INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **52**:1093–1134
Published online 19 April 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1217

A PLIC volume tracking method for the simulation of two-fluid flows

S. H. Garrioch and B. R. Baliga*,†

Department of Mechanical Engineering, McGill University, 817 Sherbrooke St. W., Montreal, Quebec, Canada H3A 2K6

SUMMARY

Numerical methodologies for computer simulations of two-fluid flows are presented. These methodologies fall into the category of volume tracking methods with piecewise-linear interface calculation (PLIC). The scope of this work is limited to laminar flows of immiscible, non-reacting, incompressible Newtonian fluids, without phase change, in planar two-dimensional geometries. The following novel or enhanced procedures are proposed: a parallelogram scheme for multidimensional advection of the volume-fraction field; a circle-fit technique for the orientation of the interface segments and the calculation of curvature; a novel contact angle treatment; and a staggered formulation for volumetric body forces that can accurately balance pressure forces in the vicinity of the interface. In addition, surface-tension-derived and hydrostatic-derived pressure adjustments are introduced as a means of accurately calculating pressure forces in cells that contain the interface, so as to minimize the non-physical flows that afflict many available volume tracking methods. The proposed method is validated using four test problems that involve simulations of pure advection, a static drop, an oscillating bubble, and a static meniscus. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: volume tracking; interface tracking; PLIC; VOF; finite volume

1. INTRODUCTION

Simulations of two-fluid flows is an active area of research in computational fluid dynamics, due to the prevalence of such flows and the modelling challenges that they pose. For example, droplet deposition, mould filling, sloshing of liquids in containers or tanks, immiscible oils floating on top of or in water, droplet and bubble formation and breakup, and liquid jets issuing into gaseous environments all involve two-fluid flows with distinct interfaces that may evolve with time, and all of these flows continue to be difficult to simulate accurately and

*Correspondence to: B. R. Baliga, Department of Mechanical Engineering, McGill University, 817 Sherbrooke St. W., Montreal, Quebec, Canada H3A 2K6.
†E-mail: bantwal.baliga@mcgill.ca

Contract/grant sponsor: Natural Sciences and Engineering Research Council of Canada

Received 8 September 2003
Revised 5 February 2006
Accepted 6 February 2006

efficiently. Numerical methods for the simulation of such flows can be categorized into two broad groups: in one, the computational mesh is deformed or adjusted to fit the interface between the fluids; and in the other, the mesh is kept fixed and a suitable technique is employed to deduce and track the location of the interface. For reviews and descriptions of several numerical methods for the simulation of two-fluid flows and interface tracking, and the associated terminology, the reader is referred, in chronological order, to the works of Harlow and Welch [1], DeBar [2], Noh and Woodward [3], Hirt and Nichols [4], Youngs [5], Hyman [6], Chorin [7], Fyfe *et al.* [8], Floryan and Rasmussen [9], Ashgriz and Poo [10], Unverdi and Tryggvason [11], Brackbill *et al.* [12], Sussman *et al.* [13], Tsai and Yue [14], Kothe [15], Pilliod and Puckett [16], Rudman [17], Mosso *et al.* [18], Rider and Kothe [19], Bugg *et al.* [20], Wu *et al.* [21], Scardovelli and Zaleski [22], Gao [23], and Gueyffier *et al.* [24], among others.

The focus in this work is on volume tracking methods formulated to work with fixed computational meshes. These methods are particularly well-suited for the modelling of two-fluid flows with gross interface deformations, such as folding and tearing. Rather than explicitly track the interface, volume tracking methods evolve fluid volumes by initializing and updating a volume-fraction field that identifies the fluid contained in each computational cell, and the interface is reconstructed from this volume-fraction field, as required. The reconstructed interface is not unique, as it depends on the reconstruction techniques. The goal of this paper is to present the formulations of several numerical techniques used in volume tracking methods for the simulation of two-fluid flows, including the effects of surface tension.

Most of the early volume tracking methods, such as the volume-of-fluid (VOF) method of Hirt and Nichols [4], used relatively low accuracy (essentially first-order accurate), piecewise-constant reconstruction techniques. DeBar [2] and Youngs [5] pioneered more accurate, piecewise-linear interface calculation (PLIC) techniques for volume tracking, in which line segments represent the reconstructed interface. In the 1990s, PLIC-based volume tracking methods grew in popularity as their capabilities were enhanced, and as more of the related research was published [10, 16, 19, 24].

While the latest PLIC methods are a great improvement over the piecewise-constant VOF and early PLIC methods, further improvements are needed. One key issue for PLIC methods is how best to reconstruct the interface from the volume-fraction field. It is essential that the interface reconstruction closely approximates the actual interface, because the interface topology directly influences volume-fraction advection and modelling of the surface tension forces. Rider and Kothe [19] describe several different approaches for performing this task, including the error minimization method of Pilliod and Puckett [16, 25] and a least-squares method based on the gradient of the volume fraction. The latter approach is of low accuracy, however, while the former approach is of better accuracy but only practical on two-dimensional orthogonal meshes [19]. In this work, a circle-fit technique is proposed and tested, and is demonstrated to be of similar accuracy to the error minimization method. In addition, this circle-fit technique can be used on non-orthogonal meshes.

Another issue is how to accurately update the volume-fraction field so as to match the evolution of the fluids. The discrete volume fractions must not be allowed to smear across many cells widths because that would increase the uncertainty in the location of the interface. Consequently, volume tracking methods use geometric advection schemes, rather than conventional algebraic advection formulas, to prevent numerical diffusion of the volume fractions [26, 27]. Early volume tracking methods moved the fluid volumes via a series of

one-dimensional fluxes, advecting the fluid first in the x-direction and then in the y-direction, for example, as in the methods of Noh and Woodward [3], Hirt and Nichols [4], and Lafaurie *et al.* [28]. This type of operator-split advection, however, can produce spurious asymmetries in the simulation and necessitate multiple interface reconstructions for each volume fraction update. Thus, a multidimensional technique is preferable and such a technique is proposed in this paper.

Another portion of this paper is devoted to an examination of the force balances across the interface. Due to the abrupt changes in density, pressure, and surface tension forces across the interface, numerical approaches akin to traditional finite volume discretizations of the conservation equations can lead to inaccurate numerical force balances, unless additional measures are taken to model them accurately. The consequences of such improper force balances are most evident at 'steady state' conditions, when the simulation exhibits non-physical flows and pressure gradients. Non-physical flows obtained in numerical simulations of physically static problems, called *parasitic currents* by Lafaurie *et al.* [28], have been noted in many papers on volume tracking [12, 15, 24, 29, 30], in the context of surface-tension dominated systems. These parasitic currents are produced by inaccuracies in both the surface tension calculations and the force balances at the interface. Additionally, as the simulation proceeds, the parasitic currents tend to grow in strength [15]. In this paper, new techniques are presented to improve the numerical approximation of force balances near the interface so as to reduce the parasitic currents. These techniques include the use of staggered body forces (to better balance the staggered pressure gradients) and pressure adjustments (to improve the accuracy of the numerically calculated pressure gradients).

The current work is limited to immiscible, non-reacting, constant-property Newtonian fluids in laminar, isothermal, two-fluid flows, without phase change. The proposed techniques are developed and tested for such two-fluid flows in planar two-dimensional geometries, on structured, rectangular grids. However, these techniques are designed to be extensible to two-dimensional axisymmetric meshes, as well as to unstructured grids. Indeed, they are developed with the idea of eventually implementing them in a control-volume finite element method (CVFEM) [31–33], which uses unstructured meshes for simulations in irregular geometries.

2. GOVERNING EQUATIONS

In this work, the one-field model assumption is invoked, whereby each fluid is assumed to move with the centre-of-mass of the aggregate fluid in a local control volume [15]. Under this assumption, the flow of both fluids is described by one velocity field and one pressure field. Thus, a single set of governing equations describes the flow of both fluids.

The stipulated flow is governed by the continuity and conservation of momentum equations, which may be written in Cartesian tensor notation as

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) + S_{g,i} + S_{\mathrm{st},i} \tag{2}$$

where t is the time, ρ is the density, u_i is a velocity component, x_i is the corresponding Cartesian coordinate, p is the pressure, μ is the dynamic viscosity, and i and j are tensor indices. $S_{g,i}$ is the volumetric gravitational force component equal to ρg_i, where g_i is a gravitational acceleration component. $S_{\mathrm{st},i}$ is a volumetric force component that approximates the effects of surface tension, as elaborated later in Section 2.1.

Volume tracking methods use a Heaviside unit function, f, hereafter referred to as an identity function, to describe the layout of the fluid regions [19]

$$f = f(x_1, x_2, t) = \begin{cases} 1 & \text{in Fluid } A \\ 0 & \text{in Fluid } B \end{cases} \tag{3}$$

As the identity function evolves with the movement of the fluids, it is governed by the following advection equation:

$$\frac{\partial f}{\partial t} + u_i \frac{\partial f}{\partial x_i} = 0 \tag{4}$$

The density and dynamic viscosity in Equation (2) are related to the individual fluid properties via the identity function, as follows:

$$\rho = f\rho_A + (1 - f)\rho_B; \quad \mu = f\mu_A + (1 - f)\mu_B \tag{5}$$

where the subscripts A and B refer to the two fluids.

2.1. Surface tension

At the boundary between two fluids, a surface tension force exists due to an imbalance of cohesive intermolecular forces at the interface [34]. The surface tension force can be expressed in terms of components that are normal and tangential to the interface, as follows [12, 35]:

$$s_{\mathrm{st},i} = \sigma\kappa n_i + \frac{\partial \sigma}{\partial x_i} \tag{6}$$

where σ is the surface tension coefficient, κ is the interface curvature, n_i is the i-direction component of the unit normal to the interface, and $\partial\sigma/\partial x_i$ is the corresponding component of the *surface* gradient of σ. Note that $s_{\mathrm{st},i}$ is a *surface* force. In this work, σ is assumed to be a constant, so the second term on the right-hand side of equation (6) is zero. Thus the surface tension force is simply

$$s_{\mathrm{st},i} = \sigma\kappa n_i \tag{7}$$

In volume tracking methods, rather than treat the interface as a boundary in the flow system, it is subsumed into the domain by using the one-field model. Therefore, incorporation of surface tension requires conversion of the surface force into a volume force. Based on the continuum surface force (CSF) model of Brackbill *et al.* [12], the volumetric surface tension force in Equation (2) is given by

$$S_{\mathrm{st},i} = s_{\mathrm{st},i}\delta_s \tag{8}$$

where δ_s is a surface delta function. Thus, the surface tension is effectively smeared in a small region about the interface; outside this region, $S_{\mathrm{st},i} = 0$. A fuller discussion of δ_s is presented

in Section 4.1. By substituting Equation (7) into Equation (8), the volumetric surface tension force is expressed as

$$S_{\mathrm{st},i} = \sigma \kappa \delta_s n_i \tag{9}$$

2.2. Initial and boundary conditions

Initial values are specified for velocity and the identity function (based on the known initial distribution of the two fluids). Initialization of pressure is not required for the two-fluid flows considered here, as the density of each of the two fluids is assumed to be constant. All of the walls are assumed to be rigid and impermeable, and the no-slip condition is applied at their interfaces with the fluids. In addition, advantage is taken of any symmetries and periodic behaviour in the flow, to reduce the size of the solution domain.

3. NUMERICAL FORMULATION OF THE PLIC VOLUME TRACKING METHOD

A summary of the proposed discretization of the governing equations and discussions of some novel techniques for the PLIC are presented in this section. The discretization is accomplished with a co-located finite volume method (FVM) [36–38] implemented, in this work, on an orthogonal rectilinear grid. A portion of this grid and related nomenclature are illustrated in Figure 1. The calculation domain is first divided into rectangular control volumes, or cells, and the grid points are placed at the geometric centres of the cells. Grid points are also placed

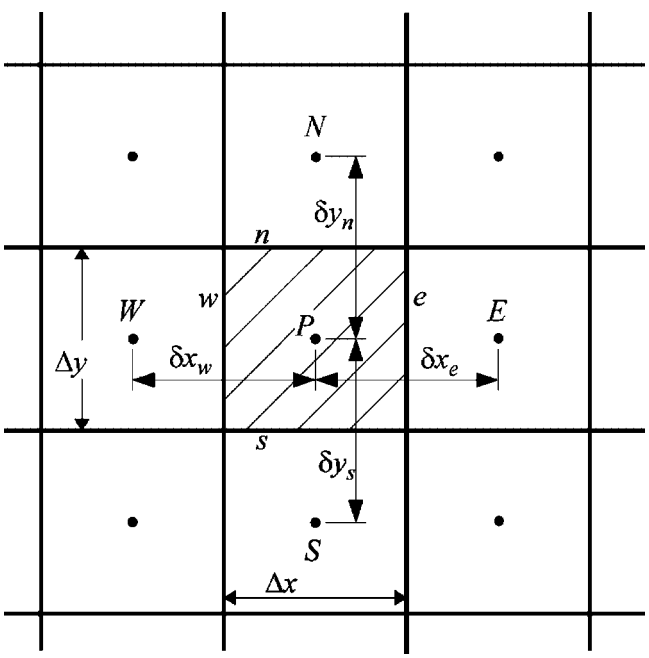

Figure 1. A portion of the grid and related nomenclature used in the FVM.

at the points where the grid lines cross the domain boundaries. All dependent variables are stored at the same grid points, or nodes, in the computational mesh.

3.1. Discretized momentum equations

With respect to the geographic node-labelling scheme given in Figure 1, the discretized x-momentum equation is expressed as follows:

$$a_P^u u_P = a_E^u u_E + a_W^u u_W + a_N^u u_N + a_S^u u_S + b^u \tag{10}$$

Following Patankar [36], the coefficients in this equation are defined as

$$\begin{aligned}
&a_E^u = D_e A(|Pe_e|) + \mathrm{Max}[-\dot{m}_e, 0]; \quad a_W^u = D_w A(|Pe_w|) + \mathrm{Max}[\dot{m}_w, 0] \\
&a_N^u = D_n A(|Pe_n|) + \mathrm{Max}[-\dot{m}_n, 0]; \quad a_S^u = D_s A(|Pe_s|) + \mathrm{Max}[\dot{m}_s, 0] \\
&a_P^{\mathrm{o}} = \frac{\rho^{\mathrm{o}} \Delta x \Delta y}{\Delta t}; \quad a_P^u = a_E^u + a_W^u + a_N^u + a_S^u + a_P^{\mathrm{o}} \\
&b^u = -\left\{ \frac{p_E - p_P}{\delta x_e} + \frac{p_P - p_W}{\delta x_w} \right\} \frac{\Delta x \Delta y}{2} + (M_x + S_x) \Delta x \Delta y + a_P^{\mathrm{o}} u_P^{\mathrm{o}}
\end{aligned} \tag{11}$$

A fully implicit formulation is used here, so all variables are at the new time level $(t + \Delta t)$ except for those with the superscript o, which are at the old time level (t). The designation $\mathrm{Max}[a, b]$ indicates that the largest of the bracketed quantities should be used. Let $\dot{m}_{\mathrm{fc}}$ denote the mass flow rate through the cell or control volume (CV) face fc, D_{fc} be the diffusion conductance for momentum across the cell face, and Pe_{fc} be the corresponding grid-related Peclet number. Then, at the east CV face, for example, the following relations apply [36]:

$$\dot{m}_e = \rho_e u_e \Delta y; \;\; D_e = (\mu_e \Delta y)/\delta x_e; \quad Pe_e = \dot{m}_e / D_e \tag{12}$$

The hybrid difference scheme was used to discretize the advection and diffusion terms at the CV faces. Thus, as discussed by Patankar [36], the following expression applies for the function $A(|Pe|)$ in Equation (11):

$$A(|Pe|) = \mathrm{Max}[0, -0.5|Pe|] \tag{13}$$

In Equation (11), S_x is the *total* x-direction body force per unit volume, and M_x is related to finite difference approximations of the partial derivatives of the viscosity, μ, and the velocity components, u and v, that appear in the x-momentum equation

$$M_x = \left\{ \begin{array}{l} \left(\dfrac{\mu_E - \mu_P}{\delta x_e} + \dfrac{\mu_P - \mu_W}{\delta x_w} \right) \left(\dfrac{u_E - u_P}{\delta x_e} + \dfrac{u_P - u_W}{\delta x_w} \right) \\ \quad + \left(\dfrac{\mu_N - \mu_P}{\delta y_n} + \dfrac{\mu_P - \mu_S}{\delta y_s} \right) \left(\dfrac{v_E - v_P}{\delta x_e} + \dfrac{v_P - v_W}{\delta x_w} \right) \end{array} \right\} \frac{\Delta x \Delta y}{4} \tag{14}$$

A discussion of the evaluation of the density at the CV faces is deferred until Section 3.3, as it is related to the advection of the volume fraction. The values of the viscosity at the CV faces are calculated using a local resistance analogy [36]. At the east CV face, for example,

the following equation applies:

$$\mu_e = \delta x_e \left[\frac{(\Delta x)_E/2}{\mu_E} + \frac{(\Delta x)_P/2}{\mu_P} \right]^{-1} \tag{15}$$

To avoid checkerboard pressure distributions in this co-located FVM, an adaptation of the momentum interpolation scheme [38] is used to obtain the values of the velocity that appear in mass flow rate expressions at the CV faces. First, the discretized momentum equations are rewritten as explicit expressions for the nodal values of the velocity components. For example, the x-momentum equation (10) is rewritten as follows:

$$\begin{aligned} u_P &= \hat{u}_P + \mathrm{d}u_P \left[\frac{1}{2} \left\{ \frac{p_E - p_P}{\delta x_e} + \frac{p_P - p_W}{\delta x_w} \right\} \right] \\ \hat{u}_P &= \frac{\sum_{nb} a^u_{nb} u_{nb} + (M_x + S_x)\Delta x \Delta y + a^{\circ}_P u^{\circ}_P}{a^u_P}; \quad \mathrm{d}u_P = \frac{\Delta x \Delta y}{a^u_P} \end{aligned} \tag{16}$$

In the calculation of the mass flow rate $\dot{m}_e$, u_e is approximated by linearly interpolating the $\hat{u}$ and du values from the P and E nodes, and by employing the pressure gradient that exists in the staggered CV between those nodes [38]

$$u_e \approx \hat{u}_e + \mathrm{d}u_e \left(\frac{p_E - p_P}{\delta x_e} \right) \tag{17}$$

The discretized y-momentum equation is similarly derived and cast in a form akin to that of Equation (10).

3.2. Discretized equations for pressure

The discretized equations for pressure are obtained by first integrating the continuity equation (1) over the control volume surrounding each node, discretizing these equations, and then substituting in them the expressions yielded by the momentum interpolation scheme for the velocity components at the CV faces. The resulting discretized pressure equation for the node P in Figure 1 can be cast in the following form:

$$\begin{aligned} a^P_P p_P &= a^P_E p_E + a^P_W p_W + a^P_N p_N + a^P_S p_S + b^P \\ a^P_E &= \mathrm{d}u_e \frac{\Delta y}{\delta x_e}; \quad a^P_W = \mathrm{d}u_w \frac{\Delta y}{\delta x_w}; \quad a^P_N = \mathrm{d}v_n \frac{\Delta x}{\delta y_n}; \quad a^P_S = \mathrm{d}v_s \frac{\Delta x}{\delta y_s} \\ a^P_P &= a^P_E + a^P_W + a^P_N + a^P_S; \quad b^P = (\hat{u}_w - \hat{u}_e)\Delta y + (\hat{v}_s - \hat{v}_n)\Delta x \end{aligned} \tag{18}$$

3.3. Discretized volume tracking equations

3.3.1. Description of the volume fractions. Information regarding the fluid contained in a cell can be obtained by integration of the identity function f defined in Equation (3)

$$F = \frac{1}{\Delta x \Delta y} \int_{\text{cell}} f \, \mathrm{d}x \, \mathrm{d}y \tag{19}$$

where F is the volume fraction of Fluid A in the cell. At the node within the cell, the density and the viscosity are related to this volume fraction as follows:

$$\rho = \rho_A F + \rho_B(1-F); \quad \mu = \mu_A F + \mu_B(1-F) \tag{20}$$

In PLIC volume tracking, a single line segment represents the interface in each cell that has F greater than zero and less than one. The equation for this line segment is given by

$$\vec{x} \,.\, \hat{n} = \beta \tag{21}$$

where $\vec{x}$ is a point on the line, $\hat{n}$ is the unit normal to the line, and β is the line constant [19].

3.3.2. Multidimensional advection of the volume fractions. In terms of the volume fraction, Equation (4) is written in conservative form as

$$\frac{\partial F}{\partial t} + \vec{\nabla} .(F\vec{u}) = 0 \tag{22}$$

Integrating over the cell associated with the node P in Figure 1 and over the time-step, Δt, the discretized advection equation for F is given by

$$(F^{n+1} - F^n)\Delta A + \{(F'_e u_e - F'_w u_w)\Delta y + (F'_n u_n - F'_s u_s)\Delta x\}\Delta t = 0 \tag{23}$$

where F^{n+1} is the volume fraction at the new time level, $t + \Delta t$, F^n is the volume fraction at the old time level, t, ΔA is the two-dimensional cell volume ($= \Delta x \Delta y$), and F' denotes volume fractions at the cell faces. The F' and velocities are all evaluated at the $n + \frac{1}{2}$ time level, $t + 0.5\Delta t$. Mosso *et al.* [39] found that this time-centred integration is necessary, because forward Euler integration is numerically expansive and backward Euler integration is contractive and, thus, both are unsatisfactory.

By rearranging Equation (23), the updated volume fraction is written in terms of the inflows and outflows as follows:

$$F^{n+1} = F^n - \left(\sum_{\text{fc}} Q_{\text{fc}} F'_{\text{fc}}\right) \Big/ \Delta A \tag{24}$$

where fc is a face of the current cell and Q_{fc} is the fluid volume *leaving* the cell through face fc (for inflows, Q_{fc} is negative). With respect to the notation in Figure 1

$$Q_e = u_e \Delta y \Delta t; \quad Q_w = -u_w \Delta y \Delta t; \quad Q_n = v_n \Delta x \Delta t; \quad Q_s = -v_s \Delta x \Delta t \tag{25}$$

For the fluid flows of interest, the *net* outflow from a cell is zero. Thus,

$$Q_e + Q_w + Q_n + Q_s = 0 \tag{26}$$

In the volume fraction update Equation (24), the F'_{fc} terms are unknown. As discussed in the introduction, volume tracking methods calculate these terms geometrically to prevent numerical diffusion of the F field. For example, in the operator-split approach, in which unidirectional fluxes are used, a rectangular *swept volume* is assumed to contain the fluid transported out through the cell face. Note that this region is generically referred to as a 'volume' even though it is an area in two-dimensions: thus, implicitly, a unit depth is implied. The contents of a swept volume are determined from the position of the interface, as discussed by Rider and Kothe [19].

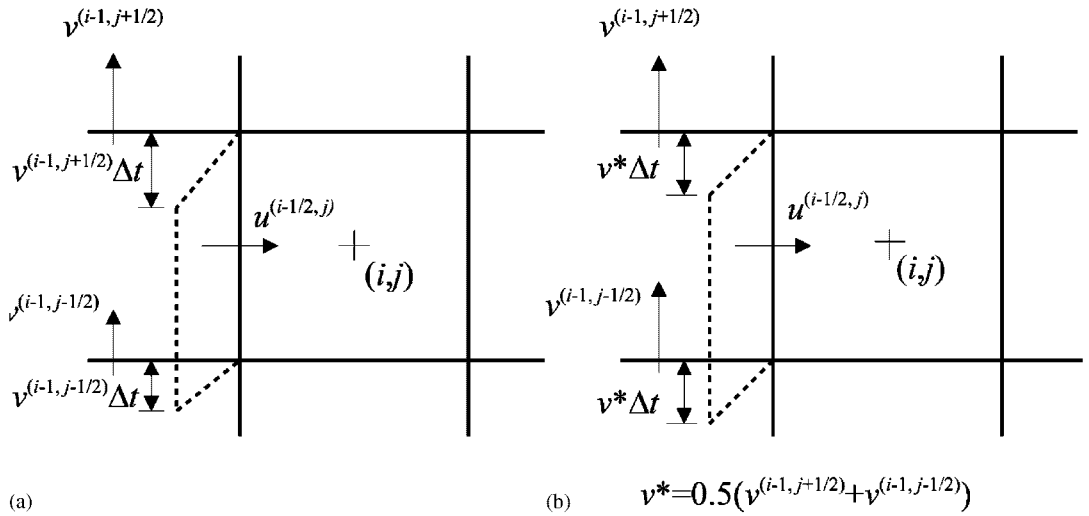

Figure 2. Swept volumes: (a) trapezoid; and (b) parallelogram.

Operator-split advection, however, can produce spurious asymmetries in the simulations. Thus, a multidimensional technique is preferred. Unfortunately, determination of the F'_{fc} is more complex in multidimensional advection. Pilliod and Puckett [16, 25, 40] have developed a second-order, but seemingly cumbersome, multidimensional advection technique. Rider and Kothe [19] use a simpler, related technique in which trapezoidal swept volumes are used to determine the fluxes between cells. As shown in Figure 2(a), the swept volumes in this approach are trapezoids generated by tracing back from the end points of the cell face, based on the velocities at local cell faces. In general, the trapezoid technique is more accurate than the operator-split approach and, because it requires only a single reconstruction and advection to update the F field, is more efficient.

The trapezoid technique, however, experiences problems with mass conservation. These problems can be manifested as an overshoot or undershoot in the volume fraction: that is, an F value greater than one or less than zero. One source of these mass gains (or losses) is an imbalance of inflows and outflows produced by the trapezoidal swept volumes. For example, cell (i,j) in Figure 2(a) experiences an inflow at its west face. The size of this inflow should be equal to $u^{(i-1/2,j)}\Delta y\Delta t$, as given by $-Q_w$ in Equation (25), where $u^{(i-1/2,j)}$ is u_w. This is not the case, however, since the influence of the transverse velocities $v^{(i-1,j+1/2)}$ and $v^{(i-1,j-1/2)}$ alters the size of the inflow and, thereby, the amount of fluid entering cell (i,j). Thus, these trapezoidal inflows and outflows cannot be expected to satisfy Equation (26). Rider and Kothe [19] use a divergence correction term and volume fraction redistribution to reduce the effects of these flow inequalities, but these measures do not address the underlying problem.

To eliminate the flow imbalance, the use of parallelogram swept volumes is proposed in this work. Rather than use an individual transverse velocity to dictate the position of each side of the swept volume, an *average* transverse velocity, v^*, dictates the shape of the parallelogram, as shown in Figure 2(b). The area of a parallelogram is determined by the lengths of its parallel sides and by the distances between those sides, and is independent of its skew.

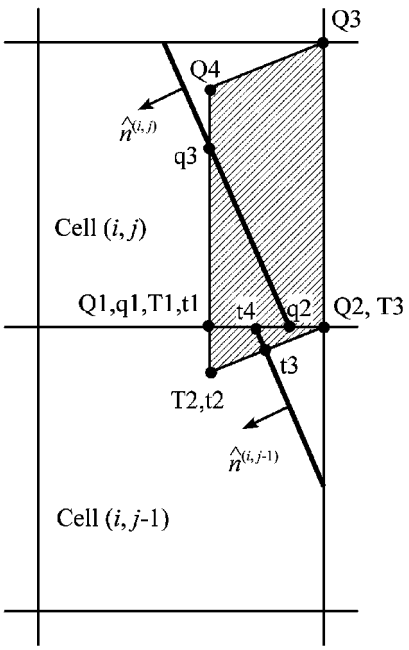

Figure 3. The polygons (Q, q, T, t) involved in the calculation of F'_e for cell (i, j).

Thus, the area of a parallelogram swept volume is equal to the area of a rectangular swept volume, as described in Equation (25), such that Equation (26) is satisfied. Consequently, with parallelogram swept volumes, the net outflow from a cell is zero which eliminates the aforementioned mass imbalance.

The volume fraction calculation for the parallelogram swept volume is similar to the calculation for the rectangular swept volume. Consider, for example, an outflow through the east face of cell (i, j), represented by the shaded swept volume in Figure 3. The volume fraction for this swept volume is given by

$$F'_e = (A_q + A_t)/(A_Q + A_T) \tag{27}$$

where A_q, A_t, A_Q, and A_T are the areas of the polygons $q1$–$q2$–$q3$, $t1$–$t2$–$t3$–$t4$, $Q1$–$Q2$–$Q3$–$Q4$, and $T1$–$T2$–$T3$, respectively. The average material density fluxed across the cell face is calculated as follows: $\rho_e = \rho_A F'_e + \rho_B(1 - F'_e)$. Such cell-face average densities are needed for calculating the coefficients in the discretized momentum equations; see, for example, Equations (10)–(12).

While the use of parallelogram swept volumes ensures volume conservation, conservation of mass can still be violated if the outflows remove from the cell more of Fluid A or B than is available. Specifically, because the swept volumes do not exactly abut one another, an outflow can remove fluid that neither resides in the current cell nor enters the cell, as shown by the overhang regions in Figure 4. Also, within a cell, the outflows can overlap, which represents a double claim on the underlying fluid (Figure 4). These phenomena can produce volume fractions greater than one or less than zero if an excess of Fluid A or B is taken by

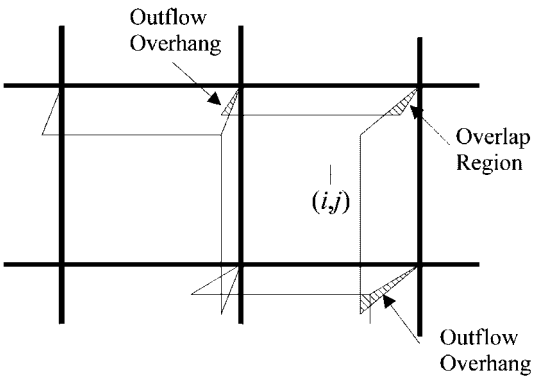

Figure 4. Potential problem regions for the outflows at the top and right faces of cell (i, j).

the outflows. It is common practice in volume tracking methods to simply reset these values to eliminate the overshoots and undershoots, but this effectively destroys or creates mass [4, 40]. Another approach is to redistribute fluid to neighbouring cells which, while conservative, is inaccurate [35].

In this work, the F'_{fc} values are iteratively scaled, if necessary, so that the amounts of Fluids A and B taken from a cell cannot exceed the amounts entering and residing in the cell [41]. This adjustment procedure is performed on each cell that contains the interface and on each cell that borders an interface cell. As any adjustment to the outflows affects multiple cells, in general, the entire procedure is repeated until no more modifications are required. With this procedure, mass conservation is strictly enforced. Thus, the advection technique presented here combines the accuracy of flow-oriented swept volumes with a strict adherence to mass conservation.

3.4. Piecewise-linear interface calculation

In PLIC volume tracking methods, a line segment defined by Equation (21) represents the interface in each interface cell. These methods are differentiated by the procedures they use to calculate the segment orientation, represented by $\hat{n}$, and the segment position, given by the line constant β. In this section, the details of the interface reconstruction are discussed.

3.4.1. Interface segment orientation: calculation of $\hat{n}$. As line segments cannot provide an exact representation of an arbitrary interface, a unique piecewise-linear representation does not exist in general. The reconstruction must, however, closely approximate the actual interface for accurate volume tracking, because the interface topology directly influences volume fraction advection and the surface tension forces. Thus, an accurate $\hat{n}$ calculation is critical to the success of the volume tracking. In this work, it was decided that an acceptable method should be at least second-order accurate. According to a criterion established by Pilliod and Puckett [25, 16], a second-order reconstruction technique can exactly represent an arbitrarily-oriented linear interface, that is, a linear interface which is not parallel to the mesh lines.

A circle-fit technique [41, 42] that satisfies this criterion was developed in this work. Drawing on earlier work by Chorin [7] and Mosso *et al.* [39], this technique uses circles

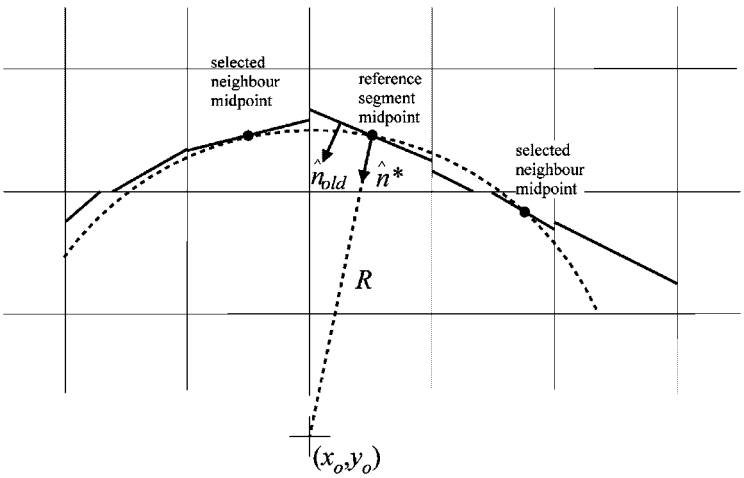

Figure 5. Illustration of the circlefit.

to orient the interface segments. With respect to each reference segment, the circle is fitted through its midpoint and the midpoints of two of its neighbours, after selecting the two neighbour segments from among those available. This circle fit provides not only a new orientation for the reference segment but also the curvature of the segment. An example of the circle-fit is shown in Figure 5.

An iterative algorithm for adjusting the reference segment orientation was designed as follows:

1. Set initial $\hat{n}$ for all interface segments using a least-squares method, such that $\hat{n} = \vec{\nabla} F / |\vec{\nabla} F|$, as discussed by Rider and Kothe [19].
2. Set the iteration counter to 1.
3. Consider an interface cell, whose segment has an orientation of $\hat{n}_{\text{old}}$, and its eight surrounding neighbour cells in a 3×3 cell block. This central segment is referred to as the reference segment.
4. Select one neighbour segment on each side of the reference segment for use in the circle-fit (see details in Section 3.4.2).
5. The three midpoints of the reference and neighbour segments uniquely define a circle. A new unit normal for the reference segment, $\hat{n}^*$, lies on the line from the midpoint of the reference segment to the centre of the circle. In this work, $\hat{n}$ always points into Fluid A, which may be inside or outside the circle in the reference cell. Thus, the proper $\hat{n}^*$ (pointing to or away from the centre of the circle) is chosen by maximizing its dot product with $\hat{n}_{\text{old}}$.
6. Use a modified *regula falsi* algorithm [43] to calculate $\hat{n}_{\text{new}}$ based on $\hat{n}^*$ and bracketing values of $\hat{n}$ (it was found that this algorithm works very well, but there may be other equally suitable or more efficient algorithms for this task).
7. Store $\hat{n}^*$ as a bracketing value, if it improves over existing brackets of the *regula falsi* algorithm.

8. Calculate the curvature for the reference segment as the inverse of the circle radius.
9. Return to Step 3 for any unadjusted segments at this iteration level.
10. Update each interface segment, by setting $\hat{n} = \hat{n}_{\text{new}}$ and by recalculating β.
11. Increment the iteration counter. If any $\hat{n}$ changed in direction by more than a prescribed tolerance, return to Step 3 and sweep through each interface cell again.

This circle-fit technique, like the method of Mosso *et al.* [39], is second-order accurate. Generally four or five iterations are needed before the segment orientations change by 0.001° or less, at which point the iterations are stopped. As segment orientation is tied directly to the location of neighbouring segments, rather than to the underlying volume fraction field, this circle-fit technique tends to keep the reconstructed interface smooth.

3.4.2. Neighbour segment selection. The proposed procedure for the selection of neighbour segments will now be described; for details of the reasoning behind this procedure, the interested reader is referred to References [41, 42]. In general, it is preferable to select those neighbour segments that are closer to the reference segment, so that the circle more nearly approximates the local interface shape. Neighbour selection is thus confined to the eight neighbour cells surrounding the reference cell in the 3×3 cell block.

The most appropriate neighbour segments should also provide a smooth transition to the reference segment, as will now be described. Consider Figure 6 which shows two potential neighbours, labelled 1 and 2, one of which must be selected for a circle fit with the reference segment, labelled r. To measure the smoothness of the transition, a connector line is drawn between the midpoints of the reference segment and its potential neighbour, as illustrated in Figure 6. The angle between the connector line and the reference segment is given by

$$\alpha_i = \cos^{-1}(\hat{n}_{c,i} \,.\, \hat{n}_r) \tag{28}$$

while the angle between the connector line and the neighbour segment is given by

$$\Omega_i = \cos^{-1}(\hat{n}_{c,i} \,.\, \hat{n}_i) \tag{29}$$

where $\hat{n}_{c,i}$ is the unit normal to the connector line to neighbour i, $\hat{n}_r$ is the unit normal to the reference segment, and $\hat{n}_i$ is the unit normal to neighbour segment i. These angles are summed and denoted by Γ as follows:

$$\Gamma_i = \alpha_i + \Omega_i \tag{30}$$

where Γ_i is a measure of the transition smoothness from the reference segment to neighbour segment i. Generally, a neighbour segment is a better candidate for selection if it produces a lower Γ and thus, in Figure 6, neighbour 1 should be selected over neighbour 2.

It is also better to avoid picking those neighbours which reside in cells with F close to 1 or 0, as they are more likely to be influenced by numerical inaccuracies. Expressed formally, the selection process should favour neighbours with F closer to one-half, which can be measured as

$$F_{\text{test}} = \min(F, 1 - F) \tag{31}$$

Neighbours with higher F_{test} are preferred.

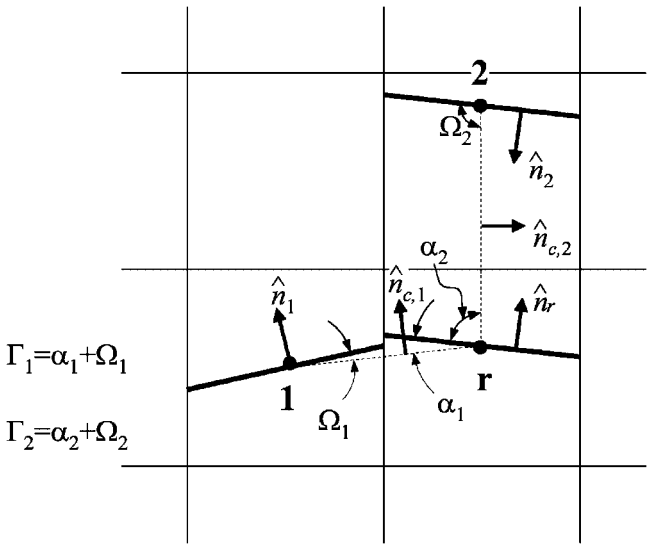

Figure 6. Example of the connector line angles used to select appropriate neighbours.

For the most accurate neighbour selection, the two selection criteria, based on a smooth segment transition and F_{test}, should be combined. Through numerous preliminary numerical experiments, a good combination was found to be a minimization of $(\Gamma/\sqrt{F_{\text{test}}})$.

Thus, selection of the two neighbours is performed as follows: the neighbour segments are divided into left and right neighbours based on their positions relative to a normal line passing through the midpoint of the reference segment. Then, on each side, the segment with the smallest value of $(\Gamma/\sqrt{F_{\text{test}}})$ is selected. To prevent neighbour selection flipping, whereby the neighbour selection continuously alternates between two or more segments and, therefore, promotes instability, a tolerance is applied to $(\Gamma/\sqrt{F_{\text{test}}})$. A tolerance of 6° was found to work well.

3.4.3. Interface segment position: *Calculation of* β. With $\hat{n}$ known, two related tasks are frequently performed: given β, calculate F; and given F, calculate β. The first task requires a single calculation of the fractional cell area cut by the interface segment, while the second task uses the same calculation approach but normally requires a few iterations. The details are available in the work of Rider and Kothe [19].

3.5. Solution of the coupled discretized equations

An outline of the overall iterative solution procedure at each time level is presented below:

1. Calculate the surface tension forces (details are given in Section 4).
2. Solve the discretized x-momentum and y-momentum equations.
3. Solve the discretized pressure equations.
4. Calculate the cell-face velocities using the momentum interpolation scheme.
5. Advect the F field (Equation (24)), reconstruct the interface, and calculate curvatures.

6. Calculate ρ and μ based on the new F field (Equation (20)).
7. If the u, v, and p fields are not converged or mass is not conserved, return to Step 1.

This overall iterative solution procedure is an adaptation of the simultaneous variable adjustment (SIVA) procedure [32, 33], which is based on ideas borrowed from SIMPLER [36]. In Step 2 of this procedure, the discretized x- and y-momentum equations (linearized and decoupled) are solved sequentially using a line-by-line tridiagonal matrix algorithm [36]. However, a more sophisticated iterative solver is needed for an efficient solution of the discretized pressure equations in Step 3: The solver used in this work is a multigrid algorithm with a Gauss–Seidel smoother, both adapted from the work of Ferziger and Peric [37].

4. NUMERICAL MODELLING OF SURFACE TENSION FORCES

4.1. Surface tension force

The derivation of the volumetric surface tension force was presented in Section 2.1. This volumetric force, given by Equation (9), is repeated below

$$S_{\mathrm{st},i} = \sigma\kappa\delta_s n_i \tag{32}$$

The surface tension coefficient, σ, is a property of the two fluids involved, while n_i is determined from the segment orientation as discussed in Section 3.4.1. Several different forms of κ and δ_s are used in volume tracking methods.

One common way of calculating the curvature is as follows [12]:

$$\kappa = -\vec{\nabla} \cdot \hat{n} \tag{33}$$

In this approach, $\hat{n}$ is calculated using the gradient of F. Thus, the above equation effectively involves second derivatives of F [22, 44]. This is problematic since the abrupt change in the volume fractions across the interface generally makes it difficult or impossible to accurately support the second derivative stencils required to evaluate Equation (33). Thus, a smoothed F field is usually employed to solve for κ using Equation (33) [30, 45]. In addition, the curvatures and surface tension forces are sometimes directly smoothed, to eliminate noise and spurious values [30, 45].

In this work, the use of Equation (33) without any smoothing was found to produce physically untenable results in surface-tension dominated flows. Smoothing the F field produces better results, but it obscures fine interface details. For example, at a region of high curvature, a large surface tension force should be generated that will tend to decrease the interface curvature. With smoothing, however, the high curvature may be hidden or smeared so that the associated large reaction force never materializes. Thus, smoothing can lead to a fractured interface, in which the interface segments are disjointed. In addition, the interface topology may be altered by the smoothing unless the grid resolution significantly exceeds the size of the smoothing filter.

Consequently, an alternate approach was chosen in this work: the curvature is calculated from the circle fit used to orient the interface segments, as suggested by Chorin [7]. The radius of the circle is assumed to be equal to the principal radius of curvature of the interface,

at the midpoint of the interface segment, from which the curvature is defined as [34]

$$\kappa = 1/R \tag{34}$$

With this approach, the curvature is obtained from the circle-fit technique with almost no additional effort. Furthermore, because no smoothing is involved, the connections between neighbouring interface segments are strong such that, if one segment were to move out of step with its neighbours, the resulting large curvatures and surface tension forces would work to reduce the gaps between the segments. The proper sense of the curvature must be explicitly assigned based on which side of the interface the circle fit lies. If the circle lies on the Fluid *A* side, the curvature is positive; if it lies on the Fluid *B* side, the curvature is negative, and a negative sign must be added to the right side of Equation (34).

The surface delta function in Equation (32) is responsible for smearing the surface tension force in a compact region about the interface, thereby creating a volumetric force. In this work, the surface delta function is chosen as

$$\delta_s = l/\Delta A \tag{35}$$

where l is the segment length and ΔA is the cell area. Thus, the expression for the volumetric surface tension force is given by

$$S_{\mathrm{st},i} = \sigma\kappa(l/\Delta A)n_i \tag{36}$$

Note that surface tension forces exist only in cells that contain the interface.

4.2. Contact angle treatment

In this work, a constant contact angle, θ_c, was imposed at the contact line formed by the intersection of the two fluids and a solid surface. This approach neglects the fact that the contact angle changes with the flow conditions [44], but it produces stable simulations and at least qualitatively accurate surface tension forces. For a discussion of other approaches, and their advantages and drawbacks, the interested reader is referred to Zhuang *et al.* [46].

Before the contact angle can be imposed, the contact line must be located in the domain. This is not an insignificant task since when the interface passes through a cell adjacent to the wall, referred to here as a *near-wall cell*, it is not known *a priori* whether or not the interface actually makes contact with the wall in that particular cell. Two simple rules are proposed and used here to identify those near-wall cells that contain the contact line:

- If $\theta_c \leqslant 90^\circ$ (wetting), the contact line exists in a non-empty, near-wall cell which is adjacent to an empty near-wall cell;
- If $\theta_c > 90^\circ$ (non-wetting), the contact line exists in a non-full, near-wall cell which is adjacent to a full near-wall cell.

Once the contact-line cell is known, the question is how to impose the contact angle. One possibility is to impose θ_c as the new orientation of the interface segment in the contact-line cell. This approach, however, was found to be overly severe and intrusive, especially with coarse grids. Since the surface tension force is directed normal to the interface segment, applying the contact angle directly to the interface segment orientation can result in a misdirected surface tension force. A better approach is to have the contact angle influence the curvature calculation.

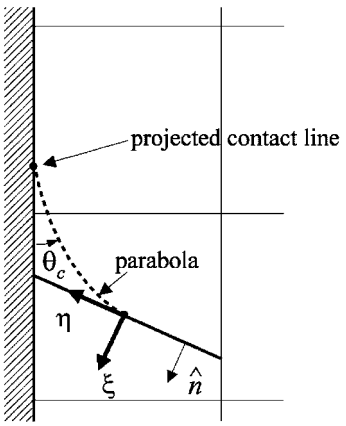

Figure 7. The projected contact line.

Since the interface segment in the contact-line cell, hereafter referred to as the *contact-line segment*, lacks a neighbour segment on the other side of the wall, the contact angle condition presents a convenient opportunity for providing the second neighbour point for the circle-fit technique of Section 3.4.2. The rationale for this approach is as follows: as the contact-line segment is a piecewise-linear representation of the interface, its intersection with the wall does not accurately represent the contact-line location. A more realistic contact-line position can be projected by defining a curve that runs from the contact-line segment to the wall and meets it at the prescribed contact angle, θ_c, as shown in Figure 7. This projected contact line is then used as the second neighbour point for the circle fit of the contact-line segment and, thus, it influences the curvature and orientation of the segment.

A parabola provides a simple means of projecting the contact line. The equation for the parabola is given by

$$\xi = a_2\eta^2 + a_1\eta + a_0 \tag{37}$$

where the (ξ, η) coordinate system is aligned with the contact-line segment and has its origin at the midpoint of this segment.

The coefficients of this equation are found by satisfying the following boundary conditions, which are designed to provide a smooth transition between the contact-line segment and the projected contact line:

1. The curve passes through the midpoint of the reference segment ($\eta = 0, \xi = 0$). Thus, a_0 equals zero.
2. The curve is tangent to the contact-line segment at the midpoint ($\eta = 0, \mathrm{d}\xi/\mathrm{d}\eta = 0$). Thus, a_1 equals zero.
3. The curve satisfies the contact angle θ_c at the wall.

The third boundary condition is used to solve for a_2. Then, the intersection point between the curve and the wall is determined. This intersection point is the projected contact-line location.

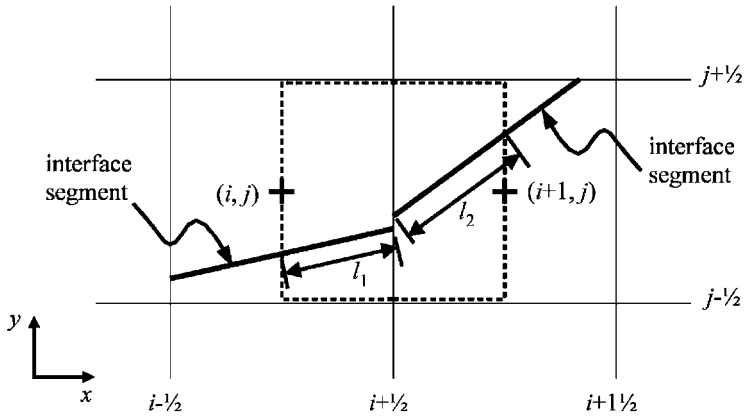

Figure 8. A control volume staggered in the x direction.

5. STAGGERED FORCES AND PRESSURE ADJUSTMENTS

5.1. Staggered body forces

One cause of poor numerical force balances at the interface, in a classical FVM formulation [36], is that the pressure gradients are defined between nodes, calculated as the slopes of the piecewise-linear fits between the nodal pressures, while the body forces are cell-centred. Away from the interface, where fluid properties and flow conditions change gradually, this arrangement does not present a problem. Near the interface, however, this numerical approximation of body forces as cell-centred values and pressure gradients as face-centred values prevents exact balances of the body and pressure forces. Proper numerical force balances are achievable only if the body forces, due to both surface tension (modelled here as a volumetric body force) and gravity, are formulated at face-centred locations, in a manner that is consistent with the treatment of the pressure gradients.

Consider the staggered CV located between nodes (i,j) and $(i+1,j)$, as shown in Figure 8. The cell faces are shown to have indices that are offset from the nodal indices by $\frac{1}{2}$. Thus, this staggered CV and its associated variables are denoted by the index $(i+\frac{1}{2},j)$. The staggered volumetric surface tension force is calculated by appropriately summing the cell-centred values of this force that reside within this staggered CV. Based on Equation (36), the components of this staggered volumetric force are given by

$$S_{\mathrm{st},i}^{(i+1/2,j)} = \frac{(\sigma\kappa n_i)^{(i,j)} l_1 + (\sigma\kappa n_i)^{(i+1,j)} l_2}{0.5(\Delta A^{(i,j)} + \Delta A^{(i+1,j)})} \tag{38}$$

where l_1 and l_2 are the lengths of the portions of the interface segments in cells (i,j) and $(i+1,j)$, respectively, that reside in the staggered CV $(i+\frac{1}{2},j)$. These lengths are illustrated in Figure 8.

The staggered volumetric gravitational force is calculated by first determining the fluid density in the staggered CV, $\rho^{(i+1/2,j)}$, based on the current interface reconstruction. From this staggered CV density, the components of the staggered volumetric gravitational force are

calculated as

$$S_{g,i}^{(i+1/2,j)} = \rho^{(i+1/2,j)} g_i \tag{39}$$

Thus, components of the total volumetric body force corresponding to the control volume staggered in the x-direction are equal to

$$S_i^{(i+1/2,j)} = S_{\mathrm{st},i}^{(i+1/2,j)} + S_{g,i}^{(i+1/2,j)} \tag{40}$$

This is the ith component of the body force, per unit volume, acting on the eastern half of cell (i,j), and the western half of cell $(i+1,j)$. The ith component of the volumetric body force corresponding to the control volume staggered in the y-direction, $S_i^{(i,j+1/2)}$, is similarly derived, and it describes the volumetric body force acting on the northern half of cell (i,j) and the southern half of cell $(i,j+1)$.

The staggered forces are easily accommodated in the numerical method presented in Section 3. Basically, they are treated in the same manner as pressure gradients: whenever the pressure gradients are used directly (such as in the calculation of velocities at the CV faces), so are the staggered forces; when the pressure gradients are averaged over the CV, so are the staggered forces in the x and y directions, and denoted as $\tilde{S}_x^{(i,j)}$ and $\tilde{S}_y^{(i,j)}$, respectively.

5.2. *Pressure adjustment terms*

Despite the use of staggered body forces, force imbalances can persist in the numerical model because, when the interface is present in a cell, a single nodal pressure cannot accurately represent the average pressure at both the vertical face of a CV staggered in the x direction and the horizontal face of a CV staggered in the y direction. Thus, in this work, separate average pressures are calculated by defining two pressure adjustments for each cell, one for the x-direction and one for the y-direction, which are added to the nodal pressure. The nodal pressures are still solved for, using the discretized pressure equations given by Equation (18), with some minor modifications as described later in this section.

5.2.1. Surface-tension-derived pressure adjustments. In this subsection, pressure adjustments based on the pressure jump caused by surface tension are developed. Gravitational effects are considered in the next subsection. For a staggered CV, the two important faces with respect to the pressure adjustments are the ones parallel to the cell faces, which pass through the nodes of the overlapped cells. As shown in Figures 8 and 9, for x-staggered CVs, these faces are represented by vertical median lines, which divide the cells in half; it should be noted that the median lines are horizontal for y-staggered CVs. The average pressure at an x-staggered CV face can be defined as

$$p_{\mathrm{avg},x}^{(i,j)} = p^{(i,j)} + p_{\mathrm{st},x}^{\prime(i,j)} \tag{41}$$

where $p_{\mathrm{avg},x}^{(i,j)}$ is the average pressure along the vertical median line in cell (i,j), $p^{(i,j)}$ is the nodal pressure in cell (i,j), and $p_{\mathrm{st},x}^{\prime(i,j)}$ is the surface-tension-derived pressure adjustment for the vertical median line in cell (i,j). If the interface does not cross the vertical median line, then the nodal pressure is a good approximation to the average pressure on that line and $p_{\mathrm{st},x}^{\prime(i,j)}$ equals zero. If, however, the interface does cross the median line, the pressure on a portion

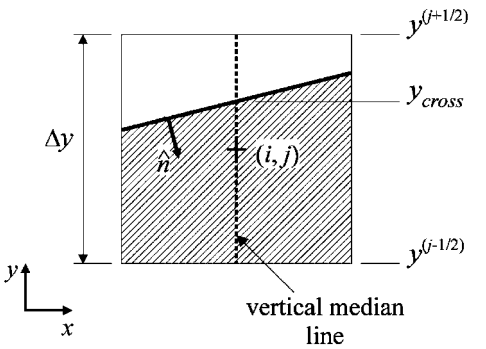

Figure 9. Geometry for the surface-tension-derived pressure adjustment in the x direction.

of this line experiences a pressure jump of $\pm\sigma\kappa$. The appropriate pressure adjustment must spread this pressure surplus or deficit along the length of the vertical median line, as follows:

$$p'_{\mathrm{st},x} = \pm\,\sigma\kappa r_{\mathrm{ml}} \tag{42}$$

where r_{ml} is the *fraction* of the median line that lies on the opposite side of the interface from the cell node. Using the notation of Figure 9, this fraction is calculated as

$$r_{\mathrm{ml}} = \min\left(\frac{y^{(j+1/2)} - y_{\mathrm{cross}}}{\Delta y}, \frac{y_{\mathrm{cross}} - y^{(j-1/2)}}{\Delta y}\right) \tag{43}$$

where $y^{(j+1/2)}$ and $y^{(j-1/2)}$ are the locations of the top and bottom cell faces, respectively, and y_{cross} is given by the intersection between the interface segment and the vertical median line. The minus sign is used on the right-hand side of Equation (42) if the node lies in Fluid A, that is, if $\hat{n}$ points towards the node; otherwise, the plus sign is used.

The pressure adjustment for the y-direction, $p'_{\mathrm{st},y}$, is calculated similarly, using the horizontal median line. Both pressure adjustments are zero in all non-interface cells. Within an interface cell, $p'^{(i,j)}_{\mathrm{st},x}$ is zero if the interface segment does not cross the vertical median line, as is $p'^{(i,j)}_{\mathrm{st},y}$ if the interface segment does not cross the horizontal median line.

To demonstrate the effectiveness of the staggered force formulation and the pressure adjustments, a system containing a nominally static drop is simulated. The initial fluid configuration consists of a circular region of Fluid A, with a radius of 0.24 m, centred in a $1.0\,\mathrm{m} \times 1.0\,\mathrm{m}$ domain and surrounded by Fluid B. Initial velocities are zero and gravity is absent. Both fluids have the same ρ and μ values, here set similar to those of air, but the surface tension coefficient at the interface is specified to be 0.0728 N/m. The simulation is performed on a 40×40 uniform mesh, with no-slip boundary conditions at the walls (outer boundaries of the calculation domain), and a time-step of 0.05 s.

Results of the simulation after 20.0 s are shown in Figure 10. On the left side are three-dimensional plots of the mesh with pressure, in Pascals, used as the z-value, while velocity vector plots are presented on the right side. In Figure 10(a), results obtained with the traditional co-located FVM formulation presented in Section 3, without staggered forces or pressure adjustments, are shown. This formulation gives rise to large non-physical pressure variations inside and outside the drop, and asymmetries are also evident due to poor convergence at

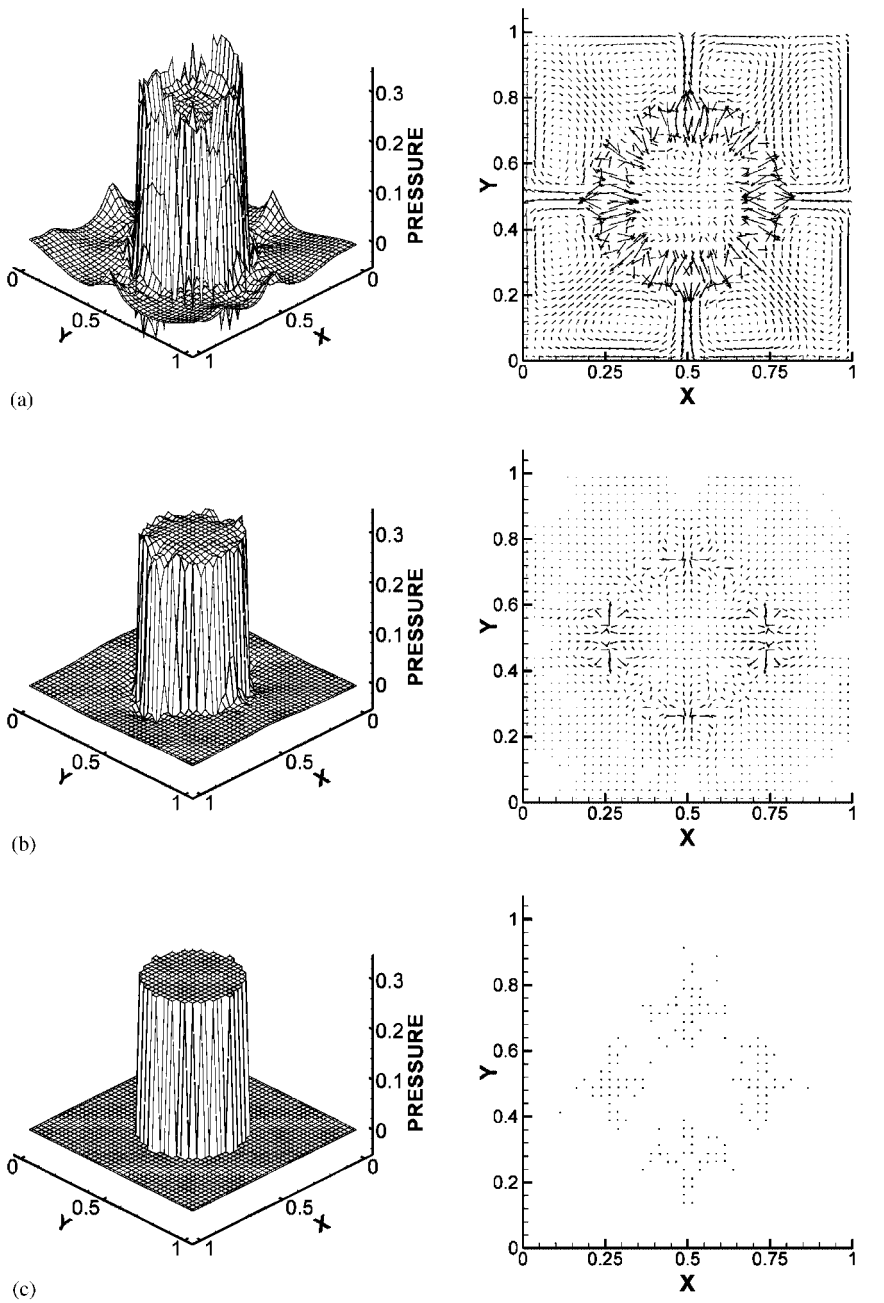

Figure 10. Pressure and velocity vector plots for the static drop simulations: (a) traditional formulation; (b) staggered-force formulation; and (c) staggered-force formulation with pressure adjustments.

some times. In addition, the velocity vector plot indicates that the levels of flow are significant throughout the domain and are especially strong near the interface. This flow is an example of the parasitic currents [28, 29] described in the introduction to this paper. They are produced by imbalances between the surface tension and pressure forces at the interface.

Volume tracking researchers who use this traditional formulation often rely on smoothing of the surface tension force to reduce the parasitic currents [24, 29]. Smoothing provides only limited relief, however, and smears the pressure change across the interface, usually over several cell widths [35]. By contrast, in this work, the staggered forces and pressure adjustments are formulated and employed to directly reduce the force imbalances, while maintaining the abrupt pressure change across the interface.

Now consider the results of the simulations with the staggered-force formulation, shown in Figure 10(b). With this formulation, the non-zero pressure gradients are largely confined to the cells near the interface and the parasitic currents are significantly reduced. By also employing the pressure adjustments, virtually uniform pressure is obtained both inside and outside the drop, as is evident from the results shown in Figure 10(c); furthermore, the parasitic currents are reduced to comparatively insignificant levels.

Thus, it is seen that the staggered forces and surface-tension-derived pressure adjustments are capable of enabling highly accurate force balances. The degree to which the pressure adjustments are effective can vary depending on the quality of the curvature calculation, but they generally prevent the appearance of jagged pressure profiles and strongly inhibit parasitic currents.

5.2.2. Hydrostatic-derived pressure adjustments. An abrupt density change across the interface can also produce force balance deficiencies in the presence of a gravitational field. In this subsection, hydrostatic-derived pressure adjustments are developed to improve the force balances. In this development, surface tension effects are ignored, as they were already considered in the previous subsection.

Consider the pressure variation on the vertical median line through cell (i, j), shown in Figure 11. A *local* vertical coordinate, y^*, is defined with its origin at the centre of the cell. Using ideas akin to those used in the development of pressure adjustments in the last subsection, the hydrostatic-derived pressure adjustment in the x-direction, $p'^{(i,j)}_{\mathrm{hy},x}$, is added to the nodal pressure, $p^{(i,j)}$, to obtain the average pressure over the vertical median line, $p^{(i,j)}_{\mathrm{avg},x}$, as follows:

$$p^{(i,j)} + p'^{(i,j)}_{\mathrm{hy},x} = p^{(i,j)}_{\mathrm{avg},x} \tag{44}$$

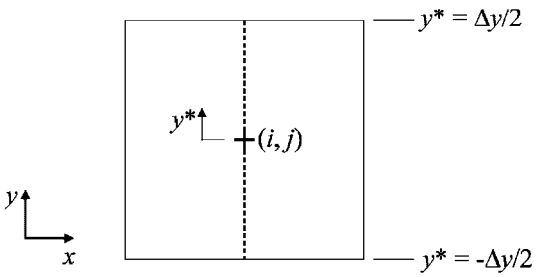

Figure 11. Geometry for the hydrostatic-derived pressure adjustment in the x direction.

The average pressure is obtained by integrating the pressure along the median line

$$p_{\text{avg},x}^{(i,j)} = \frac{1}{\Delta y}\int_{-\Delta y/2}^{\Delta y/2} p_{\text{ml}}\,\mathrm{d}y^* \tag{45}$$

where p_{ml} is the local pressure along the median line. This local pressure can be approximated in terms of the nodal pressure and the hydrostatic pressure variation in the following manner:

$$p_{\text{ml}} = p^{(i,j)} + \int_0^{y^*} \rho_{\text{ml}} g_y\,\mathrm{d}y^* \tag{46}$$

where ρ_{ml} is the local fluid density along the median line and, thus, equals ρ_A or ρ_B at any given point on the line.

Equation (46) is substituted into Equation (45), and the result into Equation (44), to obtain

$$p^{(i,j)} + p_{\text{hy},x}^{\prime(i,j)} = \frac{1}{\Delta y}\int_{-\Delta y/2}^{\Delta y/2}\left\{p^{(i,j)} + \int_0^{y^*} \rho_{\text{ml}} g_y\,\mathrm{d}y^*\right\}\mathrm{d}y^* \tag{47}$$

As the nodal pressure, $p^{(i,j)}$, is treated as having no spatial dependence in a given cell, it can be removed from the integral on the right-hand side, and thus it cancels the identical term on the left-hand side. The remaining terms define the hydrostatic-derived pressure adjustment in the x-direction, as follows, noting that g_y is assumed to be constant:

$$p_{\text{hy},x}^{\prime(i,j)} = \frac{g_y}{\Delta y}\int_{-\Delta y/2}^{\Delta y/2}\left\{\int_0^{y^*} \rho_{ml}\,\mathrm{d}y^*\right\}\mathrm{d}y^* \tag{48}$$

The solution to this double integral has two different forms, depending on whether the interface segment crosses the median line above or below the node. The location of the intersection point is represented by y_{cross}^*. If $y_{\text{cross}}^* > 0$:

$$p_{\text{hy},x}^{\prime(i,j)} = \frac{g_y}{\Delta y}\left\{\begin{array}{l}\frac{1}{2}\rho_u((y_{\text{top}}^*)^2 - (y_{\text{cross}}^*)^2) + \frac{1}{2}\rho_b((y_{\text{cross}}^*)^2 - (y_{\text{bot}}^*)^2)\\ \quad +(\rho_b - \rho_u)y_{\text{cross}}^*(y_{\text{top}}^* - y_{\text{cross}}^*)\end{array}\right\} \tag{49}$$

where ρ_u and ρ_b are the densities of the fluids in the upper and lower parts of the cell, with respect to the interface, respectively, and where $y_{\text{top}}^* = \Delta y/2$ and $y_{\text{bot}}^* = -\Delta y/2$. If $y_{\text{cross}}^* < 0$, the third term in between the braces in Equation (49) changes, as follows:

$$p_{\text{hy},x}^{\prime(i,j)} = \frac{g_y}{\Delta y}\left\{\begin{array}{l}\frac{1}{2}\rho_u((y_{\text{top}}^*)^2 - (y_{\text{cross}}^*)^2) + \frac{1}{2}\rho_b((y_{\text{cross}}^*)^2 - (y_{\text{bot}}^*)^2)\\ \quad +(\rho_u - \rho_b)y_{\text{cross}}^*(y_{\text{cross}}^* - y_{\text{bot}}^*)\end{array}\right\} \tag{50}$$

The x-direction hydrostatic pressure adjustment is non-zero only if the interface crosses the median vertical line and if, as formulated in Equations (49) and (50), the two fluid densities are different. As with the surface-tension pressure adjustments, the hydrostatic pressure adjustments are zero in the non-interface cells. The y-direction hydrostatic pressure adjustments, $p_{\text{hy},y}^{\prime(i,j)}$, are found similarly and result in equations of the same form as Equations (49) and

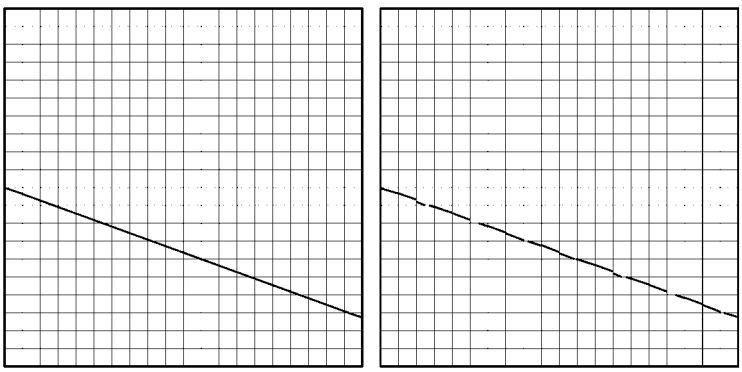

Figure 12. Interface of hydrostatic air–water system: (a) with hydrostatic pressure adjustments; and (b) without hydrostatic pressure adjustments.

(50), except that all y's on the right-hand sides are replaced with x's, and the upper and lower densities are replaced by eastern and western densities, respectively.

To demonstrate the efficacy of the hydrostatic pressure adjustments, a static air–water system in a domain tilted at 20° to the horizontal is simulated. The domain has dimensions of $1.0\,\mathrm{m} \times 1.0\,\mathrm{m}$, is bounded by solid walls, and is discretized with a 20×20 mesh of uniform cells. The initial conditions are hydrostatic and a time-step of 0.001 s is used. The gravitational acceleration is $9.81\,\mathrm{m/s^2}$, and the surface tension is neglected.

This simulation is run to $t = 0.1$ s, with and without the hydrostatic pressure adjustments, but always with the staggered-force formulation. As shown in Figure 12, a continuous linear interface is maintained when the hydrostatic pressure adjustments are used while, without these pressure adjustments, the interface is fractured. Thus, the need for hydrostatic pressure adjustments in order to accurately represent a static fluid system is clear. These adjustments are also appropriate for dynamic simulations, provided that the fluid acceleration is small compared to the gravitational acceleration, in which case Equation (46) is a reasonable approximation. If not, the hydrostatic pressure adjustments can be disabled.

Although the calculations of the surface tension and hydrostatic pressure adjustments are not especially demanding, the additional coupling that the adjustments introduce between the interface reconstruction and the pressure field can increase the number of iterations required per time-step and, thus, the overall simulation run-time. This additional cost, however, is justified by the better behaved and more accurate simulations that are obtained.

5.2.3. Implementation of the pressure adjustments. The pressure adjustments are integrated into the numerical formulation described in Sections 3.1 and 3.2 in a straightforward manner. The adjustments $p'^{(i,j)}_{\mathrm{st},x}$ and $p'^{(i,j)}_{\mathrm{hy},x}$ are calculated from Equations (42), (43), (49), and (50), while $p'^{(i,j)}_{\mathrm{st},y}$ and $p'^{(i,j)}_{\mathrm{hy},y}$ are calculated from the equivalent equations for the y-direction. During the overall iterative solution procedure, these adjustment terms are moderately under-relaxed (typically, the under-relaxation parameter $\alpha = 0.7$), as are the surface tension forces and the momentum equations, to improve the stability and convergence properties of the

simulation. After these adjustments are calculated, they are summed to create a combined pressure adjustment for each direction as follows:

$$p'_x = p'_{\mathrm{st},x} + p'_{\mathrm{hy},x}; \quad p'_y = p'_{\mathrm{st},y} + p'_{\mathrm{hy},y} \tag{51}$$

Thus, a set of pressure and pressure adjustment values, p, p'_x, and p'_y is available for each cell in the calculation domain.

The pressure adjustments are incorporated into the discretized equations by replacing p with $p + p'_x$ for the calculation of pressure gradients in the x-direction, and with $p + p'_y$ for the calculation of pressure gradients in the y-direction, immediately following the FVM discretization of the governing equations. To calculate the velocities at the cell faces, the pressure adjustments are included in the calculation of the pressure gradients for the staggered CVs. With respect to the discretized equation for pressure, the nodal pressures are treated as unknowns, and the pressure adjustments are lumped into the constant.

6. TEST PROBLEMS

6.1. Pure advection

Rider and Kothe [19, 47] have proposed four test cases, all involving pure advection (no diffusion and no surface tension), for evaluating the accuracy of interface tracking techniques, one of which is considered here. Discussions of the application of the proposed method to all four of these test cases and the results, for a range of parameters, are available in References [41, 42]. These test cases check the ability of the proposed method to not only translate an interface, but also to stretch and tear it in the presence of vortical flow. The velocity field is prescribed in each test case so that the properties of the fluids, other than their identities, are unimportant. Thus, these tests gauge the accuracy of the interface reconstruction techniques and the advection of the volume-fraction field (F).

6.1.1. Test description. This test is conducted on a $1.0\,\mathrm{m} \times 1.0\,\mathrm{m}$ domain and Fluid A is initially in the shape of a filled circle of radius $0.15\,\mathrm{m}$, centred at $(0.5\,\mathrm{m}, 0.75\,\mathrm{m})$, and the rest of the domain is considered to be filled with Fluid B. The prescribed flow field consists of a complex deformation field produced by a 4×4 array of symmetrical, counter-rotating vortices, which is described by the following velocity components [19]:

$$\begin{aligned} u &= \sin\{4\pi(x + 0.5)\} \sin\{4\pi(y + 0.5)\} \cos(\pi t/T) \\ v &= \cos\{4\pi(x + 0.5)\} \cos\{4\pi(y + 0.5)\} \cos(\pi t/T) \end{aligned} \tag{52}$$

In this equation, the trailing cosines make the flow field reversible. The maximum value of u given by the above equation is $1\,\mathrm{m/s}$ and the selected CFL number is 1.0. Thus, the time-step Δt equals $(1/n_x)\,\mathrm{s}$, where n_x is the number of cells in the x-direction. Additionally, spatially periodic conditions are assumed to apply at the domain boundaries, and the time period, T, of the flow is set equal to $2.0\,\mathrm{s}$.

As the flow is reversible, the initial circle of Fluid A should be reformed at its original location when $t = T$. Any differences between the initial and the final configurations of this fluid are due to errors caused by inaccuracies in the interface tracking techniques. These errors

can be quantified using the following L_1 error norm:

$$\|F\|_{L_1} = \sum_{i,j} \Delta A^{(i,j)} |F^{(i,j)}_{\text{final}} - F^{(i,j)}_{\text{initial}}| \tag{53}$$

where $\Delta A^{(i,j)}$ is the cell area, $F^{(i,j)}_{\text{final}}$ is the final volume fraction, and $F^{(i,j)}_{\text{initial}}$ is the initial volume fraction, all for cell (i,j). One measure of the order of these volume tracking techniques is presented in the form of a 'grid convergence rate' [47, 48], defined as

$$\text{Grid Convergence Rate} = \frac{\ln(\|F\|_{L_1,\text{fine}}/\|F\|_{L_1,\text{coarse}})}{\ln(n_{x,\text{coarse}}/n_{x,\text{fine}})} \tag{54}$$

where the subscripts fine and coarse refer to simulations conducted with higher and lower mesh resolutions, respectively.

6.1.2. Results. The evolution of the interface on a uniform 128×128 mesh is displayed as interface segment plots in Figure 13. From $t = 0.5$ s onward, the severity of the deformation is such that the volume tracking implementation is unable to correctly resolve the thin strands of drawn fluid, which instead appear as droplets. At $t = 2.0$ s, when the original circle should be reformed, a misshapen circle is observed.

Results obtained with the PLIC volume tracking method proposed in this paper were compared to those of Rider and Kothe [19]. They use trapezoid advection to evolve the F field and an error minimization method [16, 25] to calculate the interface segment orientation, $\hat{n}$. The errors in reforming the circle at $t = T$ are listed in Table I for uniform $32\times32, 64\times64, 128\times128$, 256×256, and 384×384 meshes. Both PLIC volume tracking methods have only limited success at simulating the extremely complex topological changes on the three coarsest grids, as evidenced by the relatively high L_1 error norms and low grid convergence rates. Limitations in the interface reconstruction technique are largely responsible for this poor performance in the case of the proposed method. Recall that, for the proposed method, interface segments are oriented in a two-step process, in which a least-squares approximation to $\vec{\nabla} F$ is used to calculate an initial $\hat{n}$, based on which the circle-fit technique calculates the final $\hat{n}$. On the relatively coarse meshes $(32 \times 32, 64 \times 64, 128 \times 128)$, when the stretched fluid regions become very thin, they are inadequately resolved for the $\vec{\nabla} F$ calculations. Thus, a highly inaccurate initial $\hat{n}$ is sometimes obtained, which in turn can lead the circle-fit technique to either choose incorrect neighbours or else fail in locating two suitable neighbours. In either case, the interface segments are misoriented.

Results for the 256×256 and 384×384 meshes are significantly better: low L_1 error norms and second-order grid convergence rates are observed in Table I. These results, together with the excellent results obtained in the other three (simpler) pure advection tests [41, 42], demonstrate the accuracy of the volume fraction advection and interface reconstruction techniques that form the core of the proposed method.

6.2. Static drop

This test case demonstrates the efficacy of the surface-tension-derived pressure adjustments discussed in Section 5.2.1. Variations of this test case have been used by several researchers

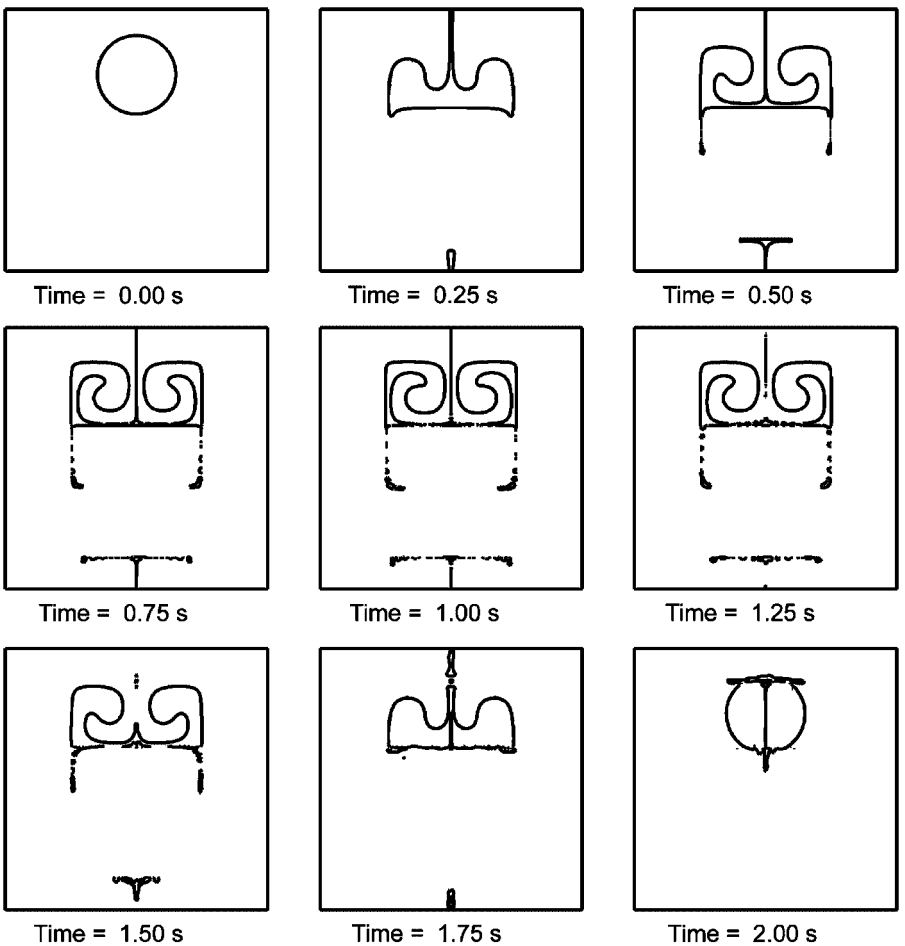

Figure 13. Interface segment plots obtained in the pure advection test problem.

Table I. L_1 error norms and order (grid convergence rates) for the pure advection problem.

	32^2 Mesh	Order	64^2 Mesh	Order	128^2 Mesh	Order	256^2 Mesh	Order	384^2 Mesh
Proposed method	2.00×10^{-2}	0.96	1.03×10^{-2}	1.14	4.66×10^{-3}	2.26	9.70×10^{-4}	2.67	3.29×10^{-4}
Rider and Kothe [19]	1.96×10^{-2}	0.81	1.12×10^{-2}	0.91	5.95×10^{-3}	—	—	—	—

[28–30, 35]. The emphasis in this subsection is on evaluations of the accuracy of the curvature calculation and the ability of the proposed method to maintain a circular interface, under physically steady-state conditions.

6.2.1. Test description. The dimensions of the calculation domain are $1\,\mathrm{m} \times 1\,\mathrm{m}$, and water is contained in a circular region centred in the middle of the domain. The radius of this circular region is 0.24 m, so the interface curvature equals $4.167\,\mathrm{m}^{-1}$. Outside of this region, the calculation domain is filled with air. The fluid properties are as follows: $\rho_{\mathrm{air}} = 1.205\,\mathrm{kg/m^3}$, $\rho_{\mathrm{water}} = 998\,\mathrm{kg/m^3}$, $\mu_{\mathrm{air}} = 1.808 \times 10^{-5}\,\mathrm{N\text{-}s/m^2}$, $\mu_{\mathrm{water}} = 1.003 \times 10^{-3}\,\mathrm{N\text{-}s/m^2}$, and $\sigma = 0.0728\,\mathrm{N/m}$, where σ is the surface tension coefficient. Gravitational acceleration is zero, and steady-state conditions prevail in the physical problem.

In order to gauge the effectiveness of the circle-fit technique specifically, the simulations are repeated using an alternate curvature calculation in place of the circle-fit technique. In this alternate technique, which is based on the work of Williams *et al.* [30], curvatures are calculated from the gradients of a smoothed volume fraction field. In fact, most available volume tracking methods, including those of Aleinov and Puckett [45], Kothe [15], and Rudman [29], use this approach or closely related approaches.

Williams *et al.* [30] use a radially symmetric, eighth-degree kernel, K_8, as the basis for smoothing the F field. The unit normal to the interface is calculated from the smoothed volume fraction field, $\tilde{F}$, as follows:

$$\hat{n} = \vec{\nabla}\tilde{F} / |\vec{\nabla}\tilde{F}| \tag{55}$$

The curvature is calculated from the divergence of the unit normal [12]

$$\kappa = -\vec{\nabla} \cdot \hat{n} \tag{56}$$

In the next subsection, Equations (55) and (56) will be used as an alternate means to calculate $\hat{n}$ and κ, respectively. This approach will be referred to as the K_8 *kernel technique*. When $\hat{n}$ and κ are determined via the proposed circle-fit technique, the method will be referred to as the *circle-fit technique*.

The static drop simulations are conducted on a uniform mesh of 40×40 cells with a constant time-step of 0.1 s. This time-step was chosen after numerous trials with different values, because it is small enough to ensure stable and convergent solutions at each time-step, it produces solutions that are essentially insensitive to further refinement of Δt, and does not entail excessive computational costs. The simulations were continued for 100 s in order to test the ability of each technique to sustain a static, or near-static, equilibrium. For the K_8 kernel technique, the radius of support was set at 0.2 m.

6.2.2. Results. Plots of the maximum, minimum, and average values of the interface curvature versus time are presented in Figure 14 for the circle-fit technique. The solid horizontal line indicates the analytic curvature for this circle, $\kappa = 4.167\,\mathrm{m}^{-1}$. The dashed line indicates the length-weighted average curvature, $\bar{\kappa}$, for the PLIC representation of the interface, calculated as

$$\bar{\kappa} = \left\{ \sum_{(i,j)} l^{(i,j)} \kappa^{(i,j)} \right\} \Bigg/ \sum_{(i,j)} l^{(i,j)} \tag{57}$$

In this equation, the summations are done over all cells in the calculation domain: It should be noted that in cells without the interface, the segment length is $l^{(i,j)} = 0$. As the curvature is

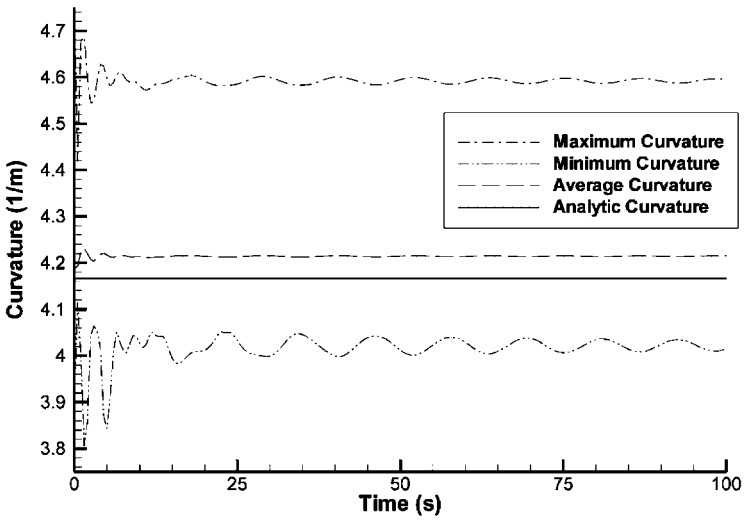

Figure 14. Curvatures obtained using the circle-fit technique in the static drop test problem.

calculated independently for each segment, the curvatures about the circumference of the reconstructed circle vary. The range of this variation is indicated in Figure 14 by the maximum and minimum segment curvatures at each time.

As shown in Figure 14, the curvatures computed by the circle-fit technique display oscillatory behaviour in time. These oscillations decay with time. In addition, the spread between the maximum and minimum curvatures is quite large, approximately equal to 15% of the analytic value of the curvature. The length-weighted average curvatures, however, are close to the analytic curvature, with an error, after 100 s, of about 1.1%. The same simulation conducted on a coarser 20×20 mesh [41] gives the same range of curvatures and a smaller average curvature error, of about 0.025%. The increase in the error in predictions of the average curvature as the mesh is refined, and the corresponding lack of improvement in the curvature spread, indicates that the proposed circle-fit technique cannot be used for curvature calculations of high accuracy: It is only able to reduce the errors in the curvature calculations with grid refinement until they are within a certain minimum radius of convergence about the exact value of curvature.

The increasing error as the mesh is refined seems to result from the decreasing distance between the three segment-midpoints that are used to perform the circle fit (see Section 3.4). As these three midpoints get closer together, the circle fit becomes more sensitive to any inaccuracies in the locations of the segments and the resultant segment-midpoints, which is reflected in greater variations in the calculated curvatures. Several attempts were made in this work to improve the accuracy of the circle-fit technique by smoothing or by incorporating additional, more distant midpoints into the calculation. While these changes initially improved the results, over time the solutions degraded and became much worse than those produced with the original circle-fit technique. In fact, one of the main strengths of the proposed circle-fit technique is that its results remain bounded throughout the simulation, unlike the results from the K_8 kernel technique, as will be demonstrated next.

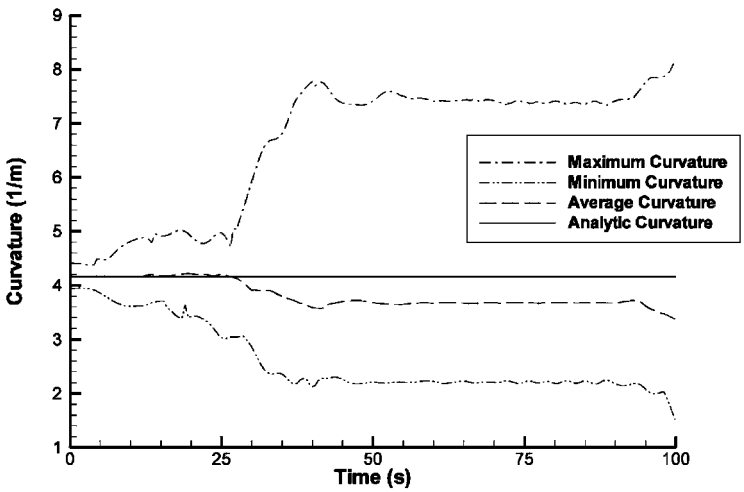

Figure 15. Curvatures obtained using the K_8 kernel technique in the static drop test problem.

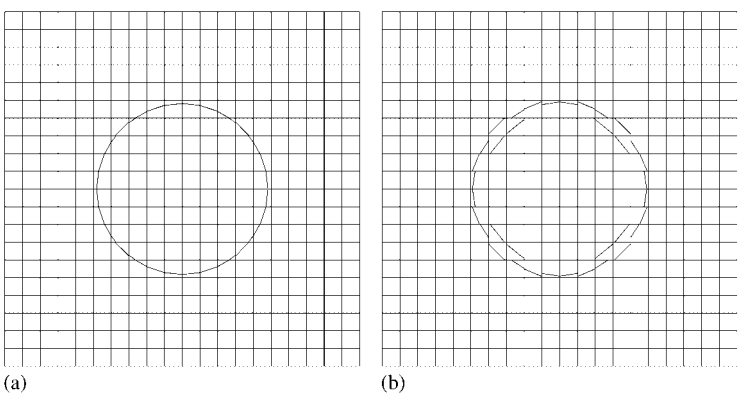

Figure 16. Interface segment plots for the static drop test problem, obtained using a uniform 20×20 mesh at $t = 100$ s: (a) proposed circle-fit technique; and (b) K_8 kernel technique.

Consider the plots in Figure 15 that present results produced using the K_8 kernel technique. At $t = 0$ s, the error in the average curvature is only 0.013%, which is much smaller than the initial curvature error in Figure 14. By the end of the simulation, however, this error increases to 19%. In addition, while the spread between the maximum and minimum curvature values is initially small, it eventually grows to more than 125% of the analytic value of the curvature.

Also consider the interface segment plots in Figure 16, generated on the 20×20 mesh at $t = 100$ s. In Figure 16(a), the results from the proposed circle-fit technique show a relatively smooth circle, that closely matches the initial circle at $t = 0$ s. However, with the K_8 kernel technique, as shown by the results given in Figure 16(b), the interface is disjointed and a double layer of segments has begun to form, due to the parasitic currents generated by bad

curvature calculations and the resultant numerical imbalances between the surface tension and pressure forces. The fact that the average curvature for this segment configuration, at $t = 100\,\text{s}$, is still close to the analytic value of the curvature is indicative of the disconnection that smoothing produces between the actual F field and the curvature.

Thus, the ability of the proposed circle-fit technique to provide accurate static drop simulations is shown to be superior to that of the K_8 kernel technique. Although it is unable to provide high-accuracy curvatures, the circle-fit technique produces reasonable curvature values and excellent results in the context of interface fidelity and suppression of parasitic currents. As the curvatures are calculated locally, using three neighbouring segments, the circle-fit technique responds immediately to local interface conditions, and thus prevents interface disruptions. In contrast, smoothing techniques, such as the K_8 kernel technique, provide superior calculations of curvature initially, but they promote the growth of parasitic currents and, therefore, produce unacceptably large errors at later times in the simulations.

6.3. *Oscillating bubble*

Numerical simulations of a planar two-dimensional bubble undergoing oscillations, due to surface tension and an initial perturbation, have been conducted by Fyfe *et al.* [8], using a moving-mesh method. Both the fluids, inside and outside the bubble, are assumed to be inviscid. Thus, in theory, the oscillations should continue indefinitely, with a constant period and amplitude. To check the accuracy of their numerical simulations, Fyfe *et al.* [8] modified a linear theory derived by Rayleigh [49] for small amplitude oscillations of cylindrical jets. With this linear theory, they were able to obtain an analytic value for the period of the oscillations to which they could compare the period yielded by their numerical simulations.

Numerical results obtained with the proposed PLIC volume tracking method are compared here to those of Fyfe *et al.* [8] and also their analytic solution based on the theory put forward by Rayleigh [49].

6.3.1. Test description. The bubble is located in a $0.001\,\text{m} \times 0.001\,\text{m}$ box, with no-slip and impermeability conditions at the walls. The initial shape of the bubble is given in polar coordinates by

$$r = a + \varepsilon \cos(2\theta) \tag{58}$$

where the origin of the polar coordinate system is at the centre of the box, a is the unperturbed radius of the bubble and ε is the amplitude of the initial perturbation. For this system, $a = 1.25 \times 10^{-4}\,\text{m}$ and $\varepsilon = 2.5 \times 10^{-5}\,\text{m}$. As the horizontal and vertical lines through the centre of the bubble are symmetry lines, and assuming that they are symmetry lines in the numerical simulations too, only one-quarter of the system domain is modelled. Additionally, gravitational effects are absent and $\sigma = 0.03\,\text{N/m}$. To simulate essentially inviscid flow, the fluid viscosities are set to $10^{-50}\,\text{N-s/m}^2$.

Two sets of fluid densities are employed to match the runs of Fyfe *et al.* [8]. In the first set, the bubble fluid has a density of $2000\,\text{kg/m}^3$ and the external fluid has a density of $1000\,\text{kg/m}^3$, giving a density ratio of 2:1. In the second set, the densities are $820\,\text{kg/m}^3$ inside the bubble and $1.26\,\text{kg/m}^3$ outside, yielding a density ratio of 650:1. The grid sizes and time-steps used for the simulation of the 2:1 density ratio flows are listed in Table II; the same grids are used for the 650:1 flows, but with time-steps half of those listed in Table II. Linear

Table II. Grid sizes and time-steps for the oscillating bubble test problem, with 2:1 density ratio.

n_x	n_y	Δt (s)
10	10	1.0×10^{-4}
20	20	2.5×10^{-5}
40	40	6.25×10^{-6}
60	60	2.75×10^{-6}
80	80	1.5625×10^{-6}

theory [8, 49] yields the following predictions of the period of oscillation: 1.134×10^{-3} s for the case in which the density ratio is 2:1; and 5.931×10^{-4} s for the case with a density ratio of 650:1.

In addition to comparing the results yielded by the proposed volume tracking method to the linear theory and numerical results of Fyfe *et al.* [8], the following subsection examines the effect of changing the neighbour selection criterion for the circle-fit technique. In the proposed method, as was discussed in Section 3.4, neighbour selection is based on *a mixed selection criterion*. The criterion is mixed in the sense that it is based both on the volume fractions, as represented by F_{test} in Equation (31), and on the angles made by the lines connecting the reference segment to its neighbour segments, as measured by Γ in Equation (30): It requires that the neighbours that minimize $\Gamma/\sqrt{F_{\text{test}}}$ be chosen. Two other neighbour selection criteria were tested: neighbour selection based on maximizing F_{test}, which is referred to as the *F selection criterion*; and neighbour selection that leads to the minimization of Γ, which is referred to as the *angle selection criterion*.

6.3.2. Results. Plots of the $F = 0.5$ contour of the oscillating bubble are displayed in Figure 17. These results were obtained with the mixed selection criterion, for the 2:1 density ratio test case, and a uniform 80×80 grid. These plots are of the $F = 0.5$ contour with the full domain shown, not just the quadrant that was used as the computational domain. These oscillations are qualitatively correct, although a decrease in the amplitude of the oscillations is apparent. At the end of the full simulation, at $t = 0.01$ s, the bubble mass changed by less than $2.3 \times 10^{-7}\%$, showing the excellent mass conservation property of the proposed volume tracking method.

The decaying amplitude of the bubble oscillations is better observed in Figure 18, where the rightmost point on the interface is plotted versus time. The amplitude steadily decreases, but the period of oscillation remains approximately constant. As the flow is inviscid, the amplitude should be constant. Thus, its decay indicates that numerical dissipation occurs due to discretization errors in the finite volume method (note that the hybrid difference scheme was used) and also inaccuracies in the volume tracking model (most likely in the curvature calculations). Fyfe *et al.* [8] noted similar decay of the amplitude of the oscillations in their simulations with a moving-mesh method.

When the simulations were repeated employing the F selection criterion, the results were very similar to those obtained with the mixed criterion (Figure 18). When the angle selection criterion was used, however, very poor performance was observed: with a 40×40 uniform mesh, the simulations were unable to sustain the oscillations; only about

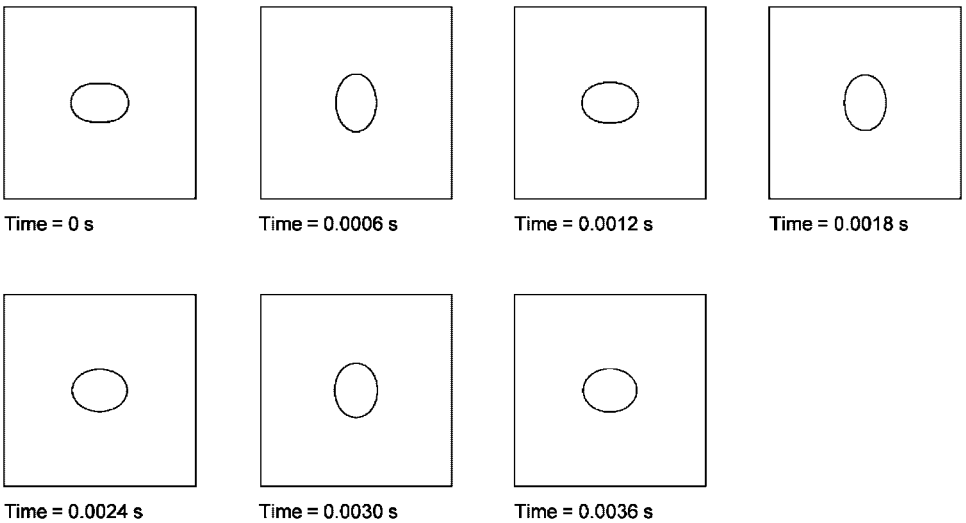

Figure 17. Bubble oscillations obtained with the mixed selection criterion and a uniform 80×80 mesh.

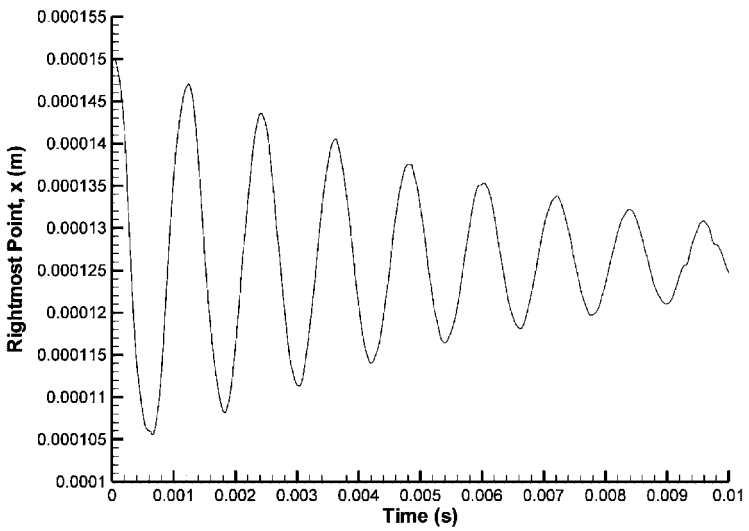

Figure 18. Position of the rightmost point on the interface of the oscillating bubble with 2:1 density ratio, obtained with the mixed selection criterion on a uniform 80×80 mesh.

one-and-a-half oscillations were completed in a discernible manner, before they damped down to levels that were essentially similar to numerical noise; and when the computational mesh was refined, this performance worsened. These bad results with the angle selection criterion are due to inappropriate neighbour selections, which create large non-physical variations in the curvature, producing many local minima and maxima. In turn, these large curvature

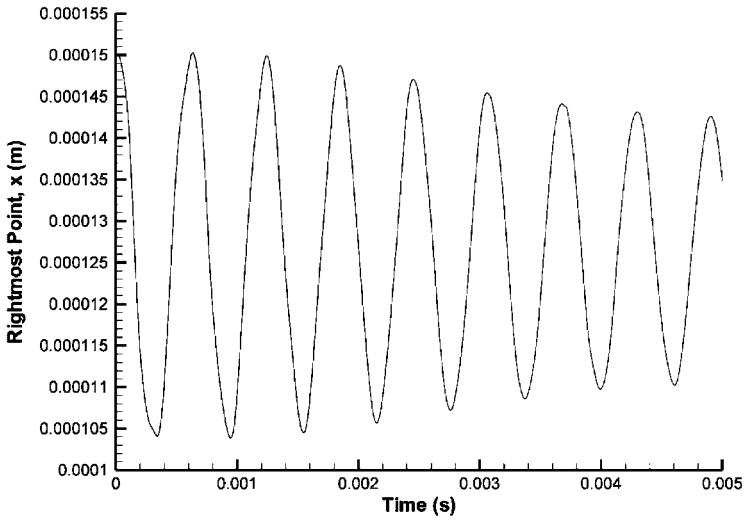

Figure 19. Position of the rightmost point on the interface of the oscillating bubble with 650:1 density ratio, obtained with the mixed selection criterion on a uniform 80×80 mesh.

variations generate additional, high-frequency oscillations that tend to dampen the main bubble oscillation. Consequently, the angle criterion cannot produce reliable results and, thus, was not considered further.

To check whether the placement of the domain boundaries influences the oscillation periods yielded by the numerical simulations, additional computations were conducted in which the domain size was doubled in the x and y directions, for the test case with the 2:1 density ratio. In order to maintain the same mesh resolution, n_x and n_y were also doubled from the values listed in Table II. The numerical results obtained with this larger domain differed insignificantly from those obtained with the smaller domain [41]. Thus, results obtained with the smaller domain were considered adequate for comparisons with the linear theory, in which the domain boundaries are absent.

Now the oscillating bubble for the test case with a 650:1 density ratio is considered. Such high-density-ratio systems have typically proven quite troublesome for volume tracking methods, producing large force imbalances near the interface in some previous numerical formulations (see, for example, the work of Kothe *et al.* [35]). These difficulties, however, have not been experienced with the proposed method, due to the use of staggered forces and pressure adjustments. A plot of the bubble oscillations for this 650:1 density ratio test case, modelled using the mixed selection criterion from $t = 0$ to 0.005 s on a uniform 80×80 mesh, is shown in Figure 19. The amplitude decay in the numerical simulations of this high density test case is considerably smaller than that shown in Figure 18 for the test case with the smaller density ratio.

In Figure 20, plots of the periods of oscillation versus n_x and n_y are given for the test case with the 650:1 density ratio. Only the mixed criterion was used for neighbour selection in this case. The proposed method performs much better than does the moving-mesh method of Fyfe *et al.* [8], who presented results for only one computational grid. Note, however, the

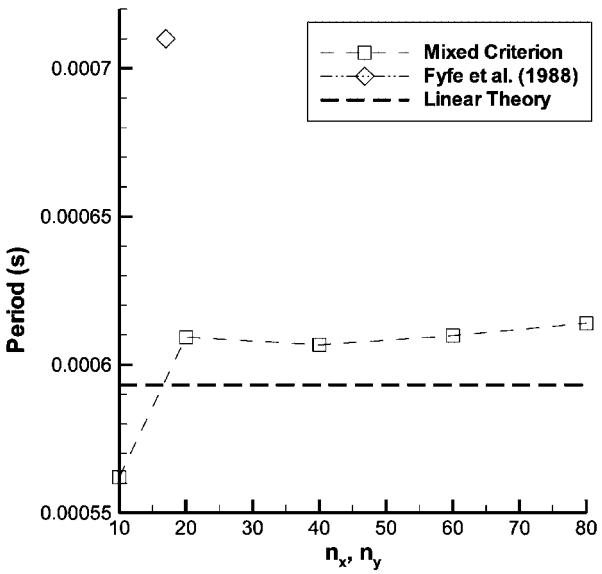

Figure 20. Periods of the oscillating bubble with 650:1 density ratio.

movement of the numerical results away from the linear theory results as the mesh is refined, although the relative difference in these results is only about 3.5% on the 80×80 mesh. This discrepancy between the numerical and analytic results may be due to the presence of non-linear effects in this test case, which are neglected in the linear theory. On the average, however, the agreement between the results produced by the proposed numerical method and the linear theory is quite encouraging, for what is a difficult surface-tension-driven test case that is not commonly used to test volume tracking methods.

6.4. Meniscus between two parallel plates

In this section, simulations of a two-dimensional meniscus formed between two infinite, vertical, parallel plates, in a water–air system, are presented. The contact angle requirement at the wall generates surface tension forces that drive the fluid flow and deform the two-fluid interface so that it makes the proper angle with the wall. Ultimately, an equilibrium state with the proper meniscus shape should be attained.

6.4.1. Analytic description of the meniscus and other test details. The meniscus formed between two infinite, vertical, parallel plates when the fluids are in static equilibrium is illustrated in Figure 21(a). In the figure, H_m is the height of the bottom of the meniscus with respect to the free surface outside the plates; W_m is the horizontal distance between the plates; and θ_c is the steady-state contact angle between the water–air interface and the plate at each end. An analytic description of the meniscus is desirable as it would permit quantification of the errors in the final meniscus shape produced by the numerical simulations. Briefly, a differential equation that satisfies the contact angle requirement was derived [41] following the mathematical model of this problem given by Batchelor [34]. An iterative algorithm and

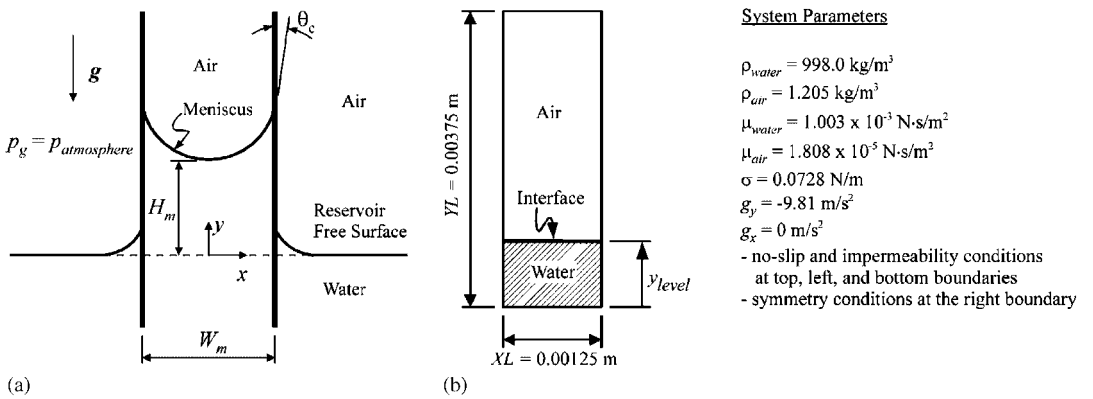

Figure 21. Meniscus formed between two infinite, vertical, parallel plates: (a) problem schematic; and (b) initial configuration and system parameters used in the simulations.

a fourth-order Runge–Kutta method (with 100 000 steps) was used to solve this differential equation numerically, and the iterations were continued until the value of the contact angle was converged to twelve significant figures [41]. For the purposes of this test case, this highly accurate numerical solution and the corresponding curvatures are considered as 'exact', and it is referred to in subsequent subsections as the analytic meniscus solution.

To quantify the accuracy of the computations of the meniscus using the proposed PLIC volume tracking method, two measures of error are used when essentially steady-state conditions prevail. The first is an area-weighted volume fraction error defined as follows:

$$\text{Volume Fraction Error} = \left\{ \sum_{(i,j)} \Delta A^{(i,j)} |F^{(i,j)} - F^{(i,j)}_{\text{analytic}}| \right\} \Big/ (XL \times YL) \tag{59}$$

where the summation is performed over every cell in the domain and $(XL \times YL)$ is the area of the computational domain. F_{analytic} is the volume fraction field derived from the analytic meniscus solution such that

$$\sum_{(i,j)} \Delta A^{(i,j)} F^{(i,j)}_{\text{analytic}} = \sum_{(i,j)} \Delta A^{(i,j)} F^{(i,j)} \tag{60}$$

Equation (60) is based on the requirement that the analytic solution possess the same total area (or mass) of water as the numerical solution, in order to ensure that these solutions are indeed comparable, with the assumption that the numerical solution does not experience significant mass loss.

The second measure of error is a length-weighted curvature error, defined as follows:

$$\text{Curvature Error} = \left\{ \sum_{(i,j)} l^{(i,j)} |(\kappa^{(i,j)} - \kappa^{(i,j)}_{\text{analytic}})/\kappa^{(i,j)}_{\text{analytic}}| \right\} \Big/ \sum_{(i,j)} l^{(i,j)} \tag{61}$$

where $l^{(i,j)}$ is the length of the interface segment in cell (i,j), and $\kappa^{(i,j)}_{\text{analytic}}$ is the analytic value of curvature evaluated at the x-location of the midpoint of the interface segment for cell (i,j).

Table III. Grid sizes and time-steps used in the meniscus simulations.

n_x	n_y	Δt (s)
10	30	1.0×10^{-4}
20	60	5.0×10^{-5}
30	90	2.0×10^{-5}

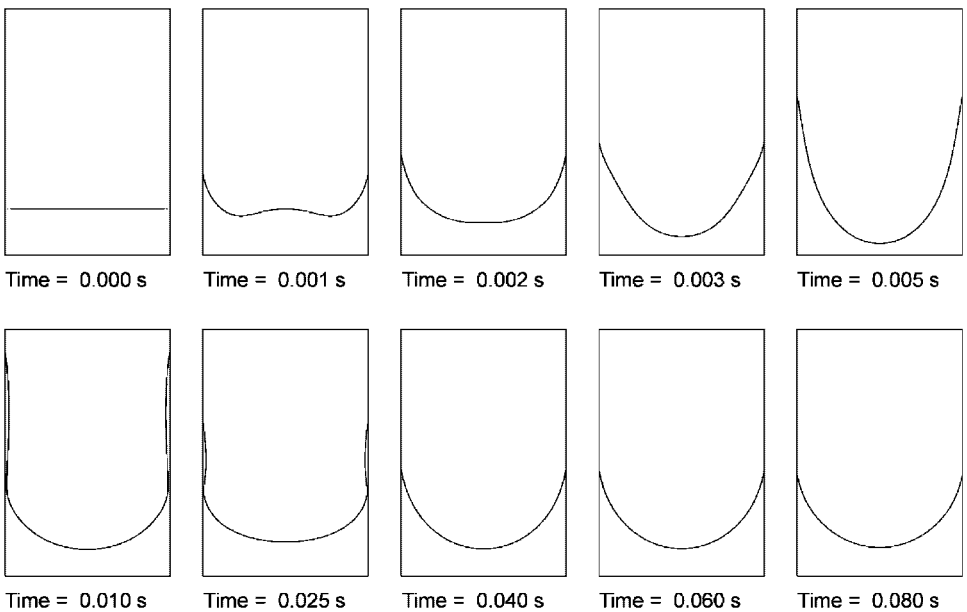

Figure 22. Plots of the interface segments in the meniscus simulations, obtained with a 30×90 mesh with $\theta_c = 10°$.

The initial configuration of the interface is shown in Figure 21(b), along with the system parameters used in the simulations. For computational efficiency, only the region from the left plate to the mid-plane between the plates is included in the computational domain. Thus, the right boundary of the computational domain is assumed to be a symmetry boundary, while the other three boundaries are modelled as solid walls, with no-slip and impermeability conditions. The height of the initially flat interface, y_{level}, is set based on the requirement that at steady-state conditions, the contact angle, θ_c, should be accommodated with a reasonable amount of water at the bottom of the computational domain. Based on many initial trial simulations, the following values of y_{level} were found to be satisfactory: for $\theta_c < 90°$, $y_{\text{level}} = 7.0832 \times 10^{-4}$ m; otherwise, $y_{\text{level}} = YL - 7.0832 \times 10^{-4}$ m. The computations were conducted on three uniform meshes: details of these meshes and the corresponding time-steps are given in Table III. In all simulations, a total run time of $t = 1.0$ s produced essentially steady-state solutions.

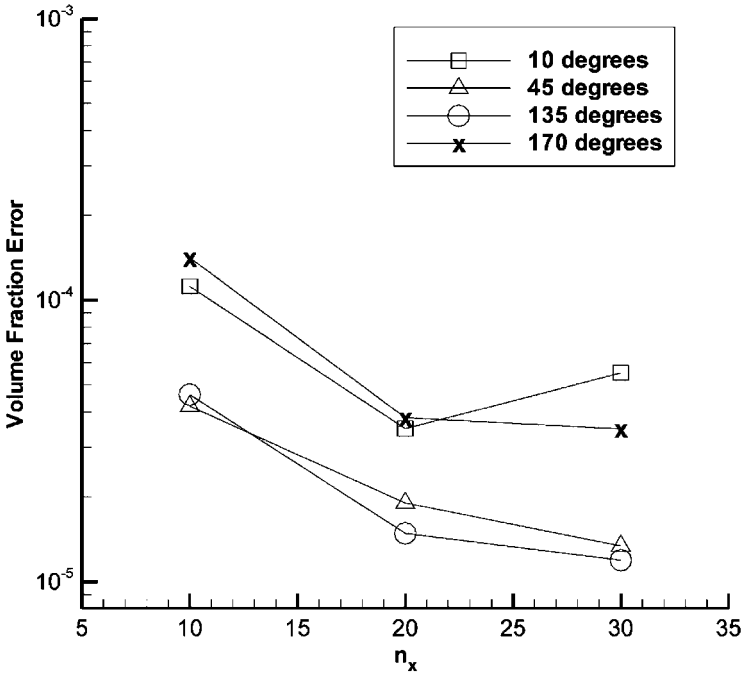

Figure 23. Volume fraction errors in the meniscus simulations.

6.4.2. Results. The computed evolution of the interface is presented as interface segment plots in Figure 22, for the 30×90 mesh with $\theta_c = 10°$. These plots show the full meniscus, by mirroring the computational domain across its right symmetry boundary. At $t = 0$ s, the contact angle is imposed at the vertical walls, according to the treatment described in Section 4.2. As a result, the segments adjacent to the walls experience very high curvatures, which in turn generate large upward surface tension forces. From $t = 0$ to 0.003 s, negative curvatures cause the interface to rise rapidly near the wall, while positive curvatures away from the wall force the interface down. Indeed, the sudden imposition of the contact angle produces such a strong reaction in the water that it climbs high up the walls (see the plot for $t = 0.01$ s) and does not fully drain back down until about $t = 0.030$ s. Thereafter, small oscillations in the interface were observed that gradually decayed with time. These oscillations were extremely small (essentially negligible) by $t = 0.5$ s, but the dynamic simulations were run to $t = 1.0$ s to ensure essentially steady-state conditions. For the simulation shown in Figure 23, after 1.0 s, the change in the mass of water (compared to its initial value) caused by numerical inaccuracies was only $1.05 \times 10^{-7}\%$.

The meniscus formation is simulated for four steady-state contact angles, $\theta_c = 10°$, 45°, 135°, and 170°. In Figure 23, the volume fraction errors (when essentially steady-state conditions have been reached) are plotted versus n_x for each of the four aforementioned steady-state contact angles (using the meshes described in Table III). The volume fraction errors presented in Figure 23 are time-averaged values from $t = 0.9$ to 1.0 s, though the errors are essentially time-invariant at this stage. Most of the errors fall below 10^{-4}, which is

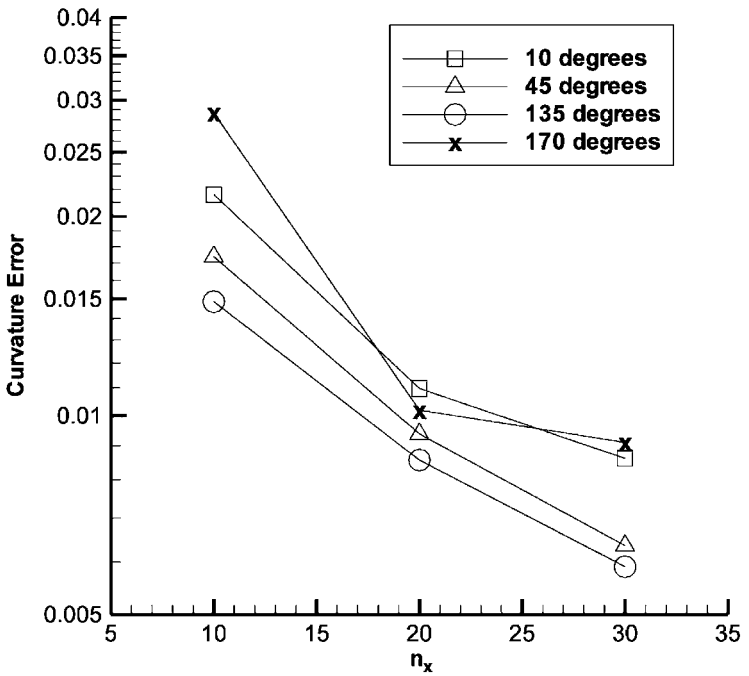

Figure 24. Curvature errors in the meniscus simulations.

equivalent to an average error of less than 0.01% in the F value of every cell in the domain. Also note that the less severe contact angles, $\theta_c = 45°$ and 135°, result in smaller volume fraction errors. In general, the errors decrease as the mesh is refined, except for $\theta_c = 10°$ on the 30×90 mesh.

Essentially steady-state curvature errors computed using the three meshes described in Table III are plotted versus n_x in Figure 24. In this case, the error always decreases as the mesh is refined. On the 30×90 mesh, represented by the rightmost points in the plot, the curvature error for each θ_c is less than 0.01, which is equivalent to a length-averaged curvature error of less than 1%.

The relatively low errors in Figures 23 and 24, plus the physically plausible interface evolution depicted in Figure 22, demonstrate that the proposed contact angle treatment is valid and accurate, at least with respect to the steady-state conditions, as is the numerical model for the surface tension forces.

7. CONCLUSION

A volume tracking method based on a piecewise-linear interface representation has been developed for the simulation of two-fluid flows. This method is based on several new or modified techniques, including a multidimensional advection technique, a circle-fit technique for orienting the interface segments and calculating curvature, and a surface tension model

tied to the interface reconstruction. In addition, staggered body forces and pressure adjustments for improving numerical force balances, and thus inhibiting parasitic currents, have been developed.

Four test problems were presented to demonstrate the capabilities of the proposed method. The first of these test problems involved pure advection and demonstrated that the proposed method is capable of simulating two-fluid flows involving complex interface deformations. The other three test problems, pertaining to a static drop, an oscillating bubble, and a meniscus, introduced the additional complexity of surface tension modelling. While it was found that curvatures of high accuracy could not be computed in the static drop problem using the circle-fit technique, the predicted curvatures were reasonable and the overall simulations remained stable, unlike the simulations conducted with the K_8 kernel. The accuracy of the proposed method was also demonstrated by the essentially constant periods observed for the oscillating bubble problem, and the low volume fraction and curvature errors produced in the meniscus simulations.

Improvements of the proposed circle-fit technique (Sections 3.4.1 and 3.4.2) and surface tension model (Section 4) may be possible by incorporating some of the ideas contained in the works of Ginzburg and Wittum [50], on the representation of the interface using spline interpolants, and Renardy and Renardy [51], on a parabolic reconstruction of surface tension (PROST). Furthermore, with regard to implementation of the surface-tension-derived pressure adjustments described in Section 5.2.1, it would be useful to review the treatment of pressure gradients at the interface proposed by Popinet and Zaleski [52] and also the recent work of Shirani *et al*. [53] on pressure calculations based on interface location (PCIL).

In order to simulate additional practical flows, the proposed two-dimensional FVM implementation of the volume tracking method would have to be extended to two-dimensional axisymmetric and three-dimensional formulations. For the three-dimensional formulation, extensive modifications would be required, but guidance can be obtained from existing three-dimensional formulations [25, 35, 48, 54]. The interface in each cell would be represented by a plane, rather than a line segment. Thus, the circle-fit technique would, in principle, be modified into a sphere fit technique, in which a sphere is passed through the midpoints of a reference plane and *three* neighbour planes. It would also be beneficial to implement the proposed volume tracking method in CVFEMs [32, 33], so as to enable simulations in complex geometries.

ACKNOWLEDGEMENTS

The authors wish to thank the Natural Sciences and Engineering Research Council of Canada for its financial support of this work.

REFERENCES

1. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 1965; **8**:2182–2189.
2. DeBar R. Fundamentals of the KRAKEN code. *Technical Report UCIR-760*, Lawrence Livermore National Laboratory, 1974.
3. Noh WF, Woodward PR. SLIC (simple line interface calculation). *Lecture Notes in Physics* 1976; **59**:330–340.
4. Hirt CW, Nichols BD. Volume of Fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
5. Youngs DL. Time-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*, Morton KW, Baines MJ (eds). Academic Press: New York, 1982; 273–285.

6. Hyman JM. Numerical methods for tracking interfaces. *Physica D* 1984; **12**:396–407.
7. Chorin AJ. Curvature and solidification. *Journal of Computational Physics* 1985; **57**:472–490.
8. Fyfe DE, Oran ES, Fritts MJ. Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh. *Journal of Computational Physics* 1988; **76**:349–384.
9. Floryan JM, Rasmussen H. Numerical methods for viscous flows with moving boundaries. *Applied Mechanics Reviews* 1989; **42**:323–341.
10. Ashgriz N, Poo JY. FLAIR: flux line-segment model for advection and interface reconstruction. *Journal of Computational Physics* 1991; **93**:449–468.
11. Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics* 1992; **100**:25–37.
12. Brackbill JU, Kothe DB, Zemach C. A continuum method for modelling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.
13. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.
14. Tsai WT, Yue DKP. Computation of nonlinear free-surface flows. *Annual Review of Fluid Mechanics* 1996; **28**:249–278.
15. Kothe DB. Perspective on Eulerian finite volume methods for incompressible interfacial flows. In *Free Surface Flows*, Kuhlmann HC, Rath HJ (eds). Springer: Berlin, 1997; 267–331.
16. Pilliod JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Technical Report LBNL-40744*, Lawrence Berkeley National Laboratory, 1997.
17. Rudman M. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
18. Mosso SJ, Swartz BK, Kothe DB, Farrell RC. A parallel, volume-tracking algorithm for unstructured meshes. In *Parallel Computational Fluid Dynamics*: *Algorithms and Results Using Advanced Computers*, Schiano P, Ecer A, Periaux J, Satofuka N (eds). Elsevier: Amsterdam, 1997; 368–375.
19. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
20. Bugg JD, Mack K, Rezkallah KS. A numerical model of Taylor bubbles rising through stagnant liquids in vertical tubes. *International Journal of Multiphase Flow* 1998; **24**:271–281.
21. Wu J, Yu ST, Jiang BN. Simulation of two-fluid flows by the least-squares finite element method using a continuum surface tension model. *International Journal for Numerical Methods in Engineering* 1998; **42**: 583–600.
22. Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* 1999; **31**:567–603.
23. Gao DM. A three-dimensional hybrid finite element-volume tracking model for mould filling in casting processes. *International Journal for Numerical Methods in Fluids* 1999; **29**:877–895.
24. Gueyffier D, Li J, Nadim A, Scardovelli R, Zaleski S. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics* 1999; **152**:423–456.
25. Pilliod JE. An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods. *Master's Thesis*, University of California, Davis, 1992.
26. Tamamidis P, Assanis DN. Evaluation of various high-order-accuracy schemes with and without flux limiters. *International Journal for Numerical Methods in Fluids* 1993; **16**:931–948.
27. Yeh D, Yeh GT. Computer evaluation of high order numerical schemes to solve advective transport problems. *Computers and Fluids* 1995; **24**:919–929.
28. Lafaurie B, Nardone C, Scardovelli R, Zaleski S, Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics* 1994; **113**:134–147.
29. Rudman M. A volume-tracking method for incompressible multifluid flows with large density variations. *International Journal for Numerical Methods in Fluids* 1998; **28**:357–378.
30. Williams MW, Kothe DB, Puckett EG. Convergence and accuracy of kernel-based continuum surface tension models. *Technical Report LA-UR-98-2268*, Los Alamos National Laboratory, 1998.
31. Baliga BR, Patankar SV. A control-volume finite element method for two-dimensional fluid flow and heat transfer. *Numerical Heat Transfer* 1983; **6**:245–261.
32. Masson C, Saabas HJ, Baliga BR. Co-located equal-order control-volume finite element method for two-dimensional axisymmetric incompressible fluid flow. *International Journal for Numerical Methods in Fluids* 1994; **18**:1–26.
33. Baliga BR. Control-volume finite element methods for fluid flow and heat transfer. In *Advances in Numerical Heat Transfer*, Minkowycz WJ, Sparrow EM (eds). Taylor & Francis: Bristol, PA, 1997; 97–135.
34. Batchelor GK. *An Introduction to Fluid Dynamics*. Cambridge University Press: Cambridge, MA, 1967.
35. Kothe DB, Rider WJ, Mosso SJ, Brock JS, Hochstein JI. Volume tracking of interfaces having surface tension in two and three dimensions. *Technical Report AIAA 96-0859*, AIAA, 1996.
36. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing: Washington, DC, 1980.
37. Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics*. Springer: Berlin, 1996.

38. Peric M, Kessler R, Scheuerer G. Comparison of finite-volume numerical methods with staggered and colocated grids. *Computers and Fluids* 1988; **16**:389–403.
39. Mosso SJ, Swartz BK, Kothe DB, Clancy SP. Recent enhancements of volume tracking algorithms for irregular grids. *LANL Report LA-CP-96-227*, Los Alamos National Laboratory, 1996.
40. Puckett EG, Almgren AS, Bell JB, Marcus DL, Rider WJ. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *Journal of Computational Physics* 1997; **130**:269–282.
41. Garrioch SH. A volume tracking method for the simulation of two-fluid flows. *Ph.D. Thesis*, McGill University, 2001.
42. Garrioch SH, Baliga BR. A PLIC volume tracking method using circle-fit segment orientation. *4th International Conference on Multiphase Flow*, paper #387. New Orleans, LA, 2001.
43. Conte SD, de Boor C. *Elementary Numerical Analysis*. McGraw-Hill: New York, 1972.
44. Kothe DB, Mjolsness RC. RIPPLE: a new model for incompressible flows with free surfaces. *AIAA Journal* 1992; **30**:2694–2700.
45. Aleinov I, Puckett EG. Computing surface tension with high-order kernels. *Proceedings of the 6th International Symposium on Computational Fluid Dynamics*, Lake Tahoe, CA, 1995.
46. Zhuang WH, Etemadi K, Benenson DM, Ashgriz N. Sensitivity of the spreading rate of a liquid droplet on a surface to the approximation of the contact angle. *International Journal of Modelling and Simulation* 1999; **19**:184–193.
47. Rider WJ, Kothe DB. Stretching and tearing interface tracking methods. *Technical Report AIAA 95-1717*, AIAA, 1995.
48. Kothe DB, Williams MW, Lam KI, Korzekwa DR, Tubesing PK, Puckett EG. A second-order accurate, linearity-preserving volume tracking algorithm for free surface flows on 3-D unstructured meshes. *Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference*, FEDSM99-7109, San Francisco, 1999.
49. Rayleigh L. *Proceedings of the Royal Society of London* 1879; **29**:71–97.
50. Ginzburg I, Wittum G. Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants. *Journal of Computational Physics* 2001; **166**:302–335.
51. Renardy Y, Renardy M. PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of Computational Physics* 2002; **183**:400–421.
52. Popinet S, Zaleski S. A front-tracking algorithm for accurate representation of surface tension. *International Journal for Numerical Methods in Fluids* 1999; **30**:775–793.
53. Shirani E, Ashgriz N, Mostaghimi J. Interface pressure calculation based on conservation of momentum for front capturing methods. *Journal of Computational Physics* 2005; **203**:154–175.
54. Bussmann M, Chandra S, Mostaghimi J. Modeling the splash of a droplet impacting a solid surface. *Physics of Fluids* 2000; **12**:3121–3132.

