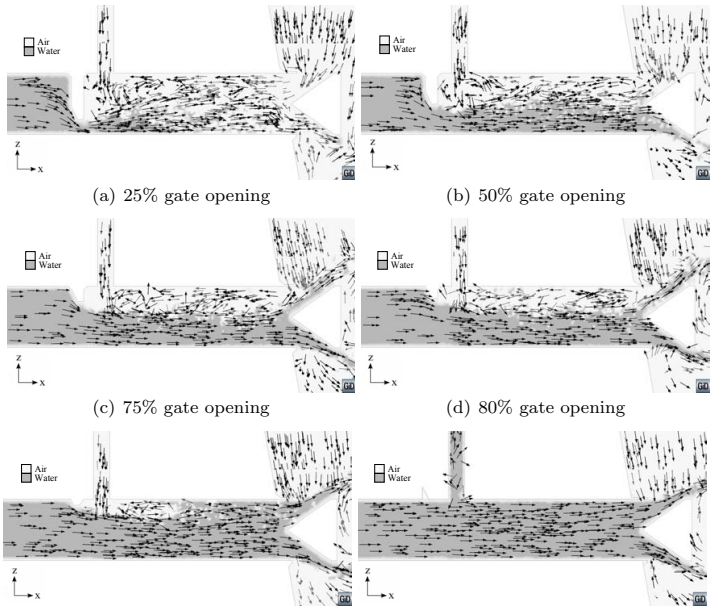
Finally, as mentioned in the previous section, the water completely filled the conduit for 100% gate opening. In this situation there was no air demand, and part of the water exited through the air vent towards the downstream face of the dam body with low pressure.

A more detailed analysis was carried out for 25, 50, 75 and 90 % gate opening. Fig. 13 shows results in four cross-sections between the gate and the deflector: a) the contour for 95% air concentration ($C = 0.95$), and b) the sign of the velocity in the x-axis direction. It can be seen that for small openings the water concentrated at the bottom central area, while part of the air was dragged towards the deflector ($v_x > 0$). By contrast, for large openings only the top-central zone featured $C \geq 95\%$.

In the Fig. 14 only the air nodes are plotted, separated by the sign of their x-velocity. The air flow from the deflector area for 25-50% gate opening can be clearly observed, as well as the small volume occupied by air for 90 % opening. The ratio of air nodes with positive x-velocity increased with large gate openings.

5.3 Air flow rate

The air vent design (axis and diameter) is mostly based on the maximum air flow. In the general case, the problem is coupled, i.e. the air demand depends



(a) 25% gate opening (b) 50% gate opening

(c) 75% gate opening (d) 80% gate opening

(e) 90% gate opening (f) 100% gate opening

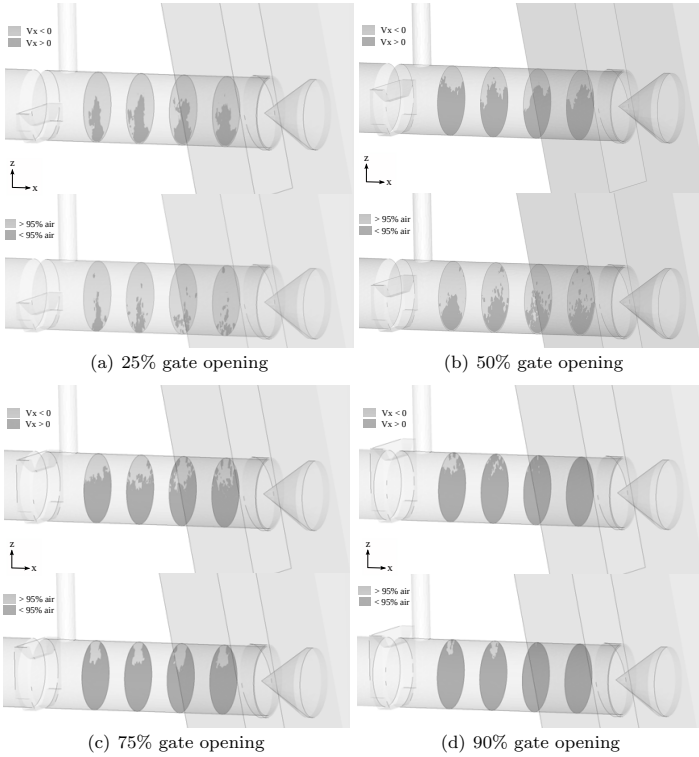
Fig. 12 Velocity vectors for different openings. Longitudinal section. The dark area depicts air concentration $\leq 50\%$

on the geometry of the air vent. In this particular case study, the design was already decided and built, and the numerical modelling was carried out as a further verification.

The air flow rate for each gate opening was computed by integrating the velocity field in a horizontal section of the air vent. The result was averaged over five seconds of simulation (from $t = 1s$ to $t = 6s$), once a pseudo-stationary regime was reached (in the beginning of the simulation, the incoming water flow pushes the air). The results are presented in Fig. 15.

It can be observed that the maximum air demand is registered for 80% gate opening. This value coincides with the practical criteria suggested by some authors (e.g. [36], [39], [4]). Nonetheless, it should be mentioned that according to some publications, the gate opening for the maximum air flow rate depends on the specific features of each facility. For instance, Tullis and Larchar reported a maximum air demand for 50% gate opening [38]. Their experimental study focused on slide gates installed on the upstream sloping face of embankment dams.

The velocity values (and consequently also the flow rate) showed high fluctuation for some gate openings along the period considered. This was also observed in experimental tests [38], and was attributed to the high turbulence and instability of the flow downstream of the gate. Fig. 15 also shows the standard deviation of the flow rate for each gate opening.



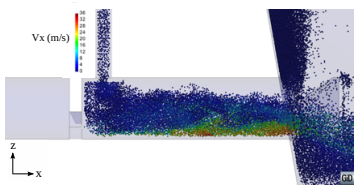
(a) 25% gate opening (b) 50% gate opening

(c) 75% gate opening (d) 90% gate opening

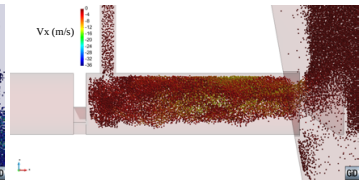
Fig. 13 Air-water interaction downstream of the air vent. Cross sections at 1.5, 2.5, 3.5 and 4.5 m from the gate. For each gate opening, the sign of the x-velocity (top) and the air concentration (bottom) are plotted. It should be noted that the darker region in the bottom figures represents the area occupied by the aerated flow (0-95% air concentration).

The maximum velocity in the air vent was also verified, as this is the fundamental parameter its design. Fig. 16 shows that the relation between gate opening and maximum velocity sensibly coincides with that for the mean air flow, although the absolute maximum was recorded for 90% gate opening: an instant value of 33 m/s was obtained. Since the typical recommendation is to limit the maximum velocity to 45 m/s ([39], [4]), it can be concluded that the aeration for Susqueda Dam was correctly dimensioned.

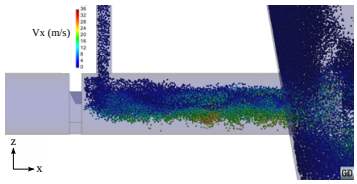
These results might seem contradictory as compared to those depicted in Fig. 13: the higher air demand coincides with gate openings for which the area occupied by water is greater. However, it should be noted that for small openings most of the air circulates with negative velocity in the x-axis, i.e., it comes from the area around the deflector. By contrast, for 75-90% gate opening all the air demand is



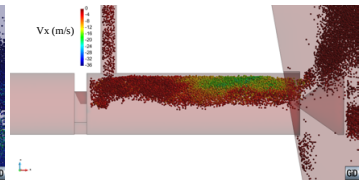
(a) 25% gate opening. Air nodes with positive x-velocity



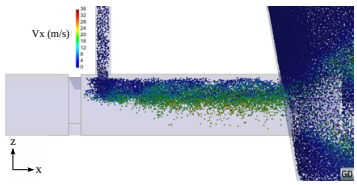
(b) 25% gate opening. Air nodes with negative x-velocity



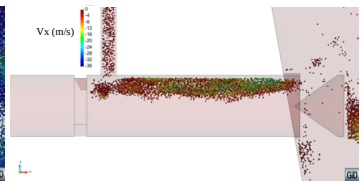
(c) 50% gate opening. Air nodes with positive x-velocity



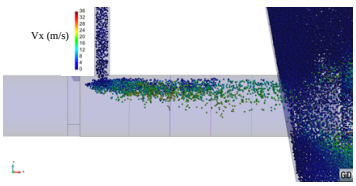
(d) 50% gate opening. Air nodes with negative x-velocity



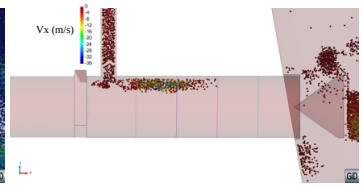
(e) 75% gate opening. Air nodes with positive x-velocity



(f) 75% gate opening. Air nodes with negative x-velocity



(g) 90% gate opening. Air nodes with positive x-velocity



(h) 90% gate opening. Air nodes with negative x-velocity

Fig. 14 Air nodes for 25, 50, 75 and 90% gate opening, separated by the sign of the x-velocity. Most of the air is supplied from the deflector area for small gate openings. (Colour figure online)

supplied by the air duct, the re-circulation area is smaller, and the air velocity is higher. These factors result in greater air flow through the air vent (Fig. 14).

6 Summary and conclusions

The PFEM was applied to simulate the air-water interaction in the downstream reach of Susqueda Dam bottom outlet. Different gate openings were analysed for

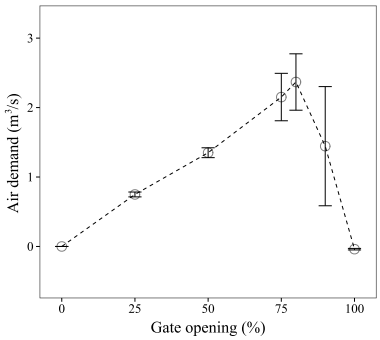


Fig. 15 Air flow rate as a function of the gate opening. The bars show the standard deviation. The maximum air demand was registered for 80% opening.

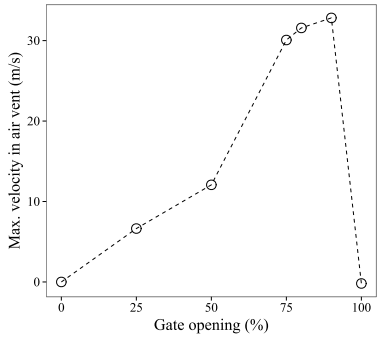


Fig. 16 Maximum instant air flow velocity in the vent as a function of the gate opening.

a constant upstream head. In the PFEM, the information is stored at the nodes of the mesh, which is regenerated at every time step. Thus, air-water mixture can be naturally modelled by means of mixed elements (formed by nodes from air and water).

The phenomenon is highly complex, with a broad range of air and water particle sizes. Although the direct simulation of every droplet and bubble is infeasible, PFEM allows considering the air drag produced by the high-velocity water flow, as well as analysing the air-water flow pattern.

The numerical results were analysed and compared to previous studies and common design criteria. The main conclusions that can be drawn from that analysis are:

– Some of the flow regimes defined by Sharma [35] were observed for increasing gate openings: from spray flow with 25% opening to fully pressurised flow for 100% opening. Free and foamy flows were also observed for intermediate gate openings.

– The maximum air demand on average was recorded for 80% gate opening, which is in accordance with the reference studies and design guides ([38], [39], [36]).
– For full opening, the conduit is fully pressurised and part of the water flow fills the air vent. Thus, the air demand is null in this situation. This behaviour was observed on site during the operation tests (Fig. 11).

The results suggest that the PFEM can be useful for calculating the air demand in dam bottom outlets. In the case study presented, the appropriateness of the existing design was verified. For new facilities, the possibilities of the PFEM for identifying the flow patterns, and for computing the pressure and velocity fields should be helpful for designing the aeration system.

Acknowledgements The authors thank Felipe Río and Francisco J. Conesa, from ENDESA GENERACION, for supplying the information about Susqueda Dam, and Francisco Riquelme for promoting this research. It was carried out with financial support received from the FLOODSAFE project funded by the Proof of Concept Program of the European Research Council.

References

1. The Particle Finite Element Method - PFEM. URL http://www.cimne.com/pfem/
2. ANSYS Inc.: Ansys Fluent theory guide (2011)
3. Aubry, R., Idelsohn, S., Oñate, E.: Particle finite element method in fluid-mechanics including thermal convection-diffusion. Computers & Structures **83**(17–18), 1459–1475 (2005)
4. Campbell, F., Guyton, B.: Air demand in gated conduits. In: IAHR Symposium, Minneapolis (1953)
5. Carbonell, J.M., Oñate, E., Suárez, B.: Modeling of ground excavation with the particle finite-element method. Journal of Engineering Mechanics **136**(4), 455–463 (2009)
6. Chanson, H.: Hydraulics of aerated flows: qui pro quo? Journal of Hydraulic Research **51**(3), 223–243 (2013)
7. Cheng, S.W., Dey, T.K., Shewchuk, J.: Delaunay mesh generation. CRC Press (2012)
8. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. ACM Transactions on Graphics **13**(1), 43–72 (1994)
9. Erbisti, P.C.: Design of hydraulic gates. CRC Press (2014)
10. FEMA: Outlet works energy dissipators. Tech. Rep. P–679 (2010)
11. Frizell, K.: Hydraulic model studies of aeration enhancements at the folsom dam outlet works: Reducing cavitation damage potential. Water Operation and Maintenance Bulletin (185) (2004)
12. Hirt, C.W.: Modeling turbulent entrainment of air at a free surface. Flow Science, Inc (2003)
13. Idelsohn, S., Mier-Torrecilla, M., Oñate, E.: Multi-fluid flows with the particle finite element method. Computer Methods in Applied Mechanics and Engineering **198**(33), 2750–2767 (2009)
14. Idelsohn, S., Nigro, N., Limache, A., Oñate, E.: Large time-step explicit integration method for solving problems with dominant convection. Computer Methods in Applied Mechanics and Engineering **217**, 168–185 (2012)
15. Idelsohn, S., Oñate, E., Del Pin, F.: A lagrangian meshless finite element method applied to fluid-structure interaction problems. Computers and Structures **81**(8), 655–671 (2003)
16. Idelsohn, S., Oñate, E., Pin, F.D.: The particle finite element method: a powerful tool to solve incompressible flows with free–surfaces and breaking waves. International Journal for Numerical Methods in Engineering **61**(7), 964–989 (2004)
17. Idelsohn, S.R., Marti, J., Becker, P., Oñate, E.: Analysis of multifluid flows with large time steps using the particle finite element method. International Journal for Numerical Methods in Fluids **75**(9), 621–644 (2014)

18. Larese, A., Rossi, R., Oñate, E., Idelsohn, S.: Validation of the particle finite element method (pfem) for simulation of free surface flows. Engineering Computations: Int J for Computer-Aided Engineering **25**(4), 385–425 (2008)
19. Larese, A., Rossi, R., Oñate, E., Idelsohn, S.R.: Validation of the particle finite element method (PFEM) for simulation of free surface flows. Engineering Computations **25**(4), 385–425 (2008)
20. Liu, T., Yang, J.: Three-dimensional computations of water–air flow in a bottom spillway during gate opening. Engineering Applications of Computational Fluid Mechanics **8**(1), 104–115 (2014)
21. Oñate, E., Idelsohn, S.R., Celigueta, M.A., Rossi, R.: Advances in the particle finite element method for the analysis of fluid–multibody interaction and bed erosion in free surface flows. Computer Methods in Applied Mechanics and Engineering **197**(19), 1777–1800 (2008)
22. Oñate, E., Rossi, R., Idelsohn, S.R., Butler, K.M.: Melting and spread of polymers in fire with the particle finite element method. International Journal for Numerical Methods in Engineering **81**(8), 1046–1072 (2010)
23. Oñate, E., Celigueta, M.A., Idelsohn, S.R.: Modeling bed erosion in free surface flows by the particle finite element method. Acta Geotechnica **1**(4), 237–252 (2006)
24. Oñate, E., Celigueta, M.A., Idelsohn, S.R., Salazar, F., Suárez, B.: Possibilities of the particle finite element method for fluid–soil–structure interaction problems. Computational Mechanics **48**(3), 307–318 (2011)
25. Oñate, E., Franci, A., Carbonell, J.M.: 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 (2014)
26. Oñate, E., Franci, A., Carbonell, J.M.: A particle finite element method for analysis of industrial forming processes. Computational Mechanics **54**(1), 85–107 (2014)
27. Oñate, E., Idelsohn, S.: The particle finite element method. an overview. International Journal of Computational Methods **1**, 267–307 (2004)
28. Pozo, D., Salazar, F., Toledo, M.: Modeling the hydraulic performance of the aeration system in dam bottom outlets using the particle finite element method. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería **30**(1), 51–59 (2014). [in Spanish]
29. Ribó, R., Pasenau, M., Escolano, E., Ronda, J., González, L.: GiD reference manual. CIMNE, Barcelona (1998)
30. Riquelme, F., Morán, R., Salazar, F., Celigueta, M., Oñate, E.: Flat-seat round-section valves: design criteria and performance analysis via numerical modelling. In: Dam Maintenance and Rehabilitation II, pp. 609–615. CRC Press (2011). [in Spanish]
31. Safavi, K., Zarrati, A., Attari, J.: Experimental study of air demand in high head gated tunnels. In: Proceedings of the Institution of Civil Engineers-Water Management, vol. 161, pp. 105–111. Thomas Telford Ltd (2008)
32. Sagar, B.: Asce hydrogates task committee design guidelines for high-head gates. Journal of Hydraulic Engineering **121**(12), 845–852 (1995)
33. Salazar, F., Irazábal, J., Larese, A., Oñate, E.: 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 (2015)
34. Salazar, F., Oñate, E., Morán, R.: Numerical modelling of landslides in reservoirs via the particle finite element method (PFEM). Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería **28**(2), 112–123 (2012). [in Spanish]
35. Sharma, H.R.: Air-entrainment in high head gated conduits. Journal of the Hydraulics Division **102**(11), 1629–1646 (1976)
36. Spanish National Comittee on Large Dams (SPANCOLD): Dam Safety Technical Guide n. 5 "Spillways and Bottom Outlets". Colegio de Ingenieros de Caminos, Canales y Puertos (1997). [in Spanish]
37. Toombes, L., Chanson, H.: Free-surface aeration and momentum exchange at a bottom outlet. Journal of Hydraulic Research **45**(1), 100–110 (2007)
38. Tullis, B.P., Larchar, J.: Determining air demand for small-to medium-sized embankment dam low-level outlet works. Journal of Irrigation and Drainage Engineering **137**(12), 793–800 (2011)
39. USACE: Hydraulic Design Criteria, Air Demand, Regulated Outlet Works. US Army Corps of Engineer (1964)

40. Valero, D., García-Bartual, R.: Calibration of an air entrainment model for cfd spillway applications. In: Advances in Hydroinformatics, pp. 571–582. Springer (2016)
41. Vischer, D., Hager, W.H., Cischer, D.: Dam hydraulics. Wiley Chichester, UK (1998)
42. Wright, N., Tullis, B.: Prototype and laboratory low-level outlet air demand comparison for small-to-medium-sized embankment dams. Journal of Irrigation and Drainage Engineering **140**(6), 04014,013 (2014)
43. Zienkiewicz, O.C., Taylor, R.L.: The finite element method for solid and structural mechanics. Butterworth-Heinemann (2005)