Journal of Computational Physics 228 (2009) 6590–6616

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Sharp interface immersed-boundary/level-set method for wave–body interactions

Jianming Yang *, Frederick Stern

IIHR – Hydroscience and Engineering, University of Iowa, Iowa City, IA 52242, USA

ARTICLE INFO

Article history:
Received 22 January 2008
Received in revised form 6 May 2009
Accepted 30 May 2009
Available online 12 June 2009

PACS:
47.11.–j
47.27.–i
47.35.–i
47.55.–t

Keywords:
Cartesian grid method
Sharp interface method
Immersed boundary method
Level-set method
Ghost–fluid method
Large-eddy simulation
Ship hydrodynamics
Two-phase flow
Moving bodies

ABSTRACT

A sharp interface Cartesian grid method for the large-eddy simulation of two-phase turbulent flows interacting with moving bodies is presented. The overall approach uses a sharp interface immersed boundary formulation and a level-set/ghost–fluid method for solid–fluid and fluid–fluid interface treatments, respectively. A four-step fractional-step method is used for velocity–pressure coupling, and a Lagrangian dynamic Smagorinsky subgrid-scale model is adopted for large-eddy simulations. A simple contact angle boundary condition treatment that conforms to the immersed boundary formulation is developed. A variety of test cases of different scales ranging from bubble dynamics, water entry and exit, landslide-generated waves, to ship hydrodynamics are performed for validation. Extensions for high Reynolds number ship flows using wall-layer models are also considered.

1. Introduction

Incompressible two-phase flows interacting with moving bodies are encountered in many scientific and engineering applications, such as ship hydrodynamics, ocean and coastal engineering, and civil and environmental engineering. The computer simulations of these flows are extremely complicated as the intricate physical phenomena involved, e.g. turbulence, breaking waves, wave impacts, and wave–structure interactions, present significant challenges to the computational methods.

In general, computational methods for free-surface/two-phase flows with complex geometries and moving bodies can be categorized into three groups: meshless methods, moving grid methods, and fixed grid methods. In the meshless methods, discrete, moving particles are used to represent the continuous fluid motion. Due to the complexities of algorithms and requirements of computer resources, the meshless methods have only limited application, although there is an increased interest recently. In the moving grid methods, body-fitted grids are attached to the solid surfaces or even the interfaces between two different fluids, whereas the overall grids can be unstructured, block-structured, or overlapping grids. In

* Corresponding author. Tel.: +1 319 335 5749; fax: +1 319 335 5238.
E-mail addresses: jianming-yang@uiowa.edu (J. Yang), frederick-stern@uiowa.edu (F. Stern).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.05.047

general, these methods were applied to free-surface flows with single stationary or moving body/structure. Although significant progress has been made in moving grid methods, the grid deformation, re-generation and overlapping interpolation are still very time-consuming and prone to errors. On the other hand, in the fixed grid methods, solid boundaries and phase interfaces can have unrestricted motions across the underlying fixed grid lines, which are usually not aligned with the solid–fluid and/or the fluid–fluid interfaces. And, in most cases, Cartesian grids, which further simplify the gridding requirements, are used to cover the whole computational domain, although some techniques developed on fixed Cartesian grids have been applied to fixed curvilinear and unstructured grids.

Considering the interest of this study, only the computational methods using fixed Cartesian grids for free-surface/two-phase flows with moving bodies will be discussed here. Basically, there are two important aspects, besides the differences in solution methods for the Navier–Stokes equations, to distinguish different methods for free-surface/two-phase flows with moving bodies on fixed Cartesian grids: methods for tracking the evolving fluid–fluid and solid–fluid interfaces and methods for enforcing the jump/boundary conditions at the phase interface/boundary that is immersed in a Cartesian grid and does not align with the underlying fixed grid lines.

Tracking a moving interface has been one of the major topics since the very beginning of computational fluid dynamics (CFD). In general, there are two different types of methods from the point of view of how an interface is identified and advected. In the Lagrangian methods, massless markers, with or without connectivity, are placed on an interface and move with it by directly integrating the evolution equation of a fluid particle. The front tracking method [42] is one Lagrangian method frequently used in two-phase flow simulations. On the other hand, in Eulerian methods, a field function with the interface information embedded is advected in an Eulerian manner. Methods of this type are usually called volume tracking or interface capturing methods as the position of the interface is not known exactly from the field function. The volume-of-fluid (VOF) method [15] and the level-set method [33] are two popular Eulerian methods for interface capturing. There are various pros and cons of each method for tracking an interface. Therefore, many hybrid interface tracking schemes have been developed to combine the advantages of different methods.

Once an immersed interface/boundary is identified using an interface tracking/capturing method, the next step is to enforce the jump/boundary conditions at the immersed interface/boundary, which is trivial for an interface- or body-fitted grid method as the grid lines are made to follow the phase interface. Therefore, a perceptible choice is to separate a grid cell cut through by the interface into, e.g. two grid cells of two different phases, such that essentially two interface- or body-fitted grids are formed with each grid for one single phase. Methods of this type are usually named cut-cell or embedded boundary methods, which have been applied to both solid–fluid and fluid–fluid interface treatments. Although the cut-cell methods are sharp interface approaches, they are very difficult to apply to three-dimensional (3D) problems, especially, when moving interfaces/boundaries are involved, due to the large number of special treatments required by the numerous different interface cells generated during the cutting and merging processes. On the other hand, in many circumstances the sharp interface requirement can be relaxed and an interface is allowed to have a finite thickness, usually, of one to several grid cells. The immersed boundary method originated by Peskin [35] is one of the widely used Cartesian grid methods with a smeared interface. In Peskin's method, the effects of solid/elastic boundaries on the fluid are modeled by a set of body forces distributed over the nearby flow field of the immersed boundaries. This idea has been extended to treat the surface tension force in two-phase flows in [42,4]. In the past few years, several sharp interface methods using forcing or correction terms have been developed for solid–fluid and fluid–fluid interface problems, such as the direct forcing immersed boundary method [9], and the immersed interface method [24], among others. In Fadlun et al. [9] the concept of body forcing was also adopted, however, in a discrete manner and a sharp interface was retained. The ghost–fluid method, which is conceptually similar to the immersed interface method but neglects the second-order derivatives in the jump conditions and is used in a dimension-by-dimension manner, was developed in [20] for the sharp interface treatment of a fluid–fluid interface. Both the direct forcing immersed boundary method and the ghost–fluid method have gained increased popularity because of their increased accuracy over the diffusive interface methods and their straightforward implementation in 3D applications.

Some computational methods have been developed for free-surface flows with a moving body as the gas phase plays a minor role and can be neglected without substantially altering the flow physics in many applications. In [22], a finite volume Cartesian grid method was developed for wave impact problems. Lin [25] also developed a finite difference Cartesian grid method for free-surface flows with a moving body. In both methods, Lagrangian methods were used to track the solid bodies and VOF methods were used to track the free surfaces. The so-called free-surface boundary conditions, i.e. atmospheric pressure is given and shear stress is neglected, were applied at the free surfaces; cut-cell methods were used to treat the solid–fluid interface. Especially, in [25], a "locally relative stationary" technique was developed for accounting for the effect of body motion. Various water entry/impact problems and other cases were shown in [22,25]. However, neither of the two methods were applied to free-surface flows with 3D moving bodies. A sharp interface Cartesian grid method for droplet impact problems was developed in [27], with both the fluid–fluid and solid–fluid interfaces defined using level-set functions. The ghost–fluid method was used to treat jump conditions at the fluid–fluid interfaces. The solid–fluid interface treatment was developed along the line of cut-cell methods, using a one-side finite difference scheme to avoid the cumbersome cell cutting and merging processes in the finite volume framework. But only 2D droplet impact problems with stationary geometry were shown in [27]. The CIP (cubic-interpolated propagation/constrained interpolation profile) method [47] is another unified treatment that track a solid–fluid interface in the same way as that for a fluid–fluid interface. In the CIP method, a color function with a definition similar to the volume fraction function in the VOF method was transformed to a smoothed profile using a tangent function, then the smoothed function was advected using the CIP schemes, and finally the smoothed function was inverted to give a sharper profile. Similar to Peskin's immersed boundary method, the fluid–fluid and solid–fluid interfaces are diffused. But instead of the smoothed delta function in Peskin's method, the color function was used to smear the interfaces and interface jump/boundary conditions. In addition, to reflect the presence of the solid phase in the flow field, the solid phase was treated in the same way as the fluid phase, but with the velocity of the solid phase obtained from prescribed values or predicted solutions by solving the governing equations for the motion of the solid phase. The CIP method has been applied to a wide range of 2D and 3D two-phase flows with moving solid bodies. However, it is memory- and CPU-intensive as the derivatives of the primitive variables are stored and solved. The algorithm is fairly complicated due to the extra equations introduced. Also, a solid body sometimes was distorted and the sharp solid–fluid interface was smeared out. For some variants and applications of the CIP method, the reader is referred to [54,48].

An additional consideration for simulating gas–liquid–solid systems is the contact angle boundary condition. In [40], an iterative method similar to the level-set reinitialization equation was developed by adding a time derivative term to the contact angle boundary condition and constructing an extension velocity field for the level-set function. By solving such an equation in pseudo-time iteratively, the level-set field was extended into the solid bodies and the contact angle boundary condition was satisfied. In [27] a local level-set reconstruction method was presented for enforcing the dynamic contact angle boundary condition. A local 2D parabola was established by searching and identifying two grid points near the contact points and then the level-set function was extended into the solid body following the parabola. A virtual surface method was proposed in [44] to enforce contact angles at the intersection of a fluid interface and a general 3D curved solid surface. In this approach, a local stencil box centered near the contact front was setup and the virtual surface was constructed as a level-set function in the solid region. This level-set function is computed at the stencil nodes directly and then propagated to off-stencil nodes via a fast marching method.

In this paper, a combined sharp interface immersed-boundary/level-set Cartesian grid method is developed for the large-eddy simulations (LES) of 3D two-phase flows interacting with moving bodies. In the current approach, a Lagrangian method is used to track the solid–fluid interface for accuracy and efficiency. The second-order direct forcing, sharp interface immersed boundary method for moving boundary problems in [49] is used to treat the moving bodies. The fluid–fluid interface is captured using a high-order level-set method with third-order TVD (Total Variation Diminishing) Runge–Kutta and fifth-order HJ (Hamilton–Jacobi) WENO (Weighted Essentially Non-Oscillatory) schemes. The ghost–fluid method [26,20] is used to handle fluid–fluid interfaces in a sharp manner. In addition, a simple contact angle boundary condition treatment is developed. The applications of interest herein are the complicated flow problems in many engineering fields, especially, ship hydrodynamics, with a wide spectrum of scales, such as wave–body interactions, wave–wave interactions, breaking waves, bubbles and droplets. Therefore, a variety of test cases of different scales ranging from bubble dynamics, water entry and exit, landslide-generated waves, to ship hydrodynamics are performed for validation purpose in this study.

For ship wave problems, several Cartesian grid methods were presented in the literature. In [32], a modified Marker and Cell method (TUMMAC) was developed for the finite difference solution of non-linear wave generation in the near field of ships. A Cartesian grid approach was presented in [40] with a coupled level-set/volume-of-fluid method for interface capturing and an embedded boundary method for the immersed geometry. Another approach using immersed-body and volume-of-fluid methods was also shown in [40] and its latest development was given in [8] for ship waves simulations. The current method is different from these methods in many aspects as will be shown in detail later. For example, it uses a second-order sharp interface immersed boundary method for two-phase flows with multiple moving bodies on fixed Cartesian grids. Although free-surface flows with a moving body were shown in [8], a non-inertial reference frame fixed with respect to the moving body, which is unable to handle multiple moving bodies. In addition, a ghost–fluid method is used in the current approach without smearing the density across the fluid–fluid interface.

2. Mathematical model

2.1. Navier–Stokes equations

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

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

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

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

$$\mathbf{T} = 2\mu \mathbf{S}, \tag{3}$$

with μ the dynamic viscosity, $\mathbf{S}$ the strain rate tensor given by

$$\mathbf{S} = \frac{1}{2}\left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right], \tag{4}$$

where the superscript T represents transpose operation.

2.2. Interface modeling

2.2.1. Interface tracking

Defining the interface Γ as the zero level set of a signed distance function, ϕ, or the level-set function, the position of the interface can be tracked by solving the level-set evolution equation [33]

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{5}$$

The reinitialization equation [39] for the level-set function is iteratively solved to keep ϕ as a signed distance function in the course of its evolution:

$$\frac{\partial \phi}{\partial \tau} + S(\phi_o)(|\nabla \phi| - 1) = 0, \tag{6}$$

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

$$S(\phi_o) = \frac{\phi_o}{\sqrt{\phi_o^2 + (\Delta h)^2}}, \tag{7}$$

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

Since the level-set function is a signed distance function, the unit normal vector, $\mathbf{n}$, and the local curvature, κ, of the interface can be readily calculated by applying standard finite difference to the level-set function:

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \tag{8}$$

and

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

respectively.

2.2.2. Physical properties

Each phase of constant density and viscosity can be easily defined by the level-set function in the computational domain and sharp jumps of the fluid properties occur at the phase interface. In the ghost–fluid method for incompressible two-phase flows [20], the discontinuous fluid properties were considered and the interface jump conditions were tackled in a sharp manner. Unfortunately, the viscous terms were treated explicitly in [20], which imposed a severe time step limitation to the simulations. And due to the coupled manner of the interface jump conditions, an implicit treatment of the viscous terms requires the solution of the entire coupled system of three velocity components [16]. Sussman et al. [41] presented a semi-implicit ghost–fluid method with a sharp viscosity jump, but neglected viscous flux jumps. Several authors chose to regularize the viscosity with a smoothed Heaviside function across the interface [43,11]. For simplicity and efficiency, the density keeps its sharp jump, whereas the viscosity is smoothed over a transition band across the interface in this study:

$$\begin{aligned} \rho &= \rho_G + (\rho_L - \rho_G)H(\phi), \\ \mu &= \mu_G + (\mu_L - \mu_G)H_\epsilon(\phi), \end{aligned} \tag{10}$$

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

$$H(\phi) = \begin{cases} 1, & \text{if } \phi \geqslant 0, \\ 0, & \text{if } \phi < 0, \end{cases} \tag{11}$$

and the smoothed Heaviside function [39] is

$$H_\varepsilon(\phi) = \begin{cases} 1, & \text{if } \phi > \varepsilon, \\ \frac{1}{2}\left[1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\varepsilon}\right)\right], & \text{if } |\phi| \leqslant \varepsilon, \\ 0, & \text{if } \phi < -\varepsilon. \end{cases} \tag{12}$$

2.2.3. Jump conditions

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

$$[\mathbf{u}] = 0, \tag{13}$$

where $[\cdot]$ indicates the jump at the interface, i.e. $f_L^I - f_G^I$ for a variable f with superscript I denotes interface.

The exact jump condition for stress is

$$[\mathbf{n}\cdot(-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\cdot\mathbf{n}]=\sigma\kappa, \tag{14}$$

where σ is the coefficient of surface tension.

The gravity term can be removed from Eq. (1) by incorporating the gravity into the jump condition as

$$[\mathbf{n}\cdot(-p_d\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\cdot\mathbf{n}]=\sigma\kappa+[\rho]\mathbf{X}\cdot\mathbf{g}, \tag{15}$$

where p_d represents the dynamic pressure (for simplicity, p is used hereafter), $[\rho]$ is the density jump at the interface, and $\mathbf{X}$ is the position vector normal to the reference plane of zero hydrostatic pressure.

With a continuous viscosity and velocity field, the stress jump conditions Eq. (14) reduce to

$$[p]=p_L^I-p_G^I=-\sigma\kappa-[\rho]\mathbf{X}\cdot\mathbf{g}. \tag{16}$$

2.3. Subgrid-scale model

In the LES approach adopted here, the Navier–Stokes equations are spatially filtered such that the large, energy carrying eddies are resolved and the small-scale, dissipative eddies are modeled by a subgrid-scale model. The following equations can be obtained after applied the filter operation to Eqs. (1) and (2):

$$\frac{\partial\bar{\mathbf{u}}}{\partial t}+\bar{\mathbf{u}}\cdot\nabla\bar{\mathbf{u}}=-\frac{1}{\rho}\nabla\bar{p}+\frac{1}{\rho}\nabla\cdot[\mu(\nabla\bar{\mathbf{u}}+(\nabla\bar{\mathbf{u}})^T)]-\nabla\cdot\bar{\tau}, \tag{17}$$

$$\nabla\cdot\bar{\mathbf{u}}=0, \tag{18}$$

where $\bar{f}$ denotes the filter operation on a variable $f, \bar{\tau}=\overline{\mathbf{u}\mathbf{u}}-\bar{\mathbf{u}}\bar{\mathbf{u}}$ is the subgrid-scale (SGS) stress tensor, whose deviatoric part is parametrized by following the Smagorinsky procedure as:

$$\bar{\tau}-\frac{1}{3}\text{trace}(\bar{\tau})\mathbf{I}=-2\nu_t\bar{\mathbf{S}}. \tag{19}$$

And the turbulent eddy viscosity is defined as

$$\nu_t=C\Delta^2|\bar{\mathbf{S}}|,\quad\text{and}\quad|\bar{\mathbf{S}}|=\sqrt{2\bar{\mathbf{S}}\cdot\bar{\mathbf{S}}}. \tag{20}$$

The model parameter C in the eddy viscosity definition has to be given to close the equations. In this paper the Lagrangian dynamic SGS model [31] is chosen as it can handle complex geometries without the requirement of homogeneous direction(s). Therefore, Eq. (17) can be rewritten as the following form:

$$\frac{\partial\bar{\mathbf{u}}}{\partial t}+\bar{\mathbf{u}}\cdot\nabla\bar{\mathbf{u}}=-\frac{1}{\rho}\nabla\bar{p}+\frac{1}{\rho}\nabla\cdot[\mu(\nabla\bar{\mathbf{u}}+(\nabla\bar{\mathbf{u}})^T)]+\nabla\cdot[\nu_t(\nabla\bar{\mathbf{u}}+(\nabla\bar{\mathbf{u}})^T)], \tag{21}$$

with the trace of SGS stress tensor $\frac{1}{3}\text{trace}(\bar{\tau})$ (and gravity as discussed in Section 2.2.3) incorporated into $\bar{p}$. Also, an effective total viscosity cannot be defined using $\mu+\rho\nu_t$ since ρ is discontinuous across the interface in the present sharp interface treatment. For simplicity, hereafter the bar signs are dropped and $\mathbf{u}$ and p are used for $\bar{\mathbf{u}}$ and $\bar{p}$, respectively, even SGS model is applied.

3. Numerical method

3.1. Navier–Stokes solver

A finite difference method is used to discretize the Navier–Stokes equations on a non-uniform staggered Cartesian grid, in which the velocity components u, v, and w are defined at centers of cell faces in the x, y, and z directions, respectively, and all other variables, i.e. p, ϕ, ρ, μ, and ν_t are defined at cell centers. Fig. 1 shows the staggered arrangement of the variables on a 2D x–y grid.

3.1.1. Fractional-step method

A four-step fractional-step method [6] is employed for velocity–pressure coupling, in which a pressure Poisson equation is solved to enforce the continuity equation. For time advancement, a second-order semi-implicit 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. The four-step algorithm can be written as follows:

(1) Predictor:

$$\frac{\hat{u}_i-u_i^n}{\Delta t}=\frac{1}{2}[3A_i^n-A_i^{n-1}]+\frac{1}{2}\left[C_i^{n+1}+C_i^n\right]-\text{Grad}_i(p^n), \tag{22}$$

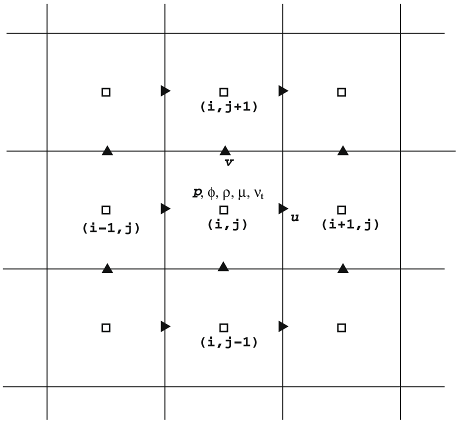


Fig. 1. Arrangement of variables on a staggered Cartesian grid.

(2) First corrector:

$$\frac{u_i^* - \hat{u}_i}{\Delta t} = \mathrm{Grad}_i(p^n), \tag{23}$$

(3) Pressure Poisson equation:

$$\frac{\partial}{\partial x_i}\mathrm{Grad}_i(p^{n+1}) = \frac{1}{\Delta t}\frac{\partial u_i^*}{\partial x_i}, \tag{24}$$

(4) Second corrector:

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = -\mathrm{Grad}_i(p^{n+1}), \tag{25}$$

where superscript n denotes time step, subscript $i = 1, 2, 3$ represents i-coordinate, $\hat{u}$ and u^* are the first and second intermediate velocities, respectively. $\mathrm{Grad}_i(p)$ is a pressure gradient term defined at the center of a cell face (collocated with the ith velocity component) with the jump conditions due to surface tension and gravity incorporated. More details about the definition and calculation of $\mathrm{Grad}_i(p)$ will be discussed in Section 3.1.3. **A** and **C** denote terms treated by the Adams–Bashforth and Crank–Nicolson schemes, respectively, i.e.

$$\mathbf{A} = -\bar{\mathbf{u}} \cdot \nabla\bar{\mathbf{u}} + \frac{1}{\rho}\nabla \cdot [\mu(\nabla\bar{\mathbf{u}})^T] + \nabla \cdot [\nu_t(\nabla\bar{\mathbf{u}})^T], \tag{26}$$

and

$$\mathbf{C} = \frac{1}{\rho}\nabla \cdot [\mu(\nabla\bar{\mathbf{u}})] + \nabla \cdot [\nu_t(\nabla\bar{\mathbf{u}})]. \tag{27}$$

The above four-step fractional-step method does not require special treatment of boundary conditions for the intermediate velocity to obtain an overall second-order splitting. The pressure correction methods are not applicable here because the density and pressure are treated in a sharp manner and the following equality cannot be applied directly

$$\frac{1}{\rho}\nabla p^{n+1} = \frac{1}{\rho}\nabla p^n + \frac{1}{\rho}\nabla\psi, \tag{28}$$

with ψ for the pressure correction or increment.

3.1.2. Momentum solver

To solve Eq. (22), the convection, diffusion, and pressure gradient terms have to be discretized. Note that the surface tension and gravity do not appear in Eq. (22) explicitly since it enters the system through the pressure jump condition. In this study, the diffusion terms are discretized with the standard second-order central difference scheme, and pressure gradient will be discussed in Section 3.1.3. An arithmetic mean is used to obtain values of velocity components, density, and viscosity at locations where these values are not defined from the neighboring collocation points with the exception of the interfacial density as defined in Section 3.1.3.

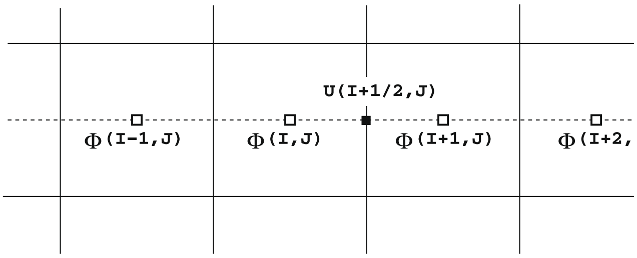


Fig. 2. Schematic of the QUICK scheme for the convection terms.

For the convective terms, a third-order QUICK (Quadratic Upwind Interpolation for Convective Kinematics) scheme [23] is used, whereas central differencing can be applied when there is enough grid resolution. As shown in Fig. 2, Φ represents a general velocity component and a different index system (I,J) than that in Fig. 1 is used to indicate the staggered variable arrangement. Using $\nabla_x(u\Phi)$ as an example, the discretization can be written as

$$\nabla_x(u\Phi) = \frac{1}{\Delta x}(U_{I+1/2,J}\Phi_{I+1/2,J} - U_{I-1/2,J}\Phi_{I-1/2,J}), \tag{29}$$

where U means a cell face advecting u velocity component from an arithmetic mean. With an upwind procedure, the right side cell face flux, $\Phi_{I+1/2,J}$, can be evaluated using the QUICK scheme as follows:

$$\Phi_{I+1/2,J} = \begin{cases} \frac{1}{8}(-\Phi_{I-1,J} + 6\Phi_{I,J} + 3\Phi_{I+1,J}), & \text{if } U_{I+1/2,J} \geqslant 0, \\ \frac{1}{8}(-\Phi_{I+2,J} + 6\Phi_{I+1,J} + 3\Phi_{I,J}), & \text{if } U_{I+1/2,J} < 0 \end{cases} \tag{30}$$

on a uniform grid. Lagrange polynomial interpolation can be used to obtained the non-constant coefficients in the above equation for non-uniform grids.

To invert the momentum equations due to the implicit treatment of the diagonal viscous terms, the approximate factorization method [3] is used. Eq. (22) can be rewritten in the following form (for illustration purpose, a uniform grid is used and terms due to SGS stresses are not included):

$$\left[1 - \frac{\Delta t}{\rho}\nabla\cdot(\mu\nabla)\right]\hat{u}_i = u_i^n + \Delta t(\mathrm{RHS})_i^n, \tag{31}$$

where $(\mathrm{RHS})_i^n$ includes all terms evaluated explicitly in Eq. (22). With the approximate factorization scheme, the above equation can be expressed as

$$\left[1 - \frac{\Delta t}{\rho}\frac{\partial}{\partial x}\left(\mu\frac{\partial}{\partial x}\right)\right]\left[1 - \frac{\Delta t}{\rho}\frac{\partial}{\partial y}\left(\mu\frac{\partial}{\partial y}\right)\right]\left[1 - \frac{\Delta t}{\rho}\frac{\partial}{\partial z}\left(\mu\frac{\partial}{\partial z}\right)\right]\hat{u}_i = u_i^n + \Delta t(\mathrm{RHS})_i^n. \tag{32}$$

A splitting error of order $O(\Delta t^3)$ is introduced into the system in this factorization procedure, which does not affect the second-order temporal accuracy of the overall algorithm.

Eq. (32) can be inverted by solving three tridiagonal linear equations. Due to the domain decomposition for parallelization, the matrix of each tridiagonal system is distributed over a number of processors. Iterative solvers can be used by solving the sub-system in each block and then updating the ghost cells. However, the performance is not optimal and an extra convergence error is introduced. In this study, the parallel tridiagonal system solver given in [30] has been implemented and proved to be efficient and robust. In this solver, two tridiagonal systems are inverted in each grid block, one for the sub-system residing in this block and one for the coupling of all blocks. The results from the two systems are combined to give the final solution with minimized number of communications between the grid blocks.

3.1.3. Jump conditions treatment

As shown in Fig. 3, the following definition (see [41] for a similar one) for the pressure gradient in the x direction can be used to implement the jump conditions given in Eq. (16):

$$\mathrm{Grad}_x(p)_{i+1/2,j} = \frac{1}{\hat{\rho}_{i+1/2,j}}\frac{(p_{i+1,j} - [p]H_{i+1,j}) - (p_{i,j} - [p]H_{i,j})}{\Delta x} = \frac{1}{\hat{\rho}_{i+1/2,j}}\frac{(p_{i+1,j} - p_{i,j}) - [p](H_{i+1,j} - H_{i,j})}{\Delta x}, \tag{33}$$

where H is the Heaviside function defined in Eq. (11), the pressure jump across the interface between cell centers (i,j) and $(i+1,j)$ is

$$[p] = -\sigma\kappa^I - [\rho](x^I g_x + y^I g_y + z^I g_z), \tag{34}$$

with (x^I, y^I, z^I) the interface position vector normal to the reference piezometric plane. The cell face density is defined as

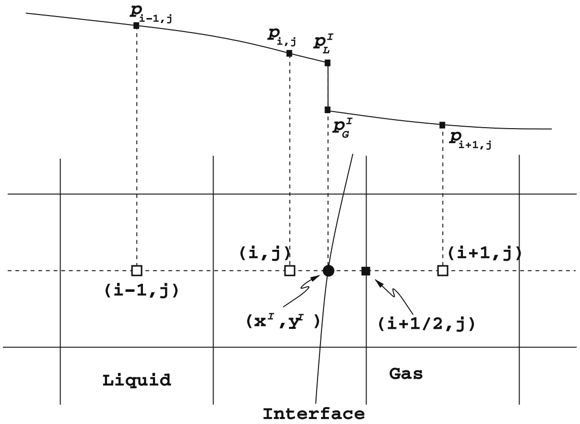


Fig. 3. Schematic of the jump condition treatment for the case $H_{i,j} = 1$ and $H_{i+1,j} = 0$.

$$\hat{\rho}_{i+1/2,j} = \rho_L\theta_{i+1/2,j} + \rho_G(1 - \theta_{i+1/2,j}), \tag{35}$$

with

$$\theta_{i+1/2,j} = \begin{cases} 1, & \text{if } \phi_{i,j} \geqslant 0 \quad \text{and} \quad \phi_{i+1,j} \geqslant 0, \\ 0, & \text{if } \phi_{i,j} < 0 \quad \text{and} \quad \phi_{i+1,j} < 0, \\ \frac{\phi_{i,j}}{|\phi_{i,j}|+|\phi_{i+1,j}|}, & \text{if } \phi_{i,j} \geqslant 0 \quad \text{and} \quad \phi_{i+1,j} < 0, \\ \frac{\phi_{i+1,j}}{|\phi_{i,j}|+|\phi_{i+1,j}|}, & \text{if } \phi_{i,j} < 0 \quad \text{and} \quad \phi_{i+1,j} \geqslant 0. \end{cases} \tag{36}$$

It is evident from the definition of the Heaviside function H that Eqs. (33) and (35) recover their usual one-fluid forms when the cell centers (i,j) and $(i+1,j)$ are in the same phase, i.e. either air or water.

3.1.4. Poisson solver

The pressure Poisson equation, Eq. (24), is discretized using a standard second-order central-difference scheme. On a staggered grid, the right-hand side of Eq. (24) can be computed readily using the cell face velocity components. The left-hand side can be discretized by applying the divergence operator to Eq. (33) [26]. Again, using the x direction and a uniform grid as an example,

$$\frac{1}{\Delta x}\mathrm{Grad}_x(p^{n+1})_{i+\frac{1}{2}j} = \frac{1}{\Delta x}\frac{1}{\hat{\rho}_{i+1/2}}\frac{\left(p^{n+1}_{i+1,j} - p^{n+1}_{i,j}\right) - [p](H_{i+1,j} - H_{i,j})}{\Delta x} = \frac{1}{(\Delta x)^2}\frac{p^{n+1}_{i+1,j} - p^{n+1}_{i,j}}{\hat{\rho}_{i+1/2,j}} - \frac{[p]}{(\Delta x)^2}\frac{H_{i+1,j} - H_{i,j}}{\hat{\rho}_{i+1/2,j}}, \tag{37}$$

where the second term due to the pressure jump condition is moved to the right-hand side of the pressure Poisson equation. Since the interface is advected before the Poisson solver as will be discussed in Section 3.6, the position and curvature of the interface and then the pressure jumps due to the gravity and surface tension can be computed in advance. Therefore, re-evaluation of the right-hand side is not necessary and standard black-box algebraic system solver can be used to solve the Poisson equation.

In this study, the pressure Poisson equation is solved using a Krylov-based multigrid solver from the PETSc library [2] with the GMRES method as the smoother. Usually, one multigrid cycle can reduce the residual norm by one order of magnitude. The solver can converge to a relative residual norm of 10^{-6} in around ten iterations on a severely stretched grid even for a density ratio of 1:1000. In general, this is the most expensive part of the whole algorithm.

3.2. Level-set solver

The local (narrow band) level-set method by Peng et al. [34] is used to solve the level-set evolution and reinitialization equations. In this approach, a narrow band of about a few grid cell widths around the zero level set is setup at each time step using a mask function. The mask function identifies grid points where the evolution equation is solved; then the level-set function is reinitialized at all these points and their neighboring points. With the use of a narrow band method, the additional solution of the level-set function with the current high-order schemes does not pose a significant overhead to the Navier–Stokes solver.

A fifth-order HJ WENO scheme [19] is used for the spatial discretization of the level-set evolution and reinitialization equations. In Eq. (5) the term $\mathbf{u} \cdot \nabla\phi$ has to be approximated at the cell center. Again, the x component of the advection term is used here to illustrate the treatment. With an upwind scheme

$$(\phi_x)_{i,j} = \begin{cases} (\phi_x^-)_{i,j}, & \text{if } u_{i,j} \geqslant 0, \\ (\phi_x^+)_{i,j}, & \text{if } u_{i,j} < 0, \end{cases} \tag{38}$$

the upwinding derivatives $(\phi_x^-)_{i,j}$ and $(\phi_x^+)_{i,j}$ are computed using a subset of $(\phi_{i-3,j}, \phi_{i-2,j}, \phi_{i-1,j}, \phi_{i,j}, \phi_{i+1,j}, \phi_{i+2,j}, \phi_{i+3,j})$ by following the HJ WENO procedure given in [19]. Note on a staggered grid as shown in Fig. 1, the velocity components are not available at the cell centers and an arithmetic mean is used to calculate the velocity components where needed.

The reinitialization equation can be rewritten as

$$\frac{\partial \phi}{\partial \tau} + \mathbf{u}_{ext} \cdot \nabla \phi - S(\phi_o) = 0, \tag{39}$$

where the extension velocity is

$$\mathbf{u}_{ext} = S(\phi_o) \frac{\nabla \phi}{|\nabla \phi|}. \tag{40}$$

Then the advection term in Eq. (39) can be evaluated similarly as done in the evolution equation.

The level-set evolution and reinitialization equations are solved using third-order TVD Runge–Kutta scheme [37] for time advancement. Both equations can be written as the following ODE:

$$\frac{d\phi}{dt} = L(\phi). \tag{41}$$

Then the third-order TVD Runge–Kutta scheme for Eq. (41) is

$$\begin{aligned} \phi^{(1)} &= \phi^{(0)} + \Delta t L(\phi^{(0)}), \\ \phi^{(2)} &= \frac{3}{4}\phi^{(0)} + \frac{1}{4}\Delta t[\phi^{(1)} + L(\phi^{(1)})], \\ \phi^{(3)} &= \frac{1}{3}\phi^{(0)} + \frac{2}{3}\Delta t[\phi^{(2)} + L(\phi^{(2)})], \end{aligned} \tag{42}$$

with $\phi^{(0)} = \phi^n$ and $\phi^{n+1} = \phi^{(3)}$. As to be discussed in Section 3.5, a CFL (Courant–Friedrichs–Lewy) number of 0.5 with all constraints including advecting velocity, gravity, and surface tension is used for the Navier–Stokes solver, which means an even smaller CFL number will be used for Eq. (41) and there is no stability issues for using the above scheme.

3.3. Immersed boundary treatment

The sharp interface immersed boundary formulation by Yang and Balaras [49] is used to treat the immersed moving boundaries/bodies on a non-uniform Cartesian grid. In this approach, the grid generation for complex geometries is trivial as the requirement that the grid points coincide with the boundary, which is imperative for body-fitted methods, is relaxed; whereas the solution near the immersed boundary is reconstructed using momentum forcing in a sharp manner. The detailed procedure is given in [49] and summarized here.

The first step is to establish the grid-interface relation with a given immersed boundary description, such as parametrized curve/surface or triangulation. In this step all Cartesian grid points are split into the three categories shown in Fig. 4: (1): *fluid points*, which are points in the fluid phase; (2) *forcing points*, which are grid points in the fluid phase with one or more neighboring points in the solid phase; and (3) *solid points*, which are points in the solid phase. The Navier–Stokes solver described in the previous section is applied to all grid nodes without distinguishing fluid, forcing, or solid points. A discrete forcing function, f_i, is introduced to the momentum equation to mimic the effect of immersed boundary on the flow field:

$$\frac{\hat{u}_i - u_i^n}{\Delta t} = \frac{1}{2}\left[3A_i^n - A_i^{n-1}\right] + \frac{1}{2}\left[C_i^{n+1} + C_i^n\right] - \mathrm{Grad}_i(p^n) + f_i^{n+\frac{1}{2}}. \tag{43}$$

For the semi-implicit time advancement scheme used here, a provisional velocity field $\tilde{u}_i$ for $\hat{u}_i$ is solved first with all terms treated by the Crank–Nicolson scheme in Eq. (22) using the explicit forward Euler scheme; then the forcing function in Eq. (43) can be evaluated in a straightforward manner by substituting $\tilde{u}_i$ with u_f [21]:

$$f_i^{n+\frac{1}{2}} = \frac{u_f - u_i^n}{\Delta t} - \frac{1}{2}\left[3A_i^n - A_i^{n-1}\right] - \frac{1}{2}\left[C_i^n + C_i^n\right] + \mathrm{Grad}_i(p^n). \tag{44}$$

If the interface and the forcing point coincide, u_f is just the velocity boundary condition one wants to apply and the local velocity of the immersed body that is already known from prescribed or predicted motion. However, usually the points on the fixed Cartesian grid and the moving interface never coincide and u_f has to be computed using the already known information with various interpolation schemes. In this study, the linear interpolation scheme given by [49] is used and an example is shown in Fig. 4. An interpolation stencil is setup for a forcing point by three points: the projection of the forcing point on the interface (point 1 in Fig. 4) and two neighbor grid points in the fluid phase (points 2 and 3 in Fig. 4). As discussed above, the local velocity of the immersed body (point 1) is known and the provisional velocity field $\tilde{u}_i$ is used

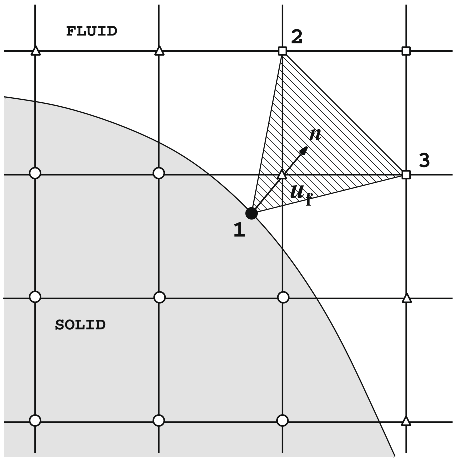


Fig. 4. Grid-interface relation and the interpolation stencil for u_f (point 1, 2, and 3). ○ *solid points*; □ *fluid points*; △ *forcing points.*

for points 2 and 3. In addition, solid body forcing is imposed on all solid points such that at a specific location a solid point is given the same velocity of the solid body. The above procedure has been extensively tested for a variety of laminar and turbulent flow problems involving complex moving immersed boundaries and fully coupled fluid–structure interactions with results in excellent agreement with reference computations and experiments [1,49,50]. Another advantage of this method is that the basic flow solver keeps intact as a discrete forcing field is applied to the flow field. Therefore, it is much easier to implement a direct forcing immersed boundary method as in [49] than the cut-cell methods in [22,27], especially, for 3D problems.

On the other hand, the velocity boundary conditions at the immersed boundary could be violated as a result of imposing the boundary conditions on the intermediate velocity field instead of the final one which satisfies the incompressibility constraint. This issue has been carefully examined and discussed by Fadlun et al. [9]. They also proposed an iterative approach to reduce the errors to round-off error, but the additional computational cost was considered unjustified by the indistinguishable difference between the results from the one-step and the iterative approaches. The former approach is adopted in this study and we found the errors at the immersed boundaries due to the above problem are similar to those in [9] (around the order of $10^{-3} \sim 10^{-4}$), which are quite small and acceptable for the applications we are interested in here.

3.4. Contact angle boundary condition

The contact angle boundary condition on the immersed boundary is implemented by following an interpolation strategy similar to the idea discussed above. As shown in Fig. 5, the contact angle boundary condition can be written as

$$\mathbf{n}_{ls} \cdot \mathbf{n}_{ib} = |\mathbf{n}_{ls}||\mathbf{n}_{ib}|cos(\pi - \alpha) = -cos(\alpha), \tag{45}$$

where $\mathbf{n}_{ls}$ is the normal of level-set function at the contact point, which is calculated using Eq. (8), $\mathbf{n}_{ib}$ is the outward normal of the immersed boundary at the contact point, which is well-defined as a basic element of the current immersed boundary method, and α is the contact angle between the interface and the immersed wall.

The following equations can be obtained by substituting $\mathbf{n}_{ls}$ in Eq. (45) with Eq. (8):

$$\frac{\nabla\phi}{|\nabla\phi|} \cdot \mathbf{n}_{ib} = \frac{1}{|\nabla\phi|}\left(\frac{\partial\phi}{\partial n}\right)_{ib} = -cos(\alpha), \tag{46}$$

or,

$$\left(\frac{\partial\phi}{\partial n}\right)_{ib} = -cos(\alpha)|\nabla\phi| = -cos(\alpha), \tag{47}$$

as ϕ is a signed distance function and $|\nabla\phi| = 1$.

Therefore, with the normal gradient of level set available, the contact angle boundary condition can be readily specified using the sharp interface immersed boundary method in [49]. In the 2D space the level-set function ϕ can be approximated as follows:

$$\phi = b_1 + b_2 x + b_3 y. \tag{48}$$

It is evident from Eq. (46) that the normal derivative of level-set function at the immersed boundary can be expressed as

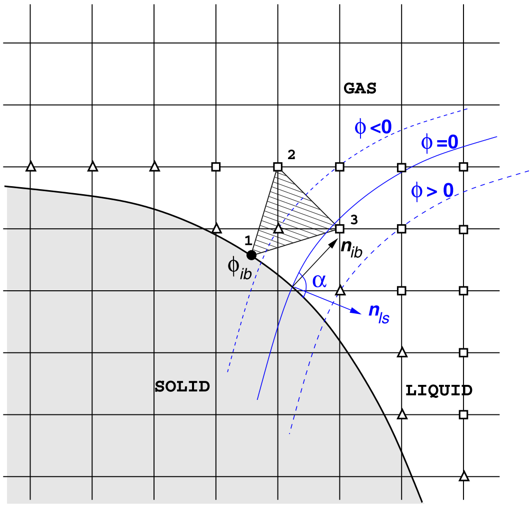


Fig. 5. Contact angle boundary condition.

$$\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial x} n_x + \frac{\partial \phi}{\partial y} n_y, \tag{49}$$

where n_x and n_y are the x and y components of the unit normal vector $\mathbf{n}_{ib}$, respectively. Subscript *ib* is dropped for simplicity in above equation. The x and y derivatives of ϕ can be easily computed from Eq. (48). Therefore, the coefficients b_1, b_2, and b_3 in Eq. (48) can be found by solving the following system:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \mathbf{A}^{-1} \begin{vmatrix} \frac{\partial \phi}{\partial n} \\ \phi_2 \\ \phi_3 \end{vmatrix} = \begin{bmatrix} 0 & n_x & n_y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \phi}{\partial n} \\ \phi_2 \\ \phi_3 \end{bmatrix}, \tag{50}$$

where (x_2, y_2) and (x_3, y_3) in the 3×3 matrix $\mathbf{A}$ are the coordinates of the two fluid points in the interpolation stencil shown in Fig. 5. The inversion of matrix $\mathbf{A}$ at every forcing point is performed every time the location of the immersed boundary is updated.

In this paper, a fixed contact angle $\alpha = 90^\circ$ is used for all cases. However, other different fixed or even dynamic contact angles can be implemented by the above scheme in a straightforward manner. A field extension procedure to extend the level-set function into the solid bodies is also applied.

The current approach for implementing the contact angle boundary condition is surprisingly simple and highly consistent with the overall immersed boundary method. As demonstrated in the results part, this approach is effective for both 2D and 3D problems and is also accurate. Comparing to the iterative method in [40], the 2D local fitted method in [27], and the local stencil box construction method in [44], this approach has some remarkable advantages as discussed above. It enables the extension of the sharp interface immersed boundary method for fluid–solid interaction problems to gas–liquid–solid interaction problems in a prompt and natural way.

However, due to the direct explicit modification of the level-set function at the forcing points, the volume fractions of different phases might be slightly changed near the immersed boundary. An additional constraint for volume conservation has to be imposed in the course of the above procedure if the strict volume conservation is sought. For the wave–body interaction problems of interest in the present work, volume conservation is a desirable property but not vital. Therefore, this issue will be left to future studies, especially, together with our recent work on improving the volume conservation properties of the level set method using the volume-of-fluid and particle approaches [45,46].

3.5. Time step

The time step Δt is restricted by the CFL condition, gravity, and surface tension. With a CFL restriction of 0.5, the following relationship can be established as discussed in [20]:

$$\Delta t \leqslant 0.5 \left(\frac{C_{cfl} + \sqrt{(C_{cfl})^2 + 4(G_{cfl})^2 + 4(S_{cfl})^2}}{2} \right)^{-1}, \tag{51}$$

with the convective time step restriction

$$C_{cfl} = \max\left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z}\right), \tag{52}$$

the time step restriction due to gravity

$$G_{cfl} = \sqrt{\frac{|g_x|}{\Delta x} + \frac{|g_y|}{\Delta y} + \frac{|g_z|}{\Delta z}}, \tag{53}$$

and the time step restriction due to surface tension

$$S_{cfl} = \sqrt{\frac{\sigma|\kappa|}{\rho_G(\min(\Delta x, \Delta y, \Delta z))^2}}. \tag{54}$$

3.6. Parallelization and solution procedure

The sharp interface Cartesian grid solver for two-phase flows interacting with moving bodies has been parallelized using a domain decomposition technique with the Message Passing Interface (MPI) library for inter-process communication. The domain decomposition is trivial as a Cartesian grid is divided into uniform blocks with equal number of grid points. The total number of grid blocks is just the number of processes to be used.

The layers of ghost points to be exchanged between neighboring blocks depends on the finite difference stencils used in the simulations. If only second-order central-difference schemes are used, then one layer of ghost cells will be adequate even with the requirement of constructing interpolation stencils for momentum forcing as discussed in [49]. In this study, a third-order QUICK scheme is used for the convection terms in the momentum equation and a fifth-order HJ WENO scheme is used for the evolution and reinitialization equations of the level-set function. Therefore, two and three layers of ghost cells for the velocity components and level-set function are used to ensure no loss of accuracy near the grid block interfaces, respectively.

In summary, the overall algorithm for one time step loop can be listed in detail as follows:

(1) Setup immersed moving boundaries: Section 3.3.
(2) Solve the level-set function ϕ: Section 3.2.
(3) Apply contact angle boundary condition of ϕ on the immersed boundaries: Section 3.4.
(4) Define the fluid properties, density ρ and viscosity μ: Section 2.2.2.
(5) Compute eddy viscosity v_t if the SGS model is used: Section 2.3.
(6) Compute the right-hand side of the momentum equation: Section 3.1.2.
(7) Apply momentum forcing for the immersed boundaries: Section 3.3.
(8) Invert the momentum equation: Section 3.1.2.
(9) Apply the first corrector to the first intermediate velocity $\hat{\mathbf{u}}$: Section 3.1.1.
(10) Compute the right-hand side of the pressure Poisson equation, $\nabla \cdot \mathbf{u}^*$: Section 3.1.1.
(11) Add the correction terms due to the jump condition in $\nabla \cdot \mathbf{u}^*$: Section 3.1.3.
(12) Setup the coefficient matrix and solve the Poisson equation using PETSc: Section 3.1.4.
(13) Update the velocity field using the second corrector: Section 3.1.1.
(14) Apply field extension for the immersed boundary treatment: Section 3.3.
(15) Compute the time step Δt: Section 3.5.

4. Results

The basic solver for unsteady incompressible two-phase flows is first validated against the micro-scale case of a 3D spherical cap bubble rising in a quiescent liquid. Then the small-scale cases of water entry and exit of a circular cylinder are simulated and compared with reference data in the literature. The case of two counter-rotating ellipses in a partially filled tank is carried out to show the capabilities of the current method for two-phase flows with multiple bodies undergoing complex motions. Then the 3D case of waves generated by sliding mass is presented to illustrate the accuracy and applicability of the current method in large-scale problems. Finally, several ship flow cases are performed to further validate the method in the applications of large-scale wave–body interaction problems in the real world. More validation cases can be found in [52,53].

4.1. A spherical cap bubble rising in a quiescent liquid

The spherical cap bubbles rising in a quiescent liquid were studied experimentally in [17]. One of the cases considered here has the following parameters: $\rho_L = 0.8755\ \mathrm{g\ cm^{-3}}, \rho_G = 0.001\ \mathrm{g\ cm^{-3}},\ \mu_L = 1.18\ \mathrm{P},\ \mu_G = 0.01\ \mathrm{P}$, and the surface tension coefficient $\sigma = 32.2\ \mathrm{dyn\ cm^{-1}}$. The volume of the bubble is 0.94 $\mathrm{cm^{-3}}$ and the corresponding effective radius is 0.61 cm. In the experiment, the terminal speed of this bubble is 21.5 $\mathrm{cm\ s^{-1}}$.

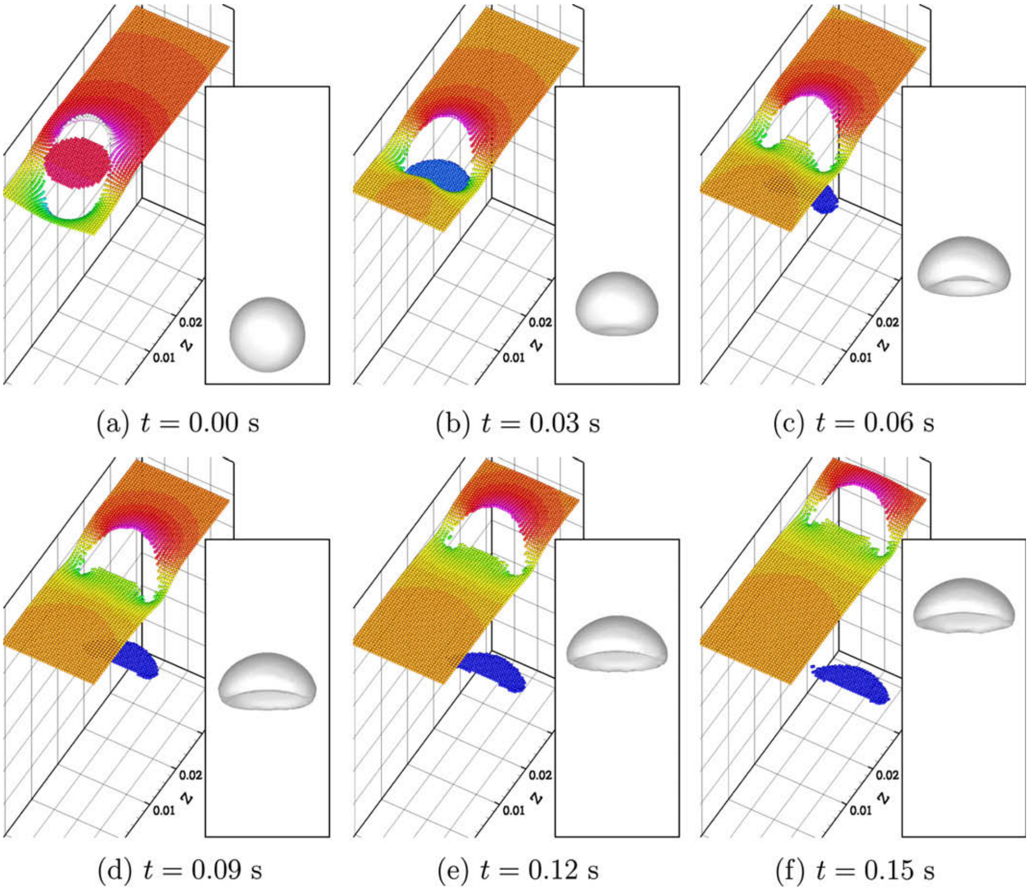
The computational domain for this case is 8 cm × 8 cm × 13.6 cm. Three different grids, $40 \times 40 \times 96$, $48 \times 48 \times 120$, and $64 \times 64 \times 160$, are used with the path of the bubble covered by uniform grid sizes of $\Delta h = 0.10$ cm, 0.08 cm, and 0.06 cm, respectively. Slip-wall boundary condition is applied to all boundaries. As the bubble rises, it changes from a spherical shape to a cap shape (see Fig. 6). The dynamic pressure $(p - \rho g_z z)$ profiles across the x–z center plane of the bubble are also shown in Fig. 6. The sharp jump of pressure due to gravity and surface tension is evident. Also, note that the (absolute) total value of the pressure jump is building up as the bubble rises due to the increasing distance of the bubble to the reference plane of zero hydrostatic pressure.

The time series of the bubble top point on three different grids are given in Fig. 7. It shows that the bubble rises at a constant speed after $t = 0.15$ s. The steady rising speed from the computation on the third grid is 21.0 cm s^{-1}. It is very close to the experimental value and there is only a 2.3% difference, considering that this grid is still a very coarse grid with only about 20 grid points covering the bubble in one direction. In [13], a 2D axisymmetric simulation using the VOF method was performed on a uniform grid of size $\Delta h = 0.014$ cm, which is more than four times finer than the third grid used here, and the exact experimental value was obtained. To have the same resolution in a 3D simulation, a non-uniform grid of more than 5 million points has to be used. In addition, the time step will be extremely small due to the severe constraint associated with surface tension, which makes the 3D case on such a grid too expensive for the purpose of validating the code. Adaptive grid refinement could be helpful but is beyond the scope of the present work. Nevertheless, there is a clear trend of monotonic grid convergence of the rising speed (slope of the trajectory curve) to the experimental value on the three different grids. It is reasonable to expect that, with the current sharp interface method, an exact match with the measured value can be reached after the grid is further refined.

4.2. Moving bodies

4.2.1. Water entry and exit

The water impact/entry problems have been numerically studied in [22,25], among others. Here the same parameters as those in [25] are used: a circular cylinder of radius $R = 1$ is placed in air and the distance of its center to the air–water



(a) $t = 0.00$ s (b) $t = 0.03$ s (c) $t = 0.06$ s

(d) $t = 0.09$ s (e) $t = 0.12$ s (f) $t = 0.15$ s

Fig. 6. Evolution of the bubble shape and the pressure profile of the vertical center plane cutting through the bubble (finest grid, portion of the computational domain shown).

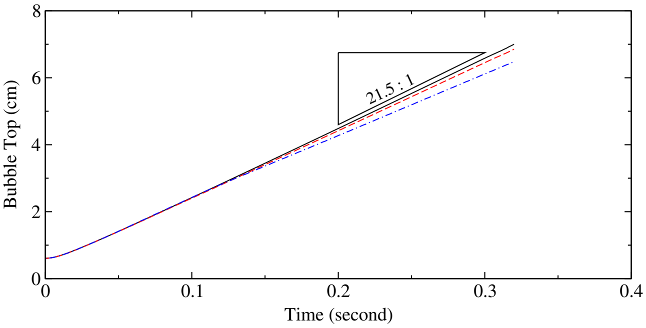


Fig. 7. Time series of the top point of a spherical cap bubble rising in a quiescent liquid on three different grids. –·– 40 × 40 × 96, – – – 48 × 48 × 120, — 64 × 64 × 160. Experimental value is also shown.

interface is $H = 1.25$. The gravity is set to be $g = -1$ and the cylinder is given a constant downward velocity $V = -1$ at $T = Vt/H = 0$ with t the time in the calculation. The computational domain is $40R \times 24R$ in the horizontal and vertical directions. A non-uniform grid 300 × 240 is generated for the whole domain and uniform grid of size $\Delta h = 0.05R$, which is the same as that in [25], is used to cover the path of the cylinder and near the air–water interface.

A series of snapshots are shown in Fig. 8. As the cylinder impacts the air–water interface, two oblique jets are generated along the left and right sides of the cylinder, respectively. The vortices shed from the shear layers along the surface of the cylinder interact with the air–water interface. It is very interesting to observe these vortices attaching to the air–water interface and following the interface motion.

The results are very close to those given in [25] up to $T = 2$. However, the current method is able to capture the entrainment of air as the two-phase model is used here and yields a stronger, higher water jet at a later time than that in [25], in which the single-phase free-surface flow is solved.

To study the effect of the contact angle boundary condition discussed in Section 3.4, the water entry problem is also simulated without enforcing the contact angle condition. Fig. 9 shows the comparison of the relative positions of the interface and the cylinder at several instants from the simulations with enforced and neglected contact angle condition, respectively. The interfaces away from the immediate vicinity of the cylinder in the two simulations match very well. In the former, the interfaces are always perpendicular to the cylinder surface as a contact angle of 90° is enforced. In the latter, however, the contact fronts essentially follow the boundary layer flow along the cylinder surface and a contact angle of about 0° is exhibited as the interfaces are tangential to the cylinder surface. What makes the enforcement of the contact angle boundary condition particularly relevant to the accuracy of the results is the comparison in Fig. 9(d). In the latter simulation, two bubbles resulting from the enclosed air cavity on top of the cylinder grow from the cylinder surface and rise in the wake of the cylinder. Apparently, this is a non-physical phenomenon induced by the neglect of contact angle boundary condition, although air entrainment does occur in the course of water entry.

For the water exit problem, the parameters given in [12] are used. The overall setup is the same as the water entry problem except that the cylinder starts its upward motion from a height of $H = -1.25$ below the calm water surface with a constant upward velocity $V = 0.39$.

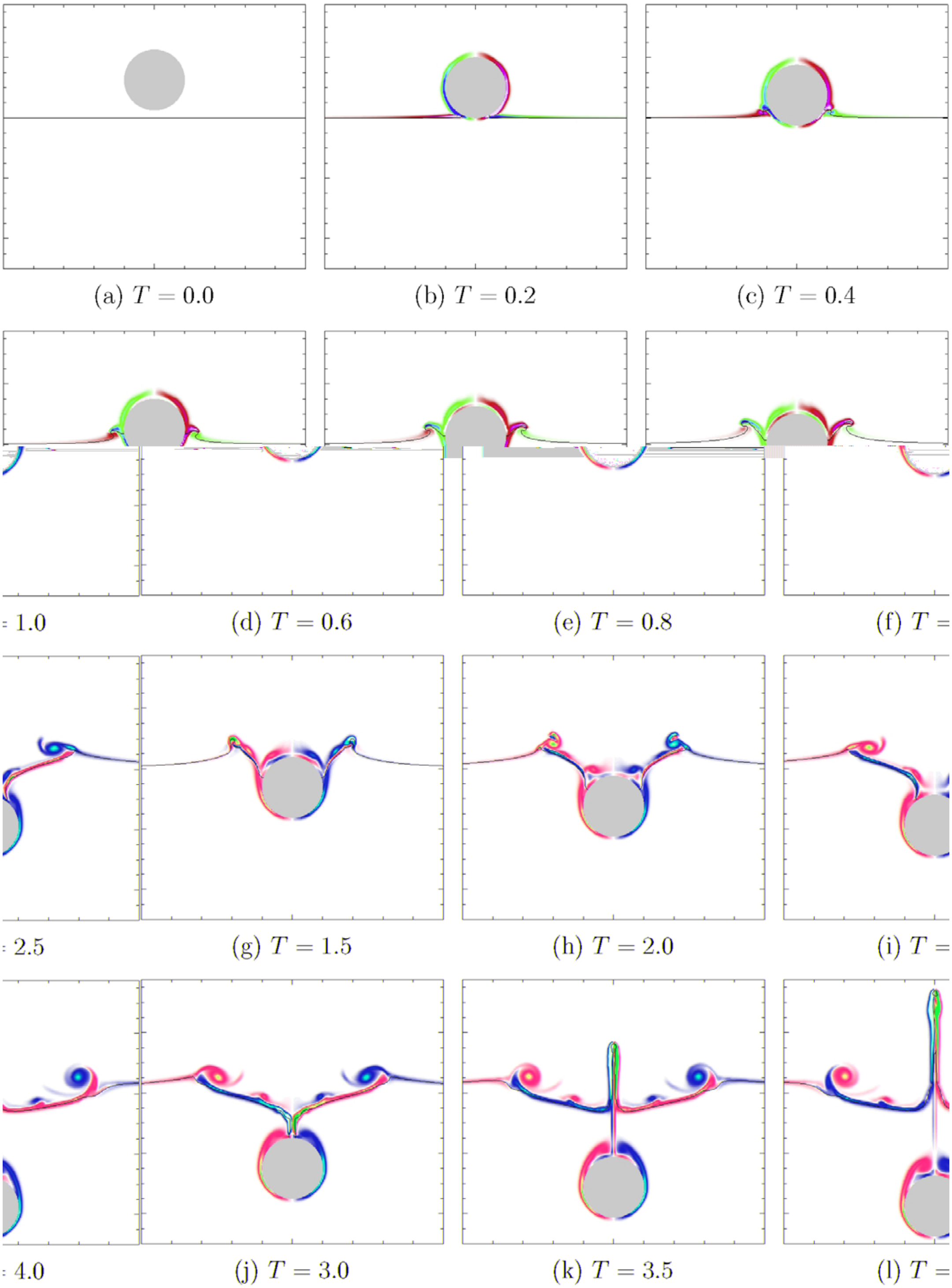
The snapshots of the interactions between the cylinder and the two-phase interface are shown in Fig. 10. As the cylinder moves upward, two shear layers are developing along the left and right sides of the cylinder. Two vortices shed from these two shear layers interact with the air–water interface and form two dipoles beneath the interface. Waves are generated in the exiting process and propagate toward the left and right sides of the domain.

To further validate the numerical method, the comparison between the current results with the boundary element simulation of [12] is shown in Fig. 11. Very good agreement is observed for the two time instances $T = 0.4$ and $T = 0.6$ given here. However, the boundary element model had to be stopped when the cylinder started to pierce the free surface in [12]; whereas the current method can simulate the whole water exit process without any special treatment.

4.2.2. Rotating ellipses in a tank

The water entry and exit cases in the previous sections are interesting; however, a single body with constant motion in a two-phase flow does not pose major challenges to many computational methods. To demonstrate the capabilities of the current computational method and the ease of setting up simulations for complicated cases, two ellipses rotating in a partially filled tank is calculated. A lot of intricate phenomena have been integrated into this conceptually "simple" case.

As shown in Fig. 12(a), a 2D tank of $4.0L \times 2.4L$ (L is a reference length scale) is filled with water of $1.3L$ height to the tank bottom. Two ellipses of the same sizes, with a semi-major axis of $0.9L$ and a semi-minor axis of $0.2L$, are placed in the tank. The left ellipse, with a vertical long axis initially, pierces the interface and starts to rotate with a constant angular speed of $\pi/3$ radians per second counterclockwise; while the right ellipse, with a horizontal long axis initially, is just submerged of $0.1L$ below the air–water interface and starts to rotate with the same angular speed, but clockwise. The density ratio and viscosity ratio between the air and water are 1.2×10^{-3} and 1.8×10^{-2}, respectively. The Reynolds number based on the



(a) $T = 0.0$ (b) $T = 0.2$ (c) $T = 0.4$

$= 1.0$ (d) $T = 0.6$ (e) $T = 0.8$ (f) $T =$

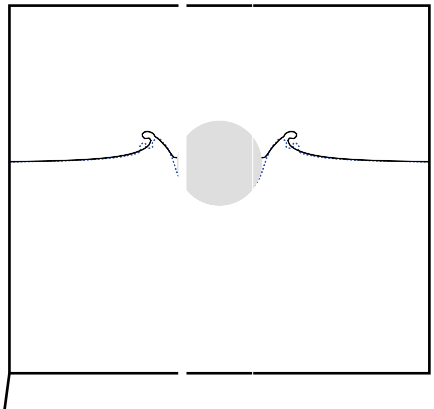
$= 2.5$ (g) $T = 1.5$ (h) $T = 2.0$ (i) $T =$

$= 4.0$ (j) $T = 3.0$ (k) $T = 3.5$ (l) $T =$

Fig. 8. Water impact and entry problem: the air–water interface position (solid black line) and vorticity contours ($-20 < \omega < 20$ with intervals of 0.2).

maximum tip velocity, the length of ellipse semi-major axis, and the density and viscosity of water is about $Re = 0.85 \times 10^5$. The Froude number based on the maximum tip velocity and water height is $Fr = 0.264$. A uniform Cartesian grid of 400×240 is used to cover the whole computational domain.

In this case, there are a lot of complicated interactions between the bodies, flow, interface, wall, and vortices. As the ellipses start to rotate, vortices shed from the tips as shown in Fig. 12(b), $t/T = 0.2$. At $t/T = 0.5$ (Fig. 12(d)), a strong upward jet is squeezed in between the left side-wall and the ellipse tip. This jet is attached to the side-wall initially and then separates



from it at the upper position. Also, instabilities develop in the air–water shear layer as the jet rises. When the rotation reaches one period as shown in Fig. 12(i), the right ellipse also squeezes a stronger, but thinner jet than the left ellipse did, in between the tip and the right side-wall. This time, the jet is highly vortical; it reaches much higher and becomes unstable and collapses in a more irregular and violent manner (see Fig. 12(h)).

To accurately capture these interactions, a versatile, high-fidelity, and robust solver that can handle moving boundaries, evolving interfaces, and gas–liquid–solid triple points (contact angle boundary condition) with ease is critical. The efficiency and ease of setup of the current solver is also remarkable. For this complicated simulation, a fixed, uniform Cartesian grid of $\Delta h = 0.01L$ is used without any extra grid manipulation. It takes about 3 s for one time step on an Intel Pentium D 3.00 GHz Linux desktop with more than 95% CPU time used by the multigrid Poisson solver.

4.2.3. Waves generated by a sliding wedge

In [28] the waves generated by 3D sliding masses were studied using both large-scale wave tank experiments and numerical methods. Here the case of a subaerial wedge slide as detailed in [28] is simulated. The setup is the same as that in the reference. However, one coordinate direction is chosen to be parallel to the slope in order to obtain better grid resolution close to the wall. The computational domain in this simulation is 6.5 m × 3.7 m × 3 m in the direction along the slope, the spanwise direction, and the direction normal to the slope, respectively. And the Cartesian grid is 256 × 144 × 120 in the corresponding directions. The motion of the sliding wedge is given by laboratory measurement.

The snapshots of air–water interface profile are shown in Fig. 13. At the beginning, the sliding wedge enters the water slowly and pulls away the water in front of it. As the wedge reaches its full speed, the water around it follows the downward motion of the wedge and a big cavity appears on top of the wedge. The wedge keeps sinking and water rushes into the cave. A big jet is generated near the shoreline when the left and right flushing waves merge at the centerline. This jet then collapses and strong waves propagate outward.

These snapshots qualitatively agree with the simulation in [28] very well. To further validate the solution, the air–water interface fluctuations are compared with the data recorded from a series of wave gauges in the experiment and shown in Fig. 14. The positions of the gauge stations are shown in Fig. 14(a). The overall agreement is very good, especially, in the beginning stage of the landslide process.

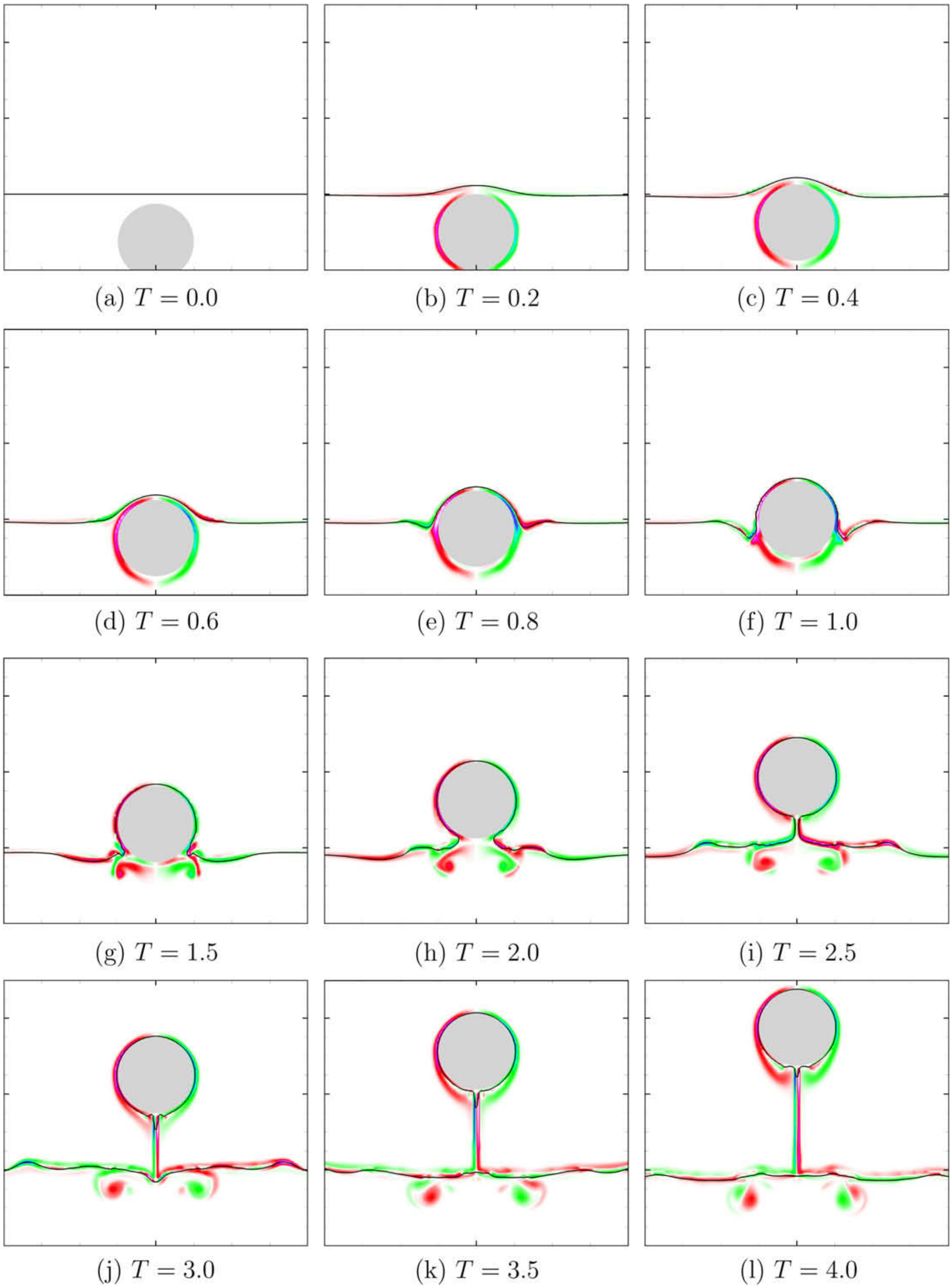
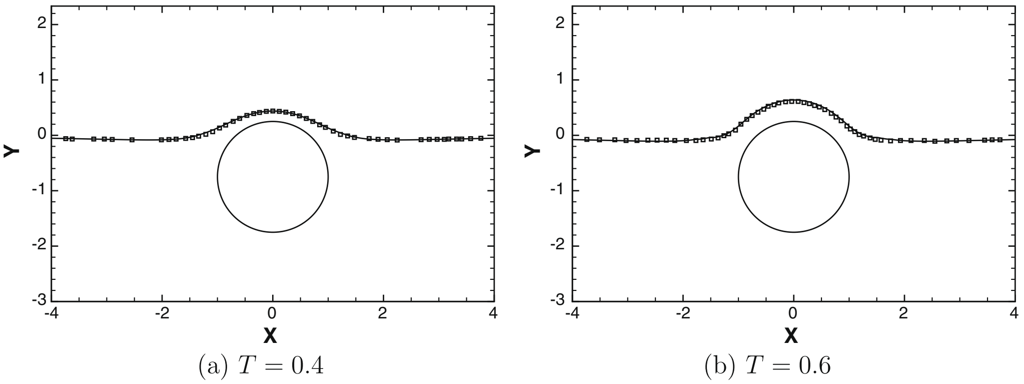


Fig. 10. Water exit problem: the air–water interface position (solid black line) and vorticity contours ($-10 < \omega < 10$ with intervals of 0.1).

4.3. Ship hydrodynamics

4.3.1. Wigley hull

The case of a Wigley hull in steady forward speed advancing in calm water at $Fr = 0.267$ (based on ship length L and forward speed U) in an even keel condition is studied first. The computational domain is $3.5L \times 1.5L \times 1.0L$ in the streamwise,



(a) $T = 0.4$

(b) $T = 0.6$

Fig. 11. Comparison of the free-surface profiles: — present simulation; □ boundary element simulation [12].

spanwise, and vertical directions, respectively. Note that in this case a half domain is used due to the symmetric shape of the Wigley hull with respect to the vertical center plane and that a steady flow condition is used. A non-uniform Cartesian grid of $512 \times 120 \times 120$ is used to cover the half domain. The grid is stretched to cluster more points near the hull surface, especially, in the bow and stern regions.

Fig. 15 shows a comparison of the wave pattern obtained with the Cartesian grid solver and the experimental data from [7]. The simulation gives a remarkably accurate wave pattern with well-matched wave length and amplitude. However, the amplitude of the second trough is slightly under-predicted as the grid there is not fine enough. Also, there is an under-prediction of the upward velocity in the wake caused by the artificially thickened boundary layer, which results in a smaller wave peak located downstream of the stern.

To further demonstrate the accuracy and efficiency of the current computational method, two additional simulations are setup for the Wigley hull case: one using the current solver on a coarser Cartesian grid of $256 \times 68 \times 68$ (1.2 million grid points), shown on the top of Fig. 16; and the other using the CFD solver CFDShip-Iowa version 4 [5] on an overset grid of the above coarse Cartesian grid and a body-fitted grid of $161 \times 79 \times 75$ (0.95 million grid points) attached to the hull surface, shown on the bottom of Fig. 16. CFDShip-Iowa version 4 is a general-purpose Reynolds-averaged Navier–Stokes (RANS) solver with a focus on ship hydrodynamics. It solves unsteady turbulent free-surface flows around moving ships on multi-block overset body-fitted grids with a blended $k - \epsilon/k - \omega$ turbulence model and a level-set method for free-surface capturing. The reader is referred to [5] for more details about this solver.

The comparison of results from both simulations is given in Fig. 16. Significantly improved wave field is obtained from the Cartesian grid solver as compared with the single-phase overset grid RANS solver. Possible reasons for the better resolution of the wave pattern with the Cartesian grid solver than the body-fitted grid solver include the following among others: (a) the spatial discretization of the Cartesian grid solver is much less diffusive, especially, a fifth-order HJ WENO scheme versus a second-order upwind scheme for the level-set solvers which are used in the former and the latter, respectively; and (b) the time-averaging RANS simulation in the latter results in more dissipation in the free-surface waves. However, the RANS solution in the wake does match the experimental data better than that of the Cartesian grid solver, although coming from an expensive body-fitted grid of another nearly 1 million points for resolving the boundary layer.

4.3.2. Model 5365

The DTMB[1] model 5365, Research Vessel (R/V) Athena, is a very streamlined ship with a transom stern, which adds complexities on the flow because it results in boundary layer separation and stern wave breaking for larger ship speeds. At the selected speed ($Fr = 0.25$), the transom is wet and separation occurs. For the case, the domain size and the grid are the same as the fine grid Wigley hull simulation shown above.

The wave field with a highly unsteady transom is shown in Fig. 17. Similar to the Wigley hull case, the Cartesian grid solver predicts phase and amplitude of the wave system quite accurately. Wave cuts at several spanwise cross planes are compared with the experimental data in Fig. 18. The overall agreement with the experimental data [10] is excellent. There is some phase difference near the bow due to inadequate grid resolution for the bow region although the very steep bow wave is generally captured. Also the waves are dissipated farther away and downstream of the ship ($y/L = 0.308, x/L \approx 1.8$) as a result of grid stretching.

4.3.3. Model 5512

The surface combatant model DTMB 5512 (hull shape shown in Fig. 19) has a sonar dome on the bow, which is an additional geometric complexity with respect to the R/V Athena model. There are extensive data available for this ship at

[1] David Taylor Model Basin, now the Naval Surface Warfare Center, Carderock Division, Bethesda, MD.

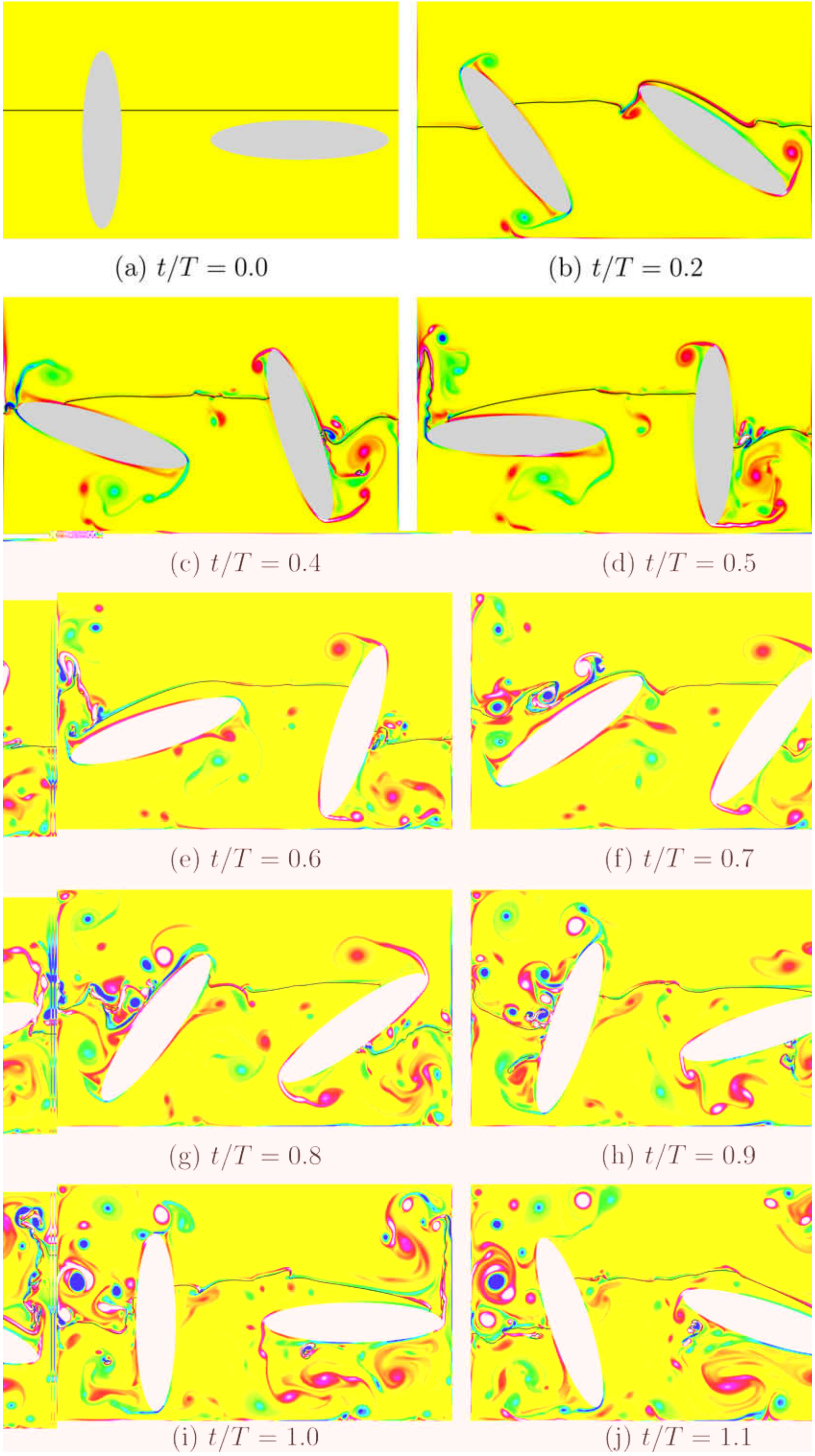


Fig. 12. Simulation of two rotating ellipses in a tank: the air–water interface profile (solid black line) and vorticity contours ($-100 < \omega < 100$ with intervals of 1).

different model scales, including wave fields [14] and velocities at the nominal propeller plane ($x/L = 0.935$) [29]. In this study, the case of $Fr = 0.28$ is used. The computational domain is $3.5L \times 1.5L \times 1.5L$ in the streamwise, spanwise, and vertical directions, respectively. A fairly coarse grid of $256 \times 128 \times 128$ (4.2 million grid points) is used to cover the whole domain.

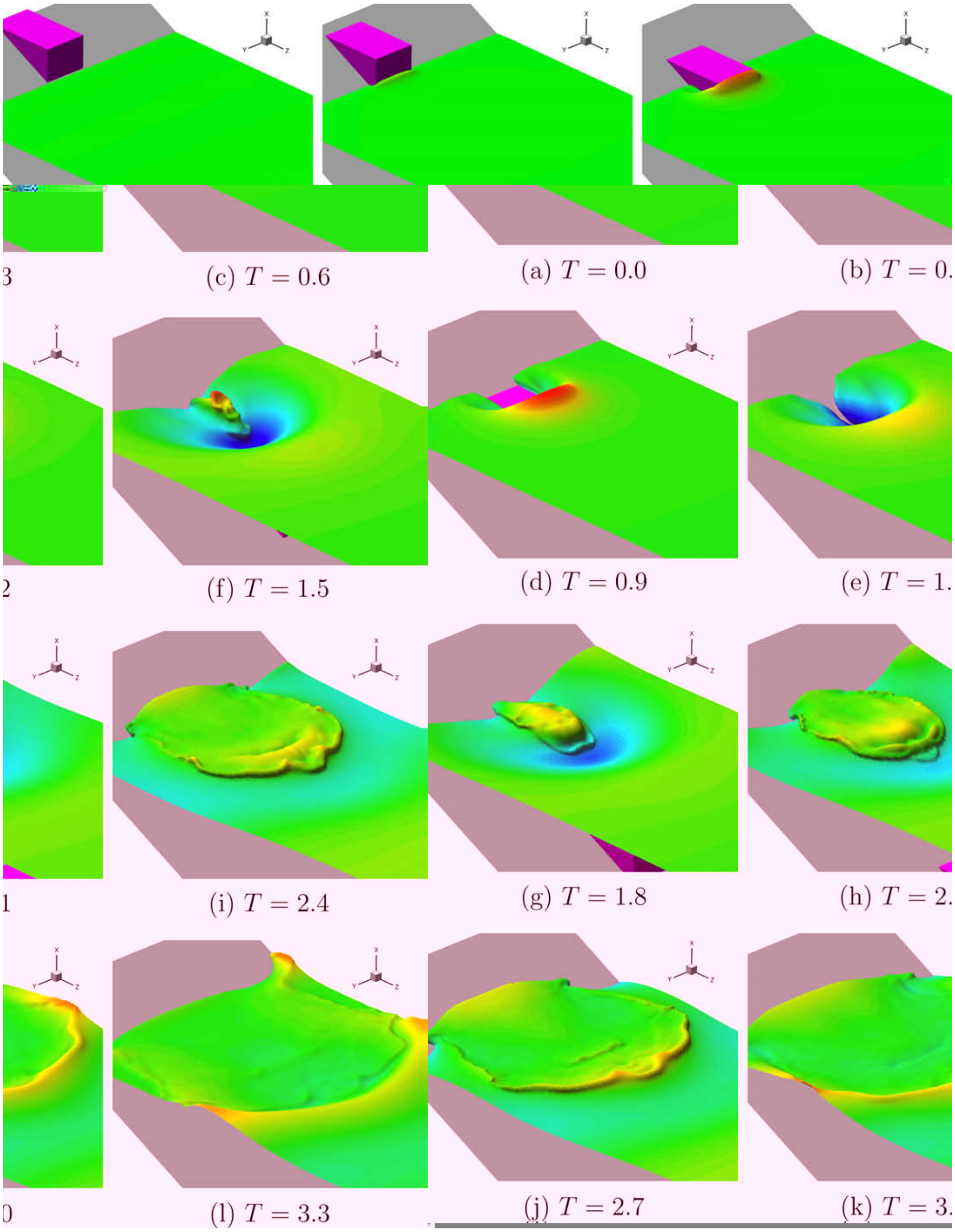
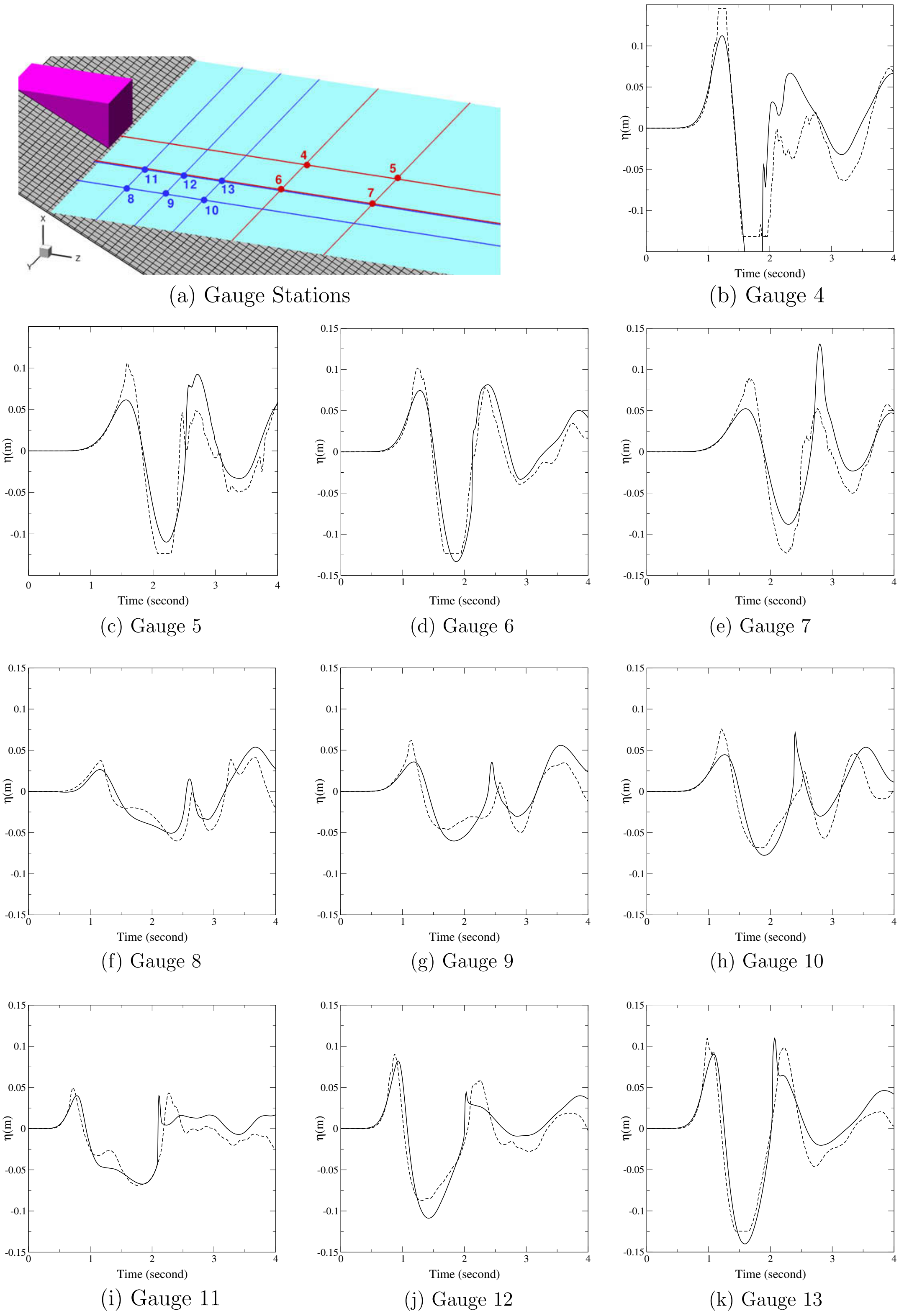


Fig. 13. Landslide-generated waves. The air–water interface is colored by the elevation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

With the current immersed boundary formulation for LES, enough grid points have to be placed within the viscous sublayer of the boundary layer. Otherwise, the effects of the near-wall turbulence is hardly captured via a linear interpolation of the velocity with no-slip wall boundary condition at the ship hull. A grid with multiple billions of grid points, which is more than ten times finer than the current grid in each direction, has to be used for performing LES of model-scale ship flows. It is achievable, but extremely expensive, even with the state-of-the-art supercomputers. Fortunately, wall-layer models [36] that represent the near-wall turbulence in a statistical or stochastic manner instead of directly resolving it can be used to alleviate the grid requirement near the wall.



(a) Gauge Stations

(b) Gauge 4

(c) Gauge 5

(d) Gauge 6

(e) Gauge 7

(f) Gauge 8

(g) Gauge 9

(h) Gauge 10

(i) Gauge 11

(j) Gauge 12

(k) Gauge 13

Fig. 14. Time series of the air–water interface fluctuations at different wave gauges for the landslide-generated waves: ⋯⋯ experimental data [28]; — present simulation. The gauge stations are shown in (a).

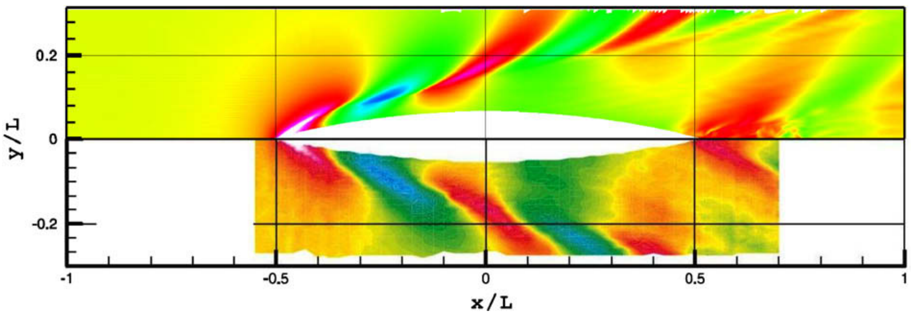


Fig. 15. Comparison of the wave fields for Wigley hull at $Fr = 0.267$ between the Cartesian grid simulation (top) and the experimental data [7] (bottom). The contours of the free-surface elevation in the simulation range from -10^{-2} to 10^{-2} with intervals of 10^{-3}.

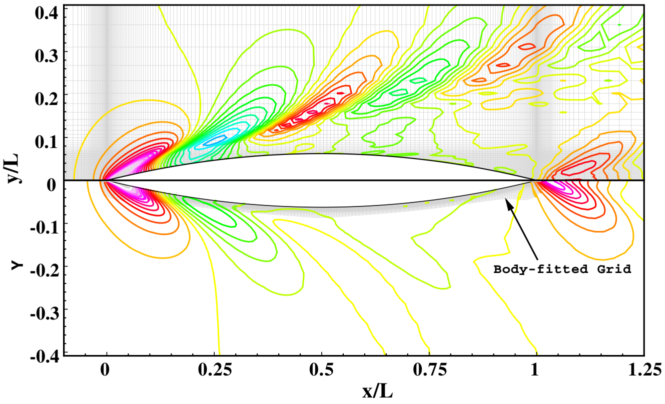


Fig. 16. Comparison of the wave fields for Wigley hull at $Fr = 0.267$ between the Cartesian grid simulation (top) and the overset body-fitted grid RANS solution [51] (bottom). The contours of the free surface elevation range from -1.2×10^{-2} to 1.2×10^{-2} with intervals of 10^{-3}.

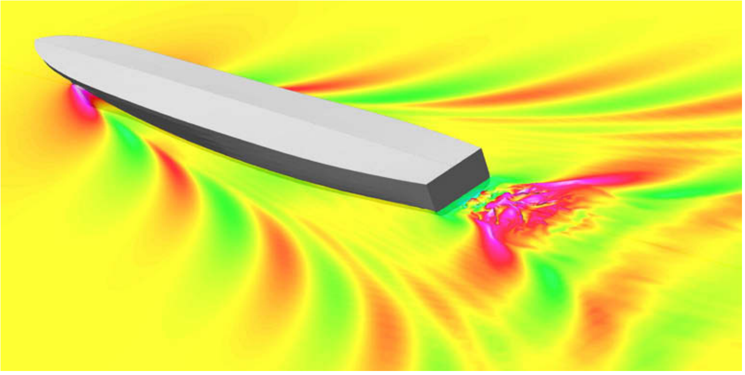


Fig. 17. Instantaneous wave field of the R/V Athena at $Fr = 0.25$, The contours of the free-surface elevation range from -10^{-2} to 10^{-2} with intervals of 10^{-4}.

As shown in the Wigley hull case, the current solver can reasonably predict the wave pattern even with a very coarse Cartesian grid that is far from resolving the boundary layer. This will also be confirmed for the model 5512 case. In addition, to demonstrate the applicability of wall-layer models with immersed boundary method for high Reynolds number ship flows, a one-way coupled hybrid RANS/LES is performed for this case: the steady solution of the boundary layer velocity field from the two-phase CFDShip-Iowa solver (version 5) [18] is directly used as the input to the outer LES field instead of a coupled, interactive manner. Therefore, the immersed boundary treatment is carried out using the RANS velocity solution on a surface (solution matching surface herein) some distance away from real hull surface. This solution matching surface is exactly one layer of body-fitted grid parallel to the ship hull. The wall distance at this surface to the ship hull ranges from $y^+ = 200$ to 900, with most of the surface area with $y^+ = 400$, well within the boundary layer, as shown in Fig. 20. Again, it must be noted that in this study the Cartesian solver does not receive instantaneous turbulent fluctuations, which have to be given in a real hybrid RANS/LES simulation, from the body-fitted RANS solver except one layer of steady RANS velocity solution.

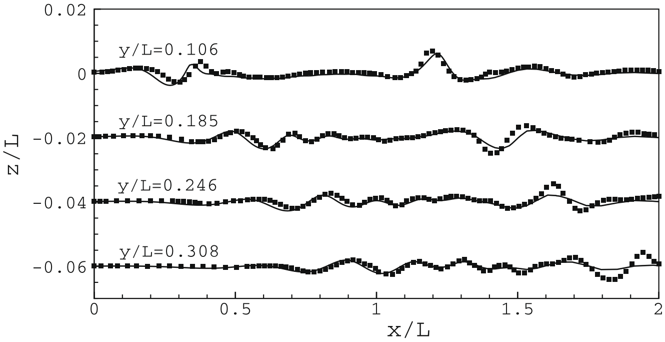


Fig. 18. Wave cuts at four different spanwise planes for the Athena R/V at $Fr = 0.25$: — cartesian grid LES results; ■ experimental data from [10].

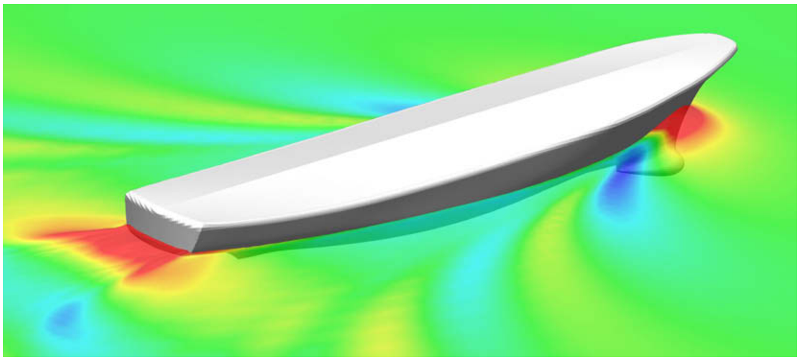


Fig. 19. Instantaneous wave field of the Model DTMB 5512 at $Fr = 0.28$ from the one-way coupled hybrid RANS/LES simulation. The contours of the free-surface elevation range from -5×10^{-3} to 5×10^{-3} with intervals of 5×10^{-5}.

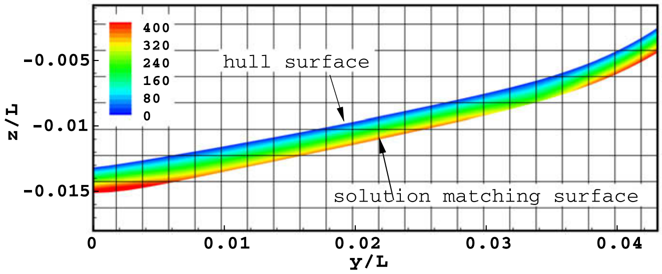


Fig. 20. y^+ distribution from the body-fitted RANS solution at the nominal propeller plane for the model DTMB 5512 at $Fr = 0.28$.

The instantaneous wave field from the one-way coupled hybrid RANS/LES approach is given in Fig. 19. The comparison of the averaged wave field and the experimental data [14] is given in Fig. 21, with a plane section of the Cartesian grid shown. In the near field, the agreement is satisfactory. The very coarse grid used in the far-field results in a poor prediction of the wave field in that region. The comparison of the wave field between the coupled approach and the one with no-slip wall (hull) boundary condition is shown in Fig. 22. The overall wave pattern of the coupled approach is very similar to the latter. However, it exhibits much better agreement with experimental data in the wake area. It indicates that the more realistic near-wall velocity distribution "borrowed" from the body-fitted RANS solver gives improved wake wave prediction, comparing to the uncoupled approach with a linear near-wall velocity distribution, which is no longer valid on such a coarse grid.

To further demonstrate the effects of a wall-layer approximation on the flow field, the mean streamwise velocity distribution at the nominal wake plane from the coupled approach is compared with the ensemble-averaged experimental data from [29] in Fig. 23. The overall boundary layer velocity distribution is very close to the experimental data. However, the current computation under-predicts the turbulence in the near hull region, which is expected as the outer LES simulation receives no turbulence information from the boundary layer. As shown in Fig. 23, the immediate consequence of the deficient turbulence activity is that the vortices induced by the sonar dome in the boundary layer still have a strong effect in

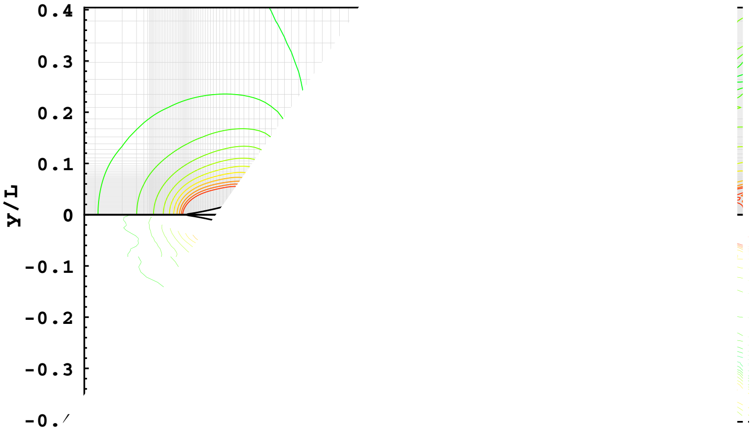


Fig. 21. Comparison of
and the experiment

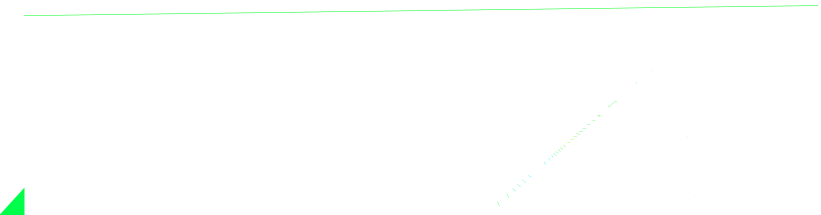


Fig. 22. [illegible] LES solution(bottom). The contours of the free-surface elevation range from− to 5×10 with intervals of 5×10 .

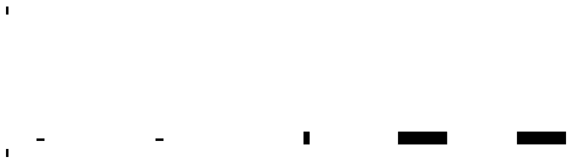
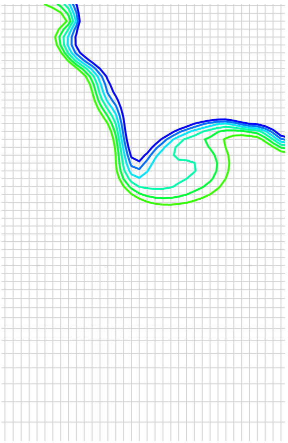


Fig. 23. Comparison of the average streamwise velocity distributions for the model DTMB 5512 at $Fr = 0.28$ between the coupled RANS/LES solution (right) and the experimental data [29] (left). U contours range from 0.55 to 0.95 with intervals of 0.05.

thr



two simulations illustrates that a wall-layer model, even the current crude approach, can produce significant improvements to the linear immersed boundary formulation on a very coarse Cartesian grid.

5. Conclusions

In this paper, a sharp interface immersed-boundary/level-set method has been developed for LES of 3D two-phase flows interacting with moving bodies on fixed Cartesian grids. This work is based on a sharp interface immersed boundary method for fluid flows interacting with moving bodies [49] and a level-set based ghost–fluid method for the sharp interface treatment of the fluid–fluid interface [26]. The mathematical formulation for two-phase incompressible turbulent flows including interface jump conditions and subgrid-scale model for LES has been presented. Detailed numerical methods for the solution of complicated gas–liquid–solid systems, such as the four-step fractional-step method for velocity–pressure coupling, momentum solver, jump conditions treatment, Poisson equation solver, level-set solver, and immersed boundary treatment have been discussed. In addition, a simple contact angle boundary condition treatment has been developed. By using the similar interpolation stencil as the momentum forcing scheme, the current treatment is consistent with the immersed boundary formulation. It is also very easy to implement and applicable to both 2D and 3D problems.

Several cases of different scales have been used to validate the numerical method. In the first case, a 3D spherical cap bubble rising in quiescent fluid has been simulated on three different grids. Monotonic grid convergence of the bubble rising speed has been observed and the result on the fine grid is within 3% of the experimental measured value. This micro-scale cases shows the accuracy of the basic Navier–Stokes solver for unsteady incompressible two-phase flows including strong surface tension effects. Then the small-scale cases of 2D water entry and exit of a circular cylinder have been carried out. The air–water interface profiles have been compared with other numerical results and excellent agreement has been obtained. To demonstrate the capabilities of the current computational method and the ease of setting up simulations for complicated cases with multiple bodies, the 2D case of two ellipses rotating in a partially filled tank has been performed. A lot of intricate interactions of wave–body, body–wall, vortex–body, and vortex–interface have been shown. To further illustrate the applicability of the method in large-scale wave–body interaction problems, a 3D case of waves generated by sliding mass has been presented. The snapshots of the air–water interface at different times agree with the numerical results in the reference very well. The time series of the air–water interface fluctuations have been compared with the data recorded from a series of wave gauges in the experiment and the overall agreement is very good.

Finally, three ship flow cases have been carried out to further validate the method in the applications of large-scale wave–body interaction problems in the real world. The Wigley hull case has been performed on two grids. The fine grid wave field agrees with the experimental wave pattern very well. The coarse grid solution has been compared with the solution from an overset grid RANS solver, which used the coarse Cartesian grid as the background grid and a body-fitted grid attached to the hull surface, and much better far- and near-field wave pattern has been obtained except the wake field due to a lack of boundary layer resolution. Then the R/V Athena model has been simulated and the near- and far-field wave cuts agree with the experimental data very well. In addition, the case of model DTMB 5512 was used to demonstrate the promise of wall-layer models with the immersed boundary method on fixed Cartesian grids for high Reynolds number ship flows. In this case, the steady RANS solution of the boundary layer velocity field from a body-fitted grid was used as the input to the outer LES field. On a very coarse grid, encouraging results of the wave pattern and the average streamwise velocity have been obtained. It shows that the effects of high Reynolds number boundary layers can be incorporated into a Cartesian grid solver through coupling with a curvilinear solver or some other means to handle the viscous boundary layers.

Currently, several approaches for capturing the effects of viscous boundary layers within the framework of a Cartesian grid solver are under consideration. In general, a series of strategies with different levels of complexity can be adopted. The simplest way is to use a wall function approach [38], in which the velocity distribution near the wall is no longer a linear profile; instead, a non-linear profile given by a specific wall function can be used. Using wall functions in Cartesian grids would render a body-fitted grid unnecessary. However, this approach would require very fine grids even for model-scale problems. A more general way is to use advanced wall-layer models on a body-fitted grid, such as solving the turbulent boundary layer equations or even RANS equations with given pressure gradients, which can be extrapolated from the Cartesian grid solution. Also, a coupled curvilinear/Cartesian grid method is proposed to combine the advantages of both Cartesian and body-fitted grids. Curvilinear grids are used to resolve high Reynolds number boundary layers for complex geometries, which are combined with Cartesian grids for the bulk of the flow both of which enable increased accuracy, speed, and scalability. The goal is to use orthogonal curvilinear grid solvers coupled with Cartesian grid solvers. An additional advantage of using orthogonal curvilinear grids instead of non-orthogonal ones is that the discretization of the governing equations is very similar to that in Cartesian grids, and therefore the implementation is greatly simplified and the overall efficiency close to that of Cartesian grid methods.

Acknowledgments

This research was sponsored by the Office of Naval Research (ONR) under Grants N00014-01-1-0073 and N00014-06-1-0420, with Dr. Patrick Purtell as the program manager. The computations were performed at the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) through the Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP). The authors are grateful to Mr. N. Sakamoto for processing the ship flow simulations, Prof. P.M. Carrica for providing the body-fitted RANS solution for the DTMB 5512 case, and their useful inputs to the ship flow cases. Also thanks are due to Prof. T.-R. Wu for providing the landslide experimental data.

References

[1] E. Balaras, J. Yang, Non-boundary conforming methods for large-eddy simulations of biological flows, ASME J. Fluids Eng. 127 (2005) 851–857.

[2] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A.M. Bruaset, H.P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.

[3] R.M. Beam, R.F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation-law form, J. Comput. Phys. 22 (1976) 87–110.

[4] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.

[5] P.M. Carrica, R.V. Wilson, F. Stern, An unsteady single-phase level set method for viscous free surface flows, Int. J. Numer. Meth. Fluids 53 (2007) 229–256.

[6] H. Choi, P. Moin, Effects of the computational time step on numerical solutions of turbulent flow, J. Comput. Phys. 113 (1994) 1–4.

[7] F. D'Este, G. Contento, Time domain linear and weakly nonlinear coupled wave resistance and seakeeping computations, Science and Supercomputing at CINECA, Report, 2003.

[8] D.G. Dommermuth, T.T. O'Shea, D.C. Wyatt, et al., An application of Cartesian-grid and volume-of-fluid methods to numerical ship hydrodynamics, in: Proc. Nineth Int. Conf. Numer. Ship Hydrodynamics, Ann Arbor, MI, 2007.

[9] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35–60.

[10] T. Fu, A. Karion, A. Pence, J. Rice, D. Walker, T. Ratcliffe, Characterization of the steady wave field of a high speed transom stern ship – Model 5365 hull form, Naval Surface Warfare Center, Carderock Division, Hydromechanics Directorate R&D Report, NSWCCD-50-TR-2005, 2005.

[11] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible Navier–Stokes equations with phase change, J. Comput. Phys. 222 (2007) 536–555.

[12] M. Greenhow, S. Moyo, Water entry and exit of horizontal circular cylinders, Philos. Trans. Math. Phys. Eng. Sci. 355 (1997) 551–563.

[13] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423–456.

[14] L. Gui, J. Longo, F. Stern, Towing tank PIV measurement system data and uncertainty assessment for DTMB model 5512, Exp. Fluids 31 (2001) 336–346.

[15] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.

[16] J.-M. Hong, T. Shinar, M. Kang, R. Fedkiw, On boundary condition capturing for multiphase interfaces, J. Sci. Comput. 31 (2007) 99–125.

[17] J.G. Hnat, J.D. Buckmaster, Spherical cap bubbles and skirt formation, Phys. Fluids 19 (1976) 182–194.

[18] J. Huang, P.M. Carrica, F. Stern, Coupled ghost–fluid/two-phase level set method for curvilinear body-fitted grids, Int. J. Numer. Meth. Fluids 55 (2007) 867–897.

[19] G.-S. Jiang, D. Peng, Weighted ENO schemes for Hamilton–Jacobi equations, SIAM J. Sci. Comput. 21 (2000) 2126–2143.

[20] M. Kang, R.P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.

[21] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (2001) 132–150.

[22] K.M.T. Kleefsman, G. Fekken, A.E.P. Veldman, B. Iwanowski, B. Buchner, A volume-of-fluid based simulation method for wave impact problems, J. Comput. Phys. 206 (2005) 363–393.

[23] B.P. Leonard, Stable and accurate convective modeling procedure based on quadratic upstream interpolation, Comput. Meth. Appl. Mech. Eng. 19 (1979) 59–98.

[24] R. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.

[25] P. Lin, A fixed-grid model for simulation of a moving body in free surface flows, Comput. Fluids 36 (2007) 549–561.

[26] X.D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poissons equation on irregular domains, J. Comput. Phys. 154 (2000) 151–178.

[27] H. Liu, S. Krishnan, S. Marella, H.S. Udaykumar, Sharp interface Cartesian grid method II: a technique for simulating droplet interactions with surfaces of arbitrary shape, J. Comput. Phys. 210 (2005) 32–54.

[28] P.L. F Liu, T.-R. Wu, F. Raichlen, C.E. Synolakis, J.C. Borrero, Runup and rundown generated by three-dimensional sliding masses, J. Fluids Mech. 536 (2005) 107–144.

[29] J. Longo, J. Shao, M. Irvine, F. Stern, Phase-averaged PIV for the nominal wake of a surface ship in regular head waves, J. Fluids Eng. 129 (2007) 524–540.

[30] N. Mattor, T.J. Williams, D.W. Hewett, Algorithm for solving tridiagonal matrix problems in parallel, Parallel Comput. 21 (1995) 1769–1782.

[31] C. Meneveau, C.S. Lund, W.H. Cabot, A Lagrangian dynamic subgrid-scale model of turbulence, J. Fluids Mech. 319 (1996) 353–385.
[32] H. Miyata, S. Nishimura, A. Masuko, Finite difference simulation of nonlinear waves generated by ships of arbitrary three-dimensional configuration, J. Comput. Phys. 60 (1985) 391–436.
[33] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[34] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, J. Comput. Phys. 155 (1999) 410–438.
[35] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[36] U. Piomelli, E. Balaras, Wall-layer models for large-eddy simulations, Ann. Rev. Fluid Mech. 34 (2002) 349–374.
[37] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[38] F. Stern, S. Bhushan, P.M. Carrica, J. Yang, Large scale parallel computing and scalability study for surface combatant static maneuver and straight ahead conditions using CFDShip-Iowa, in: Proc. 21st Int. Conf. Parallel Comput. Fluid Dynamics, Moffett Field, CA, 2009.
[39] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[40] M. Sussman, D. Dommermuth, The numerical simulation of ship waves using Cartesian grid methods, in: Proc. 23rd Symposium on Naval Hydrodynamics, Val De Reuil, France, 2000.
[41] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys. 221 (2007) 469–505.
[42] S.O. Unverdi, G. Tryggvason, A front-tracking method for, viscous, incompressible multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[43] S.P. van der Pijl, A. Segal, C. Vuik, P. Wesseling, A mass-conserving level-set method for modelling of multi-phase flows, Int. J. Numer. Meth. Fluids 47 (2005) 339–361.
[44] H. Wang, P.J. Mucha, G. Turk, Water drops on surfaces, ACM Trans. Graphics 24 (2005) 921–929.
[45] Z. Wang, J. Yang, B. Koo, F. Stern, A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves, Int. J. Multiphase Flow 35 (2009) 227–246.
[46] Z. Wang, J. Yang, F. Stern, An improved particle correction procedure for the particle level set method, J. Comput. Phys. (2009), doi:10.1016/j.jcp.2009.04.045.
[47] T. Yabe, F. Xiao, T. Utsumi, The constrained interpolation profile method for multiphase analysis, J. Comput. Phys. 169 (2001) 556–593.
[48] T. Yabe, K. Takizawa, M. Chino, M. Imai, C.C. Chu, Challenge of CIP as a universal solver for solid liquid and gas, Int. J. Numer. Meth. Fluids 47 (2005) 655–676.
[49] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, J. Comput. Phys. 215 (2006) 12–40.
[50] J. Yang, S. Preidikman, E. Balaras, A strongly coupled, embedded-boundary method for fluid–structure interactions of elastically mounted rigid bodies, J. Fluids Struct. 24 (2008) 167–182.
[51] J. Yang, N. Sakamoto, Z. Wang, P.M. Carrica, F. Stern, Two phase level-set/immersed-boundary Cartesian grid method for ship hydrodynamics, in: Proc. Nineth Int. Conf. Numer. Ship Hydrodynamics, Ann Arbor, MI, 2007.
[52] J. Yang, F. Stern, Large-eddy simulation of breaking waves using embedded-boundary/level-set method, in: Proc. 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA-2007-1455, 2007.
[53] J. Yang, F. Stern, A sharp interface method for two-phase flows interacting with moving bodies, in: Proc. 18th AIAA Computational Fluid Dynamics Conference, Miami, FL, AIAA-2007-4578, 2007.
[54] K. Yokoi, Numerical method for a moving solid object in flows, Phys. Rev. E 67 (2003) 045701.