

## A FULLY ADAPTIVE, CONSERVATIVE FRONT TRACKING METHOD FOR THE SIMULATION OF INCOMPRESSIBLE MULTIPHASE FLOWS

M. R. Pivello<sup>1</sup>, R. S. de Lima<sup>1</sup>, M. M. V. Vale<sup>1</sup>, A. M. Roma<sup>2</sup> and A. Silveira-Neto<sup>1</sup>

<sup>1</sup> Department of Mechanical Engineering, Federal University of Uberlândia  
(pivello@mecanica.ufu.br)

<sup>2</sup> Institute of Mathematics and Statistics at the University of São Paulo

**Abstract.** *This paper presents a fully adaptive formulation of the Front Tracking method for the simulation of incompressible, multiphase, bubbly flows, based on the Tryggvason formulation. The Navier-Stokes equations are discretized using a finite difference scheme, and domain discretization is carried out with Berger & Collela's structured adaptive mesh refinement (SAMR) algorithm. Time discretization is based on SBDF scheme, with adaptive time stepping. The lagrangian interface is represented using the GTS library, which provides a volume- and shape- preserving remeshing algorithm, therefore minimizing the volume change due to non-conservative interpolation of the eulerian velocity field. Nevertheless, a simple volume recovery algorithm is also provided, along with a subgrid undulation removal algorithm based on the TSUR-3D algorithm [6]. Rising bubble flows were simulated under several regimes, showing small errors when comparing to experimental results.*

**Keywords:** *two-phase flows, front tracking method, rising bubbles, adaptive mesh refinement, conservative).*

## 1. INTRODUCTION

Front-tracking methods are based on the one-fluid formulation for the simulation of multiphase flows and make use of markers, usually a set of connected points forming an unstructured mesh, to track the interface. Since the interface is explicitly represented by its coordinates, the set of markers is said to use a Lagrangian framework, and the mesh which characterizes the interface is termed Lagrangian mesh. While this interface is based on a Lagrangian framework, an Eulerian framework is used to solve the transport equations. This method is accurate but also complex to be implemented because dynamic remeshing of the Lagrangian interface mesh is required and mapping of the Lagrangian data onto the Eulerian mesh has to be carried out. Difficulties may arise when multiple interfaces interact with each other as in coalescence and breakup, both of which requiring a proper sub-grid model. This feature may be computationally expensive, but can be used to prevent artificial merge or breakup when the number of bubbles in the flow is large.

Although storing the marker points is enough to fully determine the interface position, storing the interface as a mesh is useful for computing geometrical properties such normal, area and the interfacial tension force, not to mention checking against collisions or merge/breakup phenomena.

There are two possible strategic forms of communication. Glim et al. [8] discretized the Navier-Stokes equations using a finite difference method which modifies the stencil in the vicinity of the mesh overlap, so that the vertices of the two meshes match each other. However, for multiphase flows the most widespread methodology is based on the Immersed Boundary Method (IBM) of Peskin [13, 14], which represents the interface by imposing a force field that is spread on the Eulerian mesh. The Navier-Stokes equations are solved based on one-fluid models. Among various methodologies developed after Peskin, one of the most wide spread in the context of multiphase flows is the work of Uverdi and Tryggvason [20] and Tryggvason et al. [19]. Similarly to IBM, the interface between the phases is represented by the interfacial tension force field.

In the present work the Front-Tracking method is combined with an AMR (Adaptive Mesh Refinement) framework. The concept of adaptive mesh refinement is based on the works of Berger et al [2] and Jameson [9] and consist in creating a hierarchy of Cartesian meshes with different levels of refinement which cover the entire domain, concentrating the meshes on the regions which require special attention. All meshes in a given level of refinement must have the same characteristic length.

## 2. MATHEMATICAL MODEL

The mathematical model used in the present work is composed of the Navier-Stokes and mass equations combined with a front tracking model with which problems of two phase flows are modeled. The model is presented in three parts, as follows.

### 2.1. Mathematical model for the fluid

This model, for a Newtonian fluid and for incompressible flows, is composed by the classical mass and Navier-Stokes Equations.

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

$$\rho \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \rho \mathbf{g} + \nabla \cdot (\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \mathbf{f} \quad (2)$$

where  $\mathbf{u}(\mathbf{x}, t)$  is the uid velocity,  $\rho$  is the uid density, and  $\mu$  is the viscosity,  $p$  is the pressure and  $\mathbf{g}$  is the gravity acceleration. The source term models the interfacial force that is different from zero only over an interface and equals zero everywhere. The fluid density and viscosity are calculated with the aid of an indicator function:

$$\rho(\mathbf{x}, t) = \rho_1 H(\mathbf{x}, t) + \rho_2 (1 - H(\mathbf{x}, t)) \quad (3)$$

$$\mu(\mathbf{x}, t) = \mu_1 H(\mathbf{x}, t) + \mu_2 (1 - H(\mathbf{x}, t)) \quad (4)$$

Usually the indicator function is found by solving a Poisson equation for the surface normal vector [20]. In this work, however, a geometric approach, named CPT (Closest Point Transform) is adopted [12]. Instead of solving an additional elliptic PDE, CPT finds the shortest distance between a given point in the Eulerian domain and the Lagrangian interface. Unlike the Tryggvason approach, in this case the function is calculated only in the neighbourhood of the interface [5].

## 2.2. Mathematical model for the interface

The interface is modeled by an equation for the interface force field and the equations for its Lagrangian transport:

$$\mathbf{F}(\mathbf{X}, t) = \int_{d\Omega} \sigma \kappa \mathbf{n}(\mathbf{X}, t) d\Omega = \oint_{d\Gamma} \sigma \mathbf{t} \times \mathbf{n} d\Gamma \quad (5)$$

$$\frac{d\mathbf{X}(t)}{dt} = \mathbf{u}(\mathbf{X}, t) \quad (6)$$

where  $\sigma$  is the interface force coefficient,  $\kappa$  is the curvature,  $\mathbf{n}$  is the interface normal,  $\mathbf{X}(t)$  is the spatial Lagrangian coordinate,  $d\Omega$  is an area element and  $d\Gamma$  is the boundary of  $d\Omega$ .

## 2.3. Mathematical model for the fluid-interface communication

Information exchange between fluid and interface is given by interpolation and distribution equations

$$\mathbf{f}(\mathbf{x}, t) = \int \mathbf{F}(\mathbf{X}, t) \delta(\mathbf{X} - \mathbf{x}) d\mathbf{X} \quad (7)$$

$$\mathbf{U}(\mathbf{X}, t) = \int \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}) d\mathbf{x} \quad (8)$$

where  $\mathbf{X}$  and  $\mathbf{x}$  are, respectively, positions in the Lagrangian (interface) and the Eulerian (fluid) domains.

## 2.4. Numerical method

The time discretization of the differential model is performed by the general model below:

$$\frac{\rho^{n+1}}{\Delta t} (\alpha_2 \mathbf{u}^{n+1} + \alpha_1 \mathbf{u}^n + \alpha_0 \mathbf{u}^{n-1}) = \beta_1 h(\mathbf{u}^n) + \beta_0 h(\mathbf{u}^{n-1}) + \lambda (\theta_2 \nabla^2 \mathbf{u}^{n+1} + \theta_1 \nabla^2 \mathbf{u}^n + \theta_0 \nabla^2 \mathbf{u}^{n-1}) - \nabla p + \rho^{n+1} \mathbf{g} \quad (9)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (10)$$

where  $\lambda = L_\infty(\mu)$  and  $h$  depends on the diffusive-, advective- and forcing terms, according to the following equation:

$$h(\mathbf{u}) = -\lambda \nabla^2 \mathbf{u} + \nabla \cdot (\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)) - (\mathbf{u} \cdot \nabla) \mathbf{u} + \mathbf{f} \quad (11)$$

The value of  $\alpha$ ,  $\beta$  and  $\theta$  are given by:

$$\begin{aligned}\alpha_0 &= \frac{(2\gamma - 1)\omega^2}{1 + \omega} \\ \alpha_1 &= (1 - 2\gamma)\omega - 1\end{aligned}\tag{12}$$

$$\alpha_2 = \frac{1 + 2\gamma\omega}{1 + \omega}$$

$$\beta_0 = 1 + \gamma\tag{13}$$

$$\beta_1 = -\gamma\omega$$

$$\begin{aligned}\theta_0 &= c \left( \omega - \frac{\omega}{1 + \omega} \right) \\ \theta_1 &= 1 - \gamma - c\omega\end{aligned}\tag{14}$$

$$\theta_2 = \gamma + c \frac{\omega}{1 + \omega}$$

where  $\omega = \frac{\Delta t^{n+1}}{\Delta t^n}$  is the rate between two consecutive time steps. A family of numerical schemes can be established choosing the appropriate values for  $\gamma$  and  $c$  [1]:

- Crank-Nicolson Adams-Bashforth(CNAB):  $(\gamma;c) = (0.5;0)$  ;
- Modified Crank-Nicolson Adams-Bashforth(MCNAB):  $(\gamma;c)=(0.5; 0.125)$ ;
- Crank-Nicolson Leap-Frog (CNLF):  $(\gamma;c)=(0.0;1.0)$ ;
- Semi-Backward Difference (SBDF):  $(\gamma;c)=(1.0; 0.0)$ .

In the present work the above SBDF temporal time discretization, with variable size of time step, was chosen in all cases simulated. To handle the pressure-velocity coupling between Eq. 1 and Eq. 2, the following fractional step method is proposed:

$$\begin{aligned}\frac{\rho^{n+1}}{\Delta t} (\alpha_2 \mathbf{u}^{*n+1} + \alpha_1 \mathbf{u}^n + \alpha_0 \mathbf{u}^{n-1}) &= \beta_1 h(\mathbf{u}^n) + \beta_0 h(\mathbf{u}^{n-1}) + \\ \lambda (\theta_2 \nabla^2 \mathbf{u}^{*n+1} + \theta_1 \nabla^2 \mathbf{u}^n + \theta_0 \nabla^2 \mathbf{u}^{n-1}) &- \nabla p + \rho^{n+1} \mathbf{g}\end{aligned}\tag{15}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{*n+1} + \frac{\Delta t}{\alpha_2} \frac{\nabla q}{\rho^{n+1}}\tag{16}$$

The pressure correction  $q$  is obtained using the following equation:

$$\nabla \cdot \left( \frac{1}{\rho^{n+1}} \nabla q \right) = \frac{\alpha_2}{\Delta t} \nabla \cdot \mathbf{u}^*\tag{17}$$

## 2.5. Spatial discretization

Space is discretized using a structured adaptive mesh refinement, which is based on the Immersed Boundary method introduced by Roma et al. [16], and in the hierarchical grid structure proposed by Berger and Colella [3]. In this scheme, regions of the flow being of special interest are covered by block-structured grids, defined as a hierarchical sequence of nested, progressively finer levels (composite grids). Each level is formed by a set of disjoint rectangular grids and the ratio between two successive refinement levels is constant and equal to two. Ghost cells are employed around each grid, for all the levels, and underneath finer grid patches to formally prevent the finite difference operators from being redefined at grid borders and at interior regions which are covered by finer levels.

Values defined in these cells are obtained from interpolation schemes, usually with second or third order accuracy, and not from solving the equations of the problem. A staggered composite grid is used, i.e., pressure and others scalar variables are placed at the centers of the computational cells, while vector variables are located at the respective cell faces. The discretization of the Laplacian, gradient and divergence differential operators are performed by standard, cell-centered second order finite difference stencils. Although a variable can be defined or initialized in any level, in the current work the interface must be completely covered by the finest level, ensuring that the most important physical phenomena are being captured.

The initial mesh for the Lagrangian interface is generated using GMSH [7]. Volume conservation is ensured in the surface remeshing during the simulations by applying the memoryless simplification technique by Lindstrom and Turk [11], as provided by the GTS Library [15]. Nevertheless, the interpolation of the velocity field from the eulerian framework onto the Lagrangian interface usually is non-conservative. Although the volume change in a single time step is nearly negligible, it is a cumulative process that may lead to considerable volume change in long term simulations. To overcome this issue, a simple volume recovery procedure applied at the end of the advection step will be described next.

## 2.6. Volume Recovery Algorithm

The mass conservation issue in Front Tracking arises after interpolating the Eulerian velocity field to the Lagrangian interface, because the interpolation functions are not conservative. This problem can be tackled using two different approaches: (i) implementing a conservative interpolation scheme or (ii) applying a correction step after interpolating the velocity field in order to correct the surface volume. This second approach will be followed here, since it is much simpler than (i).

Average based volume corrections are not new in Front Tracking Methods [17, 18] and usually are carried out apart from the remeshing procedure, such that no topological change is introduced in the mesh. If the volume change is small, it is safe to assume that all variation occurs in the normal direction. Since the interface is discretized by triangular elements, the volume change can be seen as the sum of small prisms and formed by the mesh elements and a small constant height  $\eta$ :

$$\Delta V = \sum_i^{Ne} A_i \eta \quad (18)$$

where  $A_i$  is the area of the  $i$ th-element of the mesh,  $\eta$  is the average length of the normal and  $Ne$  is the total number of elements in the mesh. Since  $h$  is constant, eq. 18 can be written as

$$\Delta V = \eta \sum_i^{Ne} A_i = A \cdot \eta \quad (19)$$

where  $A$  is total surface area. Also, the volume change may also be computed as the difference between the initial and current interface volume, i.e.:  $\Delta V = V_c - V_i$ ,

$$A \cdot \eta = V_c - V_i \quad (20)$$

and the length of the normal will be just

$$\eta = \frac{V_c - V_i}{A} \quad (21)$$

That is, the volume correction can be achieved by just expanding or shrinking the surface in the normal direction by a factor equal to  $\eta$ . The correction is then given by eq. 22, where  $\mathbf{x}_{old}$  and  $\mathbf{x}_{new}$  are the original and new coordinates of the mesh vertices.

$$\mathbf{x}_{new} = \mathbf{x}_{old} - \eta \cdot \mathbf{n}_v \quad (22)$$

Since this movement will be applied to the vertices, an accurate pseudo-normal computation must be provided. A good approximation is given by an weighted average of the normals of the elements sharing a given vertex [10] according to Eq. 23:

$$\mathbf{n}_v = \sum_{i=1}^{N_t} \frac{\mathbf{n}_i \sin(\alpha_i)}{\|e_{i1}\| \|e_{i2}\|} \quad (23)$$

Here,  $N_t$  is the number of triangles which share the vertex,  $\mathbf{n}_i$  is the normal of element  $i$ ,  $e_{i1}$  and  $e_{i2}$  are the edges of the  $i$ th-element which are employed to compute this element's normal, and  $\alpha_i$  is the angle between them.

### 3. RESULTS

Firstly a verification of the volume-conserving abilities of the front tracking algorithm will be made, based on an analytical shear flow test. Then, results showing the simulation of rising bubbles will be presented.

#### 3.1. Analytical Shear Flow

The test consists in subjecting a surface, usually spherical, to a solenoidal, periodic, analytic velocity field and assessing its volume change as the flow evolves. Assuming a velocity field with a period  $T$ , the surface is stretched until  $t = T/2$  and then returns to its

original position and shape. The velocity field may be calculated either directly on the Lagrangian vertices or interpolated from an auxiliary grid, allowing to assess the influence of the interpolation functions on the volume change during the simulation.

Regarding the interface advection, the two most common time integration schemes are first order Euler and second order trapezoidal rule, respectively Eqs. 24a and 24b. Note that, as long as there is a time history of the Eulerian velocity field, it is possible to implement second order methods for the advection of the Lagrangian interface.

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \cdot \mathbf{u}^{n+1} \quad (24a)$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \frac{\Delta t}{2} \cdot (\mathbf{u}^{n+1} + \mathbf{u}^n) \quad (24b)$$

Surface stretching requires remeshing, which also may also interfere on the conservation process. Regardless its volume conservation properties, any remeshing algorithm destroys the Lagrangian velocity history, requiring a new interpolation of the velocity field. Since interpolation schemes are usually non-conservative, a conservative remeshing algorithm could still lead to a non-conserving outcome owing to the need for velocity interpolation.

In order to assess the influence of such variables on the flow outcome, a series of tests was performed, in which three main parameters were analysed: (i) the velocity field interpolation, (ii) the surface remeshing and (iii) the volume recovery algorithm. These cases are summarized in table 1.

Case	Description		
	Velocity Interpolation	Volume Recovery	Remeshing
1	no	no	no
2	no	no	yes
3	yes	no	no
4	yes	no	yes
5	no	yes	no
6	no	yes	yes
7	yes	yes	no
8	yes	yes	yes

Table 1. Shear Flow. Case description for each test performed.

The velocity profile is given by equation 25 [17] and the interpolation of the velocity field was performed using equation 26 [21].

$$\begin{aligned} u(x, y, z, t) &= \cos(\pi t/T) \sin^2(\pi x) (\sin(2\pi z) - \sin(2\pi y)) \\ v(x, y, z, t) &= \cos(\pi t/T) \sin^2(\pi y) (\sin(2\pi x) - \sin(2\pi z)) \\ w(x, y, z, t) &= \cos(\pi t/T) \sin^2(\pi z) (\sin(2\pi y) - \sin(2\pi x)) \end{aligned} \quad (25)$$

$$\delta(r) = \begin{cases} \frac{1}{4}(1 + \cos(\frac{\pi}{2}r)), & r < 2 \\ 0, & r \geq 2 \end{cases} \quad (26)$$

Figure 1 shows snapshots of the sphere shape during a typical simulation at  $t = 0$ ,  $t = T/4$ ,  $t = T/2$ ,  $t = 3T/4$  and  $t = T$ . The flow stretches the surface until  $T = T/2$  and then symmetrically returns it to its initial position, so that the shape and position are the same not only  $t = 0$  and  $t = T$ , but also at  $t = T/4$  and  $t = 3T/4$ . The period is  $T = 4s$ , the same which was used in the simulations presented here.

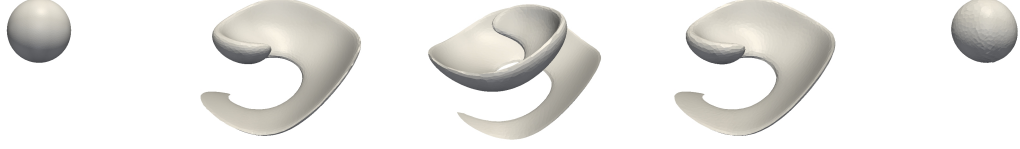


Figure 1. Geometry deformation when subjected to the analytical velocity field given by Eq. 25. Snapshots taken at  $t = 0$ ,  $t = \frac{T}{4}$ ,  $t = \frac{T}{2}$ ,  $t = \frac{3T}{4}$  e  $t = T$

Figure 2 shows the time evolution of the relative volume error ( $\varepsilon_{vol} = (V_f/V_i - 1) \times 100$ ) for all cases simulated. Graphs on the left column show the cases in which volume recovery was not applied, while the right column shows the results achieved with volume recovery. Notice that, in this case, the error was kept within the specified threshold, that is  $\pm 0.01\%$ . Regarding the non-conservative cases, both interpolation process and interface remeshing have shown to play little influence on the final volume error when the time integration is carried out with Euler scheme. Although velocity interpolation changes the time history of volume error, the final value achieved is similar in both cases. The trapezoidal rule, although being more sensitive to both surface remesh and velocity interpolation, leads to considerably smaller errors.

### 3.2. Rising Bubbles

This section presents the simulation of rising bubbles in various flow regimes. Firstly, a case study is performed in order to assess the influence of some geometry-correction tools on the outcome of the flow. Then a set of low Reynolds cases is simulated, in order to assess the influence of the Morton number on the prediction of the terminal Re number. All cases in this set have the same Eo number. Finally, a wobbling case is simulated. In all cases simulated in this paper, the initial distance from the center of the bubble to the domain boundaries is four times the bubble diameter. Also, the maximum edge length allowed for the Lagrangian mesh is half the length of the Eulerian grid spacing at its finest level.

#### Geometry-Correction Tools

The physical parameters defining the rising bubble flow regime in this section are:  $Eo = 40$ ,  $M = 0.056$ ,  $Re = 20.6$ ,  $\rho_C = 1000 kg/m^3$ ,  $\mu_C = 0.273556 Pa \cdot s$ ,  $\lambda_\rho = 100$ ,  $\lambda_\mu = 100$ . The bubble diameter is  $\phi = 0.02m$ .  $Eo$  is the Eotvos number,  $M$  is the Morton number,  $\rho_C$  and  $\mu_C$  are the density and dynamic viscosity of the continuous phase and  $\lambda_\rho$  and  $\lambda_\mu$  are the density ratio and viscosity ratio between the continuous and dispersed phase, respectively.

Besides the well-known non-conservative behaviour of the velocity interpolation process, another common problem in front tracking methods is the development of small un-

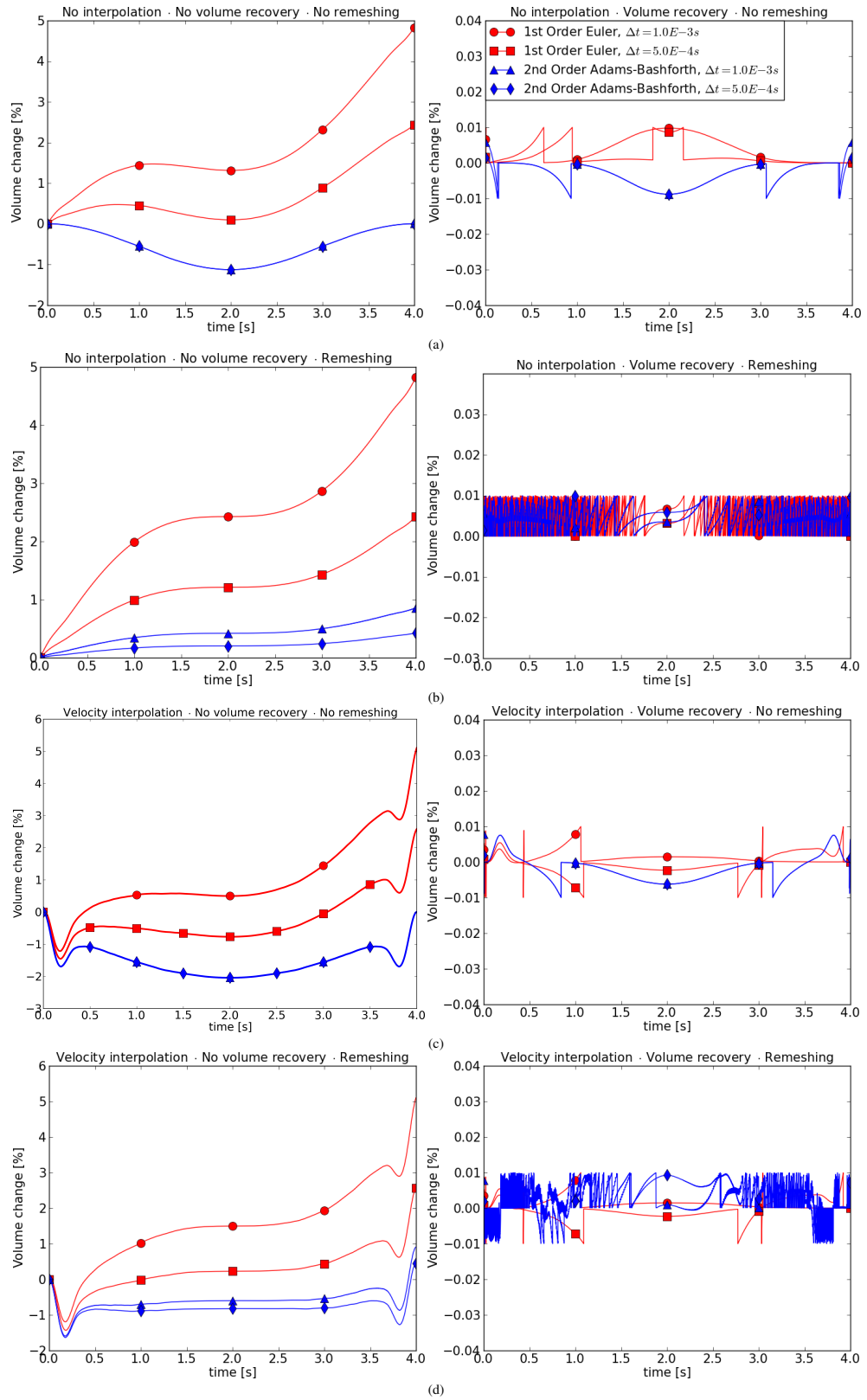


Figure 2. Influence of the volume recovery algorithm on the volume change during the analytical shear flow test. Left column: no recovery performed. Right column: volume recovery applied.

dulations or wrinkles that occurs at the rear of rising bubbles. The TSUR-3D algorithm [6] eliminates such undulations by requiring that the normal of a given edge in the mesh be parallel to the resultant of the approximate normals of the vertices of this edge.

Four cases were run in order to assess the role of the geometric correction tools on the flow simulation. In case A1 the standard Front Tracking Method was used, without any further intervention. In case A2 a volume correction step was applied at the end of the interface advection, while in case A3 only TSUR-3D was used. Case A4 applies both, volume recovery and TSUR-3D algorithms. Table 2 shows the terminal Re achieved in each simulation, as well as the relative error.

Case	Description	Re	Error (%)
A1	Standard Front Tracking Algorithm	23.5	12.3
A2	S1 + Volume Recovery	23.3	11.6
A3	S1 + Wrinkle removal	19.9	3.5
A4	S1 + Volume Recovery + Wrinkle Removal	19.8	4.0

Table 2. A-series: Cases description and error on the terminal Re number.

The effect of the geometric tools on the Re number and terminal velocity is shown in figure 3. Notice that the volume recovery algorithm has little influence at the beginning of the flow. In fact, only after reaching the terminal regime a small decrease in the Re number can be seen. Surface wrinkle removal, on the other hand, affects the early stages of the flow by smoothing the transient region and decreasing the Re maximum at the overshoot region. The final position of the bubbles, depicted on the right, shows that the algorithm affects the rise velocity, not the bubble volume.

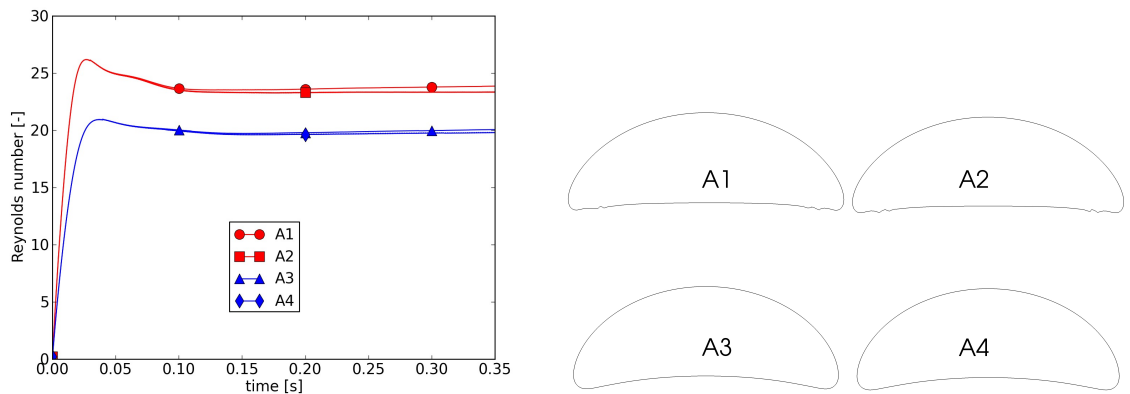


Figure 3. A-series: Terminal Reynolds number depending on the geometric intervention applied.

Figure 4 shows details of the bubble shape at the end of the flow for cases A1 and A3. Figure 5 show that the presence of the wrinkles lead to the formation of artificial vortices on the bubble rear, influencing the flow inside and outside the bubble.

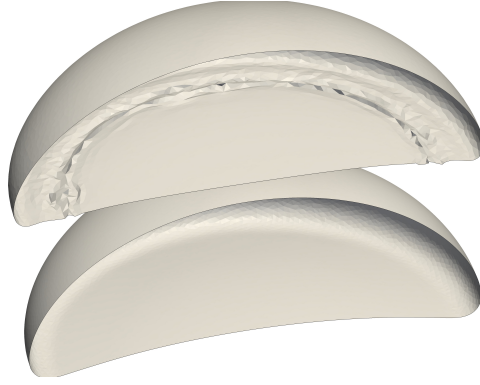


Figure 4. Bubble shape at the end of the simulation. Top: no geometric intervention. Bottom: volume recovery and wrinkle removal.

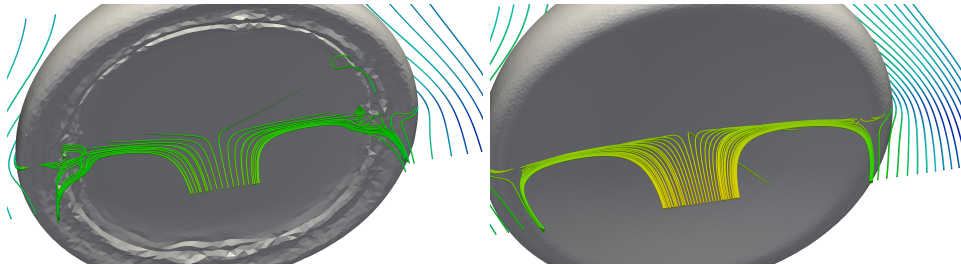


Figure 5. A-series: Streamlines at the rear of the bubble. Case A1 vs A4.

### 3.3. Low Reynolds flows

After assessing the influence of the geometry correction tools on the bubble behaviour, a set of low Re flows was simulated using the volume recovery and TSUR-3D algorithms. For all cases simulated,  $Eo=116$  while  $M$  ranged between 1.31 and 848. Figure 6 shows the time history of the Reynolds number for all cases on the left and the error on the terminal Re on the right. Table 3 shows the results of the simulations and compares the terminal bubble shape and Re number to the experimental work of Bhaga [4]. Despite the difference in the density ratio (1370 in the experiment vs 100 here), the errors were small even for high Morton numbers, which usually demands more effort on the solver. Notice also that in all cases the predicted shapes are in good agreement with the experiments.

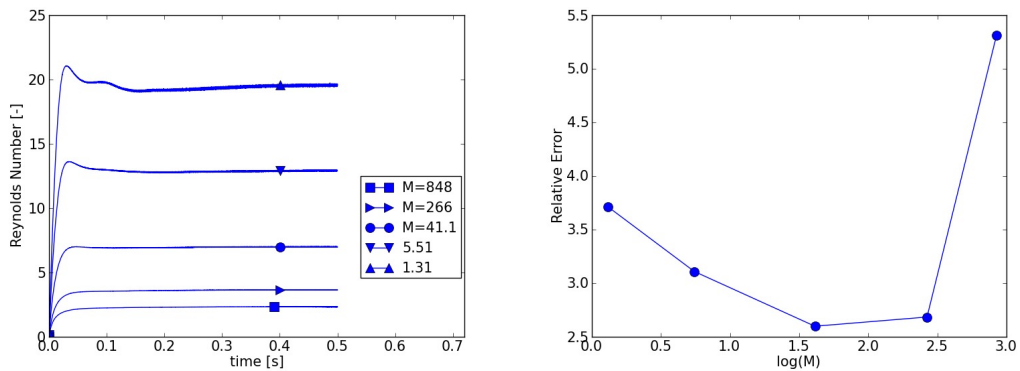


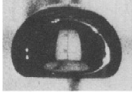


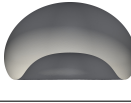

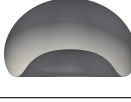

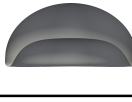


Figure 6. Reynolds profile and terminal Re error for the low Reynolds cases.

Table 3. Comparison of terminal shapes and Reynolds number observed in experiments and predicted by numerical simulations.

Experiments		Simulations	
Eo = 116 M = 848 Re = 2.47			Re = 2.34 $\varepsilon = 5.32\%$
Eo = 116 M = 266 Re = 3.57			Re = 3.67 $\varepsilon = 2.68\%$
Eo=116 M = 41.1 Re = 7.16			Re = 6.97 $\varepsilon = 2.607\%$
Eo=116 M = 5.51 Re = 13.3			Re = 12.89 $\varepsilon = 3.11\%$
Eo=116 M = 1.31 Re = 20.4			Re = 19.64 $\varepsilon = 3.71\%$

#### 4. CONCLUSIONS

The numerical method to solve the Navier-Stokes equations and the front-tracking method for the two phase flow interfaces treatment was implemented in an AMR framework. The methodology was verified and validated under comparison with experimental results. The preliminaries conclusions are that the methodology represents well the main physical characteristics of this very complex problem. The next steps for the present project is to improve this methodology in order to capture coalescence and fragmentation of bubbles in turbulent two phase flows as well as to make numerical experiments to obtain statistical parameters as drag, lift, added mass and Basset force.

#### Acknowledgements

To PETROBRAS, CNPq, FAMEMIG and UFU.

#### References

- [1] Ascher, U.M. and Ruuth, S.J. and Wetton, B.T.R. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):pp. 797–823.
- [2] M. J. Berger and R. J. Le Veque. Adaptive mesh refinement using wave propagation algorithms for hyperbolic systems. *SIAM Journal on Numerical Analysis*, 35(6):2298–2316, 1998.

- [3] Berger, M. J. and Colella, P. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.
- [4] Bhaga, D. *Bubbles in viscous liquids: shapes, wakes and velocities*. PhD thesis, Department of Chemical Engineering, McGill University, Montreal, Canada, September 1976.
- [5] Cenicerros, H.D. and Roma, A.M. A multi-phase flow method with a fast, geometry-based fluid indicator. *Journal of Computational Physics*, 205(2):391 – 400, 2005.
- [6] de Sousa, F.S. and Mangiavacchi, N. and Nonato, L.G. and Castelo, A. and Tomé, M.F. and Ferreira, V.G. and Cuminato, J.A. and McKee, S. A front-tracking/front-capturing method for the simulation of 3d multi-fluid flows with free surfaces. *Journal of Computational Physics*, 198(2):469–499, 2004.
- [7] Geuzaine, C and Remacle, J. F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [8] J. Glim, E. Isaacson, D. Marchesin, and O. McBryan. Front tracking for hyperbolic systems. *Adv. Appl. Math.*, 2:91–119, 1981.
- [9] Jameson, M. J. Berger A. Automatic adaptive grid refinement for the euler equations. *American Institute of Aeronautics and Astronautics Journal*, 23(4):561–568, 1985.
- [10] S. Jin, R. R. Lewis, and D. West. A comparison of algorithms for vertex normal computation. *The Visual Computer*, 21(1-2):71–82, 2005.
- [11] Lindstrom, P. and Turk, G. Fast and memory efficient polygonal simplification. In *Proceedings of IEEE Visualization '98*, pages 279–286, October 1998.
- [12] Mauch, S. *Efficient algorithms for solving static Hamilton-Jacobi equations*. PhD thesis, Caltech, Pasadena CA, 2003.
- [13] Peskin, C. S. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(1):220–252, 1977.
- [14] Peskin, C. S. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.
- [15] Popinet, S. *GTS Library Reference Manual*, 2000.
- [16] Roma, A. M. and Peskin, C. S. and Berger, M. J. An Adaptive Version of the Immersed Boundary Method. *Journal of Computational Physics*, 153(2):509 – 534, 1999.
- [17] Singh, R. and Shyy, W. Three-dimensional adaptive cartesian grid method with conservative interface restructuring and reconstruction. *Journal of Computational Physics*, 224(224):150–167, 2007.
- [18] Stene, J. F. *Numerical Simulation of Interfacial and Multiphase Flows using the Front Tracking Method*. PhD thesis, National University of Singapore, 2010.

- [19] Tryggvason, G. and Bunner, B. and Esmaeeli, A. and Juric, D. and Al-Rawahi, N. and Tauber, W. and Han, J. and Nas, S. and Jan, Y. J. A front-tracking method for the computations of multiphase flow. *J. Comput. Phys.*, 169:708–759, 2001.
- [20] Unverdi, S. A. and Tryggvason, G. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100:25–37, 1992.
- [21] Yokoi, K. A numerical method for free-surface flows and its application to droplet impact on a thin liquid layer. *J. Sci. Comput.*, 35:372–396, 2008.