46th AIAA Aerospace Sciences Meeting and Exhibit
7 - 10 January 2008, Reno, Nevada

AIAA 2008-530

Comparison of Particle Level Set and CLSVOF Methods for Interfacial Flows

Zhaoyuan Wang[1], Jianming Yang[2] and Frederick Stern[3]
IIHR-Hydroscience & Engineering, University of Iowa, Iowa City, IA 52242-1585, USA

The level set method is susceptible to the numerical dissipation, which usually results in serious mass loss in the under-resolved regions at the interface. In this study, a hybrid particle level set method and a coupled level set and volume-of-fluid method, have been implemented with the aim at improving the mass conservation property of the level set method. A number of standard tests are conducted to evaluate their performances. The two methods are applied to simulate a water drop impact onto a liquid pool, and wave breakings over a submerged bump which is of particular relevance to ship hydrodynamics.

I. Introduction

Interfacial and free surface flows are common phenomena in ship hydrodynamics which include complex air/water interactions, such as breaking waves, formation of spray and entrainment of air around ships. The highly dynamic interface between water and air needs to be located for the numerical simulation of such flows. The level set (LS) method, pioneered by Osher and Sethian,[1] has been widely used for interface modeling in free surface and interfacial flows.[2, 3]

In the LS method, a signed distance function is employed to capture the interface implicitly. It has been shown significant success to handle complex flow problems involving severe topological changes, such as breakup and coalescence, with its ease and simplicity of implementation. The major problem with the LS method is its susceptibility to numerical dissipation which results in a poor mass conservation property. When the LS functions are advected, the sharp edges are usually smoothed out and hence mass loss occurs, which is even worse when a coarse grid is used. This issue has been frequently addressed and many attempts have been made to remedy it,[4-12] among which some combined strategies are adopted. Examples are the coupled level set and volume-of-fluid method (CLSVOF)[4, 5] and the hybrid particle level set (PLS) method.[6, 7] In the combined method, the LS function is still used to provide the interface information but is corrected by other functions in order to keep mass conservation. The advantage of the combined method is that major modifications to the original LS based code are not needed. This will not only avoid additional coding work but also keep the desirable properties offered by the LS method.

In the CLSVOF method, volume conservation is imposed directly by solving the volume fraction transport equation. The interface is reconstructed based on the volume-of-fluid (VOF) function and the interface normal computed from the LS function. The LS function is re-distanced based on the reconstructed interface. With the correction by the VOF function, the mass can be very accurately conserved. The implementation of the CLSVOF method is not easy, because both the interface reconstruction and LS re-distance involve a very complicated geometric procedure. Moreover, the available CLSVOF algorithms are generally based on the orthogonal Cartesian grid. The implementation on curvilinear or unstructured grids will be more challenging.[12] The PLS method,[6, 7] as a combined Eulerian-Lagrangian method, utilizes massless marker particles to correct the LS function in the under-resolved areas. This PLS method has been shown remarkable improvement in mass conservation through a number of benchmark tests for both two- and three-dimensional problems with more fine features of the interface well captured. The implementation of the PLS method is relatively easy as compared to the CLSVOF method. Due to the desirable visual effects in simulating realistic fluids, the PLS method has been used in computer graphics and physical based animations.[13] Its applications in free surface flows can be found in Ref. 14.

In this study, the CLSVOF method and PLS method are implemented with the aim at improving the mass conservation property of the LS method for the numerical simulation of interfacial flows in ship hydrodynamics. For the PLS method, a generalized particle reseeding algorithm is developed instead of using the original strongly

[1] Post-doctoral Associate, AIAA Member
[2] Assistant Research Scientist, AIAA Member
[3] Professor of Mechanical Engineering

problem-dependent method.[6] The piecewise linear interface construction (PLIC) scheme for the VOF method presented in Ref. 15 is employed for the interface reconstruction, and an efficient LS re-distance algorithm is developed for the CLSVOF method. The performance of the CLSVOF and PLS methods is evaluated through a series of benchmark tests with a prescribed velocity field. Then a water drop impact onto a liquid pool and wave breakings over a submerged bump are simulated in conjunction with CFDShip-Iowa version 6, a Cartesian grid solver for two-phase incompressible flows recently developed at IIHR.[16,17] In this solver, the interface is represented by the LS method. A ghost fluid methodology is adopted to handle the jump conditions across the interface, where the density and surface tension effect are treated in a sharp way while the viscosity is smeared by a smoothed Heaviside function.

The computational methods for two-phase interfacial flows are presented in Sec. II. Sec. III is devoted to the description of both the PLS and CLSVOF methods. The numerical results computed from the two methods are compared and analyzed in Sec. IV and Sec. V, and conclusions are given in the final section (Sec. VI).

II. Computational Methods

A. Navier-Stokes Equations

The incompressible viscous flows of two immiscible fluids, e.g., air and water, are governed by the Navier-Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{\rho} \nabla \cdot \left(-p\mathbf{I} + \mathbf{T} \right) + \mathbf{g} , \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 , \tag{2}$$

where t is time, $\mathbf{u}$ is the velocity vector, p is pressure, $\mathbf{I}$ is the unit vector, ρ is density, $\mathbf{g}$ represents the gravity acceleration, and $\mathbf{T}$ is the viscous stress tensor defined as

$$\mathbf{T} = 2\mu \mathbf{S} = 2\mu \left[\frac{1}{2} \left(\nabla \mathbf{u} + \left(\nabla \mathbf{u} \right)^T \right) \right] = \mu \left(\nabla \mathbf{u} + \left(\nabla \mathbf{u} \right)^T \right), \tag{3}$$

with μ the dynamic viscosity and $\mathbf{S}$ the strain rate.

Density and viscosity are discontinuous across the interface, which is a function of time and space. They will be defined by using LS function later.

B. Interface Jump Conditions

The velocity across the interface Γ is continuous, as the fluids are viscous and no phase change is considered here:

$$\left[\mathbf{u} \right] = 0 , \tag{4}$$

and the jump condition for stress is

$$\left[\mathbf{n} \cdot \left(-p\mathbf{I} + \mu \left(\nabla \mathbf{u} + \left(\nabla \mathbf{u} \right)^T \right) \right) \cdot \mathbf{n} \right] = \sigma \kappa , \tag{5}$$

where [ ] indicates the jump at the interface, i.e., $f_L^I - f_G^I$ for a variable f with superscript I denoting interface, $\mathbf{n}$ is the unit vector normal to the interface, σ is the surface tension coefficient, and κ is the local curvature of the interface.

C. Interface Representation

The interface Γ is defined as the zero LS of a signed distance function, ϕ, or the LS function, the position of the interface can be tracked by solving the LS evolution equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 . \tag{6}$$

To keep ϕ as a signed distance function in the course of evolution, the LS function is re-initialized by iterating the following equation[18]

$$\frac{\partial \phi}{\partial \tau} + S\left(\phi_o \right) \left(\left| \nabla \phi \right| - 1 \right) = 0 , \tag{7}$$

where τ is the pseudo time and $S(\phi_0)$ is the numerically smeared-out sign function

$$S(\phi_o) = \frac{\phi_o}{\sqrt{\phi_o^2 + h^2}}, \tag{8}$$

with ϕ_o the initial values of ϕ and h a small distance, usually the grid cell size, to smear out the sign function.

The geometric properties, i.e., the normal vector and curvature, can be readily obtained from the LS function:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \tag{9}$$

and

$$\kappa = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right). \tag{10}$$

D. Fluid Properties

With the LS function defined, the fluid properties, such as density and viscosity, are given by

$$\begin{aligned} \rho &= \rho_G + (\rho_L - \rho_G) H(\phi) \\ \mu &= \mu_G + (\mu_L - \mu_G) H_\varepsilon(\phi) \end{aligned}, \tag{11}$$

where the subscripts G and L represent gas and liquid phase, respectively, and the Heaviside function is defined as

$$H(\phi) = \begin{cases} 1 & \text{if} \quad \phi \ge 0 \\ 0 & \text{if} \quad \phi < 0 \end{cases}. \tag{12}$$

In this paper, the viscosity is smoothed over a transition region across the interface as

$$\mu = \mu_G + (\mu_L - \mu_G) H_\varepsilon(\phi), \tag{13}$$

using the smoothed Heaviside function[16]

$$H_\varepsilon(\phi) = \begin{cases} 1 & \text{if} \quad \phi > \varepsilon \\ \frac{1}{2}\left[1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\varepsilon} \right) \right] & \text{if} \quad |\phi| \le \varepsilon \\ 0 & \text{if} \quad \phi < -\varepsilon \end{cases}. \tag{14}$$

With a continuous viscosity and velocity field, the stress jump condition Eq. (5) reduces to

$$[p] = p_L^I - p_G^I = -\sigma\kappa. \tag{15}$$

The flow equations are discretized on a staggered Cartesian grid with the convection terms approximated by a third-order QUICK scheme[19] and other terms by the standard second-order central difference scheme. A semi-implicit time-advancement scheme is adopted to integrate the momentum equations with the second-order Crank-Nicolson scheme for the diagonal viscous terms and the second-order Adams-Bashforth scheme for the convective terms and other viscous terms. A four-step fractional-step method is employed for velocity-pressure coupling. The LS advection and re-initialization equations are solved using the third-order TVD Runge-Kutta scheme for time advancement and the fifth-order HJ-WENO scheme[20] for spatial discretization. The resulting pressure Poisson equation is solved using the PETSc library. The code is parallelized via a domain decomposition technique using the MPI library. Details of the numerical methods can be found in the studies by Yang and Stern.[16, 17]

III. PLS and CLSVOF Methods

A. PLS Method

The PLS method[6, 7] utilizes particles to correct the LS function in the under-resolved areas of the interface in order to preserve mass. Initially, particles are placed within a band across the interface. Each particle stores its position and radius, which is used to perform error correction on the LS function. The radius is set so that the boundary is just touching the interface:

$$r_p = \begin{cases} r_{\max} & \text{\textit{if}}\ s_p\phi(\vec{x}_p) > r_{\max} \\ s_p\phi(\vec{x}_p) & \text{\textit{if}}\ r_{\min} \le s_p\phi(\vec{x}_p) \le r_{\max} \\ r_{\min} & \text{\textit{if}}\ s_p\phi(\vec{x}_p) < r_{\min} \end{cases} \tag{16}$$

where s_p is the sign of the particle, set to +1 if $\phi(\vec{x}_p) > 0$ and -1 if $\phi(\vec{x}_p) < 0$, $r_{\min} = 0.1\min(\Delta x, \Delta y, \Delta z)$ and $r_{\max} = 0.5\max(\Delta x, \Delta y, \Delta z)$.

The positions of the particles are updated using standard Lagrangian update equation:

$$\vec{x}_p(t) = \vec{x}_p(t-1) + dt\vec{u}_{t-1}(\vec{x}_p(t-1)) \tag{17}$$

Whenever a particle escapes the interface by more than its radius (Fig. 1a), it will be used to perform error correction on the interface. The particle predicted LS value for each corner of the cell containing the escaped particle is defined as:

$$\phi_p(\vec{x}) = s_p(r_p - |\vec{x} - \vec{x}_p|) \tag{18}$$

Error correction is performed using the positive particles to create a temporary grid ϕ^+ and the negative particles to create a temporary grid ϕ^- . For each positive escaped particle, ϕ_p is found for each corner of the cell that contains the particle. The value for each corner is then set to

$$\phi^+ = \max(\phi_p, \phi^+) \tag{19}$$

For each negative escaped particle, ϕ_p is similarly defined for each corner of the cell that contains the particle. The value for each corner is then set to

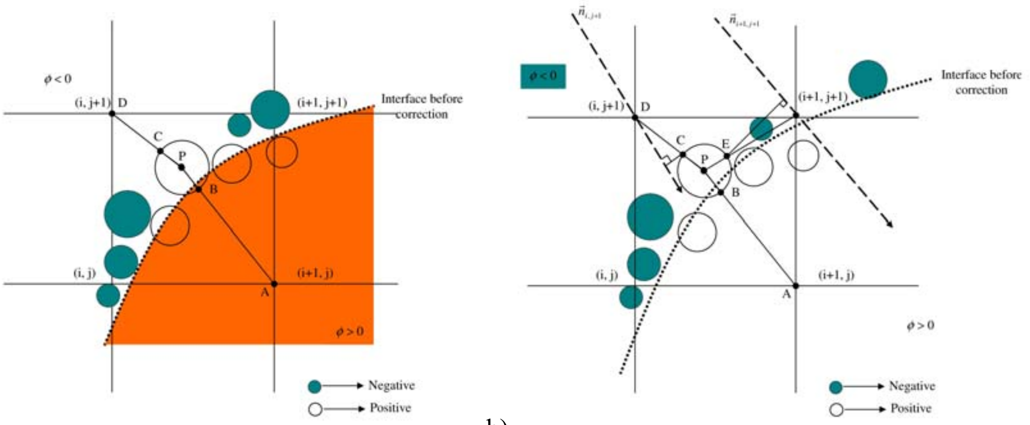
$$\phi^- = \min(\phi_p, \phi^-) \tag{20}$$

The LS is then reconstructed using ϕ^+ and ϕ^- by choosing the value with minimum magnitude at each grid node:

$$\phi = \begin{cases} \phi^+ & if|\phi^+| \le |\phi^-| \\ \phi^- & if|\phi^+| > |\phi^-|. \end{cases} \tag{21}$$

In the interfacial flows involving severe deformations, some regions may lack a sufficient number of particles whereas a large amount of particles may pile up in some other regions. It is necessary to periodically readapt particle distributions near the interface and adjust their radii accordingly.

In summary, the procedure of the operations is: evolve both the particles and the LS functions forward in time, correct errors in the LS functions using particles, reinitialize the LS function, again correct errors of the LS functions using particles, and finally adjust the particle radii and distribution.



a) b)

Figure 1. Particle correction for an escaped positive particle.

In the present study, a new interface reconstruction method based on the escaped particles is used to re-distance the LS functions across the interface. The initialization and update of the particles follow the same procedure as outlined above with exceptions of a new particle correction method and the corresponding reseeding strategy. Meanwhile, the interface is anchored when solving the re-initialization equations, the LS functions are only corrected after the advection process and additional correction after the re-initialization is not needed. This not only reduces the computational cost but also ensures the LS function to be a distance function, avoiding the otherwise disturbance caused by the correction errors. To accurately correct the LS function by the escaped particles, the particle prediction of the LS values is modified, e.g., for an escaped positive particle,

$$\phi_p(\vec{x}) = s_p(r_p - |\vec{x} - \vec{x}_p|) \quad \text{if } \phi \le 0 \tag{22a}$$

$$\phi_p(\vec{x}) = s_p(r_p + |\vec{x} - \vec{x}_p|) \quad \text{if } \phi > 0 \tag{22b}$$

This modification guarantees that the LS values on both sides of the interface can be corrected.

Note that the LS values calculated via the two equations given above are the distance from the cell corner to the particle surface, which does not guarantee the corrected LS function to be a distance function. Geometric information, normal vector, at the cell corner will be used to determine whether particle correction should be applied or not. If an escaped particle deviates too far away from the normal direction of a corner, this escaped particle will not be used to correct the LS value stored at that corner. For example, as shown in Fig. 1b, the values at both points (i, j) and $(i+1, j+1)$ are not corrected by the escaped positive particle P.

With these new particle correction algorithms, the errors due to the particle correction can be greatly reduced and more accurate interface representation can be obtained. Particles are reseeded at every time step simply by removing those particles drifting too far away from the interface and adding new particles in the regions with lack of sufficient number of particles.

B. CLSVOF method

In the CLSVOF method, the interface is reconstructed via a PLIC scheme from the VOF function and the interface normal vector computed from the LS function. Based on the reconstructed interface, the LS functions are re-distanced via a geometric procedure for achieving mass conservation. The interface reconstruction and LS re-distance are two essential procedures of the CLSVOF method.

1. *Interface reconstruction and propagation*

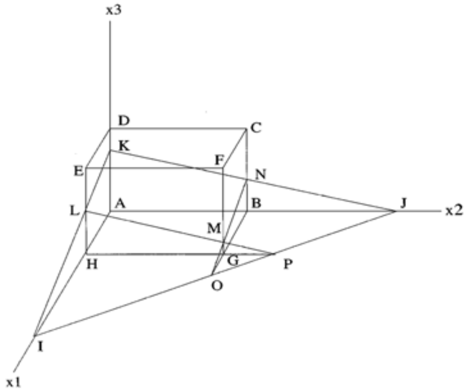


Figure 2. Interface reconstruction with a PLIC scheme.

The interface reconstruction is to locate the interface from the discrete VOF and LS values by a specific scheme, which serves two purposes: one is to calculate the VOF fluxes across each computational cell with an interface, and the other is to re-distance the LS function for achieving mass conservation. The VOF-PLIC scheme presented by Gueyffier *et al.*[15] is employed for the interface reconstruction, where the interface is represented by a plane on a 3D computational cell (see Fig. 2),

$$\mathbf{n} \cdot \mathbf{x} = n_1 x_1 + n_2 x_2 + n_3 x_3 = \alpha \,, \tag{23}$$

where α is a parameter related to the shortest distance from the origin. Different numerical methods for the normal vector estimation can be found in Ref. 21 in the context of the VOF formulation. In this study, the normal vector can be easily obtained from the LS function. Once the normal vector is known, the rest is to determine α from the *cut volume* defined by the VOF function. In Ref. 15, the expression for the *cut volume* was derived with respect to α, normal vector $\mathbf{n}$ and grid spacing $\Delta\mathbf{x}$,

$$Volume = \frac{1}{6n_1n_2n_3}[\alpha^3 - \sum_{i=1}^{3} F_3(\alpha - n_i\Delta x_i) + \sum_{i=1}^{3} F_3(\alpha - \alpha_{\max} + n_i\Delta x_i)] \,, \tag{24}$$

where $\alpha_{\max} = \sum_{i=1}^{3} n_i \Delta x_i$, and $F_n(y)$ is defined as,

$$F_n(y) = \begin{cases} y^n & \text{for y} > 0 \\ 0 & \text{for y} \le 0 \end{cases} . \tag{25}$$

Given the normal vector and grid spacing, the “forward” problem is to find the cut volume from α through Eq. (24), and the “inverse” problem is to determine α for a given *cut volume*. Detailed analytical relations connecting the *cut volume* and α for both the “forward” and “inverse’ problems are discussed in Ref. 22. With a careful investigation of the geometry and the expression (24), the possible cases are reduced to five basic configurations as shown in Fig. 3. In order to meet the requirement of a large range in spatial resolutions due to the complex flow structures of ship hydrodynamics, a non-uniform grid is adopted in this study. Although the scheme and analysis in Refs. 15, 22 are based on a unitary cubic grid, its extension to a non-uniform grid is straightforward.

Once the interface is reconstructed, the VOF function in the whole computational domain can be updated. In Ref. 15, a Lagrangian interface advection scheme is used, which has been found to be more robust and considerably simplifies programming with an operator splitting strategy to separately advance the interface in each spatial direction. The Lagrangian propagation scheme proposed in Ref. 15 is first order accurate in time, in this study, a second order Runge-Kutta scheme is used.

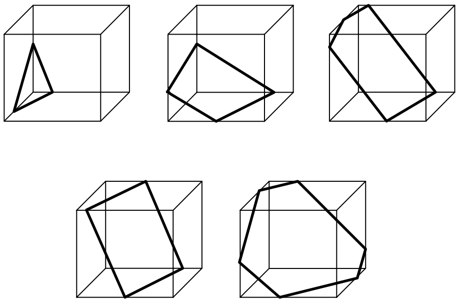


Figure 3. Different configurations of the interface reconstruction.

2. LS re-distance

After the LS advection equation is solved, the LS function needs to be re-distanced in order to achieve mass conservation. Different LS re-distance algorithms can be found in Refs. 4, 5, and 12, which are mainly focused on the 2D examples, and details for 3D cases are not available. Although the geometrical re-distance schemes presented in Refs.4, 5 can be extended to 3D cases, the procedure is more complicated and tedious. The algorithm presented in this study significantly simplifies the complicated geometric procedure by finding the nearest point on the reconstructed interface directly without considering the interface configuration.

The re-distancing of the LS function includes initial determination of the sign of the LS function and the subsequent calculation of the shortest distance from the cell centers to the reconstructed interface. The sign of the LS function, S^{ϕ}, is given by

$$S^{\phi} = sign(F-0.5)\,, \tag{26}$$

where $sign$ denotes a function that returns the sign of the numeric argument. It is obvious that when F>0.5, the cell center will fall inside the liquid where the LS function takes the positive sign, and vice versa.

Next, the magnitude of the LS function is determined, which is the most important step of the re-distancing process. The basic idea is to find the closest point on an interfacial cell to the neighboring cell centers. Although various configurations of the reconstructed interface exist, generally, all the interfacial cells can be simply divided into two cases: single-phase cells (i.e., F=0 or 1*)* and interfacial cells (i.e., 0<F<1). For two adjoining cells (i,j,k) and (i',j',k'), the closest point on cell (i,j,k) to the center of cell (i',j',k') will always be either at the corner or at the centroid of the cell faces. Therefore, these points should always be considered first when calculating the shortest distance associated with two adjoining cells. When a cell contains an interface segment, for a 2D case, the closest point on the segment will be either one of the two endpoints or the projection point of the neighboring cell center. For example, as shown in Fig. 4, for cell ($i+1,j$), the shortest distance is from the cell center to face centroid; for cells ($i,j+1$) and ($i+1,j+1$), the nearest point is projection point on the segment and the endpoint of the segment, respectively. As for 3D cases, the reconstructed interface contained in the cell is an n-sided (from 3-6) polygon, the possible closest point on the polygon can be the projection point of the neighboring cell center to the interface, one of the vertices, and the projection point to one of the polygon sides from the neighboring cell centers. For a given interfacial cell (i,j,k) and its neighboring cell (i',j',k'), the re-distance procedure is given as follows:

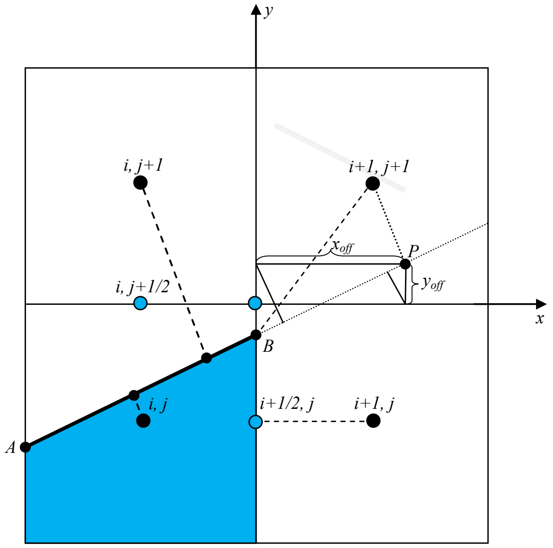


Figure 4. Schematic for LS function re-distance.

(1). The closest point on the cell boundary, $\mathbf{X}_v$, is determined first. This point is either the face centroid or the cell corner, as shown in Fig. 4. If $\mathbf{X}_v$ and the neighboring cell center are in different phases, $\mathbf{X}_v$ will be the nearest point to the neighboring cell center. Otherwise, go to step (2).

(2). Find the projection point of cell (i',j',k') center onto the interface, $\mathbf{X}_p$. If point $\mathbf{X}_p$ falls inside of cell (i,j,k), then $\mathbf{X}_p$ will be the closest point to cell (i',j',k'). If point $\mathbf{X}_p$ falls outside of cell (i,j,k), then go to step (3).

(3). Find the point on the boundary of the interface segment. For a 2D case, it will be one of the two endpoints. For the 3D problems, it will be one of the vertices or the projection point to one of the polygon sides.

IV. Numerical Tests

The PLS and CLSVOF methods, along with the pure LS method, are tested with the prescribed velocity fields first. This allows direct comparison of the interface tracking/capturing schemes alone without the involvement of the flow solver.

A. Slotted (Zalesak's) Disk

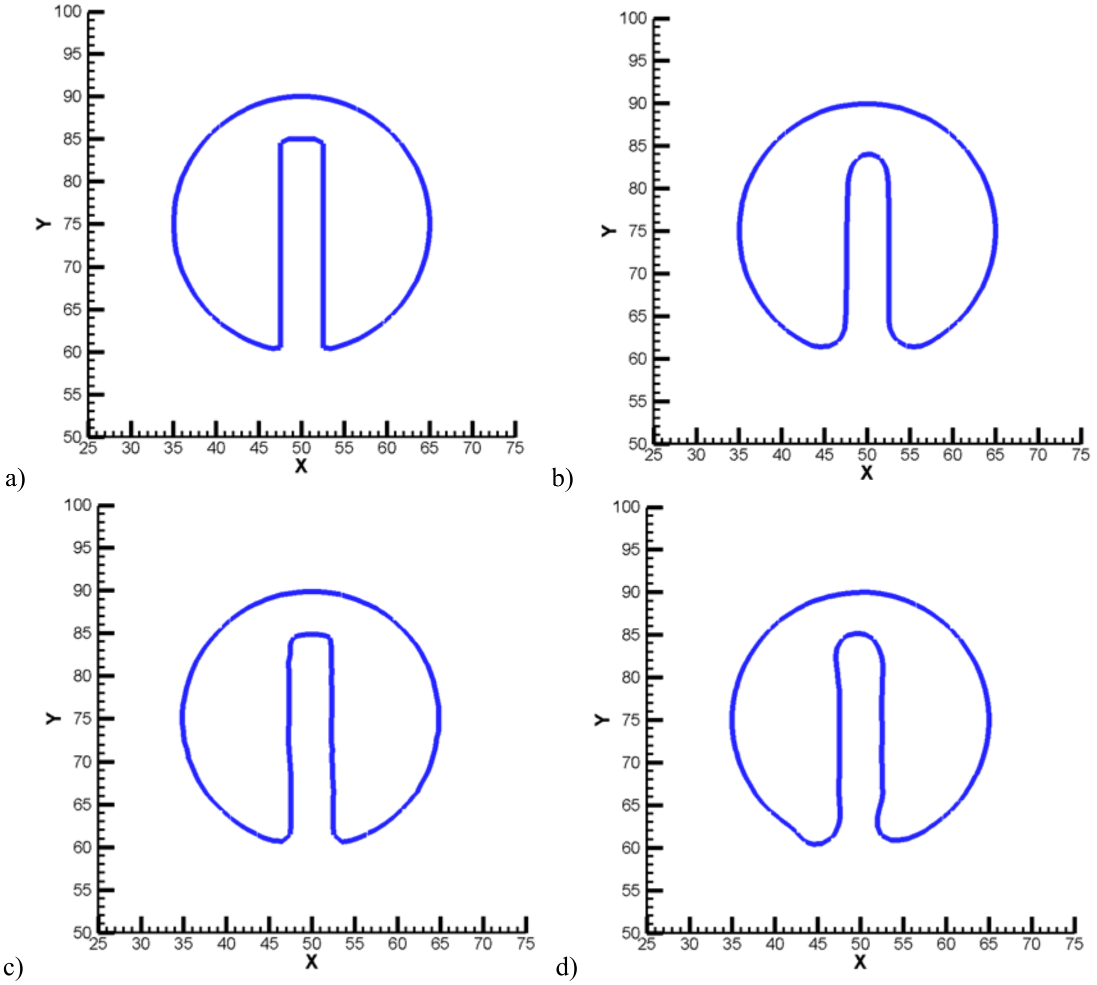


Figure 5. Comparison of the solutions for different methods after one rotation. a) Initial shape; b) LS; c) PLS; d) CLSVOF. Grid: 100×100.

A slotted disc with a radius of 15.0 and slot width of 5.0 is initially located at (50.0, 75.0) on a 100×100 computational domain. This problem, referred to as the Zalesak's problem, is often used for the interface tracking/capturing scheme test. The prescribed velocity field is given as:

$$u = (\pi/314)(50.0 - y),$$

$$v = (\pi/314)(x - 50.0),$$

with the axis of rotation centered at (50.0, 50.0).

The results after one rotation for all the three methods are shown in Fig. 5. As expected, the sharp edges of the slot are smoothed out in the pure LS method, whereas the PLS method well preserves the sharp corners. Although the CLSVOF method exactly conserves mass in this case, the edges at the corners are also smeared with the symmetry slightly lost. This is because a VOF-PLIC reconstruction smoothes the regions with high curvatures, and the discontinuity near the corner will be progressively smeared out and advected faster in the rotation flow[23]. It should be noted that for the rigid body rotation, as in this case, there is no relative movement among the particles in the PLS method, which makes particle reseeding process not necessary. All the initially prescribed particles can always return back to their original locations after every rotation. Therefore, it is not surprising that the PLS method gives excellent results in this test.

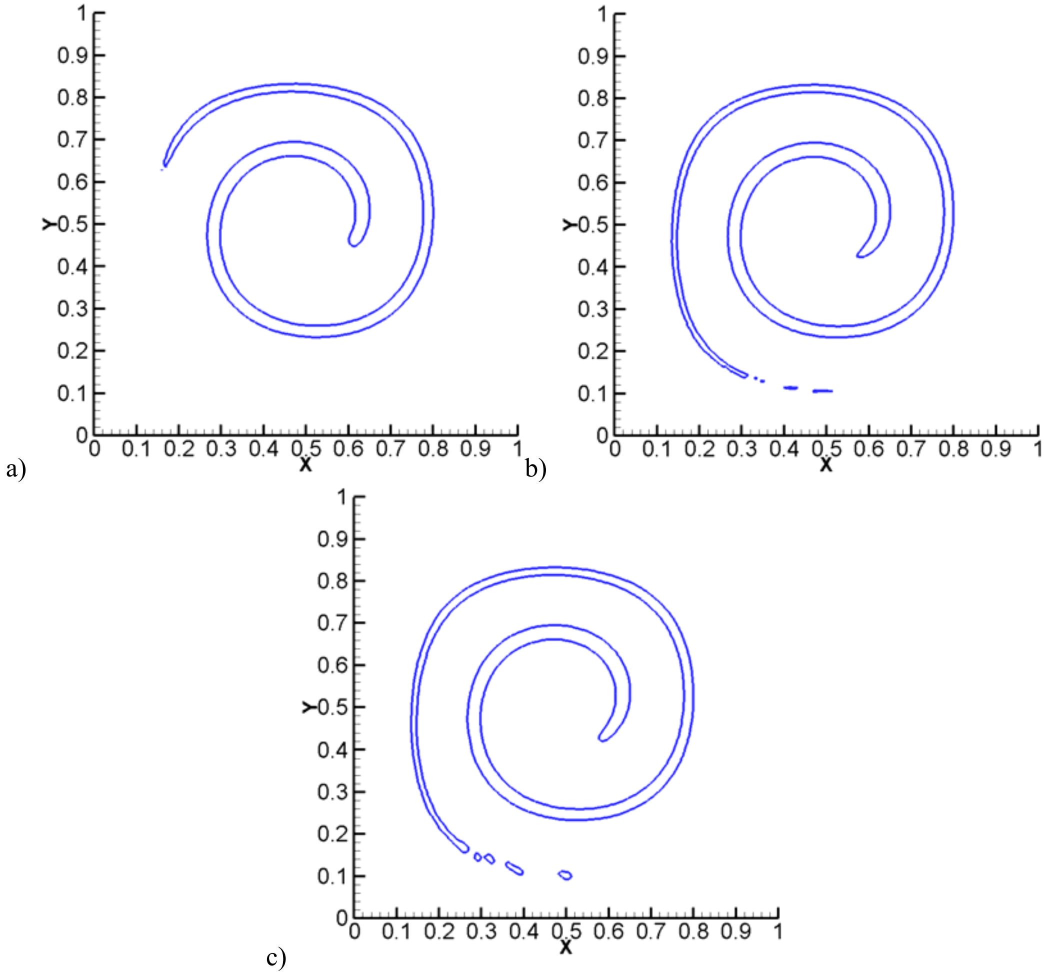


Figure 6. Single vortex flow test at *t*=4. a) LS; b) PLS; c) CLSVOF. Grid: 128×128.

B. Rider-Kothe Single Vortex Flow

A circle evolving in a shearing flow is another challenging test for interface modeling schemes, which involves severe topological changes. The circle is stretched and torn in this vortex flow where very thin filaments on the scale of the mesh can be produced. A prescribed shearing flow is given by:

$$u = (\sin(\pi x))^2 \sin(2\pi y)\cos(\pi t/T),$$

$$v = (\sin(\pi y))^2 \sin(2\pi x)\cos(\pi t/T),$$

where t is time, T is the time at which the flow returns back to its initial shape. A circle with a radius of 0.15 is prescribed at (0.5, 0.75) on a computational domain of 1.0×1.0. The resulting velocity field stretches out the circle into a very long, thin fluid element which progressively wraps itself towards the center of the domain.

The results at t=4 with different methods are shown in Fig. 6 where a maximal stretching is reached with T chosen as 8. It is clear that both the PLS and CLSVOF methods well maintain the thin, elongated ligaments on the scale of the grid spacing, whereas serious mass loss occurs at both the head and the tail of the filament for the LS method. Fig. 7 shows the results at t=8 at which the flow field returns back to its initial state. For the PLS method, it can be seen that the recovered circle is very comparable to the initial shape. The recovered shape in the CLSVOF method is roughly a circle, the interface is not smooth. As for the pure LS method, the recovered shape is far from a circle with serious mass loss.

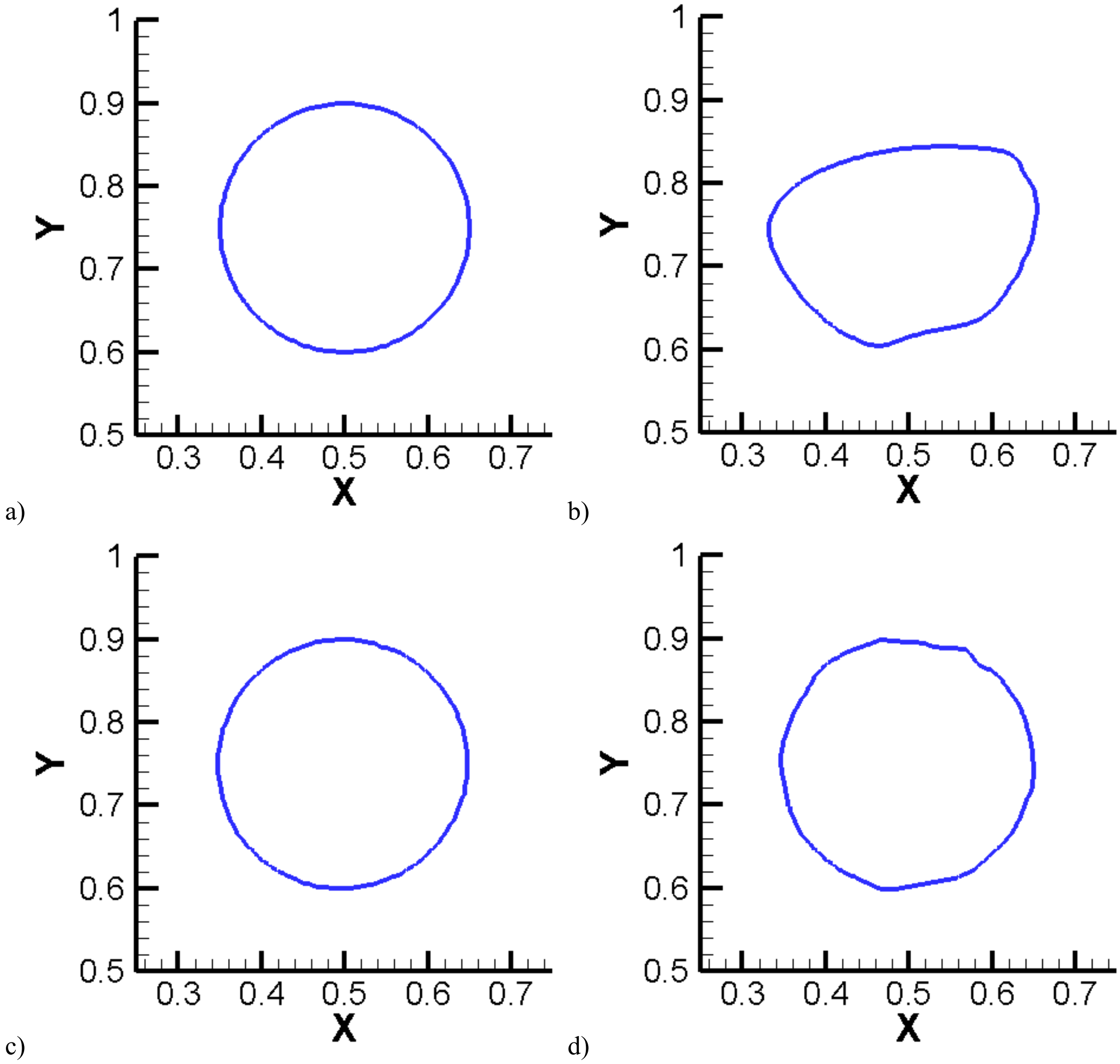


Figure 7. Single vortex flow test at *t*=8. a) Initial shape; b) LS; c) PLS; d) CLSVOF. Grid: 128×128.

C. 3D Deformation Field

To demonstrate the ability of the method to capture three-dimensional deformations, a three-dimensional incompressible flow field which combines deformations both in the x-y and x-z planes is considered. The velocity field is given by

$$u = 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/T),$$

$$v = -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos(\pi t/T),$$

$$w = -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos(\pi t / T)\,,$$

where T=3.

A sphere of radius 0.15 is placed within a unit computational domain at (0.35, 0.35, 0.35). A uniform grid of 100×100×100 is used. The sphere is stretched by two rotating vortices which initially scoop out opposite sides of the sphere and then reverse them back to the initial shape. The deformed shapes at t=1.5 when it reaches the maximum stretching are shown in Fig. 8a, and the recovered shapes after the flow returns back are shown in Fig. 8b. As shown in Fig. 8a, both the PLS and the CLSVOF methods partially resolve the thin interface produced at the middle section of the stretched shape. The PLS method recovers the sphere quite well with less volume loss. For the CLSVOF method, a sphere with a scar is obtained due to the accumulated numerical errors in the interface reconstruction process. On the other hand, the LS method fails severely for mass conservation.

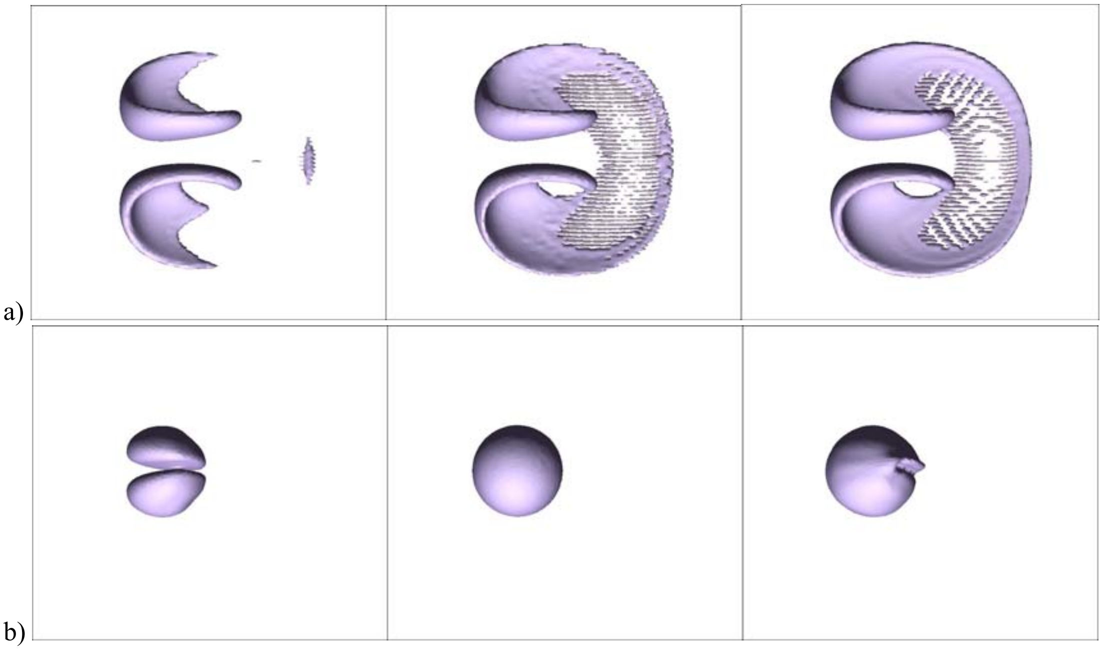


Figure 8. Deformation of a sphere. a) t=1.5; b) t=3.0. Left: LS; middle: PLS; right: CLSVOF. Grid: 100×100.

In the above two tests, the PLS method apparently performs better than the CLSVOF method in recovering the initial shape. The velocity fields used are prescribed without solving the flow equations where the interface geometrical change has no influence on the flow field. For the PLS method, the initially prescribed particles can always return back to their initial locations after the flow reverses. Therefore, if the initially prescribed particles are kept as much as possible during reseeding process, the initial shape can be recovered since the particle movement is independent to the LS function. In fact, without the reseeding operations, the initial shapes can be exactly recovered. However, in practice, particle reseeding process must be used when more complicated interface interactions, such as merging or break-up, occur. The particle reseeding operation is dependent on the LS information only, thus frequent reseeding will decrease the accuracy of PLS method.

V. Validation Studies

With the combination of the two-phase flow solvers, a water drop impact onto a liquid pool and wave breakings over a submerged bump are simulated in order to validate the accuracy of the interface modeling methods.

A. Liquid Drop Impact

In Ref. 14, a liquid drop impact onto the surface of a liquid pool was simulated using the PLS method. It is a good example for interface modeling methods since it involves complicated flow deformations such as splashing, coalescence and bouncing. In this test, a water droplet of radius 1/3 m initially positioned at (1.0 m, 1.0 m, 1.5 m) impinges upon a pool of water of 1 m in depth. The initial velocity used here is -1 m/s which is much less than that (-5 m/s) used in Ref. 14. It is computational expensive to use a large initial velocity, since it not only requires a large computational domain to capture the rebounding liquid jet but also needs a very small time step due to the time step constraints. Constant physical properties of air and water are chosen as: g=-9.8 m/s^2, σ =0.0728 kg/s^2, ρ_l =1000 kg/m^3, ρ_g =1.226 kg/m^3, μ_l =1.137$\times$10^{-3} kg/ms and μ_g =1.78$\times$10^{-5} kg/ms. The computations are carried out on a domain of 2.0m $\times$ 2.0m $\times$ 3.2m. The grid used is 60×60×96 and non-slip solid wall boundary conditions are specified at all the domain boundaries.

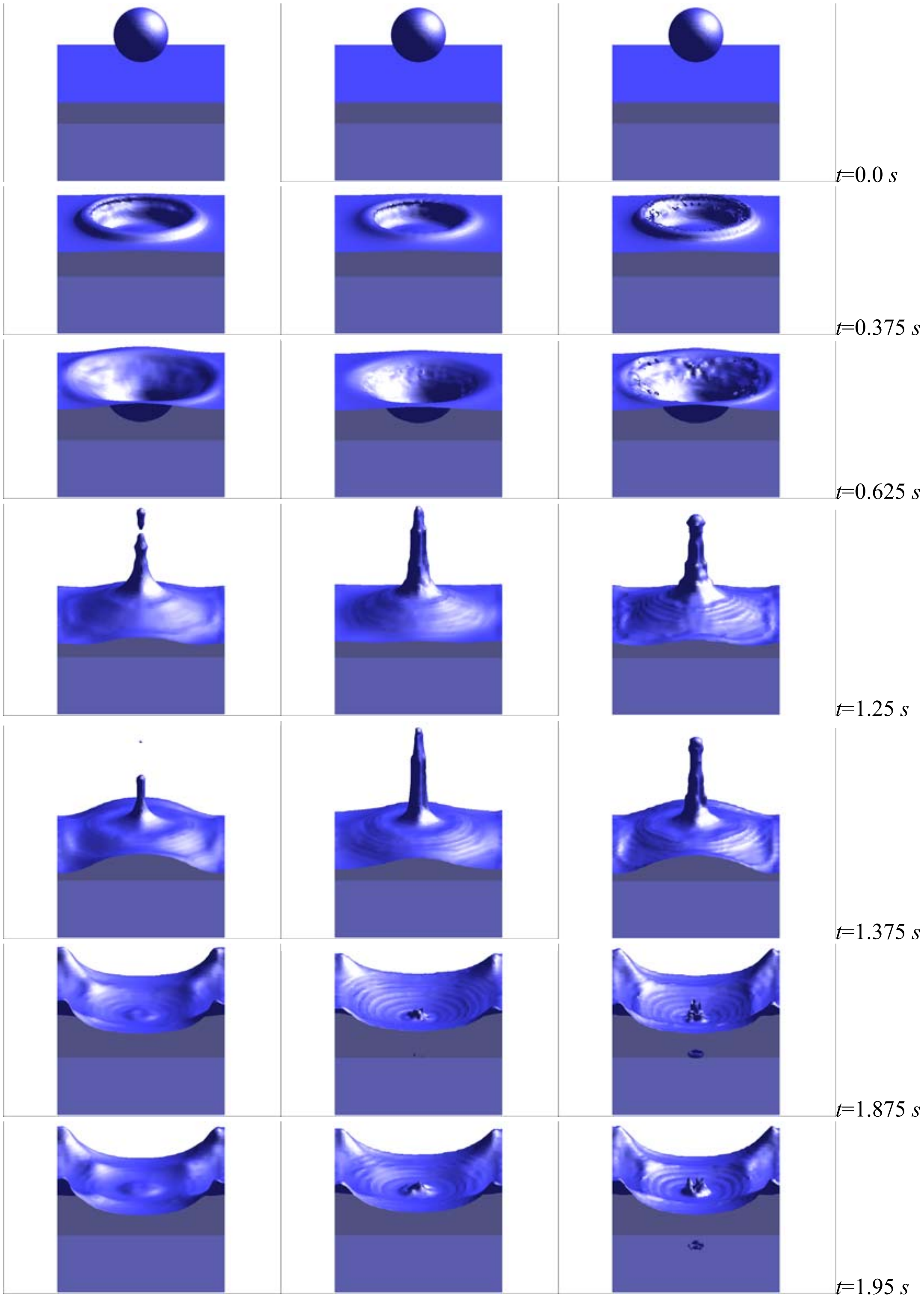


Figure 9. Time sequences of a water drop impingement onto a liquid pool. a) LS; b) PLS; c) CLSVOF.

Time sequences of the drop impingement process for all the three methods are given in Fig. 9. As shown in the figure, after the impact of the droplet onto the water surface of the pool, a big dent on the water surface is produced first, followed by the formation of a water jet. The jet then falls back to the water surface. Generally, the major events of the impingement process are demonstrated in all the three methods.

After the drop hits the water surface, a big dent is produced on the liquid surface, meanwhile, many water splashes are generated due to the drop impact which can be seen in the CLSVOF method at t=0.375 s and t=0.625 s. In both the LS and PLS methods, however, water splashes are not properly captured. Although a very large initial velocity was used in Ref. 14, water splashes were still not visible. The water surface rebounds upwards from the bottom of the dent, and then a jet is formed and reaches its maximum height between t=1.25s and t=1.375 s. The maximum jet height is about 3.6D (D, the diameter of initial drop) which is comparable to the maximum jet height of approximately 3.75D in Ref. 14 even though, where a much larger initial velocity was used. Apparent mass loss occurs in the LS method after the formation of the liquid jet, where the jet shrinks gradually and then vanishes. In both the PLS and CLSVOF methods, the jet is well maintained while it is falling down to the water surface. As the jet falls down, some air bubbles are formed due to the entrainment of air around the jet, which is clearly indicated between t=1.875s and t=1.95 s in the CLSVOF method. Air bubbles can not be found in both the PLS and the LS method. Moreover, in the PLS method, the interface is not smooth in some regions and symmetric problems can also be found. This is probably caused by the random distribution of the particles along the interface.

It is clear that the PLS method performs better than the LS method in mass conservation, and the results obtained from the current PLS method is comparable to those reported in Ref. 14 in which the original PLS method was used. As for the CLSVOF method, more details of the interface structures, such as water splashes, air entrainment, are captured in the drop impingement process. Moreover, the computational cost of the PLS method is more expensive than the CLSVOF in terms of CPU time.

B. Plunging Wave Breaking

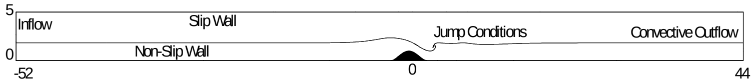


Figure 10. Computational domain for wave breaking over a submerged bump.

A complementary experimental and computational study of plunging breaking waves generated in an open channel flume using a bottom bump and impulsive accelerated flow was presented in Ref. 24. The geometry is of particular relevance to ship hydrodynamics, since the breaking involves body-wave interactions. The experiments were carried out in an open channel flume (9 m long, 0.6 m wide and 0.43 m high). The time evolution of the transient wave and its flow properties were measured using experimental fluid dynamics (EFD): upstream and downstream velocity and flow rates using pitot probes; air-water interface elevation measurements and two dimensional particle image velocimetry (PIV) in the wave breaking region. The computational fluid dynamics (CFD) study is summarized below with a focus on the wave breaking process and the comparison of the performance of the interface modeling methods. Since strong water/air interactions are involved during the wave breaking process, it is a very challenging case for interface modeling method.

The numerical simulations are conducted on a 2D computational domain of x = [-52, 44] and z = [0, 5], with a grid size of 768 × 256. The boundary conditions are given in Fig.10. The inlet velocity imposed at the left boundary is 0.87 m/s for water and zero for air. The initial free surface elevation is ζ = 1.85 and a uniform velocity field of 0.87 m/s is prescribed in the water domain at t=0 with the air phase at rest. The corresponding Re is 99441 and Fr is 0.82. In the experiments, the flow is transitional and completely impulsive and wave breaks during the flow acceleration process. The experimental flow conditions are complicated and hence are difficult to model, for simplicity, constant inlet flow condition is used in the computations. The initial and inlet velocity and free surface elevation are chosen based on the sensitivity study discussed in Ref. 24 in order to match wave breaking location in the EFD. The EFD flow conditions will be used in the future work. The time step initially is 5×10^{-5} and then is reduced to 1×10^{-5} after the flow gets violent due to wave breaking.

The time sequences of wave breaking process computed using the three different methods are given in Fig. 11. As was discussed in Ref. 24, the major events in the wave breaking process are maximum wave height, first plunge, oblique splash, vertical jet, repeated processes, chaotic motions and broken wave swept downstream. At the early stage of the computations, the general structures of the wave profile are very similar. The maximum wave height (frame 1) and the first jet plunge (frame 2) are well demonstrated in all the three methods. After the jet hits the trough surface (frame 3), an oblique splash-up with many small droplets can be found in the CLSVOF method, this is in agreement with the experimental findings. As for the LS and PLS methods, the oblique splash-up is not properly captured. The vertical jet reaches its maximum height (frame 4), which can be seen in all the three methods. In the subsequent events (frame 5-8), the CLSVOF method shows the second plunge, oblique splash-ups and vertical jet, and the third plunge at last. In both the LS and PLS methods, however, only the second jet plunge is indicated after which the flow appears to enter a chaotic motion of air and water. Moreover, in the entire wave breaking process, small scale droplets and air bubbles can hardly be found in the LS method. As for the PLS method, more fine structures of the interface have been captured than the LS method. The CLSVOF method is likely more suitable for modeling the wave breaking phenomenon, since it is able to capture more details of the wave breaking events.

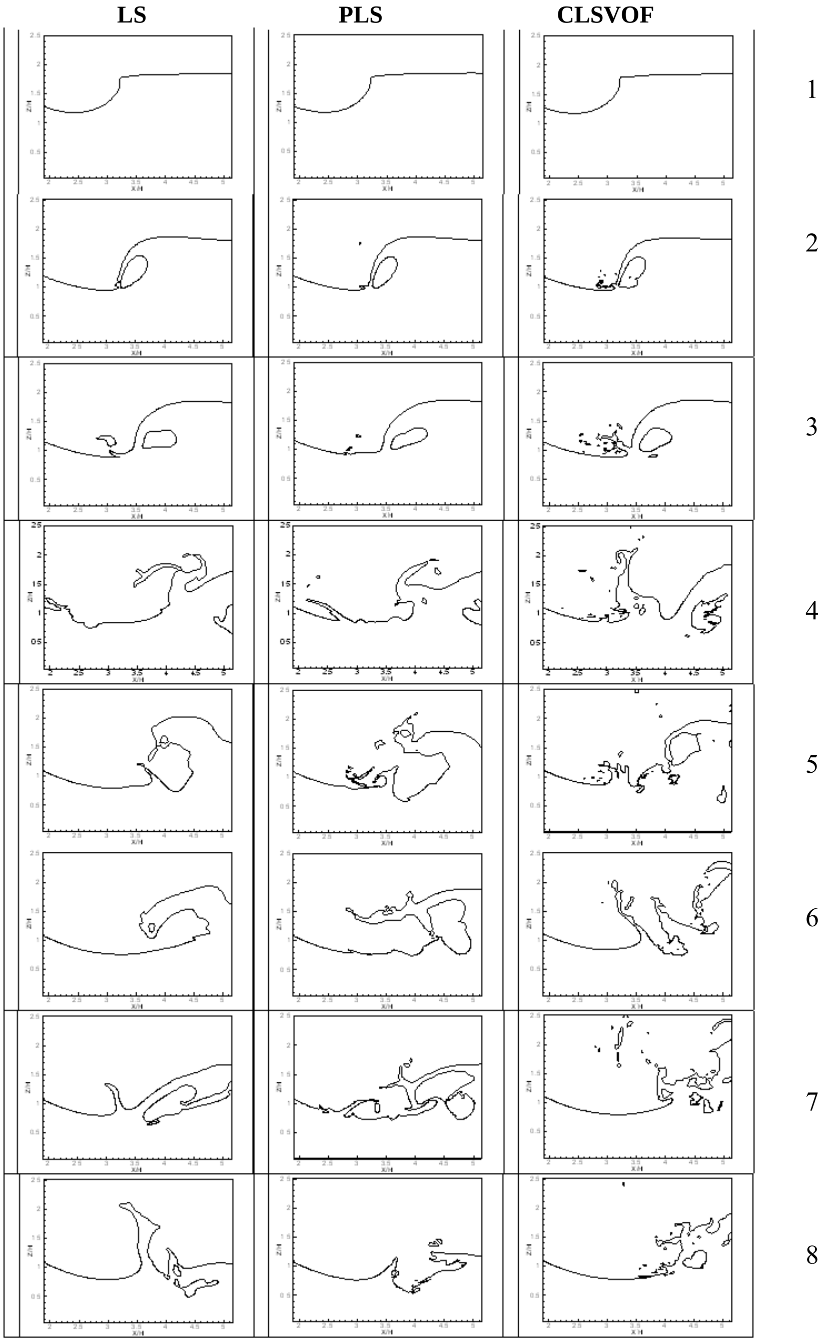
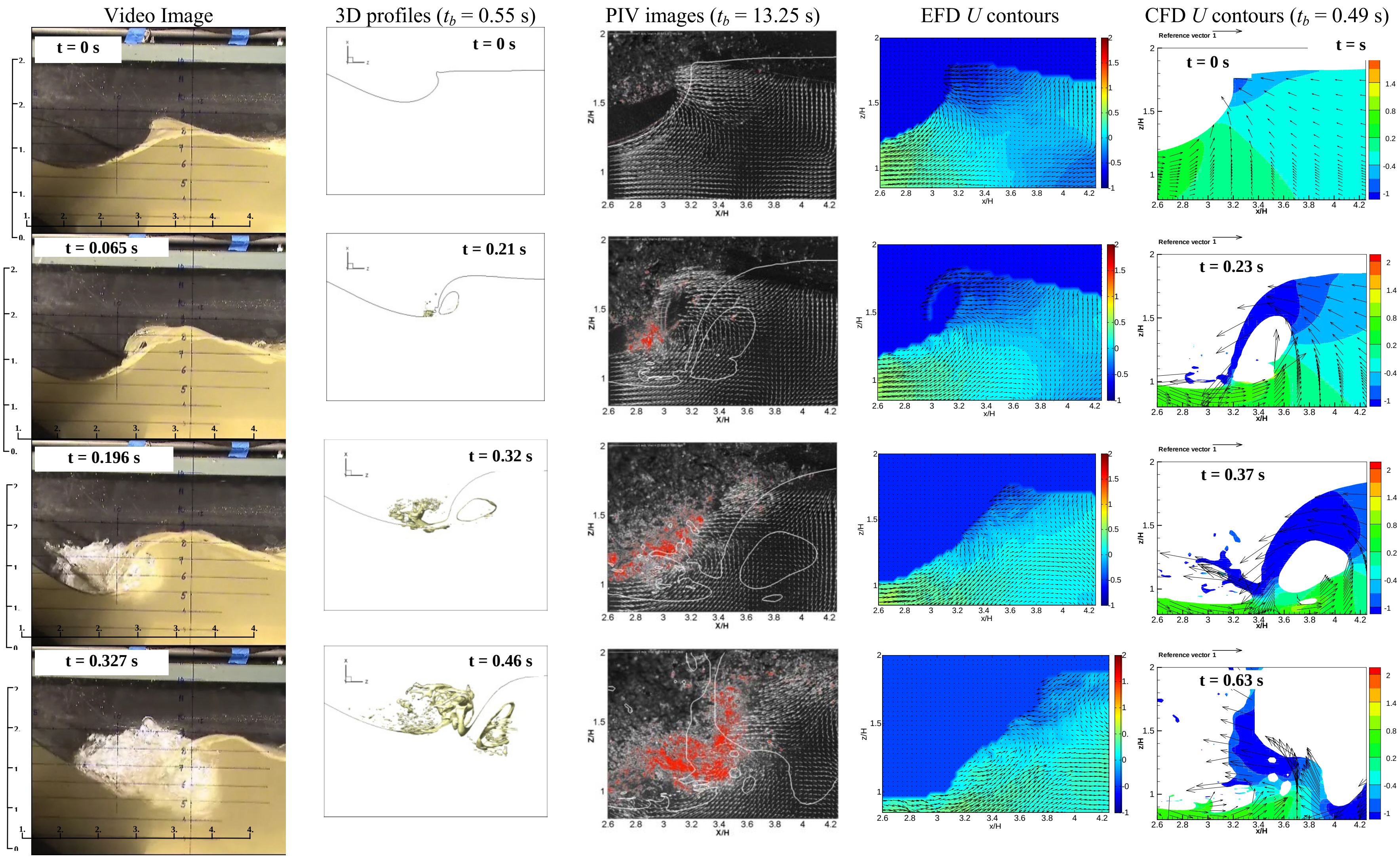


Figure 11. Instantaneous free surface profiles of the Wave breaking process.



Video Image

3D profiles (t_b = 0.55 s)

PIV images (t_b = 13.25 s)

EFD U contours

CFD U contours (t_b = 0.49 s)

Figure 12. Impulsive breaking waves behind a bump. From left to right: video image; 3D wave profile; PIV image with overlaid CFD wave profile; EFD *U* contours and CFD *U* contours.

Detailed plunging wave breaking process was discussed in Ref. 24 with the major wave breaking events identified by using the CLSVOF method. 3D Large Eddy Simulation (LES) of the wave breaking is also performed using the CLSVOF method with some preliminary results presented. The plunging wave breaking process has been characterized in the previous studies[25-27] by four major phases including steep wave formation, jet formation and overturning, splash-up and air entrainment as summarized in Table 1 of Ref. 24. The first two phases, i.e., steep wave formation and jet overturning, are very similar in most experimental and numerical studies even though the flow conditions and the mechanism that induces wave breaking are different. However, the subsequent phenomena after the jet overturning observed in the above studies vary considerably. The angle of the overturning jet just prior to the plunge varies with different studies, while the surface profile beneath the jet is approximated with the similar elliptical shape. After the jet hits the trough surface, the splash-up angles are different for different studies, and different researchers have different opinions about the origin of the splash-ups. Most of the previous studies have reported occurrence of successive splash-up cycles with reduced energy after the first plunge and the degeneration of the flow into a chaotic motion.

Herein, wave breaking is triggered by the flow over a submerged bump which differs significantly from the previously mentioned experimental studies in flow conditions, jet overturning directions and subsequent events. The following major events in the plunging wave breaking process have been identified: maximum wave height, first plunge, oblique splash, vertical jet, two repeated processes, chaotic motions and broken wave swept downstream. Fig.12 summarizes the overall PIV results (close-up views) in the wave breaking region and its comparison with CFD at various time steps that describe the most important events in the wave breaking process. Video images of the plunging breaker with reference scales and the instantaneous 3D free surface profiles of plunging waves are presented in column 1 and 2 respectively. PIV images with overlaid 2D CFD air-water interface profiles are shown in column 3. t_b is a reference time at which the wave reaches its maximum height. Time $t = 0$ refers to t_b according to Eq. (3) of Ref. 24. The CFD air-water interface profiles were initially studied to identify the overall wave breaking process and its major events. Subsequently, those events were also identified qualitatively from the video and PIV images in EFD. In the current layout of Fig. 12, the EFD and CFD results are compared side by side for those particular events and hence the time instances do not necessarily match due to the differences in the initial conditions. Although the CFD time is given in the CFD U contour plot, unless otherwise specified, the EFD time is always referred to in the discussions.

At time $t = 0$ the wave crest becomes steepest when it reaches its maximum height. As the wave crest starts to overturn the steep angle θ_E, defined as the angle between the horizontal free surface and the jet's longitudinal axis, also increases, and its edge breaks into some small droplets in air. At the instant just prior to the first plunge, the CFD steep angle is approximately 50 degrees which is less than the EFD value but matches Ref. 27. At $t = 0.065$, the first plunge occurs when the overturning jet impinges onto the free surface of the trough. The θ_E increases to approximately 75 and 85 degrees in the CFD and EFD profiles, respectively compared to the previous time step. With the overturning jet, a large amount of air below the jet is entrapped which forms a big air bubble. The entrapped air bubble can also be observed in the PIV image at $t = 0.065$. The size of the air bubble in CFD is almost 1.5 times larger than the EFD. The CFD results show that the entrapped air bubble initially resembles an ellipse with an axis ratio of 2.153 which is slightly larger than those (usually, $\sqrt{3}$) found in the EFD and previous studies. Once the jet tip touches the trough surface, splash-up initiates and develops at the location where the jet impacts. At $t = 0.196$ as shown in CFD plot, an oblique splash-up is generated towards the upstream direction with a spray region. This spray region is more clearly shown in the experimental video image. The PIV image also shows that the red (the color indicates bubble concentration) region intensifies. The splash-up angle differs from Ref. 27 in which the splash-up is initially vertical and then becomes oblique towards the original wave crest. The observations of the present study are close to those reported in Refs. 25, 26. This oblique splash-up appears to "rebound" from the overturning jet rather than originates from the trough surface, which can be clearly seen from the CFD velocity vector field. The origin of the oblique splash-up is also indicated in the 3D wave profile. The splash-up intensifies and a vertical jet can be clearly observed in the CFD profiles with its maximum height of $\zeta = 2.15$ at $t = 0.327$. The video image shows the spray zone grows in size with more intense splash-up reaching a maximum height of $\zeta = 2.22$. In between $x = 2.2$ and $x = 4.6$ the broken wave creates an aerated region. A vertical jet can also be observed at $x = 3.05$ that is ejected from the trough. The CFD velocity vector field shows that the fluid of vertical jet mainly comes from the disturbed trough surface, which can also be seen from the 3D wave profile. After the first wave plunge, the splash-up initially originates from the plunging crest which forms an oblique spray and intensifies with more water coming from the trough. This agrees with the explanation[27] on the origin of the splash-ups. The initially entrapped air bubble eventually breaks up at around $t = 0.392$ (not shown here). Severe bubble shape deformations can be seen in the CFD plot as it moves downstream. As the vertical jet decreases in height, another jet is formed which starts to initiate the second plunge, followed by the second oblique splash-ups and the second vertical jet. After the third jet impacts onto the trough surface, multiple small scale splash-ups and plunges keep occurring, and finally the broken wave is swept downstream leaving a trail of aerated region.

The CFD profiles show that the wave breaking occurs further downstream compared to the EFD profiles, which is clearly seen in the first and second plunging events. This is likely because the initial mean flow velocity is higher in the CFD simulations. It is found that the entrapped air bubble after the first plunge collapses in EFD, but remains intact in CFD. This may be because of 3D instability, which can contribute to the relatively short life of the air bubble[25] that does not exist in the 2D simulations. A clear two-phase (aerated) region is demonstrated in the experimental video images after the jet overturning. However, it is hard to define an exact two-phase region for the 2D simulations. As for the 3D case, more splash-ups can be observed, and the region of spray is well defined. The entrapped air bubble undergoes serious shape deformations and eventually collapses due to the 3D instabilities. Since the interface is modeled via an interface tracking method which is not able to capture air bubbles or droplets at the scale less than the grid spacing, a mixture model is needed in order to properly simulate the aerated region in future investigations.

The fourth and fifth columns of Fig.12 show the EFD *U* contours and the CFD *U* contours, respectively at the time steps that correspond to the wave breaking process with the velocity vectors overlaid on top. All contour plots and velocity vectors are presented in dimensional form. All EFD data is ensemble averaged and CFD data is instantaneous. The mean stream-wise velocity component within the water phase of the display area at each time step has been subtracted from the vector fields for both EFD and CFD. To be consistent with the vector representation the mean stream-wise velocity has also been subtracted from the background contours. The CFD velocity vectors of column 5 are of the same magnitude as the EFD vectors of column 3 and not the vectors in column 4 whereas, the EFD velocity contours in column 4 are of the same scales as the CFD velocity contours of column 5. Even though the CFD simulations were conducted for a much larger domain size, in the present layout the CFD x and z limits are matched with the EFD scales for comparison. At time $t = 0$, the EFD results show a positive U region extends towards downstream with the magnitude reduced in the region where steep wave is formed. A negative U region in the crest initiates and intensifies after the first plunge. For $0 \leq t \leq 0.327$, the positive U region moves towards downstream with the magnitudes continuing to increase, the EFD results show resolvable details of steep wave formation, first plunge, oblique splash and the vertical jet. Similar flow trends can be found in CFD U contours, with more detailed resolutions of plunging, splashing, vertical jet and bubble entrainment events in the wave breaking region. In the EFD, a clockwise rotating span-wise vortex is found to be shed from the bump and convected in the vertical direction by the rising fluid and also in the downstream direction by the mean flow. This bump vortex cannot be observed in the CFD results, this is due to the differences in the impulsive start conditions. The wave breaking vortex is clearly demonstrated in the CFD vorticity plot (not shown here) as part of plunging and bubble entrainment events.

VI. Conclusions

In this study, the PLS method and the CLSVOF method have been implemented in order to improve the mass conservation property of the LS method. A generalized particle reseeding algorithm is developed for the PLS method, and an efficient LS re-distance algorithm is used for the CLSVOF method. The performance of these methods has been evaluated through a series numerical tests and application examples. It has been found that both PLS and CLSVOF methods have greatly improved the mass conservation property of the LS method, which is consistent with the tests reported in the literature. The PLS method performs better than the CLSVOF method in the standard tests. In the application examples with combination of the flow solver, the CLSVOF method shows more realistic and reasonable results than the PLS method. As compared to the CLSVOF method, the implementation of the PLS method is relatively easy, but it is computationally expensive for the three dimensional problems.

In the drop impact test, the PLS method demonstrates better mass conservation property than the LS method, and the results of the current PLS method are comparable to those presented in Ref. 14. In the CLSVOF method, more details of the drop impingement process, such as water splashes, air entrainment, are captured.

Plunging wave breakings over a submerged bump is modeled with the results compared with the EFD data. More detailed interface structures are captured in the CLSVOF method. The major wave breaking events are identified from the CFD results and are qualitatively confirmed by the EFD findings. CFD studies revealed that the major events at the early wave breaking stage, such as maximum wave height, first plunge and splash-ups are similar to the observations reported in the literature. Subsequent events are examined and some distinct wave breaking events, such as vertical jet formation, two (even more) repeated processes with reduced amplitudes, chaotic motions and broken wave swept downstream, are identified in the later wave breaking stages. The number of repeated processes likely depends on the acceleration flow conditions. CFD results suggest that the first oblique splash-up appears to rebound from the overturning jet, whereas the first vertical jet originates from the disturbed trough. The flow structures are also compared with the EFD data qualitatively, similar flow trends have been observed in the CFD results. Due to the different initial flow conditions used in the CFD, bump vortex observed in the EFD cannot be found in the CFD.

For the wave breakings over a submerged bump, further investigations using the EFD initial and boundary conditions should be made in order to quantitatively validate the CFD results. Quantitative verification of the 2D-CFD results needs to be further investigated. A fully 3D large eddy simulation (LES) including surface tension is necessary for a better understanding of the physics of the plunging wave breaking phenomenon. In order to further validate the CLSVOF method, more test cases, such as wave breakings induced by a ship bow, will be investigated in the future work.

Acknowledgments

This work is sponsored by the US Office of Naval Research through research grants N00014-01-1-0073 and N00014-06-1-0420 under the administration of Dr. Patrick Purtell. The numerical tests and simulations were performed using DoD HPC resources. The authors would like to thank Bonguk Koo for his contribution to the plunging wave breaking simulations.

References

[1]Osher, S. and Sethian, J.A., “Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton--Jacobi Formulations,” *J. Comput. Phys*., Vol. 79, 1988, pp. 12-49.

[2]Sethian, J. A., *Level set methods and fast marching methods*, Cambridge University Press, 1999.

[3]Carrica, P.M., Wilson, R.V. and Stern, F., “An Unsteady Single-Phase Level Set Method for Viscous Free Surface Flows,” *Int. J. Numer. Meth. Fluids*, Vol. 53, 2007, pp. 229-256.

[4]Sussman, M. and Puckett, E.G., “A coupled level set and volume-of-fluid method for computing 3D axisymmetric incompressible two-phase flows,” *J. Comput. Phys.*, Vol. 162, 2000, pp. 301-337.

[5]Son, G., “Efficient Implementation of a coupled level-set and volume-of-fluid method for three-dimensional incompressible two-phase flows,” *Numer. Heat Transfer B*, Vol. 43, 2003, pp. 549-565.

[6]Enright, D., Fedkiw, R., Ferziger, J. and Mitchell, I., “A Hybrid Particle Level Set Method for Improved Interface Capturing,” *J. Comput. Phys*., Vol. 183, 2002, pp. 83-116.

[7] Enright, D., Losasso, F. and Fedkiw, R., “A Fast and Accurate Semi-Lagrangian Particle Level Set Method, Computers and Structures,” Vol. 83, 2005, pp. 479-490.

[8]Osher, S. and Fedkiw, R., “Level set methods: an overview and some recent results,” *J. Comput. Phys*., Vol. 169, 2001, pp. 463-502.

[9]Osher, S. and Fedkiw, R., *The Level Set Method and Dynamic Implicit Surfaces*, Springer-Verlag, New York, 2002.

[10]Sussman, M., Almgren, A., Bell, J., Colella, P., Howell, L. and Welcome, M., “An adaptive level set approach for incompressible two-phase flows,” *J. Comput. Phys*., Vol. 148, 1999, pp. 81-124.

[11]Strain, J., “Tree methods for moving interfaces,” *J. Comput. Phys*., Vol. 151, 1999, pp. 616-648.

[12]Yang, X., James, A.J., Lowengrub, J., Zheng, X. and Cristini, V., “An adaptive coupled level-set/volume-of- fluid interface capturing method for unstructured triangular grids,” *J. Comput. Phys*., Vol. 217, 2007, pp. 364-394.

[13]Enright, D., Marschner, S. and Fedkiw, R., “Animation and Rendering of Complex Water Surfaces,” *Proceedings of SIGGRAPH 2002,* ACM TOG, Vol. 21, 2002, pp. 736-744.

[14]Enright, D., Nguyen, D., Gibou, F. and Fedkiw, R., “Using the Particle Level Set Method and a Second Order Accurate Pressure Boundary Condition for Free Surface Flows,” *Proceedings of the 4th ASME-JSME Joint Fluids Eng. Conf., FEDSM2003-45144*,edited by M. Kawahashi and A. Ogut and Y. Tsuji, Honolulu, HI 2003, pp. 1-6.

[15]Gueyffier, D., Li, J., Nadim, A., Scardovelli, S. and Zaleski, S., “Volume of Fluid interface tracking with smoothed surface stress methods for three-dimensional flows,” *J. Comput. Phys*., Vol. 152, 1999, pp. 423-456.

[16]Yang, J. and Stern, F., “A Sharp Interface Method for Two-Phase Flows Interacting with Moving Bodies,” *18th AIAA Computational Fluid Dynamics Conference*, Miami, FL, 25 - 28 Jun 2007, AIAA paper 2007-4578.

[17]Yang, J. and Stern, F., “Large-Eddy Simulation of Breaking Waves Using Embedded-Boundary/Level-Set Method,” *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 8 - 11 January 2007, AIAA Paper 2007-1455.

[18]Sussman, M., Smereka, P. and Osher, S., “A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow,” *J. Comput. Phys*., Vol. 114, No. 1, 1994, pp. 146-159.

[19]Leonard, B.P., “A stable and accurate modeling procedure based on quadratic interpolation,” *Comput. Methods Appl. Mech. Engrg*., Vol. 19, 1979, pp. 58-98.

[20]Jiang, G.-S. and Peng, D., “Weighted ENO Schemes for Hamilton Jacobi Equations,” *SIAM J. Sci. Comp.*, Vol. 21, 2000 pp. 2126-2143.

[21] Aulisa, E., Manservisi, S., Scardovelli, R. and Zaleski, S., "Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry", *J. Comput. Phys*., Vol. 225, 2007, pp. 2301-2319.

[22] Scardovelli, R. and Zaleski, S., "Analytical relations connecting linear interfaces and volume fractions in rectangular grids," *J. Comput. Phys*., Vol. 164, 2000, pp. 228-237.

[23] Scardovelli, R. and Zaleski, S., "Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection," *Int. J. Numer. Meth. Fluids,* Vol. 41, 2003, pp. 251-274.

[24] Ghosh, S., Reins, G., Koo, B., Wang, Z., Yang, J. and Stern, F., "Plunging Wave Breaking: EFD and CFD," *International Conference on Violent Flows 2007*, 20-22 November 2007, Fukuoka, Japan.

[25] Peregrine, D. H., "Breaking Waves on Beaches", *Ann. Rev. of Fluid Mech*., Vol. 15, 1983, pp. 149-178

[26] Bonmarin, P., "Geometric properties of deep-water breaking waves", *J. Fluid Mech.,* Vol. 209, 1989, pp.405-433

[27] Tallent, J. R., Yamashita, T., and Tsuchiya, Y. "Transformation Characteristics of Breaking Waves", *Wave Water Kinematics*, Vol. 178, 1990, pp.509-523