



An extended validation of the last generation of particle finite element method for free surface flows [☆]



Juan M. Gimenez ^{a,*}, Leo M. González ^b

^a Centro de Investigación de Métodos Computacionales (CIMEC), UNL/CONICET, Predio Conicet-Santa Fe, Colectora Ruta Nac 168, Paraje El Pozo, Santa Fe, Argentina

^b Escuela Técnica Superior de Ingenieros Navales, Universidad Politécnica de Madrid (ETSIN-UPM), Avd. Arco de la Victoria 4, Madrid, Spain

ARTICLE INFO

Article history:

Received 20 May 2014

Received in revised form 9 October 2014

Accepted 11 December 2014

Available online 19 December 2014

Keywords:

PFEM

PFEM-2

Free surface flows

Finite elements

Large time-steps

Enrichment

ABSTRACT

In this paper, a new generation of the particle method known as Particle Finite Element Method (PFEM), which combines convective particle movement and a fixed mesh resolution, is applied to free surface flows. This interesting variant, previously described in the literature as PFEM-2, is able to use larger time steps when compared to other similar numerical tools which implies shorter computational times while maintaining the accuracy of the computation. PFEM-2 has already been extended to free surface problems, being the main topic of this paper a deep validation of this methodology for a wider range of flows. To accomplish this task, different improved versions of discontinuous and continuous enriched basis functions for the pressure field have been developed to capture the free surface dynamics without artificial diffusion or undesired numerical effects when different density ratios are involved. A collection of problems has been carefully selected such that a wide variety of Froude numbers, density ratios and dominant dissipative cases are reported with the intention of presenting a general methodology, not restricted to a particular range of parameters, and capable of using large time-steps. The results of the different free-surface problems solved, which include: Rayleigh–Taylor instability, sloshing problems, viscous standing waves and the dam break problem, are compared to well validated numerical alternatives or experimental measurements obtaining accurate approximations for such complex flows.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Every day, hybrid methods gain traction in computational fluid mechanics, where the Lagrangian framework given by a particle method, is combined with a Eulerian methodology. In these hybrid methods, a fixed or reconstructed grid supports part of the pressure and velocity calculation. The original idea, proposed by Monaghan [1] and later works applied to fluid mechanics [2], where a pure Lagrangian perspective was used during the whole meshless computation, has been in some cases completed using other well known discretization methods, such as FVM [3] or FEM [4]. The first combination of Lagrangian and FEM methods can be found in [5], where an extended Delaunay Tessellation is used to reconstruct the mesh

[☆] Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author.

E-mail address: jmarcelogimenez@gmail.com (J.M. Gimenez).

while the fluid evolves. In this method, known as MFEM, the construction of the shape functions inside each polyhedron is based on a non-Sibsonian interpolation.

The next step in this evolution was the first version of the PFEM method [6], which was a robust method designed to solve fluid–structure interaction problems including free-surface, breaking waves, flow separations, etc. where Lagrangian particles and meshing processes are alternated with the advantage of having a FEM structure that supports the differential equation solvers. An interesting difference between the PFEM and other hybrid methods as PIC (particle-in-cell) [7], MAC (marker-and-cell) [8] and MPM (material point method) [9] is that while in latter methods the particles transport mass and consequently have a volume, PFEM particles are non-material points that transport the fixed intensive properties of the fluid.

Other methods, that also combine both Eulerian and Lagrangian perspectives, are the arbitrary Eulerian–Lagrangian (ALE) [10] or semi-Lagrangian methods [11]. The Lagrangian perspective makes it possible to use a material derivative formulation where the absence of the non-linear convective terms transform the Navier–Stokes system into a transformed linear coupled problem. Methodologies, such as the backward Characteristics method [11], also give this possibility but if the process is done in a fixed mesh without any distortion, and unless high order polynomials are used, a dissipative process appears due to the interpolation of the feet of characteristics.

In contrast to the backward characteristic method where the feet of the characteristic line was searched and located in a mesh element based on the known velocity fields at past time steps, a new strategy known as X-IVAS (eXplicit Integration following the Velocity and Acceleration Streamlines) was developed by Idelsohn et al. [12]. This methodology of integrating the convection of fluid particles is based on following the streamlines of the flow in the current time step instead of the particle trajectories, which represents an alternative way to solve the non-linearities of the flow equations. Adding this strategy to the original PFEM method, a new methodology appears called Particle Finite Element Method Second Generation (PFEM-2) [13]. The X-IVAS strategy gives the possibility of solving complex flows with large time steps ($CFL > 1$), as well as the presence of the mesh allows for accurate solutions of the fractional step method.

In PFEM-2, there are two approaches to communicate particle and mesh data, each one generating two versions of the method. The first one is called *Moving Mesh*, which follows the original idea of PFEM, creating a new mesh using the new position of the particles as nodes. The second version, named *Fixed Mesh*, projects the particle states to nodes while preserving the initial background mesh. The former strategy maintains the uncomfortable drawback of previously cited methods, which leads to the necessity of constructing or controlling the mesh quality during the simulation if the accuracy of the solution has to be maintained. The evaluation of the mesh distortions or the re-meshing processes are always computationally expensive and it would be interesting to explore the possibility of avoiding that step. Consequently, the *Fixed Mesh* approach avoids the remeshing at each time-step. In this approach, mesh nodes and moving particles interchange information through different interpolation algorithms. In the context of this paper, PFEM-2 will refer to the *Fixed Mesh* version, and a detailed explanation of this algorithm is given in Section 2.

The extension of PFEM-2 to multi-fluids flows is presented by Idelsohn et al. [14] where novel features for, mainly, the treatment of the interface evolution are explained. In that work, the treatment of the free-surface has been done simulating both fluids that share the interface using a scalar function to identify each fluid, and, to improve the pressure calculation close to the free surface, a discontinuous enrichment technique is employed. However, the reference has a lack of validation because only non-viscous fluids with high density ratios were analyzed, skipping the free surface numerical problems derived when viscosity plays a dominant role and/or the density ratio is moderate.

In the current work, the mentioned PFEM-2 for multi-fluids strategy has been used to solve a wider range of free-surface flows, starting from classical benchmark problems, such as the Rayleigh–Taylor instability (Section 4.1), and finishing with problems of industrial interest, such as sloshing in closed containers (Section 4.2) or dam-break cases (Section 4.4), where the obtained results must be validated with numerical or experimental data. However, to solve all the range of Froude number situations, a novel continuous version of the enrichment methodology is presented. It can be used on general cases in order to avoid the problems encountered particularly in high Froude number flows where other versions present poor accuracy. The obtained results have been also compared to other well reputed Eulerian codes, obtaining accurate and numerically stable results while using larger time steps. Taking into account that the computational time per time-step is comparable to the one required by the other methodologies, the possibility of increasing the time-step implies shorter global computational times.

2. PFEM-2 algorithm

The aim of this section is to present a brief review of the efficient and accurate methodology, called PFEM-2, to numerically simulate the dynamics of an incompressible flow. Readers interested in more details of the method may see [12,13,15].

The governing equations are the incompressible Navier–Stokes equations, which are supplemented with the conventional boundary conditions on solid and/or open boundaries. The computational domain Ω contains both fluids, the first one, denoted by subscript 1, and the second one with its corresponding variables denoted by the subscript 2 with densities and viscosities ρ_i and μ_i ($i = 1, 2$), respectively. The governing equations, written in a Lagrangian framework, are:

$$\nabla \cdot \mathbf{v} = 0 \quad (1)$$

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f} \quad (2)$$

As expected, the convective term does not appear explicitly in this Lagrangian formulation but a kinematic problem has to be solved at each time step. Here \mathbf{v} , p are the velocity and fluid pressure and \mathbf{f} is an external body force (normally gravity $\rho\mathbf{g}$ and/or inertial force).

In order to decouple the unknown fields: velocity and pressure, the projection method known as fractional step were implemented in PFEM-2. This segregated strategy consist on three main steps: predictor, Poisson equation and correction.

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

1. Predictor step, which solves the momentum equation to predict a velocity $\hat{\mathbf{v}}^{n+1}$ on the fixed mesh

(a) Acceleration Stage: Calculates acceleration components: \mathbf{a}_τ (viscous component) and \mathbf{a}_p (pressure component) on the mesh nodes.

$$\int_{\Omega} \mathbf{a}_\tau^n \psi_j d\Omega = \int_{\Omega} \mu \nabla^2 \mathbf{v}^n \psi_j d\Omega = - \int_{\Omega} \mu \nabla \mathbf{v}^n \nabla \psi_j d\Omega + \int_{\Gamma} \mu \nabla \mathbf{v}^n \psi_j \cdot \mathbf{n} d\Gamma \quad (3)$$

$$\int_{\Omega} \mathbf{a}_p^n \psi_j d\Omega = - \int_{\Omega} \nabla p^n \psi_j d\Omega \quad (4)$$

$$\mathbf{a}^n = \mathbf{a}_p^n + (1 - \theta) \mathbf{a}_\tau^n \quad (5)$$

Where θ is a numerical parameter that rules the explicitness of the viscous term in the algorithm and \mathbf{n} is the unitary vector normal to the surface.

(b) X-IVAS Stage: Evaluates the new particle position \mathbf{x}_p^{n+1} and intermediate velocity $\hat{\mathbf{v}}_p^{n+1}$ following the velocity streamlines at t^n

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \int_n^{n+1} \mathbf{v}^n(\mathbf{x}_p^t) dt \quad (6)$$

$$\hat{\mathbf{v}}_p^{n+1} = \mathbf{v}_p^n + \int_n^{n+1} [\mathbf{a}^n(\mathbf{x}_p^t) + \mathbf{f}(\mathbf{x}_p^t)] dt \quad (7)$$

where \mathbf{f} are the external body forces. X-IVAS is presented in [12] and [13] as a strategy which significantly improves the particle trajectory integration by following streamlines. This method is able to resolve difficult details of the flow with high accuracy without a drastic time-step reduction, specially when compared to standard Lagrangian integration schemes. The temporal integration along the streamlines can be solved using analytical expressions [12] or high-order integrators [16]. However, in this work, a sub-stepping integrator inherited from STS [17] is used, which can adapt its sub-step $\delta t = \frac{\Delta t}{K \cdot CFL_h}$ depending on the local CFL number defined as $CFL_h = \frac{|\mathbf{v}| \Delta t}{h}$ and a parameter K that adjusts the minimal number of sub-steps required to cross an element. According to this N sub-stepping integration, where $N \delta t = \Delta t$, Eqs. (6) and (7) can be written as:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \sum_{i=1}^N \mathbf{v}^n(\mathbf{x}_p^{n+\frac{i}{N}}) \delta t \quad (8)$$

$$\hat{\mathbf{v}}_p^{n+1} = \mathbf{v}_p^n + \sum_{i=1}^N [\mathbf{a}^n(\mathbf{x}_p^{n+\frac{i}{N}}) + \mathbf{f}(\mathbf{x}_p^{n+\frac{i}{N}})] \delta t \quad (9)$$

(c) Projection Stage: Projects velocity from the particles onto the mesh nodes:

$$\hat{\mathbf{v}}_j^{n+1} = \frac{\sum_p \hat{\mathbf{v}}_p^{n+1} W(\mathbf{x}_j - \mathbf{x}_p^{n+1})}{\sum_p W(\mathbf{x}_j - \mathbf{x}_p^{n+1})} \quad (10)$$

Where the functions W are the typical kernel functions used in particle methods such as SPH [1]. Summations are extended to the particles p within a critical distance that depends on the election of the kernel function. For the computations presented in this paper, the Wendland kernel function [18] was used for the projections.

(d) Implicit Viscosity Stage: Implicit correction of the viscous diffusion. The fractional velocity $\hat{\mathbf{v}}_j^{n+1}$ is found on the mesh nodes.

$$\int_{\Omega} \hat{\mathbf{v}}_j^{n+1} \psi_j d\Omega = \int_{\Omega} \hat{\mathbf{v}}_j^{n+1} \psi_j d\Omega + \theta \Delta t \int_{\Omega} \mu \nabla^2 \hat{\mathbf{v}}_j^{n+1} \psi_j d\Omega \quad (11)$$

2. Poisson Stage: Computes the pressure correction δp^{n+1} on the mesh nodes by solving the Poisson equation.

$$\int_{\Omega} \nabla \cdot \left[\frac{\Delta t}{\rho} \nabla (\delta p^{n+1}) \right] \phi_j d\Omega = \int_{\Omega} \nabla \cdot \hat{\mathbf{v}}_j^{n+1} \phi_j d\Omega \quad (12)$$

$$\frac{\partial \delta p^{n+1}}{\partial n} = 0 \quad \text{in } \Gamma_D \quad (13)$$

$$\delta p^{n+1} = 0 \quad \text{in } \Gamma_N \quad (14)$$

This problem should be stabilized if the P1–P1 FEM formulation is used [13]. The pressure at time t_{n+1} is updated as $p^{n+1} = p^n + \delta p^{n+1}$.

3. Correction Stage: Updates the mesh and particle velocity with the pressure and diffusion corrections:

$$\int_{\Omega} \rho_j \mathbf{v}_j^{n+1} \psi_j d\Omega = \int_{\Omega} \rho_j \hat{\mathbf{v}}_j^{n+1} \psi_j d\Omega - \Delta t \int_{\Omega} \nabla \delta p^{n+1} \psi_j d\Omega \quad (15)$$

$$\rho_p \mathbf{v}_p^{n+1} = \rho_p \hat{\mathbf{v}}_p^{n+1} + \sum_j \delta \mathbf{v}_j^{n+1} \psi_j(\mathbf{x}_p^{n+1}) \quad (16)$$

where $\delta \mathbf{v}_j^{n+1} = \mathbf{v}_j^{n+1} - \hat{\mathbf{v}}_j^{n+1}$.

Summarizing, steps 1.a, 1.d, 2 and 3 take place on the mesh, only step 1.b is entirely done over the particles, while steps 1.c and 3 involve transference of information between nodal and particle data.

3. Free-surface flows treatment

An accurate and efficient simulation of the interface evolution is crucial in the simulation of free-surface flows. During the flow evolution, it is essential that the interface remains sharp. Large jumps of fluid density and viscosity across the interface should be correctly assumed by the numerical algorithm in order to satisfy the momentum balance at the vicinity of the interface.

There are several, albeit small, differences between the PFEM-2 algorithm for homogeneous flows and the two fluid version. Those differences stem from the density and viscosity discontinuities that appear in the fluid, consequently most of the implemented changes are related to the strategies followed to correctly capture the interface between both fluids. Although some details presented in this section have already been reported in [14], important strategies that significantly improve the accuracy and efficiency of the computation were added in this work.

Taking into account the methodology presented in Section 3.4, important considerations to manage and improve each one of the stages for the particular case of multiphases problems are presented hereafter. The following part is related with three aspects of the simulation, namely: the kinematic treatment of the fluid particles during the X-IVAS stage, the enrichment technique for the free-surface definition, and the pressure computation step. Although these three topics are listed independently, they are closely related to each other during the computation and, consequently, will be treated together in the next section.

3.1. Internal interfaces tracking

When two different fluids separated by an interface are considered, each particle p carries the information of the fluid to which it was initially assigned. This quantity, represented by a scalar function λ_p , assumes integer values -1 or 1 depending if it belongs to the first or second fluid. This value is advected, adding one equation to the *Streamline Integration Stage*: $\frac{D\lambda}{Dt} = 0$, i.e. each particle keeps its marker value during the entire simulation. This function is projected to the mesh nodes to determine the free-surface position. Mesh nodes thus obtain real values after the projection which are different to the integer values ± 1 that the particles transport. The free-surface interface is defined as the set of points that satisfy the equation $\lambda = 0$, and a Heaviside function is used to determinate intensive properties (density and viscosity) on the nodes.

In general, the movement of a particle is done by sub-steps according to Eq. (8). The velocity used in the particle movement at position \mathbf{x}_p is calculated by the equation:

$$\mathbf{v}(\mathbf{x}_p) = \frac{\sum_i \mathbf{v}_i^n \psi_i(\mathbf{x}_p)}{\sum_i \psi_i} \quad (17)$$

where the nodes included in the interpolation are the nodes of the hosting element. Two situations could happen to any particle when its position changes: all the nodes of the hosting element have the same density as the fluid particle or one or more nodes have a different density than the fluid particle. Whilst in the first case a typical finite element interpolation is performed, the second situation clearly represents a case where the fluid particle is close to the interface. In the particular case that the density ratio ρ_1/ρ_2 is larger than a first numerical parameter α , that is $\rho_1/\rho_2 > \alpha$, two situations can appear:

- $\rho_p = \rho_2$ (light particle tracking). The velocity used in the particle movement will be computed using Eq. (17).
 - $\rho_p = \rho_1$ (heavy particle tracking). Depending on the value of $A = \sum_{i(\rho_i=\rho_p)} \psi_i$ where the sum is limited to the hosting nodes that have the same density as the particle, we can have 2 possibilities:
 - $A < \beta$ the gravity force will be included in the computation of the particle trajectory, which will finally be computed as a parabolic motion.
 - $A > \beta$, the sums that appear in Eq. (17) are both restricted to the hosting nodes i that have the same density as the particle $\rho_i = \rho_p$.
- being β a second numerical parameter, independent on the density ratio, which regulates the minimum amount of information needed to move a heavy particle following the streamlines of its own phase.

This means that if a water particle is momentarily in an air regime, it will remain as a water particle for further determination of the interface position. While a better approximation to the real particle trajectory defined by the acting forces is searched, the parabolic motion is used herein as the simplest trajectory when only gravity forces are acting, an interesting alternative for the particle motion (not used in this work) could be using a water droplet drag model. It must be noticed that for similar densities $\rho_1/\rho_2 < \alpha$, Eq. (17) is used on each one particle.

Once the particles have been advected and the intermediate velocities have been determined, this updated velocity and density information has to be incorporated into the mesh nodes. During this *Projection Stage*, each mesh node j updates its intermediate velocity according to Eq. (10), analogously a similar equation is used to update the values of λ_j . Depending on the value of λ_j the instantaneous local interface inside each element is determined as the iso-line (an iso-plane in 3D) where $\lambda(\mathbf{x}) = 0$.

3.2. Shape function enrichments for pressure gradient discontinuity capturing

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

- pressure gradient gaps where density discontinuities are present [19]
- pressure gaps where viscosity discontinuities are present [20]
- gradient velocity gaps where viscosity discontinuities are present
- pressure gaps where surface tension is present

For the case of interest presented in this work where two different density fluids are simulated, the interpolation errors in the pressure and its gradient give rise to spurious velocities that can render the solution meaningless.

Enrichment methods add degrees of freedom to elements that are cut by the interface in order to reduce interpolation errors. In this work, two space enrichment methodologies are proposed in order to treat pressure gradient discontinuities.

The first enriched space is based on the one presented by Coppola [19], which is illustrated, for the two-dimensional case, in Fig. 1. The triangle conformed by the nodes at positions \mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3 is cut by the interface at points \mathbf{x}_A and \mathbf{x}_B , dividing the element into two regions Ω_1 and Ω_2 . The construction of the enrichment function ϕ^* must satisfy

$$\begin{cases} \phi^*(\mathbf{x}_A) = 1 \\ \phi^*(\mathbf{x}_1) = \phi^*(\mathbf{x}_2) = \phi^*(\mathbf{x}_3) = 0 \end{cases} \quad (18)$$

where \mathbf{x}_A is a point over the edge $\mathbf{x}_1 - \mathbf{x}_2$ such as $\lambda(\mathbf{x}_A) = 0$.

It can be demonstrated [19] that the enrichment function, which accomplishes the previously presented restrictions, can be expressed as a linear combination of the traditional shape functions, being:

$$\phi^*|_{\Omega_2} = k_1 \phi_1 \quad (19)$$

$$\phi^*|_{\Omega_1} = k_2 \phi_2 + k_3 \phi_3 \quad (20)$$

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

The second set of enrichment functions used in this work is described in Fig. 2. As in the previous case, the two new degrees of freedom can also be statically condensed. However, using this new space, it is possible to ensure continuity between elements at the cost of having to rebuild the system matrix at each time step, which could be an expensive task due to memory allocation. This enrichment space is constructed following

$$\begin{cases} \phi_A^*(\mathbf{x}_A) = \phi_B^*(\mathbf{x}_B) = 1 \\ \phi_A^*(\mathbf{x}_1) = \phi_A^*(\mathbf{x}_2) = \phi_A^*(\mathbf{x}_3) = \phi_A^*(\mathbf{x}_B) = 0 \\ \phi_B^*(\mathbf{x}_1) = \phi_B^*(\mathbf{x}_2) = \phi_B^*(\mathbf{x}_3) = \phi_B^*(\mathbf{x}_A) = 0 \end{cases} \quad (21)$$

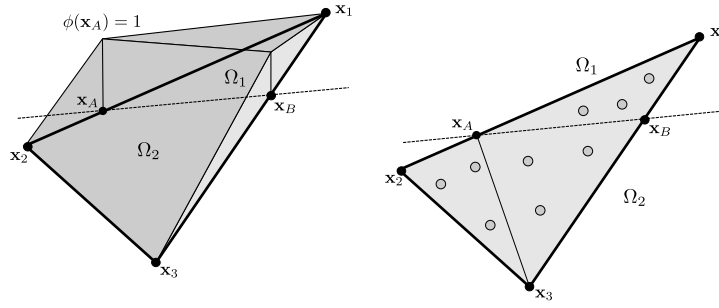


Fig. 1. 2D interface element. The interface is calculated cutting the element at the segment $A - B$. The enrichment proposed by Coppola (left) and the partition of the triangle into three sub-triangles with its own Gauss points (right).

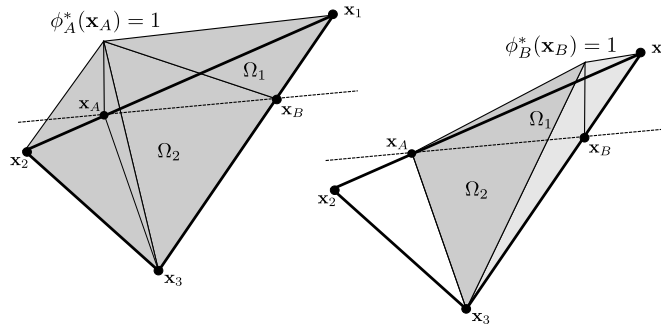


Fig. 2. 2D interface element. The interface is calculated cutting the element at the segment $A - B$. An enrichment space with two functions per interface element, which can be used to ensure continuity between elements, is presented. The integration partition is the same as presented above (Fig. 1).

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

$$p_h(\mathbf{x}) = \sum_{i=1}^{N_n} \phi_i(\mathbf{x}) p_i + \sum_{i=1}^{N_e} \phi_i^*(\mathbf{x}) p_i^* \quad (22)$$

where ϕ_i are traditional linear shape functions (a total of N_n), and ϕ_i^* are the enrichment shape functions (a total of N_e).

In order to capture the discontinuities and taking advantage of the enrichment functions used, the integration rules need to be modified in elements cut by the free surface. The method used is to divide each tetrahedra (triangles when represented in 2D) element into up to six tetrahedral (three triangular in 2D) sub elements. For each sub element, the same integration rule as for the non-cut elements is used. Fig. 1 shows such a partition where the small circles represent the Gauss points for the integration. When using enrichment functions for the pressure, the material properties ρ , μ are taken as ρ_1 , μ_1 or ρ_2 , μ_2 , depending on which part of the domain (Ω_1 or Ω_2) the integration point is found.

3.3. Pressure calculation

Particles can move across several elements and cross the interface during a time-step, consequently, the pressure gradient of the previous time-step would introduce a poor and even unstable approximation of the new pressure forces. In order to avoid these large errors in the evaluation of the pressure gradients in the initial value of the iterative process, the value of the pressure is set to zero at the beginning of each time-step. Therefore, the acceleration over the particle calculated in X-IVAS stage is only due to the gravitational force (the pressure gradient is considered null and viscous forces are treated implicitly) [14].

Starting from Eq. (12), and following the classical variational formulation for the global domain Ω with boundary Γ , through integration weighting with test functions and weakening both pressure Laplacian and velocity divergence terms, it is possible to obtain the system:

$$\left[\Delta t \int_{\Omega} \frac{1}{\rho} \nabla \phi^T \nabla \phi d\Omega \right] \delta p^{n+1} = \left[\int_{\Omega} \nabla \phi^T \psi d\Omega \right] \hat{\mathbf{v}}_j^{n+1} \quad (23)$$

where the term resulting from weakening

$$\int_{\Gamma} \phi \left[\hat{\mathbf{v}}_j^{n+1} + \Delta t \frac{1}{\rho} \nabla \delta p^{n+1} \right] \cdot \mathbf{n} d\Gamma = \int_{\Gamma} \phi \mathbf{v}_j^{n+1} \cdot \mathbf{n} d\Gamma \quad (24)$$

is zeroed to ensure the impenetrability on walls, or is added to the l.h.s. in the case of outflow, or, finally, \mathbf{v}_j^{n+1} is imposed on inflow boundaries.

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

$$\begin{pmatrix} L_{\phi,\phi} & L_{\phi,*} \\ L_{\phi,*}^T & L_{*,*} \end{pmatrix} \begin{pmatrix} \delta p^{n+1} \\ \delta p^{*,n+1} \end{pmatrix} = \begin{pmatrix} D_\phi \\ D_* \end{pmatrix} (\hat{\mathbf{v}}^{n+1}) \quad (25)$$

where

- $(L_{\phi,\phi})_{N_n \times N_n} = \Delta t \int_{\Omega^e} \frac{1}{\rho} \nabla \phi_i^T \nabla \phi_j d\Omega$
- $(L_{\phi,*})_{N_n \times N_e} = \Delta t \int_{\Omega^e} \frac{1}{\rho} \nabla \phi_i^T \nabla \phi_j^* d\Omega$
- $(L_{*,*})_{N_e \times N_e} = \Delta t \int_{\Omega^e} \frac{1}{\rho} \nabla \phi_i^{*T} \nabla \phi_j^* d\Omega$
- $(D_\phi)_{N_n \times N_n} = \int_{\Omega^e} \nabla \phi_i^T \psi_j d\Omega$
- $(D_*)_{N_e \times N_n} = \int_{\Omega^e} \nabla \phi_i^{*T} \psi_j d\Omega$

At this point, it is possible to choose between two options. First, if the enriched space expressed by Eq. (21) is used, the new degrees of freedom can be included in the global system, guaranteeing continuity between elements. The second option is to follow the classical procedure of static condensation of the system through Gaussian elimination [21], where the following reduced system is obtained:

$$[L_{\phi,\phi} - L_{\phi,*}(L_{*,*})^{-1}L_{\phi,*}^T](\delta p^{n+1}) = [D_\phi - L_{\phi,*}(L_{*,*})^{-1}D_*](\hat{\mathbf{v}}^{n+1}) \quad (26)$$

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

In the case of an enriched space, see Eq. (18), when condensation is used, the assumption of neglecting the inter-elemental boundary terms from the integration by parts of the Poisson equation is not true. From our experience, and as reported by Coppola [19], when the Froude Number is high (that is, a low density ratio), the neglected term from the weakening of the divergence of the velocity introduces severe problems of numerical diffusion.

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

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

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

$$p^{*,n+1} = (L_{*,*})^{-1} [D_* \hat{\mathbf{v}}^{n+1} - L_{\phi,*}^T p^{n+1} + L_{\phi,*}^T p^n + L_{*,*} p^{*,n}] \quad (27)$$

Finally, the equation system presented in Eq. (28) must be solved.

$$\int_{\Omega} \psi \rho \mathbf{v}^{n+1} d\Omega = \int_{\Omega} \psi \rho \hat{\mathbf{v}}^{n+1} d\Omega - \Delta t \left[\int_{\Omega} \psi \nabla \delta p^{n+1} d\Omega + \int_{\Omega} \psi \nabla \delta p^{*,n+1} d\Omega \right] \quad (28)$$

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

Although this pressure reinitialization leads to a first order temporal approximation, successive iterations on a pressure-correction loop improve the incompressibility of the solution. Idelsohn [14] ensures that the stabilization effect of the first order fractional step is lost when higher order schemes are used. The same conclusion was obtained here, where instead of using a stabilization technique, a limited number of iterations (two or three) were used in order to obtain a converged pressure field without the presence of any pressure oscillations. More iterations would tend to make the scheme unstable.

3.4. PFEM-2 algorithm for two immiscible and incompressible fluids

Considering the details presented in the above section for the treatment of free-surface, the main changes regarding the general algorithm described in this section are:

- Implicit solution of viscosity terms: $\theta = 1$
- Pressure restarting: $p^n = 0$

Since the initial pressure p^n and the factor $(1 - \theta)$ are both zero, the stage 1.a of the general formulation can be suppressed and the algorithm starts with the X-IVAS step where the acceleration is only due to body forces \mathbf{f} . Finally, [Algorithm 1](#) presents the current PFEM algorithm to solve incompressible fluids with two different densities.

Algorithm 1 Time-step PFEM-2 for two immiscible and incompressible fluids.

1. X-IVAS Stage:

$$\begin{cases} \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \sum_{i=1}^N \mathbf{v}^i(\mathbf{x}_p^{n+\frac{i}{N}}) \delta t \\ \hat{\mathbf{v}}_p^{n+1} = \mathbf{v}_p^n + \sum_{i=1}^N \mathbf{f}^i(\mathbf{x}_p^{n+\frac{i}{N}}) \delta t \\ \lambda_p^{n+1} = \lambda_p^n \end{cases} \quad (29)$$

2. Projection Stage:

$$\begin{cases} \hat{\mathbf{v}}_j^{n+1} = \frac{\sum_p \hat{\mathbf{v}}_p^{n+1} W(\mathbf{x}_j - \mathbf{x}_p)}{\sum_p W(\mathbf{x}_j - \mathbf{x}_p)} \\ \lambda_j^{n+1} = \frac{\sum_p \lambda_p^{n+1} W(\mathbf{x}_j - \mathbf{x}_p)}{\sum_p W(\mathbf{x}_j - \mathbf{x}_p)} \end{cases} \quad (30)$$

3. Implicit Viscosity Stage:

$$\int_{\Omega} \hat{\mathbf{v}}_j^{n+1} \psi_j d\Omega = \int_{\Omega} \hat{\mathbf{v}}_j^{n+1} \psi_j d\Omega + \theta \Delta t \int_{\Omega} \mu(\mathbf{x}) \nabla^2 \hat{\mathbf{v}}_j^{n+1} \psi_j d\Omega \quad (31)$$

4. Pressure-Correction Iterations:

set $\mathbf{v}_j^n = \hat{\mathbf{v}}_j^{n+1}$

for $k = 1$ to K

(a) Poisson Stage:

$$\int_{\Omega} \nabla \cdot \left[\frac{\Delta t}{\rho(\mathbf{x})} \nabla (\delta p^{(n+\frac{k}{K})}) \right] \phi_j d\Omega = \int_{\Omega} \nabla \cdot \mathbf{v}_j^{(n+\frac{k-1}{K})} \phi_j d\Omega \quad (32)$$

(b) Correction Stage:

$$\int_{\Omega} \psi \rho(\mathbf{x}) \mathbf{v}_j^{(n+\frac{k}{K})} d\Omega = \int_{\Omega} \psi \rho(\mathbf{x}) \mathbf{v}_j^{(n+\frac{k-1}{K})} d\Omega - \Delta t \left[\int_{\Omega} \psi \nabla (\delta p)_j^{(n+\frac{k}{K})} d\Omega + \int_{\Omega} \psi \nabla (\delta p^*)^{(n+\frac{k}{K})} d\Omega \right] \quad (33)$$

end for

5. Particle Correction Stage:

$$\rho_p \mathbf{v}_p^{n+1} = \rho_p \hat{\mathbf{v}}_p^{n+1} + \sum_j \delta \mathbf{v}_j^{n+1} \psi_j(\mathbf{x}_p^{n+1}) \quad (34)$$

4. Results

In this section, a wide range of free-surface problems with different ratio between densities and viscosities of the fluids involved are solved using the PFEM-2 method and the results compared with reference ones, which include numerical results, experimental analysis and/or analytical solutions.

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

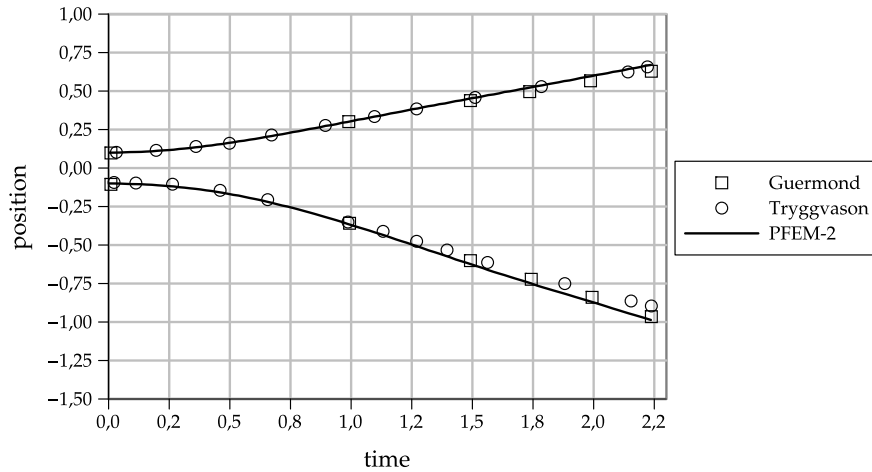


Fig. 3. Position of rising and falling bubbles versus time. Case with $Re = 1000$.

The main aim of this section is to show the capability of the method to work with large Courant numbers without stability loss and with negligible resign accuracy, reasonable large time-steps are then selected for each test. Previous PFEM-2 works (see [13,15]), showed that the computational cost of each time-step is almost equal to traditional Eulerian solvers, and the increase of time-step decreases the duration of the global computation without loss of accuracy. Herein, the efficient distributed-memory implementation presented in [15] is extended to the free-surface treatment and used to simulate each of next cases presented.

4.1. Rayleigh–Taylor instability

This problem is based on the evolution of two layers of fluids initially at rest in a gravity field. The density of the upper most fluid is larger than the one placed underneath. Due to a little disturbance in the contact surface the more dense fluid moves down and the less dense fluid does the opposite. During the evolution of the problem, a mixture is created, which is lately segregated. The final state reaches a stable equilibrium with the more dense fluid at the bottom layer and the less dense fluid at the top layer. The growth and evolution of the instability has been investigated among others by Tryggvason [22] for inviscid incompressible flows, and by Guermond and Quartapelle [23] for viscous flows.

The starting point is the problem documented by Guermond and Quartapelle [23]. The computational domain is $[-d/2, d/2] \times [-2d, 2d]$ and the initial position of the perturbed interface is $\eta(x) = 0.1d \cos(2\pi x/d)$. The density ratio is 3, which corresponds to an Atwood number of 0.5 according to Tryggvason's definition $At = (\rho_{max} - \rho_{min})/(\rho_{max} + \rho_{min})$. Other physical parameters are selected to obtain a Reynolds number $Re = \rho_{min} d^{3/2} g^{1/2} / \mu = 1000$. The computational domain is discretized into 80 000 structured triangles ($\Delta x = 0.01$) setting slip boundary conditions on each wall. The time step selected is $\Delta t = 0.01$ [s], which allows for $CFL_{max} \approx 8$. Among five and eight particles per element are used and two pressure iterations are required.

To compare solutions with reference results, the time is made dimensionless by using $\tilde{t} = t\sqrt{gAt}$. Results on the vertical position of the tip of the falling and rising fluid (spike and bubble, respectively) are shown in Fig. 3. It can be observed that current solution is in good agreement with the reference results.

On the other hand, the evolution of the instability is shown in Fig. 4 at dimensionless times $\tilde{t} = 0, 1, 1.5, 2$. Around $\tilde{t} = 1.5$ the heavy fluid begins to roll up into two counter-rotating vortices. Later, around $\tilde{t} = 2$, these two vortices become unstable and a pair of secondary vortices appear at the tails of the roll-ups. These shapes of the fluid interface obtained with PFEM-2 are similar to the ones shown as reference results, see [23].

4.1.1. Extending the time step

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

In the case of OpenFOAM®, the solver *interFoam* is chosen, which implements a Volume of Fluid (VoF) algorithm for multiphase flow [24,25]. It includes the multi-dimensional limiter for explicit solution (MULES) as a method of guaranteeing boundedness of scalar fields, in particular phase/mass-fractions (more information about MULES can be found in [26]). Since OpenFOAM® version 2.3, a new semi-implicit variant of MULES has been introduced which combines operator splitting with application of the MULES limiter to an explicit correction rather than to the complete flux. This approach would maintain boundedness and stability at an arbitrarily large Courant number. In the next simulation, the following recommended

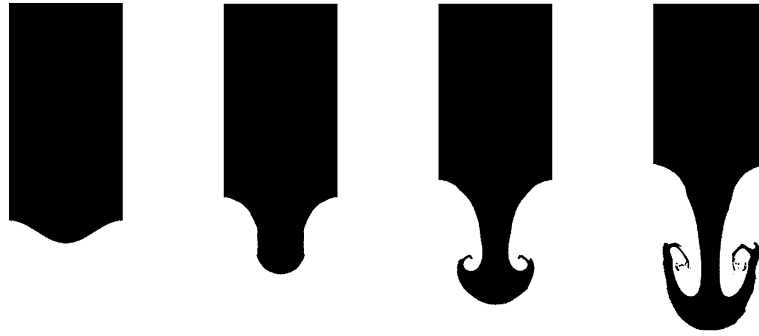










Fig. 4. Rayleigh–Taylor instability evolution. Case with $Re = 1000$. From left to right $\tilde{t} = 0.0, 1.0, 1.5, 2.0$.

Table 1

Rayleigh–Taylor instability captures at $\tilde{t} = 2.25$ for different CFL_{rt} numbers. OpenFOAM® simulation implements VoF+MULES simulation (interFoam solver).

Solver	$CFL_{rt} = 0.32$	$CFL_{rt} = 0.79$	$CFL_{rt} = 3.16$	$CFL_{rt} = 7.91$
PFEM-2				
OpenFOAM®				

simulation schemes have been used: CrankNicolson (second order, implicit) time integration, Gauss linear (second order, Gaussian integration with linear interpolation) discretization for the gradient, divergence and Laplacian operators (corrected with two nNonOrthogonalCorrectors due to the triangular mesh, for the later). Relevant VoF settings are: nAlphaSubCycles is set in order to keep the maximum local CFL of the sub-cycling around 0.5, cAlpha = 0.25 to give more stability through relaxing on some level the strong sharpness imposition, and MULESCorr is enabled to calculate the limiter in a semi-implicit way.

Table 1 presents the comparison of the solutions with PFEM and OpenFOAM® at a particular time ($\hat{t} = 2.25$) using several fixed time-steps, i.e. global CFL numbers. For normalization purposes of this particular case, that non-dimensional number is defined as $CFL_{rt} = U \Delta t / \Delta x$, being $U = \sqrt{gd}$ a reference velocity and d the fluid column of one of the phases. From captures, it can be shown that PFEM keeps approximately the same solution with each CFL_{rt} , but interFoam cannot solve with any accuracy using $CFL_{rt} > 0.5$ because the evolution of the mushroom-like interface differs from the reference results and this divergence is increased with larger time-steps.

Another relevant feature to take into account when comparing both algorithms is that similar CPU times are required to solve a time-step. Table 2 summarizes the CPU Times required to complete 1 [s] of real time in the current case. Results show that, using the same time-step, both solvers have similar performance, being OpenFOAM® faster. However, due to the capability of time-step enlargement with PFEM, shorter CPU times are achieved with similar accuracy.

4.1.2. Three-dimensional simulation

In this section, the extension of the two dimensional problem to three dimensions is presented. The third dimension is generated as a surface of revolution from the previous 2D geometry, conforming a cylindrical volume in 3D of radius

Table 2

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

Solver	Δt	CPU time
OpenFOAM®	0.001	1121 [s]
PFEM	0.0025	1011 [s]
PFEM	0.01	288 [s]
PFEM	0.025	123 [s]

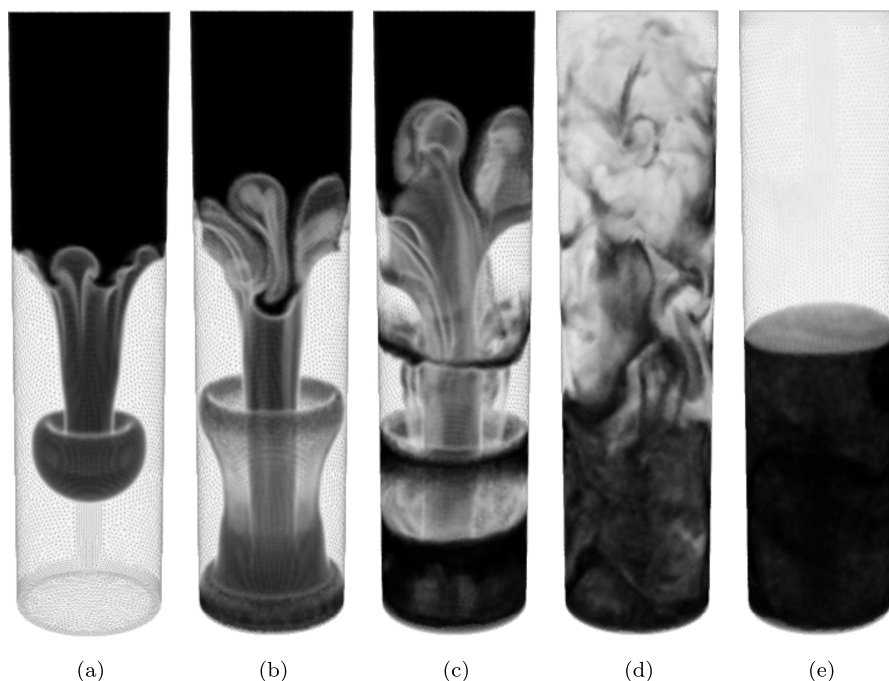


Fig. 5. Snapshots of the heavy-phase in the Rayleigh–Taylor instability solved in three-dimensions with PFEM-2. From left to right $\tilde{t} = 2$ [s], 3.3 [s], 4.4 [s], 8.8 [s], and 27.5 [s].

$R = 0.5$. This allows the same problem configuration to be kept, this is, a slip boundary condition on the wall, $At = 0.5$, and an initial perturbation of the surface $\eta(r) = 0.1d \cos(2\pi r/d)$, with $0 < r < R$.

The computational domain is discretized with a mesh size of $\Delta x = 0.03$ [m] conforming a non-structured mesh with around 1.2 million tetrahedral elements. An average of eight million particles are used during the simulation that move across the light-phase and heavy-phase domains. Simulation was carried out with a $\Delta t = 0.025$ which peaks to $CFL_{max} = 15$. Fig. 5 shows the evolution of the heavy-phase. It must be noticed that the simulation is extended until reaching the stable condition with the heavy-phase at the bottom and at rest, which is approximately 30 [s] of simulation time. To complete the entire simulation, the implementation requires around three wall-clock hours running on an AMD Opteron 6376 @ 2.3 GHz with a 64 Gb RAM using 16 processors.

The spirit of this section is to show the stability of the 3D simulation with a realistic progress. However, a way to prove the validity of the solution is that, during the simulation, the initial mass quantity is preserved. The conservation is theoretically expected since the analyzed case does not have a mass inlet and/or outlet. A more detailed analysis about the accuracy of 3D simulations will be presented in Section 4.4.1.

4.2. Non-linear sloshing in a rectangular container

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

For the studied cases in this section, the sloshing phenomenon is produced by a horizontal harmonic excitation $x = a_h \sin(\omega_h t)$, where a_h is the excitation amplitude and ω_h is the excitation frequency of the rectangular tank where the

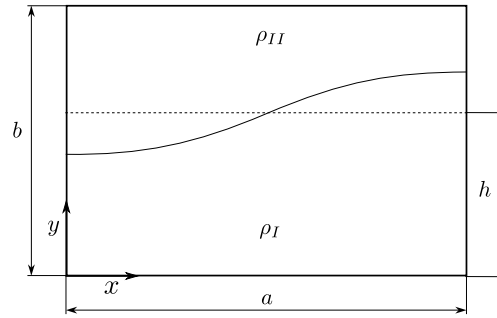


Fig. 6. Configuration of the non-linear sloshing in a rectangular container case. Initial condition is represented by a dashed line. The continuous line represents the position of the free-surface at a certain time.

two fluid phases are contained. The tank is divided in two parts, the bottom part where there is water with a density of $\rho_I = 1000$ [kg/m³] and the top part which contains a fluid with different densities $\rho_{II} = 1.3, 50, 200, 800$ [kg/m³], depending on the studied case. The dimensions of the tank are a (width) by b (height) and the initial free surface is at height h from the bottom of the tank, see Fig. 6. The free surface starts the simulation as a horizontal line and is subsequently deformed by the tank excitation and the flow dynamics.

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

Reference results for this case are taken from [27] which uses the codes STARCCM+ and OpenFOAM® to obtain numerical solutions and reports the free surface displacement on the left wall of the container. Those simulations use the same grid as presented above, but, in order to avoid numerical instabilities, the CFL number was limited to $CFL_{max} = 0.5$ which implies $\Delta t \approx 0.001$. In PFEM-2 such restrictions do not exist, Δt is therefore fixed to 0.01, reaching a $CFL_{max} \approx 5$.

Regarding to the numerical parameters α and β presented in Section 3.1, the simulations were done using a value of $\alpha = 10$ and $\beta = 0.1$. In this way, two cases where heavy particles on lighter media are moved with parabolic motion ($\sigma = 0.0013$ and $\sigma = 0.05$), and two cases where the trajectory prediction of those particles is done following streamlines ($\sigma = 0.02$ and $\sigma = 0.8$), are analyzed.

Fig. 7 presents the free surface displacement reported on the left wall of the container for different values of σ . For each one of them, PFEM-2 simulations show a good agreement with reference solutions. It is worth mentioning that the time step used is around ten times bigger than the one used in [27].

4.2.1. Enrichment and density jump issues

The PFEM-2 results presented in the previous section, have been obtained using the continuous enrichment strategy, which allows for the same formulation to be used, independently of the Froude number. However, as was mentioned before, without using a continuous formulation between elements for the enriched shape functions, the assumption about the inter-elemental boundary terms of the Poisson equation formulation should be revisited. In this subsection, problems that appear when the formulation combines non-continuous enriched shape functions and small density jump ($\rho_I / (\rho_I - \rho_{II}) \sim 1$) are presented. The latter is somewhat quantified by the Froude number, which also includes the ratio between inertial and gravitational forces, this is $Fr = \frac{U^2}{gL} \frac{\rho_I}{\rho_I - \rho_{II}}$. This phenomenon is mentioned, but not investigated, by Coppola-Owen and Codina [19].

Fig. 8 shows a comparison between the level height calculated by PFEM-2 using continuous enrichment, discontinuous enrichment and no-enrichment for two extreme density jump cases. It must be noted that next results were obtained with the enrichment shape functions presented in (18), however, using (21), and condensing the elemental matrices, similar conclusions were reached. The reference solution used in Fig. 8 was obtained using the formulation presented in (21) without condensing, which is equivalent to continuous enrichment. As was mentioned, that reference solution was previously validated in Section 4.2.

When both fluids have similar densities $\sigma \sim 1$, larger Fr values are obtained. In Fig. 8(b), a comparison with $\sigma = 0.8$ between different PFEM-2 formulations is presented. When no enrichment strategy is used, the solution presents a noisy behavior where, according to the level heights, a typical mass-loss appears, deteriorating the overall solution. When discontinuous enrichment is added (this is, condensing the equation system (25)) the solution is smoother but still excessively dissipative compared to the continuous enrichment formulation used as reference. Meaning that, when the discontinuous enrichment formulation is used, the assumption of avoiding the inter-elemental boundary terms in the right hand side of (23) is not correct and, consequently, a diffusive behavior appears. This conclusion is confirmed when discontinuous enrichment is used (but without weakening the divergence of the velocity) and a solution very similar to the continuous enrichment one is obtained (not presented in the figure due to being almost identical to the reference). By not integrating

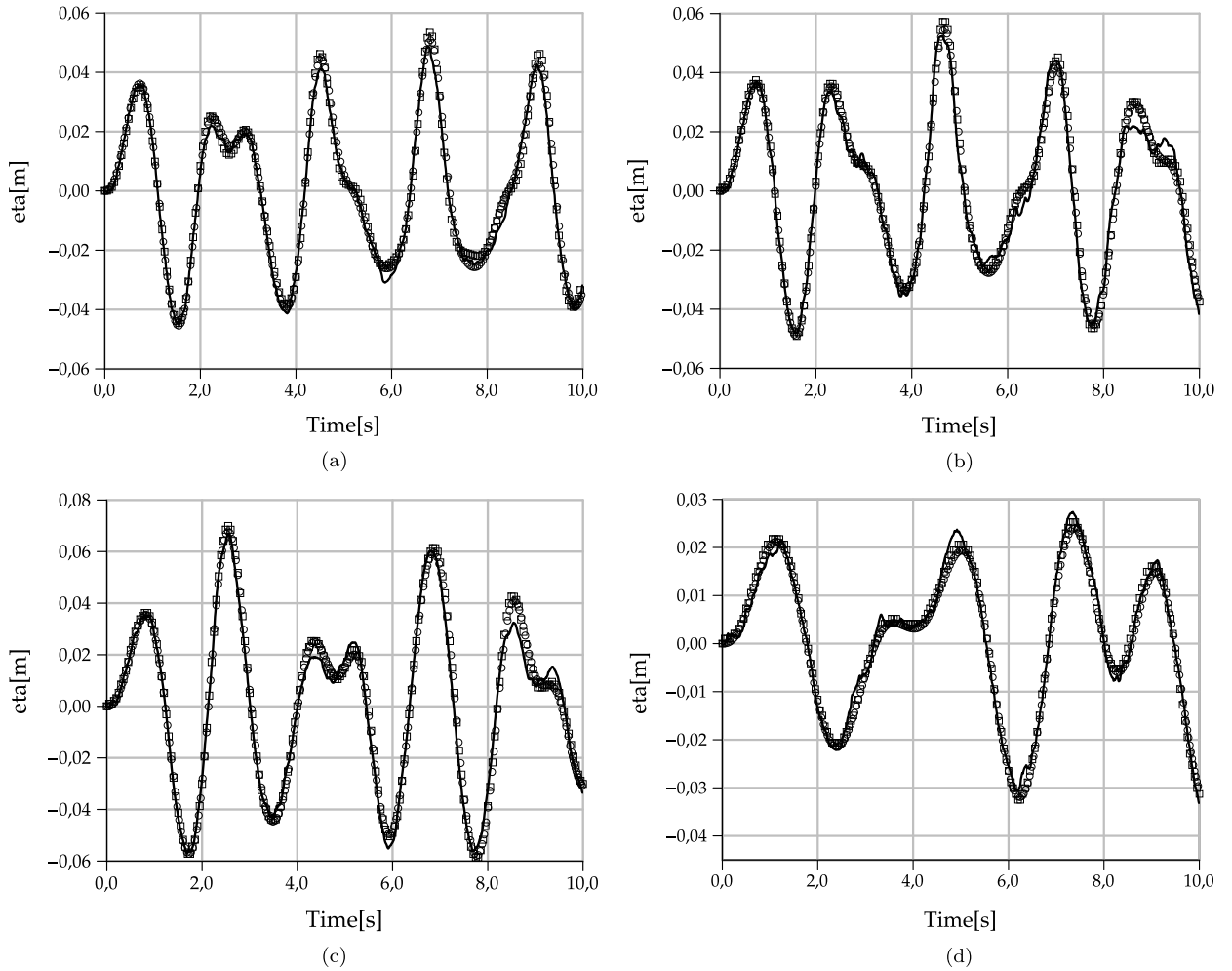


Fig. 7. Level height on the left wall for a two phase flow for different density jumps. (a) $\sigma = 0.0013$, (b) $\sigma = 0.05$, (c) $\sigma = 0.2$ and (d) $\sigma = 0.8$. References: \circ STARCCM, \square OpenFOAM and filled line PFEM-2.

by parts, the divergence term requires imposing a pressure gradient on the boundaries: if small density jumps are used, the previous gradient value ∇p^n could be imposed on the boundaries, although this approximation is not valid when there is a gravitational field, large density jumps and the fluid is not at rest.

For the other limit $\sigma \ll 1$, the case when $\sigma = 0.0013$, is presented in Fig. 8(a). The graphic shows that both the continuous and the discontinuous enrichment solutions present a very similar behavior. As before, when no enrichment is used, results become noisy, showing that a spurious velocity field appears close to the free surface when no improvements are introduced in this region. Discontinuous enrichment with a strong form of divergence of the velocity cannot be used in this case due to the wrong prediction of the pressure gradients on the boundaries which turns the simulation unstable.

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

4.3. Viscous standing waves

Computing the dissipation due to wave-breaking remains a challenging problem in the computational fluid mechanics context. In order to analyze the PFEM-2 solution when two-phase viscous incompressible flows are simulated, the evolution of a viscous standing wave has been chosen. An approximate analytical solution is available for small amplitude perturbations in the scientific literature [28] and it is of practical interest since it is related to the propagation of gravity waves.

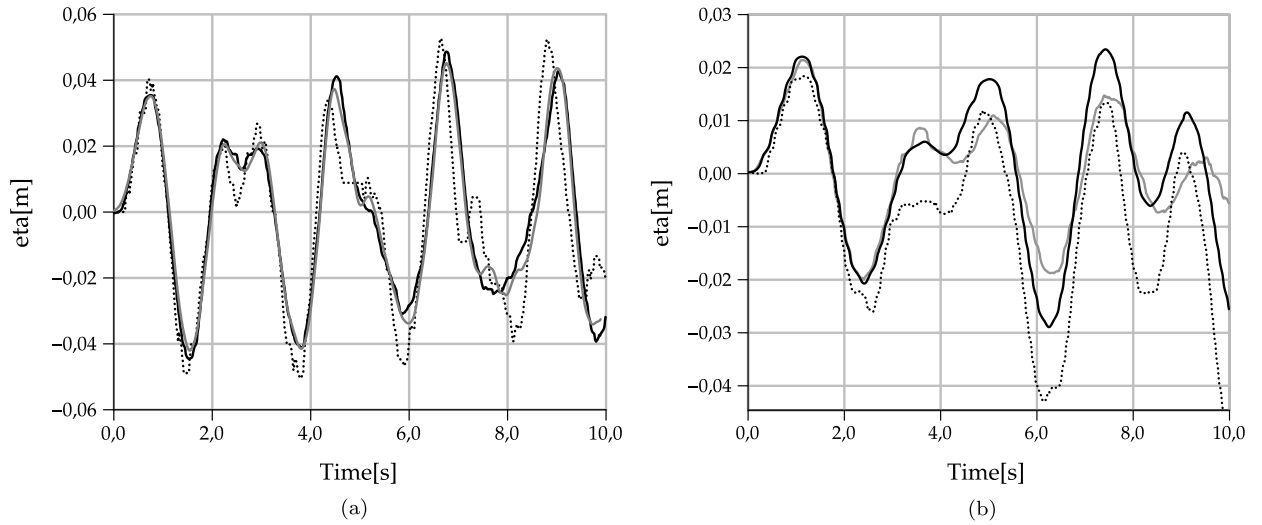


Fig. 8. Level height on the left wall of a two phase flow for different density jumps. (a) $\sigma = 0.0013$ and (b) $\sigma = 0.8$. References: filled black line continuous enrichment, filled gray line discontinuous enrichment, and dotted line without enrichment.

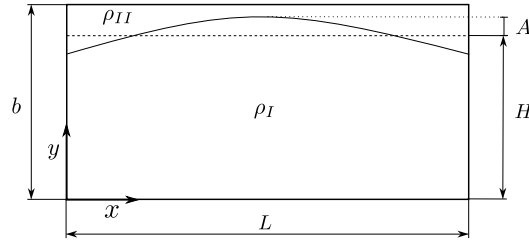


Fig. 9. Configuration scheme of the standing wave case. Initial condition is represented by dotted lines. The continuous line presents an intermediate state where the maximum amplitude A is reached.

The chosen standing wave configuration consists in a rectangular tank with length L and a water filling height of $H = L/2$. This setup has been extracted from [29], for a sketch of this configuration see Fig. 9. The wave length is $\lambda = L$, k is the corresponding wave number (i.e. $k = 2\pi/\lambda$) and A is the wave amplitude that defines the ratio $\epsilon = 2A/H$.

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

$$\varphi(x, y, t) = \varphi_0(x, y) \cos(\omega t) \quad (35)$$

$$\varphi_0(x, y) = -\epsilon \frac{Hg \cosh[k(y + H)]}{2\omega \cosh(kH)} \cos(kx) \quad (36)$$

where the circular frequency ω is given by the dispersion relation of gravity waves, that is, $\omega^2 = gk \tanh(kH)$ where g is the acceleration of gravity. At time $t = 0$, the free surface is horizontal while the initial fluid velocity is given by φ_0 .

It can be demonstrated that the approximate solution is well posed only for $Re \gg 1$ and for $Re^{-1} \ll k \ll Re^{2/3}$, where $Re = H\sqrt{gH}/\nu$ is the Reynolds number for this problem. From that solution, it is possible to obtain the formula that gives the approximate decay of the kinetic energy [28]:

$$\varepsilon_K(t) = \epsilon^2 g \frac{\lambda H^2}{32} e^{-4\nu k^2 t} [1 + \cos(2\omega t)] \quad (37)$$

The kinetic attenuation is governed by the parameter $\beta_l = 4\nu k^2$, which depends on the wave number and on the kinematic viscosity $\nu = \mu/\rho_l$. Lately, work [30] has demonstrated that generally, Eq. (37) overestimates the dissipations, especially when the Reynolds number is not very large. Then, an improved damping rate was proposed in [30] $\beta = 4\nu k^2 - 2\sqrt{2}k^{11/4}Re^{-3/2} + O(Re^{-2})$, which is used next for comparisons.

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

Several Reynolds number ($Re = 25, 50, 250, 2500$) have been chosen to compare with the approximate analytic dissipation. Problems were solved using a grid size Δx such that $H/\Delta x = 100$ and varying Δt in order to solve using a Fourier

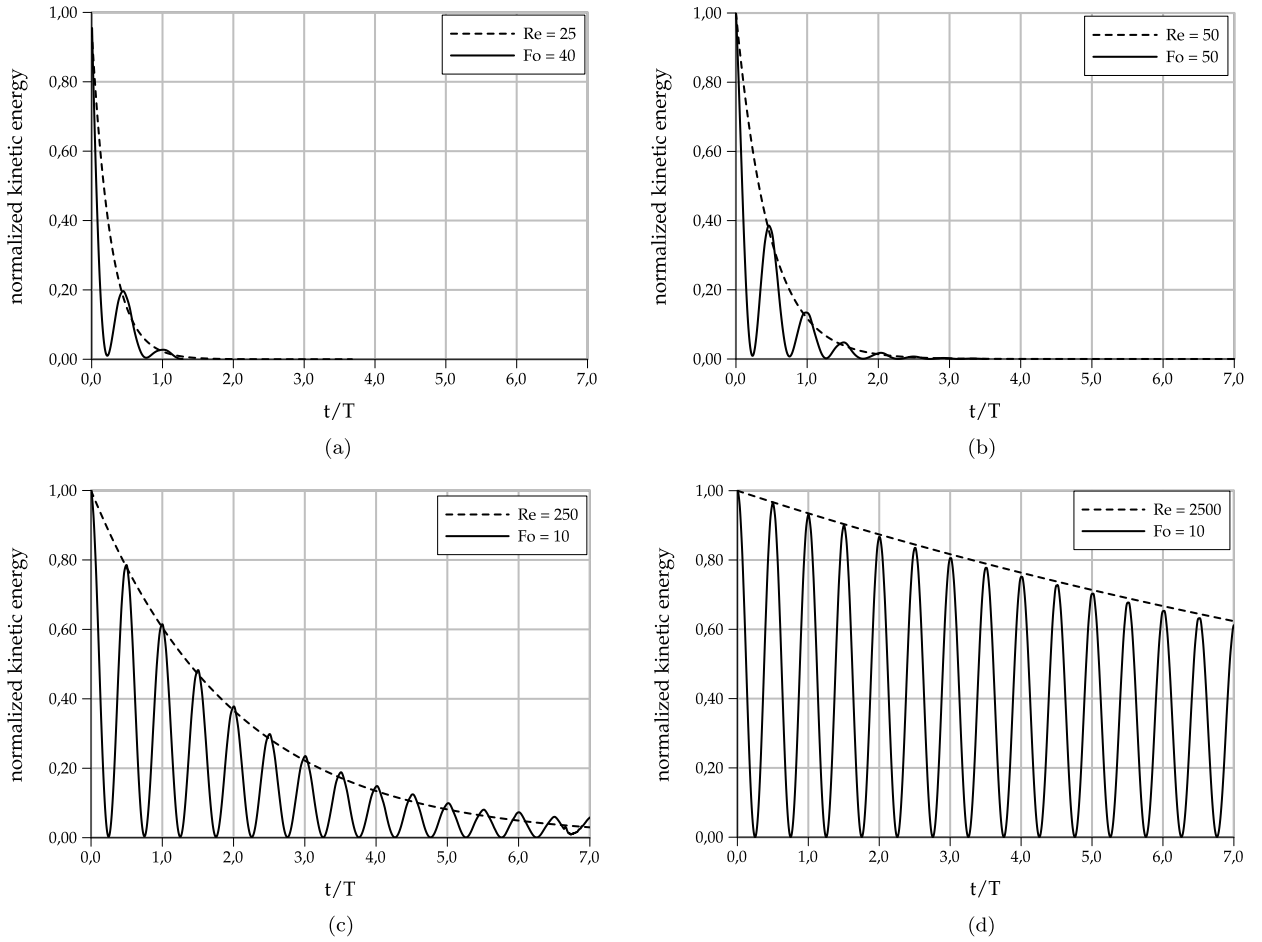


Fig. 10. Kinetic energy decay for standing wave problem with different Reynolds numbers. Dashed lines are approximate analytical solutions for total energy and continuous lines are the kinetic energy calculated with PFEM-2. Reynolds numbers analyzed: $Re = 25, 50, 250, 2500$ in (a), (b), (c), (d) respectively. Legend in each figure indicates the maximum Fourier number used in each numerical simulation.

number $Fo = \frac{\nu \Delta t}{\Delta x^2}$ with a local maximum of $Fo_{max} \approx 10$ –50. Fig. 10 shows the comparison between the expected energy dissipation (which includes the improved damping rate) and numerical results for the kinetic energy calculated with PFEM-2. Large Fourier numbers were used in order to reduce the computation times required to complete the simulations, showing that the accuracy of the method is maintained with large CFL numbers, even when diffusion is considered.

Regarding the computational effort, it is noticeable that the complete set of PFEM simulations only required a few hours on an Intel i7-2600k 3.4 GHz processor. This is a large difference with respect to the CPU times reported in other works, see [29], where a similar set of problems were solved with similar accuracy using SPH and a 30 CPUs cluster, each CPU with 8 Intel Xeon 2.33 GHz cores, running continuously for 30 days.

4.4. Dam-break problem

The objective of this section is to compare experimental measurements of a dam-break flow over a dry horizontal bed with the numerical approximation carried out with the PFEM-2 algorithm. The extensive set of experimental data is extracted from [31], where the dynamics of the dam break wave impacting a vertical wall downstream, with emphasis on the pressure loads and surface evolution after the dam burst, are presented.

A computational configuration of the tank used in experimental cases is presented in Fig. 11, where the locations of water level measuring points and pressure sensors are shown. In this report, only the case with $H = 300$ [mm] is analyzed. A two-phase non-viscous flow simulation is carried out, with $\rho_{water} = 1000$ [kg/m³], $\rho_{air} = 1$ [kg/m³] and gravity force $\mathbf{g} = -10\hat{j}$ [m/s²]. The 2D computational grid used has 322×120 nodes, conforming a mesh with around 80 000 triangles. Boundary conditions are slip on all walls, and Δt is fixed to 0.1, which allows for $CFL_{max} \approx 20$ when the free surface impacts the downstream wall.

Fig. 12 shows the comparison between experimental and numerical results for each water-level measurement. A good agreement can be observed, moreover when taking into account the capture of the back wave and splashing start events.

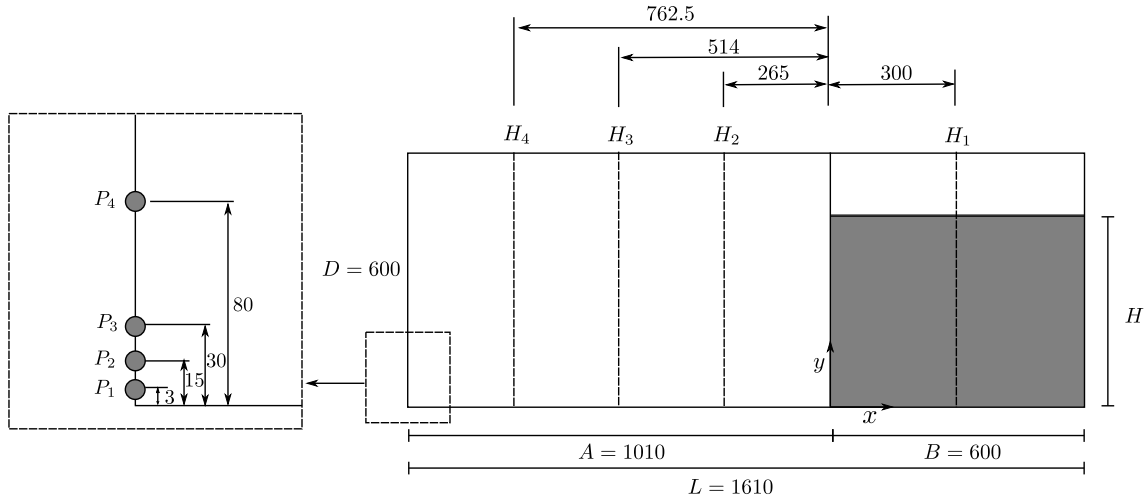


Fig. 11. Configuration scheme of the dam-break case. H_1 , H_2 , H_3 , H_4 present the locations of water level measuring positions. Also, P_1 , P_2 , P_3 , P_4 show the locations of pressure sensors at the impact wall downstream from the dam. The gray zone represents the initial water condition. Dimensions are in millimeters.

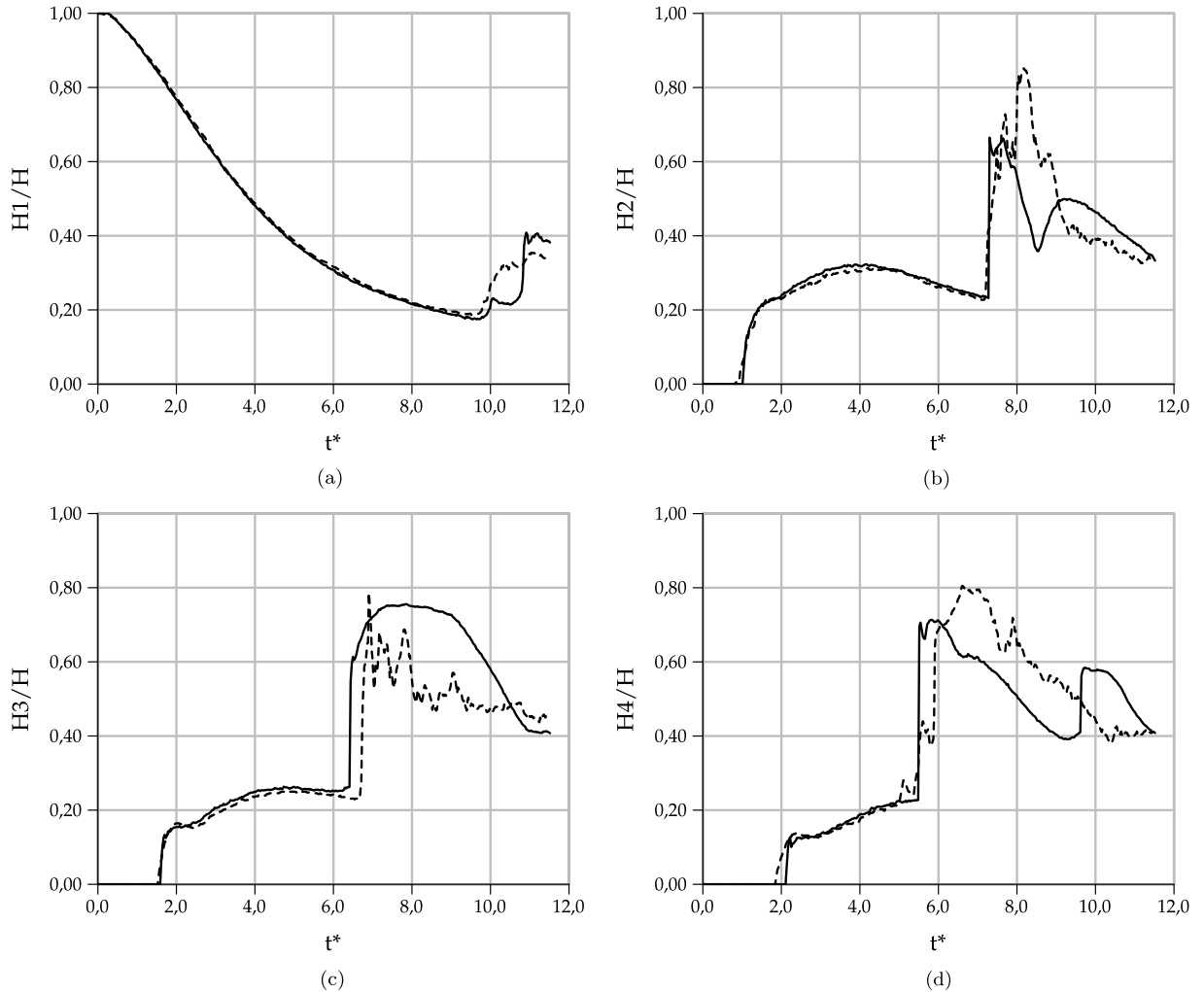


Fig. 12. Water levels at locations H_1 , H_2 , H_3 and H_4 for tests with initial filling height $H = 300$ [mm] compared to data from literature experimental results [31] (dashed lines) and numerical results with PFEM-2 (continuous lines). Time normalization is $t^* = t(g/h)^{1/2}$.

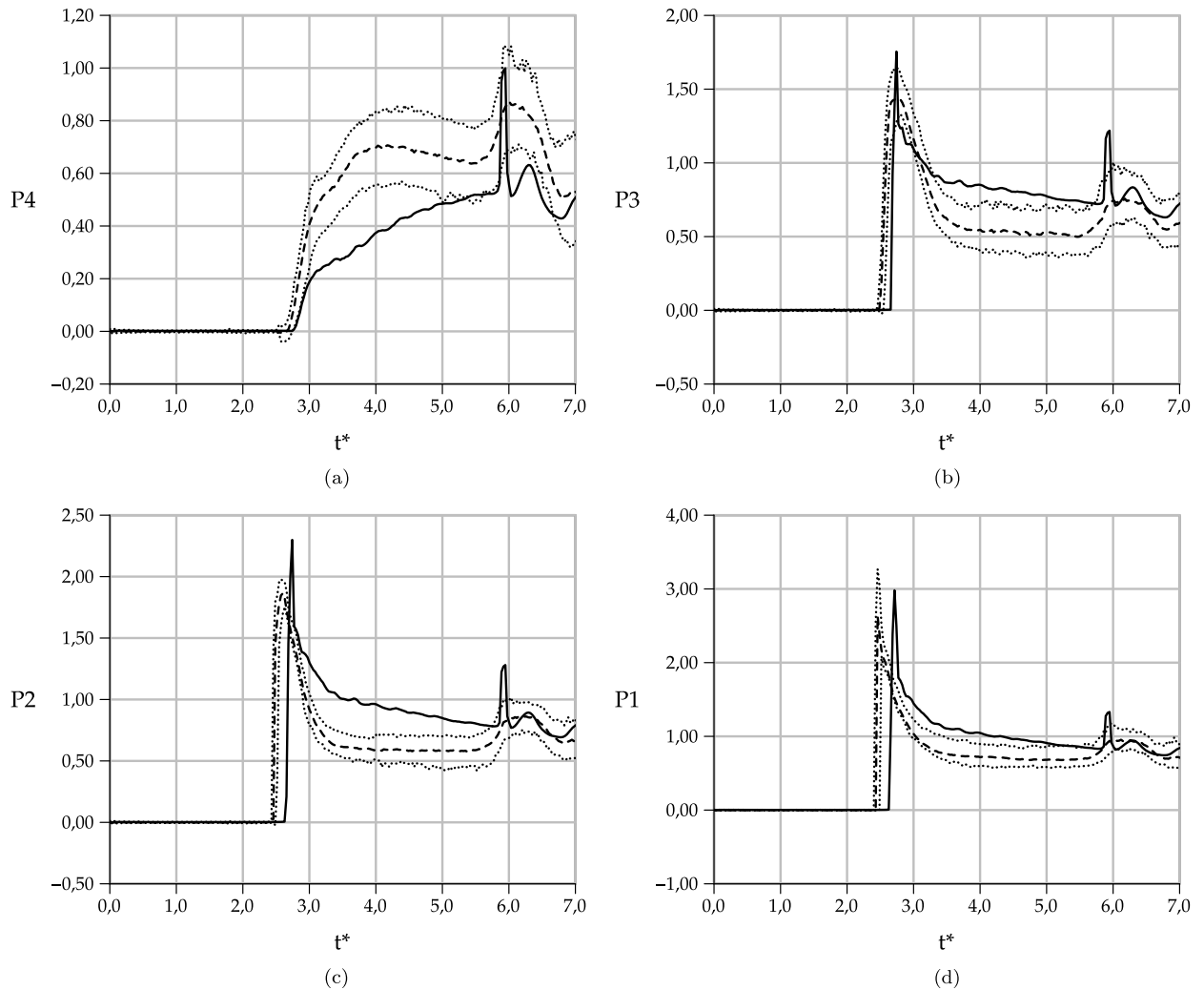


Fig. 13. Pressure time histories comparison between experimental results [31] (discontinuous lines) and numerical results with PFEM-2 (continuous lines). Values at locations P_1 , P_2 , P_3 , P_4 are presented in (a), (b), (c), (d) respectively. Experimental results shows the median (dashed lines) and percentiles 2.5 and 97.5 (dotted lines).

The impact pressure was measured with four sensors on the vertical wall at the end of the downstream flume, as described in Fig. 11. The statistical analysis of the pressure peaks, rise times and the occurrence time, i.e. the time between the opening of the dam gate and the occurrence of the impact, are presented in Fig. 13. The shown pressure P is non-dimensionalized with regards to the hydrostatic pressure at the bottom of the reservoir.

In the reference work, the analysis is focused on peak events. It can be noticed that the highest peak is recorded by sensor number 1 which is the sensor receiving the full impact, whilst the pressure of the other sensors is given by the run up of the flow. It can also be observed that sensor number 4, i.e. the sensor located at the highest position, does not show a pure impact event, see Fig. 13(d), and the maximum for this sensor is actually obtained later in time, when the water falls back after running along the wall. Numerical solution behavior follows the mentioned conclusions, although the pressure values are not between the statistical limits of experimental data. Also, a discrepancy can be observed with the peaks arrival times for sensors 1 to 3. This difference can be assigned to the numerical simplification which does not model the gate movement. However, the pressure magnitude of the peaks is well predicted giving confidence to PFEM-2 calculations.

Finally, Fig. 14 presents snapshots for the evolution of the simulated free-surface. Initial condition is shown in Fig. 14(a). Pressure peaks are related with the impact event observed in Fig. 14(c), which generates the back wave propagation that is displayed in the remaining figures.

4.4.1. Three-dimensional simulation

Despite being a problem that can accurately be simulated in 2D, the same example was run in 3D to test the ability of the PFEM-2 solver to deal with larger geometries and 3D problems. The 3D mesh, which adds a third dimension of a

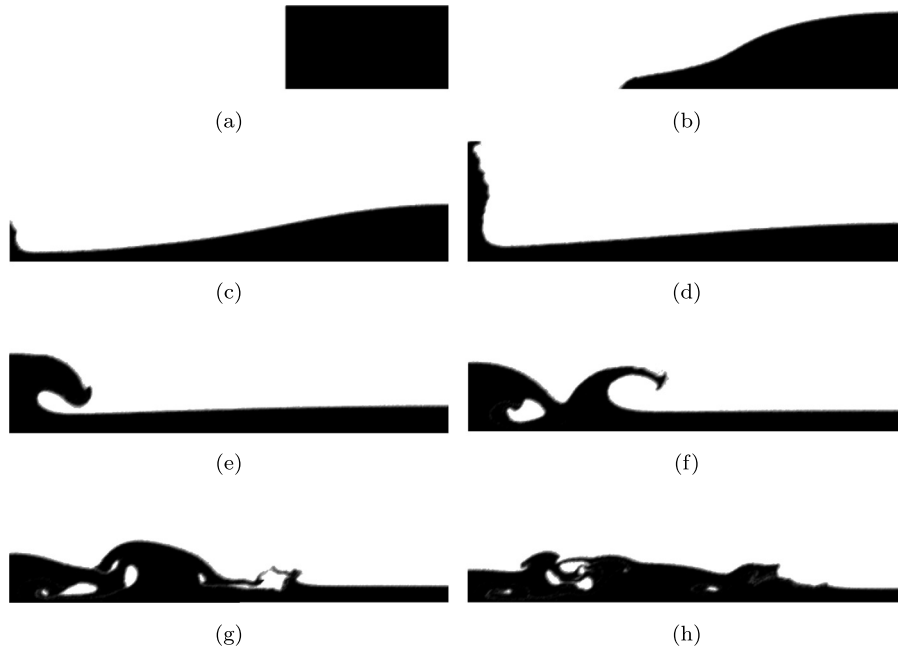


Fig. 14. Snapshots of the dam-break at times $t = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75$ [s], in (a) to (h) respectively.

thickness of 0.15 [m] with slip walls, has six million elements. These have an average $h = 0.07$, demanding more than 25 million particles for the fixed mesh approximation that fill the air and water domains. The same physical and numerical parameters of the 2D simulation were used. Sensors are placed in the same position on the left wall and in the middle position of the third dimension, see Fig. 11.

In this example, the time step used, the element sizes and the velocity of the fluid lead to a simulation with some time-steps having a Courant number larger than 12, mainly when waves impact with walls. This shows once more, the capability of the method to manage 3D geometries with large time-steps. In Fig. 15, the pressure history for each sensor is presented, showing a similar appearance to the two dimensional simulation.

5. Conclusions

The last generation of the Particle Finite Element Method (PFEM-2) is a contemporary strategy which uses a spatial discretization based on a background mesh and a cloud of particles. The dynamics equations are solved in a Lagrangian frame, where the implicit non-linearities of the equation are solved using the X-IVAS strategy. That explicit temporal integration for convective terms allows for the use of large time-steps, thus providing a very efficient way when computing times are concerned.

In the current work, a formulation to solve free-surface flows with pressure gradient discontinuities, presented in [14], was generalized and exhaustively tested. That algorithm, which is based on a continuous enriched space for pressure, has shown good accuracy when solving a wide range of multiphase problems and keeping the advantage of the possibility to use large time-steps. Several cases with a large variety of Froude numbers, density ratios and dominant dissipative cases have also been analyzed. These results were compared with other reference softwares, semi-analytical expressions and also experimental data. In each of them, PFEM-2 has proven to be accurately competitive and computationally efficient. Regarding CPU times, they can be decreased without accuracy loss if the original condensing strategy for enriched pressure degrees of freedom is used. Although that approach loses the generality of the formulation for any range of application, the computational cost is reduced, which, to our knowledge, currently makes PFEM-2 the faster algorithm for solving two phase flows.

Acknowledgements

Authors thank to Pedro Galán del Sastre (UPM-Spain), Eng. Pablo Becker (CIMNE-Spain), Dr. Norberto Nigro (CIMEC-Argentina) and Prof. Sergio Idelsohn (CIMNE-Spain and CIMEC-Argentina) for their valuable help and discussions about the method features and its implementation. Also to Dr. Santiago Márquez Damián (CIMEC-Argentina) for his contribution in OpenFOAM simulations.

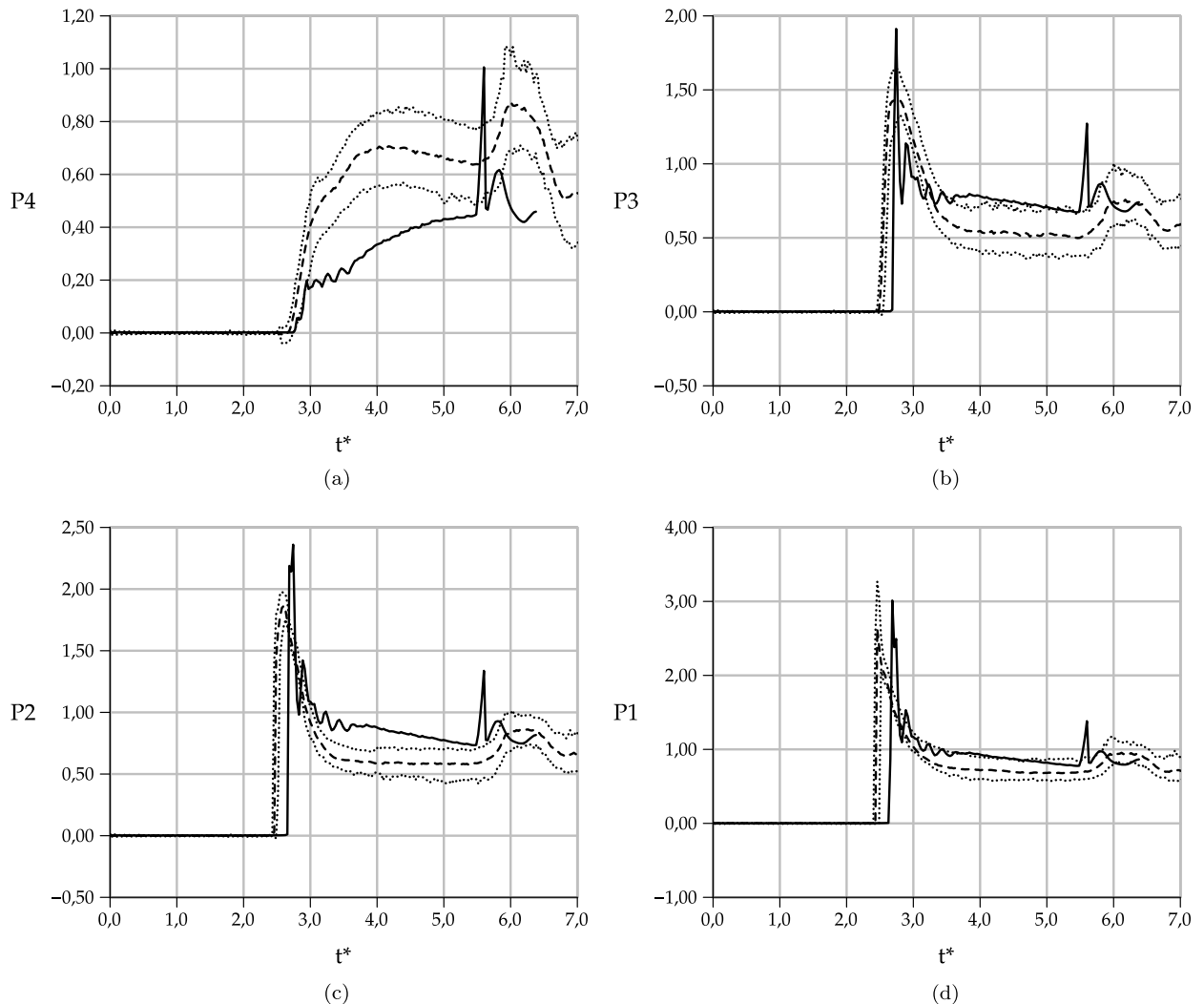


Fig. 15. A pressure time history comparison between experimental results [31] (discontinuous lines) and numerical results with PFEM-2 in three dimensions (continuous lines). Values at locations P_1 , P_2 , P_3 , P_4 are presented in (a), (b), (c), (d) respectively. Experimental results shows the median (dashed lines) and percentiles 2.5 and 97.5 (dotted lines).

J. Gimenez gratefully acknowledges the support of the Argentinian Agencia Nacional de Promoción Científica y Técnica (ANPCyT) through a doctoral grant in the FONCyT program. Also the authors would like to thank the program ERASMUS mundus action 2 ARCOIRIS project for their financial support through a six-month doctoral scholarship.

Other financial support was provided by CONICET, Universidad Nacional del Litoral (CAI+D Tipo II 65-333 (2009)), ANPCyT-FONCyT (grants PICT 1645 BID (2008)) and ERC Advanced Grant REALTIME project AdG-2009325.

The research leading to these results has also received funding from the Spanish Ministry for Science and Innovation under grant TRA2010-16988 “Optimización del transporte de Gas Licuado en buques LNG mediante estudios sobre interacción fluido-estructura”.

All the authors want to thank Mr. Hugo Gee for his valuable assistance during the preparation of this manuscript.

References

- [1] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics, theory and application to non-spherical stars, *Mon. Not. R. Astron. Soc.* 181 (1977) 375–389.
- [2] J. Monaghan, An introduction to SPH, *Comput. Phys. Commun.* 48 (1988) 89–96.
- [3] R.M. Nestor, M. Basa, M. Lastiwka, N.J. Quinlan, Extension of the finite volume particle method to viscous flow, *J. Comput. Phys.* 228 (5) (2009) 1733–1749, <http://dx.doi.org/10.1016/j.jcp.2008.11.003>.
- [4] S. Idelsohn, N. Calvo, E. Oñate, Polyhedrization of an arbitrary 3D point set, *Comput. Methods Appl. Mech. Eng.* (ISSN 0045-7825) 192 (22–24) (6 June 2003) 2649–2667, [http://dx.doi.org/10.1016/S0045-7825\(03\)00298-6](http://dx.doi.org/10.1016/S0045-7825(03)00298-6).
- [5] S. Idelsohn, E. Oñate, N. Calvo, F.D. Pin, The meshless finite element method, *Int. J. Numer. Methods Eng.* 58 (6) (2003) 893–912.

- [6] S. 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. Numer. Methods Eng.* 61 (2004) 964–989.
- [7] F. Harlow, A machine calculation method for hydrodynamic problems, Tech. rep., Los Alamos Scientific Laboratory report LAMS-1956, Nov. 1955.
- [8] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* (1958–1988) 8 (1965) 2182–2189, <http://dx.doi.org/10.1063/1.1761178>.
- [9] Z. Wieckowsky, The material point method in large strain engineering problems, *Comput. Methods Appl. Mech. Eng.* 193 (39) (2004) 4417–4438.
- [10] J. Donea, Arbitrary Lagrangian–Eulerian finite element methods, in: T. Belytschko, T. Hughes (Eds.), *Computational Methods for Transient Analysis*, 1983, pp. 474–516.
- [11] A. Allievi, R. Bermejo, Finite element modified method of characteristics for Navier–Stokes equations, *Int. J. Numer. Methods Fluids* 32 (2000) 439–464.
- [12] S. Idelsohn, N. Nigro, A. Limache, E. Oñate, Large time-step explicit integration method for solving problems with dominant convection, *Comput. Methods Appl. Mech. Eng.* 217–220 (2012) 168–185.
- [13] S. Idelsohn, N. Nigro, J. Gimenez, R. Rossi, J. Marti, A fast and accurate method to solve the incompressible Navier–Stokes equations, *Eng. Comput.* 30 (2) (2013) 197–222.
- [14] S. Idelsohn, J. Marti, P. Becker, E. Oñate, Analysis of multifluid flows with large time steps using the particle finite element method, *Int. J. Numer. Methods Fluids* 75 (9) (2014) 621–644, <http://dx.doi.org/10.1002/flid.3908>.
- [15] J. Gimenez, N. Nigro, S. Idelsohn, Evaluating the performance of the particle finite element method in parallel architectures, *Comp. Part. Mech.* 1 (2014) 103–116, <http://dx.doi.org/10.1007/s40571-014-0009-4>.
- [16] R.D. Nair, J.S. Scroggs, F.H. Semazzi, A forward-trajectory global semi-Lagrangian transport scheme, *J. Comput. Phys.* 190 (1) (2003) 275–294, [http://dx.doi.org/10.1016/S0021-9991\(03\)00274-2](http://dx.doi.org/10.1016/S0021-9991(03)00274-2).
- [17] V. Alexiades, G. Amiez, P. Gremaud, Super-time-stepping acceleration of explicit schemes for parabolic problems, *Commun. Numer. Methods Eng.* 12 (1996) 31–42.
- [18] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Adv. Comput. Math.* 389 (1995) 389–396.
- [19] A.H. Coppola-Owen, R. Codina, Improving Eulerian two-phase on finite element approximation with discontinuous gradient pressure shape functions, *Int. J. Numer. Methods Fluids* 49 (2005) 1287–1304.
- [20] S. Idelsohn, M. Mier-Torrecilla, N. Nigro, E. Oate, On the analysis of heterogeneous fluids with jumps in the viscosity using a discontinuous pressure field, *Comput. Mech.* 46 (1) (2010) 115–124, <http://dx.doi.org/10.1007/s00466-009-0448-6>.
- [21] C. Felippa, *Introduction to Finite Element Methods*, University of Colorado, 2004, Chapter 10.
- [22] G. Tryggvason, Numerical simulations of the Rayleigh–Taylor instability, *J. Comput. Phys.* 75 (1988) 253–282.
- [23] J. Guermont, L. Quartapelle, A projection fem for variable density incompressible flows, *J. Comput. Phys.* 165 (2000) 167–188.
- [24] E. Berberović, N.P. van Hinsberg, S. Jakirlić, I.V. Roisman, C. Tropea, Drop impact onto a liquid layer of finite thickness: dynamics of the cavity evolution, *Phys. Rev. E* 79 (2009) 036306, <http://dx.doi.org/10.1103/PhysRevE.79.036306>.
- [25] S. Márquez Damián, N.M. Nigro, An extended mixture model for the simultaneous treatment of small-scale and large-scale interfaces, *Int. J. Numer. Methods Fluids* (2014), <http://dx.doi.org/10.1002/flid.3906>.
- [26] S. Márquez Damián, An extended mixture model for the simultaneous treatment of short and long scale interfaces, Ph.D. thesis, Facultad de Ingeniería y Ciencias Hídricas (FICH), Instituto de Desarrollo Tecnológico para la Industria Química (INTEC), Universidad Nacional del Litoral, 2013.
- [27] J. Gómez-Gofi, C.A. Garrido-Mendoza, J.L. Cercós, L. González, Two phase analysis of sloshing in a rectangular container with volume of fluid (VoF) methods, *Ocean Eng.* 73 (2013) 208–212.
- [28] J. Lighthill, *Waves in Fluids*, Cambridge University Press, 2001.
- [29] A. Colagrossi, A. Souto-Iglesias, M. Antuono, S. Marrone, Smoothed-particle-hydrodynamics modeling of dissipation mechanisms in gravity waves, *Phys. Rev. E* 87 (2013) 023302.
- [30] M. Antuono, A. Colagrossi, The damping of viscous gravity waves, *Wave Motion* 50 (2013) 197–209.
- [31] L. Lobovský, E. Botia-Vera, F. Castellana, J. Mas-Soler, A. Souto-Iglesias, Experimental investigation of dynamic pressure loads during dam break, *J. Fluids Struct.* (ISSN 0889-9746) 48 (July 2014) 407–434, <http://dx.doi.org/10.1016/j.jfluidstruct.2014.03.009>.