High Order Level Contour Reconstruction Method

Seungwon Shin*
Department of Mechanical and System Design Engineering, Hongik University, 72-1, Sangsu-dong, Mapo-gu, Seoul 121-791, Korea
Damir Juric
Laboratoire d'Informatique pour la Mecanique et les Sciences de l'Ingenieur (LIMSI), CNRS-UPR 3251, BP133, 91403 Orsay, France

Complex interfacial physics arising from geometric curvature associated with surface tension as well as phase transformation make it a formidable task to design an accurate, reliable, and yet simple method for direct computation of multiphase flows. Hybrid methods mixing conventional, Volume-of-Fluid, Level Set, Phase Field, and Front Tracking methods have recently become popular in an attempt to overcome the shortcomings of each method alone. We developed the Level Contour Reconstruction Method (LCRM) as part of a hybrid method for treating the complex interface geometry associated with general three-dimensional multiphase flows. The main idea in that work focused on a simple and robust algorithm especially suited for dynamic interfaces in the three-dimensional case by combining characteristics of both Front Tracking and Level Set methods. In this article we describe a modification to the LCRM which introduces a high order interpolation kernel during the course of the interface reconstruction along with a new hybrid surface tension formulation. With this we can essentially eliminate any mass redistribution between regions of differing curvature and reconstruct the interface accurately and smoothly. The improvement with high order reconstruction is also noticeable vis a vis spurious currents which are further decreased by two orders of magnitude over the previous linear reconstruction method. Moreover, there is no disturbance concurrent with reconstruction and the solution fidelity is not influenced by the reconstruction time step. This High Order Level Contour Reconstruction Method retains the simplicity of the original LCRM and avoids complicated interface smoothing procedures.

Key Words : Numerical Simulation, Front Tracking, Surface Tension, Parasitic Currents, Multiphase Flow, Computational Fluid Dynamics

Nomenclature

$\mathbf{u}$: Velocity vector
$U_{\max}$: Magnitude of the maximum velocity
$\mathbf{V}$: Interface velocity vector
P : Pressure
$\boldsymbol{g}$: Gravitational acceleration
$\mathbf{F}$: Local surface tension force vector
$\Delta x, \Delta y, \Delta z$: Grid spacing in each x, y, and z direction
N_p : Number of interface elements
N_x, N_y, N_z : Number of grid in each x, y, and z direction
σ : Surface tension coefficient
μ : Viscosity
ρ : Density
κ_L : Twice the mean interface curvature
I : Indicator function
$\mathbf{n}_f$: Unit normal to the interface
δ_f : Dirac delta function
$La = \sigma\rho D/\mu^2$: Laplace number

* Corresponding Author,
E-mail : sshin@wow.hongik.ac.kr
TEL : +82-2-320-3038; **FAX** : +82-2-322-7003
Department of Mechanical and System Design Engineering, Hongik University, 72-1, Sangsu-dong, Mapo-gu, Seoul 121-791, Korea. (Manuscript **Received** June 5, 2006; **Revised** December 1, 2006)

$Ca = U_{\max}\mu/\sigma$: Capillary number

Subscripts

1 : Fluid 1
2 : Fluid 2

1. Introduction

Direct simulation of multiphase flow has drawn ever increasing attention for numerous important engineering applications from industrial flows to biological fluid interactions. Despite its importance, complex interfacial physics from geometric curvature associated with surface tension as well as phase transformation still make it a formidable task to come up with an accurate, reliable, and yet simple method.

The method using fixed grids to represent the velocity field with additional specific advection schemes in order to preserve the sharpness of interfacial front are popular due to their simplicity. These methods are further divided into the front capturing type : Volume-of-Fluid (VOF) (Brackbill et al., 1992 ; Lafaurie et al., 1994 ; Scardovelli and Zaleski, 1999), Level Set (Osher and Fedkiw, 2001 ; Son, 2001), and Phase Field (Jamet et al., 2001), which represent the interface implicitly and the Front Tracking type (Tryggvason et al., 2001 ; Unverdi and Tryggvason, 1992a, 1992b), which explicitly track the interface. Each of the above methods has some advantage over the others and varying degrees of success in modeling general two-dimensional multiphase problems over wide parameter ranges. On the other hand, it is still difficult to conduct full direct simulations for three-dimensional multiphase flows, even with current computer resources. Thus accuracy under limited memory and computational time is an essential feature in developing a numerical algorithm. Recently, there have been efforts to construct hybrids among the above mentioned methodologies with the intention of facilitating simulations of general three-dimensional problems.

Lagrangian front tracking schemes are well known to maintain sharp interface structures and mass conservation since they preserve material characteristics. On the other hand, front tracking schemes pose many difficulties in reconstruction of the interface especially in three-dimensions for the necessary operations of addition/deletion, breaking, and merging of the computational interface elements. When fluid filaments become too thin, VOF methods tend to form irregular interfaces which are not smooth and even discontinuous in the attempt to maintain local mass conservation. Level Set methods lose/gain a significant amount of mass lowering the accuracy and ultimately compromising the entire solution. Several researchers have worked to improve the accuracy of the geometrical information in the VOF method and numerous attempts have been made to improve mass conservation in level set methods using a variety of reinitialization techniques and higher order ENO/WENO convection schemes.

A number of hybrid methods have appeared in recent years. Sussman and Puckett (2000) proposed a coupled of Level Set/Volume-of-fluid (CLSVOF) method in order to alleviate some of the geometrical problems of the VOF method. It combines the accuracy in mass conservation of VOF and convenient description of topologically complex interfaces of the Level Set function. The resulting scheme still remains Eulerian, not incorporating any of the Front Tracking characteristic and still not accurate in under-resolved regions by blindly applying the VOF local mass constraint.

Enright et al.(2002) combine Lagrangian marker particle and Eulerian Level Set methods to accurately rebuild the zero level set field near the interface. This allows the Level Set method to obtain a subgrid scale accuracy near the interface and better mass conservation properties in under-resolved regions. The lack of connectivity between marker particles makes the implementation much easier than Front Tracking and unsatisfactory description of interface geometry can be overcome by using the Level Set function which maintains nice geometric properties. Marker particles are randomly positioned near the interface and are passively advected by the flow. Their "particle level set" method compares favorably with VOF and Front Tracking in mass conservation and interface resolution, respectively.

Aulisa et al.(2003) present a new hybrid method which combines markers and Volume-of-Fluid algorithms. Two distinct markers of grid intersection and mass conservation have been used to describe the interface. Both markers are advected numerically to update the volume fraction. The conservation markers inside each cell keep the local volume fraction to the reference value while the intersection markers, which locate the interface on the grid lines, eliminate the necessity of remeshing the system. Thus they obtain both smooth motion of the interface by marker methods and good mass conservation as in the standard VOF method.

As an alternate hybrid method, we developed the Level Contour Reconstruction Method (LCRM) (Shin et al., 2005 ; Shin and Juric, 2002 ; Shin, 2002) for very general complicated three-dimensional multiphase flows. The main idea was focused on simplicity and a robust algorithm especially for the three-dimensional case. It combines characteristics of both Front Tracking and Level Set methods since it primarily uses Lagrangian elements (line for 2D and triangles for 3D) to describe the interface and its motion but periodically regenerates this interface from contour field values such as the indicator function or distance function as in the Level set method if available. The Level Contour Reconstruction Method is basically a Front Tracking type method which tracks the implicitly connected individual interface elements. At the same time, we take advantage of the fact that the interface can also be represented by an Eulerian function field (indicator function, $I(\mathbf{x})$, in our case). Reconstruction of the interface at a certain level set of the indicator function enables us to naturally, automatically, and robustly model the merging and pinch off of interfaces as in the Level Set method. It also retains the subgrid accuracy of interfacial motion of Front Tracking while eliminating the burden of bookkeeping of the logistic information of neighboring elements as is necessary in original Front Tracking and which incurs a large memory and calculation cost. The interface tension has also been formulated in a hybrid form in Shin et al. (2005) for more accurate representation of the effect of curvature, and a corresponding reduction in the parasitic currents to a minimal level.

During the reconstruction, the interface is relocated using a linear interpolation of the given indicator function field. We found that reconstruction using this linear interpolation introduces a small disturbance which eventually dies out very quickly after a few time steps. This indicates that the reconstructed interface is continuous but not smooth with linear interpolation. The effect is usually negligible since reconstruction is not performed at every time step. But this slight perturbation may cause instability of the solution for simulations requiring frequent reconstruction of the interface, especially with low resolution. Moreover, slight discrepancies in the interface points can induce undesirable results where the exact location of the interface is extremely important. In this paper, we will present a method to increase the accuracy and smoothness of the Level Contour Reconstruction Method by introducing a high order interpolation kernel during interface reconstruction.

The remainder of this paper is organized as follows. The next section includes a brief description of the LCRM and a description of the new high order interface reconstruction method. We describe the governing Navier-Stokes equations in the subsequent section. We then present test cases to demonstrate the accuracy of the new scheme compared to other existing methods.

2. Numerical Development

2.1 High order level contour reconstruction method

We will start with a brief description of the Level Contour Reconstruction Method. The complete details can be found in Shin et al.(2005), Shin and Juric (2002) and Shin (2002). In the Front Tracking method a stationary Eulerian volumetric mesh is supplemented by a moving Lagrangian interface mesh. The Level Contour Reconstruction Method is a simplified version of the Front Tracking method. The interface is composed of non-stationary, not logically but implicitly connected computational points to form

a two-dimensional surface or one-dimensional line for 3D or 2D problems respectively. The interface elements are periodically discarded and then reconstructed on a certain level contour of the characteristic indicator function.

The Indicator function, I, has essentially the same characteristics as the Heaviside function and can be found by solving the following Poisson equation with a standard FFT package such as FISHPAK on a uniform Cartesian grid which takes negligible time compared to pressure iteration:

$$\nabla^2 I = \nabla \cdot \mathbf{G} \tag{1}$$

where the distribution, $\mathbf{G}$, is geometric information computed directly on the interface and then distributed onto an Eulerian grid.

$$\mathbf{G} = \int_{\Gamma(t)} \mathbf{n}_f \delta_f(\mathbf{x}-\mathbf{x}_f)\, dS \tag{2}$$

and $\mathbf{n}_f$ is the unit normal to the interface. Here, $\mathbf{x}_f = \mathbf{x}(s,t)$ is a parameterization of the interface $\Gamma(t)$, and $\delta_f(\mathbf{x}-\mathbf{x}_f)$ is a three-dimensional Dirac distribution that is non-zero only when $\mathbf{x}=\mathbf{x}_f$. Hereafter, we will denote the indicator function calculated from equation (1) as I_{org}.

The level contour of the indicator function which satisfies the exact location of the original interfacial points before reconstruction is not however, in general a constant value. At a specific time step and given a complete set of tracked interface elements, we can obtain the indicator function value at each element point by interpolating the original grid indicator function values to surface element points. We then redistribute these local indicator function values back to the nearest grid locations. The idea is to use this local indicator function value at the surface to reconstruct the element at that cell location. There may be more than one surface element in one cell. Indicator function value at each interfacial element can be found by standard interpolation technique and we will denote this value as $I_{p,e}$. After having interpolated the indicator function values to the surface for all the elements, observe that each cell is affected by several elements, $I_{p,e}$ with areas, Δs_e. Thus the local contour value to use for reconstruction, I_{local}, at that location is calculated by distributing this value back onto the Eulerian grid

$$I_{local}(i,j,k) = \frac{\sum_e I_{p,e}\Delta s_e}{\sum_e \Delta s_e} \tag{3}$$

Thus we generate a localized contour level field, $I_{local}(i,j,k)$, which will be used to reconstruct the element at that cell location (Shin et al., 2005). Finally we draw a contour line/surface of zero value with linear evaluation function, $E^L(\mathbf{x})$,

$$E^L(\mathbf{x}) = \sum_g [I_{org}(i,j,k) - I_{local}(i,j,k)]\, p(\mathbf{x}-\mathbf{x}_g) \tag{4}$$

Here, $\mathbf{x}$ is the evaluation point, $\mathbf{x}_g$ is the grid cell center, and the integral is performed across a small multiple of the mesh spacing in each direction. $P(\mathbf{x}-\mathbf{x}_g)$ is a tensor product of one-dimensional Linear interpolation kernels, L, given by:

$$\begin{aligned} &P(\mathbf{x}-\mathbf{x}_g) \\ &= L(x-x_g;\Delta x)\, L(y-y_g;\Delta y)\, L(y-y_g;\Delta y) \end{aligned} \tag{5}$$

with grid spacing $\Delta x, \Delta y$, and Δz. The Linear interpolation kernel is defined by:

$$L(x;h) = \begin{cases} 1-\dfrac{|x|}{h}, & 0 \le \dfrac{|x|}{h} \le 1 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where h is the grid spacing in each direction. Hereafter, we will refer to this as the linear reconstruction procedure as compared to the high order reconstruction which will be discussed below.

The linear reconstruction always generates a small perturbation near the reconstructed interface since we used a linear interpolation kernel function, which is continuous but not smooth, to locate the zero level of the evaluation function, even though the surface can be reconstructed accurately with small mass redistribution between different curvature regions by using the localized indicator function value. Furthermore, the values of indicator function interpolated from the $I_{local}(i,j,k)$ field at the original interface points do not exactly match the value interpolated from the given original indicator function, $I_{org}(i,j,k)$. We interpolated the indicator function value from the original interface points and then redistributed this back to the grid, so that the final local indicator function field is a somewhat averaged

value since each cell can be affected by several elements.

This perturbation eventually dies out since the reconstruction is usually not performed every time step during the course of a simulation. Thus the global sense of the error can usually be considered to be minimal. But for some problems which undergo abrupt change of interfacial motion from a surface tension dominant equilibrium state to a dynamic flow regime with large deformation, rupture, and coalescence, frequent reconstruction becomes necessary. The reconstruction time step usually depends on the specific problem at hand and the optimal time step can be difficult to obtain. With excessive reconstruction, the small disturbances from reconstruction can drive the interface to unphysical locations and the solution will depend on the chosen reconstruction frequency. The problem becomes worse for small length scales without sufficient resolution, which turns out to be a frequent scenario for three-dimensional simulations.

We had taken advantage of the fact that we have two separate representations of the interface position: 1) the explicitly tracked interface elements and 2) the certain level contour of the indicator function which has a value of one in one phase and zero in the other. Thus, beginning with a given indicator function field we can deposit a collection of interface elements on the certain contour level or, conversely, beginning with interface elements we can solve the Poisson equation for the indicator function. To locate a contour line/surface in the indicator function field, we need two ingredients: a level contour value at a specific location and the evaluation function to compute the field level. A continuous and smooth contour level can be obtained using B-spline interpolation functions. B-spline interpolation allows smoothing of the, possibly noisy, data (Monaghan, 1985). The Indicator function value at an arbitrary location can be found by:

$$I(\mathbf{x}) = \sum_g I_{org}(i,j,k)\, S(\mathbf{x}-\mathbf{x}_g) \tag{7}$$

Here, $S(\mathbf{x}-\mathbf{x}_g)$ is a tensor product of one-dimensional B-splines, M, given by:

$$\begin{aligned} &S(\mathbf{x}-\mathbf{x}_g) \\ &= M(x-x_g;\Delta x)\, M(y-y_g;\Delta y)\, M(y-y_g;\Delta y) \end{aligned} \tag{8}$$

We used both the cubic B-splines $M_3(x;h)$ and quintic B-splines $M_5(x;h)$ suggested by Torres and Brackbill (2000),

$$M_3(x;h) = \begin{cases} \frac{2}{3} - \left(\frac{|x|}{h}\right)^2 + \frac{1}{2}\left(\frac{|x|}{h}\right)^3, & 0 \le \frac{|x|}{h} \le 1, \\ \frac{1}{6}\left(2 - \frac{|x|}{h}\right)^3, & 1 \le \frac{|x|}{h} \le 2 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

and

$$\begin{aligned} &M_5(x;h) \\ &= \frac{1}{120} \begin{cases} \left(3-\frac{|x|}{h}\right)^5 - 6\left(2-\frac{|x|}{h}\right)^5 + 15\left(1-\frac{|x|}{h}\right)^5, & 0 \le \frac{|x|}{h} \le 1, \\ \left(3-\frac{|x|}{h}\right)^5 - 6\left(2-\frac{|x|}{h}\right)^5, & 1 \le \frac{|x|}{h} \le 2 \\ \left(3-\frac{|x|}{h}\right)^5, & 2 \le \frac{|x|}{h} \le 3 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{10}$$

We can obtain the evaluation function field, $E^*(\mathbf{x})$, which gives a zero contour level at interface reconstruction from:

$$\begin{aligned} &E^*(\mathbf{x}) \\ &= \sum_g [I_{org}(i,j,k) - I_{local}(i,j,k)]\, S(\mathbf{x}-\mathbf{x}_g) \end{aligned} \tag{11}$$

The evaluation field, $E^*(x)$, has continuous and smooth properties along the zero contour level. But if we compute the evaluation function value at the original interface point, the resulting value is still not exactly zero but very close due to the manner of calcuating $I_{local}\ (i,j,k)$.

To get a more precise location for the reconstructed interfacial elements, we correct by adding a trial function field, $\Psi_{i,j,k}$, which will satisfy the zero contour value at the original interface point. The corrected Eulerian evaluation field function is defined by:

$$\begin{aligned} E^H(\mathbf{x}) = &\sum_g [I_{org}(i,j,k) - I_{local}(i,j,k)]\, S(\mathbf{x}-\mathbf{x}_g) \\ &- \sum_g \Psi_{ijk} S(\mathbf{x}-\mathbf{x}_g) \end{aligned} \tag{12}$$

There are several ways of approximating the trial function, $\Psi_{i,j,k}$. We assumed a $\Psi_{i,j,k}$ function described by:

$$\Psi_{ijk}=\sum_{N_p}\delta I_p S(\mathbf{x}_g-\mathbf{x}_p) \tag{13}$$

Here, $\mathbf{x}_p$ are the location of original interface points before reconstruction, δI_p is the increment needed at the original interfacial points, and the intergration has been performed throughout the entire given element, respectively. After rearrangement of equations (12) and (13), the final form of the system of linear equations which satisfies a zero level of the evalution function is:

$$\sum_g\sum_{N_p}\delta I_p S(\mathbf{x}_g-\mathbf{x}_p)\,S(\mathbf{x}-\mathbf{x}_g) =\sum_g[I_{org}(i,j,k)-I_{local}(i,j,k)]\,S(\mathbf{x}-\mathbf{x}_g) \tag{14}$$

The idea here is similar to the Point Set Method of Torres and Brackbill (2000). However, they construct a new indicator function by adding a function interpolated directly from interface points to interface points:

$$I(\mathbf{x})=\sum_{N_p}\delta I_p S(\mathbf{x}-\mathbf{x}_p)+I_{org}(\mathbf{x}) \tag{15}$$

This was used to find the vaule of δI_p which would give a constant indicator function value throughout the interfacial points. This proves useful in two-dimensional simulations but will require a tremendous amount of work in three-dimensions. The computational effort needed to find the first part of the left hand side of equation (15) is N_p for a single grid location. For the entire computational domain, $N_p\times N_x\times N_y\times N_z$ computations are required. This would most likely be impractical to perform for three-dimensional simulations unless additional information for interface point locations is provided.

Equation (12) is a more convenient form which can be treated efficiently for function evalution. We distribute the δI_p increment thoughout the domain to construct the Eulerian field trial function and then interpolate this back to the original interfacial points to force the zero contour of $I_{org}(i,j,k)$-$I_{local}(i,j,k)$. Equation (14) can be rewriten in simple form as below:

$$\Phi[\delta I_{p,m}]=b(x_{p,m}),\ m=1,\cdots,N_p \tag{16}$$

We calculate the $\delta I_{p,m}$ using a Newton iteration scheme:

$$\delta I_{p,m}^{l+1}=\delta I_{p,m}^{l}-[\mathbf{J}]^{-1}\Phi(\delta I_{p,m}^{l}),\quad m=1,\cdots,N_p \tag{17}$$

where l is the iteration index and $\mathbf{J}$ is the Jacobian matrix of partial derivatives of the error with respect to the $\delta I_{p,m}$. These derivatives are difficult to calculate since equation (17) is implicit in regard to $\delta I_{p,m}$ and the subsequent matrix inversion would also be computationally expensive, thus a simple form of the Jacobian is used,

$$\mathbf{J}=a^{-1}\mathbf{I} \tag{18}$$

where $\mathbf{I}$ is the identity matrix and a is a constant which determines the rate of convergence of the iteration. At the optimum value of a, different for different physical parameters, the iteration converges rather quickly to a tolerance of $\varepsilon=10^{-5}$ in 10 to 100 iterations. The computational overhead associated with equation (12) is usually negligble compared to the pressure iteration in cases with an optimum value of a. Optimum values for a, which range roughly between 1 and 10, are generally different for the problem at hand and have to be determined through numerical experiment. The tolerance is calculated by

$$\varepsilon=\max(|\delta I_{p,m}^{l+1}|),\ m=1,\cdots,N_p \tag{19}$$

The high order Level Contour Reconstruction Method bears great resemblence to the Point Set Method of Torres and Brackbill (2000). They used points rather than the line or triangular elements which we use in our Level Contour Reconstruction Method. Because we utilize higher dimensions in the front description, calculation of geometric quantities such as curvature, normal, and tangent associated with the interface becomes much more straightforward to implement.

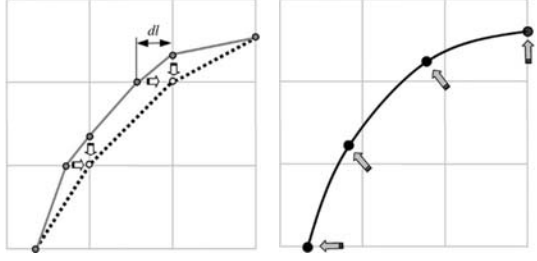


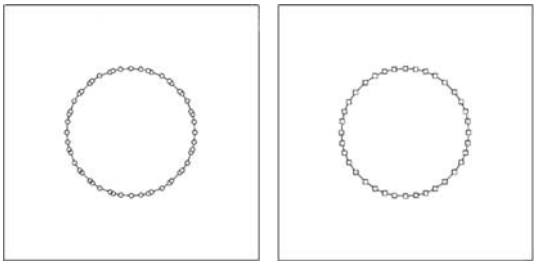
Fig. 1 Interface regularization during reconstruction: (a) Attaching to grid edge. (b) Relocation to correct position

2.2 Regularizing the surface elements

In general, the reconstructed interface contains elements of non-uniform size as seen in Figures 2(a) and 3(a) because the location of reconstruction is confined to the edge of each grid cell. The irregularity of element size does not produce any problem with the high order reconstruction but increases the required memory by generating insignificant size elements without any additional accuracy enhancement. Usually elements of the size of the spatial grid size are recommended. Thus in terms of memory optimization it is preferable to generate elements of roughly uniform size.

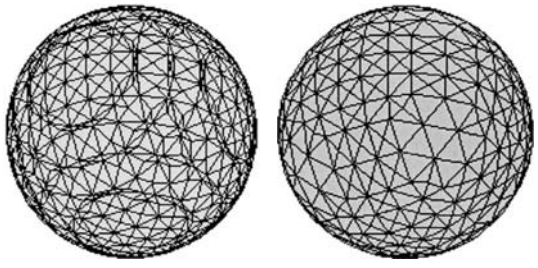
We regularize the interface elements in the following two steps: First we draw an approximate contour line neglecting relatively small elements. As can be seen in Figure 1, we draw a new line, which passes through the edge of the cell when the distance from the reconstructing point to the edge of each cell, dl, is less than a specified criteria. We call this process "attaching to the grid". Because the first step is simply a rough approximation, we can use either equation (4) or (11) to locate the reconstructed interface. The next step is to relocate this approximate contour line/point to the exact interfacial location depicted by equation (12) by movement in the normal direction.

After these two steps, the size of the reconstructed interface elements may not be exactly the same but they are quite uniform as can be seen from Figures 2(b) and 3(b). The range of element sizes is usually 1.0 or 1.5 times the spatial grid size. By regularizing the interface elements, we can reduce the number of elements to 70% of that of the previous reconstruction procedure without sacrificing the accuracy. This is quite important in three-dimensional cases where resources are highly restricted.



(a) (b)

Fig. 2 Two-dimensional interface reconstruction: (a) Using regular reconstruction method (b) After regularizing the interface element during reconstruction



(a) (b)

Fig. 3 Three-dimensional interface reconstruction: (a) Using regular reconstruction method (b) After regularizing the interface element during reconstruction

2.3 Governing Equations

The governing equations for isothermal, incompressible multifluid motion can be expressed in a single field formulation as:

$$\nabla\cdot\mathbf{u}=0 \tag{20}$$

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right) =-\nabla P+\rho\mathbf{g}+\nabla\cdot\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})+\mathbf{F} \tag{21}$$

Here, $\mathbf{u}$ is the velocity, P the pressure, $\boldsymbol{g}$ the gravitational acceleration, and $\mathbf{F}$ is the local surface tension force at the interface which can be described by the hybrid formulation developed in Shin et al.(2005) as:

$$\mathbf{F}=\sigma\kappa_L\nabla I \tag{22}$$

where σ is the surface tension coefficient (assumed constant here) and κ_L is twice the mean interface curvature field calculated on the Eulerian grid.

The expression for the curvature, κ_L, is give by:

$$\kappa_L=\frac{\mathbf{F}_L\cdot\mathbf{G}}{\sigma\mathbf{G}\cdot\mathbf{G}} \tag{23}$$

where

$$\mathbf{F}_L=\int_{\Gamma(t)}\sigma\kappa_f\mathbf{n}_f\delta_f(\mathbf{x}-\mathbf{x}_f)\,\mathrm{d}S \tag{24}$$

Material property fields can be described using the indicator function, $I(\mathbf{x},t)$. For example, the density is calculated by:

$$\rho(\mathbf{x},t)=\rho_1+(\rho_2-\rho_1)I(\mathbf{x},t) \tag{25}$$

where the subscripts 1 and 2 refer to the respective fluids. A similar equation is used to define the viscosity, μ.

The interface is advected in a Lagrangian fashion by integrating

$$\frac{\mathrm{d}\mathbf{x}_f}{\mathrm{d}t}=\mathbf{V} \tag{26}$$

where $\mathbf{V}$ is the interface velocity vector.

The method used to solve for the fluid velocity and pressure is the projection method of Chorin (1968). We use a first order, forward Euler integration in time. For the spatial discretization, we use the well known staggered mesh (MAC method) of Harlow and Welch (1965). The pressure is located at the cell centers while the x, y and z components of velocity are located at the respective faces. All spatial derivatives except the convective term are approximated by standard second-order centered differences. The convective term is discretized using a 2nd order ENO procedure (Shu and Osher, 1989; Sussman et al., 1998). On the staggered grid, quantities needed at cell centers are linearly interpolated from cell faces and vice versa. The detailed solution procedure and discretization of governing equation can be found in (Juric and Tryggvason, 1998; Shin et al., 2005; Shin and Juric, 2002).

3. Results and Discussion

In this section we perform a few tests of the high order Level Contour Reconstruction Method focusing on precise relocation of the interface points, local mass conservation, and spurious currents associated with reconstruction. For discussion below we will refer to Equation (12) as the high order reconstruction method and Equation (4) as the linear reconstruction.

3.1 Mass conservation

As we found in our earlier work (Shin et al., 2005; Shin and Juric, 2002; Shin, 2002), the Level Contour Reconstruction using a constant contour (indicator function) value leads to unphysical mass distribution from high curvature to low curvature regions, especially without sufficient grid resolution. By increasing resolution or using a localized indicator function value (Shin et al., 2005), we could minimize this error to an acceptable range.

We reconstructed a circle with radius of 0.25 in a 1×1 box domain using 25×25 resolution in each x and y direction. Figure 4 shows the relative % error of the reconstructed radius, $(R_{recon}-R_{org})/R_{org}\times100$, for both linear reconstruction and high order reconstruction. While the linear reconstruction gives a 0.6% perturbation after reconstruction, the high order reconstruction is far better than linear reconstruction. The error is almost negligible for the quintic B-spline, M_5. By using high order reconstruction, we can relocate the interface points very accurately and smoothly after reconstruction.

We repeated the simulation of Shin et al.(2005), placing two distinctive circular surfaces together in a 1×1 box to evaluate both local volume conservation and accurate interface relocation. The larger, low curvature surface has a radius of 0.25

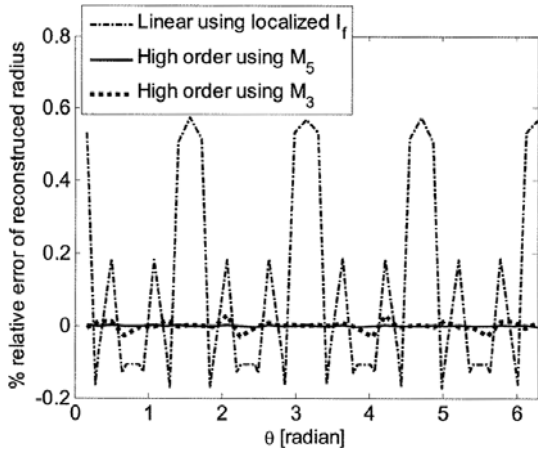


Fig. 4 % relative error of reconstructed radius, $(R_{recon}-R_{org})/R_{org}\times100$, using both linear and high order reconstruction. A circle with radius of 0.25 in 1×1 box domain using 25×25 resolution. The error is almost negligible for the quintic B-spline, M_5

while the smaller, high curvature surface has a radius of 0.05. We perform one reconstruction step and then check how much mass is redistributed between the two different sized circles as a function of grid resolution. We also investigate the radius of both the large and small circle after reconstruction. Both linear reconstruction using a constant or localized indicator function value and high order reconstruction are considered. With a coarse 25×25 mesh the diameter of the small circle is spanned by only about two grid cells and the interface consists of only 6 elements.

To quantify the error, convergence results for both circles are plotted in Figure 5. We plot the volume change of the small circle as compared to the total volume. The amount of area change of the small circle using the high order reconstruction is about one order of magnitude smaller than linear reconstruction using the local indicator function value, which is already one order of magnitude smaller than using a globally constant value. It is interesting to note that the amount of area loss of the circle with small radius is exactly the same as the area gain of the larger circle in the case of linear reconstruction. So the mass is redistributed from the high curvature region to the low curvature region. But for the high order reconstruction, the area change between the larger and smaller circle is not the same. The larger circle is relatively well resolved so most of the error is confined to the smaller circle for high order reconstruction. There is virtually no mass redistribution associated with high order reconstruction which means the error issues mainly from the number of grid points used to resolve the interface.

We can more clearly see the accuracy of the high order reconstruction in Figure 6. This is the plot of maximum relative % error of the reconstructed interface radius of the smaller circle after both linear and high order reconstruction. As can be seen in the figure, the position of the interface is not improved much by linear reconstruction using a localized indicator function value even though the local mass conservation has been greatly improved. The accuracy of the relocated interface has been dramatically improved by using high order reconstruction. With increasing resolution, both bicubic and quintic B-spline reconstruction become comparable to each other.

We need to reconstruct the surface several times during a typical simulation (but not every time step) since interface elements usually deform irregularly as they follow the flow field. Figure 7

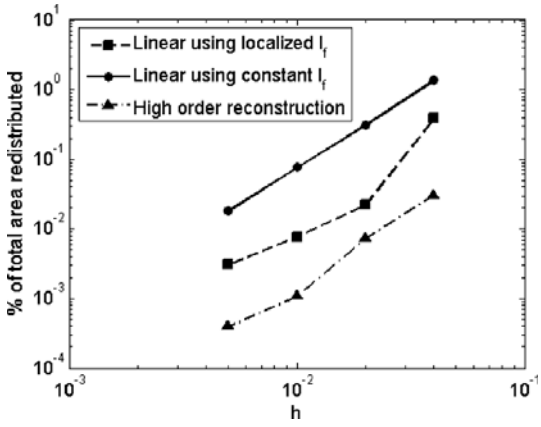


Fig. 5 Mass redistribution between two different size surfaces vs grid resolution. Two distinct circular surfaces, radii of 0.25 and 0.05, are placed in a 1×1 box. We perform one reconstruction step and then check how much mass is redistributed between the two different size circles as a function of grid resolution

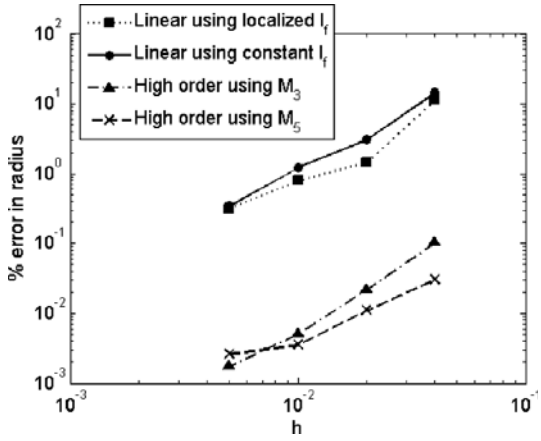
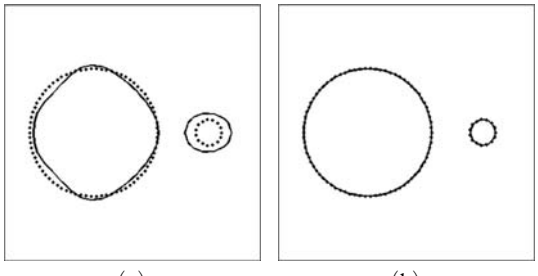


Fig. 6 % relative error of reconstructed radius of smaller circle, $(R_{recon}-R_{org})/R_{org}\times100$, for different resolution using both linear and high order reconstruction

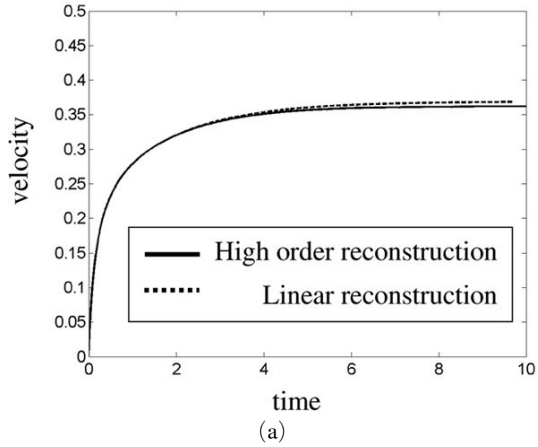
shows the effect of repeated reconstructions on the mass redistribution. The simulation geometry is exactly the same as above and we use the lowest resolution, 25×25 grid. We plot both the initial interface and the interface after 100 consecutive reconstructions. The initial and final reconstructed interfaces are nearly indistinguishable when using the high order reconstruction method (Figure 7 (b)) but with linear reconstruction using the local indicator function value, the shape of the circle has been distorted considerably (Figure 7 (a)). As we discussed before, the resolution is extremely low for this test and such a result can be expected because the error in mass redistribution has not been improved much at this lowest resolution for the linear reconstruction using localized I_f as compared to higher resolution. The high order reconstruction shows more uniform improvement over a wide span of resolution. This is a worst possible case scenario since, in actual computations, we do not reconstruct the interface at every time step but about every 100 to 1000 time steps depending on the particular problem.

We calculated the rise of a single bubble to validate the accuracy of current method. We used a domain of size 8R×8R×8R with density ratio of 1000, viscosity ratio of 100, Reynolds number of 1.34, and Weber number of 14.3 as used by Ryskin and Leal (1984). They obtained terminal bubble rise velocity of 0.374 using body fitted grid structure. In our current simulation, a relatively coarse grid, 34 grids in each direction and a time step of 10^{-3} have been used to capture interface motion. As can be seen from Figure 8, the rise velocity of the bubble reaches a steady state value of 0.365 and within 5% error compared with the results calculated by Ryskin and Leal (1984). Total mass loss during the simulation was less than 0.2%. The rising velocity by linear reconstruction shows comparable result compared to high order reconstruction. But we need to point out that high order method doesn't require volume matching procedure after interface reconstruction which is the case for linear method. Moreover, high order reconstruction generates more stable result by reconstructing relatively smooth interface. The simulation breaks down near $t=$ 10 as can be seen from Figure 8(a) for linear

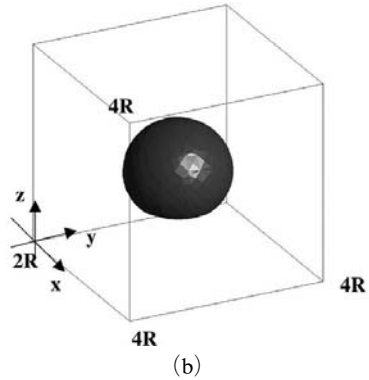


(a) (b)

Fig. 7 Mass redistribution between two different size surfaces with 100 consecutive reconstructions. Two distinct circular surfaces, radii of 0.25 and 0.05, are placed in a 1×1 box with 25×25 grid resolution. (a) Linear reconstruction using local indicator function value (b) High order reconstruction using quintic B-spline



(a)



(b)

Fig. 8 Rise of single bubble (a) Bubble rise velocity (b) Steady state interface shape for single bubble rise with high order reconstruction

reconstruction.

We conclude that with the high order reconstruction method, we can maintain the accuracy and smoothness of the interface even with the lowest possible resolution which can often be the case for three-dimensional simulations in practice.

3.2 Parasitic currents

Spurious or parasitic currents become an important problem in two-phase modeling since they impose limitations in the application of interface methods to two-phase flows. Parasitic currents plague all interface methods and numerous attempts at remediation have been implemented in order to reduce or eliminate these currents. In using the hybrid formulation of the surface tension force (Shin et al., 2005), we can suppress spurious currents to a minimal level compared to the regular Front Tracking formulation. We will check the effect of high order reconstruction on parasitic currents and compare with other endeavors.

As we discussed in Shin et al.(2005), if the initial interface is a circle or sphere with uniformly distributed elements along the interface, we can maintain the parasitic currents at zero machine for any value of Laplace, La, number even up to infinity thanks to the exact discrete balance of the pressure gradient with the surface tension force. This being given, all of the subsequent tests presented here are for interfaces with non-uniformly distributed elements, which would be the case after interface reconstruction, or for an interface initially out of equilibrium progressing to equilibrium, which requires repetitive reconstruction, as in the decaying oscillations of a droplet.

In Figure 9, we repeated the calculation in Figure 8 of Shin et al.(2005). We placed a 2D circular drop with a radius of 0.25 in a 1×1 domain resolved by a 50×50 grid with all other properties set to unity and a time step of 10^{-2} has been used. The Laplace number, $La=\sigma\rho D/\mu^2$, (related to the Ohnesorge number by $La=1/Oh^2$) is set to 250 and $U_{\max}$, the magnitude of the maximum velocity, has been obtained for both linear and high order reconstruction. Since it is not possible to generate a strictly uniform distribution of surface element sizes with our reconstruction technique at this point, we now focus on the effect of the surface reconstruction, which will act as a perturbation of the interface. Reconstruction has been performed at fixed intervals of every $\Delta t=100$ and we can observe in Figure 9 that spikes occur in accordance with reconstruction but die out quickly and the parasitic currents tend to further decrease with linear reconstruction using localized indicator function values. But using high order reconstruction, there is virtually no perturbation during reconstruction and we can maintain orders of magnitude smaller parasitic currents throughout the simulation. This behavior is quite important since it can be difficult to find an optimal reconstruction time and multiple reconstructions during a short span of time may need to be performed. We still have some amount of spurious current even though it is at an essentially negligible level. This is due to the non-uniformity of interface element sizes after reconstruction.

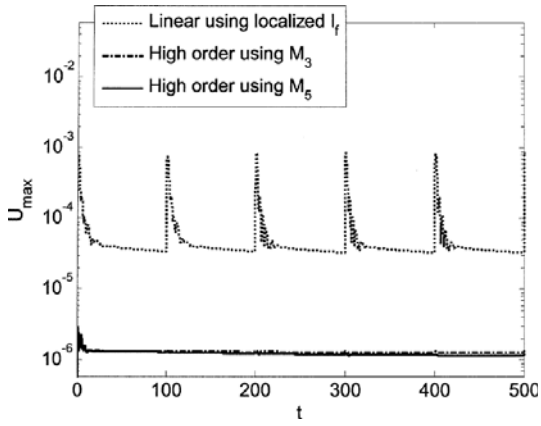


Fig. 9 Parasitic currents in a 2D equilibrium static drop calculation for a drop of radius $R=0.25$ in a 1×1 domain resolved by a 25×25 grid with a Laplace number, $La=250$. Interface reconstruction every $\Delta t=100$

In Figure 10, we repeat the 2D droplet oscillation experiment of Torres and Brackbill (2000). The figure shows a plot of kinetic energy, $1/2\int\rho\mathbf{u}\cdot\mathbf{u}dV$, versus time for simulations on a doubly periodic $[0, 0.01]^2$ domain resolved by a

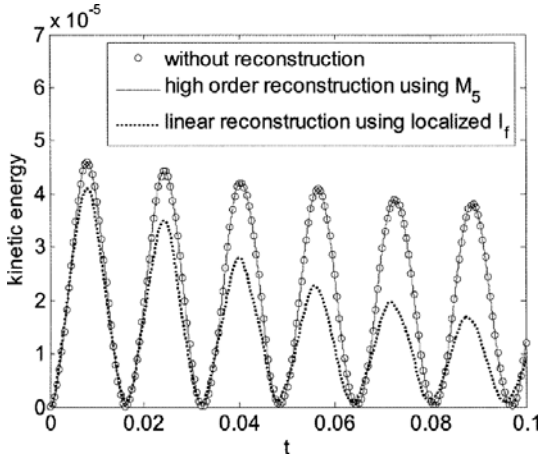


Fig. 10 Kinetic energy vs time for a 2D oscillating drop. The ellipse, $x^2/0.003^2+y^2/0.002^2=1$, is placed in a doubly periodic $[0, 0.01]^2$ domain resolved by a 64×64 grid. The density and viscosity inside the ellipse is 1000.0 and 0.001, respectively. The density and viscosity outside the ellipse is 1.0 and 10^{-4}, respectively, and $\sigma=0.1$. The number of interface elements is about 400 for the calculation without reconstruction and about 125 with reconstruction. We reconstruct the interface every 10^{th} time step for linear reconstruction using localized indicator function values and every time step for high order reconstruction using the quintic B-spline, M_5

64×64 grid. The interface is initially the ellipse $x^2/0.003^2+y^2/0.002^2=1$. The density and viscosity inside the ellipse is 1000.0 and 0.001, respectively. The density and viscosity outside the ellipse is 1.0 and 10^{-4}, respectively, and $\sigma=0.1$. Time step of 10^{-5} has been chosen for the simulation. The number of interface elements is about 400 for the calculation without reconstruction and about 125 with reconstruction. We reconstruct the interface every 10^{th} time step for linear reconstruction using localized indicator function values and every time step for high order reconstruction using the quintic B-spline, M_5. This is just about a worst possible case since we would not reconstruct the surface this often in an actual computation. Although the frequency of oscillation remains almost the same, the reconstruction using the linear kernel tends to dampen the oscillations as can be seen in Figure 10. But using high order reconstruction, there is virtually no difference to the case without reconstruction which is comparable to that in Torres and Brackbill (2000) who used the curl projection formulation to reduce parasitic currents. We conclude that the high order reconstruction does not introduce any perturbation and the result is independent of the frequency of reconstruction.

Popinet and Zaleski (1999) suppressed the parasitic currents in their Front Tracking method by improving the pressure gradient calculation in grid cells cut by the interface. They found that the Laplace number, La, is correlated to a dimensionless measure of the parasitic current, the capillary number, $Ca=U_{\max}\mu/\sigma$.

We performed comparisons with the calculations shown in Table I of Popinet and Zaleski (1999). A circle with diameter of 0.4 is located in 1×1 box resolved by a 32×32 grid. Periodic boundary conditions in the x direction and reflecting conditions in y direction have been used. The density and viscosity ratio is set to unity and the maximum amplitude of the parasitic currents was measured after 250 characteristic time scales ($t=t_{phys}\sigma/\mu D$) with time step of 10^{-5}. They also tested the convergence of the method by increasing the grid resolution for the case of $La=12{,}000$.

We reconstructed the interface 50 times during the entire simulation with high order reconstruction using the M_5 kernel. The pressure gradient correction in Popinet and Zaleski (1999) indicates a consistent level of $Ca\sim O(10^{-6})$ with increasing La from 1.2 to 12,000. As can be seen in Figure 11(a), our high order reconstruction gives a slightly better result but almost the same order of Ca of $O(10^{-6})$. High order reconstruction also shows quadratic convergence as we increase grid resolution (Figure 11(b)). The simulation is carried out for a relatively long time (2500 characteristic time scales) to obtain converged values. As shown in Figure 11(b), the order of Ca number is comparable to that of Popinet and Zaleski (1999) except for low resolution. For the coarsest mesh, 16×16 mesh has been used and thus only 6 grid cells in each

direction are used for representing the interface. With this extremely low resolution, high order reconstruction loses quadratic convergence characteristic since it is influenced greatly by both Eulerian grid resolution and Lagrangian interface spacing but still shows linear convergence.

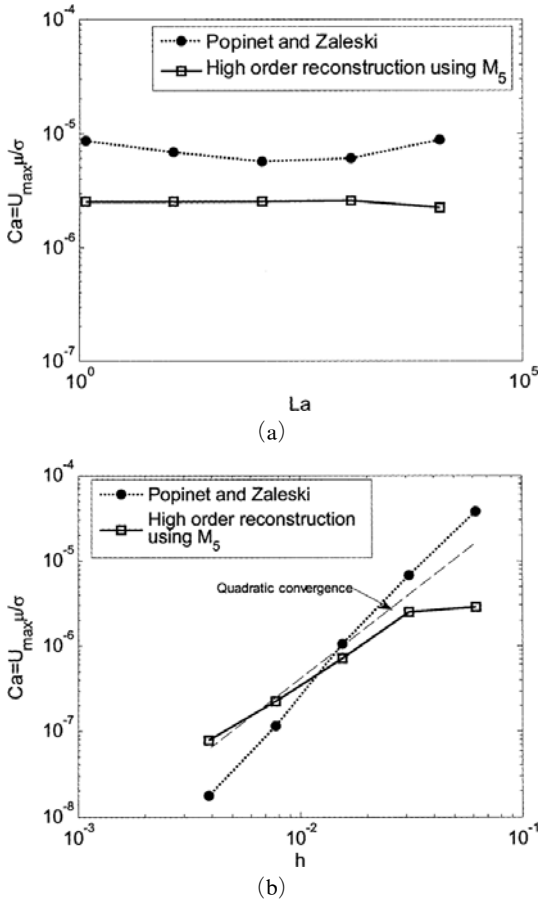


Fig. 11 Parasitic currents in a 2D equilibrium static drop compared with results from Popinet and Zaleski (1999). A circle with diameter of 0.4 is located in 1×1 box resolved by a 32×32 grid. Periodic boundary conditions in the x direction and reflecting conditions in y direction have been used. The density and viscosity ratio is set to unity and the maximum amplitude of the parasitic currents was measured after 250 characteristic time scales ($t=t_{phys}\sigma/\mu D$). (a) Amplitude of the parasitic currents for various Laplace number (b) Effect of spatial resolution on parasitic currents

Renardy and Renardy (2002) proposed an algorithm for improving the surface tension computation and thus reducing the parasitic currents in VOF methods. They calculated the interface curvature from an optimal fit for a quadratic approximation to the color function over groups of cells near the interface. Using this, parasitic currents are reduced by two orders of magnitude compared to conventional VOF-CSF (Brackbill et al., 1992) or VOF-CSS (Lafaurie et al., 1994; Scardovelli and Zaleski, 1999) methods. We repeated the simulation in table I of Renardy and Renardy (2002). We placed a spherical drop centered at (0.5, 0.5, 0.5), with radius of 0.125 and surface tension of 0.357 in a 1×1×1 box. The boundary condition is no slip at the top and bottom walls, and periodic in the x and y directions. Both fluids have equal density of 4 and viscosity of 1. The initial velocity field is zero and a time step of 10^{-5} has been used for the simulation. The exact solution is zero velocity for all time. Figure 12 shows the L_∞, L_1, and L_2 norms of the velocity field for the spurious currents. Each norm is slightly higher because the interface after reconstruction is usually more irregular in three-dimensions than two-dimensions, but the convergence rate is almost the same as the results from Renardy and Renary (2002). Because of the same reason as Figure 10(b), the results slightly deviate for lowest grid resolution.

Shirani et al.(2005) developed a method referred to as pressure calculation based on the interface location (PCIL) for VOF methods. It calculates the pressure force at each interfacial cell face using the exact pressure due to the portion of the cell face occupied by each fluid. They devised a special form of the pressure at the cell face for different orientations of an interface. By doing this, they can reduce the spurious current by up to three orders of magnitude compared to conventional methods. We compared our method to Shirani et al.(2005). The drop which has a radius of 0.25 is placed in a 1×1 box resolved by a 66×66 mesh and a time step of 10^{-5} has been chosen for the simulation. The material properties have been taken for a water droplet in air ($\rho_1/\rho_2=830.545$, $\mu_1/\mu_2=54.945$). The Ohnesorge

number based on the water properties is 1.17×10^{-4}. As we can see in Figure 13, the maximum value of the spurious current is almost three orders of magnitude lower than that of Shirani et al.(2005).

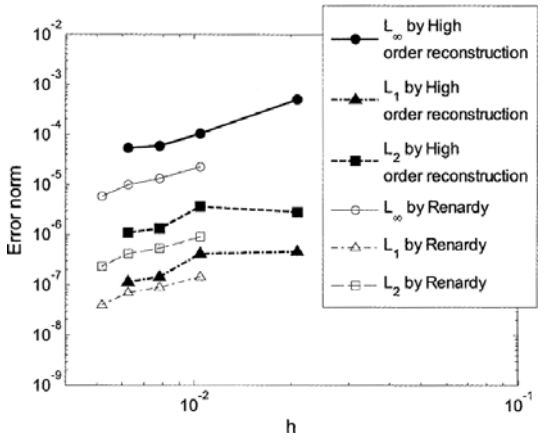


Fig. 12 Velocity norms of spurious current vs grid resolution compared to results from Renardy and Renardy (2002). Spherical surface drop centered at (0.5, 0.5, 0.5), with radius of 0.125 and surface tension of 0.357 in $1\times 1\times 1$ box. The boundary condition is no slip at the top and bottom walls, and periodic in x and y directions. Both fluids have equal density of 4 and viscosity of 1

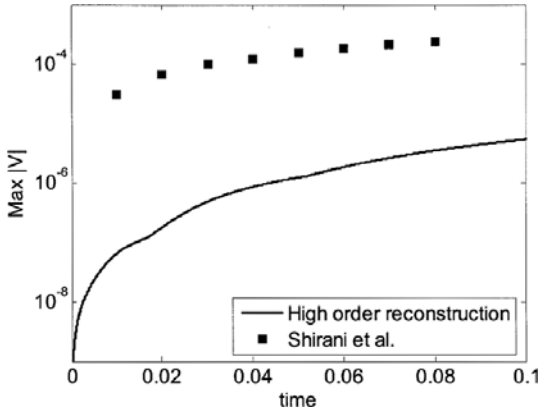


Fig. 13 Maximum spurious velocity compared to results from Shirani et al.(2005). The drop which has a radius of 0.25 is placed in 1×1 box. A 66×66 mesh and a time step of 10^{-5} has been chosen for the simulation. The material properties have been taken for a water droplet in air ($\rho_1/\rho_2=830.545$, $\mu_1/\mu_2=54.945$)

We conclude that parasitic currents, although not negligible, are small in the standard front tracking method (Tryggvason et al., 2001; Unverdi and Tryggvason, 1992a, 1992b), generally of order $O(10^{-4})$. By using the hybrid formulation (Shin et al., 2005), we reduce this to about one order smaller, $O(10^{-5})$ and render the scheme stable to the perturbations generated by the reconstruction. We can further decrease this value to $O(10^{-7})$ by using the high order reconstruction method without introducing any disturbance during reconstruction.

4. Conclusions

Even though the linear reconstruction method using a localized I_f value has good overall performance with local mass conservation, it has a relatively large error in the exact interface location after reconstruction. This usually leads to instability of the solution where repetitive reconstruction is necessary. By using high order reconstruction, we can eliminate the mass redistribution between two distinct interfaces or widely differing curvature regions and relocate the interface accurately and smoothly. The improvement using high order reconstruction is evident in tests of the magnitude of spurious currents. Usually reconstruction introduces a disturbance which dies out quickly when using the hybrid surface tension formulation (Shin et al., 2005). But since there is virtually no perturbation caused by the high order reconstruction, the parasitic currents are almost two orders of magnitude smaller than with the linear reconstruction method. Using the high order reconstruction method, we can achieve an accurate surface shape even with exceptionally low grid resolution without worrying about the reconstruction time step. This can be very important in 3D simulations where the use of sufficient grid resolution is restricted due to the available computational resources.

We point out that our LCRM does not include any optimal fit or approximation of the interface surface. The procedure is extremely simple as discussed previously and the results are comparable

to other methods which use complicated procedures for smoothing the interface.

References

Aulisa, E., Manservisi, S. and Scardovelli, R., 2003, "A Mixed Markers and Volume-of-fluid Method for the Reconstruction and Advection of Interfaces in Two-phase and Free-boundary Flows," *J. Comput. Phys*., Vol. 188, pp. 611～639.

Brackbill, J. U., Kothe, D. B. and Zemach, C., 2002, "A Continuum Method for Modeling Surface Tension," *J. Comput. Phys*., Vol. 100, pp. 335～354.

Chorin, J., 1968, "Numerical Solution of the Navier-Stokes Equations," *Math. Comput*, Vol. 22, pp. 745～762.

Enright, D., Fedkiw, R., Ferziger, J. and Mitchell, I., 2002, "A Hybrid Particle Level set Method for Improved Interface Capturing," *J. Comput. Phys*., Vol. 183, pp. 83～116.

Harlow, F. H. and Welch, J. E., 1965, "Numerical Calculation of Time Dependent Viscous Incompressible Flow of Fluid with Free Surface," *Phys. Fluids*, Vol. 8, pp. 2182～2189.

Jamet, D., Lebaigue, O., Coutris, N., and Delhaye, J. M., 2001, "The Second Gradient Method for the Direct Numerical Simulation of Liquid-Vapor Flows with Phase-change," *J. Comput. Phys*., Vol. 169, pp. 624～651.

Juric, D. and Tryggvason, G., 1998, "Computations of Boiling Flows," *Int. J. Multiphase Flow*, Vol. 24, pp. 387～410.

Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S. and Zanetti, G., 1994, "Modeling Merging and Fragmentation in Multiphase Flows with SURFER," *J. Comput. Phys*., Vol. 113, pp. 134～147.

Monaghan, J., 1985, "Particle Methods for Hydrodynamics," *Comput. Phys. Rep*., Vol. 3, pp. 71～124.

Osher, S. and Fedkiw, R. P., 2001, "Level Set Methods : An Overview and Some Recent Results," *J. Comput. Phys*., Vol. 169, pp. 463～502.

Popinet, S. and Zaleski, S., 1999, "A Front-Tracking Algorithm for Accurate Representation of Surface Tension," *Int. J. Num. Meth. Fluids*, Vol. 30, pp. 493～500.

Renardy, Y. and Renardy, M., 2002, "PROST : A Parabolic Reconstruction of Surface Tension for the Volume-of-fluid Method," *J. Comput. Phys*., Vol. 183, pp. 400～421.

Ryskin, G. and Leal, L. G., 1984, "Numerical Solution of Free-Boundary Problems in Fluid Mechanics. Part 1. The Finite-Difference Technique," *J. Fluid Mech*., Vol. 148, pp. 1～17.

Ryskin, G. and Leal, L. G., 1984, "Numerical Solution of Free-Boundary Problems in Fluid mechanics. Part 2. Buoyancy-Driven Motion of a Gas Bubble Through a Quiescent Liquid," *J. Fluid Mech*., Vol. 148, pp. 19～35.

Scardovelli, R. and Zaleski, S., 1999, "Direct Numerical Simulation of Free-Surface and Interfacial Flow," *Ann. Rev. Fluid Mech*., Vol. 31, pp. 567～603.

Shin, S. and Juric, D., 2002, "Modeling Three-Dimensional Multiphase Flow Using a Level Contour Reconstruction Method for Front Tracking Without Connectivity," *J. Comput. Phys*., Vol. 180, pp. 427～470.

Shin, S., 2002, "A Level Contour Reconstruction Method for Three-Dimensional Multiphase Flows and its Application," *Ph.D. Thesis*, Georgia Institute of Technology.

Shin, S., Abdel-Khalik, S. I., Daru, V. and Juric, D., 2005, "Accurate Representaion of Surface Tension Using the Level Contour Reconstruction Method," *J. Comput. Phys*., Vol. 203, pp. 493～516.

Shirani, E., Ashgriz, N. and Mostaghimi, J., 2005, "Interface Pressure Caluclation Based on Conservation of Momentum for Front Capturing Methods," *J. Comput. Phys*., Vol. 203, pp. 154～175.

Shu, C. W. and Osher, S., 1989, "Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes II," *J. Comput. Phys*., Vol. 83, pp. 32～78.

Son, G., 2001, "Numerical Study on a Sliding Bubble During Nucleate Boiling," *KSME International Journal*, Vol. 15, pp. 931～940.

Sussman, M. and Puckett, E. G., 2000, "A Coupled Level Set and Volume-of-fluid Method for Computing 3D and Axisymmetric Incom-

pressible Two-Phase Flows," *J. Comput. Phys*., Vol. 162, pp. 301～337.

Sussman, M., Fatemi, E., Smereka, P. and Osher, S., 1998, "An Improved Level Set Method for Incompressible Two-Phase Flows," *Computers and Fluids*, Vol. 27, pp. 663～680.

Torres, D. J. and Brackbill, J. U., 2000, "The Point-Set Method : Front-Tracking Without Connectivity," J. *Comput. Phys*., Vol. 165, pp. 620～644.

Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S. and Jan, Y. -J., 2001, "A Front Tracking method for the Computations of Multiphase Flow," *J. Comput. Phys*., Vol. 169, pp. 708～759.

Unverdi, S. O. and Tryggvason, G., 1992a, "A Front-Tracking Method for Viscous, Incompressible, Multi-Fluid Flows," *J. Comput. Phys*., Vol. 100, pp. 25～37.

Unverdi, S. O. and Tryggvason, G., 1992b, "Computations of Multi-Fluid Flows," *Physica D*., Vol. 60, pp. 70～83.