Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXIII, págs. 387-405 (artículo completo)
Graciela Bertolino, Mariano Cantero, Mario Storti y Federico Teruel (Eds.)
San Carlos de Bariloche, 23-26 Setiembre 2014

# MULTIPHASE FLOWS SIMULATION WITH THE PARTICLE FINITE ELEMENT METHOD AND ITS COMPARISON WITH EULERIAN ALTERNATIVES

**Juan M. Gimenez[a], Santiago Márquez Damián[a,b], Horacio Aguerre[a], Norberto M. Nigro[a,b] and Sergio Idelsohn[a,c]**

[a]*Centro de Investigación de Métodos Computacionales (CIMEC) - UNL/CONICET, Predio Conicet-Santa Fe Colectora Ruta Nac 168 Paraje El Pozo, Santa Fe, Argentina,http://www.cimec.org.ar*

[b]*Facultad de Ingeniería y Ciencias Hídricas - Universidad Nacional del Litoral. Ciudad Universitaria. Paraje "El Pozo". Santa Fe. Argentina. http://www.fich.unl.edu.ar*

[c]*ICREA Research Professor at the International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain.http://www.cimne.edu.es/*

**Keywords:** particle, tracking, turbulence, CFD.

**Abstract.**

The latest version of the Particle-Finite Element Method (PFEM), which incorporates the novel explicit integration strategy named eXplicit Integration of Velocity and Acceleration following Streamlines (X-IVAS), has proven to be fast and accurate to solve homogeneous flows, mainly thanks to the possibility of using large time-steps. In this work the extension of this strategy to solve multiphase flows is presented, where the calculation of the interface evolution is of fundamental importance.

In Eulerian formulations, one of the most used strategies to determine the interface position is the advection of an indicator function. This approach is followed, by example, in the Volume of Fluid (VoF) technique, which can add limiters as a method of guaranteeing boundedness and/or sharpness of phase-fractions. On the other hand, Lagrangian frames use typically marker particles. In the case of PFEM, the same set of particles transported for flow calculation allows to carry a marker function to determinate the interface position without any extra cost. In order to compare the accuracy of PFEM interface evolution strategy with the Eulerian one implemented in the widely used OpenFOAM® suite, several classical tests are presented.

In addition to capture the sharpen interface evolution, PFEM algorithm includes the use of an enriched finite element space to avoid spurious solutions due to the discontinuity of pressure gradients, which also requires some changes in the streamline integration strategy. The description of those improvements are also presented in this work. Finally the PFEM algorithm is tested for a number of problems involving free surface flows with different ratio between the densities and viscosities of the fluids involved. The accuracy of the results are compared with reference ones, focusing in the capability of using large time-steps in contrast with Eulerian solvers, including those with implicit phase-fractions advection treatment, represented by the OpenFOAM® suite.

## 1 INTRODUCTION

The efficient solution of multi-phase flows is still an open challenge. Although the dynamics of single phase flows are well understood and can be solved accurately without loss efficiency, the computational modeling of two or more phases is a underdeveloped field with growing interest. In multi-phase flows the most important stage is the calculation of the interface evolution (Hieber and Koumoutsakos, 2005). Discontinuities of the fields or of its gradients often introduce numerical problems at the internal interfaces, therefore it is essential calculating accurately the interface position and this critical zone must remain sharp (Coppola-Owen and Codina, 2005).

In the Eulerian strategies to solve multiphase problems, a widely used tool is the Finite Element Method (FEM). However standard FEM is incapable of being accurate since no discontinuity is allowed in the shape functions field. An alternative to overcome this limitation is to use Enriched Finite Elements, which adds degrees of freedom to elements that are cut by the interface in order to capture the part of the solution that escapes from the standard shape function field. Coppola-Owen and Codina (Coppola-Owen and Codina, 2005) proposed a simple enrichment function that is capable of capturing accurately gradient discontinuities, while that to capture discontinuities in the value of the unknown Ausas (Ausas et al., 2012) proposed a set of three enrichment functions that are able to capture both a gradient discontinuity and a jump in the field. Another option in the Eulerian framework is the Finite Volume Method (FVM), which is more widely used than the FEM for fluid dynamics. Its discretization leads to a formulation conservative on the fluxes, unlike FEM.

Once a strategy to solve the fluid dynamics has been selected, an accurate and efficient computational methods to describe the evolution of interfaces must be chosen. There are two methodologies to do this work, namely: interface capturing and interface tracking methods. Purely Eulerian algorithms use capturing methods: in this approach the interface is determined by an implicit function that is advected through the computational domain. Popular methods of this type are the Level Set Method (LSM) (Osher and Sethian, 1988), which have become widely used when the interface undergoes extreme topological changes, e.g. merging or pinching off; and the Volume of Fluid (VoF) technique (Hirt and Nichols, 1981), which is widely implemented in the FVM. The LSM consist of using a distance function that is convected according to the fluid velocity. This function represents the distance from a point to the interface. By definition, the interface will be located where its value is zero. The main drawback is that the level set function field degrades when advancing in time and after some time steps a reinitialization of the level set must be done to guarantee accuracy on the interface capturing. On the other hand, VoF defines a function that is the fraction occupied by one of the phases in each cell of the domain. Once this phase-fraction has been advected along the entire domain, the interface position can be reconstructed. The FVM is very robust and is likely to be the most used one in commercial/widespread codes. As an example of application, OpenFOAM®(Weller et al., 1998) uses this strategy to solve multi-fluid problems. Completing with purely Eulerian approaches, coupled strategies as CLSVoF (Sussman and Puckett, 2000) are used to overcome the problems that appear when the interface develops structures whose length scales are smaller than those afforded by the Eulerian mesh.

Formulations clustered in the Lagrangian framework are a more natural choice for simulations where there are large deformations. Using particles that are advected carrying its own properties over the domain, they are able to almost avoid the numerical diffusion. Also, the Lagrangian perspective makes it possible to use a material derivative formulation

where the absence of the non-linear convective term transform the Navier-Stokes system into a transformed linear coupled problem. However, most of Lagrangian formulations have the uncomfortable drawback of requiring a particle position treatment: search algorithms, evaluation of the mesh distortions or the re-meshing processes, which are always computationally expensive, and it would be interesting to explore the possibility of avoiding those steps.

The original idea with Lagrangian frameworks, proposed by Monaghan (Gingold and Monaghan, 1977) and later works applied to fluid mechanics (Monaghan, 1988), was a explicit meshless method named Smoothed Particle Hidrodynamics (SPH) which was strongly limited by the Courant number. Evolutions start to include mesh calculations to solve implicit systems, being one of the most known the Particle Finite Element Method (PFEM) (Idelsohn et al., 2004). The PFEM consist of using a set of particles that define the nodes of a finite element mesh. Since fluids have no deformation limit, remeshing must be done at each time step.

Alternatives, as PFEM, that combines both Eulerian and Lagrangian tools, have shown to be a choice respect to pure methods. In the work of Enright et al. (Enright et al., 2002) a pure Eulerian solver for the fluid is used, but Lagrangian marker particles are used to improve the LSM, then the interface tracking. On the other hand, combining the original idea of Particle in Cell (PIC) (Evans et al., 1957) where a fixed mesh is used to calculate forces and pressures and moving particles to convect properties (which avoids the remeshing), and improving the particle integration following the fluid streamlines, recently has born the so-called PFEM-2 method (Idelsohn et al., 2012, 2013). In addition, using an improved explicit integration named X-IVAS (eXplicit Integration following the Velocity and Acceleration Streamlines), there is no limitation in the time step, being the required precision the only bound for the time-step (Gimenez et al., 2014). The enhanced PFEM-2 version to solve multiphase problems, presented in (Idelsohn et al., 2014), also includes enrichment strategies to capture discontinuities.

In this work a review of PFEM-2 for multiphase systems will be presented in Section 2.1. In Section 2.2 a brief explanation of the OpenFOAM®strategies for two-phase problems will be shown. In Section 3, the results show a comparison of the accuracy of PFEM-2 interface evolution strategy with the one implemented in the widely used Eulerian suite. On the other hand, in Section 4, PFEM-2 is tested for a number of problems involving free surface flows with different ratio between the densities of the fluids involved. The accuracy of the results are compared with reference ones, focusing in the capability of using large time-steps in contrast with Eulerian solvers, including those with implicit phase-fractions advection treatment, represented, again, by the OpenFOAM®suite.

## 2 BRIEF DESCRIPTION OF THE NUMERICAL METHODS

### 2.1 PFEM-2 for multi-phase flows

The PFEM-2 method is an efficient and accurate methodology to simulate numerically the dynamics of a flow of two immiscible fluids. In the particular case of using one fluid, the methodology still holds. The governing equations are the incompressible Navier-Stokes equations for both fluids, which are supplemented with the conventional boundary conditions on solid and/or open boundaries. The equations written in a Lagrangian framework are:

$$\nabla \cdot \mathbf{v} = 0 \tag{1}$$

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

Here $\mathbf{v}$, $p$ are the velocity and fluid pressure and $\mathbf{f}$ is a external body force (usually gravity $\rho\mathbf{g}$ and/or inertial force).

Similarly to other Navier-Stokes algorithms, a classical fractional-step scheme is employed, which consists on three main stages: predictor, Poisson equation and correction. Since the use of a hybrid domain discretization, with a fixed background mesh and a cloud of particles, the predictor step is divided in four sub-steps:

1. An acceleration calculation stage over the mesh.
2. The X-IVAS stage to convect the fluid properties using the particles.
3. The projection of the particle data to the mesh nodes.
4. The implicit calculation of the diffusion term.

The predictor step ends with a predicted velocity $\widehat{\mathbf{v}}^{n+1}$ on the mesh. After that, a Poisson equation to find the current pressure $p^{n+1}$ is solved. Finally, the velocity prediction is corrected to find the zero divergence field $\mathbf{v}^{n+1}$.

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 changes implemented are related to the strategies devised to correctly capture the interface between both fluids. An in deep description of these changes can be found in (Idelsohn et al., 2014), being possible to mention three main improvements, 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.

Regarding to interface evolution simulation, each particle $p$ carries the information of the fluid to which it was initially assigned. This quantity, represented by a scalar function $\lambda_p$, has integer values $-1$ or $1$ depending if it belongs to the first or second fluid. This value is advected, adding one equation to the *X-IVAS 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, which is defined as the set of points that satisfy the equation $\lambda = 0$. This strategy, which almost does not include extra computational work to compute the interface evolution, is based on the LSM, with the advantage of not being necessary to restart the distance function.

As it was mentioned, interpolation errors on the interface in the pressure and its gradient due to the density jump give rise to spurious velocities that can render the solution meaningless. To reduce these errors, enrichment methods, which add degrees of freedom to elements that are cut by the interface, are employed. In the Figure 1 the enriched element and an enhanced integration, which cuts the elements into sub-elements, are presented.

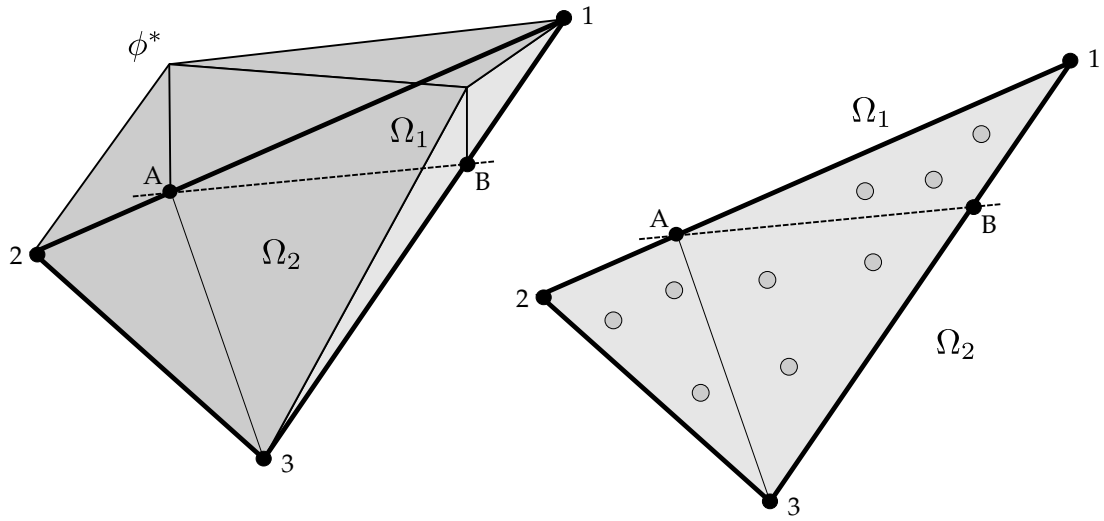



Figure 1: 2D interface element. The interface is calculated cutting the element at the segment $A - B$. An enriched shape function used in PFEM-2 (left) and the partition of the triangle into three sub-triangles with its own Gauss points to enhace the integration (right).

Finally, the Algorithm 1 presents the current PFEM algorithm to solve incompressible fluids with two different densities.

### 2.2 Volume of Fluid solver in OpenFOAM®

In the OpenFOAM®case, the solver `interFoam` is chosen, which implements a Volume of Fluid (VoF) algorithm for multiphase flow (Berberović et al., 2009). 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 (Márquez Damián, 2013)). Since OpenFOAM® version 2.3, a new semi-implicit implementation of MULES is introduced which combines operator splitting with application of the MULES limiter to an explicit correction rather than to the complete flux. This approach allows for boundedness and stability at an arbitrarily large Courant number.

## 3 INTERFACE EVOLUTION TESTS

### 3.1 Rigid Body Rotation of Zalesak's Disk

This test consists in the advection of a region composed of a circle with a slot (Zalesak, 1979). If the interface track is accurate enough, after several revolutions, the shape must remain identical. The computational domain employed is $\Omega \in R^2 : [0; 100] \times [0; 100]$. The advected region is a circle centered at $(50; 75)$ with a radius of $15$ and a slot of width $5$ and height $25$. The velocity field is a rigid body rotation around the center of the domain with a period of 628 time units:

$$u = (\pi/314)(50 - y), \tag{9}$$

$$v = (\pi/314)(x - 50) \tag{10}$$

The grid has $100$ points in each direction, conforming a cartesian mesh (in the case of PFEM simulations the mesh was split into 20000 triangles). The Courant number used in simulations is aproximately $CFL = 4.5$. The initial field and after two revolution are shown on Figure 2. In

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

1. X-IVAS Stage:

$$\begin{cases} \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \sum_{i=1}^{N} \mathbf{v}^n(\mathbf{x}_p^{n+\frac{i}{N}})\delta t \\ \widehat{\widehat{\mathbf{v}}}_p^{n+1} = \mathbf{v}_p^n + \sum_{i=1}^{N} \mathbf{f}^n(\mathbf{x}_p^{n+\frac{i}{N}})\delta t \\ \lambda_p^{n+1} = \lambda_p^n \end{cases} \tag{3}$$

2. Projection Stage:

$$\begin{cases} \widehat{\widehat{\mathbf{v}}}_j^{n+1} = \dfrac{\sum_p \widehat{\widehat{\mathbf{v}}}_p^{n+1} W(\mathbf{x}_j - \mathbf{x}_p)}{\sum_p W(\mathbf{x}_j - \mathbf{x}_p)} \\ \lambda_j^{n+1} = \dfrac{\sum_p \lambda_p^{n+1} W(\mathbf{x}_j - \mathbf{x}_p)}{\sum_p W(\mathbf{x}_j - \mathbf{x}_p)} \end{cases} \tag{4}$$

3. Implicit Viscosity Stage:

$$\int_\Omega \widehat{\mathbf{v}}_j^{n+1}\psi_j \, d\Omega = \int_\Omega \widehat{\widehat{\mathbf{v}}}_j^{n+1}\psi_j \, d\Omega + \theta\Delta t \int_\Omega \mu(\mathbf{x})\nabla^2\widehat{\mathbf{v}}_j^{n+1}\psi_j \, d\Omega \tag{5}$$

4. Pressure-Correction Iterations:
   `set` $\mathbf{v}_j^n = \widehat{\mathbf{v}}_j^{n+1}$
   `for k=1 to K`

   (a) Poisson Stage:

$$\int_\Omega \nabla\cdot\left[\frac{\Delta t}{\rho(\mathbf{x})}\nabla(\delta p^{(n+\frac{k}{K})})\right]\phi_j \, d\Omega \;=\; \int_\Omega \nabla\cdot\mathbf{v}_j^{(n+\frac{k-1}{K})}\phi_j \, d\Omega \tag{6}$$

   (b) Correction Stage:

$$\begin{aligned}\int_\Omega \psi\rho(\mathbf{x})\mathbf{v}_j^{(n+\frac{k}{K})} \, d\Omega = \int_\Omega \psi\rho(\mathbf{x})\mathbf{v}_j^{(n+\frac{k-1}{K})} d\Omega \\ - \Delta t \left[\int_\Omega \psi\nabla(\delta p)_j^{(n+\frac{k}{K})} d\Omega + \int_\Omega \psi\nabla(\delta p^*)^{(n+\frac{k}{K})} \, d\Omega\right]\end{aligned} \tag{7}$$

   `end for`

5. Particle Correction Stage:

$$\rho_p\mathbf{v}_p^{n+1} \;=\; \rho_p\widehat{\widehat{\mathbf{v}}}_p^{n+1} + \sum_j \delta\mathbf{v}_j^{n+1}\psi_j(\mathbf{x}_p^{n+1}) \tag{8}$$

the case of PFEM simulation, approximately five particles by element were used. Most relevant OpenFOAM®settings are: `SuperBee` SuperBee as the divergence scheme for the linear term in volume fraction advection equation, MULES as the time integration scheme, the number of `alphaSubCycles` is 20 (to guarantee interface Courant number less than 0.5) and the interfase-compression factor `cAlpha` is set to 1.

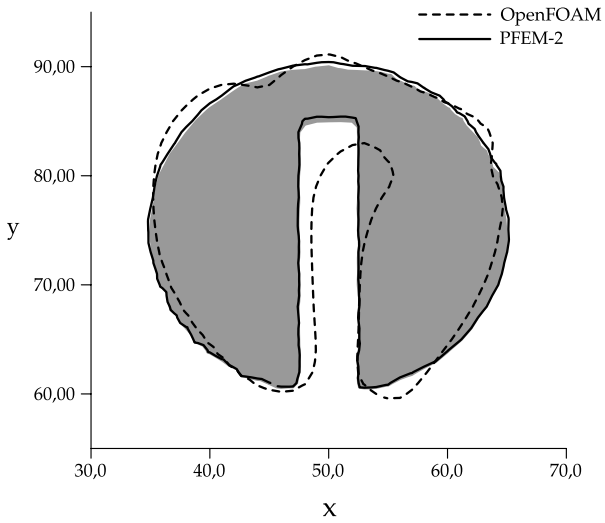



Figure 2: Zalesak's disk results after two full revolutions with 100 grid point per direction and $CFL = 4.5$. The grey region represents the initial condition.

PFEM evolution shows a good agreement with the expected result (shape preservation). Some small errors, which are more evident when the magnitude of velocity is higher, appear due to approximate a curve with a sequence of straight trajectories. Even though in OpenFOAM®simulation the interfase-compression method combined with the advetion scheme avoids numerical diffusion, they modify the disk shape excessively, finishing in a poor prediction of the final status.

### 3.2 Single Vortex Case

While Zalesak's disk test is a good indicator of numerical diffusion in an interface-capturing method, it does not test the ability to preserve small scale structures of the fluid flow. A well known test to evaluate the ability of the method to solve structures of different sizes and their evolution is given by the *vortex-in-a-box* problem introduced by Puckett et al. (Puckett et al., 1997). The difficulty of this tests is that requires the solution of an interface stretching problem. The computational domain is $\Omega \in R^2 : [0;1] \times [0;1]$, where the interest region is a circle centered at $(0.5; 0.75)$ with a radius of $0.15$, advected with a velocity field defined by the stream function

$$\psi(\mathbf{x}) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos(\frac{\pi t}{T})$$

being the velocity components

$$u = \psi_x = \sin^2(\pi x) \sin(2\pi y) \cos(\frac{\pi t}{T})$$
$$v = -\psi_y = -\sin^2(\pi y) \sin(2\pi x) \cos(\frac{\pi t}{T})$$

The grid has $256$ points in each direction, and the Courant number used in simulations is aproximately $CFL = 4.8$.

The setting employed for each numerical method in this case is almost equal to the previous test, with the only one difference that in OpenFOAM®the interfase-compression factor `cAlpha` is set to 0.25 to give more stability through relaxing in some level the strong sharpness imposition. Using a larger factor, the simulation turns unstable.

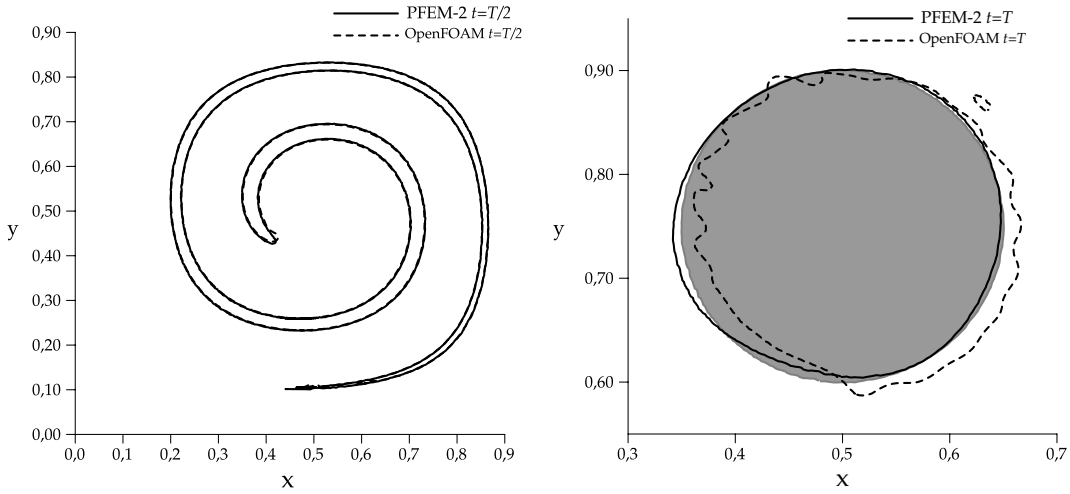



Figure 3: Single vortex test using 256 grid points per direction and $CFL = 4.8$ ($T = 8$). Grey region represents the initial condition.

The results presented in the Figure 3 shows, in PFEM, good agreement with the expected result (shape preservation) after the cycle. Although the first half of the evolution is well captured by OpenFOAM®, the reconstruction of the original shape is not good.

## 4 TWO-PHASE TESTS

### 4.1 Non-linear sloshing in a rectangular container

When big amounts of liquid are transported, the sloshing phenomenon produces violent impacts of the fluid which can affect the structural integrity of the container. This case is widely used as an evaluation test of a two-phase numerical method. Typically, the density of the upper fluid is several orders of magnitude less than the bottom one, however the same numerical method must also keep accurate when the density jump is not too large.

For 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 $\omega_h$ is the excitation frequency of the rectangular tank where the two fluid phases are contained. The tank is divided in two parts, the bottom part is water with a density of $\rho_I = 1000\ [\mathrm{kg/m^3}]$ and the top part contains a fluid with different densities $\rho_{II} = 1.3, 50, 200, 800\ [\mathrm{kg/m^3}]$, 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 Figure 4. 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 work, a 2D rectangular tank $a = 1.0\ [\mathrm{m}]$ width by $b = 1.0\ [\mathrm{m}]$

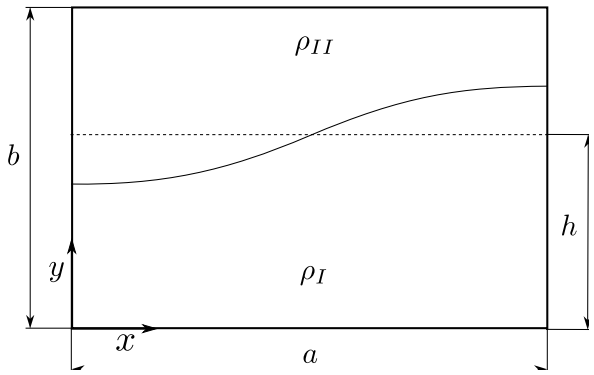



Figure 4: Configuration of the Non-linear sloshing in a rectangular container case. Initial condition is represented by dashed line. Continuous line represents the position of the free-surface for a certain time.

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 used and slip boundary conditions are taken. The density ratio $\sigma = \frac{\rho II}{\rho I}$ was modified to study the density influence on the free surface evolution. A two dimensional Cartesian mesh of $450 \times 225$, splitted into triangles in the case of PFEM, has been used in all cases.

Reference results for this case are taken from (Gómez-Goni et al., 2013) which uses the codes STARCCM+ and OpenFOAM® (both implementing the VoF strategy) 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 simulations such restrictions do not exist, then $\Delta t$ is fixed to $0.01$, reaching $CFL_{max} \approx 5$. Figure 5 presents the free surface displacement reported on the left wall of the container for different density ratios. 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 the reference work.

### 4.2 Rayleigh-Taylor Instability

This problem consists on the evolution of two layers of fluids initially at rest in the gravity field. The density of the fluid placed at the top is larger than the one placed at the bottom. Due to a little disturbance in the contact surface the more dense fluid goes 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 an 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 (Tryggvason, 1988) for inviscid incompressible flows, and by Guermond & Quartapelle (Guermond and Quartapelle, 2000) for viscous flows.

The starting point is the problem documented by Guermond. The computational domain is $[-d/2, d/2] \times [-2d, 2d]$ and the initial position of the perturbed interface is $\eta(x) = 0.1\, d\cos(2\pi x/d)$. 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 $Re = \rho_{min} d^{\frac{3}{2}} g^{\frac{1}{2}}/\mu = 1000$. The

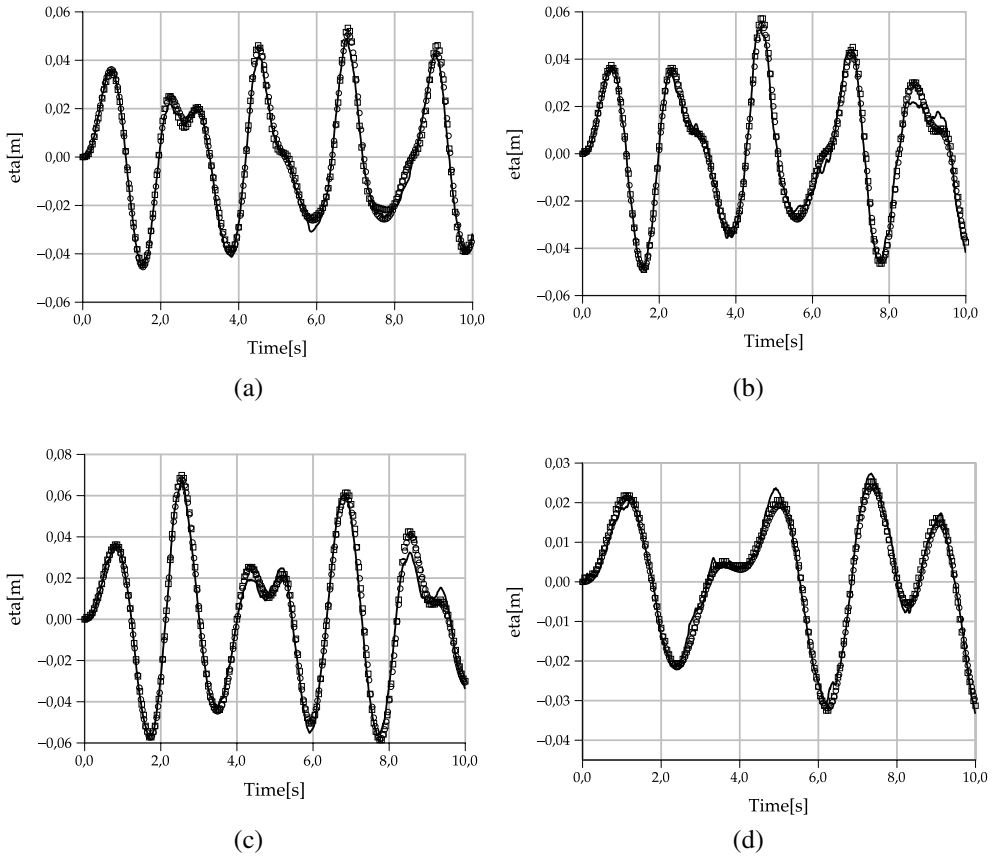



Figure 5: Level of height on the left wall for a two phase flow for different density ratios. Figure 5a: $\sigma = 0.0013$, Figure 5b: $\sigma = 0.05$, Figure 5c: $\sigma = 0.2$ and Figure 5d: $\sigma = 0.8$. References: ○ STARCCM, □ OpenFOAM and filled line PFEM-2.

computational domain is discretized into $80,000$ structured triangles ($\Delta x = 0.01$) setting slip boundary conditions on each wall. Time step selected is $\Delta t = 0.01$, which allows to reach $CFL_{max} \approx 8$. Between five and eight particles per element are used and two pressure iterations are required.

To compare with reference results, the time is made dimensionless by using $\widetilde{t} = t\sqrt{g\ At}$. Results on the vertical position of the tip of the falling and rising fluid (spike and bubble, respectively) are shown in Figure 6. It can be observed that current solution is in good agreement with the reference results.

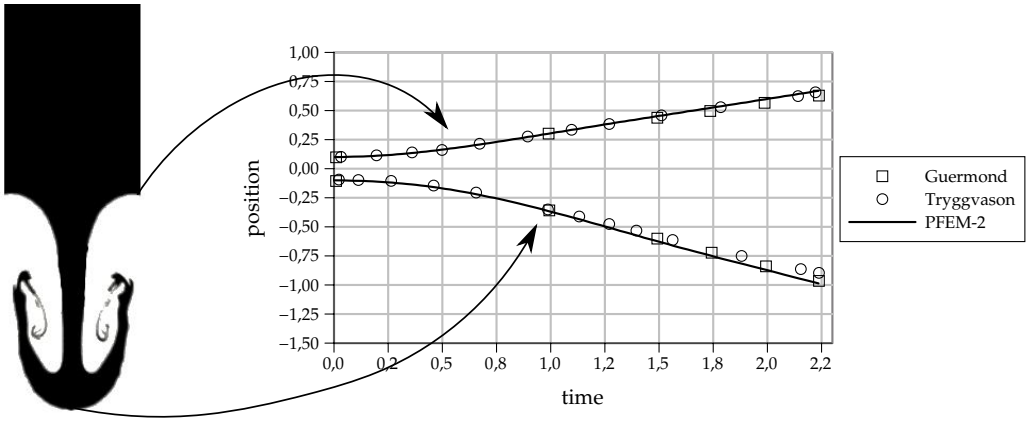



Figure 6: Position of rising and falling bubbles versus time. Case with $Re = 1000$.

### 4.2.1 Extending the time step

In order to make emphasis in the capability of the method to manage large time-steps, the current case is also simulated with a large range of $\Delta t$ using the in-house implementation of PFEM and comparing with results obtained by the widely known suite OpenFOAM®. The problem setup and domain discretization is the same as presented above and the PFEM settings are preserved.

In the OpenFOAM®simulation, the following schemes have been for the momentum equations: `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 CFL of the sub-cycling around $0.5$, `cAlpha`$= 0.25$ to give more stability through relaxing in some level the strong sharpness imposition, and `MULESCorr` and MULESCorr is enabled to integrate the phase fraction equation in a semi-implicit way.

Table 1 presents the comparison of the solutions with PFEM and OpenFOAM®at a particular time ($\widehat{t} = 2.25$) using several fixed time-steps (with the largest time-step a $CFL_{max} = 15$ is reached). From the figure, it is possible to assure that PFEM keeps approximately the same solution for all time-steps, but `interFoam` can not solve the problem with any accuracy using $\Delta t > 0.001$ because the evolution of the mushroom-like interface differs form the reference results and the differences increase with large time-steps. Moreover, each simulation of each simulation of `interFoam` diverges when $CFL_{mean} > 0.5$ is reached (this happens at different times, depending on selected time-step).

| $\Delta t$ [s] | OpenFOAM® | PFEM-2 |
|---|---|---|
| 0.001 |  |  |
| 0.0025 |  |  |
| 0.01 |  |  |
| 0.025 |  |  |

Table 1: Rayleigh-Taylor instability captures for $\widetilde{t} = 2.25$. OpenFOAM®simulation implements VoF+MULES simulation (`interFoam` solver).

Another relevant feature to take into account is that similar CPU times are required to solve a time-step when both algorithms are compared. Table 2 summarizes the CPU Times required to complete $1[\mathrm{s}]$ 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 capability of time-step enlargement of PFEM, it achieves shorter CPU times with similar solutions.

| Solver | $\Delta t$ | CPU Time |
|---|---|---|
| OpenFOAM® | 0.001 | 1121[s] |
| PFEM | 0.0025 | 1011[s] |
| PFEM | 0.01 | 288[s] |
| PFEM | 0.025 | 123[s] |

Table 2: Total computing times to simulate $1[\mathrm{s}]$ of real time of the 2D Rayleigh-Taylor instability, running over an Intel i5-3230M CPU @ 2.60GHz with a 8Gb RAM in one processor.

### 4.3 Towards a simulation of a Jet Atomization

Liquid atomization is an important process which found interest in several engineering applications such as aerospace propulsion systems, automotive engines, food processing, and ink-jet printing. Its numerical simulation allows to investigate physical processes of the atomization because our understanding on physical mechanisms of such phenomena is still not sufficient. Our investigation group is doing its first steps in this research area and we report in this work our early results using the tools presented above.

The main properties of the case analyzed are the following: the size of the domain is $(2.1[\mathrm{mm}],\ 0.3[\mathrm{mm}],\ 0.3[\mathrm{mm}])$, where the first dimension is the streamwise direction and the other two, the spanwise directions. At the injection level, the jet diameter $D$ is equal to $0.1[\mathrm{mm}]$, while the liquid jet Reynolds number is equal to $Re = 4659$. A summary of the physical

parameters, for this configuration, can be found in Table 3. Also, the geometry and boundary conditions are presented in the Figure 7. Boundary condition over *borders* is slip, over *bottom* velocity zero gradient and pressure equal to zero with alpha inlet-outlet where $\alpha = 0$ to entering flows is imposed. Top boundary has two patches: on *inlet* a mean inlet of $\hat{U}_z = -100[\mathrm{m/s}]$ is imposed and over *wall* no-slip condition is set.

| Parameter | Symbol / Unit | Value |
|---|---|---|
| Gas density | $\rho_g$ $[\mathrm{kg/m^3}]$ | 25 |
| Liquid density | $\rho_l$ $[\mathrm{kg/m^3}]$ | 696 |
| Gas viscosity | $\mu_g$ $[\mathrm{kg/m\ s}]$ | $1 \times 10^{-5}$ |
| Liquid viscosity | $\mu_l$ $[\mathrm{kg/m\ s}]$ | $1.18 \times 10^{-3}$ |
| Surface Tension Coefficient | $\sigma$ $[\mathrm{N/m}]$ | 0.06 |
| Injection Diameter | $D_0$ $[\mu\mathrm{m}]$ | 100 |
| Mean flow rate velocity Parameter | $U_d$ $[\mathrm{m/s}]$ | 79 |
| Liquid Reynolds | $Re_l$ | 4659 |
| Liquid Weber | $We_l$ | 7239 |
| Turbulent Intensity | $\overline{u\prime u\prime}/U^2$ | 0.05 |
| Turbulent Scale | $L_t$ $[\mathrm{m}]$ | $0.1D_0$ |

Table 3: Simulation parameters.

As a first reference result, we can cite the work of Menard and co-workers (Ménard et al., 2007; Chesnel et al., 2011), which employ the LSM to track the interface added to the Ghost Fluid Method (GFM) to describe the interface discontinuites and manage the pressure, density and viscosity jumps. Also, the Level Set method is coupled with the Volume of Fluid method (VoF) to ensure mass conservation. The mesh used by Menard in (Chesnel et al., 2011) is a $2048 \times 256 \times 256$ Cartesian grid with regularly spaced nodes ($\Delta = 1.17[\mu\mathrm{m}]$). Liquid surface instabilities close to the injector are visible. Their deformation leads to the formation of ligaments and droplets of various sizes. At the end of the domain, the liquid core has almost disappeared and a dense spray of droplets leaves the computational domain. The key of the quickly drop production is the use of a space-time correlated turbulent flow at the inlet: Menard uses a syntetized correlated turbulence with a method proposed by Klein (Klein, 2003). In the work of Desjardins (Desjardins et al., 2008), the author employes a forerunner simulation to impose the inlet turbulent boundary condition, obtaining similar results to the above mentioned. Both works have a relevant conclusion: by the end of the computational domain, the liquid core has been fully disintegrated.

Another approach in the numerical characterization of jet atomization is reported by Shinjo in (Shinjo and Umemura, 2010, 2011). In his work, the author reports that the grid resolution used by Menard was coarse for the chosen Reynolds and Weber numbers, so this was not a direct numerical simulation in a true sense: the produced ligaments and droplets did not exhibit smooth shapes or wave dynamics driven by surface tension, but the overall liquid jet motion was captured in that simulation. Figure 9b shows the results obtained by Shinjo solving with a mesh with 400 million of cells ($\Delta = 1.5[\mu\mathrm{m}]$) with a domain of $21.7 \times 8.0 \times 8.0[\mu\mathrm{m}]$. The ligament drop is done far from the inlet: the main responsible is the plain velocity front at the inlet imposed by Shinjo instead of using a turbulent-induced flow (Trinh and Chen, 2005).

Our initial simulations using the algorithms presented above (PFEM-2 and `interFOAM`) show some similarities with both results, depending on the inlet condition imposed. It must be

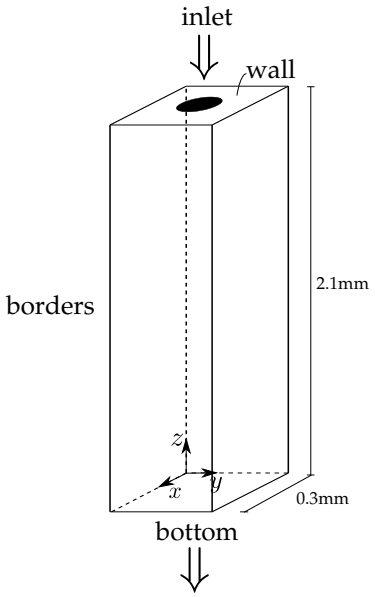



Figure 7: Geometry and boundary conditions for the case of the 3d jet.

taken into account that in the most refined case simulated with OpenFOAM®, the geometry was meshed with a cartesian grid of $128 \times 128 \times 1024$ ($\Delta x \approx 2.3[\mu\mathrm{m}]$), while the PFEM simulations has a $\Delta x \approx 7.5[\mu\mathrm{m}]$ (7 millons of tetrahedra), far from the refinement degree used in reference works.

Figure 8 shows a comparison between our simulation with our minimum reachable mesh-size compared with Shinjo reference results. In this simulation a plain flow is imposed at the inlet. The picture shows that the droplet formation and the like-mushroom shape are comparable, but the minimum drop size is better described using a finer mesh. In that figure also appears the results obtained with PFEM-2, showing the need for a refined mesh, but also demonstrating the potential of the methodology to solve this type of problems.

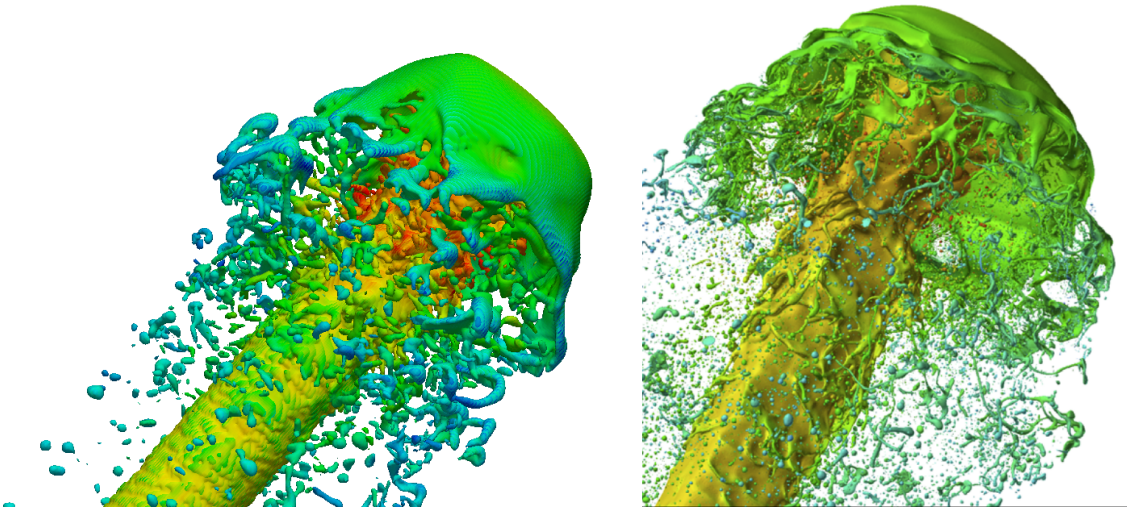
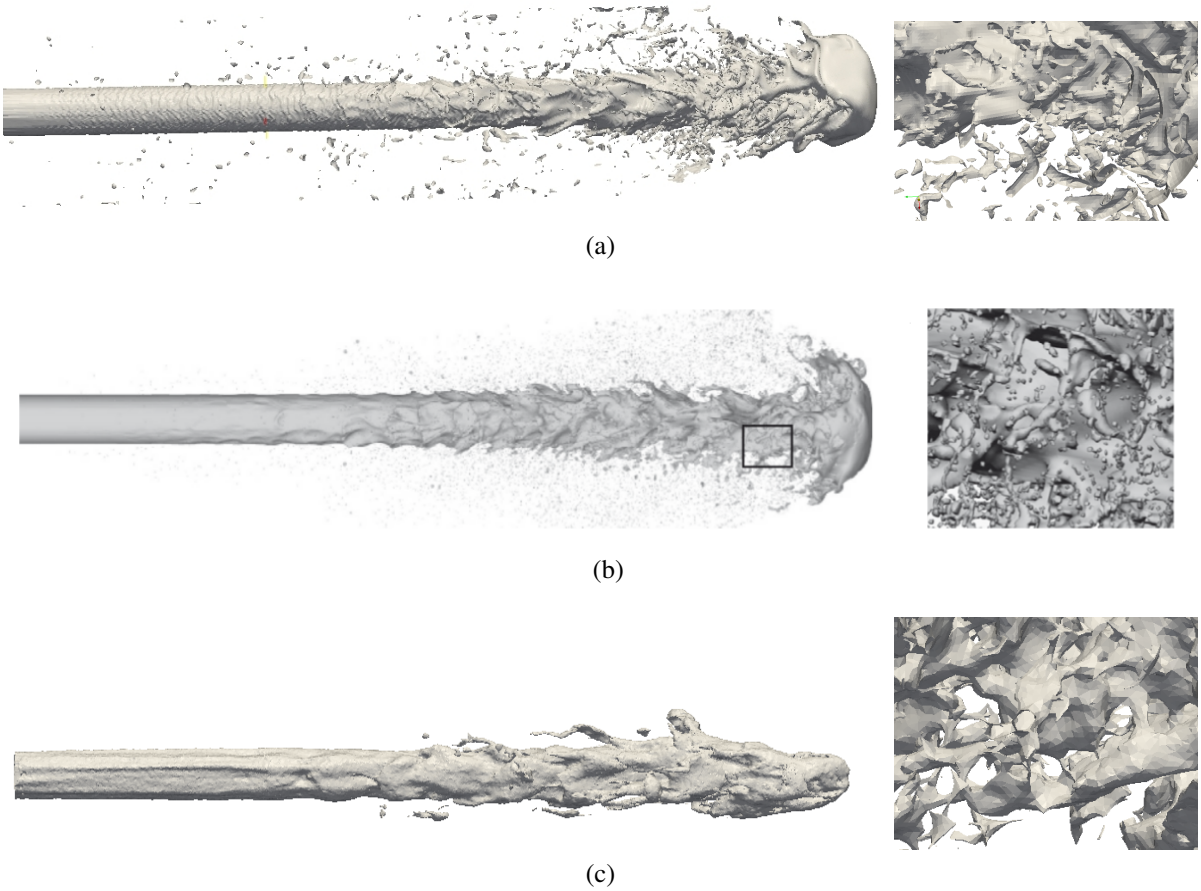


Figure 8: The overall flow structure for each case solving with a mesh $\Delta x = 2.3[\mu\mathrm{m}]$ using OpenFOAM®and compared with Shinjo (Shinjo and Umemura, 2010) which uses a mesh with $\Delta x = 0.35[\mu\mathrm{m}]$. The color indicates the axial velocity in $[\mathrm{m/s}]$.



(a)

(b)

(c)

Figure 9: Overall shape of the liquid jet atomization with: 9a OpenFOAM®in a finer mesh $\Delta x = 2.3[\mu\text{m}]$, 9b reference by Shinjo with $\Delta x = 1.5[\mu\text{m}]$ and 9c PFEM in a coarse mesh $\Delta x = 7.5[\mu\text{m}]$. Figures correspond to iso-surfaces of $\alpha = 0.5$.

### 4.3.1 Vortex-Method

As it was mentioned, the fluid behavior within the domain is determined in large part by the inlet behavior. In the primary jet breakup case, the solution presented by Menard is really different from the fluid evolution reported by Shinjo, being the main responsible the heavy-phase inlet condition. Then, in order to obtain a more realistic behavior of a fully developed flow in an unsteady simulation it is necessary to introduce turbulent velocity time histories at the inlet boundary conditions. In this work, the *2D vortex method* (Sergent, 2002) is employed as a turbulence synthesizer to generate a space-time correlated inlet flow.

The 2D vortex method is based on the modification of the velocity field normal to the streamwise direction: a transverse perturbation is added on a specified mean velocity profile via a fluctuating vorticity field calculated from the specified mean turbulent kinetic energy, $k$. The vortex method is based on the Lagrangian form of the 2D evolution equation of the vorticity:

$$\frac{d\vec{\omega}}{dt} = \vec{\omega} \cdot \nabla \vec{u} + \nu \nabla^2 \vec{\omega} \tag{11}$$

where $\nu$ is the kinematic viscosity. A particle discretization is used to solve this equation. These particles, or *vortex points* are convected randomly and carry information about the vorticity field. If $N$ is the number of vortex points and $A$ is the area of the inlet section, the amount of vorticity carried by a given particle $i$ is represented by the circulation $\Gamma_i$ and an

assumed spatial distribution $\eta$:

$$\Gamma_i(x,y) = 4\sqrt{\frac{\pi A k(x,y)}{3N[2\ln(3) - 3\ln(2)]}} \tag{12}$$

$$\eta(\vec{x}) = \frac{1}{2\pi\sigma^2}\left(2e^{-|x|^2/2\sigma^2} - 1\right)2e^{-|x|^2/2\sigma^2} \tag{13}$$

where $k$ is the turbulence kinetic energy. The parameter $\sigma \approx \frac{C_\mu^{3/4}k^{3/2}}{\epsilon}$ provides control over the size of a vortex particle. The vorticity at a position $\vec{x} = (x,y)$ on the inlet patch generated by the sum of all vortex points $i = 1..N$ is given by

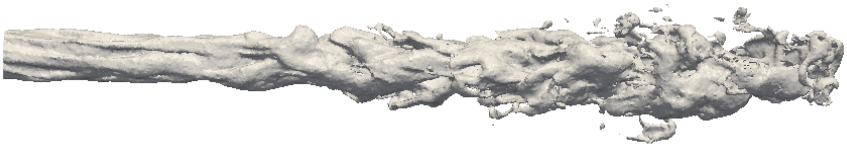
$$\vec{\omega}(x,y) = \sum_{i=1}^{N} \Gamma_i(t)\eta(|\vec{x} - \vec{x}_i|, t) \tag{14}$$

Finally, using the Biot-Savart law which relates the velocity to the vorticity, the resulting discretization for the velocity field is given by

$$\vec{u}(\vec{x}) = \frac{1}{2\pi}\sum_{i=1}^{N}\Gamma_i\frac{((\vec{x}_i - \vec{x}) \times \vec{z})\left(1 - e^{|\vec{x}_i - \vec{x}|^2/2\sigma^2}\right)}{|\vec{x} - \vec{x}_i'|^2} \tag{15}$$

The figure 10 shows a comparison between Menard results and PFEM-2. The imposition of induced turbulence at the inlet allows to PFEM-2 simulation to generate eddies and broken structures near to the inlet zone, facilitating the atomization and liquid disintegration. However, in order to reproduce accurately the drop size and another typical structures, more mesh details is really required.

(a)



(b)

Figure 10: Liquid jet atomization. 10a reference case by Menard, 10b PFEM and vortex method BC in the inlet, using a mesh with $\Delta x = 7.5[\mu\text{m}]$. Figures correspond to iso-surfaces of $\alpha = 0.5$.

## 5 CONCLUSIONS

In this work two different methodologies to solve two phase flows have been presented and compared. The PFEM-2 method includes all the advantages of the Lagrangian strategies to capture accurately the interface evolution, either purely advection and in two-phase problems. The possibility of time-step extension gives to PFEM-2 more computational efficiency to solve some problem: computing times were compared with the widely used FVM software OpenFOAM®, and PFEM-2 has demonstrated being the fastest.

The last example of the jet atomization reveals the potenciality of both strategies for solving that type of problems, being PFEM-2 a interesting alternative due to the capacity of longer time-steps, and then, shorter CPU times. The main drawback in PFEM simulations is that the strategy uses only tetrahedrical elements, requiring meshes with more elements. That must be added to the need for more RAM memory to store all the particle data. Beyond the strategy selected, the computational resources requirement to simulate accurately the drop formation and ligament breakup largely exceeds the computing capacity available today in our research group, however we expect run this cases in larger HPC clusters presenting the results in future publications.

## ACKNOWLEDGMENT

Authors want to thank the financial support of CONICET through doctoral scholarships, Universidad Nacional del Litoral (CAI+D PI PACT 83-A (2009)), PIP 112 201 101 00331 and ERC Advanced Grant REALTIME project AdG-2009325.

## REFERENCES

Ausas R.F., Buscaglia G.C., and Idelsohn S.R. A new enrichment space for the treatment of discontinuous pressures in multi-fluid flows. *International Journal for Numerical Methods in Fluids*, 70(7):829–850, 2012. ISSN 1097-0363.

Berberović E., van Hinsberg N.P., Jakirlić S., Roisman I.V., and Tropea C. Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. *Phys. Rev. E*, 79:036306, 2009.

Chesnel J., Menard T., Reveillon J., and Demoulin F.X. Subgrid analysis of liquid jet atomization. *Atomization and Sprays*, 21(1):41–67, 2011. ISSN 1044-5110.

Coppola-Owen A.H. and Codina R. Improving eulerian two-phase on finite element approximation with discontinuous gradient pressure shape functions. *International Journal for Numerical Methods in Fluids*, 49:1287 – 1304, 2005.

Desjardins O., Moureau V., and Pitsch H. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395 – 8416, 2008.

Enright D., Fedkiw R., Ferziger J., and Mitchell I. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83 – 116, 2002. ISSN 0021-9991.

Evans M., Harlow F., and Bromberg E. The particle-in-cell method for hydrodynamic calculations. Technical Report, DTIC Document, 1957.

Gimenez J., Nigro N., and Idelsohn S. Evaluating the performance of the particle finite element method in parallel architectures. *Journal of Computational Particle Mechanics*, 1:X–X, 2014.

Gingold R.A. and Monaghan J.J. Smoothed particle hydrodynamics, theory and application to non-spherical stars. *Royal Astronomical Society*, 181:375–389, 1977.

Gómez-Goni J., Garrido-Mendoza C.A., Cercos J.L., and González L. Two phase analysis of sloshing in a rectangular container with volume of fluid (vof) methods. *Ocean Engineering*, 73:208 – 212, 2013.

Guermond J. and Quartapelle L. A projection fem for variable density incompressible flows. *Journal of Computational Physics*, 165:167–188, 2000.

Hieber S.E. and Koumoutsakos P. A lagrangian particle level set method. *Journal of Computational Physics*, 210(2):342 – 367, 2005.

Hirt C. and Nichols B. Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comput. Phys*, 39(1):201–225, 1981.

Idelsohn S., Nigro N., Gimenez J., Rossi R., and Marti. J. A fast and accurate method to solve the incompressible navier-stokes equations. *Engineering Computations*, 30-Iss:2:197–222, 2013.

Idelsohn S., Nigro N., Limache A., and Oñate. E. Large time-step explicit integration method for solving problems with dominant convection. *Comp. Meth. in Applied Mechanics and Engineering*, 217-220:168–185, 2012.

Idelsohn S., Oñate E., and Del Pin F. The particle finite element method a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal of Numerical Methods*, 61:964–989, 2004.

Idelsohn S.R., Marti J., Becker P., and 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. ISSN 1097-0363.

Klein. A digital filter based generation of inflow data for spatially developing direct numerical

or large eddy simulations. *Journal of Computational Physics*, 186(2):652 – 665, 2003.

Márquez Damián S. *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.

Monaghan J. An introduction to sph. *Computational Physics Communications*, 48:89–96, 1988.

Ménard T., Tanguy S., and Berlemont A. Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet. *International Journal of Multiphase Flow*, 33(5):510 – 524, 2007.

Osher S. and Sethian J.A. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton–jacobi formulations. *Journal of Computational Physics*, 79:12 – 49, 1988.

Puckett E., Almgren A., Bell J., Marcus D., and Rider W. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *J. Comput. Phys.*, 130(2):269 – 282, 1997.

Sergent M. *Vers une Méthodologie de Couplage Entre la Simulation des Grandes Echelles et les Modèles Statistiques*. Ph.D. thesis, Ecole Central de Lyon, 2002.

Shinjo J. and Umemura A. Simulation of liquid jet primary breakup: Dynamics of ligament and droplet formation. *International Journal of Multiphase Flow*, 36(7):513 – 532, 2010.

Shinjo J. and Umemura A. Surface instability and primary atomization characteristics of straight liquid jet sprays. *International Journal of Multiphase Flow*, 37(10):1294 – 1304, 2011.

Sussman M. and Puckett E.G. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301 – 337, 2000. ISSN 0021-9991.

Trinh H.P. and Chen C.P. Modeling of turbulence effects on liquid jet atomization. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*. 2005.

Tryggvason G. Numerical simulations of the rayleigh-taylor instability. *Journal of Computational Physics*, 75:253–282, 1988.

Weller H.G., Tabor G., Jasak H., and Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620–631, 1998.

Zalesak S. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31(3):335 – 362, 1979.