AN IMPROVED THREE-DIMENSIONAL LEVEL SET METHOD FOR GAS-LIQUID TWO-PHASE FLOWS

Hiroyuki TAKAHIRA
Department of Energy Systems Engineering
Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
Email: takahira@energy.osakafu-u.ac.jp

Tomonori HORIZUCHI
Graduate School of Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
Email: horiuchi@energy.osakafu-u.ac.jp

Sanjoy BANERJEE
Department of Chemical Engineering
University of California
Santa Barbara, CA 93106, USA
Email: banerjee@engineering.ucsb.edu

ABSTRACT
In the present study, we developed a three-dimensional numerical method based on the level set method that was applicable to the two-phase systems with high-density ratio. We considered the pressure jump due to the surface tension, as well. The present solver for the Navier-Stokes equations was based on the projection method with a non-staggered grid. A semi-implicit time-advancement scheme with Adams-Bashforth method was used for the explicit terms and the Crank-Nicolson for the implicit terms. We improved the treatment of the convection terms and the interpolation method that was used to obtain the intermediate volume flux defined on the cell faces. We also improved the solver for the pressure Poisson equations and the reinitialization procedure of the level set function. It was shown that the present solver worked very well even though the density ratio of two fluids was 1:1000. We simulated the coalescence of two rising bubbles under gravity and the gas bubble bursting at a free surface to evaluate the mass conservation of the present method. It was also shown that the volume conservation (i.e., mass conservation) of bubbles was very well even after the bubble coalescence.

NOMENCLATURE
Fr: Froude number

Re: Reynolds number

S: sign function

t: time

U: volume flux

u: velocity

V: Volume

We: Weber number

x, y, z: Cartesian coordinates

ρ: density

σ: surface tension

τ: shear stress or iteration time

ϕ, d: level set function (distance function)

Subscripts

g: gas

l: liquid

INTRODUCTION
Developing direct numerical simulations of two-phase fluids is very useful to complement difficult experiments on the interactions between turbulent flows...
and phase change at the boundaries e.g., convective condensation, boiling and electrochemical gas generation. There are several numerical methods to treat the interface of two-phase fluids, such as VOF method (Hirt and Nichols, 1981) and Front Tracking method (Juric and Tryggvason, 1998). One of the useful methods among them is the level set method. The level set method is based on an Eulerian formulation that describes the interface by the zero level of a Lipschitz-continuous function (Osher and Sethian, 1988, Sussman and Smerela, 1997, Son and Dhir, 1998, Sussman, et al., 1998, Sussman, et al., 1999, Himeno and Watanabe, 1999, Sussman and Puckett, 2000). The interface is captured implicitly on the Eulerian grid by the zero level set. There are many advantages to such a formulation. However, the level set method has been applied mainly to two-dimensional and two-dimensional axisymmetric problems, and little work has been done on the three-dimensional problems for two-phase flows with high-density ratio. In the three-dimensional level set method, it is difficult to analyze the two-phase fluids with high-density ratio due to the numerical instability near the interface. The mass conservation of the two fluids systems is another difficulty in the level set method. Overall objective of this study is to develop a three-dimensional numerical method based on the level set method for the two-phase systems with high-density ratio in the curvilinear coordinates.

The present solver for the Navier-Stokes equations is based on the numerical scheme developed by Zang et al. (1994). They used a fractional step method (Projection Method) with a non-staggered grid in which the volume flux is defined on its corresponding face of the cell in addition to the Cartesian velocity components at the cell center. This non-staggered grid was used to calculate the flow field in the curvilinear coordinates with high accuracy and a small amount of memory.

When we applied the above numerical method developed for the single incompressible flow to the multiphase flow, serious numerical instability was found when the density ratio was high. The main reason for the instability was the interpolation method for cell faces and the solver for pressure Poisson equation. Therefore, to analyze the two-phase systems, we improved the treatment of the convection terms and the interpolation method that was used to obtain the intermediate volume flux defined on the cell faces. We also improved the solver for the pressure Poisson equations. To achieve the perfect mass conservation of two fluids systems, we also improved the reinitialization procedure developed by Sussman et al. (1998).

**NUMERICAL METHOD**

**Basic equations**

The Navier-Stokes equations are extended so as to consider the variable density and viscosity of two fluids. The governing equations in the present study are:

\[
\frac{\partial u_i}{\partial x_j} = 0, \quad (1)
\]

\[
\rho(\varphi)\frac{\partial u_i}{\partial t} + \rho(\varphi)\frac{\partial}{\partial x_j}(u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial \tau_{i,j}}{\partial x_j} - \left(\sigma \kappa(\varphi)\delta \varphi \right)_{\delta_{i,j}}, \quad (2)
\]

where

\[
\tau_{i,j} = \mu(\varphi) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \quad (3)
\]

\[
\kappa(\varphi) = \frac{\partial n_j}{\partial x_j}, \quad n_i = \frac{\partial x_i}{\partial \varphi}, \quad (4)
\]

\( u \) is the velocity component, \( \rho \) the density, \( p \) the pressure, \( \sigma \) the surface tension, \( \mu \) the viscosity, \( g \) the gravity, \( \delta \) the delta function, and \( \delta_{i,j} \) the Kronecker’s delta. The gravity acts in the \(-z\) direction. \( \varphi \) is the distance function from the interface and is called the level set function.

Introducing the following dimensionless variables,

\[
x^* = L x^*, \quad u^*_i = U_0 u_i, \quad t^* = (L/U_0)t, \quad p^* = p' \rho U_0^2, \quad \rho = \rho \mu', \quad \mu = \mu \mu',
\]

and substituting these variables into Eq. (2), and dropping the primes, we have

\[
\rho(\varphi)\frac{\partial u_i}{\partial t} + \rho(\varphi)\frac{\partial}{\partial x_j}(u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial \tau_{i,j}}{\partial x_j} - \left(\frac{1}{We}\sigma \kappa(\varphi)\delta \varphi \right)_{\delta_{i,j}} - \frac{\rho g}{F_r}, \quad (5)
\]

The dimensionless groups used above are the Reynolds number, the Weber number and the Froude number. These are given, respectively, by

\[
Re = \frac{\rho L U_0}{\mu}, \quad We = \frac{\rho L U_0^2}{\sigma}, \quad Fr = \frac{U_0^2}{gL}.
\]

![Fig. 1: Non-staggered grid](image-url)
where $L$ and $U_0$ are the characteristics length and velocity, respectively. The dimensionless density and viscosity are function of $\varphi$ and are given, respectively, as

$$\rho(\varphi) = \{\lambda + (1 - \lambda)H(\varphi)\},$$

(6)

$$\mu(\varphi) = \{\eta + (1 - \eta)H(\varphi)\},$$

(7)

where $\lambda = \rho_2/\rho_1$ and $\eta = \mu_2/\mu_1$. $H$ is a smoothed Heaviside function and is written as

$$H(\varphi) = \begin{cases} 0, & \text{if } \varphi < -\varepsilon, \\ \frac{1}{\varepsilon} \left(1 + \frac{\varphi}{\varepsilon} + \frac{1}{\pi} \sin(\pi \varphi/\varepsilon)\right), & \text{if } -\varepsilon \leq \varphi \leq \varepsilon, \\ 1, & \text{if } \varphi > \varepsilon, \end{cases}$$

$\varepsilon$ is the thickness of the interface and is taken to be $\alpha \times h$ where $h$ is the grid spacing. $\alpha$ is taken to be 2 in the present calculation. Smoothed delta function $\delta(\varphi)$ is defined as $dH(\varphi)/d\varphi$. The viscous terms are separated into two terms as

$$\frac{1}{\rho} \frac{\partial}{\partial x_j} \left( \mu(\varphi) \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) = \frac{1}{\rho} \frac{\partial}{\partial x_j} \left( \mu(\varphi) \frac{\partial u_i}{\partial x_j} \right) + \frac{1}{\rho} \frac{\partial}{\partial x_j} \left( \mu(\varphi) \frac{\partial u_j}{\partial x_i} \right).$$

(8)

The second term of right-hand side of Eq. (8) has non-zero values only inside the interface because $\mu$ is constant far from the interface.

The level set function is obtained from solving the following equation:

$$\frac{\partial \varphi}{\partial t} + u_i \frac{\partial \varphi}{\partial x_j} = 0.$$  

(9)

**Discretization**

The present solver for the Navier-Stokes equations is based on the numerical scheme developed by Zang et al. (1994) and Zang and Street (1995). Their solution method is developed for the incompressible Navier-Stokes equations for single-phase flow in the general curvilinear co-ordinate system. The continuity equation (1) is, for example, transformed into the general curvilinear co-ordinate system in strong law form as,

$$\frac{\partial U_m}{\partial t} = 0,$$

(10)

where $m=1,2,3$, the volume flux $U_m$ and the Jacobian $J$ are given by

$$U_m = J^{-1} \frac{\partial \xi_j}{\partial \xi_i} U_i,$$

(11)

$$J^{-1} = \det \left( \frac{\partial \xi_i}{\partial \xi_j} \right).$$

(12)

They used a non-staggered grid to discretize the governing equations. As well known, solution methods for the incompressible N-S equations based on a traditional non-staggered grid, in which all the variables are defined only at the cell center, produce spurious oscillations in the pressure field, i.e., "the checkerboard" pattern (Patankar, 1980). One of the fundamental causes is that, in a traditional non-staggered grid, a straightforward discretization of the continuity equation does not enforce mass conservation in the cell and causes decoupling of the pressure field. To prevent the decoupling, in their non-staggered grid, the volume flux is defined on its corresponding face of the cell in addition to the Cartesian velocity components at the cell center as in Fig. 1. Therefore, both the momentum and the continuity equations are enforced in the same control volume. Also, this non-staggered grid is useful in the analysis for the curvilinear coordinates with high accuracy and a small amount of computer memory.

In the present two-phase codes, we used the same non-staggered grid. Although our present solver is written for the Cartesian co-ordinate system with unequal spacing, to simply the derivation of the discretized equation, the following formulation is restricted to the Cartesian co-ordinate system with equal spacing. We also use $u$ as the Cartesian velocity components and $U$ as the volume flux (the velocity on the cell face in the Cartesian coordinates) to distinguish both of them.

We use a semi-implicit time-advancement scheme with Adams-Bashforth method for the explicit terms and the Crank-Nicolson for the implicit terms. A fractional step method (Projection Method) is used to solve the N-S equations.

The discretized equations are written as:

$$\frac{\delta U_i}{\delta x_i} = 0,$$

(13)

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{3}{2} \left[ C_i^n + D_{Ei}^n \right] - \frac{1}{2} \left[ C_i^{n-1} + D_{Ei}^{n-1} \right] + R_i(u_i^{n+1})$$

$$+ \frac{1}{2} \left[ D_i(u_i^{n+1} + u_i^n) \right] + F_i^{n+1},$$

(14)

where $\delta/\delta x_i$ denotes the discrete finite difference operator, superscripts represent the time step, $C_i$ is the convective term, $R_i$ is the discrete operator for the pressure gradient terms and $D_{Ei}$ is the explicitly treated viscous terms and $D_t$ is the discrete operator representing the implicitly treated viscous terms, $F_i$ is the sum of the surface tension term and the gravity term. $C_i$, $R_i$, $D_t$ and $D_{Ei}$ are, given, respectively by

$$C_i = - \frac{\delta}{\delta x_i} (U_j u_i),$$

$$R_i = - \frac{1}{\rho(\varphi)} \left( \frac{\delta}{\delta x_i} \right),$$

$$D_t = \frac{1}{\rho(\varphi)} \frac{\delta}{\delta x_i} \left( \mu(\varphi) \left( \frac{\delta}{\delta x_i} \right) \right).$$

Copyright © 2003 by ASME
Application of the fractional step method to Eq. (14) leads to the following predictor-corrector solution procedure.

1. Predictor

\[ \left( I - \frac{\Delta t}{2} D_t \right) (u_i^n - u_i^*) = \Delta t \left\{ \frac{3}{2} C_{n+1}^n + D_{E_i}^n \right\} - \frac{1}{2} [C_{n+1}^{n-1} + D_{E_i}^{n-1}] + D_I (u_i^n) + F_i^{n+1} \],

where \( I \) is the identity matrix and \( u_i^* \) is the intermediate velocity. The variable \( \psi \) is related to the pressure \( p \) as

\[ R_t(p) = \left( I - \frac{\Delta t}{2} D_t \right) R_t(\psi), \]

(17)

hereafter \( \psi \) is referred as the 'pressure'.

We discretize the convection terms \( C_t \) using the 2nd order ENO scheme (Sussman, at al., 1998). The first viscous term of RHS of Eq. (8) is discretized implicitly using the Crank-Nicolson as:

\[
\frac{1}{\rho} \left( \frac{\partial}{\partial x_j} \right) \left\{ \mu(\varphi) \left( \frac{\partial u_i^*}{\partial x_j} \right) \right\} \approx \frac{1}{2} \frac{1}{\rho^{n+1}} \left( \mu^{n+1} \left( \frac{\delta u_i^*}{\delta x_j} \right) \right) + \frac{1}{\rho^n} \left( \mu^n \left( \frac{\delta u_i^*}{\delta x_j} \right) \right).
\]

(18)

On the other hand, the second term of RHS of Eq. (8) is discretized explicitly. The second order central difference is used to calculate viscous terms. It should be noted that we can obtain \( \varphi, \rho, \) and \( \mu \) at time \( t = t_i^{n+1} \) by solving the level set equation (9) before calculating the N-S equations.

The surface tension terms are discretized using \( \varphi \) at \( t = t_i^{n+1} \) as:

\[
\frac{1}{\rho(\varphi)} \left( \sigma \kappa(\varphi) \delta(\varphi) \frac{\partial \varphi}{\partial x_j} \right) \approx \frac{1}{\rho^{n+1}} \left( \sigma \kappa^{n+1} \delta^{n+1} \frac{\delta \varphi^{n+1}}{\delta x_j} \right).
\]

(19)

The second order central differences are used for the surface tension terms.

Equation (15) is solved with the approximate factorization technique in which the LHS of Eq. (15) is factorized into the tridiagonal matrices as

\[
\left( I - \frac{\Delta t}{2} D_t \right) \left( I - \frac{\Delta t}{2} D_2 \right) \left( I - \frac{\Delta t}{2} D_3 \right) (u_i^n - u_i^*) = RHS \text{ of \ (15)},
\]

(20)

where \( D_k \) (k = 1, 2, 3) is the discrete one-dimensional diagonal viscous operator and \( D_I = D_1 + D_2 + D_3 \).

To update the velocity components at the cell center, the pressure gradient in Eq. (16) is discretized as

\[
R_t(\psi^{n+1}) \approx -\frac{1}{2} \frac{1}{\rho_t^{n+1/2}} \frac{\psi^{n+1} - \psi^{n+1}}{\Delta x_i} + \frac{1}{\rho_t^{n+1/2}} \frac{\psi^{n+1} - \psi^{n-1}}{\Delta x_i}.
\]

(21)

When we use the standard central difference:

\[
R_t(\psi^{n+1}) \approx -\frac{1}{\rho_t^{n+1/2}} (\psi^{n+1/2} - \psi^{n-1/2}),
\]

(22)

numerical oscillations occur when the density ratio is high although the equations (21) and (22) are identical with each other if the density is constant. After correcting step (16) the volume flux on the cell face (=\( U_{i+1/2}^{n+1} \)) is updated with

\[
U_{i+1/2}^{n+1} = U_{i+1/2}^* - \Delta t \left( \frac{1}{\rho^{n+1/2}} \frac{\delta \psi^{n+1}}{\delta x_i} \right)
\]

\[
\approx U_{i+1/2}^* - \frac{\Delta t}{\rho^{n+1/2}} \frac{\psi^{n+1} - \psi^{n+1}}{\Delta x_i},
\]

(23)

where \( U_{i+1/2}^* \) is the intermediate volume flux defined on the cell faces. The discretization with Eq. (23) is consistent with Eq. (21). To calculate \( U_{i+1/2}^* \), the value of \( u_i^* \) which is defined at the cell center is interpolated onto cell faces with third-order quadratic upwind interpolation scheme to that used in QUICK (Zang, et al., 1994). However, as shown in the later section, when we used \( U_{i+1/2}^* \) to determined the flow direction in the upwind interpolation, numerical oscillations occur when the density ratio of two fluid is high and the surface tension is considered. To avoid this problem, we use the following velocity to determine the flow direction on the cell faces inside the interface for the upwind interpolation scheme:

\[
\Delta U_i = -\frac{\Delta t}{\rho^{n+1}} \left( \sigma \kappa^{n+1} \delta^{n+1} \frac{\delta \varphi^{n+1}}{\delta x_i} \right).
\]

(24)

\( \Delta U_i \) is regarded as the velocity increase induced by the pressure gradient due to the surface tension. The flow direction outside the interface is determined from the direction of the volume flux at \( t = t_i^{n+1} \) on the cell faces.

Before \( u_i^{n+1} \) can be obtained from Eq. (16) or \( U_{i+1/2}^{n+1} \) from Eq. (23), we need to solve \( \psi^{n+1} \) from the following pressure Poisson equations:

\[
\frac{\delta}{\delta x_j} \left( \frac{1}{\rho^{n+1}} \frac{\delta \psi^{n+1}}{\delta x_j} \right) = \frac{1}{\Delta t} \frac{\delta U_i^*}{\delta x_i}.
\]

(25)

The LHS of Eq. (25) is discretized as

\[
\frac{\delta}{\delta x_j} \left( \frac{1}{\rho^{n+1}} \frac{\delta \psi^{n+1}}{\delta x_j} \right) \approx \frac{1}{\Delta x} \left[ \frac{1}{\rho_{i+1/2,j,k}^{n+1}} \psi_{i+1/2,j,k}^{n+1} - \psi_{i,j,k}^{n+1} \Delta x \right].
\]
The density ratio is 1:10000. When we applied the multigrid method for the pressure Poisson solver, the BiCGSTAB method works very well for the two fluids with high-density ratio even though the density ratio is 1:10000. We used ILU decompositions (L is lower triangular and U (Van der Vorst, 1992, Fujino and Zhang, 1996). We solved Equation (25) is solved using the preconditioning BiCGSTAB (BiConjugate Gradient Stabilized) method.

When we applied this procedure for the rising bubble problem, about 10% mass was lost. To recover the mass, we add the multiplier of the order of one to the constraint term, and we solve the following equation:

\[ d^{k+1} = \frac{d^k + \tau_{k+1}\lambda_{i,j,k}(\vec{d}^{k+1})f(\varphi)}{1 + \tau_{k+1}\lambda_{i,j,k}(\vec{d}^{k+1})} \]

where \( \lambda_{i,j,k} \) is the grid cell. The last term in Eq. (27) is used to keep the bubble volume constant during the reinitialization procedure. We solve Eq. (27) until the \(|d| = 1 \) near the interface and then replace \( \varphi(x) \) by the \( d(x, \tau_{\text{steady}}) \) when Eq. (27) converges near the interface for the next time step of Eq. (9). In the present calculation, the convection term in Eq. (27) is discretized using the 2nd order ENO scheme. The 2nd order Runge-Kutta method is used in order to advance in \( \tau \).

When we applied this procedure for the rising bubble problem, about 10% mass was lost. To recover the mass, we add the multiplier of the order of one to the constraint term, and we solve the following equation:

\[ d^{k+1} = \frac{d^k + \tau_{k+1}\lambda_{i,j,k}(\vec{d}^{k+1})f(\varphi)}{1 + \tau_{k+1}\lambda_{i,j,k}(\vec{d}^{k+1})} \]

where \( \varphi \) denotes the initial value of \( d \), and \( \vec{d}^{k+1} \) denotes the distance function \( d \) at \( \tau = \tau_{k+1} \) in the Runge-Kutta procedure of Eq. (27) without the constraint term.

When we applied this procedure for the rising bubble problem, about 10% mass was lost. To recover the mass, we add the multiplier of the order of one to the constraint term, and we solve the following equation:

\[ d^{k+1} = \frac{d^k + \tau_{k+1}\lambda_{i,j,k}(\vec{d}^{k+1})f(\varphi)}{1 + \tau_{k+1}\lambda_{i,j,k}(\vec{d}^{k+1})} \]

where \( c \) is taken to be 1 \( \sim 1.05 \). Since this multiplier works so as to increase the mass, wrong multiplier may increase the mass too much. Therefore we apply the multiplier only when the mass at each iteration step is less than the initial (or actual) mass. This simple algorithm works very well.

We continue the above iteration until the following criterion is satisfied:

\[ E = \sum_{|d| < \varepsilon} \frac{|d^{k+1} - d^k|}{M} < C \Delta \tau h^2 \]

where \( M \) is the number of grid points where \(|d| < \varepsilon\) and \( h \) is the grid spacing. \( C \) and \( \Delta \tau \) are taken to be 0.1 and 0.5\( h \), respectively.

**NUMERICAL RESULTS AND DISCUSSION**

Some numerical tests

We tested the present solver for some simple problems. Figure 2 shows the influence of the density ratio on the pressure Poisson equations. Test conditions are as follow. A spherical bubble with unit radius is at rest in the \( 4 \times 4 \times 4 \) cubic box. No-slip boundary conditions are applied to the every sides of the cube. We set \( R_e=50 \), \( \rho_e=1 \), and \( F_r=0 \). Therefore, the actual pressure jump at the interface is 2. The time interval \( \Delta t \) is taken to be 1.05.
be 0.002. Figure 2 is the pressure distribution at $t=\Delta t$. Solid lines, dashed lines, dashed lines with one dot and dotted line denote the results when the density ratio $\lambda = \rho_g/\rho_l = 0.1, 0.01, 0.001$ and 0.0001, respectively. We used $51\times51\times51$ mesh with equal spacing in Fig. 2(a) and $102\times102\times102$ mesh with equal spacing in Fig. 2(b), respectively. As the density ratio becomes small in the coarse grid, a undershoot is found near the interface. However, even though the density ratio is 1:10000, the error of the pressure jump is less than about 8 %. The finer the mesh becomes, the smaller the undershoot becomes. In Fig. 2(b), the error is less than 5 % when the density ratio is 1:10000.

Figure 3 shows the influence of the interpolation method for the intermediate volume flux. Test conditions are the same as Fig. 2 except for $Fr=1$. The pressure contour at $t=0.2$ is shown in Fig. 3. We used the 3rd order upwind quadratic interpolation in every cases. However, in Fig. 3(a) the flow direction on the cell faces inside the interface is evaluated with the velocity (volume flux) on the cell faces at the previous time step ($=U_i^n$). While in Fig. 3(b) and Fig. 3(c), the flow direction on the cell faces is evaluated with Eq. (24). As evident from Fig. 3(a), serious pressure oscillations appear when using the volume flux at previous time step. On the other hand, the numerical oscillations diminish when we evaluate the flow direction with Eq. (24). This may be because that the flow direction of the intermediate volume flux is different from the previous time step because the intermediate volume flux is accelerated by the pressure gradient due to the surface tension. Using the finer mesh, the interface is captured more sharply, and the numerical oscillations are not appeared.

**Merger of two rising bubbles**

We simulated the bubble coalescence when the two bubbles rose in a liquid due to buoyancy force. Test conditions are as follows. Computational domain is $4\times4\times12$ (-2$\leq x\leq$2, -2$\leq y\leq$2, -6$\leq z\leq$6) rectangular box.
Fig. 4: Merger of two rising bubbles: $51\times51\times153$ mesh.

Fig. 5: Volume conservation for two rising bubbles.

Two spherical bubbles with unit radius are initially located at $(x, y, z) = (0.25, 0, -4.5)$ and $(x, y, z) = (-0.25, 0, -2.3)$, respectively. We use $51\times51\times153$ mesh with equal spacing. No-slip boundary conditions are used on the every sides of the box. We set $Re=50$, $We=1.25$, and $Fr=1$. The density ratio, the viscosity ratio, and $\Delta t$ are taken to be 0.001, 0.01816, and 0.002, respectively. The multiplier $c$ in Eq. (35) is take to be 1.05.

As found from Fig. 4, both bubbles become cap-like shape at about $t=1.6$ and then the upper bubble becomes like saucer. Then at about $t=5.6$, they merge together because the vortex flow created by the upper bubble accelerates the rising speed of the lower bubble. After coalescence, the depression is found at the bottom surface of the merged bubble.

Time histories of the volume (or mass) of two bubbles are shown in Fig. 5. We compare the influence of the multiplier $c$ in Eq. (35) for reinitialization procedure on the mass conservation. The volume of bubbles are defined by

$$V(t) = \int \int \int (1 - H) \, dx \, dy \, dz.$$

Solid lines, dotted lines and dashed line with one dot correspond to $c=1.05$, $c=1.03$, and $c=1$, respectively. When $c=1$, the mass is lost gradually and the final mass becomes about 10% smaller than the initial mass. When $c=1.03$, the mass conservation is better than that when $c=1$. In this case, the mass starts to be lost when the distance between the two bubbles becomes short and they merge with each other. The maximum error at that time is about 1%. When $c=1.05$, the mass conservation is very well.

Gas bubble bursting at a free surface

We also simulated the problem of a gas bubble rising to the free surface of a liquid. Figure 6 shows the evolution of a liquid jet resulting from the submerged gas bubble. The time histories of the volume of the gas phase are shown in Fig. 7. The computational domain
Fig. 6: Evolution of a liquid jet resulting from the submerged gas bubble.

The mesh is $60 \times 60 \times 120$ with equal spacing. A spherical bubble with unit radius is released below the free surface at $(x,y,z) = (0, 0, -3.2)$. We used the periodic boundary conditions in both the $x$ and $y$ directions. The no-slip boundary conditions are used in the $z$ direction. We set $Re=474$, $We=1$ and $Fr=0.64$. The density ratio $\lambda$ and the viscosity ratio $\eta$ are taken to be 0.001 and 0.01, respectively. As shown in Fig. 6, the liquid jet starts to break up into a drop at $t=1.2$. Then the second drop is formed at $t=1.64$. As evident in Fig. 7, the mass conservation is good. However, the present mesh is too coarse to resolve the detail of the drops.

CONCLUSIONS

We developed a numerical method for the three-dimensional two-phase flows with unequal spacing using level set method. To analyze the two-phase systems, we improved the treatment of the convection terms and the interpolation method that was used to obtain the intermediate volume flux defined on the cell faces. We also improved the solver for the pressure Poisson equations. To achieve the perfect mass conservation of two fluids systems, we also improved the reinitialization procedure developed by Sussman et al. (1998). It was shown that the present solver worked very well even though the density ratio of two fluids was 1:1000. We succeeded in simulating the merger of two rising bubble and the burst of a submerged gas bubble. The mass conservation of bubbles was good even after the bubble coalescence.

Fig. 7: Volume conservation for the gas bubble bursting at a free surface.

REFERENCES


