A VOF-Based Conservative Interpolation Scheme for Interface Tracking (CISIT) of Two-Fluid Flows

Yeng-Yung Tsui [a] & Shi-Wen Lin [a]

[a] Department of Mechanical Engineering , National Chiao Tung University , Hsinchu , Taiwan , Republic of China
Published online: 15 Apr 2013.

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis
Taylor & Francis Group

A VOF-BASED CONSERVATIVE INTERPOLATION SCHEME FOR INTERFACE TRACKING (CISIT) OF TWO-FLUID FLOWS

Yeng-Yung Tsui and Shi-Wen Lin

Department of Mechanical Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China

A simple interface-reconstruction scheme (CISIT), based on the volume-of-fluid (VOF) formulation and applicable to unstructured grids with arbitrary topology, is developed to track the interface in free-surface flows. The interface is represented by the contour surface of VOF value f = 0.5, which is constructed simply by interpolation. The advancing of the interface is made by imposing mass conservation in the cells containing the interface. However, problems arise when the interface moves across the cell face and into another cell. The cell originally occupied by the interface may be overfilled (f > 1) or underfilled (f < 1) as the interface advances (moving in the positive direction of the interface) to leave the cell. In contrast, the cell may be overdepleted (f < 0) or underdepleted (f > 0) as the interface retreats (moving in the negative direction of the interface) to move out of the cell. These situations are remedied via adjusting the transported volume of fluid across the cell face by following the conservation law so that the VOF becomes 1 or 0 in these cells, depending on advancing or retreating of the interface. The resulting VOF distribution is uniform, either in 1 or 0, in the region outside the interface cells, and the interface occupies only one cell in its width. To smooth the fluid properties across the interface region in the velocity calculation, an average smoothing technique is adopted. Application to a number of test cases, including model cases with known velocities and a number of real flows, reveal that this method is accurate and robust.

INTRODUCTION

Multifluid or multiphase flow features an interface separating fluids with different fluid properties. Accurate prediction of this interfacial flow presents a challenging task in numerical simulation because there exists a jump in fluid density and viscosity across the interface. This jump phenomenon can be identified as a Heaviside (step) function in theory. Different from the high-speed compressible flow, in which the flow velocity, pressure, and temperature present a discontinuity across the shock

NOMENCLATURE

C	volumetric fluid flux		τ_{ij}	viscous stress
C_n	Courant number		ϕ	Cartesian velocity components
Eo	Eotvos number		Φ	level set function
f	volume fraction			
F	flux		**Subscripts**	
Fr	Froude number		D	downstream cell
g_i	gravitational acceleration		f	cell face
Mo	Morton number		fi	ith cell face
P	pressure		g	gas
r	gradient ratio		l	liquid
Re	Reynolds number		Ni	ith neighboring cell
S	surface area		P	primary cell
$\vec{S}$	surface vector		U	upstream cell
$\vec{T}$, T_i	surface tension force		UU	far upstream cell
$\vec{V}$, V_j	flow velocity vector		v	vertex of the primary cell
w	weighting factor		vi	ith vertex
$\gamma(r)$	flux limiter function			
$\vec{\delta}$	distance vector		**Superscripts**	
Δt	time-step size		c	convection
Δv	cell volume		d	diffusion
μ	fluid viscosity		n	new time step
ρ	fluid density		o	old time step
σ	surface tension coefficient		w	wetted area of the face

wave, these variables are continuous in interfacial flows. Direct employment of shock-capturing schemes may not be appropriate for these flows.

The geometric shape of the interface continues to change during fluid flow. The large surface distortion and surface breaking and merging rule out the use of Lagragian-grid methods in which grids are attached to the interface. An alternative is to adopt Eulerian grids (fixed grids) with a method to track the movement of the interface. Tryggvason and co-workers [1, 2] developed a front-tracking method to fulfill this job. A fixed grid is assigned to the velocity field and a grid of lower dimension to the fluid front, or the interface. The interface grid is composed of triangular elements for a 3-D surface or line segments for a 2-D curve. The fluid front moves in a Lagrangian manner, with the propagating velocity obtained by interpolation from the velocities on the Eulerian mesh. Since the interface grid may be greatly distorted in the front movement, re-meshing is required from time to time. An obvious shortcoming is that the masses of the fluids are not conserved during front propagation. In general, the coding of the method is rather complicated due to the interaction between the interface grid and the Eulerian grid, along with the requirement of re-meshing for the former.

Another method to track the interface movement is based on the use of a level function defined as a signed distance away from the interface. The interface is represented by the set of zero-level value points. Therefore, it was called the level set method by Osher and Sethian [3]. Different from the above front-tracking method, the advancing of the interface is made by solving the advection equation for the level set function Φ on an Eulerian grid as

$$\frac{\partial \Phi}{\partial t} + \vec{V} \cdot \nabla \Phi = 0 \tag{1}$$

where $\vec{V}$ is the velocity of the flow. Hence, it is much easier to implement than the former one. A drawback is that it is prone to numerical errors in solving the hyperbolic equation when the interface experiences severe stretching or tearing. Similar to the explicit tracking method described above, the major weakness of the method lies in the lack of mass conservation, especially when the interface is highly distorted. One of the sources that cause mass nonconservation is that the level set function will no longer be a distance function after some period of time [4]. Therefore, a redistancing or reinitialization becomes necessary to maintain it as a distance function after each time step. It has been shown that after the reinitialization process is done, the error of mass loss or gain is largely reduced [5, 6]. Another tactic to reduce the mass conservation problem is to reduce spatial discretization errors using high-order schemes together with adaptive grid technology [7].

Another class of methods using Eulerian grids is based on the volume-of-fluid (VOF) function, which is defined as the fraction of volume in the mesh cell occupied by one of the two fluids. The VOF function f is either unity or zero in the cells containing a single fluid and less than 1 and greater than 0 in the cells where the interface is located. Its evolution follows the same form given in Eq. (1).

$$\frac{\partial f}{\partial t} + \vec{V} \cdot \nabla f = 0 \tag{2}$$

Introducing the continuity equation, it can be cast into a conservation form.

$$\frac{\partial f}{\partial t} + \nabla \cdot (\vec{V} f) = 0 \tag{3}$$

Although eqs. (1) and (2) are of the same form, it is more difficult to solve the equation for the VOF function than that for the level set function due to the former being a step function while the latter is continuous and smooth. A variety of schemes have been developed in the past, which, in general, can be classified into two categories. One is to solve the VOF in a way similar to the shock capturing for supersonic flows. It is essential for this kind of method to limit numerical diffusion and dispersion which occur in the region near the jump. It is known that the first-order upwind scheme is most stable but causes large diffusion, while the first-order downwind scheme has negative artificial viscosity. Although the downwind scheme is unstable, the antidiffusive character tends to compress the jump region to make it sharp. Hirt and Nichols [8] were the first to combine the upwind and downwind schemes in such a way as to minimize diffusion in interfacial flow calculations. Rudman [9] adopted a method based on the flux-corrected transport algorithm [10]. It consists of two stages. In the first stage, the VOF value is calculated using the upwind scheme. This is followed by a correction stage in which the downwind scheme is used to reduce the diffusion generated in the first stage. High-order schemes have also been introduced, including the HRIC of Muzaferija and Peric [11], CICSAM of Ubbink and Issa [12], STACS of Darwish and Moukalled [13],

and the latest FBICS of Tsui and Wu [14]. These schemes are based on the NVD formulation and satisfy the convection boundedness criteria [15]. They blend a high-resolution (bounded high-order) scheme with a compressive (bounded-downwind) scheme. The blending strategy is to switch from the high-resolution scheme to the compressive scheme in a continuous manner, depending on the angle between the interface and the grid lines. Another merit of these schemes is that they are developed with the aim of application to unstructured grids. The width of the resulting interface can be restricted to a region of several mesh sizes.

In another category of VOF method, the interface is approximately reconstructed using VOF information in local cells. The fluid fluxes crossing the faces of these cells are then determined with the help of the interface geometry in the cells. A simple way is to reconstruct the interface such that it is aligned with the grid line. In the SLIC algorithm of Noh and Woodward [16], this approximate interface is assumed to be vertical or horizontal, depending on the sweep direction of the direction-splitting solution method employed in this algorithm. A higher-order representation, introduced by Young as PLIC [17], is to approximate it by an oblique line segment. Comparing with the SLIC, the PLIC is more accurate, but more difficult to implement. To construct the surface using the available volume fraction and its gradient (surface normal vector) in local cells, the use of iterative procedures is most common [18, 19]. It is further complicated by having a large number of possible interface configurations that need to be considered for determination of fluid fluxes through the cell faces. Different from the PLIC, in which the interface is fitted inside the cell, the interface represented by a line segment is drawn across the face boundary in the FLAIR algorithm [20]. The linear equation of the approximate surface is determined by the volume fractions of the two cells on either side of the face. The fluid flux through this face during a time step can then easily be obtained by integration. This algorithm was successfully applied to 2-D and axisymmetric flow problems [21, 22]. The major limitation of this technique is that it cannot be extended to be applicable to 3-D problems.

The above volume-tracking methods have become widely used, partly due to their capability to handle complex free-surface flow problems in a relatively easy way. The accuracy of the interface-reconstruction VOF methods is highly dependent on the calculation of the interface normal and curvature from volume fractions, which is difficult to compute accurately because of the discontinuity property of this function. In contrast, the level set function is continuous and smooth. The derivatives of this smooth function can be easily calculated with high-order accuracy. However, as noted, the fluid mass is not conserved. To take advantage of the merits of the VOF and level set methods, it is natural to combine the two to form a coupled method [23–27]. In these kinds of methods, the interface normal and curvature are evaluated from the level set function and the evolution of the surface is made using the VOF to enforce mass conservation.

The PLIC algorithm is one of the most popular methods used in many applications. However, the rebuilding of the interface from the VOF is not straightforward even for rectangular grids. In addition, as required for the fluxing across cell faces, it is necessary to consider 16 interface configurations for 2-D flows and 64 configurations for 3-D flows, though these numbers of configurations can be reduced after regulating the cell geometry [28]. The extension of the method for unstructured meshes

with irregular cell geometry is not easy, partly due to the increased difficulty in reconstructing the interface and also due to the increased complexity in estimating the flow fluxes. To circumvent the latter problem, a Lagrangian-Eulerian advection algorithm was introduced by Mosso et al. [29]. In this method, the Eulerian grid cells containing the interface are moved in the Lagrangian sense. This is followed by reconstructing the interface in these cells because they are deformed during advection. The last step is to remap the volume fraction from the Lagrangian cells to the original Eulerian cells. This method has been applied to triangular grids by Shahbazi et al. [30] and Yang et al. [26]. Instead of the mesh cell, it is the fluid portion in the cell being advected in the study by Ashgriz et al. [31].

In view of the above survey, methods that are more accurate or more robust become more complicated. In this study, a simple method, within the framework of the VOF, is developed. It is robust because it is applicable to unstructured grids with arbitrary geometry and equally applicable to 3-D problems without causing any further complication. The interface is based on the contour surface of volume fraction 0.5, which can be easily constructed by interpolation. The advection step is fulfilled in a prediction-and-correction way. In the following, details of this solution method are described first. This is followed by a brief description of the method for the velocity field. Then, fidelity of the algorithm and verification of the accuracy are presented. Finally, concluding remarks are given.

SOLUTION METHOD FOR VOF

The present solution method is inspired by the work of Yeh [32, 33], in which the interface is represented by the contour line $f = 0.5$. In Yeh's study, the VOF value is originally stored at the centroid of each mesh cell. To calculate the fluid flux, the VOF is sought on grid nodes where velocities are located. The VOF on the grid node is first obtained using interpolation from the surrounding cells which share this node as a common vertex. The nodal value is then reset to 1 if the volume fraction is greater than 0.5 and to 0 if it is less than 0.5. The advection equation is solved using the finite-element method. An obvious shortcoming of this method is that the VOF is not conserved. A conservative algorithm is proposed in the following.

Interface Reconstruction

In the solution, the contour surface, or contour line in 2-D problems, of $f = 0.5$ is used as an indicator of the interface. It can easily be reconstructed by an interpolation practice. The original values of f are stored at the centroid of each grid cell. A linear weighting is employed to find f_v at each cell vertex,

$$f_v = \frac{\sum_j w_j f_j}{\sum_j w_j} \qquad (4)$$

where the subscript j denotes the cells surrounding the grid node and w_j is the weighting factor corresponding to the cell j. The calculation of the weighting factor can be

based on either the inverse of the volumes of the surrounding cells or the inverse of the distances between this node and those cell centroids.

After the nodal VOFs are obtained for the two end vertices of each edge of a cell, it is checked whether one of them is less than and the other greater than the value 0.5. If this is true, the interface intersects with this edge. The intersecting point, at which the VOF is 0.5, can then be determined by linear interpolation from the two nodes. After examining all the edges, the interface in the cell can be determined. This procedure proceeds in each cell of the mesh. In the end, a continuous piecewise-linear interface is constructed. This process is equally applicable to 2-D and 3-D cells.

Time Advancing

For a cell in a Eulerian mesh, the VOF Eq. (3) in the conservation form can be discretized using the finite-volume method as

$$\frac{\Delta V}{\Delta t}(f_P^n - f_P^o) + \sum_j F_{fj} = 0 \tag{5}$$

Here the superscripts n and o denote the new and old time steps, respectively, the subscript P designates the considered cell, the subscript f indicates a face of the cell, F_f is the flux through this face, and the subscript j denotes the face number. The summation is over all the surrounding faces of the cell. The scheme being explicit or implicit depends on whether the fluxes are evaluated using new time-level or old time-level values. In the following, the explicit scheme is employed. It is noted that the advection Eq. (2) rewritten in the following form was used for discretization in some studies.

$$\frac{\partial f}{\partial t} + \nabla \cdot (\vec{V}f) = (\nabla \cdot \vec{V})f \tag{6}$$

The reason for using this nonconservation expression is that the velocities on the cell faces do not satisfy the continuity constraint in these methods. The term on the right-hand side simply eliminates this error. For the method employed in this study, mass conservation is strictly obeyed in each cell.

In interface-capturing methods, the VOF flux is approximated as

$$F_f = C_f f_f = (\vec{V}_f \cdot \vec{S}_f)f_f \tag{7}$$

where $\vec{V}_f$ and $\vec{S}_f$ are the velocity and surface vectors at the face, respectively, C_f is the volumetric fluid flux, and f_f is the fluxing value of the VOF. By these algorithms, the issue of controlling numerical diffusion and dispersion becomes most important, which depends on the approximate estimation of the fluxing value f_f using the VOFs of neighboring cells [11–14].

In the present interface-reconstruction method, the face flux is the volumetric flux through the portion of the face wetted by the fluid:
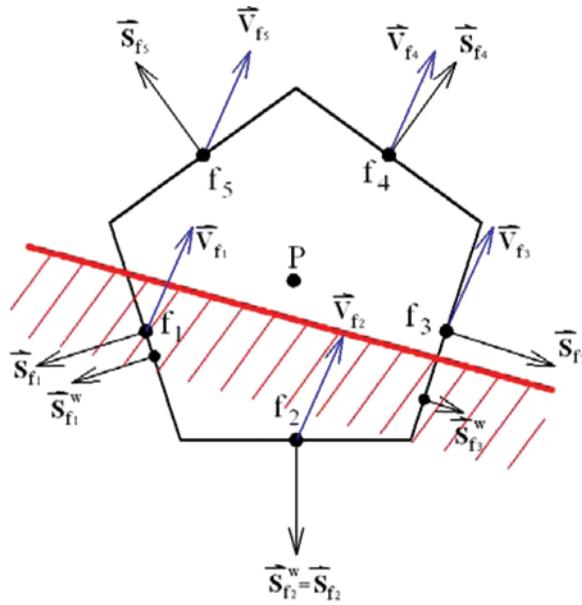
Figure 1. A control volume partially filled with fluid (color figure available online).

$$F_f = \vec{V}_f \cdot \vec{S}_f^w \tag{8}$$

where $\vec{S}_f^w$ represents the wetted area on the face. As an illustration, a 2-D cell is shown in Figure 1. The intersecting of the interface with the edges of the cell leads to three wetted areas. These wetted areas function as inlets or outlets to the cell for the fluid. The conservation law applied to this control volume results in the following equation:

$$\frac{\Delta V}{\Delta t}(f_P^n - f_P^o) + \sum_j \vec{V}_{fj} \cdot \vec{S}_{fj}^w = 0 \tag{9}$$

It can be seen that there is no need to estimate the face value f_f, as required in Eq. (7), and the calculation is undertaken only in the cells where the interface is located. Thus, numerical diffusion does not prevail in this method.

Cell Adjustment

It is ideal for a simulation based on VOF methods that the interface region covers a cell width and the VOF is either zero or one in the rest of the regions. The solution of the above time-marching step does not meet this ideal situation. In general, four kinds of problems may be encountered which occur when the interface leaves a cell and moves into another cell.

(a) 1-D flow   (b) 2-D flow
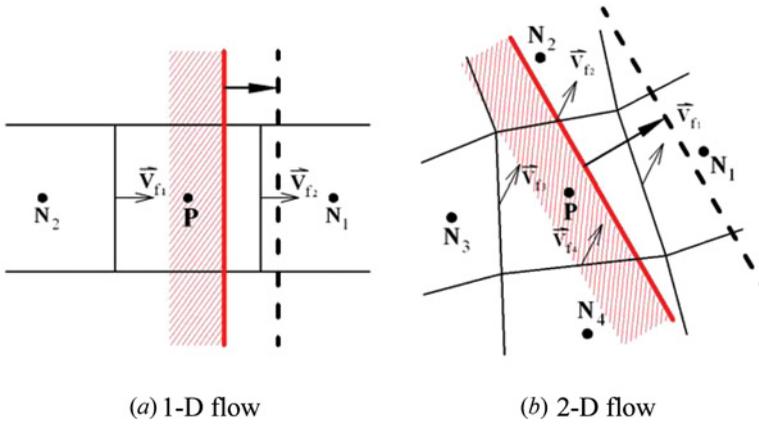
Figure 2. Advancing of interface through the face of a control volume (color figure available online).

1. Overfilling ($f_p > 1$). After solving Eq. (9), it is possible for the value of f_P^n to be greater than 1. This situation can be identified from Figure 2a, in which the interface is advancing to the right in a 1-D flow. In the beginning, the VOF value in the cell increases with the advancement of the interface, because fluid flows into the cell from the left face and no fluid leaves from the right face. By the time the interface crosses the right face, the VOF becomes greater than 1 because the wetted area on the right face remains zero. To force the interface to move across the cell face, the excessive fluid $\Delta f (= f_P - 1)$ must be reallocated to the downstream cell N1. For a multidimensional flow, as shown in Figure 2b, the excess must be portioned and assigned to the two downstream cells N1 and N2. The weighting factor w_{Ni} for the portioning is based on the ratio of the efflux through the corresponding face to the total efflux:

$$w_{N1} = \frac{C_{f1}}{C_{f1} + C_{f2}} \quad w_{N2} = \frac{C_{f2}}{C_{f1} + C_{f2}} \tag{10}$$

where $C_{fi} (= \vec{V}_{fi} \cdot \vec{S}_{fi})$ is the volumetric fluid flux through the face f_i. The formulation for the weighting factor can be written in a general form, which can be applied to all neighboring cells:

$$w_{Ni} = \frac{\max(C_{fi}, 0)}{\sum_j \max(C_{fj}, 0)} \tag{11}$$

Here the sum is over all the faces. It is obvious that the weighting factor is zero for upstream cells. The VOFs for the downstream cells are corrected by

$$f_{Ni} = f_{Ni} + w_{Ni}(f_P - 1)\frac{\Delta v_P}{\Delta v_{Ni}} \tag{12}$$

(a) 1-D flow   (b) 2-D flow
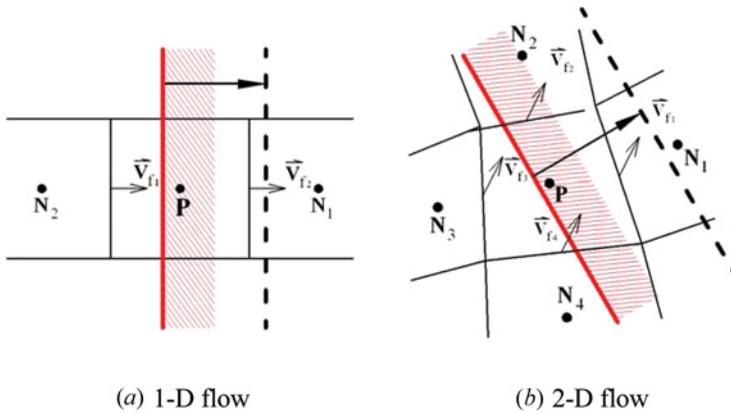
Figure 3. Retreating of interface through the face of a control volume (color figure available online).

After the neighboring cells are adjusted, the VOF for this overfilling cell is reset to 1.

2. Overdepleting ($f_p < 0$). When the interface retreats from a cell, as seen in Figure 3a in a 1-D flow, the VOF becomes less than zero at the time of crossing the right face. The overdepleted volume of fluid must be retrieved from the downstream cell N1. Similar to the overfilling situation, the overdepleted volume needs to be portioned and reallocated for multidimensional problems such as the one shown in Figure 3b. The weighting factor is given by Eq. (11). The VOFs for the neighboring cells are adjusted in the following way:

$$f_{Ni} = f_{Ni} + w_{Ni} f_P \frac{\Delta v_P}{\Delta v_{Ni}} \tag{13}$$

Afterwards, zero value is assigned to f_P.

3. Underfilling ($f_p < 1$). A shear flow has large velocity gradients in the transverse direction. In such flows, different from overfilling, the value of VOF may become less than 1 when the interface advances to leave the cell (Figure 2b). Since the interface does not lie in this cell any more, the VOF value remains less than 1 in later time. Hence, whenever it is found that $f_P < 1$ while the interface moves out of this cell, i.e., $f_{vi} > 0.5$ (f_{vi} represents the VOFs of all the cell vertices), fluid must be retrieved from the downstream cells to fill this cell so that f_P is equal to 1. The weighting factor and the adjustment of the VOF are the same as those shown in Eqs. (11) and (12).

4. Underdepleting ($f_p > 0$). Another situation encountered in high-shear flows is that the VOF may remain greater than zero when the interface retreats to leave the cell (Figure 3b). Similar to the underfilling cell, the volume of fluid remains unchanged and the fluid in the cell becomes stagnant. Therefore, whenever it is found that $f_P > 0$ and the interface is not situated in this cell any more ($f_{vi} < 0.5$ for all cell vertices), fluid in this cell must be allocated to the downstream cells so that

f_P becomes zero. The weighting factor and the adjustment are given by Eqs. (11) and (13).

The solution procedure described above is summarized as follows.

1. The interface is reconstructed first using the VOF from the last time step.
2. New values of VOF are sought via conserving fluid mass in interface cells.
3. The VOFs are adjusted to ensure that no cells are overfilled, underfilled, overdepleted, or underdepleted. It is noted that after the correction process for underfilling and underdepleting cells, it is possible for some of the neighboring cells to be overfilled or overdepleted. More similar correction steps can then be undertaken. Usually, two sweeps of such adjustment are enough to wipe out this problem.

This completes the calculation of the VOF in one time step. It can be seen that the conservation law is obeyed in each of the predictor and corrector steps. The resulting interface region is sharp and lies in one cell only.

SOLUTION METHOD FOR VELOCITY FIELD

In each time step of calculation, the VOF is obtained first, followed by solving for the velocity field. The flow in both fluids is assumed to be laminar and incompressible. The conservation of mass and momentum for the two fluids can be written in one set of equations:

$$\frac{\partial V_j}{\partial x_j} = 0 \tag{14}$$

$$\frac{\partial \rho V_i}{\partial t} + \frac{\partial}{\partial x_j}(\rho V_j V_i) = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i + T_i \tag{15}$$

where V_j is the velocity, ρ the density, P the pressure, τ_{ij} the viscous stress, g_i the gravitational acceleration, and T_i the force due to surface tension. Surface tension is a surface force at the free interface in two-phase flows. With the CSF (continuum surface force) formulation [34], it can be modeled as a body force:

$$\vec{T} = -\sigma \nabla \cdot \left(\frac{\nabla f}{|\nabla f|}\right) \nabla f \tag{16}$$

where σ is the surface tension coefficient.

Smoothing of Fluid Properties

In the one-equation model for two-fluid flows, the fluid properties, such as density and viscosity, are expressed as functions of the local volume fraction:

$$\rho = f\rho_1 + (1-f)\rho_2 \tag{17a}$$

$$\mu = f\mu_1 + (1 - f)\mu_2 \tag{17b}$$

where the subscripts 1 and 2 denote the two fluids. When the difference between the fluid properties, especially the density, of the two fluids is large, the sharp gradient across the interface region may cause numerical instability. To alleviate this problem, the gradient of the properties needs to be smeared artificially. Various smoothing techniques can be found in [5, 35, 36]. In this study, an averaging smoother is used. First, the property values at all cell vertices are calculated using the interpolation method given by Eq. (4). A new value at the cell centroid is then obtained by averaging over these vertices. In this manner, the interface region is expanded to the two adjacent cells on either side of the interface cell. This smoothing process can be repeated to enhance the smearing effects. In our calculations, no more than two such smoothing steps are taken.

Numerical Method

Discretization of the equations is performed using the finite-volume method. The convective momentum flux is expressed as

$$F^c = (\rho_f \vec{V}_f \cdot \vec{S}_f)\phi_f \tag{18}$$

where ϕ designates the Cartesian velocity components. The face flux value ϕ_f is approximated using neighboring nodal values,

$$\phi_f = \phi_U + \frac{\gamma(r)}{2}(\phi_D - \phi_U) \tag{19}$$

where, as seen in Figure 4, the subscript U denotes the cell node upstream of the considered face and the subscript D is the one downstream. The symbol γ represents the flux limiter, which depends on the gradient ratio r defined by

$$r = \frac{\phi_U - \phi_{UU}}{\phi_D - \phi_U} \tag{20}$$
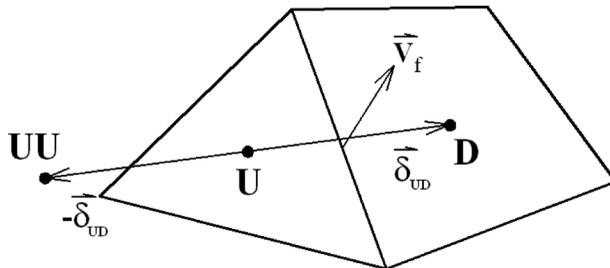
Figure 4. Illustration for determination of the far upstream node UU.

Here ϕ_{UU} is the value at a node far upstream (see Figure 4), estimated by

$$\phi_{UU} = \phi_D - 2(\nabla\phi)_U \cdot \vec{\delta}_{UD} \tag{21}$$

where $(\nabla\phi)_U$ denotes the gradient at the upstream node U and $\vec{\delta}_{UD}$ is the distance vector directing from U to D. A number of TVD and NVD schemes are available by assigning different expressions to the function $\gamma(r)$ [37, 38]. In the following calculations, the Van Leer scheme is adopted, which is given by

$$\gamma(r) = \frac{r + |r|}{r + 1} \tag{22}$$

The diffusive momentum flux, suitable for unstructured grids, is approximated by

$$F^d = \frac{\mu S_f^2}{\vec{\delta}_{PN} \cdot \vec{S}_f}(\phi_N - \phi_P) + \mu(\nabla\phi)_f \left(\vec{S}_f - \frac{S_f^2}{\vec{\delta}_{PN} \cdot \vec{S}_f} \vec{\delta}_{PN} \right) \tag{23}$$

where the subscripts P and N denote the considered cell and neighboring cell, respectively, $\vec{\delta}_{PN}$ is the vector connecting nodes P and N, and $(\nabla\phi)_f$ is the gradient at the face.

All variables, including all velocity components and pressure, are stored at the centroid of each grid cell. The coupling between momentum equations and the continuity equation is tackled using the noniterative technique of the PISO algorithm [39]. This is a kind of predictor-corrector method. First, the momentum equations are solved to obtain velocities at cell centroids in the predictor step using the prevailing velocity and pressure fields. A momentum interpolation method is then employed to find the velocities on the cell face and the mass flux through the face. A pressure-correction equation can be obtained by forcing the corrected mass flux to obey the conservation law in each cell. After the pressure correction is solved for, the pressure and velocity are upgraded accordingly. This adjustment procedure takes place in two corrector steps, after which both the momentum and continuity equations are better satisfied. The way to derive the face mass flux and the pressure-correction equation can be found in [40].

RESULTS AND DISCUSSION

The solution method for the VOF is first tested in two model problems with given velocities. One is the advection of a square or a circular cylinder in a uniform velocity field and the other is the advection of a circular cylinder in a shear flow. Real flows are then considered. The realistic flows selected for testing include collapse of a water column, rising of a gas bubble, and the Rayleigh-Taylor instability problem.

Advection of Cylinders in a Uniform Velocity Field

The computational domain is a unit square with a constant velocity $\vec{V} = u\vec{i} + v\vec{j}$ being imposed. Either a square cylinder or a circular cylinder is initially

placed at a position $x = 0.25$ and $y = 0.25$. The cylinder has a side of length (or diameter for the circle) 0.3. It is moved to the position corresponding to $x = 0.75$. The domain is partitioned into various rectangular and triangular grids. Theoretically, the geometry of the cylinders remains unchanged during translation by the uniform velocity field. To quantify the error caused by the numerical method, the normalized L^1 norm is used.

$$E = \frac{\sum \left| f_j^n \Delta v - f_j^e \Delta v \right|}{\sum f_j^i \Delta v} \tag{24}$$

where the superscript n designates the numerical solution, the superscript e the exact solution, and the superscript i the initial condition. The sum is over all the cells in the domain.

The L^1 errors for the case $u = v = 1$ calculated by using rectangular grids with $50 * 50$, $100 * 100$, and $200 * 200$ cells and various sizes of time step, corresponding to Courant numbers 0.1, 0.25, 0.5, and 0.75, are shown in Figure 5 for the square and circle translations. The Courant number is defined as

$$C_n = \frac{\sum \max(C_{fj}, 0)\Delta t}{\Delta v} \tag{25}$$

where C_{fj} is the volumetric flux through the face f_j and the sum is taken over all the faces of the cell. It is obvious that the error decreases as the grid size or time interval is reduced. In view of the figure, the accuracy of the calculation for the transport of the circle is generally higher than that for the square. Comparing the resulting shapes of the two cylinders given in Figure 6a (for $C_n = 0.5$) and Figure 6b (for $C_n = 0.1$) demonstrates this point. Large errors occur at the corners of the square where the slope is discontinuous. Wavy forms appear at the two corners on the upper left and lower right. This situation can be soothed by reducing the Courant number and grid spacing. For the case of circle the interface is much smoother. However,
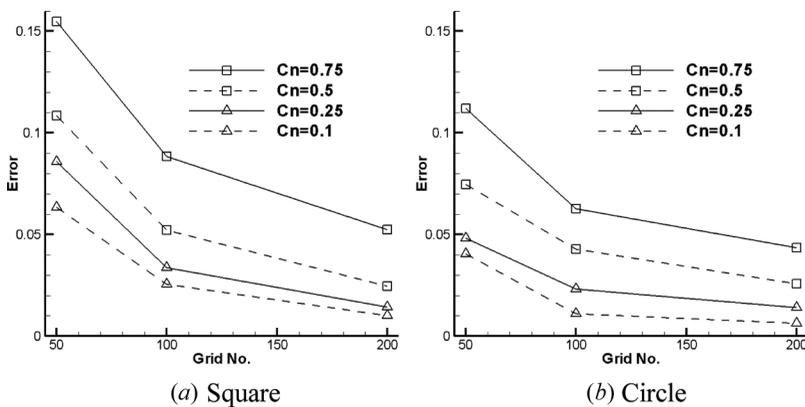
Figure 5. Numerical error against grid number for advection of cylinders in uniform flow.

(a) $C_n = 0.5$ (b) $C_n = 0.1$ (c) Triangular grid
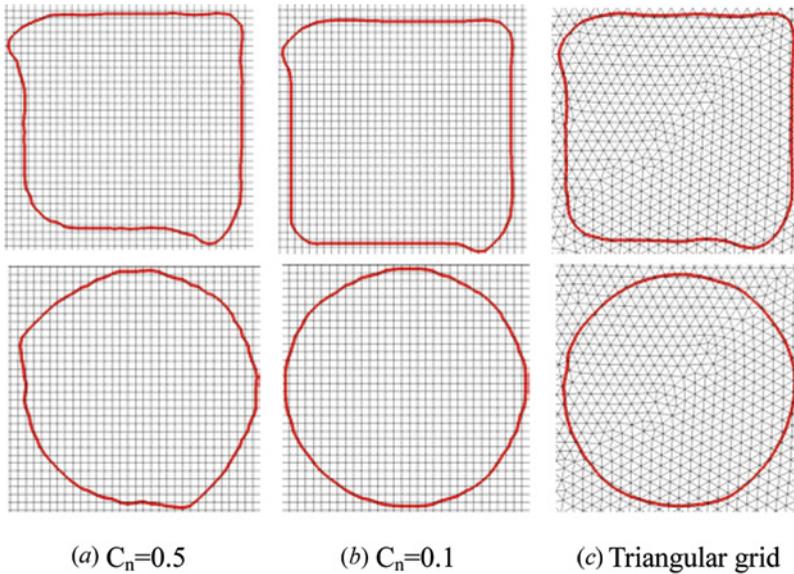
Figure 6. Resulting cylinders in uniform flow (color figure available online).

as seen from Figure 6a ($C_n = 0.5$), nonsmoothness still can be identified when the Courant number is large. It was mentioned that the present method is applicable to grids of arbitrary geometry. Figure 6c presents the results using a grid with 22,474 triangular cells.

Advection of a Circle in a Shear Flow

In real flows, the flow velocity is nonuniform. The interface is subject to straining and, thus, deforms continuously. To mimic this situation, the following velocities are assumed:

$$\vec{V} = \sin x \cdot \cos y \vec{i} - \cos x \cdot \sin y \vec{j} \tag{26}$$

A circle of diameter 0.4π is placed in a square with side length π. The initial center of the circle is located at $(\pi/2, (1+\pi)/5)$. It is advected by the shear flow in 16 units of time first, followed by reversing the velocity field for another 16 units of time. The circle will be stretched and deformed by the flow straining in the forward step and recovers its original shape at the end of the backward step.

In the calculation, the time step is determined in the way that the maximum cell Courant number is less than 0.1. The resulting geometries after the forward and backward steps are given in Figure 7. The circle is gradually stretched to become serpentine-like during the forward stage. With low grid densities the tail is shortened. It is broken and separated form the main body for the 50 * 50 grid. It can also be seen that the recovered circle becomes irregular for the lowest level of grid by the end of the backward stage. The prediction accuracy deteriorates by using triangular grids.

Rectangular grids			Triangular grids		
50*50	100*100	200*200	5626	22550	89776

Figure 7. Advection in shear flow. The results in the first row are obtained after the forward stage and those in the second row after the backward stage.

The tail of the serpentine becomes shorter and the circle is not so round even for the grid with 89,776 cells. The same flow problem has been tested by the present authors in [14] using the flux blending schemes FBICS and CICSAM along with a 100 * 100 rectangular grid. Comparing the results indicates that the recovered shape obtained by the present method is closer to a round circle.

Collapse of a Water Column

This case becomes a classical problem to test numerical methods for tracing a water front because experimental data are available in [41]. A water column 0.146 m in width and 0.292 m in height is initially placed at the left corner if a tank of size 0.584 m * 0.340 m. The large density difference between the two phases often causes computational instability. Without a smoothing activity, calculations simply fail unless the highly diffusive, first-order upwind difference scheme is used. Figure 8 presents the flow field obtained by using the smoothing procedure described in Section 3.1. It is clear that with only one smoothing sweep the flow on the gas side

(a)   *(b)*
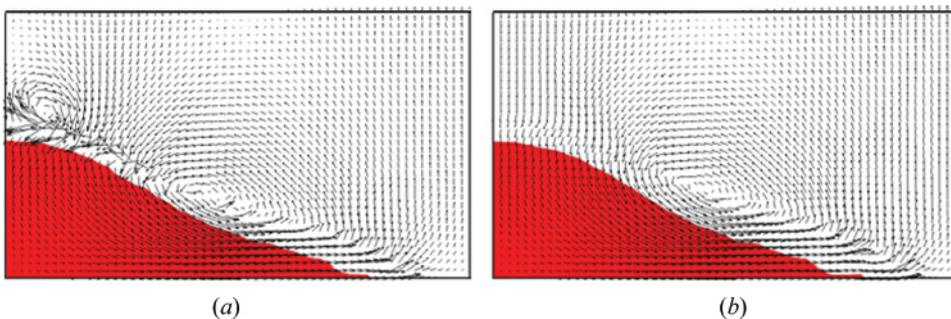
Figure 8. Flow field for the collapse of water column: (*a*) with one smoothing step; (*b*) with two smoothing steps (color figure available online).
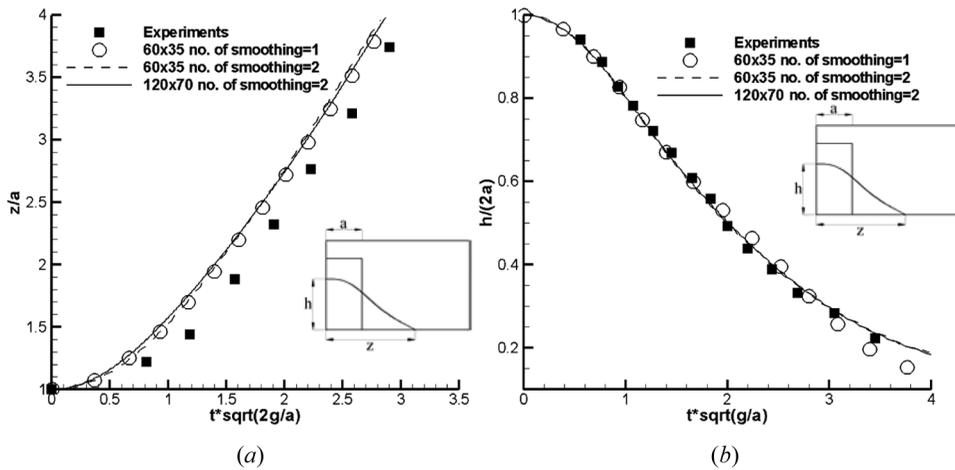
Figure 9. The collapse of water column: (*a*) position of the leading front edge along the floor; (*b*) height of the retreating front along the wall.

of the interface is unstable. The velocities vary in an irregular manner on the part of the interface where the water front is retreating and the gas flows toward it. A secondary vortex appears near the left boundary. These oscillating phenomena can be suppressed when one more smoothing step is taken, as seen in Figure 8*b*.

Comparison of the predicted interface position with experiments [41] is made in Figure 9. The one shown in Figure 9*a* is the edge position of the advancing leading front along the floor, which is slightly overpredicted. This may be attributed to the uncertainties in determining the exact location of the front edge in experiments. Such an overprediction can also be found in other studies [11, 14]. The agreement in terms of the height of the collapsing column along the side wall shown in Figure 9*b* is good. However, as is not surprising, the curve obtained by one smoothing step exhibits oscillating behavior.

Rising of a Gas Bubble

A circular gas bubble of diameter 0.01 m is released in a chamber of size 0.05 m * 0.15 m. The densities of the gas and liquid phases are assumed to be 1,000 and 1 kg/m^3 and the viscosities to be 5.56e-1 and 5.56e-4 Ns/m^2. Three different values of surface tension coefficient are under consideration: 9.79e-01, 9.79e-2, and 9.79e-3 N/m. This leads to Eotvos numbers 1, 10, and 100 and Morton numbers 0.001, 1, and 1,000. The Eotvos number is defined as $Eo = gd_e^2(\rho_l - \rho_g)/\sigma$ and the Morton number as $Mo = g\mu_l^4/\rho_l\sigma^3$, where the subscripts l and g denote the liquid and gas phases, respectively, and d_e is the initial bubble diameter. The evolution of the bubble pattern at t = 0.05, 0.15, and 0.25 s for the three cases is shown in Figure 10. It is recognized that the interfacial tension force tend to minimize the surface area of the bubble. Therefore, the bubble pattern remains nearly a circle when the Eotvos number is sufficiently small. It is transformed into the shape of a circular cap as Eo is enlarged. Further increase of this number brings about a crescent shape.

(*a*) E$_o$=1   (*b*) E$_o$=10   (*c*) E$_o$=100
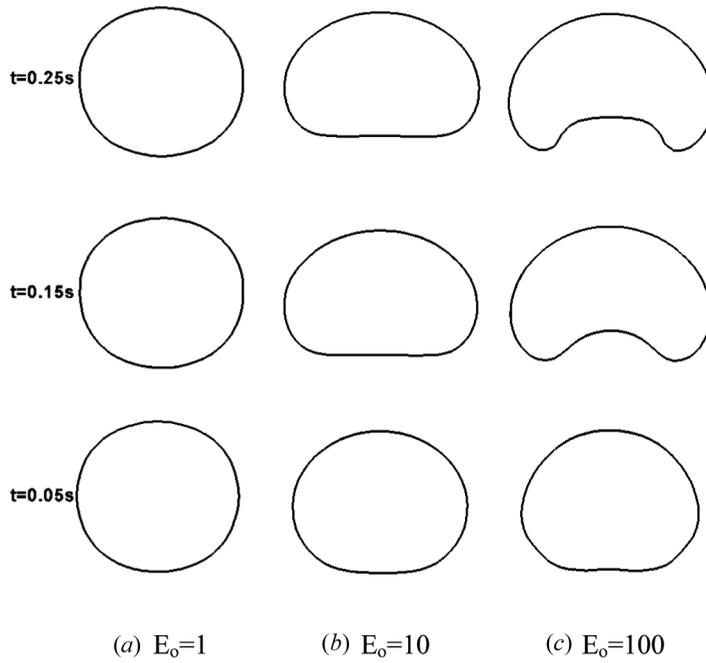
Figure 10. Evolution of the rising bubble for the three cases.

The time variation of the position of the bubble center is shown in Figure 11. The relation becomes linear after a short initial stage. The uprising velocity of the bubble, represented by the slope of the curve, is lower initially, followed by a gradual increase.
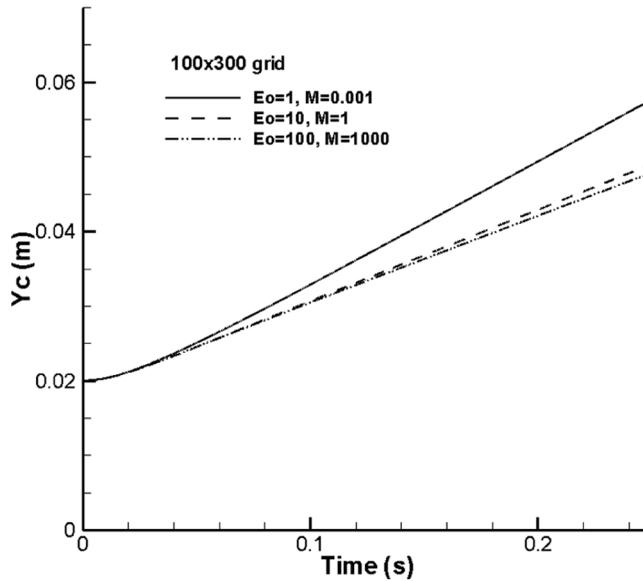
Figure 11. Center positions of rising bubbles.
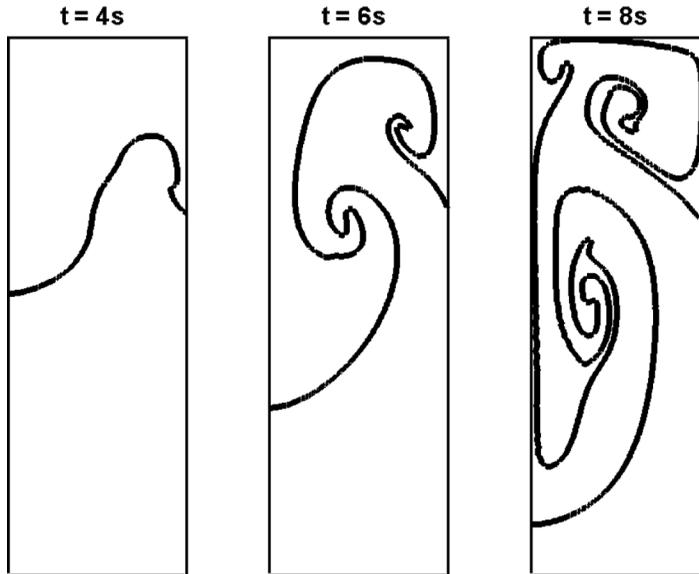
Figure 12. Evolution of Rayleigh-Taylor flow obtained using 96 * 288 grid.

Finally, it reaches a constant value in the linear stage, which is termed terminal velocity. The terminal velocities are 0.166, 0.12, and 0.115 m/s for the three cases. The gradual decrease of this speed with the Eotvos number is expected, because the deformation of the bubble results in an increase of drag.

Rayleigh-Taylor Instability

The computational domain is of dimensions 1×3. Initially, the top one-third is filled with a heavier fluid of density 1.2 and the lower two-thirds with another fluid of lower density 1.0. The governing equations are solved with the Froude number $[\text{Fr} = U/(gL)^{1/2}]$ set at 0.5 and the Reynolds number $(\text{Re} = \rho UL/\mu)$ at 500. Symmetric condition is specified on the left boundary, while no-slip condition on the other boundaries. A small perturbation of the form $0.02\cos(1 + x)\pi$ is imposed on the initial interface between the two fluids. The evolution of the unstable flow is shown in Figure 12 for times $t = 4$, 6, and 8 s. In the early stage, the perturbation is enlarged in the linear sense. Then the nonlinear interaction of the descending flow of the heavier layer and the ascending flow of the lighter layer results in a mushroom-like structure when time reaches 4 s. The flow patterns are further complicated by wrapping around the mushroom cap at t = 8 s. This flow field is very similar to the calculations of Rudman [9].

CONCLUSIONS

A conservative interpolation scheme for interface tracking (CISIT), within the framework of the VOF, has been introduced to overcome the difficulties encountered

at the interface embedded in two-fluid flows. The formulation of the method is very simple. It is easy to code and applicable to grids of arbitrary geometry without causing any further complication. The resulting distribution of fluid fraction is a good approximation to the Heaviside function. It has been seen from real-flow calculations that stable solutions are obtained with two sweeps of the smoothing process.

REFERENCES

1. S. O. Unverdi and G. Tryggvason, A Front-Tracking Method for Viscous, Incompressible, Multi-fluid Flows, *J. Comput. Phys.*, vol. 100, pp. 25–37, 1992.

2. G. Tryggvasson, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, A Front-Tracking Method for the Computations of Multiphase Flow, *J. Comput. Phys.*, vol. 169, pp. 708–759, 2001.

3. S. Osher and J. A. Sethian, Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.

4. M. Sussman, P. Smereka, and S. Osher, A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow, *J. Comput. Phys.*, vol. 114, pp. 146–159, 1994.

5. M. Sussman, E. Fatami, P. Smereka, and S. Osher, An Improved Level Set Method for Incompressible Two-Phase Flows, *Comput. Fluids*, vol. 27, pp. 663–680, 1998.

6. M. Sussman, and E. Fatami, An Efficient, Interface-Preserving Level Set Redistancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow, *SIAM J. Sci. Comput.*, vol. 20, pp. 1165–1191, 1999.

7. R. R. Nourgaliev, S. Wiri, N. T. Dinh, and T. G. Theofanous, On Improving Mass Conservation of Level Set by Reducing Spatial Discretization Errors, *Int. J. Multiphase Flow*, vol. 31, pp. 1329–1336, 2005.

8. C. W. Hirt and H. D. Nichols, Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, *J. Comput. Phys.*, vol. 39, pp. 201–225, 1981.

9. M. Rudman, Volume-Tracking Methods for Interfacial Flow Calculations, *Int. J. Numer. Meth. Fluids*, vol. 24, pp. 671–691, 1997.

10. S. T. Zalesak, Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids, *J. Comput. Phys.*, vol. 31, pp. 335–262, 1979.

11. S. Muzaferija, M. Peric, P. Sames, and T. Schellin, A Two-Fluid Navier-Stokes Solver to Simulate Water Entry, *Proc. 22th Symp. on Naval Hydrodynamics*, Washington, DC, pp. 638–649, 1998.

12. O. Ubbink and R. I. Issa, A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes, *J. Comput. Phys.*, vol. 153, pp. 26–50, 1999.

13. M. Darwish and F. Moukalled, Convective Schemes for Capturing Interfaces of Free-Surface Flows on Unstructured Grids, *Numer. Heat Transfer B*, vol. 49, pp. 19–42, 2006.

14. Y.-Y. Tsui, S.-W. Lin, T.-T. Cheng, and T.-C. Wu, Flux-Blending Schemes for Interface Capture in Two-Fluid Flows, *Int. J. Heat Mass Transfer*, vol. 52, pp. 5547–5556, 2009.

15. P. H. Gaskell and A. K. C. Lau, Curvature-Compensated Convective Transport: SMART, A New Boundedness-Preserving Transport Algorithm, *Int. J. Numer. Meth. Fluids*, vol. 8, pp. 617–641, 1988.

16. W. F. Noh and P. Woodward, SLIC (Simple Line Interface Calculation), *Lecture Notes Phys.*, vol. 59, pp. 330–340, 1976.

17. D. L. Youngs, Time-Dependent Multi-material Flow with Large Fluid Distortion, in K. W. Morton and M. J. Baines (eds.), *Numerical Methods for Fluid Dynamics*, vol. 24, pp. 273–285, Academic Press, New York, 1982.

18. W. J. Rider, and D. B. Kothe, Reconstructing Volume Tracking, *J. Comput. Phys.*, vol. 141, pp. 112–152, 1998.

19. J. E. Pilliod, Jr., and E. G. Puckett, Second-Order Accurate Volume-of-Fluid Algorithms for Tracking Material Interfaces, *J. Comput. Phys.*, vol. 199, pp. 465–502, 2004.

20. N. Ashgriz and J. Y. Poo, FLAIR: Flux Line-Segment Model for Advection and Interface Reconstruction, *J. Comput. Phys.*, vol. 93, pp. 449–468, 1991.

21. F. Mashayek and N. Ashgriz, Advection of Axisymmetric Interfaces by the Volume-of-Fluid Method, *Int. J. Numer. Meth. Fluids*, vol. 20, pp. 1337–1361, 1995.

22. F. Mashayek and N. Ashgriz, A Hybrid Finite-Element-Volume-of-Fluid Method for Simulating Free Surface Flows and Interfaces, *Int. J. Numer. Meth. Fluids*, vol. 20, pp. 1363–1380, 1995.

23. M. Sussman and E. G. Puckett, A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows, *J. Comput. Phys.*, vol. 162, pp. 301–337, 2000.

24. G. Son and N. Hur, A Coupled Level Set and Volume-of-Fluid Method for the Buoyancy-Driven Motion of Fluid Particles, *Numer. Heat Transfer B*, vol. 42, pp. 523–542, 2002.

25. S. P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling, A Mass-Conserving Level-Set Method for Modelling of Multi-Phase Flows, *Int. J. Numer. Meth. Fluids*, vol. 47, pp. 339–361, 2005.

26. X. Yang, A. J. James, J. Lowengrub, X. Zheng, and V. Cristini, An Adaptive Coupled Level-Set/Volume-of-Fluid Interface Capturing Method for Unstructured Triangular Grids, *J. Comput. Phys.*, vol. 217, pp. 364–394, 2006.

27. D. L. Sun and W. Q. Tao, A Coupled Volume-of-Fluid and Level Set (VOSET) Method for Computing Incompressible Two-Phase Flows, *Int. J. Heat Mass Transfer*, vol. 53, pp. 645–655, 2010.

28. R. Tavakoli, R. Babaei, N. Varahram, and P. Davami, Numerical Simulation of Liquid/Gas Phase Flow during Mold Filling, *Comput. Meth. Appl. Mech. Eng.*, vol. 196, pp. 697–713, 2006.

29. S. J. Mosso, B. K. Swartz, D. B. Kothe, and S. P. Clancy, Recent Enhancement of Volume Tracking Algorithm for Irregular Grids, Los Alamos Natl. Lab. Rep. LA-CP-96-227, Los Alamos, NM, 1996.

30. K. Shahbazi, M. Paraschivoiu, and J. Mostaghimi, Second Order Accurate Volume Tracking Based on Remapping for Triangular Meshes, *J. Comput. Phys.*, vol. 188, pp. 100–122, 2003.

31. N. Ashgriz, T. Barbat, and G. Wang, A Computational Lagrangian-Eulerian Advection Remap for Free Surface Flows, *Int. J. Numer. Meth. Fluids*, vol. 44, pp. 1–32, 2004.

32. J.-T. Yeh, Simulation and Industrial Application of Inkjet, *7th Natl. CFD Conf.*, Pingtung, Taiwan, 2000.

33. J.-T. Yeh, A VOF-FEM, and Coupled Inkjet Simulation, *Proc. ASME FEDSM*, New Orleans, LA, pp. 1–5, 2001.

34. J. U. Brackbill, D. B. Kothe, and C. Zemach, A Continuum Method for Modeling Surface Tension, *J. Comput. Phys.*, vol. 100, pp. 335–354, 1992.

35. C. S. Peskin, Numerical Analysis of Blood Flow in the Heart, *J. Comput. Phys.*, vol. 25, pp. 220–252, 1977.

36. J. J. Monaghan, Simulating Free Surface Flow with SPH, *J. Comput. Phys.*, vol. 110, pp. 399–406, 1994.

37. Y.-Y. Tsui and T.-C. Wu, A Pressure-Based Unstructured-Grid Algorithm Using High-Resolution Schemes for All-Speed Flows, *Numer. Heat Transfer B*, vol. 53, pp. 75–96, 2008.

38. Y.-Y. Tsui and T.-C. Wu, Use of Characteristic-Based Flux Limiters in a Pressure-Based Unstructured-Grid Algorithm Incorporating High-Resolution Schemes, *Numer. Heat Transfer B*, vol. 55, pp. 14–34, 2009.

39. R. I. Issa, Solution of the Implicitly Discretised Fluid Flow Equations by Operator-Splitting, *J. Comput. Phys.*, vol. 62, pp. 40–65, 1986.
40. Y.-Y. Tsui and Y.-F. Pan, A Pressure-Correction Method for Incompressible Flows Using Unstructured Meshes, *Numer. Heat Transfer B*, vol. 49, pp. 43–65, 2006.
41. J. C. Martin and W. J. Moyce, An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane, *Phil. Trans. Roy. Soc. Lond., Ser. A*, vol. 244, pp. 312–324, 1952.