Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 218 (2006) 479–494
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

New finite-element/finite-volume level set formulation for modelling two-phase incompressible flows

Sergey V. Shepel *, Brian L. Smith

Thermal-Hydraulics Laboratory, Paul Scherrer Institut, CH-5232, Villigen PSI, Switzerland

Received 4 October 2004; received in revised form 13 February 2006; accepted 20 February 2006
Available online 31 March 2006

Abstract

We present a new level set method for modelling two-phase incompressible flows with moving interface boundaries. The method is based on a finite element (FE) treatment of the equations governing the motion of the interface separating two immiscible phases, and a finite volume (FV) description of the fluid flow equations. The performance of the method is demonstrated and validated using several sample problems involving two-phase, isothermal flows with density ratios of the order of 10^2–10^3, and viscosity ratios as high as 7×10^4. Surface tension effects are included. The present formulation displays good convergence properties, and it is relatively simple to program. This allows the method to be implemented into existing commercial FV-based computational fluid dynamics (CFD) software, thereby extending their application range. We have successfully implemented the method in the code CFX-4.

Keywords: Two-phase flow; Interface tracking; Level set; Finite element; Finite volume; Surface tension; Validation; CFX

1. Introduction

Nowadays, there is a large choice of interface-tracking methods that can be used for modelling of flows with moving inter-phase boundaries, for instance: the front-tracking method [1,2], the boundary integral method [3,4], the phase-field method [5], the volume-of-fluid (VOF) method [6–8], and the level set method (see book by Sethian [9]). Each method has its own advantages and disadvantages, and for this reason it is not possible to assert which method is generally superior. We believe that the success of using a particular interface-tracking method depends ultimately on the problem under consideration and the computer resources available, and for each method there is a class of problems for which one method performs better than the others.

In the present work, we are concerned with further development of the level set method, which belongs to the group of Eulerian interface-tracking methods. The central idea of this method is the use of a continuous, multi-dimensional, scalar function, the spatial distribution of which contains the embedded information on

* Corresponding author. Present address: Laboratory for Hydraulic Machines, Swiss Federal Institute of Technology, Avenue de Cour, 33 bis, CH-1007 Lausanne, Switzerland. Tel.: +41 021 693 2558; fax: +41 021 693 3554.
E-mail addresses: sergey.shepel@epfl.ch (S.V. Shepel), brain.smith@psi.ch (B.L. Smith).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.02.008

the location of the interface. The evolution of this function in space and time is governed by an advection equation, combined with a special renormalization procedure [9].

There has been much progress [9–21] in the construction of various level set formulations (the number of publications on this subject is so large that we refer only to those which are directly relevant to the work presented here). In our paper, we assess the performance of a new level set formulation based on a *non-conservative* FE method for solving the level set equations, and a *conservative* FV method for solving the fluid flow equations. The novelty of our approach is the coupling of the two methods into a single Finite-Element/Finite-Volume formulation. Specifically, we apply the formulation to model two-phase, isothermal, incompressible flows in which the phases are separated by a distinct interface. The phases are assumed to have different values for density and viscosity.

The main motivation for our work was the decision to implement an advanced interface-tracking numerical method into the commercial, FV-based CFD code CFX-4 [22]. The necessity for such an upgrade of the code arose from the fact that CFX-4 has an out-of-date interface-tracking algorithm as standard, and which is extremely diffusive [23], although the solver for the Navier–Stokes equations is quite robust, and accurate to second-order if desired. Since we are interested in the most general kind of applications for complex geometries, we chose the Streamline-Upwind/Petrov-Galerkin (SUPG) FE formulation of the level set method [15], because it is readily extendible to three-dimensions, is second-order accurate, and may be adjusted to non-regular, body-fitted grids.

Connecting the non-conservative SUPG FE level set interface-tracking method to the conservative FV Navier–Stokes solver raises questions regarding accuracy, robustness and mass conservation. It should be noted that such *mixed* level set formulations have been used before for compressible flows (e.g. [11,16]). On the other hand, review of the literature shows that, when modelling incompressible flows, the fluid flow equations and the level set interface-tracking equations are traditionally written in the non-conservative form [12–14,18–21]. The conservative form of the fluid flow and interface-tracking equations for incompressible flows has been employed and validated in the coupled level set/VOF (CLSVOF) method developed by Sussman and Puckett [17]. For the mixed formulations, however, there is a lack of development, though such approaches have a strong potential from the point of view of implementation into the framework of existing commercial CFD codes, many of which incorporate conservative FV Navier–Stokes solvers.

The first results of our investigations have already been reported [23]. In the present work, we provide new results for the validation study of the FE/FV level set interface-tracking method. The method is tested on several selected problems for which there are available experimental data, or solutions obtained analytically/numerically. Special attention is paid to the mass conservation properties of the scheme, since the level set method is often challenged on this issue. Although we have implemented the FE/FV level set method into the commercial code CFX-4, it should be emphasized that the formulation is general, and is not restricted to this particular code.

2. Problem formulation

The local equations governing the motion of an unsteady, viscous, incompressible fluid (either liquid or gas) are the Navier–Stokes equations, which in a conservative formulation are given as

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) = \mathbf{B} - \nabla p + \nabla\cdot(2\mu\mathbf{D}), \tag{1a}$$

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0, \tag{1b}$$

where $\mathbf{u}$ is the velocity vector, t is the time, ρ is the density, $\mathbf{B}$ is the body force, p is the pressure, μ is the dynamic viscosity, and $\mathbf{D}$ is the rate of deformation tensor, with the components $D_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$. The stress tensor $\boldsymbol{\tau}$ is given by $\boldsymbol{\tau} = -p\mathbf{I} + 2\mu\mathbf{D}$, where $\mathbf{I}$ is the identity matrix.

In the level set method, the interfacial surface is represented using a scalar function $\phi(\mathbf{x}, t)$, which is set equal to the minimum signed distance to the interface; i.e. $\phi = -d(\mathbf{x}, t)$ on one side of the interface, $\phi = +d(\mathbf{x}, t)$ on the other, and $\phi = 0$ on the interface itself, where $d(\mathbf{x}, t)$ is the shortest distance from the point $\mathbf{x}$ to the interface. Therefore, the physical interface is associated with the zero level set contour of ϕ, whereas the sign of ϕ iden-

tifies the phases. The volumes occupied by the two phases are correspondingly given by the integrals over the domain: $\int_\Omega H(\phi)\,\mathrm{d}\Omega$ and $\int_\Omega (1 - H(\phi))\,\mathrm{d}\Omega$, where $H(\phi)$ is the discontinuous Heaviside function.

In order to avoid the appearance of instabilities near the interface, the jump in phase properties across the interface needs to be smoothed [12]. The density and viscosity of the two-phase fluid are thus interpolated across the interface as follows:

$$\rho(\mathbf{x}, t) = \rho_2 + (\rho_1 - \rho_2) H_\varepsilon(\phi(\mathbf{x}, t)), \tag{2a}$$

$$\mu(\mathbf{x}, t) = \mu_2 + (\mu_1 - \mu_2) H_\varepsilon(\phi(\mathbf{x}, t)), \tag{2b}$$

in which subscripts 1 and 2 denote the values on the two sides of the interface. Here, $H_\varepsilon(\phi)$ is a smoothed Heaviside function, the use of which aids numerical stability. Following the work of Sussman et al. [12], we use the following expression for $H_\varepsilon(\phi)$:

$$H_\varepsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\varepsilon, \\ (\phi + \varepsilon)/(2\varepsilon) + \sin(\pi\phi/\varepsilon)/(2\pi) & \text{if } |\phi| \leqslant \varepsilon, \\ 1 & \text{if } \phi > \varepsilon, \end{cases} \tag{3}$$

where ε is a small parameter of the order $O(h)$, and h is the average size of mesh cells near the interface. By using the smoothed Heaviside function defined by Eq. (3), one effectively assigns to the numerical interface a fixed, finite thickness of 2ε, over which the phase properties are interpolated. For each application, the value of the interface half-thickness ε is chosen to be as small as possible for accuracy of the simulation, within the constraints of numerical stability of the solution procedure. We have found that for the FE/FV level set method $\varepsilon = 2h$ suffices in most cases, so this value is used in all applications presented below, unless stated otherwise.

The interface is carried by the flow, so the evolution of ϕ is determined by the following transport equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{4}$$

The surface tension is modelled as a body force, $\mathbf{B}_{\mathrm{st}}$, concentrated at the interface. The functional form of $\mathbf{B}_{\mathrm{st}}$ that leads to stable simulations is given by

$$\mathbf{B}_{\mathrm{st}} = \sigma\kappa\mathbf{n}\delta_\varepsilon(\phi), \tag{5}$$

where σ is the surface tension, κ is the local curvature of the interface, and $\delta_\varepsilon(\phi)$ is a smoothed delta function, defined as the derivative of $H_\varepsilon(\phi)$ with respect to ϕ. The gradient to the interface, $\mathbf{n}$, and the interface curvature, κ, are given by

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \quad \kappa(\phi) = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}. \tag{6}$$

Earlier [24], we have analyzed an alternative model in which the surface tension force was imposed only in the computational cells containing the interface. However, we found this model to be less robust and less general than the model defined by Eq. (5), in which the surface tension is distributed in a narrow band around the interface.

When solving Eq. (4) for the advection of the level set function, ϕ , it ceases to be the signed distance function, although this property is tacitly employed in Eqs. (2) for interpolation of the phase properties. Consequently, during a transient simulation, the level set field ϕ needs to be reinitialized, preferentially at every time step. An efficient reinitialization method that can be cast into the SUPG FE formulation has been proposed by Sussman et al. [14], and is based on solving for the steady-state solution of the following equation:

$$\frac{\partial \phi}{\partial \tau} + \mathrm{sign}(\phi_0)(|\nabla\phi| - 1) = \lambda\delta_\varepsilon(\phi)|\nabla\phi_0|, \tag{7}$$

where τ is a time-like variable (different from physical time, t), ϕ_0 is the initial distribution of the level set function before reinitialization, and λ is a correction coefficient ensuring mass conservation up to the first-order term in the Taylor expansion of the integral $\partial_\tau \int_\Omega H_\varepsilon(\phi)\,\mathrm{d}\phi$ (see [14] for details of the calculation of λ).

Regarding the reinitialization procedure based on solving Eq. (7), there is one aspect that needs to be mentioned. Eq. (7) is the modified version of the original reinitialization equation proposed by Sussman et al. [12],

the only difference being the presence of the non-homogeneous term. This term was added by Sussman et al. [14] to improve stability and mass conservation. We have found that, in the case of interfaces without sharp corners, the SUPG FE method solves Eq. (7) to second-order accuracy [23], and the inhomogeneous term in the equation does improve stability and mass conservation. However, this conclusion is true only if the finite elements in the vicinity of the interface are approximately of the same size. If the mesh size distribution is not uniform, the presence of the non-homogeneous term in Eq. (7) can actually degrade accuracy compared to that obtained using the homogeneous version. This observation is demonstrated in the first test case (filling a container with water) described later. Ultimately, we have adopted the following strategy: for grids with a uniform cell-size distribution, the level set field is reinitialized by solving Eq. (7), whereas the homogeneous version of equation is used otherwise.

Since the level set function needs to be advected and reinitialized only in the vicinity of the interface, we employ the so-called *narrow-band approach* [9], in which Eqs. (4) and (7) are solved only in a band of elements located around the zero level set of ϕ. For details of solving Eq. (7) using the SUPG FE method, see [25].

3. Numerical scheme

3.1. FE–FV coupling

The SUPG FE level set method is implemented into the commercial code CFX-4 via the standard set of user subroutines. The CFX-4 Navier–Stokes solver is used to generate the flow velocity field, whereas the level set subroutines provide the fluid configuration. The code CFX-4 uses non-staggered, structured grids consisting of hexahedral cells, with all the variables, including velocity, pressure and bulk body forces, defined at cell centres. The cells, if necessary, can be body-fitted to accommodate domains of complex geometry. Boundary conditions are imposed by introducing so-called boundary nodes, located at the centres of the cell faces bordering the fluid domain, and by using dummy cells on the other side of the boundary [22].

The Navier–Stokes equations (1) are here solved using a second-order-accurate FV method. The flux quantities on cell boundaries are computed using the Rhie–Chow algorithm [26]. Space differencing is centered, except for the advection quantities, which are computed using a second-order scheme for which velocities on control-volume faces are obtained by extrapolation from two upwind points. A second-order, backward scheme is used for the time discretisation. The discretised equations are solved using a nested (inner and outer) iteration technique based on the SIMPLEC velocity–pressure coupling algorithm [27].

In CFX-4, the properties of the two-phase mixture in a cell containing the interface are approximated as

$$\rho(\mathbf{x},t) = \rho_1 F_1(\mathbf{x},t) + \rho_2 F_2(\mathbf{x},t), \tag{8a}$$

$$\mu(\mathbf{x},t) = \mu_1 F_1(\mathbf{x},t) + \mu_2 F_2(\mathbf{x},t), \tag{8b}$$

where F_i, $i = 1, 2$, are the volume fractions of the respective phases in the cell. Hence, in order to approximate the phase properties according to the level set model defined by Eqs. (2), the F_1 and F_2 need to be assigned new values according to the following prescription:

$$F_1(\mathbf{x},t) = H_\varepsilon(\phi(\mathbf{x},t)), \quad F_2(\mathbf{x},t) = 1 - H_\varepsilon(\phi(\mathbf{x},t)). \tag{9}$$

These quantities are specified at every time step, overwriting the corresponding entries generated internally by CFX-4.

An alternative procedure for interpolation of phase properties across the interface could be based on estimating the cell-volume-averaged values of quantities F_1 and F_2 using a numerical quadrature, and substituting these into Eqs. (8). However, this approach remains to be investigated.

In the SUPG FE method, variables are defined at element vertices. However, as was mentioned above, in the FV mesh used by the CFX-4 solver, the variables are defined at cell centres. Therefore, if the same mesh were used to solve both the fluid flow equations (1) and the level set equations (4) and (7), interpolation procedures would have to be invoked. Such interpolation is not desirable, since it inevitably introduces inaccuracies. For this reason, we introduce an additional FE mesh whose element vertices are constructed from the centre and boundary nodes of the FV mesh. Fig. 1 illustrates this idea. Note that construction of the FE mesh

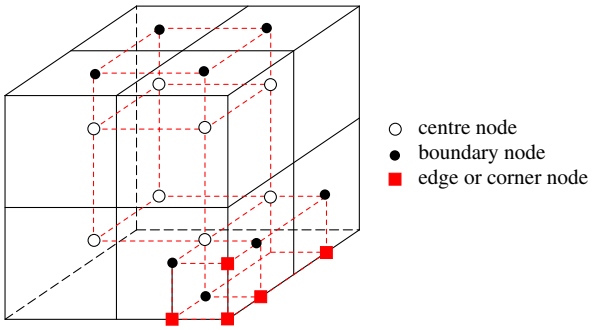

Fig. 1. Construction of the FE mesh by overlaying the FV grid. Schematic of grids, nodes and elements.

requires the introduction of a new type of node not present in the FV mesh. These new nodes are placed on the edges and corners of the external boundaries of the fluid domain.

Evidently, overlaying of the FV grid by the FE mesh increases the computer storage. In the present work, we overlay the FE mesh for the entire FV grid, which then doubles the computer storage requirements. To use computer resources more efficiently, the finite elements should be introduced only in the narrow band around the interface, where Eqs. (4) and (7) are solved. Currently, we are upgrading our level set implementation with a procedure for dynamic allocation of finite elements near the interface. The results of this work will be reported in a future communication.

3.2. Calculation of interface curvature

Here, we employ and compare two second-order methods for calculating $\nabla\phi$ and κ. In the first, which we refer to later in the text as Method A, we compute the values of the interface gradient and curvature at each node by computing the harmonic average of the contributions from neighboring elements. Consider the node i and the group of elements $K_j^i, j = 1, \ldots, N^e$, all sharing the same vertex i. Denote the element volumes as V_j. In each element K_j^i, we first calculate the gradient $\mathbf{n}_j$ at the centre of the element. The vectors $\mathbf{n}_j$ are then averaged to obtain the mean gradient $\langle \mathbf{n}^i \rangle$ at vertex i. Later, the nodal distribution of $\langle \mathbf{n}^i \rangle$ is used in an analogous manner to calculate the mean curvature $\langle \kappa^i \rangle$. Thus,

$$\langle \mathbf{n}^i \rangle = \left[\sum_{j=1}^{N^e} \frac{1}{V_j} \right]^{-1} \sum_{j=1}^{N^e} \frac{\mathbf{n}_j}{V_j}, \quad \langle \kappa^i \rangle = \left[\sum_{j=1}^{N^e} \frac{1}{V_j} \right]^{-1} \sum_{j=1}^{N^e} \frac{\nabla \cdot \langle \mathbf{n}_j \rangle}{V_j}. \tag{10}$$

An analogous method for calculating the normal to the interface has been used before by Sethian and Strain [10] in their Finite Difference (FD) level set model.

The second approach for calculating the interface gradient and curvature, which we refer to as Method B, is based on minimization of the following functionals in the least-squares sense:

$$I_g = \int_\Omega \frac{1}{2} (\mathbf{n} - \nabla\phi)^2 \, d\Omega, \quad I_c = \int_\Omega \frac{1}{2} \left(\kappa - \nabla \cdot \left(\frac{\mathbf{n}}{|\mathbf{n}|} \right) \right)^2 d\Omega, \tag{11}$$

which in turn reduces to the solving of two systems of linear equations. The gradient vector components and the curvature are approximated using bilinear shape functions.

4. Results and discussion

Due to computer power limitations, all examples presented in the following sections are two-dimensional problems, and some are axisymmetric. All the simulations have been carried out using structured grids, since these are the only type permitted by the CFX-4 code. However, both regular and non-regular grids are employed. The extension of the validation process to three dimensions will be described in a subsequent paper.

4.1. Filling a container with water

Fig. 2 is a schematic diagram of the container, which has a circular obstacle in the middle; the model is two-dimensional Cartesian. As shown in the figure, water enters the container from below with the uniform velocity profile $V_{\mathrm{FILL}} = 1$ cm/s. Initially, at $t = 0$, the water depth is 1 cm, the rest of the container being filled with air. The container is open at the top.

This particular problem configuration was chosen for three reasons. First, the shape of the container requires the use of a non-regular, body-fitted grid. Second, the water–air interface experiences separation and coalescence as it is flowing around the obstacle. Finally, the flow is slow enough not to generate noticeable waves on the interface, allowing comparison of the numerical solution with an exact solution obtained assuming that the interfacial surface stays horizontal at all times.

The initial velocity field is set to zero in both phases. A constant pressure boundary condition is imposed on the top of the container, through which the air is expelled as the water level rises. Free-slip boundary conditions are imposed around the obstacle and on the vertical walls. Due to symmetry, the problem is solved only for one half of the container.

To estimate the rate of convergence of the solution, simulations were performed on three grids with different resolution, characterized by the average cell size h: 1.19, 0.59, and 0.30 mm. The difference in solutions obtained for successive mesh resolutions is defined as follows:

$$E_1^i = \int_\Omega |H(\phi_{\mathrm{f}}(\mathbf{x},t)) - H(\phi_{\mathrm{c}}(\mathbf{x},t))|\,\mathrm{d}\Omega, \tag{12}$$

where, ϕ_{f} is the solution using a refined mesh, and ϕ_{c} is the solution from a coarser mesh. The error estimate given by Eq. (12) is well suited for problems involving interfacial flows. The results of the convergence study for the solution at $t = 3$ s, the time at which the free surface rises above the obstacle, are given in Table 1. Here, the average water elevation L_{W} is used as a target parameter. We have found that the level set method employing the mass-correction algorithm of Sussman et al. [14] results in a first-order convergence rate of the solution. In contrast, when using the homogeneous reinitialization equation (i.e. with $\lambda = 0$ in Eq. (7)), the solution converges at a second-order rate. As was pointed out earlier, the reason for the lack of convergence in the case when $\lambda \neq 0$ is the use of the non-regular grid with non-uniform cell size distribution. In our simulations, the ratio of the sizes of the largest to smallest cells crossed by the interface is about 3.5.

The fluid configurations and velocity field at different times obtained with $\lambda = 0$ are shown in Fig. 2. The free surface of the water stays practically flat during the entire filling process, as expected. Fig. 3(a) shows the average water elevation in the container as a function of time for the numerical and exact solutions; the agreement between them is excellent. Fig. 3(b) shows the deviation of the mass of water in the container from the exact value, $M_{\mathrm{exact}}(t)$, vs time. As one can see, the mass error is only a fraction of a percent, and it is smaller than in the simulation employing the mass correction procedure.

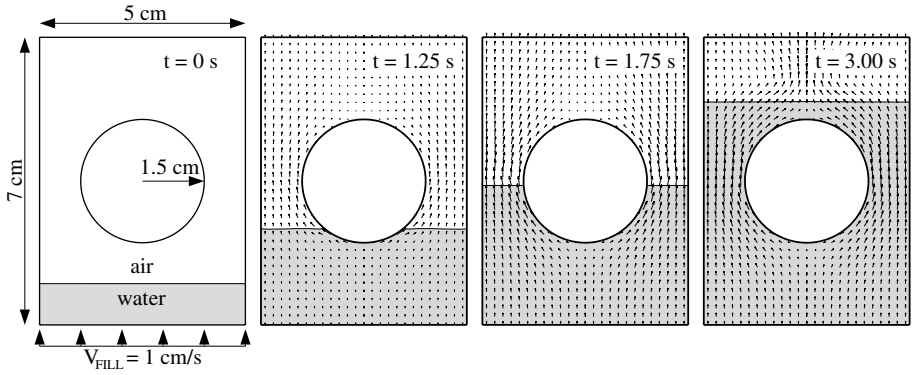

Fig. 2. Filling a two-dimensional container with water: schematic diagram of the container, velocity field and fluid configurations. The mesh is non-regular, structured, with the average size $h = 0.3$ mm. The velocity field shown here was obtained by projecting the calculated flow velocities on a regular grid with resolution 2 mm.

Table 1
Convergence study for the problem of filling a container with water

h (mm)	Without mass correction ($\lambda = 0$)		With mass correction	
	$E_1^i(3\text{ s})/w^2$	L_W (cm)	$E_1^i(3\text{ s})/w^2$	L_W (cm)
1.19	N/A	5.42	N/A	5.43
0.59	3.65×10^{-3}	5.42	3.10×10^{-3}	5.43
0.30	2.13×10^{-3}	5.41	2.33×10^{-3}	5.42

The exact solution at $t = 3$ s is $L_W = 5.41$ cm; $w(= 5$ cm) is the width of the container.

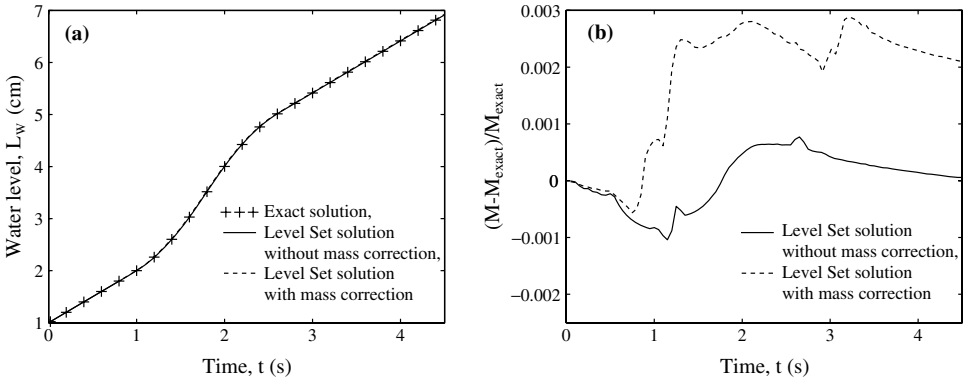

Fig. 3. Filling a two-dimensional container with water: (a) position of the free surface of water vs time; (b) mass of water vs time. Here, $h = 0.3$ mm.

4.2. Sloshing in an open tank

This application is based on the experiment of Bredmose et al. [28], consisting of a rectangular glass tank of size $1.48 \times 0.4 \times 0.75$ m (length × width × height), fixed on a support table, and shaken in a horizontal direction. Initially, the tank is motionless, and filled with water to a depth of $D_W = 155$ mm. Then, the support table is moved horizontally (in the direction of the largest side of the container) according to the driving signal shown in Fig. 4(a). In the experiment, the profiles of the water surface were recorded using a stationary video camera, and the hydrostatic pressure of the water on the left wall of the tank was measured with a pressure transducer.

In our model, the frame of reference is associated with the tank. Surface tension effects are neglected. Simulations were performed with the free-slip boundary condition on the tank walls for three regular grids of different resolution, the finest mesh size being $h = 4.96$ mm. We found from parameter studies that the

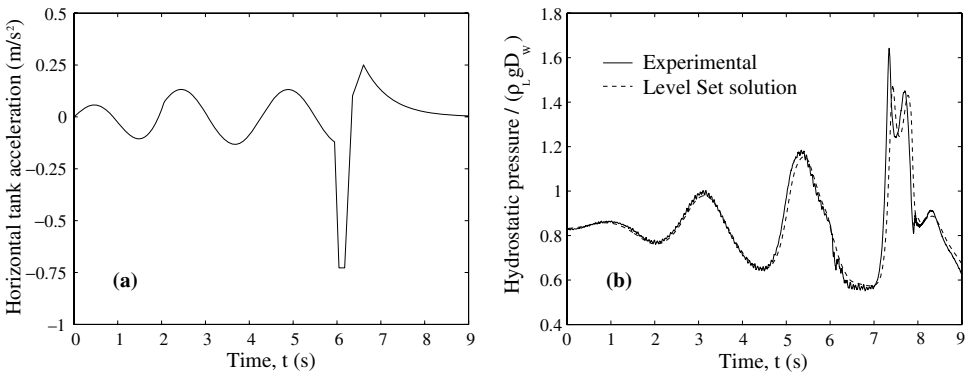

Fig. 4. Sloshing of water in the oscillating tank: (a) horizontal acceleration of the tank as a function of time; (2) pressure histories on the left vertical wall, 26 mm above the tank floor. Experimental data were obtained by Bredmose et al. [28].

difference between the free surface profiles obtained over 9 s of sloshing using the free-slip and no-slip boundary conditions on the tank walls was negligibly small. Evidently then, the friction on the walls has a negligible effect on the flow in this case. For details of implementing the no-slip boundary condition in the FE/FV level set model, see [23].

Fig. 4(b) shows the pressure evolution on the left wall of the tank, 26 mm above the floor. There is a slight phase shift between the experimental and numerical curves, but the overall agreement is very good. Fig. 5 gives the free-surface profiles. In this figure, the experimental profiles of the water surface are shown in greyscale, because the photographs taken during the experiment are not of sufficient resolution to determine the location of the free surface with higher accuracy. Nevertheless, we note good agreement between the numerical and experimental results.

Denoting the elevation of the free surface at the right wall of the tank as L_{RW}, and that at the left wall as L_{LW}, Table 2 gives the values of L_{RW} at $t = 6.52$ s, and L_{LW} at $t = 7.64$ s, as well as the values of E_1^i (Eq. (12)) obtained for different mesh resolutions. Overall, the results indicate that, in the absence of surface tension effects, the FE/FV level set method is second-order accurate. The mass of water grows with time monotonically; however, the gain of mass decreases as the mesh is refined. In the simulation for $h = 4.96$ mm, the gain of mass does not exceed 1%, and may be considered satisfactory for many industrial applications.

4.3. Zero-gravity drop oscillation

In this study, small oscillations of a two-dimensional drop of liquid around its equilibrium shape in an infinite, gaseous medium in the absence of gravity are computed; the drop is of radius a, and is oscillating at its fundamental mode. The analytical solution to the linearized problem has been found by Lamb [29], according to which the interfacial position of the drop, in polar coordinates (r, θ), is given by

$$\frac{r(\theta,t)}{a} = 1 + \tilde{\varepsilon}\cos(n\theta)\sin(\omega_n t)\exp(-t/\tau_n), \tag{13}$$

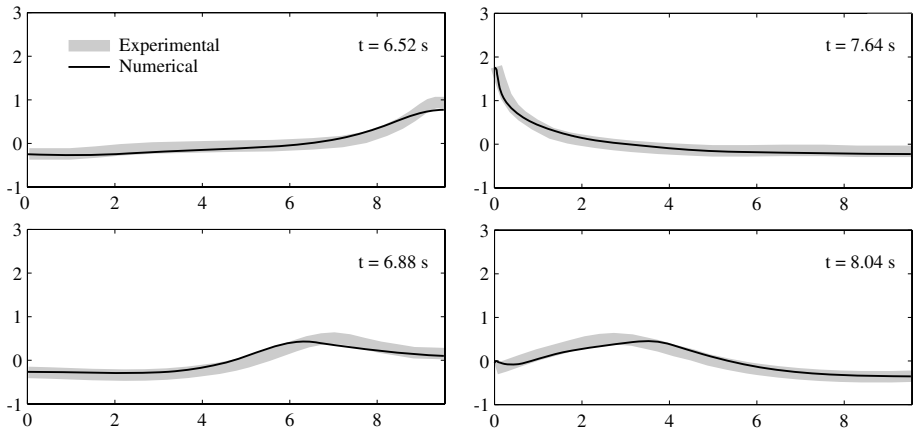

Fig. 5. Comparison of the experimental and numerical free-surface profiles for the problem of sloshing water in a tank. The tank dimensions are non-dimensionalized with the value of initial depth of water, D_{W}. Experimental data were obtained by Bredmose et al. [28].

Table 2
Convergence study for the problem of sloshing water in an open tank

h (mm)	$E_1^i(6.52\text{ s})/D_{\mathrm{W}}^2$	$L_{\mathrm{RW}}(6.52\text{ s})/D_{\mathrm{W}}$	$E_1^i(7.64\text{ s})/D_{\mathrm{W}}^2$	$L_{\mathrm{LW}}(7.64\text{ s})/D_{\mathrm{W}}$
19.84	N/A	0.74	N/A	1.32
9.92	1.08×10^{-1}	0.77	1.72×10^{-1}	1.71
4.96	3.22×10^{-2}	0.77	4.98×10^{-2}	1.76

$D_{\mathrm{W}}(= 155$ mm) is the initial water depth at $t = 0$.

where $\tilde{\varepsilon}$ is the initial perturbation, n is the number of the harmonic, ω_n is the frequency and τ_n is the characteristic time of decay due to viscous damping:

$$\omega_n^2 = \frac{n(n^2-1)\sigma}{a^3(\rho_L+\rho_G)}, \quad \tau_n = \frac{a^2}{2n}\frac{(\rho_L+\rho_G)}{((n-1)\mu_L+(n+1)\mu_G)}. \tag{14}$$

The subscripts L and G indicate the liquid and gas phases, respectively. The fundamental mode corresponds to $n = 2$.

We introduce dimensionless variables as follows:

$$\mathbf{x}^* = \mathbf{x}/l_c, \quad \mathbf{u}^* = \mathbf{u}/u_c, \quad t^* = t/t_c, \quad h^* = h/l_c, \quad \rho^* = \rho/\rho_L, \quad \mu^* = \mu/\mu_L, \tag{15}$$

where quantities in the denominators are the corresponding characteristic scales. Here and later, the asterisk superscript denotes dimensionless values. We set the velocity scale to $u_c = \sqrt{2\sigma/(a\rho_L)}$. The length and time scales are set to $l_c = a$ and $t_c = l_c/u_c$, respectively. Hence, the Reynolds and Weber numbers characterizing the problem are given by $Re = \rho_L u_c l_c/\mu_L$ and $We = \rho_L u_c^2 l_c/\sigma$. With the chosen scales, these values are $Re = \sqrt{2\sigma a\rho_L}/\mu_L$ and $We = 2$.

For the simulations, we choose parameters, configuration and initial conditions analogous to those adopted earlier by other groups [1,2,12]: $\rho_L/\rho_G = 100$, $\mu_L/\mu_G = 100$ and $Re = 20$. The interface $r(\theta,0)$ of the drop at $t = 0$ is initialized with $\tilde{\varepsilon} = 0.05$ and $n = 2$. Due to symmetry, the problem is solved in the half-domain $\Omega^* = \{(x^*,y^*)|\ 0 \leqslant x^* \leqslant 4,\ 0 \leqslant y^* \leqslant 2\}$, with the centre of gravity of the drop located at $(2,0)$. A free-slip boundary condition is imposed on the symmetry line, and the pressure is set to $p^* = 0$ on the open boundary. During scoping studies, we found that increasing the size of the domain Ω^* leads to negligibly small change in the solution.

Simulations are performed for the two methods of computing the interface gradient and curvature, designated earlier as Methods A and B. We have found that, in order to achieve stable convergence for this application, Method B requires a 2–3 times smaller time step and a larger value of the interface thickness ε compared to Method A. Here, we use $\varepsilon = 2h$ for Method A and $\varepsilon = 2.5h$ for Method B.

The evolution of the interface at $\theta = \pi/2$ obtained for a 200×100 grid and timestep $\Delta t^* = 0.007$ is compared against the analytical solution in Fig. 6(a). The frequency of the oscillations for the two numerical approaches is almost the same: $\omega_2^* = 1.68$, and this agrees well with the analytical solution, $\omega_2^{*AN} = 1.72$. Both numerical solutions display a slightly smaller rate of decay, τ^*, compared to the solution of the linearized problem, and in both numerical simulations there is a clear gradual shift of the equilibrium interface position to which the solution ultimately asymptotes. In the solution obtained using Method A, this shift is positive, whereas for Method B it is negative. Similarly, but less noticeably, this shift can be seen in the solutions of the axisymmetric oscillating-drop problem obtained using the ENO level set method by Sussman and his co-workers [13,14]. Fig. 6(b) shows the variation of the mass of the drop, M, as a function of time, for two mesh resolutions. Apparently, the mass conservation properties of Methods A and B are essentially the same. The mass increases linearly with time, but at a slower rate for the finer mesh.

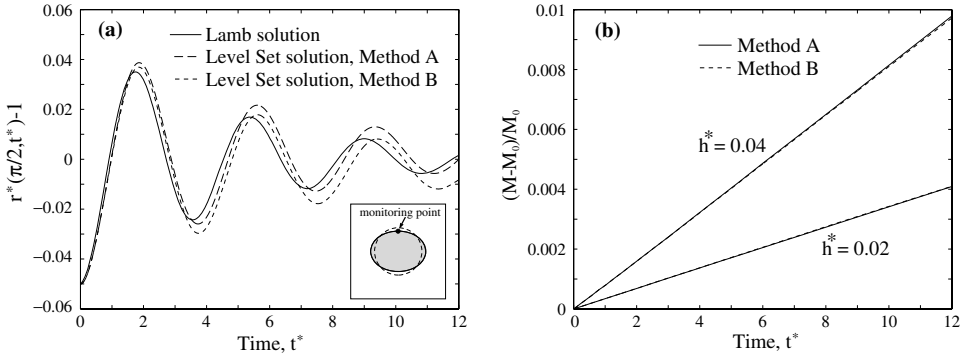

Fig. 6. 2-D drop oscillating in zero gravity: (a) time evolution of point $r^*(\frac{\pi}{2},t^*)$ on the interface (the inset is a schematic of the initial conditions); (b) mass of the drop as a function of time. Here, $\rho_L/\rho_G = 100$, $\mu_L/\mu_G = 100$, $Re = 20$, and $We = 2$.

Table 3
Convergence study for the problem of the oscillating drop in zero gravity: $\rho_L/\rho_G = 100$, $\mu_L/\mu_G = 100$, $Re = 20$, and $We = 2$

Method	h^*	E_1	E_∞	T_2^*
FE/FV level set, A	0.08	N/A	N/A	3.77
	0.04	5.72×10^{-3}	3.29×10^{-3}	3.74
	0.02	3.03×10^{-3}	1.71×10^{-3}	3.73
FE/FV level set, B	0.08	N/A	N/A	3.77
	0.04	6.37×10^{-3}	3.88×10^{-3}	3.74
	0.02	2.89×10^{-3}	1.83×10^{-3}	3.74

The convergence rate of the solution is examined using the relative errors E_1 and E_∞ calculated respectively from the traditional norms L_1 and L_∞:

$$E_1 = \int_0^{T_2^{*\mathrm{AN}}} |r_h^*(\pi/2, t^*) - r_{2h}^*(\pi/2, t^*)| \, \mathrm{d}t^*, \quad E_\infty = \max_{0 \leqslant t^* \leqslant T_2^{*\mathrm{AN}}} |r_h^*(\pi/2, t^*) - r_{2h}^*(\pi/2, t^*)|, \tag{16}$$

where $T_2^{*\mathrm{AN}} = 2\pi/\omega_2^{*\mathrm{AN}}$ is the period of oscillation predicted by the analytical solution. The results are given in Table 3. From the T_2^* values, it would seem that the solution obtained with the finest grid can be considered as converged, both Methods A and B displaying first-order convergence in norms L_1 and L_∞. This fact agrees with the results of Sussman and Puckett [17], who also obtained first-order convergence for the oscillating-drop problem, though formally their CLSVOF level set method is of higher accuracy. We believe that the development of strong parasitic currents (discussed below) is responsible for the observed shift of the equilibrium interface position, and the lack of convergence as compared to applications without surface tension effects, such as those presented in the previous sections.

4.4. Parasitic currents

It is well known that the performance of numerical interface-tracking methods applied to configurations close to equilibrium, and for which surface tension effects are significant, can suffer from the appearance of strong parasitic currents (e.g. [7]). These are unphysical flows developing around the interface as a result of a local imbalance of the pressure and stresses in the interfacial region, resulting from discretisation errors.

In order to estimate the intensity of the parasitic currents, we apply the level set formulation to simulate a static drop with the zero initial disturbance, $\tilde{\varepsilon} = 0$, and zero initial velocity field. Except for the value of $\tilde{\varepsilon}$, all the physical and numerical parameters are set identical to those used for the oscillating-drop problem. Theoretically, such a drop and the gas around it should remain motionless. However, soon after the start of the simulation, the parasitic flow develops. We characterize the parasitic flow by calculating its kinetic energy $E_{\mathrm{PC}}^* = \int_{\Omega^*} \frac{1}{2} \rho^* u^{*2} \, \mathrm{d}\Omega^*$ and the maximum velocity $u_{\max}^*$ observed in the computational domain.

We have found that Method B is more vulnerable to parasitic currents than Method A. In particular, if ε, the half-thickness of the interface, satisfies $\varepsilon \leqslant 2h$, the use of Method B results in fast growth of the intensity of the parasitic flow, which in turn leads to instability of the interface. However, increasing ε suppresses the parasitic currents. This can be seen in Fig. 7(a), which shows E_{PC}^* as a function of time for Method B using $\varepsilon = 2h$ and $\varepsilon = 2.5h$, and for Method A using $\varepsilon = 2h$. Fig. 7(b) gives $u_{\max}^*$ vs time for the two methods. As can be seen, Method A using $\varepsilon = 2h$ and Method B using $\varepsilon = 2.5h$ develop parasitic currents of approximately the same intensity, and show similar flow patterns (Figs. 7(c) and (d)). Except for minor aberrations next to the interface, both methods predict constant pressure inside the drop, $p_L^* = 0.496$, and $p_G^* = 0$ outside of the drop: the values do not vary with time. The predicted pressure jump across the interface agrees well (within 0.8%) with the exact value given by $\Delta p^* = 1/We$ (here, $We = 2$).

We have also performed simulations of the static drop in which the second-order approximations for the time derivatives and advective terms in the FV Navier–Stokes solver were replaced by the corresponding first-order schemes. Surprisingly, this resulted in negligible change in the solution (about 0.5% in the values of $u_{\max}^*$ and E_{PC}^*), indicating that the parasitic currents are very little affected by the order of discretisation employed in the FV flow solver.

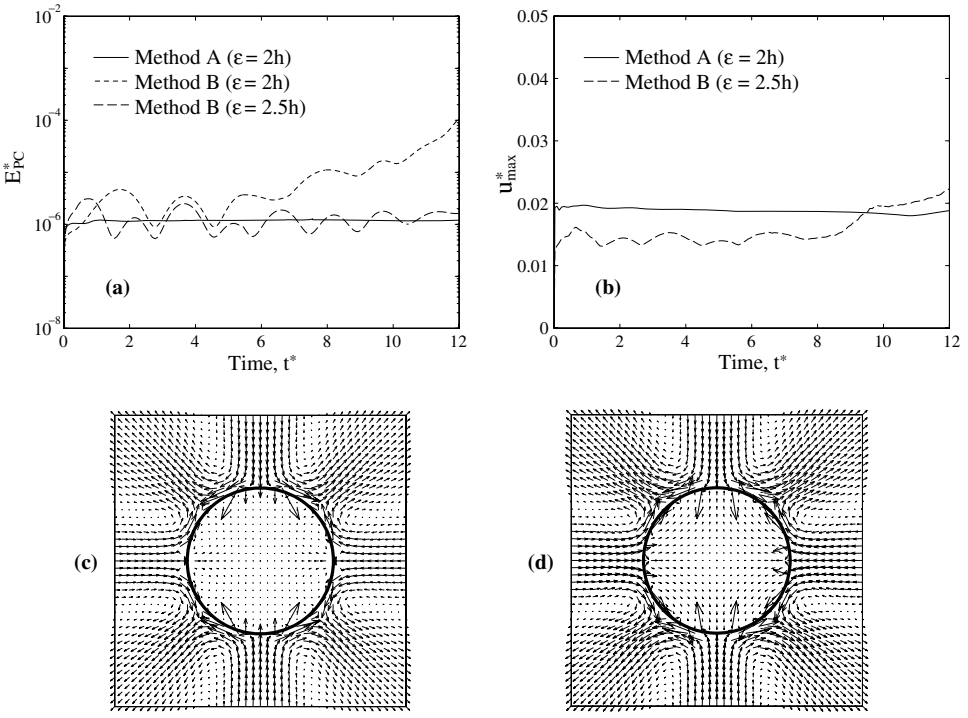

Fig. 7. Parasitic currents in the problem of the static drop: (a) kinetic energy E^*_{PC}; (b) maximum velocity u^*_{max}; (c) flow at $t^* = 12$ obtained using Method A; (d) flow at $t^* = 12$ obtained using Method B. The mesh is regular, with $h^* = 0.02$.

To estimate the effect of mesh resolution h on the intensity of the parasitic currents, we have carried out additional simulations of the static drop using a coarser, but still regular, grid with $h^* = 0.04$. Approximately the same results were obtained for Method A and Method B: $u^*_{max} \approx 0.03$ and $E^*_{PC} \approx 4 \times 10^{-6}$. Comparison of these data with the results obtained using $h^* = 0.02$ (Figs. 7(a) and (b)) indicates that the intensity of the parasitic currents is suppressed by refining the grid, and that the characteristic velocity of the parasitic flow appears to depend linearly on h.

Overall, reviewing the advanced interface-tracking methods in the literature, we note that the Lagrangian methods seem to suffer the least from parasitic currents. In particular, Tryggvason et al. [1] and Shin and Juric [2] reported values of u^*_{max} of $O(10^{-4})$ and $O(10^{-3})$, respectively, for the static-drop problem of a similar configuration to ours. According to Meier et al. [8] and Lafaurie et al. [7], the parasitic currents of the Piecewise Linear Interface Construction (PLIC) VOF method, which is a second-order accurate Eulerian algorithm, are well described by the empirical relationship $u_{max} \approx K\sigma/\mu$; although, the value of K (10^{-4} and 10^{-2}, respectively) was different for the two works. Noticing that the maximum velocities are typically observed in the cells containing the interface [7], and further assuming that on average a typical such cell is half full of fluid, we have $\mu = (\mu_L + \mu_G)/2$, and hence we can estimate that the use of the PLIC VOF method for our static-drop problem would give u^*_{max} in the range $2 \times 10^{-3} \leqslant u^*_{max} \leqslant 0.2$. Thus, we conclude that the magnitude of the parasitic velocities of the FE/FV level set method falls in the range manifested by the advanced VOF schemes.

4.5. Gas bubble rising in a viscous fluid

Modelling of a gas bubble rising in a viscous fluid under buoyancy is a useful validation test, since there is a considerable amount of experimental data on the subject in the literature. Here, we have used the data of Bhaga and Weber [30] and Hnat and Buckmaster [31], who studied terminal rise velocities and topologies of air bubbles in, respectively, sugar–water solutions and mineral oil. Table 4 gives the properties of the liquids and the volumes of the bubbles used in the experiments. The same values have been adopted for the present numerical simulation. In the table, the two bubble experiments are given labels b1 and b2. The fluid–gas

Table 4
Bubble volume, bubble rise velocity and properties of the liquids used in the experiments of Bhaga and Weber [30] and Hnat and Buckmaster [31]

Bubble	Experiment	Volume (cm^3)	Rise velocity, U_t^{exp} (cm/s)	Density, ρ_L (kg/m^3)	Viscosity, μ_L (kg/m s)	Surface tension, σ (N/m)
b1	Bhaga and Weber	9.3	26.0	1.35×10^3	2.73	0.078
b2	Hnat and Buckmaster	0.94	21.5	0.876×10^3	0.118	0.0322

density and viscosity ratios for bubbles b1 and b2 are correspondingly given as: $\rho_L/\rho_G = 1162$ and 714; $\mu_L/\mu_G = 6.93\times10^4$ and 6.67×10^3.

Two dimensionless groups are used to characterize the problem: the Morton number, $Mo = g\mu_L^4/\rho_L\sigma^3$, and the Eötvös number, $Eo = 4gr_e^2\rho_L/\sigma$. Depending on the values of Mo and Eo, the rising bubble can attain different terminal shapes, due to the interplay of the viscous, inertia and surface tension forces (e.g. [30]). The values of Mo and Eo for bubbles b1 and b2 are given, respectively, as: $Mo = 41.1$ and 0.065; $Eo = 116$ and 39.4.

Initially, at time $t = 0$, the bubble is modelled as a sphere of radius r_e, with the value of r_e calculated from the bubble volume (b1: 1.31 cm and b2: 0.61 cm). The initial velocity field is set to zero everywhere. The problem is axially symmetric, so in each case we model only half of the bubble. The computational domain is rectangular of size $5r_e \times 20r_e$ (width × height) in cylindrical coordinates (r, z), with the origin located at the bottom of the domain. At $t = 0$, the centres of the bubbles are placed at $z = 4r_e$. At the sides of the fluid domain, except for the symmetry line at $r = 0$, a far-field boundary condition is used, according to which the pressure distribution is set to $p = \rho_L g(20r_e - z)$.

Modelling of rising bubbles in liquids with such high fluid-gas density ratios necessitates the use of small time steps. As previously, we found that Method B for calculating the normal and curvature required smaller time steps than Method A. To avoid excessive CPU times, we have chosen Method A for this simulation.

Due to the limited computer resources available, convergence is studied by successively refining of the mesh size using a ratio of 2/3 (rather than 1/2), with the smallest mesh size being $h = 0.044r_e$. The terminal rise velocity of the bubble, U_t, was chosen as the target parameter. Results are given in Table 5. As can be seen, for the finest mesh, the difference between the numerical and experimental values of U_t is less than 1% for bubble b1, and is less than 3% for bubble b2. The convergence rate of the solution is close to the second order.

The transient development of the bubbles can be seen in Table 6 and Fig. 8. Both bubbles attain steady-state configurations very quickly: within 0.25 s. For comparison, Table 6 also gives the terminal bubble profiles observed experimentally: the numerical and experimental bubble shapes do indeed look quite similar. Note that the experimental images show only the side view of the bubbles, so that the characteristic indentation on the bottom surfaces cannot be seen. The terminal shape of bubble b1 belongs to the ellipsoidal cap type, whereas that of b2 is of the spherical cap type (the names of these topological configurations originate from the shapes of the bubbles that look very similar to segments cut from, respectively, an oblate spheroid and a sphere [30]). As can be seen from Fig. 8(b), the resulting loss of the bubble mass is ≈2%, the mass loss decreasing as the mesh is refined.

Denoting the height of the bubble as l_b and the width as w_b, the experimentally measured aspect ratios, w_b/l_b, of the fully developed bubbles b1 and b2 are 1.9 and 2.7, respectively. These are in good agreement with the values obtained numerically: $w_b/l_b = 1.8$ for bubble b1 and 2.6 for b2.

Table 5
Convergence study for the problem of a rising bubble in a viscous fluid

h/r_e	Bubble b1		Bubble b2	
	U_t (cm/s)	$E_1^i(0.25\text{ s})/r_e^2$	U_t (cm/s)	$E_1^i(0.25\text{ s})/r_e^2$
0.1	25.4	N/A	20.1	N/A
0.067	25.7	6.39×10^{-2}	20.6	4.71×10^{-1}
0.044	25.8	2.92×10^{-2}	21.0	2.53×10^{-1}

Table 6
Results of the simulation of rising bubbles b1 and b2

Bubble	$t = 0.05$ s	$t = 0.10$ s	$t = 0.15$ s	$t = 0.20$ s	$t = 0.25$ s	Exp.	Bubble type
b1							Ellipsoidal cap
b2							Spherical cap

Mesh size is $h = 0.044r_e$. Columns 2–6 show cross-sections of the bubble at different times, as computed using the FE/FV level set method. Column 7 shows the side view of the bubble as observed experimentally.

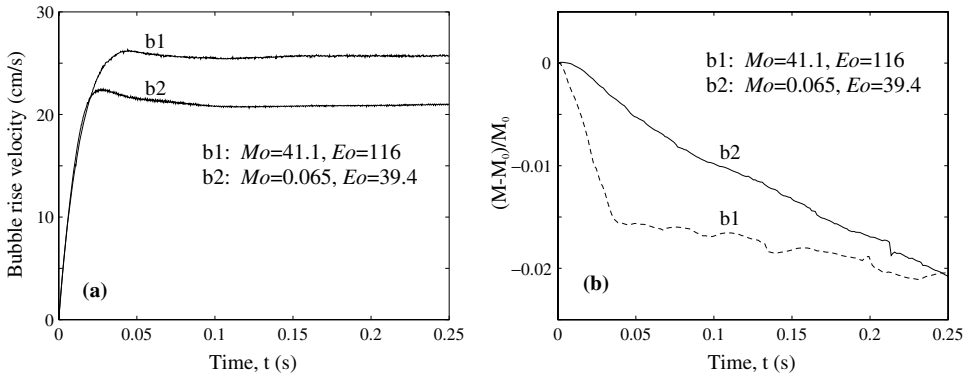

Fig. 8. Results of the simulation of the air bubble rising in a viscous fluid: (a) bubble rise velocity vs time; (b) bubble mass vs time.

In order to determine if the velocity boundary layer around the bubble surface is well resolved, we note that, according to the experimental results of Bhaga and Weber [30], the boundary layer thickness δ of ellipsoidal-cap bubbles is given by $\delta \approx w_b/4$. This corresponds to $\delta \approx 0.6r_e$ for bubble b1. To estimate the value of δ for bubble b2, we make use of the boundary-layer model developed by Moore [32] for spherical bubbles. Although shapes of spherical and spherical-cap bubbles are different, Moore's model accurately describes the flow over the upper surfaces of spherical-cap bubbles [30]. The boundary layer thickness δ has a minimum value at the stagnation point (uppermost point on the bubble) which, according to Moore [32], can be estimated as follows: $\delta \approx \sqrt{\mu_L R_{st}/(\rho_L U_t^{exp})}$, where R_{st} is the radius of curvature of the bubble surface. In turn, the value of R_{st} can be estimated using the semi-empirical model of Davies and Taylor [33] as follows: $R_{st} = 9(U_t^{exp})^2/(4g)$. For bubble b2, we obtain $\delta \approx 0.43r_e$. Hence, we conclude that the finest mesh resolution $h = 0.044r_e$ adequately resolves the velocity boundary layers in the numerical solution for both bubbles.

4.6. Gas bubble bursting through a free surface

The final application is of a gas bubble rising to and breaking through the free surface of a liquid. Once the top of the bubble reaches the free surface and opens up (see 3rd frame in Fig. 9), the pressure force acting on the lower part of the bubble becomes unbalanced, and a liquid jet is formed, which is accelerated through the hole in the free surface. This jet can subsequently break up into several drops. From a numerical point of view, this is a rather challenging problem, since the interface of the bubble experiences complex topological changes in the presence of strong surface-tension effects.

We consider the case where the bubble is initially located at a very close distance to the surface. This ensures that the rise velocity of the bubble at the time of breaking through the surface is rather small, so that surface tension effects are dominant over inertial effects. Development of the jet caused by small air bubbles bursting in water has been investigated by Boultone-Stone and Blake [3] using the boundary integral method, and by Duchemin et al. [34] using a Lagrangian marker-chain method. However, the simulations were terminated at the moment of jet break-up due to the difficulty in modelling the subsequent topological changes. Chessa and

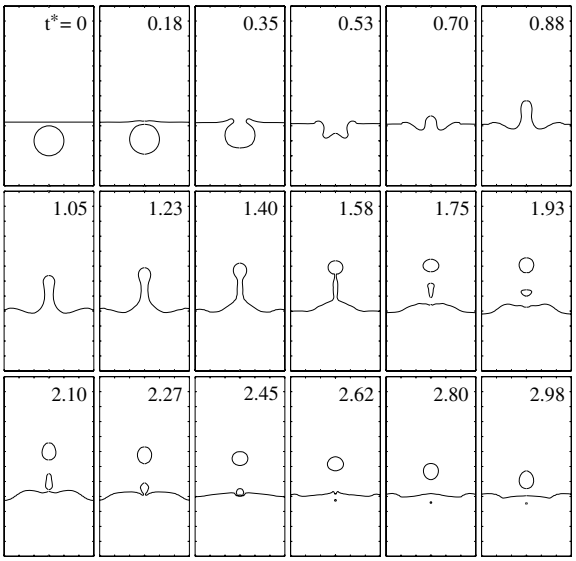

Fig. 9. Snapshots of the water jet resulting from a submerged air bubble bursting through a free surface: $\rho_L/\rho_G = 1000$, $\mu_L/\mu_G = 100$. The values shown for the time are dimensionless, with $t^* = t/t_c$ and $t_c = 0.042$ s.

Belytschko [18] modelled bursting bubbles using a FE level set method, but for relatively low density and viscosity ratios: $\rho_L/\rho_G \leqslant 100$ and $\mu_L/\mu_G \approx 1$. Sussman and Smereka [13] and Takahira et al. [19] were able to model a bursting air bubble in water before and after jet break-up using FD level set formulations.

In order to provide data for comparison, we apply our FE/FV level set model to the identical problem formulation and grid layout as that used by Sussman and Smereka [13], since the results of this work were reported in detail. Specifically, we consider an air bubble of radius $r_e = 5$ mm in a water bath. The surface tension of water is $\sigma = 0.072$ N/m. The water/air density and viscosity ratios are set to $\rho_L/\rho_G = 10^3$ and $\mu_L/\mu_G = 10^2$. The computational domain is defined as $\Omega = \{(r,z)|0 \leqslant r \leqslant 3r_e, 0 \leqslant z \leqslant 12r_e\}$ in cylindrical coordinates (r,z), with the origin located at the centre of the bath floor. Initially, the centre of the bubble is at $z = 3r_e$, with the water surface at the level $z = 4.15r_e$ (see the first frame in Fig. 9). The bubble is initially motionless. The computational mesh is 44×176 cells, corresponding to a cell width of $h = 0.068r_e$. Free-slip boundary conditions are imposed on walls. At the top of the domain, a constant pressure boundary condition is imposed. To our knowledge, there are no experimental data for such configurations.

In order to maintain consistency with [13], we non-dimensionalize the time t using the time scale $t_c = 0.042$ s; see Eq. (15). Fig. 9 gives snapshots of the solution at progressive times t^*. Only the solution obtained using Method B is provided here. The waves on the free surface, and the shape of the jet, look quite similar to those obtained by Sussman and Smereka [13], particularly prior to the moment of jet break-up. In our simulation, the jet break-up occurs at $t^*_{BR} = 1.6$. Sussman and Smereka did not explicitly report the value of t^*_{BR}, though a similar value to ours can be estimated from the snapshots of their solution. During the entire simulation, the variations of mass of water in the bath did not exceed 0.2% in our case.

Defining the jet velocity V_J as the velocity of the uppermost point on the jet, Sussman and Smereka [13] reported a maximum value of the jet velocity of $V_J^{max} = 1.6$ m/s. In our simulation, we obtained $V_J^{max} = 1.1$ m/s. As mentioned, the level set simulations are performed here with the time and space derivatives discretized using second-order approximations. Sussman and Smereka [13] employed third-order FD approximations for the advective terms. However, the observed discrepancy in the values of V_J^{max} is too large to be explained in terms of order of discretization schemes. This emphasizes the need for quality experimental data.

5. Conclusions

A new Finite-Element/Finite-Volume level set method for solving two-phase incompressible flows with moving interfacial boundaries has been presented. The method is based on a Finite Element treatment of

the level set equations which govern the motion of the interface and a Finite Volume solution of the Navier–Stokes equations. The latter are written in the conservative form, whereas the Finite Element method for level set equations employs a non-conservative formulation. This is different from the traditional level set approaches, in which the non-conservative form is adopted for all governing equations, but is more favourable in regard to implementation of the level set methods within the framework of existing commercial CFD codes, most of which employ conservative Finite Volume schemes.

The FE/FV level set method has been implemented into the commercial CFD code CFX-4, and validated against experimental and numerical data from sources involving two-phase flows with density ratios of the order of 10^2–10^3, and viscosity ratios as high as 7×10^4. Surface tension effects have been included in the model. The issue of parasitic currents has been addressed. The method has a convergence rate close to the second order, provided there are no strong parasitic currents, and displays good mass conservation properties, provided the mesh resolution is adequate for the problem under study.

Acknowledgments

The authors are sincerely grateful to Dr. H. Bredmose of the School of Mathematics, University of Bristol, UK for providing us the experimental data and photographs for the problem of sloshing water in a tank.

References

[1] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.
[2] S. Shin, D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, J. Comput. Phys. 180 (2002) 427–470.
[3] J.M. Boultone-Stone, J.R. Blake, Gas bubbles bursting at a free surface, J. Fluid Mech. 254 (1993) 437–466.
[4] T.Y. Hou, J.S. Lowengrub, M.J. Shelley, Boundary integral methods for multicomponent fluids and multiphase materials, J. Comput. Phys. 169 (2001) 302–323.
[5] V.E. Badalassi, H.D. Ceniceros, S. Banerjee, Computation of multiphase systems with phase field models, J. Comput. Phys. 190 (2003) 371–397.
[6] C.W. Hirt, B.D. Nichols, Volume of fluid method (VOF) for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[7] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, J. Comput. Phys. 113 (1994) 134–147.
[8] M. Meier, G. Yadigaroglu, B.L. Smith, A novel technique for including surface tension in PLIC-VOF methods, Eur. J. Mech. B/Fluids 21 (2002) 61–73.
[9] J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press, Cambridge, UK, 1999.
[10] J.A. Sethian, J. Strain, Crystal growth and dendritic solidification, J. Comput. Phys. 98 (1992) 231–253.
[11] W. Mulder, S. Osher, J.A. Sethian, Computing interface motion in compressible gas dynamics, J. Comput. Phys. 100 (1992) 209–228.
[12] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[13] M. Sussman, P. Smereka, Axisymmetric free boundary problems, J. Fluid Mech. 341 (1997) 269–294.
[14] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows, Comput. Fluids 27 (1998) 663–680.
[15] T.J. Barth, J.A. Sethian, Numerical schemes for the Hamilton–Jacobi and level set equations on triangulated domains, J. Comput. Phys. 145 (1998) 1–40.
[16] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[17] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301–337.
[18] J. Chessa, T. Belytschko, An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension, Int. J. Numer. Meth. Eng. 58 (2003) 2041–2064.
[19] H. Takahira, T. Horiuchi, S. Banerjee, An improved three-dimensional level set method for gas–liquid two-phase flows, J. Fluids Eng. 126 (2004) 578–585.
[20] A. Smolianski, Finite-element/level-set/operator-splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces, Int. J. Numer. Meth. Fluids 48 (2005) 231–269.
[21] X. Zheng, J. Lowengrub, A. Anderson, V. Cristini, Adaptive unstructured volume remeshing-II: application to two- and three-dimensional level-set simulations of multiphase flow, J. Comput. Phys. 208 (2005) 626–650.
[22] CFX-4, User documentation, AEA Technology, Harwell, Oxfordshire, OX11 0RA, UK, 2000.

[23] S.V. Shepel, B.L. Smith, S. Paolucci, Implementation of a level set interface tracking method in the FIDAP and CFX-4 codes, J. Fluids Eng. 127 (2005) 674–686.
[24] S.V. Shepel, B.L. Smith, Two surface tension formulations for the level set interface-tracking method, in: Proceedings of the 11th International Topical Meeting on Nuclear Reactor Thermal-Hydraulics (NURETH-11), Paper No. 371, Avignon, France, 2–6 October, 2005.
[25] S.V. Shepel, Numerical Simulation of Filling and Solidification of Permanent Mold Castings, PhD thesis, University of Notre Dame, Notre Dame, IN, USA, 2002.
[26] C.M. Rhie, W.L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J. 21 (1983) 1527–1532.
[27] J.P. Van Doormaal, G.D. Raithby, Enhancements of the SIMPLE method for predicting incompressible fluid flows, Numer. Heat Transfer 7 (1984) 147–163.
[28] H. Bredmose, M. Brocchini, D.H. Peregrine, L. Thais, Experimental investigation and numerical modelling of steep forced water waves, J. Fluid Mech. 490 (2003) 217–249.
[29] H. Lamb, Hydrodynamics, Dover, New York, USA, 1945.
[30] D. Bhaga, M.E. Weber, Bubbles in viscous liquids: shapes, wakes, and velocities, J. Fluid Mech. 105 (1981) 61–85.
[31] J.G. Hnat, J.D. Buckmaster, Spherical cap bubbles and skirt formation, Phys. Fluids 19 (1976) 182–194.
[32] D.W. Moore, The boundary layer on a spherical gas bubble, J. Fluid Mech. 16 (1963) 161–176.
[33] R.M. Davies, G. Taylor, The mechanics of large bubbles rising through extended liquids and through liquids in tubes, Proc. Roy. Soc. Lond. A 200 (1950) 375–390.
[34] L. Duchemin, S. Popinet, C. Josserand, S. Zaleski, Jet formation in bubbles bursting at a free surface, Phys. Fluids 14 (2002) 3000–3008.