AN ALE FINITE ELEMENT METHOD FOR THE SIMULATION OF 3D MULTIPHASE FLOWS

G. Rabello, gustavo.rabello@gmail.com
J. Pontes, jopontes@metalmat.ufrj.b
UFRJ

L.M. Portela, L.Portela@tudelft.nl
TU-Delft

F. S. Sousa, f.s.sousa@gmail.com
Department of Applied Mathematics and Statistics, ICMC, University of São Paulo, São Carlos, Brazil.
fsimeoni@lcad.icmc.usp.br

N. Mangiavacchi, norberto.mangiavacchi@gmail.com
GESAR/UERJ, Department of Mechanical Engineering, State University of Rio de Janeiro, Rio de Janeiro, Brazil.

Abstract. *Multiphase bubbly flows are important in many industrial applications. One important application is the simulation of hydrate formation in oil and gas industry. In fact, hydrate formation is considered one of the most important problems in the oil industry. As off-shore production migrates to deeper zones, hydrate formation in lines and production equipment becomes a more severe restriction. The study of the first stages of hydrate formation requires the knowledge of lift and drag forces over bubbles and drops. Hence, detailed modeling of bubble flows and interface phenomena is required for the understanding of hydrate formation dynamics.*
This work presents a novel method to solve 3D bubble flows. The method is based on a ALE approach on an unstructured Finite Element discretization using the Taylor-Hood Mini element. The surface tension effects are included in a consistent way, avoiding spurious oscillations and parasitic currents.
The convective terms are discretized using a semi-Lagrangian technique, which is unconditionally stable for arbitrary integration time steps and Reynolds numbers. The pressure and velocity computations are decoupled employing a projection method based on an approximate block LU factorization, producing a very efficient methodology.
Interface is represented by faces, edges and vertices of the tetrahedralization. The interfacial vertices are moved in a Lagrangian fashion, while the other vertices are moved in order to keep the good quality of the mesh, thus avoiding remeshing procedures.
The method is an extension of the method described in [Sousa et al.] for the simulation of 3d flows.
The method was validated against analytical solutions in simple cases, as well as experimental results with bubbles, showing excellent mass conservation properties, stability, and computational performance.

Keywords: *moving mesh; level-set; projection method; finite element; two-fluid flows*
FAPESP and CNPqFAPESP 01/12623-5 and 00/03385-0, CNPq 460473/2001-8

1. INTRODUCTION

Simulations of multi-fluid flows are known to be difficult to perform due to discontinuities at the fronts separating the different fluids. A number of methods have been developed to approximate the fronts. For fixed meshes, front-tracking and front-capturing methods [15] are the most used. In front-tracking methods, the fronts are represented by computational elements, usually a mesh of connected marker particles, moving through the domain with the fluid velocity field. A number of papers dealing with front-tracking methods can be found in the literature (e.g. [3, 15]). The front-tracking methodology is more accurate than front-capturing, introducing very small mass variation of the fluids involved in the simulation. However, its implementation is more difficult, in particular when the flows undergo topological changes, like either coalescence or splitting of the interfaces.

On the other hand, front-capturing methods represent the interfaces by a region of high gradient variation, where the fronts are reconstructed at each time-step. Among these, the level-set method, introduced by Osher and Sethian [7], has acquired popularity because of its algorithmic simplicity. In this method, the fronts are represented by the zero level set of a function ϕ, that is advected by solving $\phi_t + \mathbf{u} \cdot \nabla\phi = 0$, where **u** is the velocity field. Most numerical procedures designed to solve this equation will introduce artificial diffusion leading to pronounced mass conservation errors. Improvements for the mass conservation of the level-set method can also be found in literature (e.g. [9]).

Hybrid methods were also developed, trying to combine the precision of front-tracking with the simplicity of front-capturing. Tornberg and Engquist [14] developed an hybrid method called Segment Projection method, where good accuracy and mass conservation were achieved while still dealing with topological changes. More recently, a work published by Sousa et. al. [11] present a front-tracking/front-capturing technique, which is basically a finite-difference front-tracking technique which allows automatic coalescence at a grid level, but not interface splitting.

With the increasing in the computational resources available, moving-mesh techniques are becoming popular and gaining interest from many researchers. Although this is still a new methodology nowadays, we can find succesfull moving-mesh and Lagrangian methods developed in early 80's and 90's. For instance, Ryskin and Leal describe in [10] a finite difference technique where the mesh is adapted to the deformation of the front, computed in a boundary-fitted coordinate system. This technique is applied for the computation of deformable particles, but resolving only the continuous fluid. In their work from 1990, Ogũz and Prosperetti [6] developed a Lagrangian boundary integral method to predict the bubble entrainment by the impact of drops on liquid surfaces, resolving only the free-boundary. More recently, Perot and Nallapati [8] designed a moving-mesh technique to solve free-surface flows, where springs are used to control the quality of the unstructured mesh.

In this work, we propose a new moving mesh methodology, which is an ALE method for two-fluid flows simulations. It uses the same pseudo-concentration function ϕ to represent the fronts, but its advection is performed indirectly by moving all the vertices of the interface with the velocity field, where the values of the Heaviside function are stored, and values of ϕ are recomputed. The computation of the interface curvature, used to determine the surface tension, uses a level-set method. With this approach the mass conservation problems of the level-set method are avoided, while still keeping the simplicity of the interface representation. However, as the mesh is moved, elements can be distorted, degrading the finite element solution. This requires a mesh control procedure to ensure the good quality of the elements.

2. FORMULATION

The conservation equations modelling incompressible two-fluid flows are the equation of motion

$$\frac{D(\rho\mathbf{u})}{Dt} + \nabla p = \frac{1}{Re}\nabla\cdot\left[\mu\left(\nabla\mathbf{u}+\nabla\mathbf{u}^T\right)\right] + \frac{\rho}{Fr^2}\mathbf{g} + \frac{1}{We}\mathbf{f} \qquad (1)$$

and the equation of continuity $\nabla\cdot\mathbf{u}=0$, where $\mathbf{u}$ is the velocity field, p is the pressure field, μ and ρ are the discontinuous viscosity and density, $\mathbf{g}$ represents the gravitational acceleration field and $\mathbf{f}$ is a source term representing the surface tension. In this equation, Re, Fr and We are the non-dimensional Reynolds, Froude and Weber numbers. Using the CSF model according to [1], the source term can be written as $\mathbf{f}=\sigma\kappa\delta\mathbf{n}$, where σ is the surface tension coefficient, κ is the curvature and $\mathbf{n}$ is the surface unit outward normal vector. Additionally, δ is the Dirac delta function with support on the interface. In the Lagrangian approach the material derivative is approximated by a discretized form of $D(\rho\mathbf{u})/Dt$ computed in a reference frame moving with the material particles. Since the convective term does not explicitly appear in the Lagrangian formulation, high Reynolds numbers are allowed without any special stabilization technique. If Ω_i represents the domain of fluid i, then $\rho=\rho(\mathbf{x})=\rho_i$ if $\mathbf{x}\in\Omega_i$ and $\mu=\mu(\mathbf{x})=\mu_i$ if $\mathbf{x}\in\Omega_i$, where ρ_i and μ_i are the properties of fluid i.

3. NUMERICAL METHOD

The numerical procedure implemented to solve the conservation equations is based on the *Projection-1* method, initially proposed by Chorin [2], and formalized by Gresho and his co-workers [4, 5]. The general idea is to split the velocity field in two fields, one divergence-free and another rotational-free. This procedure decouples the acceleration and pressure fields from the equations. Thus, instead of solving one large system, we solve two smaller decoupled systems of equations, reducing the time of computation.

The method presented below is the Chorin's Projection-1 method, with the treatment of the boundary conditions given by Gresho [4]. The method was adapted for the two-fluid flow case, following the ideas stated by Sousa et al [11]. Consider the conservation equations (1), with initial condition $\mathbf{u}(\mathbf{x},0)=\mathbf{u}_0(\mathbf{x})$ in Ω with $\nabla\cdot\mathbf{u}_0=0$ in Ω. Let $\mathbf{u}^n=\mathbf{u}(\mathbf{x},t)$ be the velocity field computed in the time step t, with $\nabla\cdot\mathbf{u}^n=0$. Given $\mathbf{u}^n$, the procedure to find $\mathbf{u}^{n+1}=\mathbf{u}(\mathbf{x},t+\Delta t)$ in the next time step $t+\Delta t$ is given by the following algorithm:

1. Solve $\tilde{\mathbf{u}}^{n+1}$, with $\tilde{\mathbf{u}}^n=\mathbf{u}^n$, from

$$\frac{D(\rho\tilde{\mathbf{u}})}{Dt} = \frac{1}{Re}\nabla\cdot\left\{\mu\left(\nabla\tilde{\mathbf{u}}+\nabla\tilde{\mathbf{u}}^T\right)\right\} + \frac{\rho}{Fr^2}\mathbf{g} + \frac{\sigma\kappa\delta}{We}\mathbf{n}\,; \qquad (2)$$

2. Solve φ from $\nabla\cdot\rho^{-1}\nabla\varphi=\nabla\cdot\tilde{\mathbf{u}}^{n+1}$;

3. Compute $\mathbf{u}^{n+1}=\tilde{\mathbf{u}}^{n+1}-\rho^{-1}\nabla\varphi$;

4. Update the time step and continue until the final time or convergence are reached.

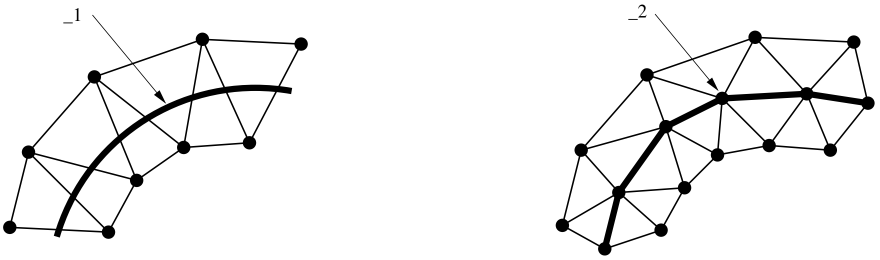

Figure 1. Interface representation: standard level-set (left) and the Lagrangian approach (right).

4. DISCRETIZATION

The domain is discretized by an unstructured triangular mesh which is initially a Delaunay triangulation. The shape functions interpolating the discrete approximations are assumed to be linear, and each triangle of the mesh has at most three degrees of freedom for the velocities and for the pressure. Considering the following spaces: $V = \left\{\mathbf{v} \in L^2(\Omega) : v_i \in H^1(\Omega), \forall\, i\right\}$ and $P = \left\{q \in L^2(\Omega) : \int_\Omega q\, d\Omega = 0\right\}$, where $H^1(\Omega)$ is a Sobolev space, and the sub-spaces $V_{\mathbf{u}_\Gamma} = \{\mathbf{v} \in V : \mathbf{v} = \mathbf{u}_\Gamma \text{ em } \Gamma_1\}$ and $V_{\mathbf{0}} = \{\mathbf{v} \in V : \mathbf{v} = \mathbf{0} \text{ em } \Gamma_1\}$, the weak formulation of the problem can be written as: find $\mathbf{u}(\mathbf{x}, t) \in V_{\mathbf{u}_\Gamma}$ and $p(\mathbf{x}, t) \in P$ such that

$$m(\frac{D(\rho\mathbf{u})}{Dt}, \mathbf{w}) - c(p, \mathbf{w}) + k(\frac{\mu}{Re}, \mathbf{u}, \mathbf{w}) - m(\frac{\rho}{Fr^2}\mathbf{g}, \mathbf{w}) - f_I(\frac{\sigma}{We}, \mathbf{w}) = 0 \tag{3}$$

$$c(q, \mathbf{u}) = 0, \tag{4}$$

for all $\mathbf{w} \in V_{\mathbf{0}}$, $q \in P$. The functionals in (3) are from the integration of the material derivative (m), pressure gradient (c), viscous term (k), and interfacial force term (f_I), respectively.

The discretization of (3)-(4) is made using linear shape functions and Galerkin weighting functions. Integrating over the triangular elements results in the following ODE system M $\dot{u} + Ku + K_{xx}u + K_{xy}v + C_x p - Mg_x + f = 0$
$M\dot{v} + Kv + K_{yx}u + K_{yy}v + C_y p - Mg_y + f = 0$
$C_x u + C_y v = 0$ In (4)-(4), the quantities u, v, p, g, and f are vectors of dimension s, where s is the total number of free nodes in the discretized domain. The ODE's (4)-(4). are solved using the projection method described above. The time derivatives are integrated by an implicit scheme. As in the Lagrangian formulation the non-linear terms do not appear, the matrices are symmetric positive definite and thus the conjugate gradient method can be applied to solve the linear systems.

5. LEVEL-SET

5.1 Interface representation

The interface is computed explicitly, and is part of the moving mesh. However, for the computation of the curvature a level-set methos is employed [12, 13], in which the front is represented by the zero level set of a pseudo-concentration function $\phi : R^2 \to R$. In symbols, $I = \{\mathbf{x} : \phi(\mathbf{x}, t) = 0\}$. With this formulation, the unit normal vector and the curvature of the interface I can be easily calculated by the expressions: $\mathbf{n} = \nabla\phi/|\nabla\phi|$ and $\kappa = \nabla \cdot (\nabla\phi/|\nabla\phi|)$.

This function is advected through the domain by the velocity field. As the interface mesh is moved with the fluid particles and the values of the Heaviside function H are stored in each vertex of the mesh, there is no need to solve any additional equation for the evolution of ϕ. We use additional vertices and edges of the computational mesh to explicitly represent the interface, so that the exact discrete position of the interface is known. Figure 1 shows the representation of the interface in our method compared to standard Eulerian approach.

To avoid problems with discontinuous properties at the interface, we use a regularized *Heaviside* function to compute μ and ρ.

5.2 Reinitialization

The Level-set function ϕ is computed as a signed distance function, which means that $||\nabla\phi|| = 1$. Keeping ϕ a distance function is important because then $\kappa = \nabla^2\phi$, which provides a more accurate computation of the curvature. This is achieved by the recomputation of ϕ. Periodically, the new values of ϕ are computed as

$$\tilde{\phi}(\mathbf{x}) = \min_{y \in I} |\mathbf{x} - \mathbf{y}| \cdot (\phi) \tag{5}$$

This operation is done every time-step and it costs $N_D \times N_I$ computations of two-dimensional distances, where N_I is the number of interface vertices and N_D is the number of the remaining vertices in the mesh. This is an acceptable value since the interface has few points compared to the whole domain. Also, the requirement of keeping ϕ a distance function is only important in the vicinity of the interface which reduces drastically the number of distance computations.

6. NUMERICAL RESULTS

In this section, results for the static and oscillating bubble computation are presented, demonstrating the advantages of this method in terms of mass conservation compared to standard Eulerian level-set methods.

6.1 Static bubble

A static bubble immersed into another fluid is simulated to verify the surface tension calculation and measure the influence of parasitic currents in the flow.

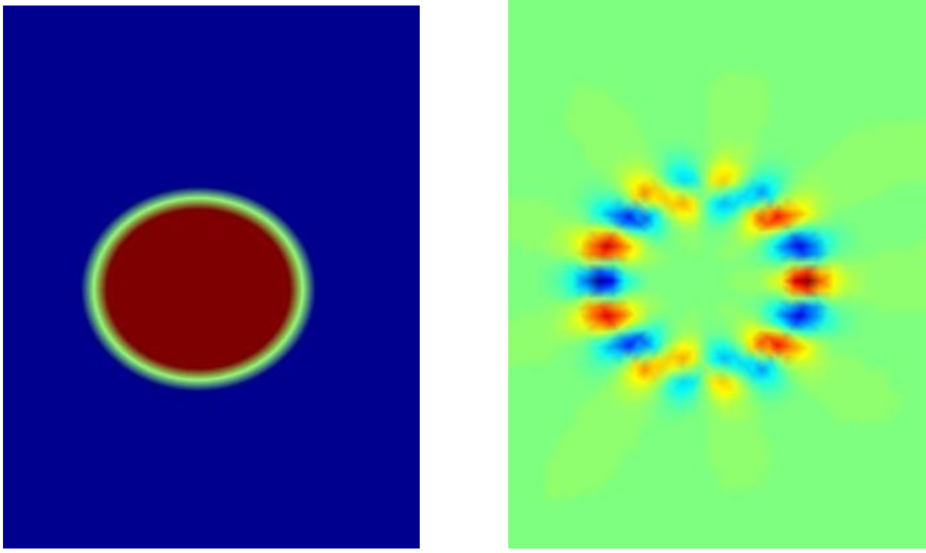

Figure 2. Pressure (left) and x-velocity (right) for a static bubble. The parasitic velocities are of the order of $u\mu/\sigma = 10^{-5}$
.

The parasitic currents are $O(10^{-5})$ in this simulation, which is an acceptable value for our simulations.

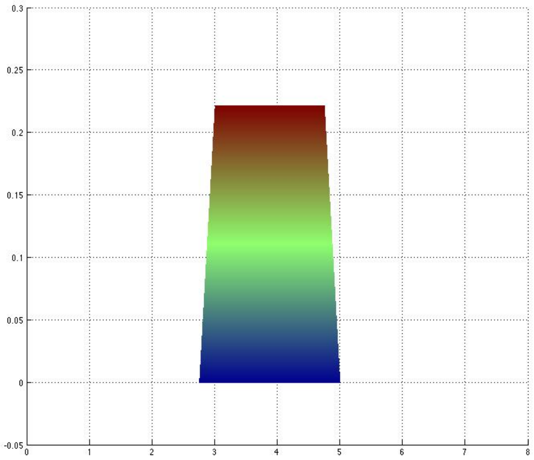

Figure 3. Pressure in a vertical plane accros the center of the bubble. The graphic shows a sharp variation from the zeo value, outside of the bubble, to the value at the horizontal plateau, inside the bubble. The relative error in the value of the pressure jump is about 10%

.

6.2 Oscillating bubble

A oscillating bubble immersed into another fluid is simulated to verify the surface tension calculation in a dynamic situation.

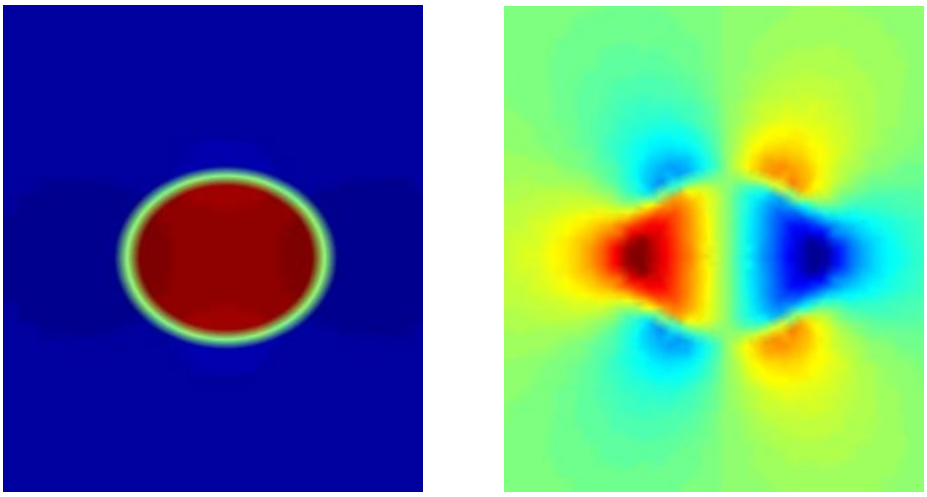

Figure 4. Pressure (left) and x-velocity (right) for an oscillating bubble. The ratio between the horizontal and vertical dimensions of the bubble is 1.2.

.

This result shows good qualitative agreement with other results reported in the literature[11].

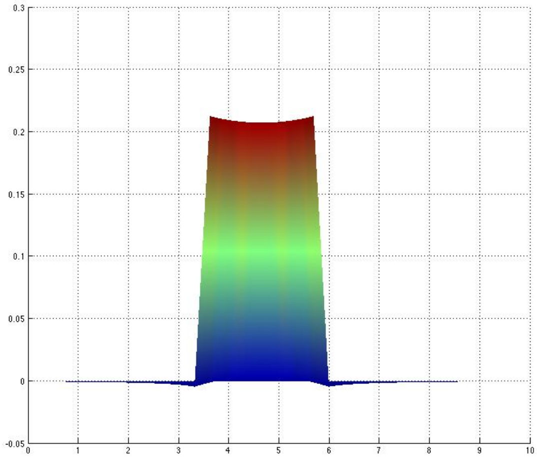

Figure 5. Pressure in a vertical plane accros the center of the bubble. The graphic shows the typical "hat" shape distribution, with sharp variation and absence of oscillations.

.

7. CONCLUSION

This paper presents a method to simulate incompressible multi-fluid flows in which the mesh moves in an ALE fashion. The interface between fluids was represented by vertices and edges of the triangulation plus a level-set of a Heaviside function, which is also used to compute the interfacial force distribution. The discussion focuses the computation of the mesh velocity, as well as implementations on moving finite element unstructured meshes, using remeshing procedures to avoid bad elements. The conservation equations are solved by a projection method based on approximated block *LU* decomposition of the system of equations, to decouple the acceleration and pressure.

Validations were performed for the approximation of the interfacial force, by the simulation of static bubble and oscillating drop. The static bubble simulation showed that the parasitic currents are very small, as expected, and a convergence study reveals second order convergence for the curvature calculation. The oscillating drop simulation also showed good agreement for the oscillation frequency, with error comparable to other works in literature.

Results for the rising bubble are compared for several mesh velocities, showing the better mass conservation properties obtained with the introduction of an elastic velocity, computed by a Laplacian filter.

Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics*, 1992; **100**:335–354.

Chorin AJ. Numerical solution of the Navier-Stokes equations. *Math. of Computation*, 1968; **22**:745–762.

Esmaeeli A, Tryggvason G. Direct numerical simulations of bubbly flows. Part 1. Low Reynolds number arrays. *Journal of Fluid Mechanics*, 1998; **337**:313–345.

Gresho P. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via the finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory. Part 2: Implementation. *Int. Journal for Numerical Methods in Fluids*, 1990; **11**:587–659.

Gresho P, Sani R. On pressure boundary conditions for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 1987; **7**:1111–1145.

Ogũz HN, Prosperetti A. Bubble entrainment by the impact of drops on liquid surfaces. *Journal of Computational Physics*, 1990; **219**:143–179.

Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton Jacobi formulations. *Journal of Computational Physics*, 1988; **79**:12–49.

Perot B, Nallapati R. A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows. *Journal of Computational Physics*, 2003; **184**:192–214.

Pijl SP, Segal A, Vuik C. Modelling of three-dimensional bubbly flows with a mass-conserving level-set method. Proceedings of ECCOMAS 2004, Jyvaskyla, Finland, 2004; Volume II, CD-ROM.

Ryskin G, Leal LG. Numerical solution of free-boundary problems in fluid mechanics. *Journal of Fluid Mechanics*, 2004; **148**:1–43.

Sousa FS, Mangiavacchi N, Nonato LG, Castelo A, Tome MF, Ferreira VG, Cuminato JA, McKee S. A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *Journal of Computational Physics*, 2004; **198**:469–499.

Sussman M, Smereka P. Axisymmetric free boundary problems. *J. of Fluid Mech.*, 1997; **341**:269–294.

Tornberg AK, Engquist B. A finite element based level set method for multiphase flow applications. *Computing and Visualization in Science*, 2000; **3**:93–101.

Tornberg AK, Engquist B. The Segment Projection Method for Interface Tracking. *Communications on Pure and Applied Mathematics*, 2003; **56**:47-79.

Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 1992; **100**:25–37.

Zienkiewicz OC, Taylor RL. *The finite element method volume 1: The basics.* Butterworth-Heinemann, 5th edition, 2000.