International Journal of Multiphase Flow 58 (2014) 72–82

Contents lists available at ScienceDirect

International Journal of Multiphase Flow

journal homepage: www.elsevier.com/locate/ijmulflow

A fully adaptive front tracking method for the simulation of two phase flows

M.R. Pivello [a,*], M.M. Villar [a], R. Serfaty [c], A.M. Roma [b], A. Silveira-Neto [a]

[a] Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Av. João Naves de Ávila, 2121, Sta. Mônica, Uberlândia, MG, Brazil
[b] IME-USP, Instituto de Matemática e Estatística da USP, Rua do Matão, 1010, Cidade Universitária, São Paulo, SP, Brazil
[c] CENPES, Cidade Universitária, Avenida 1, Quadra 7, sala 2118, Ilha do Fundão, Rio de Janeiro, RJ, Brazil

ARTICLE INFO

Article history:
Received 11 January 2013
Received in revised form 21 June 2013
Accepted 25 August 2013
Available online 7 September 2013

Keywords:
Front tracking
Adaptive mesh refinement
Volume preserving
Multiphase flow
Rising bubbles
Wobbling

ABSTRACT

This work presents a computational methodology for the simulation of three-dimensional, two-phase flows, based on adaptive strategies for space discretization, as well as a varying time-step approach. The method is based on the Front-Tracking method and the discretization of the *Eulerian* domain employs a Structured Adaptive Mesh Refinement strategy along with an implicit–explicit pressure correction scheme. Modelling of the *Lagrangian* interface was carried out with the GNU Triangulated Surface (GTS) library, which greatly reduced the difficulties of interface handling in 3D. The methodology was applied to a series of rising bubble simulations and validated employing experimental results and compared to literature numerics. Finally, the algorithm was applied to the simulation of two cases of bubbles rising in the wobbling regime. The use of adaptive mesh refinement strategies led to physically insightful results, which otherwise would not be possible in a serial code with a uniform mesh.

1. Background

A bubble rising in a quiescent liquid reaches its terminal velocity when the forces acting on it (drag, buoyancy and weight) are in equilibrium. However, unlike rigid bodies, deformation can take place as a result of the surrounding flow and, also, the transfer of momentum across the interface may induce vortices inside the bubble. Therefore, the bubble shape will depend on the viscous forces, interface forces, and also on the forces from the surrounding flow (de Vries, 2001).

Rising bubble flows can be described in terms of the Eotvos ($Eo = g\Delta\rho\phi^2/\sigma$), Morton ($M = g\Delta\rho\mu_c^4/\rho_c^2\sigma^3$) and Reynolds ($Re = \rho_c U\phi/\mu_c$) numbers, written as functions of the gravity acceleration (g), the density of the continuous phase (ρ_c), the difference between densities of the continuous and disperse phases ($\Delta\rho$), the bubble equivalent diameter (ϕ), the dynamic viscosity of the continuous phase (μ_c), the characteristic flow velocity (U) and the interface tension at the fluid–fluid interface (σ).

Bubbles tend to deform when subjected to external flow fields until normal and shear stresses balance at the fluid–fluid interface. Their shape under the action of gravity in an initially quiescent liquid can be grouped into three large categories: spherical, ellipsoidal and spherical- or ellipsoidal-cap. If the interfacial tension and/or viscous forces are much more significant than inertial forces, bubbles are termed *spherical*. Clift et al. (1978) classify a rising bubble as spherical if its height to width ratio lies within 10% of unity. *Ellipsoidal* bubbles are oblate with a convex shape when viewed from inside and may present axi-symmetry. As inertia forces become more important, ellipsoidal bubbles may undergo periodic dilatation or random wobbling motion, making shape characterization a difficult task (Bhaga and Weber, 1981). Large bubbles usually have flat or indented bases, without fore-and-aft symmetry. Their fore-shape may resemble segments of oblate spheroids of low eccentricity, thence the names *spherical-cap* or *ellipsoidal-cap*. Bubbles in this regime may also develop thin envelopes of dispersed fluid at their bases, usually referred to as skirts (Brennen, 2005).

In the present work, the motion of a single bubble rising in a quiescent liquid is simulated under various flow regimes, ranging from low-*Re*, spherical bubbles, to high-*Re*, low-Morton wobbling bubbles. The simulations were carried out using a Front-Tracking (FT) method coupled with a Structured Adaptive Mesh Refinement for solving the Navier–Stokes equations.

FT method was chosen over VOF (Hirt and Nichols, 1981) or Level Set methods (Sethian and Smereka, 2003) because of its superior capabilities for accurately representing the interface. As a result, more control over fragmentation and/or coalescence is possible. Also, calculating the interface tension force is significantly

* Corresponding author.
E-mail addresses: pivello@mecanica.ufu.br (M.R. Pivello), mmvillar@mecanica.ufu.br (M.M. Villar), rserfaty@petrobras.com.br (R. Serfaty), roma@ime.usp.br (A.M. Roma), aristeus@mecanica.ufu.br (A. Silveira-Neto).

0301-9322/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ijmultiphaseflow.2013.08.009

simpler in FT methods than in VOF and Level Set methods. Annaland et al. (2006) provide a detailed description of pros and cons about the most commonly used methods for multiphase flow simulations.

The Front-Tracking method of Unverdi and Tryggvason (1992) is based on the One-Fluid Formulation (OFF) and on the Immersed Boundary (IB) method of Peskin (1977), Peskin (2002). The Navier–Stokes equations are solved in an *Eulerian* grid and the interface is tracked by an independent, triangular, surface mesh, usually termed *Lagrangian* mesh, on which the interface tension force is calculated. Grid communication is performed based on the IB method, which represents the interface by imposing a force field that is computed on the *Lagrangian* mesh and then spread on the *Eulerian* grid. The velocity field is interpolated onto the Lagrangian mesh by means of Dirac kernels.

2. The Front-Tracking method

In the OFF approach, the Navier–Stokes equations are solved as if a single fluid, with space-dependent physical properties, were used. The presence of the fluid–fluid interface is modelled using a source term for the interface tension force, as shown as follows:

$$\rho\left[\frac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right]=\nabla\cdot[\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)]+ \\ -\nabla p+\rho\mathbf{g}+\mathbf{f}_\sigma, \tag{1}$$

$$\nabla\cdot\mathbf{u}=0. \tag{2}$$

Here, $\mathbf{u}$ is the fluid velocity field, $\rho(\mathbf{x},t)$ is the fluid density, $\mu(\mathbf{x},t)$ is the dynamic viscosity, p is the pressure, $\mathbf{g}$ is the gravity acceleration, and $\mathbf{f}_\sigma$ is the interface tension force, which appears due to both the surface tension between the flow phases and to the surface curvature. The interface is represented by an unstructured, triangulated *Lagrangian* surface mesh, on which the interface force is calculated by integrating the interface tension on a surface element ΔS, as $\delta\mathbf{F}_\sigma=\int_{\Delta S}\sigma\kappa\mathbf{n}ds$, where σ is the surface tension coefficient, κ is twice the mean curvature for three-dimensional domains and $\mathbf{n}$ is the local normal to the surface. By replacing the geometrical relation $\kappa\mathbf{n}=(\mathbf{n}\times\nabla)\times\mathbf{n}$ (Tryggvason et al., 2011) on this equation and using the Stokes theorem, the force on a surface element can be computed without explicitly calculating the surface curvature, via

$$\delta\mathbf{F}_\sigma=\oint_{\delta\Gamma}\sigma\mathbf{t}\times\mathbf{n}d\Gamma, \tag{3}$$

where $\delta\Gamma$ is the boundary of the integration element, $\mathbf{t}$ is the unit tangent and $\mathbf{n}$ is the outward unit normal, both computed at the element boundary.

The force in Eq. (3) may be computed in various ways, and usually the integration is performed directly over the mesh elements. Tryggvason et al. (2001) compute the tangent vectors from the end points of the element edges, but perform a local surface fit in order to calculate the normal vectors. Deen et al. (2004) compute the tangent vectors likewise, but use the normals at the adjacent elements. Shin and Juric (2002), on the other hand, use the resultant of $\mathbf{t}\times\mathbf{n}$ computed on both elements sharing an edge. In the present work, the latter approach will be employed, and Eq. (3) is discretized as

$$\mathbf{F}_\sigma=\sum_j\sigma(\mathbf{t}_j\times\mathbf{n}_j), \tag{4}$$

where $\mathbf{F}_\sigma$ is the vector force acting on a given element, $\mathbf{n}_j$ is the outer unit-normal associated to edge j and $\mathbf{t}_j$ is the non-normalized tangent vector at edge j. Notice that the indices are not related to Einstein notation.

After being computed on the interface mesh, the force is spread on the *Eulerian* grid, as in the IB method (Peskin, 2002), in the vicinity of the interface position. The Navier–Stokes equations can then be solved, yielding the pressure and velocity fields. Interface advection is performed in a *Lagrangian* fashion, using the velocity field interpolated from the *Eulerian* domain onto the mesh surface vertices.

Interpolation and spreading processes are performed as described in Tryggvason et al. (2001), using the following equation as a Dirac kernel (Peskin, 1977):

$$W(r)=\begin{cases}\frac{1}{4}(1+cos(\frac{\pi}{2}r)), & r<2,\\ 0, & r\geqslant 2,\end{cases} \tag{5}$$

and

$$r=\frac{x-X}{h_x},\quad\frac{y-Y}{h_y},\quad\frac{z-Z}{h_z}. \tag{6}$$

Physical properties such as viscosity and density are not literally advected. Instead, the position of the *Lagrangian* interface, explicitly tracked in time, is used to locate the constant, but different, material properties defined in the interior and exterior of the bubble. This is achieved by means of an indicator function, which yields a scalar field associated to each flow phase. Most researchers use a Poisson equation in this step, since the scalar field is obtained simultaneously in the entire domain. The high computational cost of solving such equation, however, is well known.

Alternatively, Ceniceros and Roma (2005) employed the Closest Point Transform (CPT) as the basis for an indicator function. CPT consists in generating an implicit representation of the interface geometry by computing its distance field (Mauch, 2003). When used as an indicator function, however, its computation is limited to an interval $[-\gamma,+\gamma]$, where γ is the absolute value of the largest distance from the interface to a given point in the *Eulerian* domain. The remaining of the domain is assigned with a constant value (e.g.: $+\gamma$ inside the dispersed phase and $-\gamma$ outside it). A smoothed Heaviside function is then applied to it, so that the interval $[-\gamma,+\gamma]$ is mapped to $[0,1]$. In the present work, expression (7) is used (Yokoi, 2008; Ceniceros et al., 2010b).

$$H(\varphi)=\begin{cases}1, & \varphi>\gamma\\ \frac{1}{2}\left(1+\frac{\varphi}{\gamma}+\frac{1}{\pi}sin\left(\frac{\pi\varphi}{\gamma}\right)\right), & \|\varphi\|\leqslant\gamma\\ 0, & \varphi<-\gamma.\end{cases} \tag{7}$$

Finally, the indicator function can be used for calculating the distribution of the density (ρ) and viscosity (μ) fields as

$$\rho(\varphi)=H(\varphi)\rho_1+(1-H(\varphi))\rho_2, \\ \mu(\varphi)=H(\varphi)\mu_1+(1-H(\varphi))\mu_2. \tag{8}$$

3. Temporal discretization

The time discretization scheme employed in the present work is based on Ceniceros et al. (2010a) and Ceniceros et al. (2010b). A few changes were introduced, so that various second-order, semi-implicit schemes were parametrized according to the following equation:

$$\frac{1}{\Delta t}\rho^{n+1}(\phi)(\alpha_2\mathbf{u}^{n+1}+\alpha_1\mathbf{u}^n+\alpha_0\mathbf{u}^{n-1})=\beta_1 f(\mathbf{u}^n)+\beta_0 f(\mathbf{u}^{n-1}) \\ +\lambda\left[\theta_2\nabla^2\mathbf{u}^{n+1}+\theta_1\nabla^2\mathbf{u}^n+\theta_0\nabla^2\mathbf{u}^{n-1}\right]-\nabla p^n+\rho^{n+1}\mathbf{g},\nabla\cdot\mathbf{u}^{n+1}=0. \tag{9}$$

In this equation, α_i, β_i and θ_i are given by: $\alpha_0=(2\gamma-1)\omega^2/(1+\omega)$, α_1 -= $(1-2\gamma)\omega-1$, $\alpha_2=1+2\gamma\omega/1+\omega$, $\beta_0=1+\gamma$, $\beta_1=-\gamma\omega$, $\theta_0=c/2$,