ADAPTIVE FINITE ELEMENT BASED LEVEL SET APPROACH FOR THE SIMULATION OF BUBBLE INTERACTION

Nestor M. Solalinde*, Norberto Mangiavacchi°

*GESAR/UERJ, Department of Mechanical Engineering, State University of Rio de Janeiro, Rio de Janeiro, Brazil, Email:manolosolalinde@gmail.com
° GESAR/UERJ, Department of Mechanical Engineering, State University of Rio de Janeiro, Rio de Janeiro, Brazil, Email: norberto.mangiavacchi@gmail.com

ABSTRACT

In this work, we present a level set approach for the simulation of bubble-surface interaction. The problem is focused on the hydrodynamic interaction of two air bubbles immerse in water. The level set method is used to capture the interface between the bubbles. The model consists on the incompressible Navier-Stokes equations, coupled with an advection equation for the level set function. The so-called Continuum Surface Force approach (CSF) is used to model the effect of surface tension. The spatial discretization is based on an unstructured Finite Element mesh that uses adaptive mesh refinement for higher accuracy. Refinement levels are set to be high on regions near the interface and low away from it. For the discretization of velocity, pressure and the level set function we use standard (LBB stable) finite element spaces. Discretization of the time derivative is given by backward differentiation resulting in an implicit approximation. The finite element algorithm is implemented using the *libmesh* library.

INTRODUCTION

Many natural processes involve bubbles. They play a major role in the interaction of the oceans with the atmosphere, for example, and both air bubbles near a free surface and cavitation bubbles are of major importance for the detection of submarines in naval applications. The study of bubble interaction is also important for the development of industrial equipment such as bubble-driven circulation systems used in metal processing operations such as steel making, ladle metallurgy, and the secondary refining of aluminum and copper. [1]

The main difficulty encountered performing multiphase fluid simulations is related to the discontinuities at the fronts separating different fluids. A number of methods have been developed to approximate the fronts. Among these, the level set method, introduced by Osher and Sethian [2], has acquired popularity because of its algorithmic simplicity. In this method, the fronts are represented by a zero level set of a function $\emptyset$, that is advected by solving $\emptyset_t + \boldsymbol{u} \cdot \nabla\emptyset = 0$, where $\mathbf{u}$ is the velocity field. Most numerical procedures designed to solve this equation will introduce artificial diffusion leading to pronounced mass conservation errors.

In this work, the method is based on a level-set formulation discretized by a finite element technique. Surface tension forces acting at the interfaces separating the two fluids, as well as density and viscosity jumps across such interfaces have been integrated into the finite element framework. This method is based on the weak-formulation of the Navier Stokes Equations, where the singular surface tension forces, with the strength directly defined by the interface shape, are included through line integrals along the interfaces. The discontinuous density and viscosity are included in the finite element integrals.

As higher computational capabilities and resources become available day by day, adaptive mesh refining techniques are becoming popular and gaining interest from many researchers. These techniques are meant to reduce the computational costs of the algorithm and increase overall accuracy at the same time. As it is well known, the level set approach is commonly known to present mass conservation problems. By implementing adaptive mesh refinement, together with a mass conserving reinitialization function for the level set equation, we have observed that mass conservation problems of the level-set method are reduced to just a tiny fraction.

GENERAL FORMULATION

Let us consider a domain $\Omega \subset \mathbb{R}^2$, that contains two different immiscible incompressible Newtonian phases (eg. fluid – gas). The time-dependent domains which contain the phases are denoted by $\Omega_1 = \Omega_1(\mathrm{t})$ and $\Omega_2 = \Omega_2(\mathrm{t})$ with $\overline{\Omega_1} \cup \overline{\Omega_2} = \overline{\Omega}$. The interface between the two phases $(\partial\Omega_1 \cap \partial\Omega_2)$ is denoted by $\Gamma = \Gamma(t)$. The equations describing the immiscible multiphase incompressible flow are essentially the Navier Stokes equations for incompressible flow. The contribution of the surface tension forces $\mathbf{f}$, is in addition to the gravity forces added as a source term. The equations, assuming continuity of the velocity across the interface, can be written:

$$\rho(\phi)\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla\mathbf{u}\right) = -\nabla p + \frac{\rho(\phi)}{Fr^2}\boldsymbol{g} + \frac{1}{We}\boldsymbol{f} + \\ + \frac{1}{Re}\nabla \cdot [\mu(\phi)(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^{\boldsymbol{T}})] \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

$$\emptyset_t + \boldsymbol{u} \cdot \nabla\emptyset = 0 \quad (3)$$

where $\boldsymbol{u}$ is the velocity field, p is the pressure field, μ and ρ are the discontinuous viscosity and density, $\boldsymbol{g}$ represents the gravitational acceleration field. In equation (1), *Re, Fr* and *We* are the non-dimensional Reynolds, Froude and Weber numbers. Furthermore, we assume Dirichlet boundary conditions for the velocity and Newman for the pressure, with

one Dirichlet point exception to remove the non-trivial null space of constant pressure solutions.

The effect of the surface tension can be expressed in terms of a localized force at the interface, by the so-called continuum surface force (CSF) model. [5]

$$\boldsymbol{f} = \sigma k \delta_{\Gamma} \boldsymbol{n} \tag{4}$$

where σ is the surface tension coefficient, k is the curvature, δ_{Γ} is the Dirac delta function with support on Γ, and $\boldsymbol{n}$ is the surface unit outward normal vector. The unit normal and the mean curvature are then easily computed. They are given by

$$\boldsymbol{n} = \frac{\nabla \emptyset}{|\nabla \emptyset|} \tag{5}$$

$$k = \nabla \cdot \frac{\nabla \emptyset}{|\nabla \emptyset|} \tag{6}$$

The jumps in the coefficients, μ and ρ can be described using the level set function (which has its zero level precisely at the interface Γ) in combination with the Heaviside function,

$$H(\zeta) = 0 \quad \text{for} \quad \zeta < 0, \qquad H(\zeta) = 1 \quad \text{for} \quad \zeta > 0 \tag{7}$$

$$\rho(\phi) = \rho_1 + (\rho_2 - \rho_1) H(\phi) \tag{8}$$

$$\mu(\phi) = \mu_1 + (\mu_2 - \mu_1) H(\phi) \tag{9}$$

where we can take $H(0) = 1/2$. However, as a result of the discontinuous density and viscosity jumps, numerical instabilities are presented, and since $\delta_{\Gamma} = 0$ almost everywhere, except on the interface (which has measure 0), it seems unlikely that any standard numerical approximation based on sampling will give a good approximation of an integral of equation (4). Thus, we define the smeared-out Heaviside function

$$H_{\epsilon}(\emptyset) = \begin{cases} 0 & \emptyset < -\epsilon \\ \frac{1}{2} + \frac{\emptyset}{2\epsilon} + \frac{1}{2\pi} sin\left(\frac{\pi \emptyset}{\epsilon}\right) & -\epsilon \le \emptyset \le \epsilon \\ 1 & \epsilon < \emptyset \end{cases} \tag{10}$$

where ϵ is a parameter that determines the size of the bandwidth of numerical smearing. A typically good value is $\epsilon = 1.5\Delta x$ [11], where Δx can be taken as an approximate measure of the grid. Then, the smeared-out delta function is defined as

$$\delta_{\epsilon} = \frac{\partial \mathrm{H}_{\epsilon}}{\partial \emptyset} = \begin{cases} 0 & \emptyset < -\epsilon \\ \frac{1}{2\epsilon} + \frac{1}{2\epsilon} \cos\left(\frac{\pi \emptyset}{\epsilon}\right) & -\epsilon \le \emptyset \le \epsilon \\ 0 & \epsilon < \emptyset \end{cases} \tag{11}$$

where ϵ is determined as above, and we can replace equation (4) with

$$\boldsymbol{f} = \sigma k \delta_{\epsilon} \boldsymbol{n} \tag{12}$$

To complete the general formulation given by equations (1), (2) and (3), we must now include the condition of keeping the level set function a signed distance function. We do this by reinitializing the level set function with the following equation introduced by Sussman et. al. [9]

$$\frac{\partial \varphi}{\partial \tau} = S(\varphi_0)(1 - |\nabla \varphi|) \tag{13}$$

where φ_0 is the solution of equation (3) for a given time, $\varphi(\tau)$ is the reinitialized level set function at artificial time τ and $S(\varphi)$ is a sign function, given by

$$\mathrm{S}(\varphi) = \begin{Bmatrix} -1 & \text{for } \varphi < 0 \\ 0 & \text{for } \varphi = 0 \\ 1 & \text{for } \varphi > 0 \end{Bmatrix} \tag{14}$$

Numerical tests, indicate that better results are obtained when $\boldsymbol{S}(\emptyset_0)$ is numerically smeared out, so a better choice is determined by [12] as

$$\mathrm{S}(\varphi) = \frac{\varphi}{\sqrt{\varphi^2 + |\nabla \varphi|^2 (\Delta x)^2}} \tag{15}$$

which has shown to yield improved and stable results, especially when the initial $\emptyset_0$ is a poor estimate of signed distance.

NUMERICAL IMPLEMENTATION AND SUMMARY OF THE ALGORITHM

The numerical procedure implemented to solve the conservation equations (1) and (2) is based on a nonlinear Newton solver. This yields a highly accurate solution but computational expensive procedure as well, since we are solving a coupled system, a fair price to pay. The level set equation (3) is solved based on a streamline diffusion method, using Crank-Nicholson time discretization scheme, and the reinitialization function is solved also based on the streamline diffusion method, but using an implicit scheme to achieve faster convergence.

Assuming that the initial level set function $\emptyset_0$, the initial pressure and the initial velocity field is known, the algorithm can be summarized as follows,

1) Calculate density and viscosity fields from equations (7) and (8).
2) Solve discrete Navier-Stokes equations (1) and (2) through a Newton nonlinear iterative procedure to obtain the new velocity field **u**. Solve the symmetric linear systems with an ILU preconditioned GMRES method.
3) Solve equation (3) to advect the level set function $\emptyset$ by the velocity field **u**.
4) Reinitialize the level set function by solving equation (13) with an iterative procedure to restore the distance function property in the region close to the zero level set.
5) Save results, and go back to 1)

An important observation of this method lies within the difficulty of accurately solving equation (13) on step 4). Because of the fact of being purely advective, we must add to the equation small amounts of artificial diffusion, which leads to significant mass loss. We'll see that it is possible to correct this error on each artificial time step, with a method presented by [13].

ADVECTION OF THE LEVEL SET FUNCTION

The level set function is initialized as a signed distance function, carrying information about the closest distance to any interface separating the two fluids. To determine the evolution of the interfaces, the level set function is advected by the domain of the velocity field. The advection equation is given by (3), which is equivalent to a linear convection-diffusion problem, where the diffusion term is equal to zero.

The strong form of the problem is given as follows: Given a vector field $\boldsymbol{b}$ defined on a region $\Omega \subset R^d$, find $\emptyset(x,t)$ such that:

$$\begin{aligned} \emptyset_{\mathrm{t}} + k\Delta\emptyset + \mathbf{b}\cdot\nabla\emptyset &= f \quad \text{on } \Omega \times \mathrm{I} \\ \emptyset(\mathrm{x},0) &= \emptyset_0 \quad \text{on } \Omega \end{aligned} \tag{16}$$

with $f = 0$, $\boldsymbol{b} = \boldsymbol{u}$, and $k = 0$. It is obvious that we can proceed to the discretization, setting f and k equal to zero, but for practical reasons, we'll keep f a time function and k a generic constant. For this problem, we obtain the semi-discrete analog by multiplying (16) by $w = v + \tau\,\boldsymbol{u}\cdot\nabla v,\ v \in V = H_0^1(\Omega)$ integrating over Ω and using the usual way Green's formula. This method is known as the streamline diffusion method, the τ coefficient is the stabilization parameter, depends on the velocity field and its calculated for each element (see [15] for further details). Then, we obtain the following variational formulation: Find $\emptyset(t) \in V,\ t \in I$ such that:

$$\begin{aligned} \left(\dot{\emptyset}(\mathrm{t}),\mathrm{w}\right) + \mathrm{a}(\emptyset(\mathrm{t}),\mathrm{w}) &= (f(t),w) \quad \forall \mathrm{v} \in \mathrm{V} \\ \emptyset(\mathrm{x},0) &= \emptyset_0 \quad \text{on } \Omega \end{aligned} \tag{17}$$

with the bilinear operators defined as

$$\begin{aligned} \mathrm{a}(\emptyset,\mathrm{w}) &= \int_\Omega [k\nabla\emptyset\cdot\nabla w + (\mathbf{b}\cdot\nabla\emptyset)\mathrm{w}]\mathrm{d}\Omega \\ &= \int_\Omega [k\nabla\emptyset\cdot\nabla v - k\tau\,\Delta\emptyset(\boldsymbol{b}\cdot\nabla\emptyset) + (\mathbf{b}\cdot\nabla\emptyset)w]d\Omega \\ (\emptyset,w) &= \int_\Omega \emptyset w d\Omega \end{aligned}$$

where the term $-k\tau\,\Delta\emptyset(\boldsymbol{b}\cdot\nabla\emptyset)$ has been integrated by parts. Now, let V_h be a finite-dimensional subspace of V with basis $\{N_1,\dots,N_M\}$. Replacing V by the finite-dimensional subspace V_h we get the following semi-discrete analog: Find $\emptyset_h(t) \in V_h,\ t \in I$ such that

$$\begin{aligned} \left(\dot{\emptyset}_{\mathrm{h}}(t),w_h\right) + a(\emptyset_{\mathrm{h}}(t),w_h) &= (f(t),w) \quad \forall v_h \in V_h \\ \emptyset(x,0) &= \emptyset_0 \quad on\ \Omega \end{aligned}$$

Applying the Crank-Nicholson time-discretization method we obtain the following problem: Find $\emptyset_h^n \in V_h,\ n = 0,\dots,N$ such that,

$$\begin{aligned} \left(\frac{\emptyset_h^n - \emptyset_h^{n-1}}{\Delta t},w\right) + a\left(\frac{\emptyset_h^n + \emptyset_h^{n-1}}{2},w\right) &= \left(\frac{f^n + f^{n-1}}{2},w\right) \quad \forall v \in V_h \\ (\emptyset_h(0),w) &= (\emptyset^0,w) \quad \forall v \in V_h \end{aligned}$$

With $f = 0$, and with the velocity $\boldsymbol{b}$ set equal to the velocity $\boldsymbol{u}$ given by the navier stokes equation, this is the streamline petrov-galerkin finite element discretization of the level set equation (3). Ideally, we should be able to solve this system setting the diffusion term $k = 0$, but for practical implementations this is not the case. Thus, since we are appliying streamline diffusion we are enable to use a very low diffusion coefficient k.

REINITIALIZATION OF THE LEVEL SET FUNCTION

It has been shown [9] that it is critical that the level set equation remain a distance function in regions close to interface, in order to obtain acceptable accuracy in the computation of the unit normal and the mean curvature. After advection, the level set function does not necessarily correspond to a distance function any more. Keeping $\emptyset$ a distance function is achieved by reinitialization of the level set function, given by equation (13). By solving this equation, we obtain a distance function with the same zero level set of φ_0. Equation (13) can also be written [8],

$$\frac{\partial\varphi}{\partial\tau} + \mathrm{w}\cdot\nabla\varphi = \mathrm{S}(\varphi), \qquad \mathrm{w} = \mathrm{S}(\varphi_0)\frac{\nabla\varphi}{|\nabla\varphi|} \tag{18}$$

that we can see it is a purely convective system, with the velocity field defined by $\boldsymbol{w}$ and the force function f defined by $f = \mathrm{S}(\varphi)$. Therefore, we can apply the same procedure we used for the advection of the level set equation, just that for this case, we'll use and implicit time discretization scheme, because we're looking for fast, robust convergence instead of accuracy.

Same as with the advection of the level set equation, we are enforce to include some artificial diffusion to ensure stability, but since we're solving many artificial time steps, the level set function diffuses more and more on each iteration resulting in significant mass loss.

A method presented by [13] suggests an improvement to the standard reinitialization procedure. The method states, that if the interface does not move during reinitialization, the area is preserved. On the other hand, one can preserve the area while allowing the interface to move, implying that the proposed constraint is weaker than it should be. The local constraint is implemented by the addition of a correction term to the right hand side of equation (13),

$$\frac{\partial\varphi}{\partial\tau} = \mathrm{S}(\varphi_0)(1 - |\nabla\varphi|) + \lambda\delta_\epsilon(\varphi)|\nabla\varphi| \tag{19}$$

We can rewrite this as follows,

$$\frac{\partial\varphi}{\partial\tau} = \frac{\partial\tilde{\varphi}}{\partial\tau} + \lambda\delta_\epsilon(\varphi)|\nabla\varphi| \quad \text{with,} \tag{20}$$

$$\frac{\partial\tilde{\varphi}}{\partial\tau} = \mathrm{S}(\varphi_0)(1 - |\nabla\tilde{\varphi}|) \tag{21}$$

The mentioned constrain is defined by

$$\int_{\Omega_{\mathrm{i,j}}} \mathrm{H}_\epsilon(\varphi)\mathrm{d}\Omega = 0 \tag{22}$$

where $\Omega_{\mathrm{i,j}}$ is an individual cell and $H_\epsilon(\emptyset)$ is the smeared-out heaviside function defined by equation (10). This is equivalent to

$$\int_{\Omega_{\mathrm{i,j}}} \dot{\mathrm{H}}_\epsilon(\varphi)\ \varphi_{\mathrm{t}}\,\mathrm{d}\Omega = \int_{\Omega_{\mathrm{i,j}}} \delta_\epsilon(\varphi)(\tilde{\varphi}_{\mathrm{t}} + \lambda\delta_\epsilon(\varphi)|\nabla\varphi|)\mathrm{d}\Omega = 0$$

This way, a particular $\lambda_{i,j}$ is determined on each cell by the following equation,

$$\lambda_{i,j} = -\frac{\int_{\Omega_{i,j}} \delta_\epsilon(\varphi)\left(\frac{\tilde{\varphi}^{n+1}-\tilde{\varphi}^{n}}{\Delta t}\right)d\Omega}{\int_{\Omega_{i,j}} {\delta_\epsilon}^2(\varphi)|\nabla\varphi|\,d\Omega} \quad (22)$$

where equation (20) is used to compute $\tilde{\varphi}^{n+1}$ from $\tilde{\varphi}^{n}$.

In summary, the initial guess $\tilde{\varphi}^{n+1}$ obtained from equation (20) is replaced with a corrected $\tilde{\varphi}^{n+1} + \Delta t\,\lambda\delta_\epsilon(\varphi)|\nabla\varphi|$, where λ is calculated on each cell using equation (22).

THE NAVIER STOKES EQUATIONS

Proceeding to the weak formulation of the conservation equations, we introduce the spaces

$$V = \{v \in H^1(\Omega)^2 : v = \{v_i\}, v_i \in H^1(\Omega), \forall i\}$$
$$P = \{q \in L^2(\Omega) : \int_\Omega q\,d\Omega = 0\}$$
$$H^1(\Omega) = \{v \in L^2(\Omega) : \frac{\partial v}{\partial x_i} \in L^2(\Omega), \forall i\}$$
$$V_{u\Gamma} = \{\boldsymbol{v} \in V : \boldsymbol{v} = \boldsymbol{u}_\Gamma \; on \; \partial\Omega\}$$
$$V_0 = \{\boldsymbol{v} \in V : \boldsymbol{v} = \boldsymbol{0} \; on \; \partial\Omega\}$$

Furthermore, we obtain the following variational formulation of (1)-(2): Find $\boldsymbol{u}(x,t) \in V_{u\Gamma}$ and $p(x,t) \in P$ such that $\forall t \in [0,T]$,

$$m\left(\rho, \frac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v}\right) + a(\mu, \boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) + c(\rho, \boldsymbol{u}, \boldsymbol{u}, \boldsymbol{w}) = f_\Gamma(\boldsymbol{v}) + d(\rho, \boldsymbol{g}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in V_0 \quad (23)$$

$$b(\boldsymbol{u}, q) = 0 \quad \forall q \in P \quad (24)$$

where these forms are defined by

$$m\left(\rho, \frac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v}\right) = \int_\Omega \rho\left(\frac{\partial \boldsymbol{u}}{\partial t} \cdot \boldsymbol{v}\right) d\Omega$$

$$a(\mu, \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \frac{\mu}{Re}\{tr(\nabla \boldsymbol{u}^T : \nabla \boldsymbol{v}) + tr(\nabla \boldsymbol{u} : \nabla \boldsymbol{v})\}\, d\Omega$$

$$b(\boldsymbol{u}, p) = -\int_\Omega (\nabla \cdot \boldsymbol{u}) p\, d\Omega$$

$$c(\rho, \boldsymbol{u}, \boldsymbol{u}, \boldsymbol{w}) = \int_\Omega \rho(\boldsymbol{u} \cdot \nabla \boldsymbol{u}) \cdot \boldsymbol{w}\, d\Omega$$

$$d(\rho, \boldsymbol{g}, \boldsymbol{v}) = \int_\Omega \frac{\rho}{Fr^2}(\boldsymbol{g} \cdot \boldsymbol{v})\, d\Omega$$

$$f_\Gamma(\boldsymbol{v}) = \int_\Omega \frac{1}{We} k\delta_\Gamma \boldsymbol{n}\, v d\Omega$$

The discretization of the time derivative is given by the θ-scheme, where $\theta = 1$ gives an implicit scheme, $\theta = 0.5$ a Crank-Nicholson scheme and $\theta = 0$ a fully explicit scheme. Then, equations (23) and (24) become

$$m(\rho, 0.5(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n), \boldsymbol{v}) + a(\mu, (\theta \boldsymbol{u}^{n+1} + (1-\theta)\boldsymbol{u}^{n+1}), \boldsymbol{v}) + b(\boldsymbol{v}, p) + c(\rho, (\theta \boldsymbol{u}^{n+1} + (1-\theta)\boldsymbol{u}^{n+1}), \boldsymbol{u}, \boldsymbol{w}) = f_\Gamma(\boldsymbol{v}) + d(\rho, \boldsymbol{g}, \boldsymbol{v})$$

$$b\left((\theta \boldsymbol{u}^{n+1} + (1-\theta)\boldsymbol{u}^{n+1}), q\right) = 0$$

For our problem, we set $\theta = 1$. After time discretization, equations (23) and (24) can be written in residual form for the unknown nodal values $\boldsymbol{U}_{n+1}$ as the nonlinear algebraic system

$$R(\boldsymbol{U}_{n+1}) = 0 \quad (25)$$

with $\boldsymbol{U}_{n+1} = (\boldsymbol{u}, p)^T$

We can see now, that for any $\theta \neq 0$ we need to solve a nonlinear system. To do so, we use a nonlinear Newton solver. As we are solving the pressure-velocity coupled system, we are enforced to select LBB stable elements. For our problem, we choose quadrilateral elements with 8 degrees of freedom, with second order accuracy for the velocity and first order accuracy for the pressure.

The goal is to define a sequence of linear problems that, when solved, converge to obtain the solution $\boldsymbol{U}_{n+1}$ of the nonlinear system (25). We can achieve this goal with the Newton-Raphson method defined as

$$\left[\frac{\partial \boldsymbol{R}(\boldsymbol{U}^l_{n+1})}{\partial \boldsymbol{U}_{n+1}}\right] \delta \boldsymbol{U}^{l+1}_{n+1} = -\boldsymbol{R}(\boldsymbol{U}^l_{n+1}) \quad (26)$$

that results in an implicit linear system for $\delta \boldsymbol{U}^{l+1}_{n+1}$ and a sequence of iterates $l = (0, 1, \ldots)$ which converges to $\boldsymbol{U}_{n+1}$[14]. It's worthwhile to recall that Newton's method exhibits second-order conditional convergence, meaning that the magnitude of the residual decreases quadratically at successive iterates provided that the initial guess is sufficiently close to the unknown. For this reason, our algorithm adopts linear extrapolation, using previous solutions to estimate the initial Newton's iterate. This has shown marked improvement on the convergence rate, but even with this implementation, sometimes the initial guess is far from the exact solution and the algorithm diverges. Our algorithm deals with this problem by reducing the time step, therefore looking for the unknown closer to the previous solution.

Solving a large system of equations such as the one presented by equation (25) can be a hard task to the linear solver, as it can lead to a bad-conditioned system of equations. An approximate block ILU factorization is employed.

PROGRAM STRUCTURE

The numerical finite element simulation code was written on C++ language using the *libmesh* library as main tool. The *libmesh* library is a parallel adaptative finite element library for simulating partial differential equations. Among the many advantages of the library are: the possibility of selecting different generic 1D, 2D and 3D finite element types and interpolation functions; runtime selection of different quadrature rules; mesh creation, modification, input, output and format translation utilities; all with support for adaptative mesh refinement (AMR) computations in parallel platforms. With this tool, the program implementation became much easier and was performed in short period of time.

The mesh implementation that was used to solve this problem is basically a structured mesh (can be unstructured as well if needed) that is refined and coarsen at each time step, for higher reliability and efficiency.

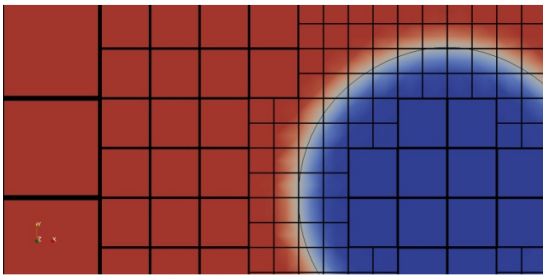

Figure1. Refined mesh.

The derivative of the Heaviside function is used as an estimator to flag elements for refinement and coarsening. For our problem, we use second order QUAD8 elements, tenth order Gaussian quadrature rule for the reinitialization function and its corrector, and fifth order quadrature rule for the level set and Navier-Stokes equations.

NUMERICAL RESULTS

In this section, we present the results of the computation of two numerical experiments. The nondimensional parameters that were used to characterize each problem are: The Morton number *Mo*, the Eötvös number *Eo* and the Froude number Fr, in addition to viscosity and density rations μ_1/μ_2 and ρ_1/ρ_2. Here, subscript 1 denotes the viscosity and density of the external fluid that surrounds the bubble and subscript 2 the fluid inside the bubble.

The Froude number can be calculated as follows,

$$Fr = \frac{U}{\sqrt{gL}} \tag{27}$$

where U is the characteristic velocity and L is the characteristic length. It is related to the Reynolds and Weber number through the following equations,

$$We = Eo\, Fr^2 \tag{28}$$

$$Re^4 = \frac{We^3}{Fr^2 Mo} \tag{29}$$

where We and Re are the adimentional Weber and Reynolds numbers.

Following, we present the numerical results of a rising bubble simulation on figures 2 to 5 and the results of bubbles coalescence simulation on figures 6 to 13.

Rising Bubble

The first experiment consists on an initially stationary bubble that rises due the effect of gravitational forces. The parameters that were used for these simulations are $Fr = 10.01$, $Eo = 1000$ and $Mo = 0.01$. The time step used was variable.

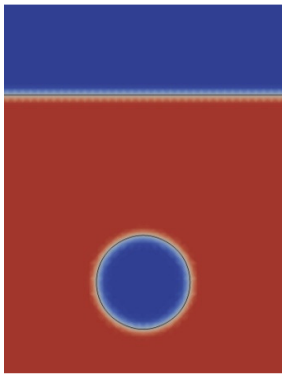

Figure 2. (t=0.0)

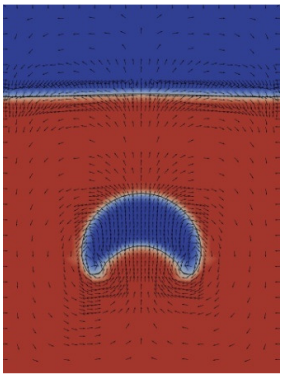

Figure 3. (t=15.0)

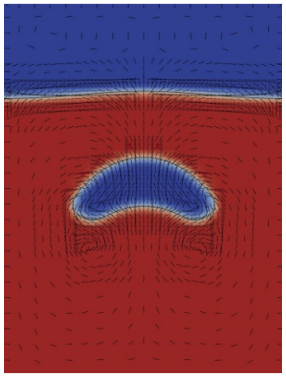

Figure 4. (t=25.1)

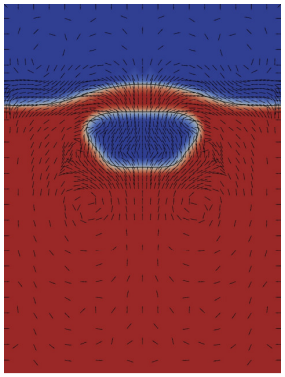

Figure 5. (t=40.2)

Bubble Coalescence

The second experiment focuses on bubble coalescence. The parameters that were used for these simulations are $Fr = 0.319$, $Eo = 10$ and $Mo = 0.1$.

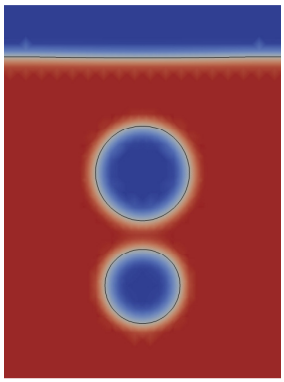

Figure 6. (t=0)

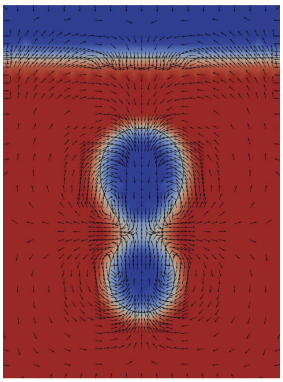

Figure 7. (t=0.077)

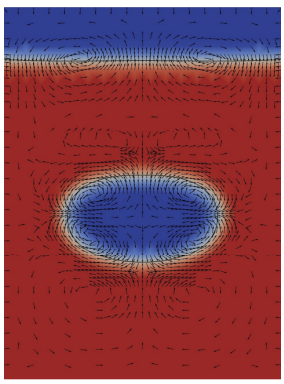

Figure 8. (t=0.179)

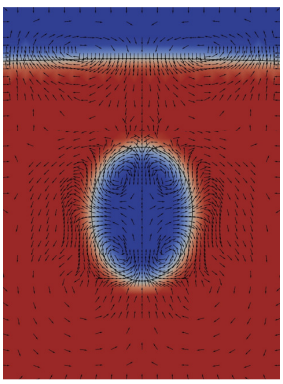

Figure 9. (t=0.262)

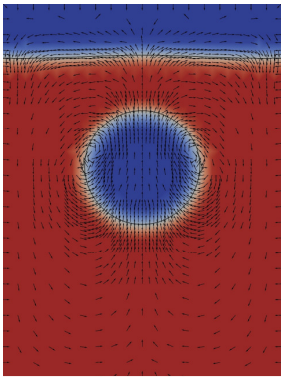

Figure 10. (t=0.707)

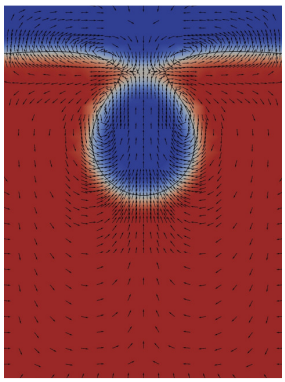

Figure 11. (t=0.937)

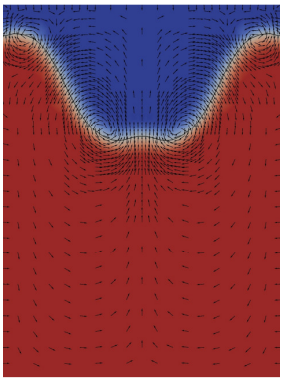

Figure 12. (t=1.030)

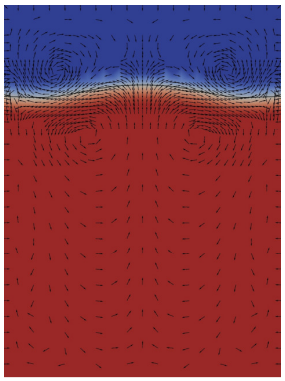

Figure 13. (t=1.150)

CONCLUSIONS

The method presented in this paper is an adaptative finite element level set approach that deals with the mass conservation problems that are inherent to the level set method, through mesh refinement capabilities and accurate correction techniques for keeping the level set function a sign function. This method results on effectively conserving mass properties while easily dealing with topological changes of the fronts, such as bubble coalescence and changes on free surface. The code implements a moving finite element unstructured mesh as well as a control procedure using *libmesh*, an adaptative C++ finite element library for simulating partial differential equations.

Future work

We can see from the last simulation that the bubbles start the merging process when they are close enough so that the smeared out surface tension force of one bubble cancels with the one of the other, reducing the surface tension in that region. In reality, this should happen when the region that defines one bubble intersects the region defined by the other bubble. From equation (12) we can see that the surface tension is defined by the smeared out derivative of the surface tension, thus accuracy can be increased by reducing ϵ. However, by doing this we are introducing numerical instabilities, and we are back to the original problem. Now, with the difficulties of the method identified, we can think on new methods to achieve even more accurate results, such as the ghost fluid method shown in [11] to directly define the pressure jump across the interface or an SUPG approach for the Navier-Stokes equations.

REFERENCES

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

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

[3] G. Rabello, L. Portella, F. Sousa, N. Mangiavacchi. An ALE finite element method for the simulation of 3D multiphase flows. *12th Brazilian Congress of Thermal Engineering and Sciences* 2008. Belo Horizonte, MG.

[4] 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.

[5] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100, pp. 335–354, 1992.

[6] F. Sousa, N Mangiavacchi. A Lagrangian level-set approach for the simulation of incompressible two-fluid flows. *International Journal for Numerical Methods in Fluids* 2005; **47**:1393-1401.

[7] S. Groß, V. Reichelt, A. Reusken. A finite element based level set method for two-phase incompressible flows. *Computing and Visualization in Science* 2006; **9/4**:239-257.

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

[9] M. Sussman, M. Fatemi, P. Smereka, S. Osher. A level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics* 1994. **114**:146-159.

[10] B. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey, libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, vol. 22, no. 3--4, pp. 237--254, 2006.

[11] S. Osher, R. P. Fedkiw. Level set methods and dynamic implicit surfaces. *Springer Verlag.* 2003.

[12] Peng, D., Merriman, B., Osher, S., Zhao, H.-K., and Kang, M., A PDE-Based Fast Local Level Set Method, *J. Comput. Phys.* 155, 410.438 (1999).

[13] Sussman, M. and Fatemi, E., An Efficient Interface-Preserving Level Set Redistancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow, *SIAM J. Sci. Comput*. 20, 1165.1191 (1999).

[14] Kirk, B.S. Adaptive finite element simulation of flow and transport applications on parallel computers. *D., The University of Texas at Austin*. 2007.

[15] A.N. Brooks and T.J.R. Hughes. Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 32(1-3):199–259, 1982.

[16] J.C. Heinrich and D.W. Pepper. *Intermediate finite element method: fluid flow and heat transfer applications*. Taylor & Francis Group, 1999.