INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **47**:1393–1401
Published online 31 January 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.899

A Lagrangian level-set approach for the simulation of incompressible two-fluid flows‡

F. S. Sousa[1,*,†] and N. Mangiavacchi[2]

[1]*Department of Computer Science and Statistics, ICMC, University of São Paulo, São Carlos, Brazil*
[2]*Department of Mechanical Engineering, State University of Rio de Janeiro, Rio de Janeiro, Brazil*

SUMMARY

A Lagrangian level-set method to solve incompressible two-dimensional two-fluid flows is presented. The Navier–Stokes equations are discretized by a Galerkin finite element method. A projection method is employed to decouple the system of non-linear equations. The interface between fluids is represented by the zero level set of a function ϕ plus additional marker points of the computational mesh. In the standard Eulerian level-set method, this function is advected through the domain by solving a pure advection equation. To reduce mass conservation errors that can arise from this step, our method employs a Lagrangian technique which moves the nodes of the finite element mesh, and consequently, the information stored in each node. The quality of the mesh is controlled by a remeshing procedure, avoiding bad triangles by flipping edges, inserting or removing vertices from the triangulation. Results of numerical simulations are presented, illustrating the improvements in mass conservation and accuracy of this new methodology. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: moving mesh; level-set; projection method; finite element; two-fluid flows

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 [1] 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

*Correspondence to: F. S. Sousa, Department of Computer Science and Statistics, ICMC, University of São Paulo, São Carlos, Brazil.
†E-mail: fsimeoni@lead.lcmc.usp.br, F.S.Sousa@klft.tn.tudelft.nl
‡Short listed for The Bill Morton Prize.

Contract/grant sponsor: FAPESP and CNPq; contract/grant number: FAPESP 01/12623-5 and 00/03385-0, CNPq 460473/2001-8

Received 27 April 2004
Revised 30 November 2004
Accepted 7 December 2004

the literature (e.g. References [1, 2]). 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 [3], 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 $\mathbf{u}$ 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 the literature (e.g. Reference [4]).

Hybrid methods were also developed, trying to combine the precision of front-tracking with the simplicity of front-capturing. Tornberg and Engquist [5] 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.* [6] presents 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 increase of the available computational resources, moving-mesh techniques are becoming popular and gaining interest from many researchers. Although this is still a new methodology nowadays, we can find successful moving-mesh and Lagrangian methods developed in early 1980s and 1990s. For instance, Ryskin and Leal describe in Reference [7] a finite difference technique where the mesh is adapted to the deformation of the front, computed in a boundary-fitted co-ordinate system. This technique is applied for the computation of deformable particles, but resolving only the continuous fluid. Ogũz and Prosperetti [8] 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 [9] 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 a Lagrangian level-set method for two-fluid flow simulations. It uses the same pseudo-concentration function ϕ to represent the fronts, but its advection is performed by moving all the vertices of the mesh by the velocity field, where the values of ϕ are stored. With this approach, the mass conservation of the level-set method is improved, 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

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

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 Reference [10], 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 $\mathrm{D}(\rho\mathbf{u})/\mathrm{D}t$ 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 [11], and formalized by Gresho and his co-workers [12, 13]. 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. 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 [12]. The method was adapted for the two-fluid flow case, following the ideas stated by Sousa *et al.* [6]. 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 $\mathrm{D}(\rho\tilde{\mathbf{u}})/\mathrm{D}t=(1/Re)\nabla\cdot\{\mu(\nabla\tilde{\mathbf{u}}+\nabla\tilde{\mathbf{u}}^{\mathrm{T}})\}+(\rho/Fr^2)\mathbf{g}+(\sigma\kappa\delta/We)\mathbf{n}$;
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.

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: $\mathbb{V}=\{\mathbf{v}\in L^2(\Omega):v_i\in H^1(\Omega),\forall i\}$ and $\mathbb{P}=\{q\in L^2(\Omega):\int_\Omega q\,\mathrm{d}\Omega=0\}$, where $H^1(\Omega)$ is a Sobolev space, and the sub-spaces $\mathbb{V}_{\mathbf{u}_\Gamma}=\{\mathbf{v}\in\mathbb{V}:\mathbf{v}=\mathbf{u}_\Gamma \text{ em } \Gamma_1\}$ and $\mathbb{V}_{\mathbf{0}}=\{\mathbf{v}\in\mathbb{V}:\mathbf{v}=\mathbf{0} \text{ em } \Gamma_1\}$, the weak formulation of the

problem can be written as: find $\mathbf{u}(\mathbf{x},t)\in\mathbb{V}_{\mathbf{u}_\Gamma}$ and $p(\mathbf{x},t)\in\mathbb{P}$ such that

$$m\left(\frac{\mathrm{D}(\rho\mathbf{u})}{\mathrm{D}t},\mathbf{w}\right)-c(p,\mathbf{w})+k\left(\frac{\mu}{Re},\mathbf{u},\mathbf{w}\right)-m\left(\frac{\rho}{Fr^2}\mathbf{g},\mathbf{w}\right)-f_{\mathrm{I}}\left(\frac{\sigma}{We},\mathbf{w}\right)=0 \tag{2}$$

$$c(q,\mathbf{u})=0 \tag{3}$$

for all $\mathbf{w}\in\mathbb{V}_{\mathbf{0}}$, $q\in\mathbb{P}$. The functionals in (2) 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 (2)–(3) is made using linear shape functions and Galerkin weighting functions. Integrating over the triangular elements results in an ODE system, which is 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

The technique used to capture the interface is a level-set method [14, 15], in which the front is represented by the zero level set of a pseudo-concentration function $\phi:\mathbb{R}^2\to\mathbb{R}$. 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 mesh is moved with the fluid particles and the values of ϕ 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. For instance, the viscosity is computed as $\mu=\mu_0+(\mu_1-\mu_0)H_\varepsilon(\phi)$, where $H_\varepsilon(\phi)=\frac{1}{2}+\frac{1}{32}(45\varepsilon-50\varepsilon^3+21\varepsilon^5)$, for $|\phi|\leqslant\varepsilon$, is the smoothed Heaviside function extracted from Reference [15]. The parameter ε is usually kept very small to avoid spreading the interface over too many cells.

The level-set function ϕ is initialized as a signed distance function, which means that $\|\nabla\phi\|=1$. As it is advected through the domain, it does not necessarily correspond to a distance function any more. Keeping ϕ as a distance function is important to ensure that the interface has finite thickness. This is achieved by the reinitialization of ϕ. Periodically,

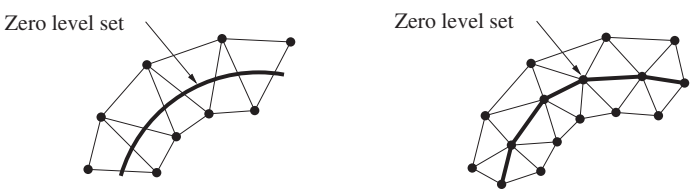

Figure 1. Interface representation: standard level-set (left); and the Lagrangian approach (right).

the new values of ϕ are computed as $\tilde{\phi}(\mathbf{x}) = \min_{y \in I} |\mathbf{x}-\mathbf{y}| \cdot \mathrm{sgn}(\phi)$. This operation is not done at every time-step and it costs $N_D \times N_\mathrm{I}$ 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 drastically reduces the number of distance computations.

6. MESH CONTROL PROCEDURE

As the mesh is moved, bad elements can appear, compromising the finite element approximation. To avoid this problem, a mesh control procedure is employed. Elements are tested by their aspect ratio, which is measured by the ratio between the radius of the circumscribing circle and the length of the shortest edge of the triangle. A triangle can be classified as bad if its ratio is greater than β, for some $\beta > \sqrt{3}/3$. In that case, it has to be removed from the triangulation.

A bad triangle can be also classified in two types: a *cap triangle* possesses a large circumscribing circle radius comparing to its edges. It can be removed by flipping the largest edge, inserting a new point in the middle of the largest edge or deleting the vertex which possesses the largest angle; a *thin triangle* possesses a very small edge comparing to the other edges and to the circumscribing circle radius. It can be removed from the triangulation by deleting the shortest edge or inserting a point in the largest edge.

The insertion, deletion and flipping operations are also controlled by additional parameters, like the maximum and minimum sizes for the edges. They are also restricted by the interface. Insertion and deletion of interface vertices are allowed to keep it smooth. Flippings of interface edges are not allowed to avoid loss of information at the interface. Insertion and, in particular, deletion work in the sense of keeping the mesh regular. The singularity of the mesh, as interfaces begin to merge or breakup, is limited by the choice of the smallest edge size. If a very accurate description of the merge or break-up process is desired, it suffices to prescribe a very small value for the shortest allowable edge, but then the computational cost will be accordingly larger. In this sense, insertion and deletion ameliorate the singularity of the geometry, and they provide a convenient approach to select the relevant scales for the analysis.

This procedure is applied periodically over the entire mesh. The vertices and edges are inserted into and removed from the mesh such that the resulting triangulation is Delaunay in the vicinity of the changed region. Although the entire mesh is not Delaunay, this ensures that the local triangulation is optimal.

7. NUMERICAL RESULTS

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

Table I. Comparison for the static bubble simulation.

	Analytic	Eulerian level-set	Coarse mesh	Fine mesh
Elements	—	3032	2056	5910
Δp	2	2.030	2.014	2.010
Error	—	1.5%	0.7%	0.5%

7.1. 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. This problem was simulated in a 2×2 domain discretized by 2056 elements in the coarse mesh and by 5910 elements in the fine mesh. Table I shows the comparison of the results for the pressure jump between our technique and standard level-set representation of the interface. We compute the pressure jump at the interface as the difference between the pressures inside and outside the bubble. In symbols, $\Delta p = p_0 - p_1$, where p_0 is the pressure inside and p_1 is the pressure outside the bubble. The analytical pressure jump at the interface is known to be $\Delta p^* = \sigma/R$, where R is the radius of the bubble. In Table I we show the error as the percentage deviation of the calculated values from analytical value.

The comparison shows better results for our method because the interface is explicitly represented by nodes and edges in the computational mesh, giving a more accurate position of the interface. The parasitic currents are $O(10^{-4})$ in the fine mesh, which is an acceptable value for our simulations.

7.2. Rising bubble

The rising of a single bubble immersed in a heavier fluid is simulated. The simulation was performed in a 3×6 domain initially discretized by 2438 elements in the coarse mesh and 4246 elements in the fine mesh. The non-dimensional parameters for this problem are the Morton and Eötvös numbers, given by $M = 7.5 \times 10^{-5}$ and $Eo = 2.5$, respectively. The bubble diameter is $D = 1$ and the bubble is about 2 times lighter and less viscous than the surrounding fluid. Figure 2 shows the mesh configuration at the non-dimensional times $t = 0$, 8 and 16, the streamlines and u-velocity contours at $t = 16$.

The volume of the bubble was computed and compared to the classic Eulerian level-set method. In two dimensions, this volume is calculated as $V_b = \int_V \mathrm{d}v$. In the classical level-set representation, the position of the interface is not well known, so linear interpolations are used to compute the bubble volume. In our method, the position of the interface is defined by edges and vertices of the triangulation, so that the bubble volume is computed directly by summing the volumes of all the interior triangles of the bubble. In this case, our method obtained an error around 1.4%, while the classical level-set formulation gives an error about 7% in mass conservation.

The rising Reynolds number for the bubble was computed for both coarse and fine mesh, giving $Re \approx 4.7$, but a better approximation was obtained for the fine mesh. The percentage of bubble volume variation from the initially cylindrical bubble was computed as 3% for the coarse mesh and 1.4% for the fine mesh. Part of this error can be associated to the removal

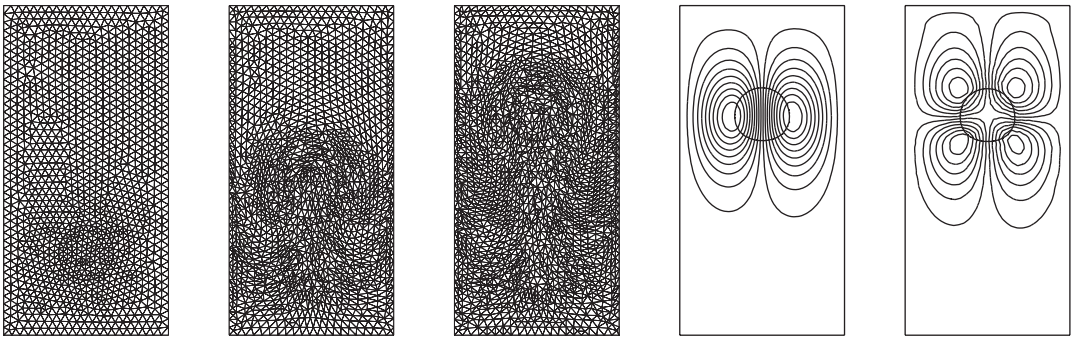

Figure 2. Rising of a single bubble, from left to right: initial mesh, mesh at $t = 8$, mesh at $t = 16$, stream lines, and u-velocity contours.

of interface vertices, which is not a mass conserving procedure, but it is necessary to keep the good quality of the mesh.

An estimate of the numerical rate of convergence of the method for the problem of the single rising bubble can be obtained based on the bubble mass conservation. Considering the bubble mass (volume) conservation error for two different meshes, the numerical rate of convergence can be estimated by $n = \log(E_1/E_2)/\log(h_1/h_2)$ where E_1, E_2, h_1 and h_2 are the mass conservation errors and representative edge sizes of meshes 1 and 2. Employing data obtained from this simulation, we obtain $n = 2.04$, which is consistent with the expected second-order accuracy of the method.

7.3. Bubble coalescence

To show that the method can easily deal with topological changes at the interface, we simulated the coalescence of two rising bubbles. The parameters for this problem are the same as the previous simulation. As two interfaces get closer to each other, at a distance of one element, coalescence takes place. We simply check for the existence of elements in the surrounding fluid that have three interface vertices, and change the material of this element. Removal of vertices that no longer belongs to the interface is also performed in order to avoid singularities. Figure 3 illustrates the coalescence of two bubbles. Details of the mesh configuration before and after coalescence can also be seen in this figure.

8. CONCLUSION

The method presented in this paper is a Lagrangian approach for the level-set method, which produces better mass conservation properties while still easily dealing with topological changes of the fronts. The code implements a moving finite element unstructured mesh as well as a control procedure to avoid bad elements in the mesh. The conservation equations are solved by a projection method to decouple the acceleration and pressure.

Results show good mass conservation and accuracy when compared with standard Eulerian level-set approach. The static bubble simulation was performed to validate the surface tension

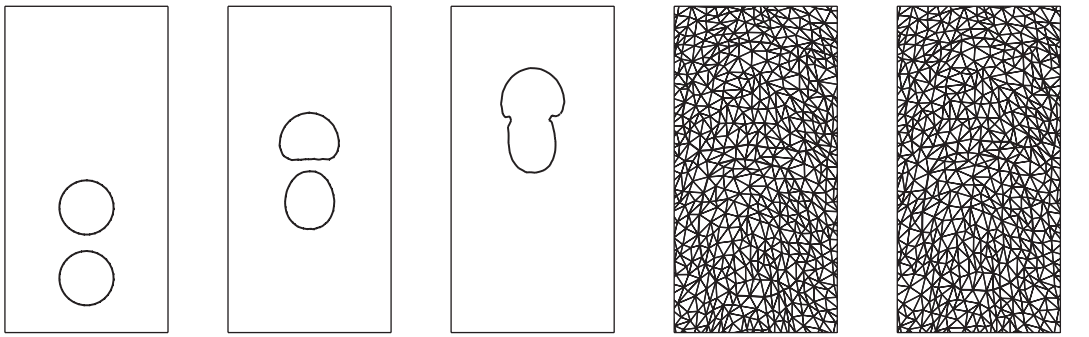

Figure 3. Coalescence of two bubbles, from left to right: interface at times $t = 0$; $t = 7.5$; and $t = 12.4$; details of the mesh configuration before and after coalescence.

calculation and to compute the parasitic currents that may appear in the flow. The results show better accuracy than the Eulerian approach, and small values for the parasitic currents. The results for a rising bubble are also presented, where the mass conservation errors are found to be much smaller than the standard Eulerian approach. A bubble coalescence simulation was also carried out, showing that the method can easily deal with topological changes at the fronts. The presented results are for low Reynolds number flows. The performance of the method for high Reynolds number flows will be addressed in the future.

ACKNOWLEDGEMENTS

During the final part of this work F. S. Sousa was at the Kramers Laboratorium, Delft University of Technology.

REFERENCES

1. Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics* 1992; **100**:25–37.
2. Esmaeeli A, Tryggvason G. Direct numerical simulations of bubbly flows. Part 1. Low Reynolds number arrays. *Journal of Fluid Mechanics* 1998; **337**:313–345.
3. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.
4. Pijl SP, Segal A, Vuik C. Modelling of three-dimensional bubbly flows with a mass-conserving level-set method. *Proceedings of ECCOMAS 2004*, vol. II, CD-ROM, Jyvaskyla, Finland, 2004.
5. Tornberg AK, Engquist B. The segment projection method for interface tracking. *Communications on Pure and Applied Mathematics* 2003; **56**:47–79.
6. 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.
7. Ryskin G, Leal LG. Numerical solution of free-boundary problems in fluid mechanics. *Journal of Fluid Mechanics* 2004; **148**:1–43.
8. Ogũz HN, Prosperetti A. Bubble entrainment by the impact of drops on liquid surfaces. *Journal of Computational Physics* 1990; **219**:143–179.
9. 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.
10. Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.

11. Chorin AJ. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**: 745–762.
12. 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. *International Journal for Numerical Methods in Fluids* 1990; **11**:587–659.
13. 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.
14. Sussman M, Smereka P. Axisymmetric free boundary problems. *Journal of Fluid Mechanics* 1997; **341**: 269–294.
15. Tornberg AK, Engquist B. A finite element based level set method for multiphase flow applications. *Computing and Visualization in Science* 2000; **3**:93–101.
16. Zienkiewicz OC, Taylor RL. *The Finite Element Method Volume 1: The Basics* (5th edn). Butterworth-Heinemann: Stoneham, MA, 2000.

