Journal of Computational Physics **183,** 233–273 (2002)
doi:10.1006/jcph.2002.7196

# A High-Resolution Characteristics-Based Implicit Dual Time-Stepping VOF Method for Free Surface Flow Simulation on Unstructured Grids

Yong Zhao,* Hsiang Hui Tan,† and Baili Zhang‡

*Nanyang Technological University, School of MPE, Nanyang Avenue, Singapore 639798; †DSO National Labs, Singapore; and ‡Institute of High Performance Computing, Singapore
E-mail: myzhao@ntu.edu.sg

Received February 25, 2002; revised August 13, 2002

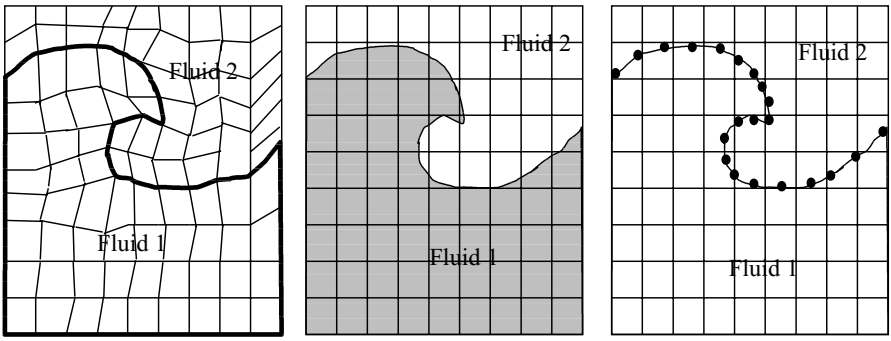
In this paper, a new unstructured-grid/finite volume incompressible Navier–Stokes solver, based on a high-order characteristics-based method, the artificial compressibility method, and a matrix-free implicit dual time-stepping scheme, has been developed to study unsteady free surface flows. The free surface effects are calculated using the volume of fluid (VOF) method and continuum surface force. The transport equation for the VOF is numerically solved using exactly the same method for the flow solver. A number of test cases have been studied to validate the proposed method and demonstrate its capability by comparison with other schemes and published results. 

## 1. INTRODUCTION

Simulation of free surface flows is difficult, primarily because boundary conditions are required on arbitrarily moving surfaces which have to be located in the flow field accurately over a long period of time. Generally, there are three approaches to compute free surface flows, namely: (i) the surface fitting method, (ii) the surface capturing method, and (iii) the surface tracking method. Schematic representations of these methods are given in Fig. 1. In the surface fitting method (Fig. 1a), the interface is tracked by attaching it to a mesh surface which is forced to move with the interface. This can be considered a Lagrangian mesh method. The surface capturing method (Fig. 1b) uses an indicator function to mark the fluids on both sides of the free surface. And the mesh remains fixed. Therefore it is a Eulerian mesh method. In the surface tracking method (Fig. 1c), the interface is represented and tracked explicitly by special marker points. The grid also remains fixed. Thus it can be considered a hybrid Eulerian–Lagrangian mesh method.



0021-9991/02 $35.00




(a) Surface Fitting Method (b) Surface Capturing Method (c) Surface Tracking Method

**FIG. 1.** Surface fitting, surface capturing, and surface tracking methods.

Surface fitting methods are efficient and accurate in the study of shaping processes of materials with a certain stiffness, such as extrusion and embossing. However, these methods rapidly become inappropriate, indeed impossible, to apply with less viscous materials that undergo large deformations. Mesh regenerations are necessary when mesh cells and elements become highly skewed, to prevent computational failures. This can make the surface fitting scheme become very complicated and inefficient. On the other hand, various surface tracking techniques are also efficient and accurate in locating the interface in the fluids because of the Lagrangian nature of the surface tracking. Implementation of the surface tension force is also straightforward given the location and curvature of the free surface. The limitation of these methods is their inability to cope naturally with folding or rupturing interfaces.

By comparison, the surface capturing method is relatively simple and versatile for computation of free surface flows because the indicator function can be updated by solving a convection equation and the interface can be treated as an internal surface. In addition no mesh regeneration is necessary. The most popular surface capturing methods include the marker and cell (MAC) method [3], the level set (LS) method [4] and the volume of fluid (VOF) method [1]. The MAC method, which allows arbitrary free surface flows to be simulated, is widely used and can be readily extended to three dimensions. Its drawback lies in the fact that it is computationally demanding to trace a large number of particles, especially in 3D simulation. In addition, it may result in some regions void of particles because the density of particles is finite. The LS method is another method which is also being widely used. It has been applied to the modeling of flame advection [4] and bubble and droplet motions [5, 6]. The level set function keeps its initial minimum and maximum values; thus the maximum value between two merging interfaces remains the same, causing a steep gradient and an impenetrable sheet between them. It is therefore necessary to reinitiate the distance function after each time step [5, 6].

The use of volume fractions to mark the fluid regions is adopted because of its efficiency and simplicity and the natural way in which complex interfaces can be treated as internal moving boundaries. The volume fractions are convected through the flow domain by solving a scalar convection equation. The method of using volume fractions is robust enough to handle the breakup and coalescence of fluid masses. Furthermore, because this method uses a continuous function it does not suffer from the lack of divisibility that discrete particles

exhibit. On the other hand, differencing schemes applied to the volume fraction convection equation can introduce numerical diffusion and smear the step profile of the interface over several cells. To avoid this smearing, special care needs to be taken to minimize the numerical diffusion. The following approaches are used to accomplish this: (1) free surface reconstruction using line techniques [7–11]; (2) high-resolution differencing schemes [12–17]. It becomes complicated to use a line reconstruction scheme on unstructured grids and more difficulty is encountered in extending it to three-dimensional calculations. The use of high-resolution schemes is promising, but existing schemes are still too diffusive for the capturing of sharp interfaces. Implementation in multiple dimensions is also restricted to the technique of operator splitting. A generalization for implementation on unstructured meshes is still required. Recently an unstructured-grid finite volume method based on the projection scheme [34] was developed for free surface flow simulation. In this method, a computational cell is considered a logical cube and cell vertices are allowed to coincide in physical space; thus unstructured grids of various types can be supported. The advantage of this method is that it can support a mixture of different cell types. But the disadvantage is that the meshes may not be generated easily and fully automatically, and if a fully unstructured grid, such as a triangular or tetrahedral mesh, is used, the data structure may not be efficiently utilized.

In this study, the VOF method is adopted and implemented into a finite volume (FV) procedure utilizing fully unstructured (triangular) meshes. A new unstructured FV incompressible Navier–Stokes (NS) solver based on a high-order characteristics-based scheme and a matrix-free implicit dual time-stepping scheme was developed for calculating free surface flows. The interface-capturing method using VOF and continuum surface factor (CSF) [2] was adopted to treat surface tension as a localized volume force, thus eliminating the need for detailed interface information. The VOF equation was also discretized by the same high-order characteristics-based FV scheme and integrated in time also by the implicit dual time-stepping method. As a result, the new solver has the following unique features compared with other finite volume schemes for computing free surface flows: (1) it can handle arbitrary boundary shapes easily, and the unstructutred meshes used can be generated fully automatically. And the method can be readily extended for three-dimensional computation. (2) It can tackle highly complex free surface flow problems accurately as a result of the use of the high-order characteristics-based finite volume method which is directly based on the conservation laws. (3) It is stable and efficient due to the use of the matrix-free implicit dual time-stepping scheme for all the equations because the time-step size is not limited by the stability condition and no matrix manipulation is required. To further improve the efficiency of the solver, an edge-based method is adopted for the calculation of convection fluxes while viscous fluxes are calculated using a cell-based method.

## 2. MATHEMATICAL MODELS

Here we consider incompressible flows with two different fluids. The density of one fluid is $\rho_l$ and the density of the second fluid is $\rho_g$. The nondimensional governing 2D equations, modified by the artificial compression method (ACM) [33], are given in vector form as

$$\mathbf{C}\frac{\partial \mathbf{W}}{\partial \tau} + \mathbf{K}\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}_c = \nabla \cdot \vec{\mathbf{F}}_v + \vec{S}, \tag{2.1}$$

where

$$\mathbf{W} = \begin{bmatrix} p \\ \vec{U} \\ \varphi \end{bmatrix}; \quad \vec{F}_{\mathrm{c}} = \begin{bmatrix} \vec{U} \\ \vec{U}\vec{U} + \dfrac{p}{\rho}\delta_{ij} \\ \varphi\vec{U} \end{bmatrix}; \quad \vec{\mathbf{F}}_{\mathrm{v}} = \begin{bmatrix} 0 \\ \frac{1}{\mathrm{Re}}\nabla\vec{U} \\ 0 \end{bmatrix};$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \mathbf{C} = \begin{bmatrix} \dfrac{1}{\beta} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \vec{\mathbf{S}} = \begin{bmatrix} 0 \\ \dfrac{1}{\rho}\vec{F}_{\mathrm{sv}} + \vec{F}_{\mathrm{g}} \\ 0 \end{bmatrix}.$$

In all the equations above, $\mathbf{W}$ is the vector of dependent variables. $\vec{U}$ is the velocity vector; u and v are velocity components in $x$ and $y$ coordinate directions; $p$ and $\rho$ are pressure and density, respectively. $\vec{F}_{\mathrm{c}}$ and $\vec{F}_{\mathrm{v}}$ are the convective flux and viscous flux vectors. $\vec{S}$ contains the surface tension and gravity terms. The first term on the left-hand side of Eq. (2.1) is a partial derivative with respect to pseudo-time $\tau$ (the artificial compression term), which is introduced to couple velocity and pressure fields for the calculation of pressure based on the divergence-free condition. $\mathbf{C}$ is a preconditioning matrix that arises with the implementation of the artificial compressibility method. $\mathbf{K}$ is the unit matrix with its first element being zero. On the other hand, $t$ is the physical time. Since the surface tension and gravity play significant roles during the development of a free surface flow process, the combined effects for surface tension and gravity are included. $\vec{F}_{\mathrm{g}}$ is the gravity force per unit mass and is given as

$$\vec{F}_{\mathrm{g}} = \frac{\vec{n}_{\mathrm{g}}}{Fr}, \tag{2.2}$$

where $Fr$ is the Froude number and $\vec{n}_{\mathrm{g}}$ is the unit vector along the prescribed direction of gravity. $\vec{F}_{\mathrm{sv}}$ is the surface tension force per unit volume, given by

$$\vec{F}_{\mathrm{sv}} = \frac{\kappa\nabla\varphi}{[\varphi]}\frac{1}{We}, \tag{2.3}$$

where $\kappa$ is the curvature of the interface and $We$ is the Weber number. $[\varphi]$ denotes the amount of jump in the fluid volume indicator function $\varphi$ across the interface. The curvature can be calculated as

$$\kappa = -\nabla\cdot\vec{n} = -\nabla\cdot\left(\frac{\nabla\cdot\varphi}{|\nabla\varphi|}\right). \tag{2.4}$$

$\varphi$ is a function of space and time, which is advected in the flow field according to the advection equation

$$\frac{d\varphi}{dt} = \frac{\partial\varphi}{\partial t} + (\vec{U}\cdot\nabla)\varphi = 0. \tag{2.5}$$

The fluid volume indicator function $\varphi$ is set to 1 in the liquid region and 0 in the gaseous region. For the interface between liquid and gas, $\varphi$ is between 0 and 1. Hence we have

$$\begin{aligned} \varphi(x,t)\begin{cases} =1 & \text{If } x \in \text{ the liquid (fluid 1)} \\ =0 & \text{If } x \in \text{ the gas (fluid 2)} \end{cases} \\ 0 < \varphi(x,t) < 1 \quad \text{If } x \in \text{ the interface.} \end{aligned} \tag{2.6}$$

The above equations are closed with the constitutive relations for the density and dynamic viscosity

$$\rho = \varphi\rho_l + (1-\varphi)\rho_{\mathrm{g}} \tag{2.7}$$

$$\mu = \varphi\mu_l + (1-\varphi)\mu_{\mathrm{g}}. \tag{2.8}$$

The total density conservation equation is replaced by the convection equation for the fluid volume indicator, and they are equivalent if $\rho_{\mathrm{g}}$ and $\rho_{\mathrm{l}}$ are constant.

The nondimensional variables used in above equations are defined as

$$x = \frac{x^*}{L^*};\ y = \frac{y^*}{L^*}; \qquad u = \frac{u^*}{u^*_\infty};\ v = \frac{v^*}{u^*_\infty}; \qquad \rho = \frac{\rho^*}{\rho^*_\infty};$$

$$p = \frac{p^*}{p^*_\infty};\ p^*_\infty = \rho^*_\infty (u^*_\infty)^2 \qquad \vec{U} = \frac{\vec{U}^*}{u^*_\infty}; \qquad t = \frac{t^*}{L^*/u^*_\infty};$$

$$Fr = \frac{u^*_\infty}{\sqrt{gL^*}}; \qquad \mathrm{Re} = \frac{\rho^* u^*_\infty L^*}{\mu^*}; \qquad We = \frac{\rho^*_\infty (u^*_\infty)^2 L^*}{\sigma^*_\infty},$$

where $L^*$ denotes the reference length and $u^*_\infty$ denotes the reference velocity. Terms with superscript $*$ indicate dimensional quantities, and the subscript $\infty$ indicates the other reference terms.

Equation (2.1) can be recast in integral form as follows:

$$\mathbf{C}\frac{\partial}{\partial\tau}\iint_S \mathbf{W}\,dS + \mathbf{K}\frac{\partial}{\partial t}\iint_S \mathbf{W}\,dS + \iint_S \nabla\cdot\vec{\mathbf{F}}_{\mathrm{c}}\,dS = \iint_S \nabla\cdot\vec{\mathbf{F}}_{\mathrm{v}}\,dS + \iint_S \mathbf{S}\,dS. \tag{2.9}$$

When convergence is reached in time-marching in pseudo time, the derivative terms associated with pseudo time $\tau$ disappear and the original unsteady incompressible Navier–Stokes equations are recovered as shown below:

$$\mathbf{K}\frac{\partial}{\partial t}\iint_S \mathbf{W}\,dS + \iint_S \nabla\cdot\vec{\mathbf{F}}_{\mathrm{c}}\,dS = \iint_S \nabla\cdot\vec{\mathbf{F}}_{\mathrm{v}}\,dS + \iint_S \mathbf{S}\,dS. \tag{2.10}$$

## 3. NUMERICAL METHODS

### 3.1. Unstructured Finite Volume Method

In this work, the finite volume (FV) method is used to discretize the governing equations. In an FV method, the computational domain is covered by a set of time-invariant, nonoverlapping control volumes which are the median dual of a triangular mesh (see Fig. 1d). On each of these control volumes, the integral form of the governing equations (Eq. (2.9)) is discretized.

Here a cell-vertex scheme is adopted, i.e., all computed variables in vector **W** are stored at vertices of the triangular cells. The vertices of the median dual mesh are located at the centroids of the triangles and their edges. Each dual edge is thus composed of two line segments as shown in Fig. 1d. The scheme is fully conservative at both the cell and global levels since the discretized equations are derived from the integral forms of conservation laws over the control volumes. This property is particularly important for the accurate calculation of fluid flows with discontinuities.

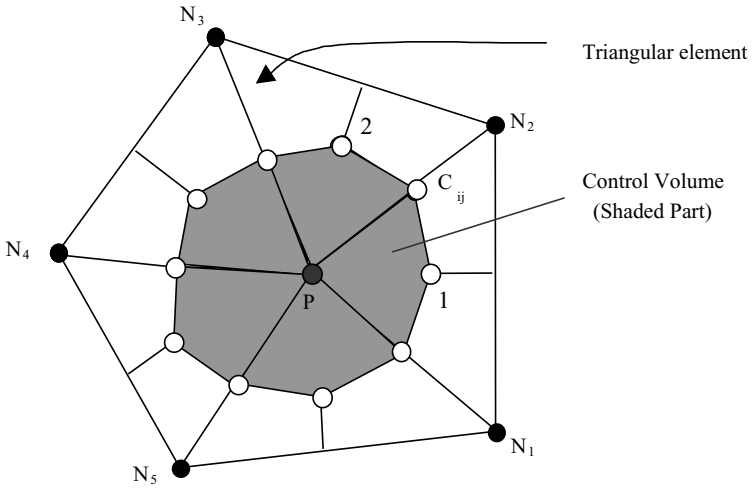



**FIG. 1d.** Construction of control volume of vertex *P*.

Spatial discretization is performed by using the integral form of the governing equations over the control volume surrounding the node or vertex, *P*:

$$\mathbf{C}\frac{\partial}{\partial \tau}\iint_{S_{\mathrm{cv}}} \mathbf{W}_P \, dS + \mathbf{K}\frac{\partial}{\partial t}\iint_{S_{\mathrm{cv}}} \mathbf{W}_{\mathrm{p}} \, dS + \iint_{S_{\mathrm{cv}}} \nabla \cdot \mathbf{F}_{\mathrm{c}} \, dS = \iint_{S_{\mathrm{cv}}} \nabla \cdot \mathbf{F}_{\mathrm{v}} \, dS + \iint_{S_{\mathrm{cv}}} \mathbf{S}_P \, dS. \tag{3.1}$$

With the introduction of the artificial compressibility method of Chorin [32], a pseudo time derivative of pressure is added to the continuity equation. Thus, the governing equations are transformed into a system of hyperbolic equations. The computation of convective terms in these equations can be performed using simple upwind, central difference, and characteristics-based schemes. In the simple upwind scheme, the wave propagation properties of the underlying hyperbolic system are taken into account. An advantage of this method is that they are naturally dissipative and no additional artificial viscosity term is required. The resulting flow variables at the boundary center, $C_{ij}$, are given as

$$\vec{U}_{ij} \begin{cases} = \vec{U}_P & \text{If } (\vec{F}_{\mathrm{c}})^k_{ij} \geq 0 \\ = \vec{U}_{Ni} & \text{If } (\vec{F}_{\mathrm{c}})^k_{ij} < 0, \end{cases} \tag{3.2}$$

where $(\vec{F}_{\mathrm{c}})^k_{ij}$ is the convection flux through part of the control volume boundary 1-$C_{ij}$-2 in Fig. 1d. It is known that simple upwind schemes introduce a large amount of numerical diffusion in the solution.

The central difference scheme does not attempt to account for the wave propagation properties present in the governing equations. In this method, a simple average of the two fluxes on both sides of the interface is used. However, artificial viscosity must be added to stabilize the central difference scheme. A standard scalar dissipation model by Jameson and Marriplis [33] was introduced. A first-order-accurate (second difference) artificial viscosity term is used to suppress oscillations in the vicinity of shock, and a third-order-accurate (fourth difference) viscosity term is used to suppress odd–even modes in the smooth regions of the flow. Although this approach is relatively simple, the coefficients in the artificial

viscosity terms must be adjusted to obtain stable results with minimum contamination of unphysical viscosity.

To introduce the upwind scheme using an edge-based procedure, the convection term is transformed as

$$\iint_{S_{\mathrm{cv}}} \nabla \cdot \vec{\mathbf{F}}_{\mathrm{c}}\, dS = \oint_{L_{\mathrm{cv}}} \vec{\mathbf{F}}_{\mathrm{c}} \cdot \vec{n}\, dl = \sum_{k=1}^{nedge} \left[ (\vec{\mathbf{F}}_{\mathrm{c}})_{ij}^{k} \cdot \vec{n} \Delta l_k \right], \tag{3.3}$$

where *nedge* is the number of edges associated with vertex $P$, $(\vec{F}_{\mathrm{c}})_{ij}^{k}$ is the convection flux through the part of the control volume boundary (similar to 1-$C_{ij}$-2 in Fig. 1d) whose length is $\Delta l_k$. Therefore, all the fluxes are calculated for the edges and then collected at the two ends of each edge for updating of flow variables in time-marching. This edge-based method leads to higher efficiency in computation and reduced data storage requirements compared with other methods, such as the cell-centerd method.

The viscous term is calculated by a cell-based method

$$\iint_{S_{\mathrm{cv}}} \nabla \cdot \vec{\mathbf{F}}_{\mathbf{v}}\, dS = \oint_{L_{\mathrm{cv}}} \vec{\mathbf{F}}_{\mathbf{v}} \cdot d\vec{l} = \sum_{i=1}^{ncell} (\vec{\mathbf{F}}_{\mathbf{v}} \cdot \Delta \vec{l}_{\mathbf{c}})_i = \frac{1}{2} \sum_{i=1}^{ncell} (\vec{\mathbf{F}}_{\mathbf{v}} \cdot \Delta \vec{l}_p)_i, \tag{3.4}$$

where $\Delta \vec{l}_{ci}$ is the part of the control volume boundary in cell $C_i$ (as shown in Fig. 1d), and $\Delta \vec{l}_{pi}$ is the edge vector of the cell edge apposite to vertex $P$ of the triangle under consideration. Here $(\vec{F}_{\mathrm{v}})_i$ is calculated at the centre of the triangle cell with a vertex $P$, which can be obtained by using Green's Theorem based on the variables at the three vertices of the triangle. The gradient of a flow variable $\Phi$ at the center of a cell is evaluated as

$$\begin{aligned} \nabla \Phi|_{\mathrm{cell}i} &= \frac{\iint_{Scell} \nabla \Phi dS}{\iint_{Scell} dS} \\ &= \frac{\int_{Lcell} \Phi \vec{n} \cdot dl}{\iint_{Scell} dS}, \end{aligned} \tag{3.5}$$

where *Scell* is the area of cell $i$ and *Lcell* is the boundary edges of cell $i$. For flow variables at a node, the calculation is based on the area (denoted by $S$) averaging of those at the centers of the surrounding cells:

$$\nabla \Phi|_{\mathrm{node}k} = \frac{\sum_{i=1}^{N} (\nabla \Phi|_{\mathrm{cell}\,i} \cdot S_{\mathrm{cell}\,i})}{\sum_{i=1}^{N} S_{\mathrm{cell}\,i}}. \tag{3.6}$$

$S_{celli}$ is the area of the triangular cell $i$ as shown in Fig. 2 and $N$ is the number of cells associated with node $k$.

When using the upwind scheme in the FV method, one needs to know the flow properties at the left and right sides of the control volume surface. Here the left and right state vectors $W_{\mathrm{L}}$ and $W_{\mathrm{R}}$ at a control volume surface are evaluated using an upwind-biased interpolation scheme, the MUSCL (monotonic upstream schemes for conservation laws) scheme [14],

$$W_{\mathrm{L}} = W_i + \frac{1}{4}[(1-\kappa)\,\Delta_i^{-} + (1+\kappa)\,\Delta_i^{+}] \tag{3.7}$$

$$W_{\mathrm{R}} = W_i + \frac{1}{4}[(1-\kappa)\,\Delta_i^{+} + (1+\kappa)\Delta_i^{-}], \tag{3.8}$$

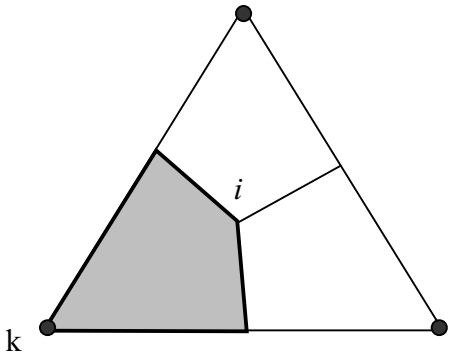



**FIG. 2.** Viscous flux calculation.

where

$$\begin{aligned}
\Delta_i^+ &= \Delta_j^- \\
&= W_j - W_i \\
\Delta_i^- &= W_i - W_{i-1} \\
&= 2\vec{ij} \cdot \nabla W_i - (W_j - W_i) \\
&= 2\vec{ij} \cdot \nabla W_i - \Delta_i^+ \\
\Delta_j^- &= W_{j+1} - W_j \\
&= 2\vec{ij} \cdot \nabla W_j - (W_j - W_i) \\
&= 2\vec{ij} \cdot \nabla W_j - \Delta_i^-.
\end{aligned}$$

Therefore,

$$W_{\mathrm{L}} = W_i + \frac{1}{2}[(1-\kappa)\vec{ij} \cdot \nabla W_i + \kappa \Delta_i^+] \tag{3.9}$$

$$W_{\mathrm{R}} = W_j + \frac{1}{2}[(1-\kappa)\vec{ij} \cdot \nabla W_j + \kappa \Delta_i^-], \tag{3.10}$$

where $\kappa$ is set to $1/3$ which corresponds to a nominally third-order accuracy scheme.

### 3.2. Characteristics-Based Scheme for Arbitrary Unstructured Grids

In this section, a characteristics-based scheme is derived using the modified Euler equations as a result of the introduction of the artificial compressibility method (ACM). This is possible because the modified equations become hyperbolic with the pseudo-time terms. The advantage of the scheme is that physics-based multidimensional upwinding can be introduced naturally and implicitly to ensure stability of numerical solutions without the need to adjust any coefficients.

The modified Euler equations are rewritten in partial differential form in a Cartesian coordinate system for the derivation of the method of characteristics,

$$\frac{\partial p}{\partial \tau} + \beta \frac{\partial u_i}{\partial x_i} = 0 \tag{3.11}$$

$$\frac{\partial u_i}{\partial \tau} + u_j \frac{\partial u_i}{\partial x_j} + u_i \frac{\partial u_j}{\partial u_j} + \frac{\partial p}{\partial x_i} = 0 \tag{3.12}$$

$$\frac{\partial \varphi}{\partial \tau} + u_j \frac{\partial \varphi}{\partial x_j} = 0, \tag{3.13}$$

where subscripts $i$ and $j$ equal 1 or 2, representing two spatial coordinates. Suppose that $\xi$ is a new coordinate outward normal to the boundary of a control volume that surrounds a particular vertex. In order to extend the method of characteristics to the unstructured grid solver, it is assumed that flow in the $\xi$ direction is approximately one-dimensional, and the above equations can then be transformed into

$$\frac{\partial p}{\partial \tau} + \beta \frac{\partial u_j}{\partial \xi} \xi_{x_j} = 0 \tag{3.14}$$

$$\frac{\partial u_i}{\partial \tau} + u_j \frac{\partial u_i}{\partial \xi} \xi_{x_j} + u_i \frac{\partial u_j}{\partial \xi} \xi_{x_j} + \frac{\partial p}{\partial \xi} \xi_{x_i} = 0 \tag{3.15}$$

$$\frac{\partial \varphi}{\partial \tau} + u_j \frac{\partial \varphi}{\partial \xi} \xi_{x_j} = 0, \tag{3.16}$$

where $\xi_{x_i} = \frac{\partial \xi}{\partial x_i}$ and $\xi_{x_j} = \frac{\partial \xi}{\partial x_j}$.

In the $\tau - \xi$ space as shown in Fig. 3, flow variable $W$ at pseudo-time level $n+1$ can be calculated along a characteristic $k$ using a Taylor-series expansion and the initial value at pseudo-time level $n$ ($W^k$)

$$W = W^k + W_\xi \xi_\tau \Delta\tau + W_\tau \Delta\tau$$

and

$$W_\tau = \frac{W - W^k}{\Delta\tau} - W_\xi \xi_\tau.$$

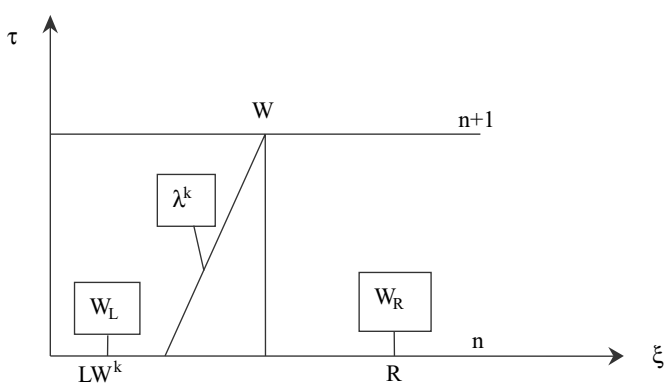



**FIG. 3.** $\tau - \xi$ coordinate.

A wave speed $\lambda^k$ is introduced,

$$\xi_\tau = \lambda^k \sqrt{\xi_{x_i}\xi_{x_i}},$$

and the normal vector components are

$$n_{x_j} = \frac{\xi_{x_j}}{\sqrt{\xi_{x_i}\xi_{x_i}}}.$$

Substituting components of $W_\tau$ into Eqs. (3.14), (3.15), and (3.16), we have

$$\frac{1}{\sqrt{\xi_{x_i}\xi_{x_i}}}\frac{(p-p^k)}{\Delta\tau} - p_\xi\lambda^k + \beta(u_\xi n_x + v_\xi n_y) = 0 \tag{3.17}$$

$$\frac{1}{\sqrt{\xi_{x_i}\xi_{x_i}}}\frac{(u-u^k)}{\Delta\tau} - u_\xi(\lambda^0 - \lambda^k) + u(u_\xi n_x + v_\xi n_y) + p_\xi n_x = 0 \tag{3.18}$$

$$\frac{1}{\sqrt{\xi_{x_i}\xi_{x_i}}}\frac{(v-v^k)}{\Delta\tau} - v_\xi(\lambda^0 - \lambda^k) + v(u_\xi n_x + v_\xi n_y) + p_\xi n_y = 0 \tag{3.19}$$

$$\frac{1}{\sqrt{\xi_{x_i}\xi_{x_i}}}\frac{(\varphi-\varphi^k)}{\Delta\tau} + \varphi_\xi(\lambda^0 - \lambda^k) = 0, \tag{3.20}$$

where $\lambda^0$ is the contra-variant velocity

$$\lambda^0 = un_x + vn_y.$$

In order to derive the compatibility equations, the spatial derivatives, such as $u_\xi$, $v_\xi$, and $p_\xi$ have to be eliminated from the above equations. Following the approaches of Eberle [19] for compressible flow equations and Drikakis *et al.* [18] for incompressible flow equations on structured grids, each of the above four equations is multiplied by an arbitrary variable and all the resulting equations are summed to form a new equation,

$$\frac{1}{\Delta\tau\sqrt{\xi_{x_i}\xi_{x_i}}}A - p_\xi B + u_\xi C + v_\xi D + \varphi_\xi E = 0, \tag{3.21}$$

where

$$\begin{aligned} A &= a(p-p^k) + b(u-u^k) + c(v-v^k) \\ B &= -a\lambda^k + bn_x + cn_y \\ C &= an_x\beta + b(\lambda^0 - \lambda^k + un_x) + cvn_x \\ D &= an_y\beta + bun_y + c(\lambda^0 - \lambda^k + vn_y) \\ E &= d(\lambda^0 - \lambda^k), \end{aligned} \tag{3.22}$$

and a, b, c, and d are the arbitrary variables used to multiply the equations. We define the

coefficients of the partial space derivatives to be zero, i.e., A, B, C, D, and E are zero:

$$A = 0 \tag{3.23}$$

$$B = 0 \tag{3.24}$$

$$C = 0 \tag{3.25}$$

$$D = 0 \tag{3.26}$$

$$E = 0. \tag{3.27}$$

Equations (3.23) to (3.27) constitute a linear system $\Phi X = 0$ with $X = \{a, b, c, d\}$. Variables a, b, c, and d are generally nonzero; thus the system of equations has a nontrivial solution. This means that $\det(\Phi) = 0$, and the following eigenvalues can be derived:

$$\lambda^1 = \lambda^2 = \lambda^0$$
$$\lambda^1 = \lambda^0 + \sqrt{(\lambda^0)^2 + \beta}$$
$$\lambda^2 = \lambda^0 - \sqrt{(\lambda^0)^2 + \beta}.$$

For each eigenvalue or characteristic speed, characteristic equations can be derived from Eqs. (3.23) to (3.27). For example, for $\lambda^k = \lambda^0$, we have

$$a = \frac{bn_x + cn_y}{\lambda^0}.$$

Substituting this into Eq. (3.22), we obtain

$$\frac{bn_x + cn_y}{\lambda^0}(p - p^0) + b(u - u^0) + c(v - v^0) + d(\varphi - \varphi^0) = 0,$$

i.e.,

$$b[n_x(p - p^0) + \lambda^0(u - u^0)] + c[n_y(p - p^0) + \lambda^0(v - v^0)] + d\lambda^0(\varphi - \varphi^0) = 0.$$

For any b, c, and d, the above equation is always satisfied. Therefore all the terms in square brackets are zero.

As a result, we have

$$(u - u^0)n_y - (v - v^0)n_x = 0 \tag{3.28}$$

$$\varphi = \varphi^0. \tag{3.29}$$

For $\lambda = \lambda^1$,

$$p - p^1 = -\lambda^1[(u - u^1)n_x + (v - v^1)n_y]. \tag{3.30}$$

For $\lambda = \lambda^2$,

$$p - p^2 = -\lambda^2[(u - u^2)n_x + (v - v^2)n_y]. \tag{3.31}$$

Finally, all the variables are determined using the above characteristic Eqs. (3.28) to (3.31);

$$\begin{aligned}\varphi &= \varphi^0\\ u &= f n_x + u^0 n_y^2 - v^0 n_x n_y\\ v &= f n_y + v^0 n_x^2 - u^0 n_y n_x\\ p &= p^1 - \lambda^1[(u-u^1)n_x + (v-v^1)n_y]\end{aligned}$$

or

$$p = p^2 - \lambda^2[(u-u^2)n_x + (v-v^2)n_y],$$

where

$$\begin{aligned}f &= \frac{1}{2C}[p^1 - p^2 + n_x(\lambda^1 u^1 - \lambda^2 u^2) + n_y(\lambda^1 v^1 - \lambda^2 v^2)]\\ C &= \sqrt{(\lambda^0)^2 + \beta}.\end{aligned}$$

Flow quantities at $n+1$ psuedo-time level obtained from the above equations on the characteristics are then used to calculate convection fluxes at the control volume interface. Those on different characteristics at $n$ time level are approximately evaluated by an upwind scheme using the signs of the characteristics as suggested in [18].

$$W^j = \frac{1}{2}[(1 + \mathrm{sign}(\lambda^j))W_{\mathrm{L}} + (1 - \mathrm{sign}(\lambda^j))W_{\mathrm{R}}],$$

where $W_{\mathrm{L}}$ and $W_{\mathrm{R}}$ are obtained by the upwind-biased MUSCL interpolation.

### 3.3. Matrix-Free Implicit Dual Time-Stepping Scheme

To improve computational efficiency, implicit schemes can be used in order to use the largest possible time-step sizes permitted by accuracy considerations. However, most implicit schemes require a large amount of computing effort per time step compared with explicit schemes, making them less attractive for many unsteady flow simulations. Here a matrix-free implicit scheme is derived, which is found to be efficient in terms of memory required and computing effort per time step. This is due to the fact that no matrix manipulation is required by the scheme.

For a certain vertex $P$, the spatially discretized equations form a system of coupled ordinary differential equations, which may be written as

$$\begin{aligned}\frac{\partial}{\partial t}(W_p) &= -\frac{1}{\Delta S_{\mathrm{cv}}}\left\{\sum_{k=1}^{nedge}\left[(\vec{F}_c)_{ij}^k \cdot (\vec{n}\Delta l_k)\right] - \frac{1}{2}\sum_{i=1}^{ncell}(\vec{F}_v \cdot \vec{n}\Delta l_P)_i + \mathbf{S}_P \Delta S_{\mathrm{cv}}\right\}\\ &= -\frac{1}{\Delta S_{\mathrm{cv}}} R(W_P),\end{aligned} \tag{3.32}$$

where the right-hand side represents the residual. An implicit scheme is adopted to approximate Eq. (3.32), and the semidiscrete form is given as

$$\frac{\partial}{\partial t}\left(\Delta S_{\mathrm{cv}} W_p^{n+1}\right) = -R\left(W_P^{n+1}\right). \tag{3.33}$$

The superscript $(n+1)$ denotes the time level at $(n+1)\Delta t$, and all the variables are evaluated at this time level. In this work, $\frac{\partial}{\partial t}$ is discretized using a second-order-accurate backward differencing formula, and Eq. (3.33) can be rewritten as follows

$$\frac{1.5\,\Delta S_{\mathrm{cv}} W_p^{n+1} - 2.0\,\Delta S_{\mathrm{cv}} W_p^{n} + 0.5\,\Delta S_{\mathrm{cv}} W_p^{n-1}}{\Delta t} + R\left(W_P^{n+1}\right) = \tilde{R}\left(W_p^{n+1}\right) = 0, \tag{3.34}$$

where $\tilde{R}(W_p^{n+1})$ is the new modified residual which contains both the time derivative and flux vectors.

The advantage of the above implicit scheme is that the physical time-step size is restricted only by numerical accuracy, not by numerical stability. This is especially useful in unsteady flow simulation where the maximum time-step size is much smaller than the size permitted by accuracy considerations. The derivative with respect to a fictitious pseudo-time $\tau$ is added to the above equation to reformulate the unsteady Navier–Stokes equation as

$$\Delta S_{\mathrm{cv}} \frac{W_p^{n+1,\,m+1} - W_p^{n+1,\,m}}{\Delta \tau} = \tilde{R}\left(W_p^{n+1,\,m}\right), \tag{3.35}$$

whose solution is sought by marching to a pseudo steady state in $\tau$. The superscripts $(m+1)$ and $(m)$ denote time levels in pseudo time. Once the artificial steady state is reached, the derivative of $W_p$ with respect to $\tau$ becomes zero, and the solution will satisfy $\tilde{R}(W_p^{n+1}) = 0$. This is actually the solution of Eq. (3.34). Hence the original unsteady Navier–Stokes equations are fully recovered. Therefore instead of solving each time step in the physical time domain $(t)$, the problem is transferred into a sequence of steady-state computations in the artificial time domain $(\tau)$.

An approximate flux function is introduced here to simplify the implicit time-stepping calculation. Following [20], the total flux (including inviscid and viscous fluxes) across a control-volume surface associated with a certain edge $ij$ can be approximated as

$$F_{ij} \approx \frac{1}{2}\left[\vec{F}_i^{\,\mathrm{c}} \bullet \vec{\mathrm{n}} + \vec{F}_j^{\,\mathrm{c}} \bullet \vec{\mathrm{n}} - |\lambda_{ij}|(W_i - W_j)\right],$$

where $\lambda_{ij}$ is the spectral radius associated with edge $ij$.

$$\lambda_{ij} = \vec{\mathrm{U}} \cdot \vec{\mathrm{n}}_{ij} + \sqrt{(\vec{\mathrm{U}} \cdot \vec{\mathrm{n}}_{ij})^2 + \beta^2}.$$

A Taylor series expansion is performed for the residual in Eq. (3.35) with respect to the pseudo time for node $i$:

$$\tilde{R}\left(W_i^{m+1}\right) = \tilde{R}\left(W_i^{m}\right) + \frac{\partial \tilde{R}_i}{\partial W_i} + \sum_{j=j1}^{jm} \frac{\partial \tilde{R}_i}{\partial W_j} \Delta W_j.$$

And in all the $R$ terms, the Taylor series expansions of the fluxes are

$$\frac{\partial F_{ij}}{\partial W_i} = \frac{1}{2}\left[\frac{\partial F_{ij}^{\mathrm{c}}}{\partial W_i} - |\lambda_{ij}|\right];$$
$$\frac{\partial F_{ij}}{\partial W_j} = \frac{1}{2}\left[\frac{\partial F_{ij}^{\mathrm{c}}}{\partial W_j} + |\lambda_{ij}|\right].$$

And for the physical time-dependent terms, we have

$$W_i^{m+1} = W_i^m + \Delta W_i.$$

After combining all the residuals at every node in the flow field into a vector, we have

$$R(W^{m+1,n+1}) = R(W^{m,n+1}) + A\Delta W,$$

where

$$A = \left\{\frac{\partial R_i}{\partial W_j}\right\}.$$

And the whole-field equivalent of Eq. (3.35) can then be rewritten as

$$\Delta S_{\mathrm{cv}}^{n+1}\left(\frac{\Delta t + 1.5\Delta\tau}{\Delta t} - \frac{A\Delta\tau}{\Delta S_{\mathrm{cv}}^{n+1}}\right)\frac{\Delta W}{\Delta\tau} = R^{n+1,m} - \frac{1.5W^{n+1,m}\Delta S_{\mathrm{cv}}^{n+1} - 2.0W^n\Delta S_{\mathrm{cv}}^n + W^{n-1}\Delta S_{\mathrm{cv}}^{n-1}}{\Delta t};$$

i.e.,

$$\Delta S_{\mathrm{cv}}^{n+1}\tilde{A}\frac{\Delta W}{\Delta\tau} = \tilde{R}^{n+1,m}. \tag{3.36}$$

Thus,

$$\Delta S_{\mathrm{cv}}^{n+1}\frac{\Delta W}{\Delta\tau} = \tilde{\tilde{R}}^{m+1,n}; \tag{3.37}$$

here

$$\tilde{\tilde{R}}^{m+1,n} = \tilde{A}^{-1}\tilde{R}^{n+1,m};$$
$$\tilde{A} = \frac{\Delta t + 1.5\Delta\tau}{\Delta t} - \frac{A\Delta\tau}{\Delta S_{\mathrm{cv}}};$$
$$\tilde{R}^{n+1,m} = \tilde{R}^{n+1,m} - \frac{1.5W^{n+1,m}\Delta S_{\mathrm{cv}}^{n+1} - 2.0W^n\Delta S_{\mathrm{cv}}^n + W^{n-1}\Delta S_{\mathrm{cv}}^{n-1}}{\Delta t}.$$

Further approximation can be introduced in order to achieve maxtrix-free computation. If we employ point implicit treatment to the above equations, then only the diagonal terms in $\tilde{A}$ are used in the pseudo time-stepping. As a result, the equation for every node can now be written as

$$\Delta S_{\mathrm{cv}i}^{n+1}\frac{\Delta W_i}{\Delta\tau} = \tilde{\tilde{R}}_i^{m+1,n}; \tag{3.38}$$

here

$$\tilde{\tilde{R}}_i^{m+1,n} = \tilde{A}_{ii}^{-1} \tilde{R}_i^{m+1,n};$$

and

$$\tilde{A}_{ii}^{-1} = \left( \frac{\Delta t + 1.5\Delta\tau}{\Delta t} - \frac{A_{ii}\Delta\tau}{\Delta S_{\text{cv}i}} \right)^{-1}.$$

Pseudo time-stepping is then performed on Eq. (3.38). For a five-stage scheme, the stage coefficients are

$$\alpha_1 = 1/4, \quad \alpha_2 = 1/6, \quad \alpha_3 = 3/8, \quad \alpha_4 = 1/2, \quad \alpha_5 = 1.$$

To speed up the convergence rate, an implicit residual-smoothing scheme developed for unstructured grids was employed. The idea behind this scheme is to replace the residual at one point of the flow field with a smoothed or weighted average of the residuals at the neighboring points. The averaged residuals are calculated implicitly in order to increase the maximum CFL number, thus increasing the convergence rate. Normally this procedure allows the CFL number to be increased by a factor of 2 or 3. The smoothing equation for a vertex $k$ can be expressed as

$$\bar{R}_k = R_k + \varepsilon \delta^2 \bar{R}_k, \tag{3.39}$$

where $R$ is the original residual, $\bar{R}$ is the smoothed residual, and $\varepsilon$ is the smoothing coefficient.

$$\varepsilon = \max \left\{ \frac{1}{4} \left[ \left( \frac{CFL}{CFL^*} \right)^2 - 1 \right], \ 0 \right\}, \tag{3.40}$$

where $CFL^*$ is the maximum CFL number of the basic scheme.

The solution to the above equations can be obtained on an unstructured grid by a Jacobi iterative method,

$$\bar{R}_k^{(m)} = R_k^{(0)} + \varepsilon \cdot \sum_{i=1}^{numnod(k)} \left[ \bar{R}_i^{(m)} - \bar{R}_k^{(m)} \right]; \tag{3.41}$$

$$\bar{R}_k^{(m)} = \frac{R_k^{(0)} + \varepsilon \cdot \sum_{i=1}^{numnod(k)} \bar{R}_i^{(m-1,m)}}{1 + \varepsilon \cdot numnod(k)}, \tag{3.42}$$

i.e., where $numnod(k)$ is the number of neighboring nodes of vertex $k$.

### 3.4. Initial and Boundary Conditions

For the nodes along a solid wall, solid wall boundary conditions apply. A nonslip boundary condition is imposed for all nodes on the wall. The values of the other flow properties, such as the indicator function and the pressure, are directly calculated at the wall nodes. An inlet boundary is a boundary where the fluid has a specified velocity distribution. The pressure

at the inlet is unknown and can be calculated directly. The position of the interface at the start of the simulation needs to be defined. For nodes that lie along the outlet boundary, the pressure is fixed at a constant value. Velocity is calculated along the outlet. For transient calculations, all the initial variables need to be consistent with each other. This can be achieved by specifying the velocity field as well as the indicator function directly. It is also necessary to initialize the pressure field to a distribution consistent with the velocity and density fields in order to reduce computational effort.

## 4. RESULTS AND DISCUSSION

### 4.1. Rayleigh–Taylor Instability

The test case considered is similar to that in [22], in which the numerical method is also based on the ACM and the dual time-stepping scheme, but with structured grids. The Rayleigh–Taylor instability is considered for two viscous, incompressible fluid layers with a prescribed density ratio ($\rho_1/\rho_2 = 998/499 = 2$) and a uniform kinematic viscosity ($\nu 1/\nu 2 = 1$). The fluids are confined within a periodic domain of width $L$ and height $H(=4L)$ which is bounded by impermeable walls at the top and bottom of the flow domain. The flow field is assumed to be symmetric about both its left and right boundaries.

A single wavelength perturbation is introduced at the fluid interface using the following nondimensional initial velocity field (adapted from the work of Daly [23]):

$$\frac{u}{U_{\text{ref}}} = \begin{cases} \alpha \sin\left(\frac{\pi x}{L}\right) \exp\left(-\frac{\pi |y|}{L}\right), & \frac{y}{L} > 0, \\ -\alpha \sin\left(\frac{\pi x}{L}\right) \exp\left(-\frac{\pi |y|}{L}\right), & \frac{y}{L} < 0, \end{cases}$$

$$\frac{v}{U_{\text{ref}}} = \alpha \cos\left(\frac{\pi x}{L}\right) \exp\left(-\frac{\pi |y|}{L}\right).$$

In [22], $\alpha = \frac{\pi A \Delta y}{2 U_{\text{ref}} L}$, where $A$ is a perturbation amplitude and $\Delta y$ is a representative mesh increment in the vertical direction. However, in this study, we simply assume that $\alpha = 0.25$. The velocity field corresponds to a sinusoidal perturbation of wavelength $2L$ and the initial pressure was set to a hydrostatic distribution. Figure 4 illustrates the important conditions of the problem.

Nondimensional initial pressure is calculated as

$$P^* = \frac{P(y) - P_{\text{ref}}}{\rho U_{\text{ref}}^2} = \frac{\rho g (H - y)}{\rho U_{\text{ref}}^2},$$

where $U_{\text{ref}} = \sqrt{gL}$, $\frac{H}{L} = 2$.
Therefore,

$$P^*_{\text{lower}} = 2 - \frac{\rho_2}{\rho_1} y^*$$

$$P^*_{\text{upper}} = 2 - y^*.$$

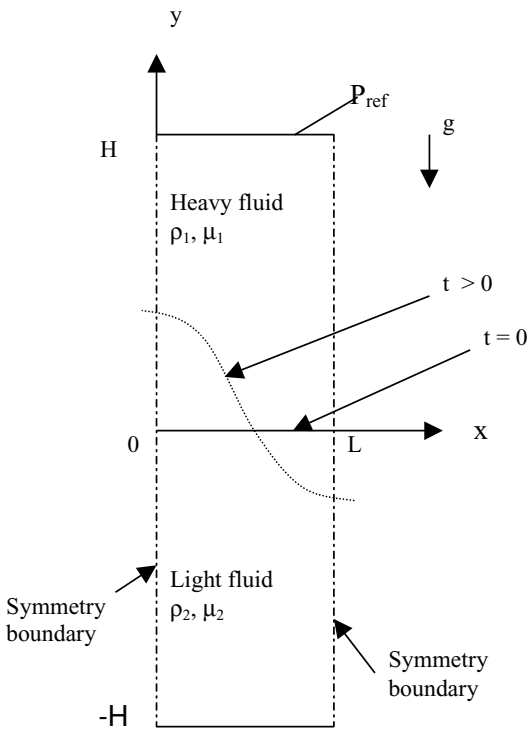



**FIG. 4.** Illustration of the Rayleigh–Taylor instability problem.

The three governing dimensionless parameters for the flow are the Reynolds number, the Weber number, and the Froude number, which are:

$$Re = \frac{U_{\text{ref}} L^*}{\upsilon}; \quad We = \frac{\rho U_{\text{ref}}^2 L^*}{\sigma}; \quad Fr = \frac{U_{\text{ref}}^2}{g L^*}.$$

In this case, the Froude number $Fr = 1$ due to the definition of the reference velocity, and the Weber number is 5000.

An initial perturbation causes the light fluid to rise along the left boundary and the heavy fluid to sink along the right boundary. The displacement of the interface is seen to be nearly symmetric during the early growth phase of the instability. As the amplitude of the instability increases, the characteristic mushroom shape emerges in the vicinity of the central vortex. Eventually, the walls begin to influence the solution during the latter stages of the transient, especially in the high Reynolds number case. To permit the interface to develop unimpeded for longer times, a larger computational domain would be required.

The characteristics-based scheme with third-order interpolation is used for this case. Various meshes of different densities are used for the grid convergence test and are listed in Table I. Figures 5 and 6 show the predicted interfaces for $Re = 100$ and 283 with the meshes, at nondimensional times $t = 3.2$ and 4. Very small discrepancies are observed between the medium-grid and fine-grid results, and the fine grid is adopted in all other simulations. Figure 7 shows the effect of different time-step sizes (0.005 and 0.01) on the predicted interfaces at $Re = 100$ and $t = 3.2$. It is also found that the two time-step sizes produce

**TABLE I**
**Details of Grids Used**

| Unstructured grids | Number of nodes | Number of elements |
|---|---|---|
| 1 (coarse) | 5,151 | 10,000 |
| 2 (medium) | 10,251 | 20,000 |
| 3 (fine) | 17,226 | 33,800 |

exactly the same results. Therefore the larger time-step size is adopted. Presented in Fig. 8 is the predicted motion of the free surface at $Re = 283$ using the fine grid. The characteristic mushroom shape of the interface is pronounced at this Reynolds number after time $t = 3$. The results obtained are comparable to those presented in [22], although the $\alpha$ value is not exactly the same for the two tests. In this test, different numbers of subiterations were used and it was found that 150 subiterations per physical time step can ensure convergent results, which means that a further increase in the number of subiterations would not change the results.

### 4.2. The Broken Dam Problem

In order to assess the accuracy and robustness of the solver in calculating free surface flow problems, the broken dam problem has also been selected as a benchmark test case. The broken dam problem is a classic problem in free surface hydrodynamics, and as its name implies, it is basically the sudden collapse of a rectangular column of fluid onto a horizontal surface. The broken dam problem gets its name due to its use in modelling the sudden failure of a dam. The relatively simple geometry and initial conditions associated with the problem have made it a popular validation case for various surface tracking and capturing schemes. The experimental results of the broken dam problem by Martin and Moyce [24] have been used in the validation of various numerical schemes

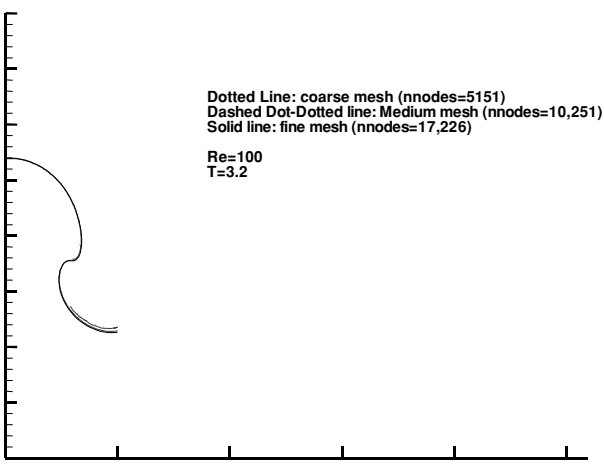



**FIG. 5.** Grid convergence study for $Re = 100$.

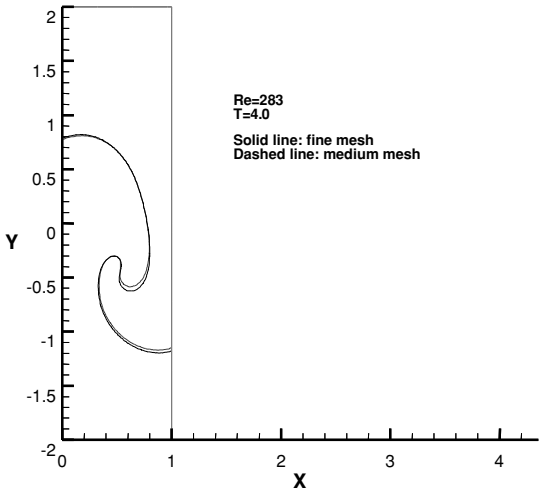



**FIG. 6.** Grid convergence study for $Re = 283$.

by Davis [12], Ramaswamy and Kawahara [26], Hirt and Nichols [1], and Koshizuka *et al*. [27].

The specific geometry employed in the present work is illustrated in Fig. 9. A rectangular water column 0.5 m wide and 0.9 m high is enclosed within an air-filled container. It can be imagined that the water column is initially kept in place by a thin partition or dam on the right-hand side of the column. At time $t = 0+$, the partition is removed instantaneously, thereby allowing the water column to collapse under the influence of gravitational force.

For comparison purposes, our broken dam problem was simulated using two different schemes, namely (1) the central difference scheme with artificial viscosity and (2) the characteristics-based upwind scheme. Both schemes were employed using various orders of accuracy in interpolation. These combinations of schemes and orders of accuracy are summarized in Table II. The limiter is the Minmod flux limiter.

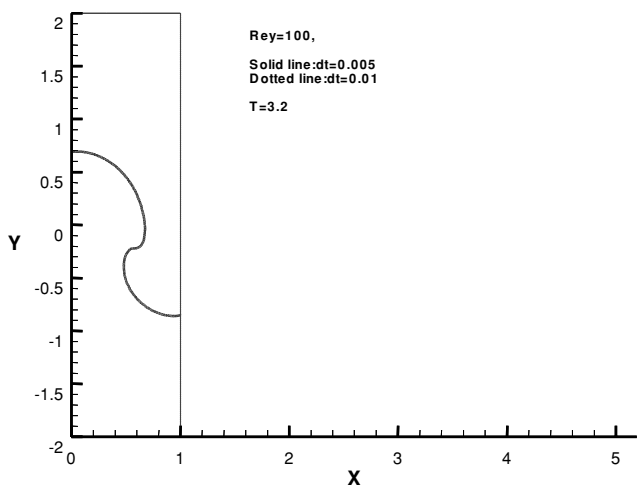



**FIG. 7.** Effect of time-step sizes.

**TABLE II**
**Schemes Tested**

| Schemes | Orders of Accuracy |
|---|---|
| (1) Central difference scheme (Central) | (i) Second order<br>(ii) Third order<br>(iii) Third order with limiter |
| (2) Characteristic upwind (Cha) | (i) Second order<br>(ii) Third order<br>(iii) Third order with limiter |

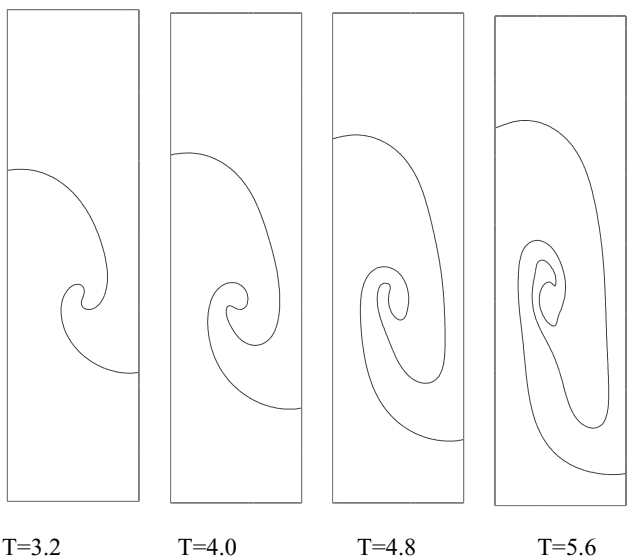



**FIG. 8.** Time evolution of the Rayleigh–Taylor instability at $Re = 283$ (fine grid).

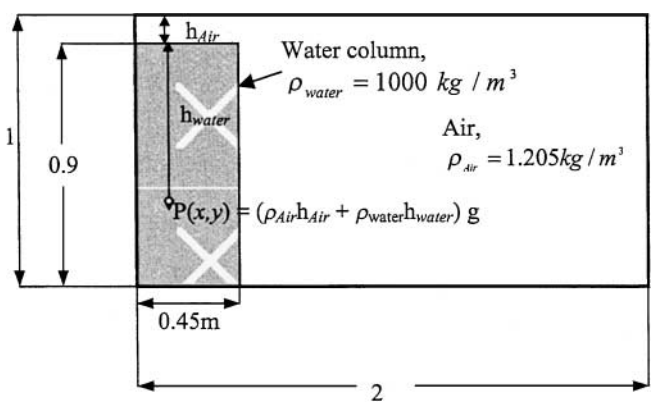



**FIG. 9.** Illustration of the 2D broken dam problem at $t = 0+$.

The initial conditions for the simulation of the broken dam problem were prescribed as follows. The density field was initialized with values appropriate for each fluid as shown in Fig. 9. The water column was assigned the density value of 1000 kg/m$^3$ and the remainder of the computational domain was given the density of air at a value of 1.205 kg/m$^3$. The initial velocity components were set to zero everywhere and the pressure distribution was defined to be hydrostatic pressure relative to the top surface of the water column. The pressure, $P(x, y)$, is initialized as shown in Fig. 9.
For any node within the water column ($x \leq 0.45$ m and $y \leq 0.9$ m),

$$P(x, y) = (\rho_{\text{air}} h_{\text{air}} + \rho_{\text{water}} h_{\text{water}}) g.$$

For any node outside the water column, pressure is given as

$$P(x, y) = \rho_{\text{air}} h_{\text{air}} g.$$

No-slip boundary conditions were applied to the four walls of the container. The viscous coefficient of water is $1.01 \times 10^{-3}$ Pa.s and the viscous coefficient of the surrounding air is $1.81 \times 10^{-5}$ Pa.s. Three sets of meshes are used which have different mesh densities: (a) mesh A = 1404 nodes; (b) mesh $B$ = 5151 nodes; (c) mesh C = 9660 nodes. In numerical experiments, results with mesh B and mesh C are basically independent of their respective mesh sizes; thus mesh B is used for all the calculations.

The profiles of the free surface of the water column at regular time interval ($t > 0+$) for simulations by different schemes were then visualized by choosing $\varphi = 0.5$. Finally, the length ($L$) and height ($H$) of the water column were compared with experimental results obtained by Martin and Moyce [24]. See Fig. 10 for an illustration of an the motion of the water column at $t > 0$.

Figure 11 presents a comparison of the calculated surge front and column height as functions of time using the third-order characteristics-based scheme, which shows an excellent agreement between the two sets of data. With these encouraging validation results, the other

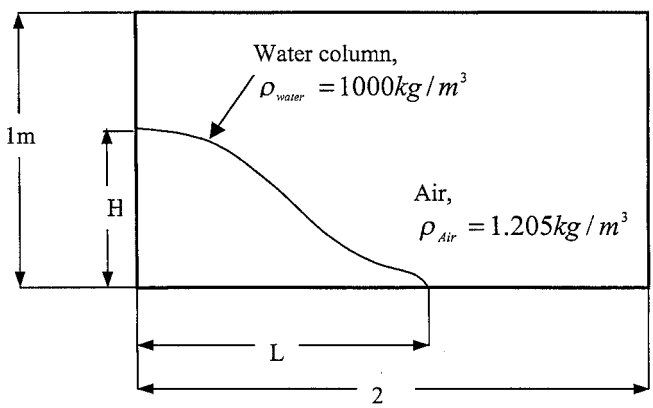



**FIG. 10.** Illustration of the 2D broken dam problem at $t > 0+$.

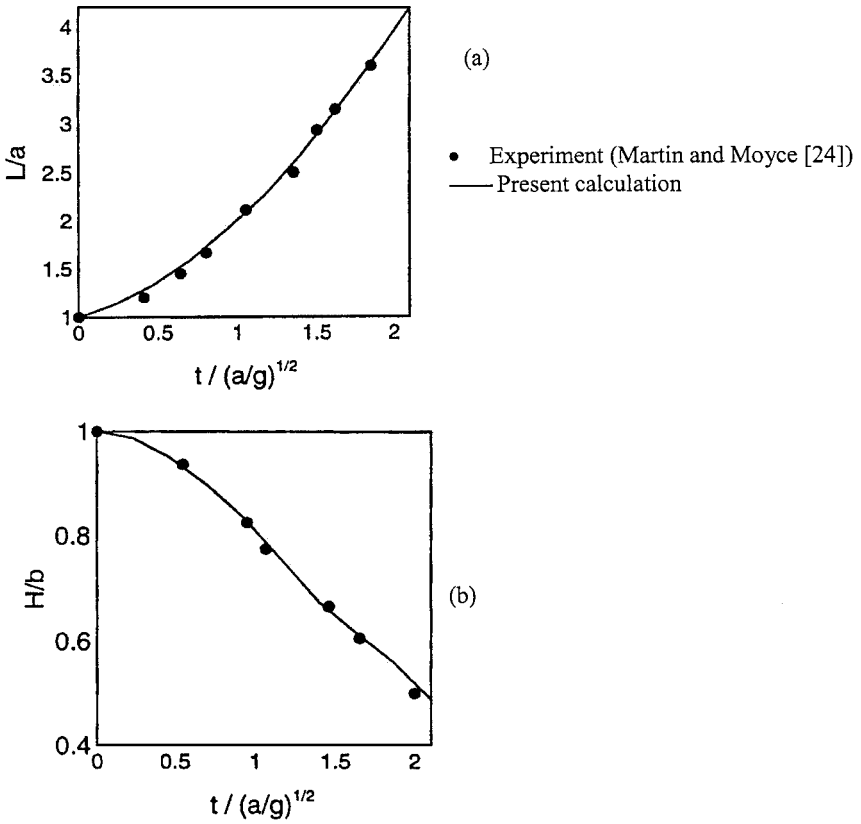



**FIG. 11.** Comparisons of calculated and experimental results: (a) nondimensionalized length of surge front against time; (b) nondimensionalized column height against time.

schemes with different orders of accuracy listed in Table II are then tested and their results are compared in Figs. 12 to 15. By observation, the profiles obtained by schemes with a flux limiter exhibit slower surge front movements as compared to schemes without a limiter. This is because the addition of a flux limiter tends to smooth out the VOF value near the interface and makes the schemes diffusive and less accurate. It is noted that even without the use of a limiter all the schemes with various orders of discretization accuracy remain stable, thus the limiter is considered redundant here. It is found that there is no distinct difference in the free surface profiles before $t = 0.24$. The explanation is that the length of time into the process is far too short for the different schemes to show any clear-cut visual discrepancies. However, the differences in results begin to show in the plots after $t = 0.24$, as illustrated in Fig. 12. The free surface profiles of the schemes having only second-order accuracy begin to slow and lag behind. When the all the simulations end at $t = 0.6$, it is obvious from Fig. 13 that the free surface profiles generated using the second-order schemes suffer from a time lag compared to the profiles generated using third-order schemes. This difference can also be seen in Figs. 14 and 15 in which the length ($L$) of the surge front and the column height ($H$), respectively, are plotted against time. From the plots, it could be seen that the profiles generated using third-order-accurate schemes do not show much difference.

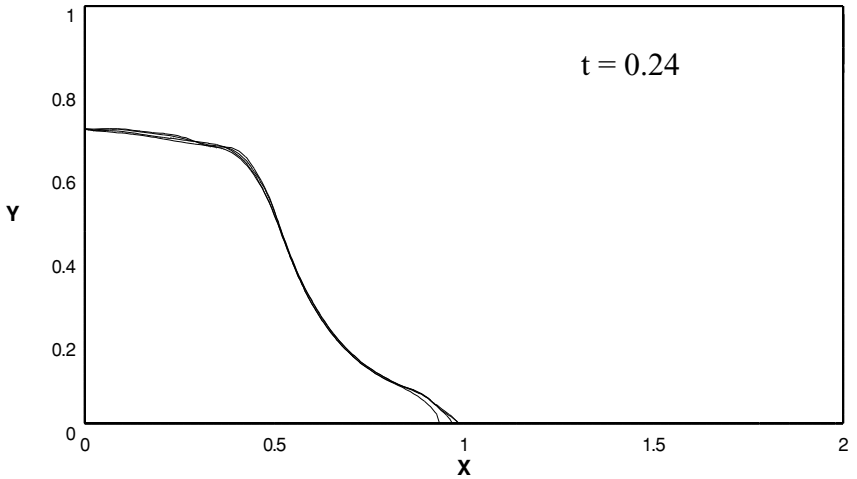



**FIG. 12.** Free surface profile at $t = 0.24$: results from different schemes begin to differ.

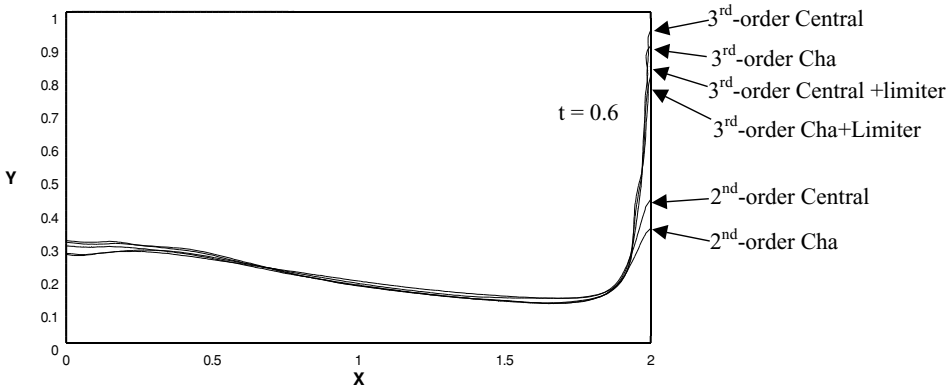



**FIG. 13.** Free surface profile at $t = 0.6$: results from different schemes.

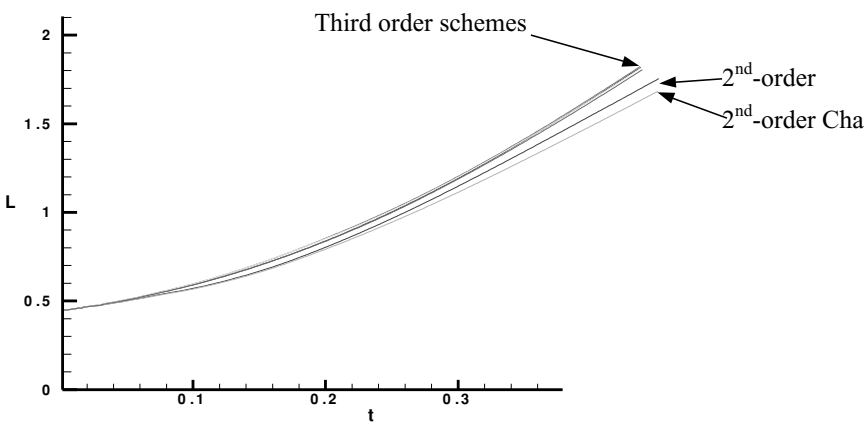



**FIG. 14.** Graph of length of surge front ($L$) against time ($t$).

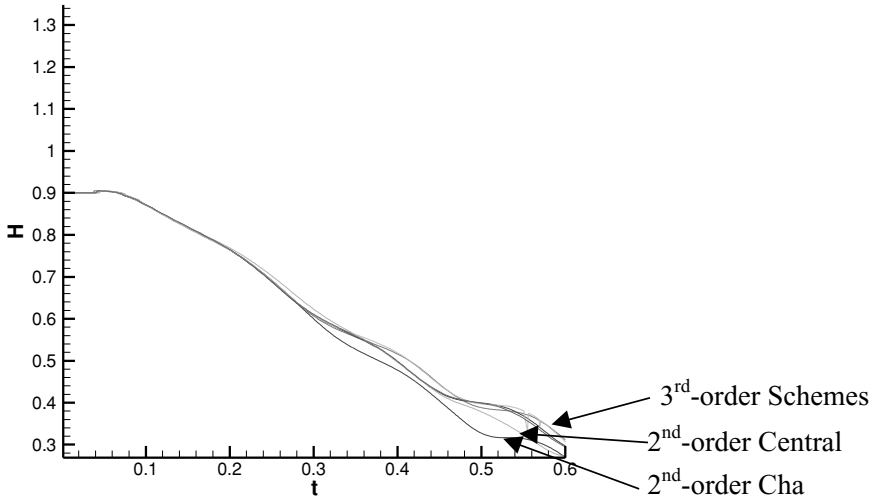



**FIG. 15.** Graph of column height ($H$) against time ($t$).

### 4.3. Bubble Rising in a Partially Filled Container

In this example, a two-dimensional bubble rising in a container partially filled with water is studied. This problem is modelled with a two-fluid system with a density ratio of 1.0 : 2.0. The heavier fluid is water while the lighter fluid is oil, and the two fluids have densities of 1000 kg/m$^3$ and 500 kg/m$^3$, respectively. Two computer simulations are performed here. The first computation is performed without the inclusion of surface tension while the second computation demonstrates the effect of surface tension. The computational domain for both cases is a 6 $D$-by-7 $D$ rectangle with $D$ being the initial diameter of the bubble. The computational mesh adopted after mesh-independent testing consists of 33,600 triangular mesh elements and 17,061 nodes.

A similar test case has been performed by Pan and Chang [25] also using the ACM and the dual time-stepping scheme on structured grids. The main difference between their work and ours is that their problem is modeled by a three-fluid system. Also, surface tension has not been included in their method. However, their results can still serve as a useful reference.

Initially, the circular oil bubble is immersed in the water that occupies the lower half of the closed container. A layer of the oil floats above the water. The initial depth of the water in the container is 5 $D$ with a 1 $D$ thickness of oil floating on it. The center of the oil bubble is located at 0.5 $D$ below the water surface. The initial velocity field of the entire computation domain is zero. The Froude number is also set to 1.0. The ratios of both density and dynamic viscosity of water and oil are 2 : 1. Uniform pressure is assigned to the bubble that is based on the hydrodynamic pressure at the center of the bubble. The Reynolds and Weber numbers are defined as

$$Re = \frac{\rho_{\text{water}} \sqrt{g} (2R_0)^{\frac{3}{2}}}{\mu_{\text{water}}}$$

$$We = \frac{\rho_{\text{water}} g (2R_0)^2}{\sigma},$$

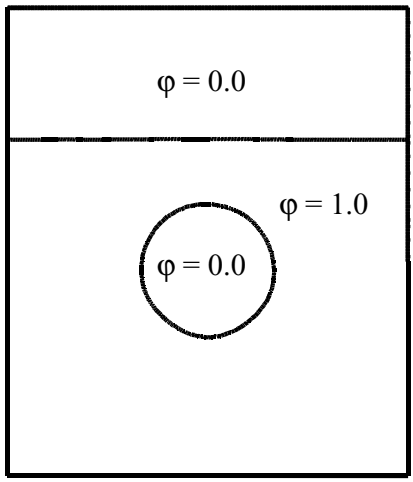


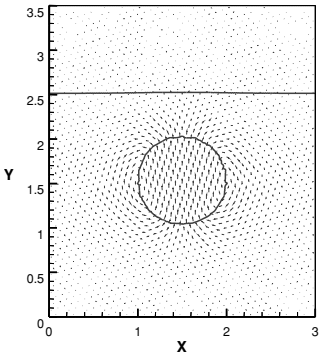

**FIG. 16.** Contour plot showing the initialization of $\varphi$ values.

where $R_0$ denotes the initial radius of the water bubble and the subscript water refers to physical properties of water.

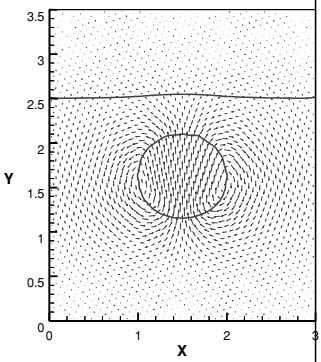
The Reynolds number used in the calculation is 200, which is the same as in [25]. The initialization of the fluid volume indicator at $t = 0$ is shown in Fig. 16. Fluid 1 is water and is assigned $\varphi = 1.0$ while the oil is assigned $\varphi = 0.0$. Figures 17 and 18 show the time evolution of the oil bubble in the partially filled container for the two cases. The free surface is shown at a time interval of 0.5. Figure 17 shows the rising bubble test case without surface tension. The upward velocity at the bottom flattens the lower half of the bubble at $t = 1.5$. The bubble deforms into a kidney shape at around $t = 2.0$. The bubble continues to deform as it approaches the water–oil interface because the upper portion of the oil bubble experiences less buoyancy and is being pushed aside by the lower half of the bubble. This elongates the oil bubble to both sides. At $t = 4.0$, the oil bubble almost hits the water–oil interface. Wave motion is observed on the water–oil surface due to the water flow beneath the surface.

A comparison of our results in Fig. 17 with those obtained by Pan and Chang [25] (shown in Fig. 19) can be made. It is noted that the two sets of results are very similar. The discrepancies in the later stages of computation can be attributed to the fact that their problem is modelled by a three-fluid system and that the fluid floating above the water is air and not oil.

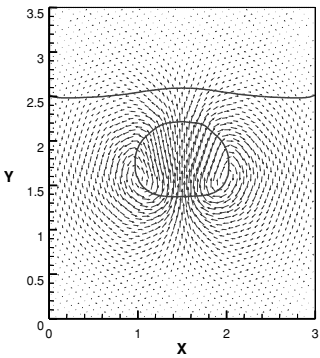
The time sequence of the rising oil bubble with surface tension is shown in Fig. 18. As in the earlier test case, the bubble accelerates from its initial position due to the buoyancy force. No significant differences between the two test cases are observed before $t = 1.5$. At times greater than 1.5, the effect of surface tension becomes obvious. With the inclusion of surface tension, the spherical oil bubble evolved into a saucer shape. Owing to the surface tension, the deformation caused by the buoyancy force on the oil bubble is less significant and no kidney-shaped bubble is observed. The lower surface of the oil bubble is also kept from catching up with the upper surface. Two vortices below the bubble can be observed and the oil bubble "bursts" at its edge instead.



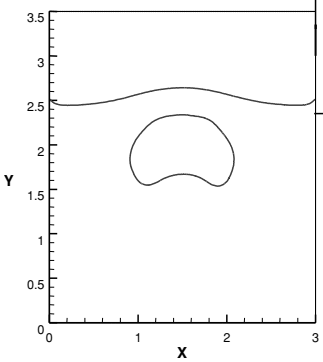

**N-D Time = 0.5**

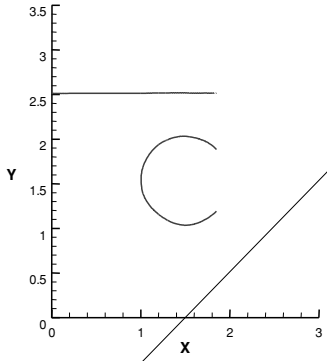


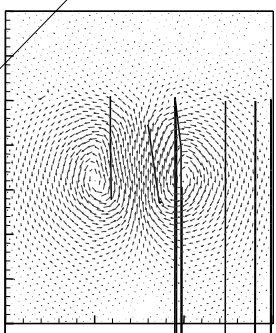

**N-D Time = 1.0**

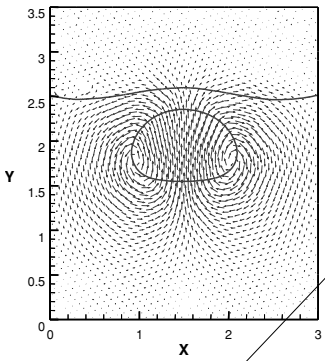



**N-D Time = 1.5**



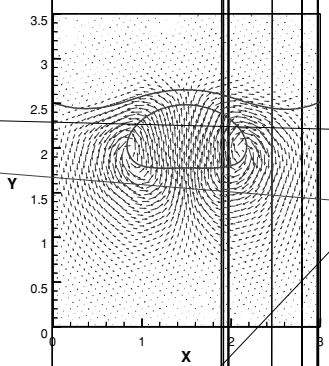


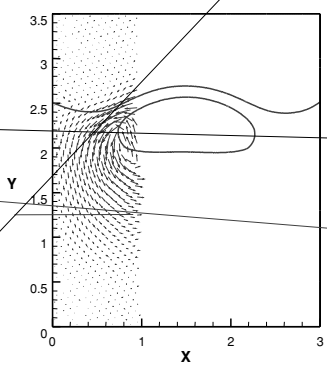






N-D Time = 1.5

N-D Time = 2.0




N-D Time = 2.5

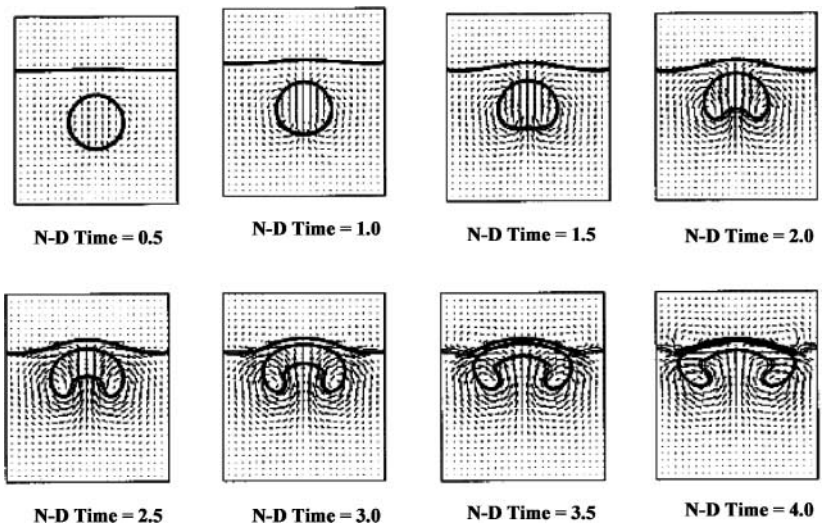



**FIG. 19.** Results of rising bubble problem by Pan and Chang [25].

### 4.4. Bubble Rising in a Fully Filled Container with Bubble Shedding

In a manner similar to the previous example, the time evolution of a two-dimensional rising bubble with surface tension was studied. The difference is that the water container is fully filled with water in this test case. In this case, the time evolution of bubbles is studied in greater depth. Parametric study is made to understand the effects of viscosity and surface tension and their interplay, as well as the effect of density ratios in a rising bubble problem. The shedding of small bubbles is also studied.

Literature review shows that numerous studies have been performed to investigate the shape evolution of a rising bubble. Experimental observations of bubbles have been reported by Walters and Davidson [28, 29] and Bhaga and Weber [30]. Lundgren and Mansour [31] used a boundary-integral method to study the evolution of a bubble from spherical to mushroom-shape, as well as the formation of a toroidal shape. Sussman *et al*. [6] used a level-set approach for computing the shape evolution of a rising bubble.

In this study, computations are performed with the five sets of nondimensionalized parameters listed in Table III. To study the effects of density ratio, Weber number, and Reynolds number, comparisons are made between the five sets of results. The computational domain for all the cases in this study is a 4 $D$-by-6.48 $D$ rectangle with $D$ being the diameter of the

**TABLE III**
**List of Test Cases**

| Test number | Figure number | $Re$ | $We$ | $\rho_{\text{water}}/\rho_{\text{air}}$ |
|---|---|---|---|---|
| 1 | 22 | 100 | 200 | 100/1 |
| 2 | 23 | 100 | 200 | 5/1 |
| 3 | 24 | 100 | 50 | 100/1 |
| 4 | 25 | 500 | 50 | 100/1 |
| 5 | 26 | 1000 | 50 | 100/1 |

initial circular bubble. The computational mesh adopted after mesh-independence testing consists of 93,312 triangular mesh elements and 47,089 nodes, which is a fine mesh due to the need to capture the fine details of the shedding of tiny bubbles.

Initially, the circular air bubble is immersed in the water that occupies the entire closed container. The air bubble is initially a perfect circle with a radius of 1.0 $D$. The center of the air bubble is located 5.0 $D$ below the top of the container. The initial velocity field of the entire computation domain is zero. The Froude number is $Fr = 1.0$. The pressure field is initialized using static hydrodynamic pressure. Uniform pressure is assigned to the bubble that is based on the hydrodynamic pressure at the center of the bubble (see Fig. 20).

The static hydrodynamic pressure at the center of the bubble is given as

$$P(x, y) = (\rho_{\text{air}} R_0 + \rho_{\text{water}} h_{\text{water}}) g.$$

At nodes outside the bubble and within the container, pressure is calculated by

$$P(x, y) = \rho_{\text{water}} h_{\text{water}} g.$$

The initialization of the fluid volume indicator at $t = 0$ is shown in Fig. 20. Fluid 1 is water and is assigned $\varphi = 1.0$, while the air bubble is assigned $\varphi = 0.0$.

When the bubble begins to accelerate from its initial position owing to buoyancy, the pressure gradient on the lower surface is greater than that on the upper surface. A vortex sheet is observed to develop and the circulation induces a jet of water that pushes the bubble from its lower surface. This motion of water jet causes the lower surface to deform and a "liquid tongue" appears. This is shown in Figs. 22 to 26. Initially the water jet does not affect the upper surface while the lower surface approaches the upper one. Due to the presence

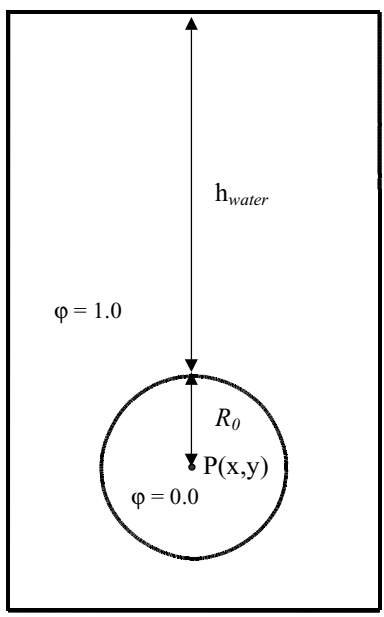



**FIG. 20.** Initialization of $\varphi$ values.

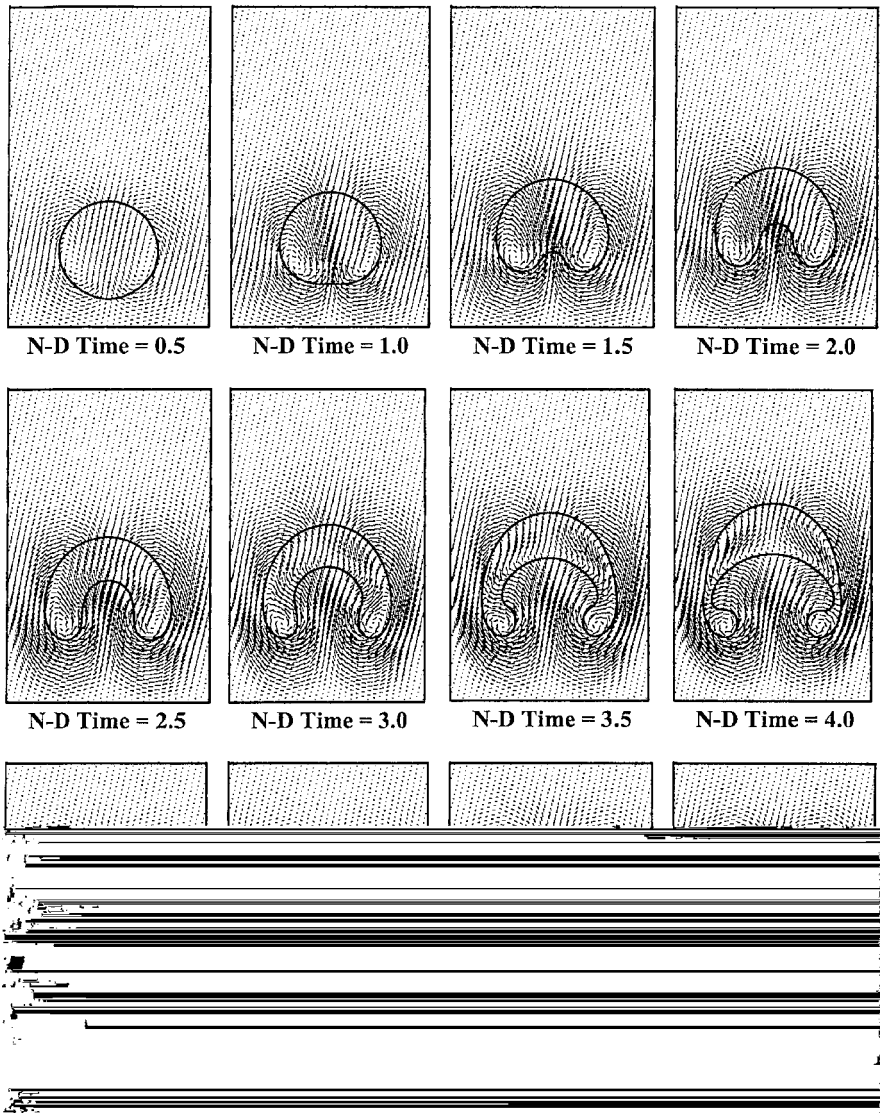



**FIG. 21.** Time evolution of rising bubble; $Re = 100$, $We = 200$, density ratio $= 100/1$.

of surface tension, the lower surface is unable to penetrate the upper one and the layer of air between them thickens again. As time evolves, the spherical bubble takes the form of a mushroom shape. Similar to results in [6], our results from Fig. 21 show the "liquid tongue" and the formation of a bubble skirt. The skirt folds inward, due to the circulation current under the rising bubble, and eventually detaches from the bubble at some time between $t = 4.5$ and $t = 5.0$. Results in [6] also show that the bubble skirt detaches from the bubble during this period ($4.40 < t < 4.80$). Finally, at $t = 6.0$, two small bubbles trailing behind the rising bubble are found in the results for Test 1.

In all our tests, it can be observed that the circulation current through the center of the bubble causes the bubble to deform, resulting in the formation of a mushroom-shaped

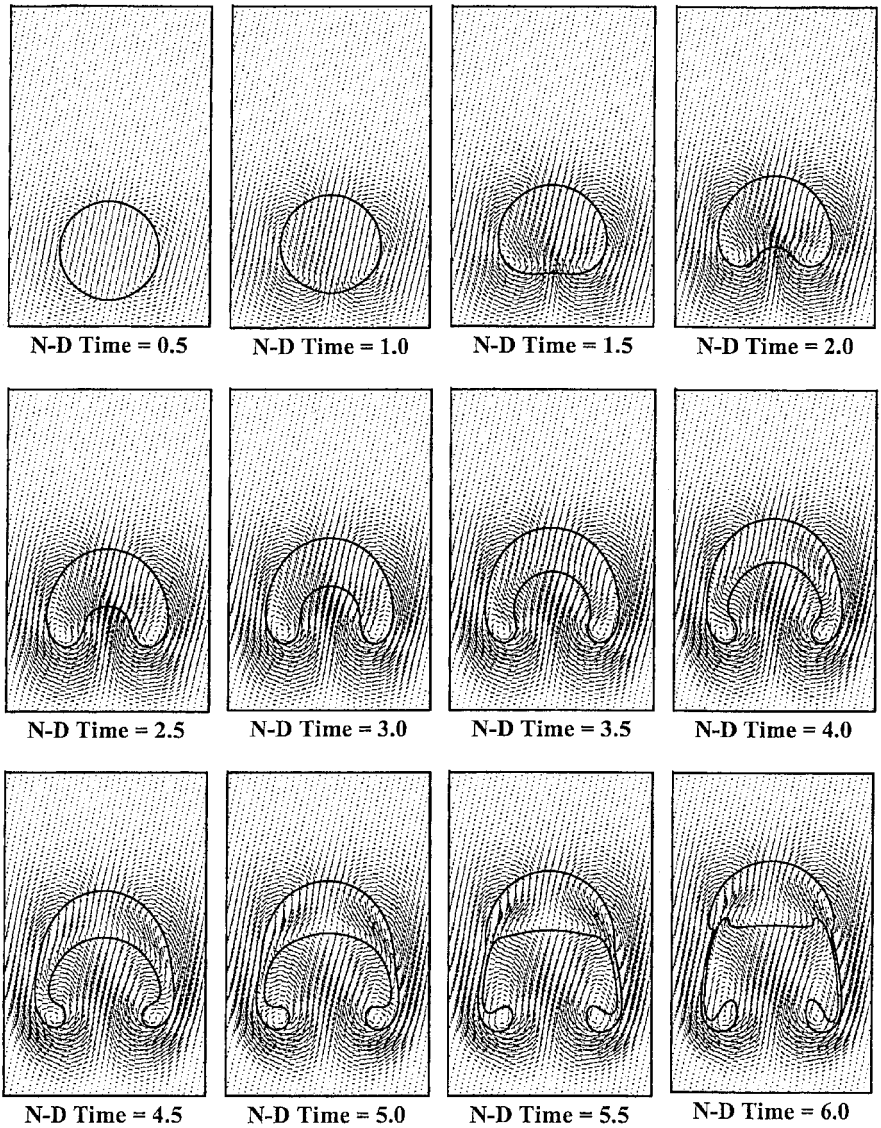



**FIG. 22.** Time evolution of rising bubble; $Re = 100$, $We = 200$, density ratio $= 5/1$.

bubble. The diameter of the deformed bubble increases as it rises. Also, the inner and outer radii of the mushroom-shaped bubble increase as it rises and the bubble is observed to "fan out." This may be explained in terms of the requirement of a stabilized circulation [31]. During the rise of their two-dimensional bubbles, Walters and Davidson [28] observed the detachment of small bubbles at the lower extremes. In all our computations, these detachments of small bubbles at the lower extremes are captured.

*Effect of Density Ratios*

The effect of density ratios of the two fluids, namely water and air, can be observed by comparing Figs. 21 and 22. In both cases, the Reynolds number ($Re = 100$) and Weber

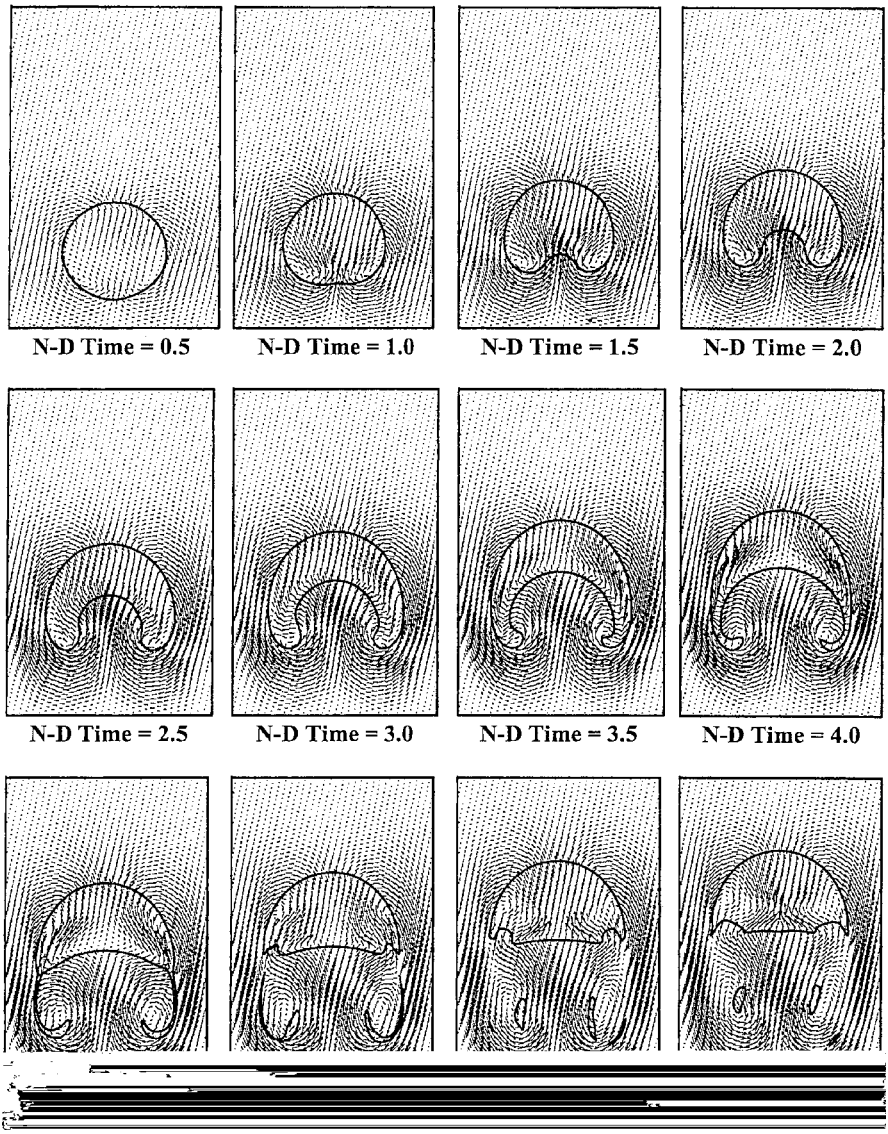



**FIG. 23.** Time evolution of rising bubble; $Re = 100$, $We = 50$, density ratio $= 100/1$.

number ($We = 200$) are identical. The only difference in the two cases is the density ratios of the two fluids. In Fig. 21, the density ratio of the two fluids is given as $\rho_{\text{water}}/\rho_{\text{air}} = 100/1$, while in Fig. 22, it is $\rho_{\text{water}}/\rho_{\text{air}} = 5/1$.

The two computations show that the circular bubble evolves into similar shapes. However, it can be observed that the bubble depicted in Fig. 21 rises faster. Figure 26 gives a graphical comparison of the maximum position reached by the bubble as a function of time for the two density ratios. An increase in density ratio leads to a faster rise of the bubble. This is because the resultant buoyancy is greater for the bubble in Fig. 21 as its density is only a hundredth of that of water. Thus it is "lighter" and rises faster as compared to the bubble depicted in Fig. 22. It can also be observed that in both test cases, the bubbles develop skirts

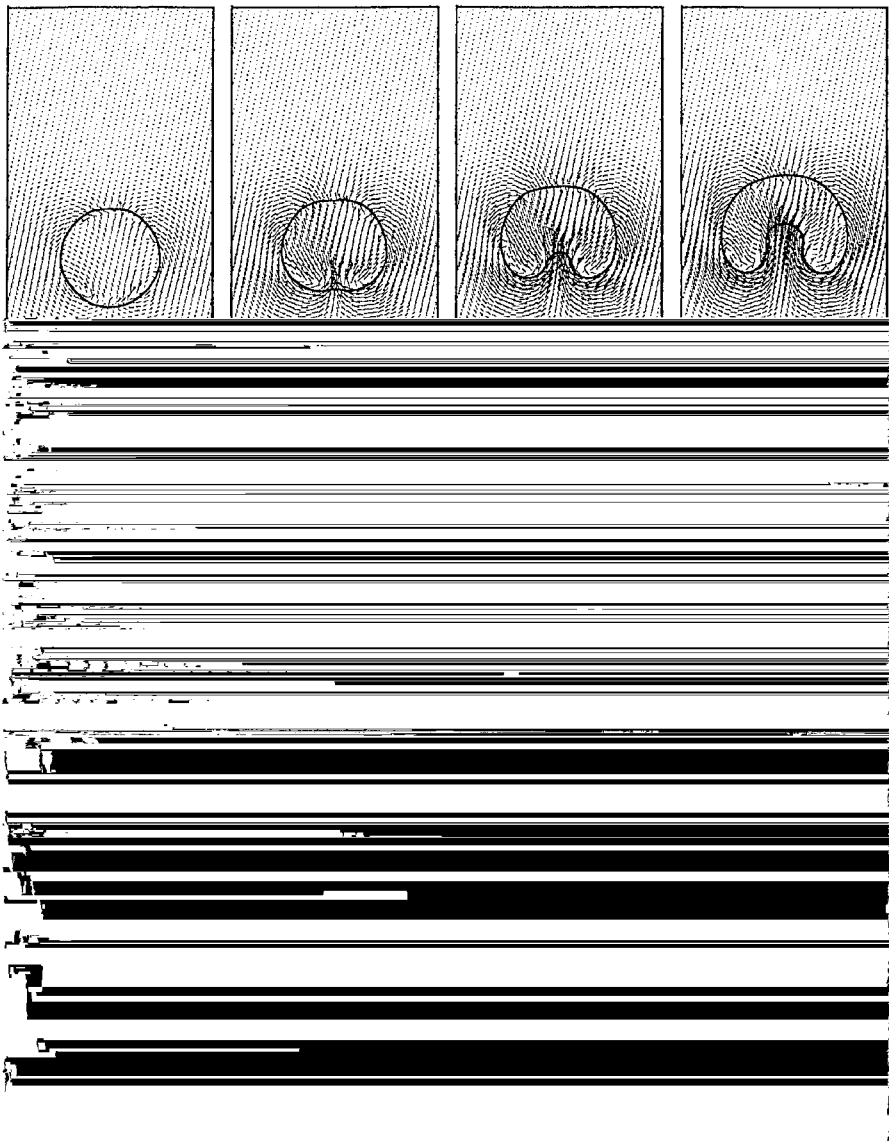


**FIG. 24.** Time evolution of rising bubble; $Re = 500$, $We = 50$, density ratio $= 100/1$.

at the two extremes. However, the bubble with the lower density ratio, shown in Fig. 22, allows the skirt to grow longer than that in Fig. 21. In Fig. 21, the skirt is broken off from the bubble at a time of about 5s. It should also be noted that the change in bubble velocity due to the change in density ratio is relatively small considering the large change in density ratio.

*Effect of Weber Number*

The effect of the Weber number is illustrated by the time evolution of bubbles in Fig. 21 ($We = 200$) and Fig. 23 ($We = 50$), which is Test 3. The Reynolds number ($Re = 100$) and

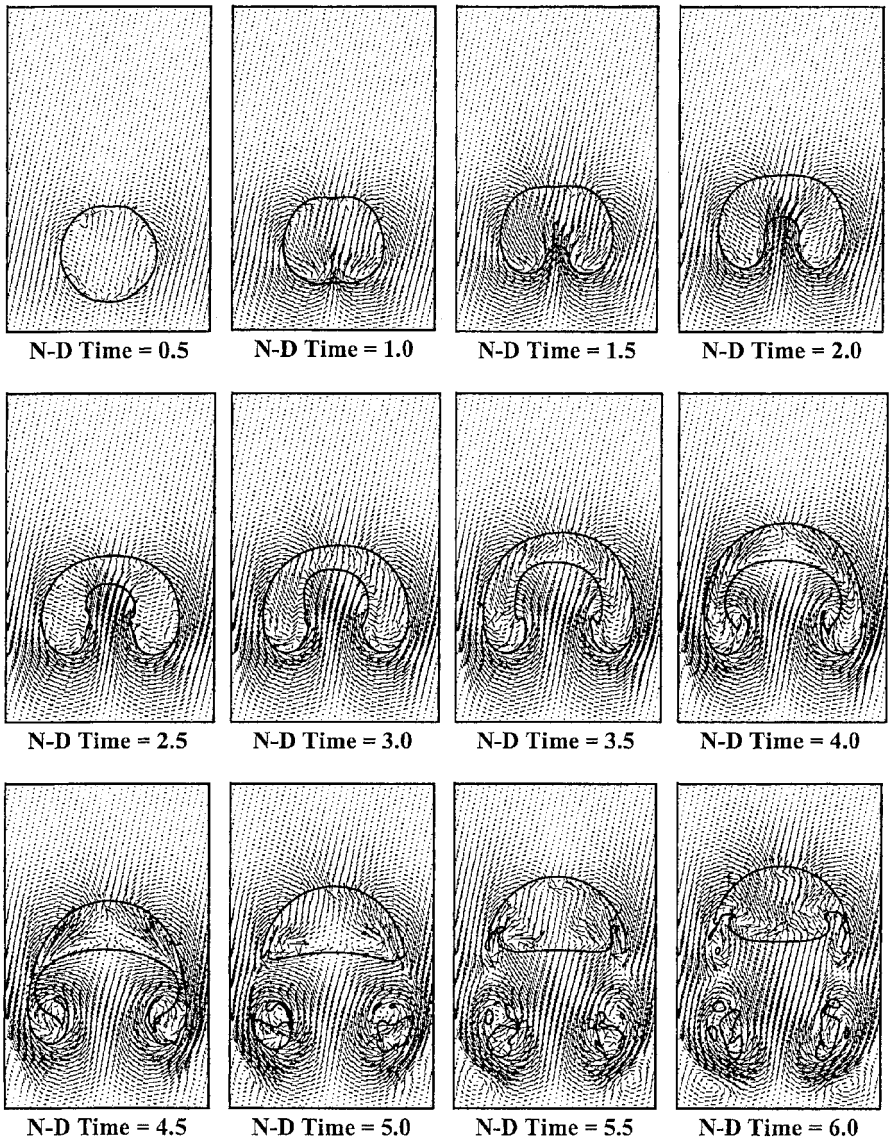



**FIG. 25.** Time evolution of rising bubble; $Re = 1000$, $We = 50$, Density ratio $= 100/1$.

density ratio are identical for both cases. It can be observed that the curvature of the deformed lower surface in Fig. 23 is lower and the "liquid tongue" piercing into the lower surface of the bubble is not as sharp as that in Fig. 21. Owing to the surface tension, the deformation caused by the "liquid tongue" on the lower surface of the air bubble is less significant. As a result, the lower surface of the bubble in Fig. 23 is unable to come as close to the upper surface as in Fig. 21. As surface tension increases, the change in shape is similar to that in [20], where the bubble skirts become thinner and the indentation at the bottom flattens out. The sizes of small bubbles that detach from the main bulk of air at the two extremes are also smaller, as shown in Fig. 21. With a higher surface tension, the integrity of the bubble increases with less shedding of small bubbles.

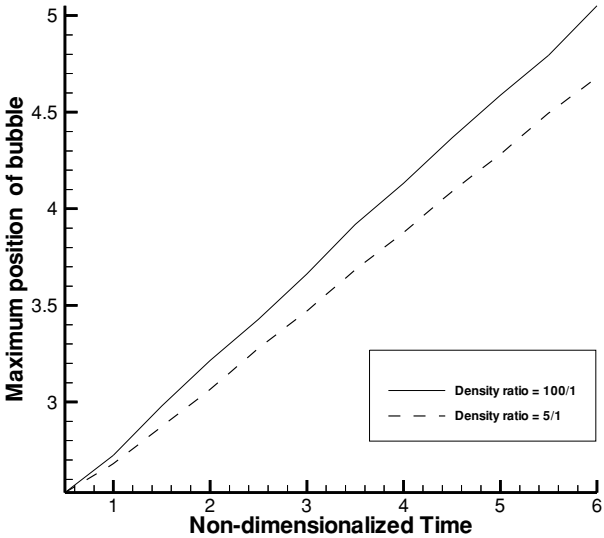



**FIG. 26.** Maximum position of bubble as a function of time.

*Effect of Reynolds Number*

To study the effect of the Reynolds number, comparisons are made of the results shown in Fig. 23 ($Re = 100$), Fig. 24 ($Re = 500$), and Fig. 25 ($Re = 1000$). For a high Reynolds number, the liquid jet below the bubble is strong and the "liquid tongue" is sharper. At a lower Reynolds number, the higher viscosity increases the form drag and reduces the liquid jet velocity. This results in a broader and more diffuse velocity distribution. From Fig. 23 ($Re = 100$), Fig. 24 ($Re = 500$), and Fig. 25 ($Re = 1000$), it can be observed that the thickness of the bubble skirt increases with increasing Reynolds numbers. The time that the skirt breaks from the main bubble is about the same (N–D Time = 4.5). However, the detached bubbles at the two extremes in Figs. 24 and 25 further disintegrate into smaller bubbles. This is due to the development of a wake at the trailing end. The magnitude of circulation is higher in the wakes at higher Reynolds numbers, causing the disintegration of the detached bubble. In Fig. 24 ($Re = 500$) and Fig. 25 ($Re = 1000$), smaller bubbles are observed to be continuously shedded from the main bubble due to higher velocity.

### 4.5. Micro Ink-Droplet Ejection

Finally a simulation of ink-droplet ejection through a micro nozzle is performed using the solver developed. The effect of ink properties on the quality of print is investigated. Two

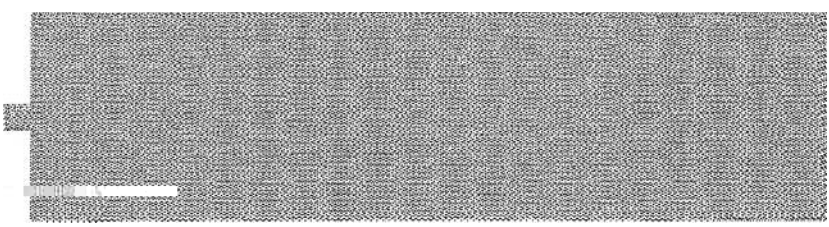


**FIG. 27.** Computational grid of nozzle.

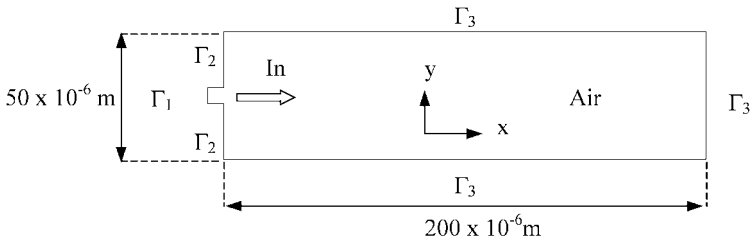



**FIG. 28.** Squirt of ink into air from a nozzle.

sets of simulations on ink-droplet ejection are performed to show how material properties of the ink would affect the size and shape of the ink droplet. The computational domain for the present simulation is shown in Fig. 27. This grid is made up of 37,264 triangular elements and 18,978 nodes. The nozzle is situated at the center of the left vertical wall and measures just 5 microns. The overall height and length of the computational domain are 50 microns and 200 microns (see Fig. 28).

In the first two simulations the effect of surface tension is investigated. The surface tension coefficients of ink for the first and second simulations are $1.73 \times 10^{-3}$ N/m and $7.36 \times 10^{-2}$ N/m, respectively. In the third and fourth simulations, the Reynolds number is varied to study the effect of Reynolds number on the motion of the droplet. Aside from the surface tension coefficient and the Reynolds number, the remaining conditions applied to the two simulations are identical. The density of the ink is set at 1000 kg/m$^3$ for all the simulations. At the outlet $\Gamma_1$ of the nozzle, the following boundary conditions are applied:

$$u_x = \hat{U}(t), \quad u_y = 0.$$

Here $\hat{U}(t)$ represents a prescribed velocity variation for the ejected ink and is assumed to be a function of time. The velocity profile of the ink-ejection pulse used in the simulation is shown in Fig. 29. From $t = 0$ to $t = 30\,\mu$s, ink is ejected with positive velocity and the

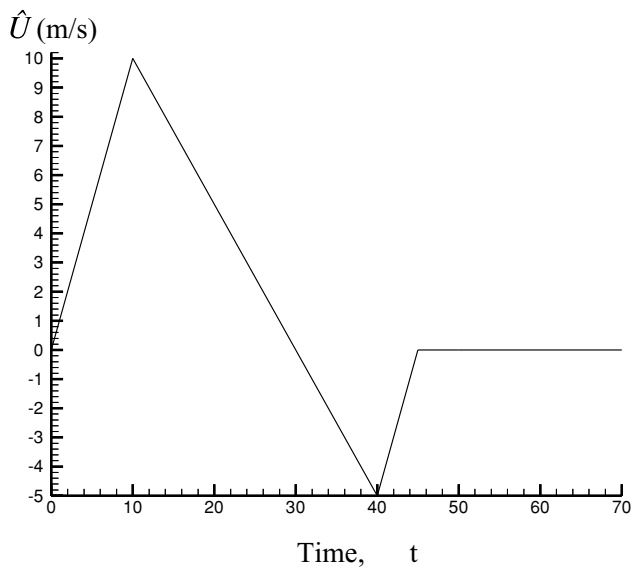



**FIG. 29.** Definition of inlet velocity $\hat{U}(t)$ at the nozzle.

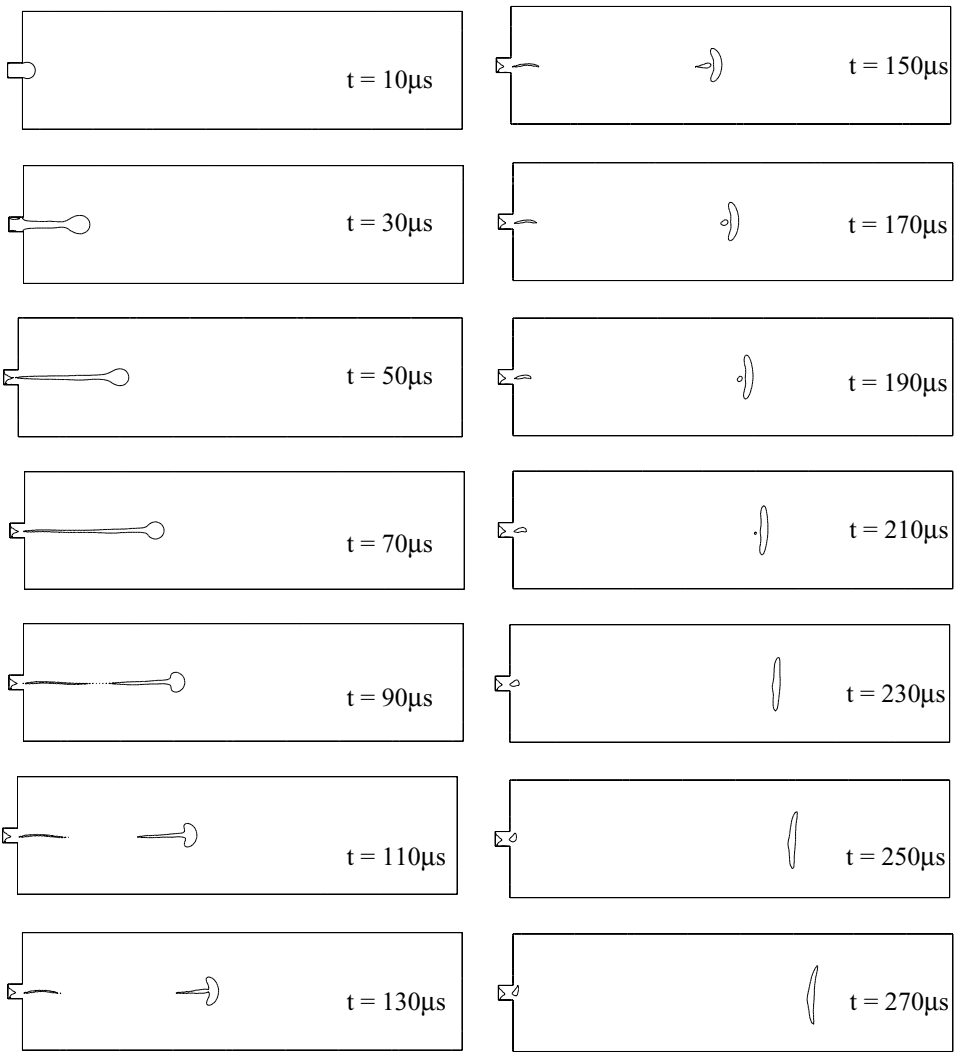



**FIG. 30.** Formation of an ink droplet ($\sigma = 1.73 \times 10^{-3}$ N/m).

peak value is set to be 10 m/s. $\varphi$ is assigned a value of 1 at boundary $\Gamma_1$. Between 30 $\mu$s and 45 $\mu$s, there is a negative velocity based on experimental observations that ink rushes back into the chamber to fill up the vacuum created by the collapse of the bubble inside the ejector chamber. Boundaries $\Gamma_2$ and $\Gamma_3$ denote the solid wall and the outflow boundaries (with a fixed pressure), respectively. The computational domain is filled with stationary air at the initial state.

The results from the simulations are illustrated in Figs. 30 to 32. By varying the value of the surface tension coefficient while keeping the remaining parameters constant, the effect of surface tension on droplet ejection could be studied. Figures 30 and 31 show the emergence of a droplet from the nozzle driven by the positive velocity due to bubble growth inside the print head chamber, which is normally produced by heating a heater. When the heating stops and the bubble collapses, the suction causes a reverse flow of the ink column. From here, the necking of the ejected volume of ink is observed. Due to the momentum, the leading tip of the ejected ink would continue in its straight line of motion, causing further necking and finally the detachment of the ink droplet.

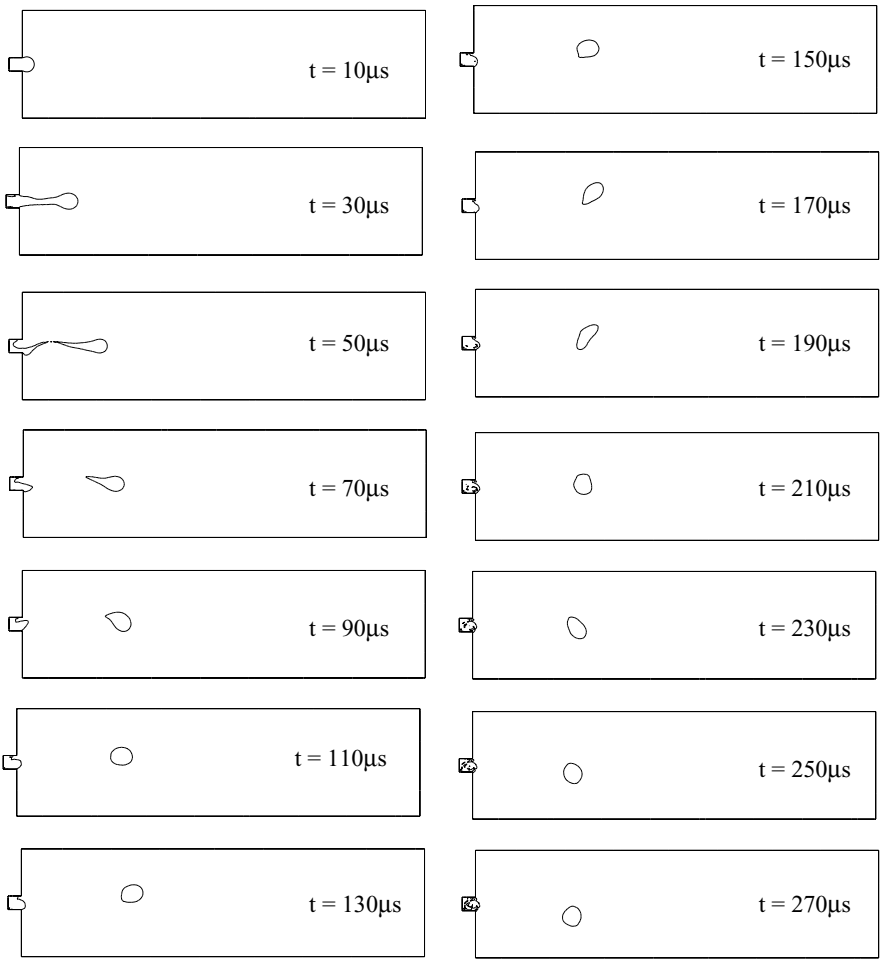



**FIG. 31.** Formation of an ink droplet ($\sigma = 7.36 \times 10^{-3}$ N/m).

In Fig. 30, it can be observed that the tail trailed the droplet front relentlessly. The shortening of this tail is much slower than that observed in Fig. 31. At a time between 130 $\mu$s and 150 $\mu$s, the tail breaks off from the bulk volume of the ejected ink droplet to form a satellite droplet. Formation of satellite droplets is a well known phenomenon in a practical inkjet printing system. They are undesirable for printing applications as they cause blurring of images because the hitting location of the main ink droplet is different from that of the satellite droplets.

In Fig. 31, the disappearance of the tail is almost instantaneous due to the higher surface-tension coefficient used. The simulation shows no sign of formation of satellite droplets and the droplet gradually forms a spherical droplet due to the effect of surface tension. The frontal area of the spherical droplet generated in this simulation is very much lower than that generated in Fig. 30. For printing applications, the effective surface area of impact of such a droplet on the print surface would be significantly smaller, thus improving the quality of print.

The effect of the Reynolds number on the motion of the ejected ink droplet is illustrated in Fig. 32. For a lower Reynolds number, the relatively larger viscosity increases the form drag and reduces the droplet velocity in addition to lowering the fluid moment. This leads

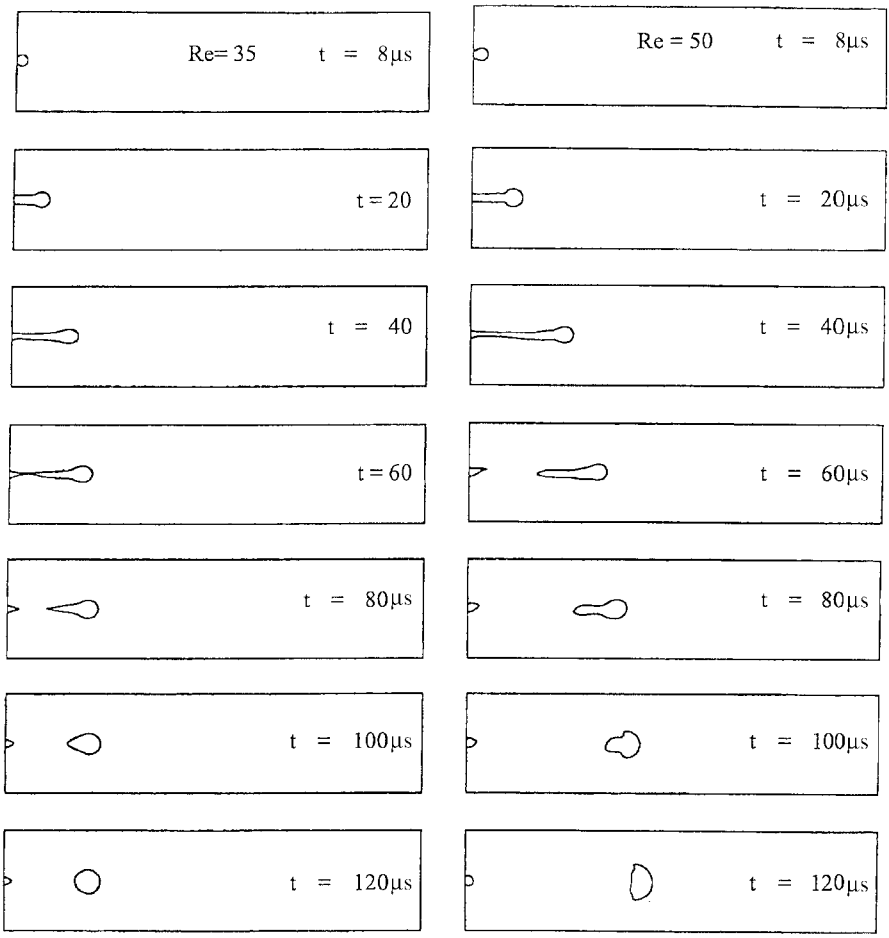



**FIG. 32.** Droplet motion with different Reynolds numbers. $t$ measured in microseconds.

to a more diffuse velocity distribution, thus causing the droplet to travel a shorter distance in a given period. The droplet with a lower Reynolds number is also observed to assume a spherical shape faster than its counterpart. This could be attributed to the surface tension force becoming more dominant given the lower Reynolds number. From the above, it is obvious that chemical engineers have to control the physical properties of the ink used, such as viscosity and the surface tension coefficient, by changing the components of the ink so that droplets with optimal properties can be produced.

## 5. CONCLUDING REMARKS

In this paper, a new method has been successfully developed to simulate complex free surface flows efficiently. The unstructured grid approach makes the method very flexible in dealing with complex boundaries. The higher order characteristic-based finite-volume scheme and the matrix-free implicit dual time-stepping in conjunction with the VOF method makes it possible to handle complex free surface flows accurately and efficiently. In particular, the high-order characteristics-based scheme is found to be more accurate than the

second-order central scheme in numerical tests. Several cases of free surface flows have been studied with the developed method and the results are excellent in comparison with experimental measurements and other numerical results. Based on the method, indepth and parametric studies are performed of bubble rising in liquid and the ejection of micron-sized droplets from a micro nozzle.

## REFERENCES

1. C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* **39**, 201 (1981).
2. J. U. Brackbill, D. B. Kothe, and C. Zemach, A continuum method for modelling surface tension, *J. Comput. Phys.* **100**, 335 (1992).
3. J. E. Welch, F. H. Harlow, J. P. Shannon, and B. J. Daly, *The MAC Method. A Computing Technique for Solving Viscous, Incompressible, Transient Fluid-Flow Problems Involving Free Surfaces*, LASL Report LA-3425, Los Alamos (1966).
4. S. Osher and J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).
5. M. Sussman,

22. F. J. Felecy and R. H. Pletcher, Numerical simulation of free surface flows in closed containers using free surface capture approach, in *Advances in Computational Methods in Fluid Dynamics* (Am. Soc. Mech. Eng., 1994), FED-Vol. 196, pp. 33–51.
23. B. J. Daly, Dynamics of liquids in moving containers, *Phys. Fluids* **10**, 297 (1967).
24. J. C. Martin and W. J. Moyce, An experimental study of the collapse of liquid columns on a rigid horizontal plane, *Philos. Trans. R. Soc. London, Ser. A* **244**, 312 (1952).
25. D. Pan and C. H. Chang, The capturing of free surfaces in incompressible multi-fluid flows, *Int. J. Numer. Meth. Fluids* **33**, 203 (2000).
26. B. Ramaswamy and M. Kawahara, Lagrangian finite element analysis applied to viscous free-surface fluid flow, *Int. J. Numer. Meth. Fluids* **7**, 953 (1987).
27. S. Koshizuka, H. Tamako, and Y. Oka, A particle method for incompressible viscous flow with fluid fragmentation, *Comput. Fluid Dyn. J.* **4**, 29 (1995).
28. J. K. Walters and J. F. Davidson, The initial motion of a gas bubble formed in an inviscid liquid. Part 1. The two-dimensional bubble, *J. Fluid Mech.* **12**, 408 (1962).
29. J. K. Walters and J. F. Davidson, The initial motion of a gas bubble formed in an inviscid liquid. Part 2. The three-dimensional bubble and the toroidal bubble, *J. Fluid Mech.* **17**, 321 (1963).
30. D. Bhaga and M. E. Weber, Bubbles in viscous liquids: Shapes, wakes and velocities, *J. Fluid Mech.* **105**, 61 (1981).
31. T. S. Lundgren and N. N. Mansour, Vortex bubbles, *J. Fluid Mech.* **224**, 177 (1991).
32. A. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* **2**, 12 (1967).
33. A. Jameson and D. J. Mavriplis, Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh, *AIAA J.* **24**, 611 (1986).
34. D. B. Kothe, R. C. Ferrel, J. A. Turner, and S. J. Mosso, *A High Resolution Finite Volume Method for Efficient Parallel Simulation of Casting Processes on Unstructured Meshes*, Los Alamos National Laboratory Report LA-UR-97-30 (1997).