INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **42**:853–884 (DOI: 10.1002/fld.555)

Numerical simulation of unsteady multidimensional free surface motions by level set method

Wusi Yue, Ching-Long Lin*,† and Virendra C. Patel

Department of Mechanical and Industrial Engineering and IIHR—Hydroscience & Engineering, The University of Iowa, Iowa City, IA 52242-1527, U.S.A.

SUMMARY

This paper presents a numerical method that couples the incompressible Navier–Stokes equations with the level set method in a curvilinear co-ordinate system for study of free surface flows. The finite volume method is used to discretize the governing equations on a non-staggered grid with a four-step fractional step method. The free surface flow problem is converted into a two-phase flow system on a fixed grid in which the free surface is implicitly captured by the zero level set. We compare different numerical schemes for advection of the level set function in a generalized curvilinear format, including the third order quadratic upwind interpolation for convective kinematics (QUICK) scheme, and the second and third order essentially non-oscillatory (ENO) schemes. The level set equations of evolution and reinitialization are validated with benchmark cases, e.g. a stationary circle, a rotating slotted disk and stretching of a circular fluid element. The coupled system is then applied to a travelling solitary wave, and two- and three-dimensional dam breaking problems. Some interesting free surface phenomena are revealed by the computational results, such as, the large free surface vortices, air entrapment and splashing of the water surge front. The computational results are in excellent agreement with theoretical predictions and experimental data, where they are available. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: free surface flows; level set method; fractional step method; solitary wave; broken dam

1. INTRODUCTION

Unsteady flows involving free surfaces receive special attention in computational fluid dynamics because of the challenge in dealing with moving surfaces. A variety of numerical methods have been developed over past four decades. They can be classified into two general categories: moving-grid and fixed-grid methods. The moving-grid method is basically a Lagrange-type method that treats the free surface as the boundary of a moving surface-fitted

*Correspondence to: Ching-Long Lin, Department of Mechanical and Industrial Engineering, University of Iowa, Iowa City, IA 52242-1527, U.S.A.

†E-mail: ching-long-lin@uiowa.edu

Contract/grant sponsor: ONR; contract/grant number: N00014-01-1-0262

Received 4 September 2002
Revised 11 March 2003

grid which can be either structured or unstructured, including strictly Lagrangian methods, free Lagrangian methods, mixed Lagrangian–Eulerian methods, etc. [1]. The free surface remains sharp and is computed precisely in the moving-grid method. When grids are highly distorted due to a strongly deformed free surface topology, rezoning (or remeshing) becomes inevitable. Methods by using rezoning are referred to as arbitrary Lagrangian–Eulerian (ALE) methods first developed by Hirt *et al.* [2]. Since flow information in the new grid is transferred from the old grid, excessive numerical diffusion may be induced by frequent rezoning.

Fixed-grid methods can be further divided into two categories: surface-tracking method and surface-capturing method. Both make use of a fixed stationary grid covering both the liquid and gas regions. In the surface-tracking method the free surface is explicitly identified and tracked by means of pre-defined markers or interface-fitted grid cells. In the surface-capturing method the free surface is implicitly captured by a contour of a certain scalar function. In most surface-tracking methods the governing equations are solved only for the liquid and the free surface grid cells, while in the surface-capturing method the equations are solved on both the liquid and gas regions. The surface-tracking method has many variants including front-tracking methods and marker methods. The front-tracking methods by Glimm *et al.* [3] and Unverdi and Tryggvason [4] represent the interface by a connected set of points. An additional unstructured grid is constructed in the vicinity of the interface to explicitly evolve the interface. Restructuring of the interface grid has to be performed dynamically during the computation. The marker methods include the marker-and-cell (MAC) method by Harlow and Welch [5] and the volume-of-fluid (VOF)-family of methods. Both versions track the free surface with volume markers. In MAC-class methods, the markers are Lagrangian massless particles that move with the local fluid velocity to update the free surface front. The VOF-family methods employ an auxiliary function, namely the volume fraction or color function, as the volume marker. This auxiliary function is then advected with the local velocity field to simulate the free surface propagation. One important procedure in the VOF algorithms is that the surface has to be reconstructed in terms of the volume fraction. The choice of different reconstructed interface geometry distinguishes members of the VOF-family, e.g. SLIC (simple line interface calculation) method by Noh and Woodward [6], SOLA-VOF by Hirt and Nichols [7], PLIC (piecewise linear interface calculation) by Youngs [8], etc. VOF-family methods have been widely used in both academic and industrial communities because of its relatively simple implementation and robustness in tackling varieties of interface topologies.

There are a large class of numerical methods based on the surface-capturing approach. Among them are artificial compressibility method, phase field methods and level set method, etc. The scalar functions used to implicitly capture the free surface are density in the artificial compressibility method [9], order parameter in the phase field method [10] and the level set function in the level set method (LSM) devised by Osher and Sethian [11]. The original notion of LSM is to define a smooth (at least Lipschitz continuous) function $\phi(\mathbf{x},t)$ (level set) that represents the interface at $\phi(\mathbf{x},t)=0$ (zero level set) [12]. The level sets are advected by the local velocity field. The interface can be captured at any time by locating the zero level set, which alleviates the burden of increasing grid resolution at the interface in many other numerical approaches. Usually, ϕ is defined as a signed distance function to the interface. This provides the great convenience of handling topological merging, breaking and even self-intersecting of interfaces in a natural way by taking advantage of the smoothness of the level set function. Information about the interface, such as orientation and curvature, can be conveniently obtained by examining the zero level set so that surface tension can be accurately estimated. Surface tension can be either diffused over the interface as a δ-function-like volume force in the momentum equations [13] or exactly treated as a jump condition incorporated in the pressure Poisson equation [14, 15]. Another advantage of LSM is that the extension from two to three dimensions is straightforward. LSM has been applied widely in incompressible fluid mechanics [13, 16–19], detonation shock dynamics [20], combustion [21], solidification [22], crystal growth [23], boiling [24], etching and deposition [25], to name just a few applications.

In this paper, we present a numerical method that couples three-dimensional (3D) incompressible Navier–Stokes equations with LSM in a generalized curvilinear co-ordinate system for study of free surface flows. The governing equations are discretized by the finite volume method on a non-staggered grid with a four-step fractional step method. We compare different numerical schemes for the advection of the level set function in curvilinear co-ordinates, including the third-order QUICK scheme, the second-order ENO scheme and the third-order ENO scheme. The numerical schemes for the level set equations of evolution and reinitialization are validated with benchmark cases, including a stationary circle, a rotating slotted disk and stretching of a circular fluid element. We show the necessity of the third-order ENO for the accurate capturing of the interface based on the results of benchmark tests. We then apply and validate the numerical code in several benchmark problems, e.g. a travelling solitary wave, and two-dimensional (2D) and 3D dam breaking problems. In particular, in the 3D dam breaking problem, different boundary conditions in the spanwise direction are applied to study air entrapment and splashing of the water surge front.

This paper is organized as follows. Section 2 describes the smoothing method of the two-phase flow system, the coupling of Navier–Stokes equations with LSM, and the reinitialization procedure for the level set function and mass conservation. Section 3 presents the Navier–Stokes solver and numerical methods for the level set function. Section 4 starts with the single-phase lid-driven cavity flow for the validation of the four-step fractional step method. The numerical schemes for level set function are then validated by means of benchmark tests, such as Zalesak's problem and the stretched fluid in a swirling shear velocity field. Finally, the coupled system is applied to three free-surface flows: propagation of a solitary wave, and breaking of 2D and 3D dams.

2. MATHEMATICAL MODEL

2.1. Smoothing of two-phase flow system

In the level set method, free surface flows are modelled as immiscible gas–liquid two-phase flows. However, the sharp jumps in density and viscosity at gas–liquid interfaces can cause numerical instabilities if not treated properly. To ease this problem, fluid properties, such as density, viscosity, etc., are smeared over a narrow transition zone around the free surface. The free surface is identified as a zero level set, i.e. $\phi(\mathbf{x},t)=0$, where $\mathbf{x}=(x,z)$ in two dimensions or (x,y,z) in three dimensions.

At a free surface, there exist kinematic and dynamic boundary conditions. The kinematic boundary condition can be interpreted in a Lagrangian way: a particle on the surface always stays on the surface. This can be expressed in terms of the advection of the level set function,

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{1}$$

where $\mathbf{u} = (u, w)$ in two dimensions or (u, v, w) in three dimensions is the fluid velocity. Free surface motion is represented by the propagation of the zero level set embedded in the equation.

The dynamic boundary condition represents the jump in the normal stress at the free surface balanced by surface tension, known as Laplace–Young equation,

$$(\mathbf{S}_\mathrm{l} - \mathbf{S}_\mathrm{g}) \cdot \mathbf{n} = \sigma \kappa \mathbf{n} \tag{2}$$

where $\mathbf{S} = -p\mathbf{I} + 2\mu\mathbf{D}$ is the stress tensor, $\mathbf{I}$ is the identity matrix, μ is the fluid viscosity, $\mathbf{D}$ is the rate of deformation tensor, $\mathbf{n}$ is the unit normal to the free surface, $\boldsymbol{\sigma}$ is the coefficient of surface tension, κ is the total curvature of the free surface, and the subscripts l and g represent liquid and gas, respectively.

Similar to the approaches employed by Brackbill *et al.* [26] and Unverdi and Tryggvason [4], Equation (2) is implemented in the momentum equations as a volume force by distributing the surface tension smoothly over the transition zone so that there is no longer a jump in the normal stress at the free surface. The two-phase flow system can then be treated as a single-fluid system by applying the single set of Navier–Stokes equations in the whole computational domain.

The level set function ϕ is initially assigned with a signed distance function,

$$\phi = \begin{cases} -d & \text{for} \quad \mathbf{x} \in \boldsymbol{\Omega}_{\mathrm{gas}} \\ 0 & \text{for} \quad \mathbf{x} \in \boldsymbol{\Gamma} \text{ (free surface)} \\ d & \text{for} \quad \mathbf{x} \in \boldsymbol{\Omega}_{\mathrm{liquid}} \end{cases} \tag{3}$$

where d is the absolute normal distance to the free surface. For immiscible incompressible fluids, the density and viscosity are taken as constant along the trajectories of fluid particles, i.e.

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = 0, \quad \frac{\mathrm{D}\mu}{\mathrm{D}t} = 0 \tag{4}$$

where $\mathrm{D}/\mathrm{D}t \equiv \partial/\partial t + \mathbf{u} \cdot \nabla$ is the material derivative. Numerical instability may be induced by the direct discretization of Equation (4) because of large density and viscosity jumps, particularly the density jump. This problem can be eased by smoothing out the density and viscosity in the transition zone defined as $|\phi| \leqslant \varepsilon$, where ε, the half-thickness of interface, is typically one or two grid distances. By defining an infinitely differentiable smoothed Heaviside function $H(\phi)$ [13],

$$H(\phi) = \begin{cases} 0 & \text{if } \phi < -\varepsilon \\ \frac{1}{2}[1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi} \sin(\frac{\pi\phi}{\varepsilon})] & \text{if } |\phi| \leqslant \varepsilon \\ 1 & \text{if } \phi > \varepsilon \end{cases} \tag{5}$$

The density and viscosity are smoothed in such a way that they are $(\rho_l+\rho_g)/2$ and $(\mu_l+\mu_g)/2$ at the surface front ($\phi=0$), respectively, and

$$\begin{aligned}\rho(\phi) &= \rho_g + (\rho_l - \rho_g)H(\phi)\\ \mu(\phi) &= \mu_g + (\mu_l - \mu_g)H(\phi)\end{aligned} \tag{6}$$

The surface tension is spread over the transition zone as a δ-function-like volume force in the momentum equations [16],

$$\sigma\kappa(\phi)\delta(\phi)\mathbf{n}$$

where $\mathbf{n}$ and $\kappa(\phi)$ can be computed in terms of ϕ,

$$\begin{aligned}\mathbf{n} &= \left.\frac{\nabla\phi}{|\nabla\phi|}\right|_{\phi=0}\\ \kappa(\phi) &= \nabla\cdot\mathbf{n} = \nabla\cdot\left.\frac{\nabla\phi}{|\nabla\phi|}\right|_{\phi=0}\end{aligned} \tag{7}$$

and the delta function $\delta(\phi)$ is obtained by taking the gradient of the smoothed Heaviside function

$$\delta(\phi) = \nabla_\phi H(\phi) = \begin{cases} 0 & \text{if } |\phi| > \varepsilon \\ \frac{1}{2\varepsilon}[1+\cos(\frac{\pi\phi}{\varepsilon})] & \text{if } |\phi| \leqslant \varepsilon \end{cases} \tag{8}$$

Thus, the kinematic and dynamic boundary conditions at free surface are automatically embedded in the formulation of LSM.

2.2. Coupling of Navier–Stokes equations with level set function

To model immiscible incompressible free surface flows in complex geometries, we consider the incompressible Navier–Stokes equations in boundary-fitted curvilinear co-ordinates in which the boundaries of the physical domain are accurately represented and the boundary conditions are simply applied in a transformed computational domain. Another advantage of the curvilinear system is that numerical fluxes can be conveniently estimated for non-orthogonal grids. To couple with LSM, the incompressible Navier–Stokes equations are modified with variable density and viscosity and include a volume force to represent the surface tension:

$$\frac{\partial U_m}{\partial \xi_m} = 0 \tag{9}$$

$$\begin{aligned}\frac{\partial(J^{-1}u_i)}{\partial t} + \frac{\partial(U_m u_i)}{\partial \xi_m} = &-\frac{1}{\rho(\phi)}\frac{\partial}{\partial \xi_m}\left(J^{-1}\frac{\partial \xi_m}{\partial x_i}p\right) - J^{-1}g_i - J^{-1}\frac{\sigma\kappa(\phi)\delta(\phi)\nabla\phi}{\rho(\phi)}\\ &+\frac{1}{\rho(\phi)}\frac{\partial}{\partial \xi_m}\left(\mu(\phi)T^{mn}\frac{\partial u_i}{\partial \xi_n}\right)\\ &+\frac{1}{\rho(\phi)}\frac{\partial}{\partial \xi_m}\left(J^{-1}\frac{\partial \xi_m}{\partial x_j}\frac{\partial \xi_n}{\partial x_i}\mu(\phi)\frac{\partial u_j}{\partial \xi_n}\right)\end{aligned} \tag{10}$$

where u_i is the velocity component in Cartesian space, J^{-1} is the inverse of the Jacobian defined as

$$\det\left(\frac{\partial x_i}{\partial \xi_m}\right)$$

in which x_i and ξ_m are the components of Cartesian co-ordinates and curvilinear co-ordinates, respectively, and det represents determinant, g_i is the gravitational acceleration component in i-direction. U_m is the volume flux normal to the surface of constant ξ_m defined as

$$J^{-1}\frac{\partial \xi_m}{\partial x_j}u_j$$

T^{mn} is called the mesh skewness tensor,

$$J^{-1}\frac{\partial \xi_m}{\partial x_j}\frac{\partial \xi_n}{\partial x_j}$$

The second to last term on the right-hand side of Equation (10) is the primary viscous force, while the last term is the subsidiary viscous force due to the variability of viscosity and exists only in the transition zone. The Cartesian velocity u_i is kept as a dependent variable such that Equation (10) is in a conservative form, as suggested by Zang *et al.* [27]. This facilitates discretization of the equations and also eliminates the extra source terms resulting from the introduction of contravariant velocity.

The curvilinear form of the evolution equation of the level set function, Equation (1), reads

$$\frac{\partial(J^{-1}\phi)}{\partial t}+\frac{\partial(U_m\phi)}{\partial \xi_m}=0 \tag{11}$$

The motion of the free surface is then embedded in this equation. As the information of level set function is needed only within the transition zone, it is not necessary to solve Equation (11) in the whole domain; we solve it only within a narrow band around free surface. For this reason, it is also called the 'narrow band' approach [28].

In summary, LSM is coupled with the variable fluid property Navier–Stokes equations by solving Equations (6), (7), (10) and (11) together.

2.3. Reinitialization of level set function and mass conservation

A free surface can be mathematically parameterized as a signed distance function ($|\nabla\phi|=1$). The properties of a free surface, such as the unit normal, curvature, etc., can be derived from the signed distance function. While ϕ is initialized as a signed distance function from the free surface, Equation (11) does not ensure ϕ as a signed distance function as time proceeds. Equation (1) can actually be written in a Hamilton–Jacobi type of equation:

$$\frac{\partial \phi}{\partial t}+u_n|\nabla\phi|=0 \tag{12}$$

where $u_n=\mathbf{u}\cdot\mathbf{n}$.

However, Equation (12) does not preserve the distance function. In a complex non-uniform flow field, it is possible for ϕ to develop steep gradients from Equation (12), especially when the free surface itself has a steep slope. As a consequence, it becomes difficult to maintain a finite thickness of the transition zone. The computation of the unit normal and curvature [Equation (7)] is no longer accurate, and the surface tension term becomes a source of numerical instabilities. For the same reason, the Heaviside function and the delta function are not in a well-behaved shape any more, which severely distorts the distribution of density and viscosity in the transition zone. Consequently, a significant loss or gain of mass occurs, and conservation of mass breaks down.

One cure for this problem is to perform a reinitialization (or redistancing) procedure for ϕ to recover $|\nabla\phi| = 1$. There are basically two algorithms: PDE (partial differential equation)-based approach [13] and geometry-based Fast Marching Method [29] firstly proposed by Tsitsklis for optimal trajectory [30]. The former solves a non-linear PDE to steady state by an iterative method. The latter solves the Eikonal equation by making use of the efficient Huygens' principle. Both algorithms have achieved great success in redistancing. We apply the PDE-based algorithm in this paper.

The signed distance function can be obtained by solving for the steady solution of the following PDE

$$\begin{gathered}\frac{\partial d}{\partial \tau} + s(d_0)(|\nabla d| - 1) = 0 \\ d_0(\mathbf{x}, 0) = \phi(\mathbf{x}, t)\end{gathered} \tag{13}$$

where $d(\mathbf{x}, \tau)$ shares the same zero level set with $\phi(\mathbf{x}, t)$, τ is an artificial time, $s(d_0)$ is the smoothed sign function defined as [31]

$$s(d_0) = \frac{d_0}{\sqrt{{d_0}^2 + (|\nabla d_0|\varepsilon)^2}} \tag{14}$$

where ε usually is one grid distance.

Equation (13) is a non-linear hyperbolic equation and can be recast as

$$\frac{\partial d}{\partial \tau} + \mathbf{F} \cdot \nabla d = s(d_0) \tag{15}$$

where $\mathbf{F} = s(d_0)\nabla d/|\nabla d|$ is the characteristic velocity pointing outward from the free surface so that redistancing always starts from the free surface. Since we apply a narrow band LSM here, we only need to obtain the signed distance function within the transition zone, i.e. only $\varepsilon/\Delta\tau$ iteration steps are needed, where $\Delta\tau$ is the artificial time step. If $\Delta\tau$ takes a quantity equal to one grid distance, only one or two iterations are needed depending on the transition zone width.

In Equation (13), the free surface captured by the zero level set does not move during the reinitialization procedure in theory because $s(0) = 0$. However, this is not guaranteed in numerical implementation. Mass error may be induced during the redistancing procedure. A remedy is to preserve the volume in each cell for one fluid by adding a volume constraint in Equation (13). The modified equation becomes [32]

$$\frac{\partial d}{\partial \tau} + s(d_0)(|\nabla d| - 1) = C\delta(d)|\nabla d| \tag{16}$$

where

$$C = \frac{-\int_{\Omega_c} s(d_0)(1 - |\nabla d|)\delta(d)\, \mathrm{d}\Omega}{\int_{\Omega_c} \delta^2(d)|\nabla d|\, \mathrm{d}\Omega}$$

Ω_c is the cell volume. After redistancing by Equation (16), ϕ is re-assigned with d and the next time step starts.

3. NUMERICAL METHODS

3.1. Solver for Navier–Stokes equations

A finite volume method is employed to solve the Navier–Stokes Equations (10) on a non-staggered grid, in which u_i, p, ρ, μ, κ and $\mathbf{n}$ are all defined at cell centres, except that U_m is defined at the centres of cell faces to ensure strong pressure–velocity coupling and enforce the mass conservation in each cell through Equation (9). A semi-implicit scheme is used for time marching of Equation (10), in which Crank–Nicolson scheme is used to advance the diagonal part of the primary viscous terms, while all other terms, including the convective terms, the surface tension terms, the off-diagonal part of the primary viscous terms and the subsidiary viscous terms, are all marched using the second-order Adams–Bashforth scheme. The resulting formula is

$$u_i^{n+1} - u_i^n = \frac{\Delta t}{J^{-1}}\left[\left(\frac{3}{2}E_i^n - \frac{1}{2}E_i^{n-1}\right) + G_i(p^{n+1}) + \frac{1}{2}D_\mathrm{d}(u_i^{n+1} + u_i^n)\right] \tag{17}$$

where $E_i^n = C_i^n + D_\mathrm{o}(u_i^n) + B_i^n + D_\mathrm{s}(u_i^n)$, C_i represents the i-component of the convective terms, D_d and D_o are the respective diagonal and off-diagonal diffusion operators of the primary viscous terms divided by the density, B_i represents the i-component of gravity acceleration and surface tension forces, D_s is the diffusion operator of the subsidiary viscous force divided by the density, and G_i is the i-component of the negative gradient operator divided by the density.

The convective term C_i is discretized with the modified QUICK scheme of Perng and Street [33] in space in which the upwind direction is determined by the volume flux U_m. The spatial central difference is applied to all diffusion terms. Equation (17) is solved with a four-step fractional step method in the following procedures:

Predictor:

$$\left(I - \frac{\Delta t}{2J^{-1}}D_\mathrm{d}\right)(u_i^* - u_i^n) = \frac{\Delta t}{J^{-1}}\left[\left(\frac{3}{2}E_i^n - \frac{1}{2}E_i^{n-1}\right) + G_i(\psi^n) + D_\mathrm{d}(u_i^n)\right] \tag{18}$$

where u_i^* is the first intermediate velocity, ψ is a pressure-like variable. The operator $(I - (\Delta t/2J^{-1})D_\mathrm{d})$ is approximated by the approximate factorization technique [27] such that the above linear equations can be solved with the tridiagonal-matrix algorithm (TDMA) or periodic TDMA depending on boundary conditions.

First corrector:

$$\hat{u}_i - u_i^* = -\frac{\Delta t}{J^{-1}}G_i(\psi^n) \tag{19}$$

where $\hat{u}_i$ is the second intermediate velocity.

Solving for the pressure-like variable:

$$\frac{\delta}{\delta \xi_1}\left(\frac{T^{lm}}{\rho(\phi)}\frac{\delta \psi^{n+1}}{\delta \xi_m}\right)=\frac{1}{\Delta t}\frac{\delta \hat{U}_1}{\delta \xi_1} \tag{20}$$

where δ is a differencing operator. This is a variable-coefficient Poisson equation, which is solved with a multigrid technique [33]. The inner gradient operator is estimated at the cell faces while the outer divergence operator is estimated at the cell centre.

Second corrector:

$$u_i^{n+1}-\hat{u}_i=\frac{\Delta t}{J^{-1}}G_i(\psi^{n+1}) \tag{21}$$

The above procedure is also called the four-step time advancement scheme by Choi and Moin in Reference [34] in which all the spatial derivatives are approximated with the second-order central differencing on a staggered grid. As compared with the fractional step method by Kim and Moin [35], the pressure gradient is added to the prediction step Equation (18) and one more correction step Equation (19) is used here. As a result, the relationship between the pressure-like variable ψ and the true pressure p becomes

$$\begin{aligned} G_i(p^{n+1}) &= G_i(\psi^{n+1})-\frac{\Delta t}{2J^{-1}}D_{\mathrm{d}}G_i(\psi^{n+1}-\psi^n)\\ &= G_i(\psi^{n+1})+O(\Delta t^2) \end{aligned} \tag{22}$$

In general, the implicit time advancement of the diffusion terms results in a splitting error in fractional step methods, determining the global order of time accuracy. Equation (22) implies that the pressure-like function ψ in this four-step fractional step method is of the second-order temporal accuracy of the real pressure p. The splitting error is as well second-order accurate in time, seen by summing Equations (18), (19) and (21). For a general three-step fractional step method, e.g. Kim and Moin [35] and Perot [36], the pressure-like function ψ is only of the first-order temporal accuracy of the pressure. In addition, the relationship between the first intermediate velocity and the true velocity obtained by summing Equations (19) and (21) reads

$$\begin{aligned} u_i^{n+1} &= u_i^* + \frac{\Delta t}{J^{-1}}G_i(\psi^{n+1}-\psi^n)\\ &= u_i^* + O(\Delta t^2) \end{aligned} \tag{23}$$

Equation (23) implies that the physical boundary velocity can be used as the boundary condition of u_i^* in solving Equation (18) without degrading the overall time accuracy. For a general three-step fractional step method. e.g. Kim and Moin [35] and Perot [36], the boundary value of u_i^* has to satisfy a modified equation to maintain the second-order accuracy. A recent improvement of three-step fractional step methods by Brown *et al.* [37] achieves fully second-order temporal accuracy by adequately coupling the approximation of pressure gradient in the prediction step, pressure update equation and boundary condition. Three conclusions about the four-step fractional step method can be drawn. First, the splitting error is of the second order in time. Second, the pressure-like variable is of the second-order temporal accuracy of the real pressure. Third, there is no special treatment on the boundary condition of u_i^* to maintain a consistent second-order temporal accuracy.

3.2. Numerical methods for level set function

As mentioned above, Equation (11) is a Hamilton–Jacobi type equation. Discontinuities in derivatives are easily produced even if the initial conditions are smooth, and furthermore, such solutions are in general not unique. Numerical schemes must be specifically designed to converge to a unique viscosity solution that satisfies the entropy condition singling out a physically generalized solution [38]. The key idea to the entropy condition is monotonicity preserving. Such schemes include TVD (total variation diminishing), ENO and weighted ENO (WENO). TVD schemes generally degenerate to first-order accuracy at smooth extrema while ENO and WENO maintain the global high-order schemes [39]. The ENO scheme was originally developed for hyperbolic conservation laws by Harten *et al.* [40] and later extended to Hamilton–Jacobi equations by Shu and Osher [41] motivated by the observation of the close relationship between conservation laws and Hamilton–Jacobi equations. The basic idea of ENO is to choose the locally 'smoothest' stencil among several candidates to approximate the numerical fluxes at the cell faces, so that the numerical viscosity is adjusted adaptively by measuring the local smoothness of the solution to eliminate the Gibbs phenomenon, i.e. spurious oscillation, near the discontinuity. ENO schemes maintain a uniform high order accuracy even at a discontinuity. The higher order scheme of ENO is obtained inductively on the lower order ENO by making use of a hierarchy of divided differences, and that makes the implementation of high order ENO schemes rather straightforward. The multidimensional ENO scheme can be conveniently extended from the one-dimensional ENO scheme in a dimension-by-dimension way.

Equation (11) is advanced with the third-order TVD Runge–Kutta scheme by Shu and Osher [41] which is TV (total variation) stable.

$$\begin{aligned}
\phi^{(1)} &= \phi^n - \frac{\Delta t}{J^{-1}} R(\phi^n) \\
\phi^{(2)} &= \frac{3}{4}\phi^n + \frac{1}{4}\phi^{(1)} - \frac{\Delta t}{4J^{-1}} R(\phi^{(1)}) \\
\phi^{n+1} &= \frac{1}{3}\phi^n + \frac{2}{3}\phi^{(2)} - \frac{2\Delta t}{3J^{-1}} R(\phi^{(2)})
\end{aligned} \tag{24}$$

where $R(\phi) = \delta(U_i\phi)/\delta\xi_i$.

Let (U, V, W) and (ξ, η, ζ) denote the components of U_i and ξ_i, where $i = 1$, 2, and 3, respectively. We set the grid distance on the transformed computational grid to unity in all dimensions, i.e. $\Delta\xi = \Delta\eta = \Delta\zeta = 1$. The spatial operator R is discretized for the control volume (i, j, k) in a conservative manner,

$$\begin{aligned}
\frac{\delta(U_m\phi)}{\delta\xi_m} = {} & (U\phi)_{i+1/2,j,k} - (U\phi)_{i-1/2,j,k} + (V\phi)_{i,j+1/2,k} - (V\phi)_{i,j-1/2,k} \\
& + (W\phi)_{i,j,k+1/2} - (W\phi)_{i,j,k-1/2}
\end{aligned} \tag{25}$$

The volume fluxes U, V and W are defined at the cell faces, whereas ϕ is defined at the cell centre. Thus, the cell face values of ϕ are constructed by the third-order ENO interpolation scheme [41].

Denote

$$\delta\phi_i^- = \phi_{i,j,k} - \phi_{i-1,j,k}, \quad \delta\phi_i^0 = \phi_{i+1,j,k} - \phi_{i,j,k}, \quad \delta\phi_i^+ = \phi_{i+2,j,k} - \phi_{i+1,j,k}$$
$$\delta^2\phi_i^- = \phi_{i-2,j,k} - 2\phi_{i-1,j,k} + \phi_{i,j,k}, \quad \delta^2\phi_i^0 = \phi_{i-1,j,k} - 2\phi_{i,j,k} + \phi_{i+1,j,k}$$
$$\delta^2\phi_i^+ = \phi_{i,j,k} - 2\phi_{i+1,j,k} + \phi_{i+2,j,k}, \quad \delta^2\phi_i^{++} = \phi_{i+1,j,k} - 2\phi_{i+2,j,k} + \phi_{i+3,j,k}$$

The second-order ENO is formulated as

$$\begin{aligned}\phi^{(2)}_{i+1/2,j,k} = \phi^{\text{up}}_{i+1/2,j,k} &+ 1/2 \max[\text{sign}(U_{i+1/2,j,k}),0]m(\delta\phi_i^-,\delta\phi_i^0)\\ &+ \tfrac{1}{2}\min[\text{sign}(U_{i+1/2,j,k}),0]m(\delta\phi_i^0,\delta\phi_i^+)\end{aligned} \tag{26}$$

where

$$\phi^{\text{up}}_{i+1/2,j,k} = \begin{cases}\phi_{i,j,k} & \text{if } U_{i+1/2,j,k} \geqslant 0\\ \phi_{i+1,j,k} & \text{otherwise}\end{cases} \quad \text{is the first-order upwind value.}$$

$$m(a,b) = \begin{cases} a & \text{if } |a| \leqslant |b|\\ b & \text{otherwise}\end{cases}$$

$$\text{sign}(a) = \begin{cases} 1 & \text{if } a>0\\ 0 & \text{if } a=0\\ -1 & \text{if } a<0\end{cases}$$

The third-order ENO is formulated as

$$\begin{aligned}\phi^{(3)}_{i+1/2,j,k} = \phi^{(2)}_{i+1/2,j,k} &+ \tfrac{1}{3}\max[\text{sign}(U_{i+1/2,j,k}),0]\{\max[c_i^-,0]m(\delta^2\phi_i^-,\delta^2\phi_i^0)\\ &+ \tfrac{1}{2}\min[c_i^-,0]m(\delta^2\phi_i^0,\delta^2\phi_i^+)\}\\ &+ \tfrac{1}{3}\min[\text{sign}(U_{i+1/2,j,k}),0]\{\tfrac{1}{2}\max[c_i^+,0]m(\delta^2\phi_i^0,\delta^2\phi_i^+)\\ &+ \min[c_i^+,0]m(\delta^2\phi_i^+,\delta^2\phi_i^{++})\}\end{aligned} \tag{27}$$

where

$$c_i^- = c(\delta\phi_i^-,\delta\phi_i^0)$$
$$c_i^+ = c(\delta\phi_i^0,\delta\phi_i^+)$$
$$c(a,b) = \begin{cases} 1 & \text{if } |a| \leqslant |b|\\ -1 & \text{otherwise}\end{cases}$$

The other cell face values of ϕ, such as $\phi_{i,j+1/2,k}$, $\phi_{i,j,k+1/2}$, etc., are approximated in the same way. Clearly, Equation (16) is also a Hamilton–Jacobi equation. We apply a second-order ENO scheme as follows:

Denote

$$d_x \equiv \frac{\delta d}{\delta x}, \quad \Delta x_i^- \equiv x_{i,j,k} - x_{i-1,j,k}, \quad \Delta x_i^+ \equiv x_{i+1,j,k} - x_{i,j,k}$$

(1) *Approximation of derivatives*

First-order approximation:

$$d_x^{(1)-} = \frac{\partial\xi}{\partial x}(d_{i,j,k} - d_{i-1,j,k}) + \frac{\partial\eta}{\partial x}(d_{i,j,k} - d_{i,j-1,k}) + \frac{\partial\zeta}{\partial x}(d_{i,j,k} - d_{i,j,k-1})$$

$$d_x^{(1)+} = \frac{\partial\xi}{\partial x}(d_{i+1,j,k} - d_{i,j,k}) + \frac{\partial\eta}{\partial x}(d_{i,j+1,k} - d_{i,j,k}) + \frac{\partial\zeta}{\partial x}(d_{i,j,k+1} - d_{i,j,k})$$

Second-order approximation:

$$d_x^{(2)-} = d_x^{(1)-} + \frac{\Delta x^-}{2}\frac{\delta^2 d^-}{\delta x^2}$$

$$d_x^{(2)+} = d_x^{(1)+} - \frac{\Delta x^+}{2}\frac{\delta^2 d^+}{\delta x^2}$$

where

$$\frac{\delta^2 d^-}{\delta x^2} = \mathrm{minmod}\,(d_1, d_2)$$

$$\frac{\delta^2 d^+}{\delta x^2} = \mathrm{minmod}\,(d_2, d_3)$$

$$\mathrm{minmod}(a,b) = \begin{cases} \mathrm{sign}\,(a)\min(|a|,|b|) & \text{if } a\cdot b > 0 \\ 0 & \text{otherwise} \end{cases}$$

d_1, d_2 and d_3 are the central difference approximations of $\delta^2 d/\delta x^2 \equiv (\partial\xi_m/\partial x)\delta/\delta\xi_m((\partial\xi_n/\partial x)\ \delta d/\delta\xi_n)$ on stencils $1(x_{i-2,j,k}, x_{i-1,j,k}, x_{i,j,k})$, $2(x_{i-1,j,k}, x_{i,j,k}, x_{i+1,j,k})$ and $3(x_{i,j,k}, x_{i+1,j,k}, x_{i+2,j,k})$, respectively. $d_y^{(2)-}$, $d_y^{(2)+}$, $d_z^{(2)-}$ and $d_z^{(2)+}$ are computed in the same way.

(2) *Compute* $|\nabla d|$

Let

$$a = d_x^{(2)-}, b = d_x^{(2)+}$$

$$c = d_y^{(2)-}, d = d_y^{(2)+}$$

$$e = d_z^{(2)-}, f = d_z^{(2)+}$$

Define

$$a_+ = \max(a,0), a_- = \min(a,0)$$

and the same subscripts for b, c, d, e, and f.

The computation of $|\nabla d|$ is performed based on Godunov's method [13],

$$|\nabla d| = \begin{cases} D^+ & \text{if } s(d_0) > 0 \\ D^- & \text{if } s(d_0) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

where

$$D^+ = \sqrt{\max(a_+^2, b_-^2) + \max(c_+^2, d_-^2) + \max(e_+^2, f_-^2)}$$
$$D^- = \sqrt{\max(a_-^2, b_+^2) + \max(c_-^2, d_+^2) + \max(e_-^2, f_+^2)}$$

Apply the upwind scheme for Equation (16) to obtain

$$s(d_0)|\nabla d| = \max[s(d_0), 0]D^+ + \min[s(d_0), 0]D^- \tag{29}$$

Equation (16) is also advanced in time with the third order TVD Runge–Kutta scheme (24).

3.3. Restriction on time step

Since the convective terms, the surface tension terms, the gravity, the off-diagonal primary viscous terms and the subsidiary viscous terms in Equation (18) are advanced with the explicit scheme in time, the time step must be restricted to enforce the stability of the numerical schemes.

According to Reference [26], the time step for the surface tension reads

$$\Delta t_\sigma = \min_{\Omega_\delta} \sqrt{\frac{(\rho_l + \rho_g)\Delta h^3}{4\pi\sigma}}$$

where $\Delta h = \min(\Delta x, \Delta y, \Delta z)$ and Ω_δ represents the transition zone where $\delta(\phi) > 0$.

The time step for the convective terms must satisfy the Courant–Friedrichs–Lewy (CFL) condition:

$$\Delta t_u = \min_{\Omega} \frac{J^{-1}\mathscr{C}}{|U| + |V| + |W|}$$

where $\mathscr{C} = 0.5$ is used here and Ω represents the whole computational domain.

The time step restrictions due to the gravity and subsidiary viscous terms are

$$\Delta t_g = \min_{\Omega} \sqrt{\frac{\Delta z}{g}}$$
$$\Delta t_\mu = \min_{\Omega_\delta} \frac{\rho \Delta h^2}{2\mu}$$

The eventual restriction on the time step is then

$$\Delta t^{n+1} = \min(\Delta t_u, \Delta t_\sigma, \Delta t_g, \Delta t_\mu) \tag{30}$$

4. COMPUTATIONAL RESULTS

4.1. 2D lid-driven cavity flow

To validate the four-step fractional step method presented in Section 3.1, we compute a 2D lid-driven cavity single phase flow on a unit square with a non-uniform grid of 48×48. The boundary grid cells take the minimum grid width of 0.01, and the grid distance is expanded toward the centre lines with a stretching factor of about 2. The Reynolds number $\mathrm{Re} = U_0 h/v = 1000$, where U_0 is the velocity of the top lid, h is the cavity length and v is the kinematic viscosity of the fluid. All the computations in this paper are performed on a Dell Dimension 8200 PC with 1.8G Pentinum CPU. The computation ends after 10 000 time steps when the residual of the pressure Poisson equation becomes of the order of 10^{-9}. Streamline patterns for primary, secondary, and corner vortices are shown in Figure 1(a). They are very similar to those obtained by Ghia *et al.* [42]. The velocities along the vertical and horizontal centrelines (u and w, respectively) are compared with the benchmark data by Ghia *et al.* [42] in Figure 1(b). The agreement is excellent, showing the high accuracy of the four-step fractional step method in computing the steady vortical flow.

4.2. Evolution and reinitialization of level set function

4.2.1. A stationary circle. Reinitialization is a key procedure in LSM. To validate the numerical scheme described in Section 3.2, we apply the reinitialization procedure on a stationary

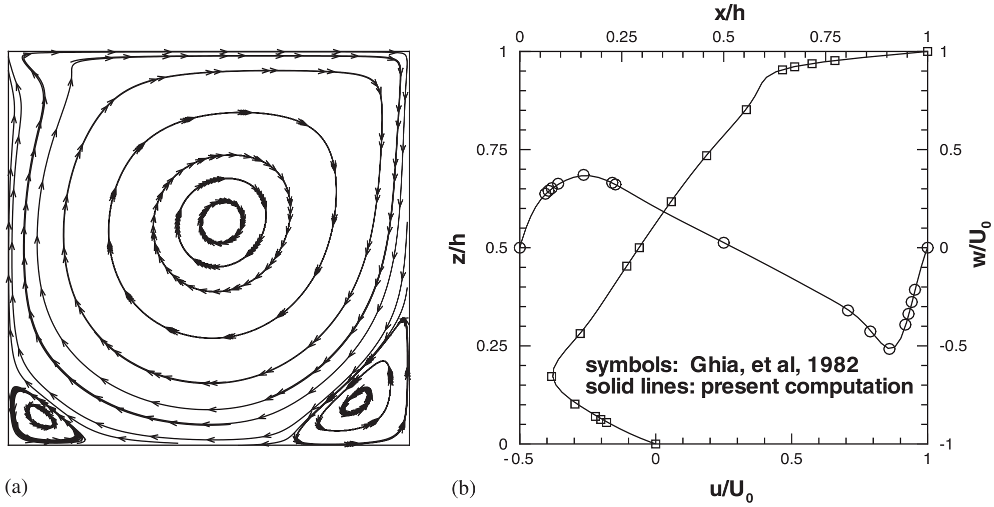

Figure 1. Lid-driven cavity flow: (a) streamline pattern for primary, secondary, and corner vortices; (b) comparison of velocity components (u, w) along the centre lines with the benchmark data.

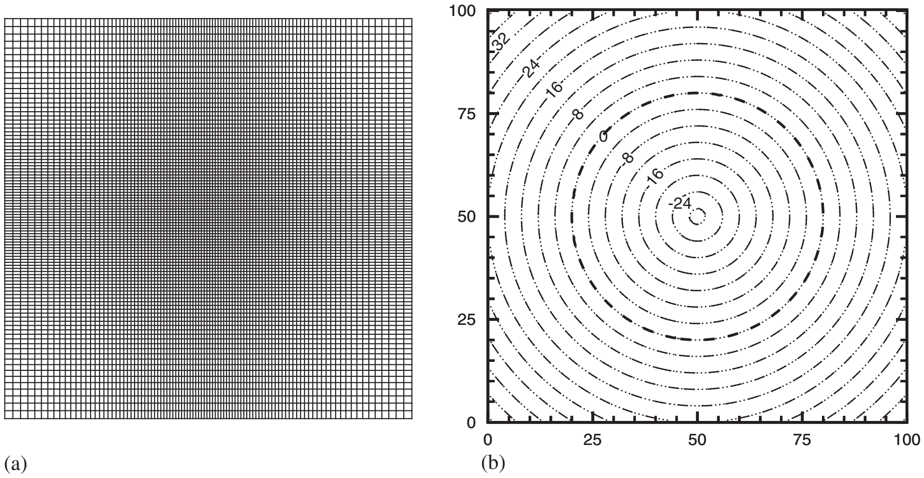

Figure 2. (a) Non-uniform grid for reinitialization of a circle; (b) contours of level sets at $t = 25$. Contour interval $\Delta\phi = 4.0$; dashed lines, uniform grid; dotted lines, non-uniform grid; thick dashed line, initial position of the circle.

circle with an initially discontinuous level set function. The domain size is 100×100. We employ a uniform grid and also the non-uniform grid shown in Figure 2(a), both having a grid size of 100×100. The centre of the circle is located at the centre of the domain and the radius of the circle is 30. The level set function is initially assigned a value of $+100$ outside the circle and -100 inside the circle. For both grids, the thickness of interface is fixed with $\varepsilon = 1$. In the uniform grid, there are exactly two grid cells in the transition zone, while the number of grid cells in the transition zone may vary in the non-uniform grid. Figure 2(b) shows the contours of ϕ at $t = 25$ time units. ϕ is fairly accurately redistanced as a signed function ($|\Delta\phi| = 1$) in the whole domain except at the centre of the circle where a singularity exists. The reinitialization performed on both grids produces the same results. The contour of $\phi = 0$ is not altered from the initial position (the thick dashed line in the figure) by the reinitialization procedure. That is just what is required for the purpose of area preservation. From this case we conclude that the accuracy of the reinitialization scheme for non-uniform grid is the same as that for a uniform grid.

4.2.2. Rotation of a slotted disk. The Zalesak's problem of a rotating slotted disk [43] has become a benchmark case for testing an advection scheme. A slotted solid disk rotates around a centre with a constant angular velocity. We use this problem to measure the diffusive error of the third-order ENO scheme on the evolution of the level set function. Here the slotted disk has a radius of 15 and a slot width of 6. It is initially located at (50,75) in the domain of size (100,100). The angular velocity Ω is set to 0.01 so that the disk returns to its original position at every $200\pi(\approx 628)$ time units. The diffusive errors can be evaluated by checking the degree of distortion of the disk boundary. For the sake of comparison, we employ three grids: a uniform grid of 100×100, a non-uniform grid of 100×100 as shown in Figure 3(a), and a refined uniform grid of 200×200. The time step $\Delta t = 0.5$ is used for the refined uniform

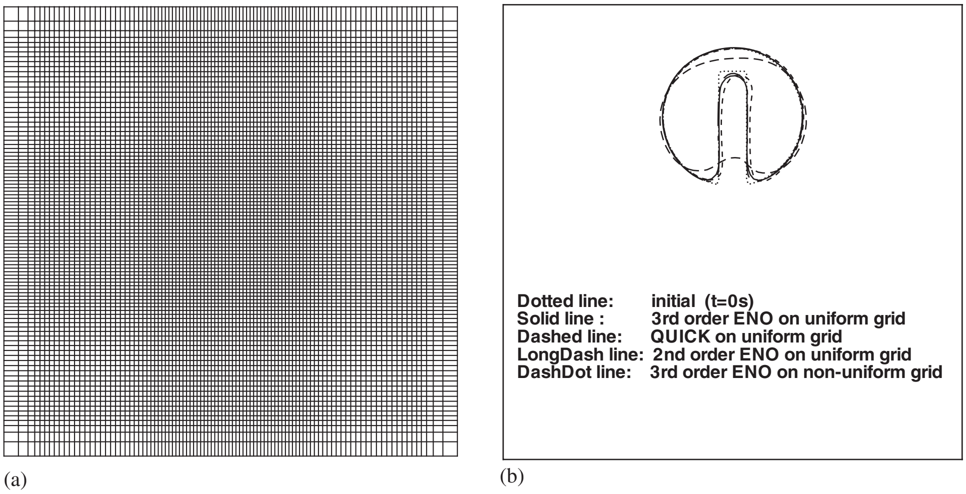

Figure 3. Zalesak's problem (rotation of a slotted disk), domain size of 100×100: (a) non-uniform grid 100×100; (b) the disk boundaries by different advection schemes after one revolution ($t = 628$), $\Delta t = 1$, uniform grid size of 100×100.

grid and $\Delta t = 1.0$ for the other two grids. For two uniform grids, the thickness of interface is fixed with two grid distance, i.e. $\varepsilon = 1$ for the 100×100 one and $\varepsilon = 0.5$ for the 200×200 one. For the non-uniform grid, the thickness of interface is fixed with $\varepsilon = 1$. Because this is a pure advection problem with a uniform velocity field, we expect that a good evolution scheme without reinitialization should adequately preserve the disk geometry. To illustrate the effect of different schemes on preserving complex geometries, we apply and compare the second-order ENO scheme of Equation (26), the third-order ENO scheme of Equation (27), and the third-order QUICK scheme for the approximation of the cell face values of ϕ in the evolution Equation (25).

Figure 3(b) shows that the second-order ENO is very diffusive because the slot is totally smeared out. Thus a higher order scheme with at least third-order accuracy is required for the evolution of complex boundaries (or interfaces). The third-order ENO evolves both the circular boundary and the slot boundary quite accurately without significant distortion except near the sharp corners. While QUICK is as well a third-order upwind scheme, the slot boundary deviates from its original position to some degree and a small distortion near the top of the slot is also observed. This is because QUICK tends to generate over- and under-shoots in the vicinity of a discontinuity. With the non-uniform grid, the advection of the disk boundary is improved at the corners and the top of the slot due to finer grid resolution. Figure 4 shows the rotation process obtained by the third-order ENO scheme on the fine grid at $t = 0$, 157, 314, 471 and 628. It is seen that the disk boundaries even in the vicinity of the corners are precisely advected. The reinitialization scheme with area preserving is also applied for comparison. A slight improvement is observed. A further improvement of this problem can be obtained by combining LSM with a marker particle Lagrangian scheme developed by Enright *et al.* [9]. A set of massless particles are placed around the interface, advected by the velocity interpolated from that of the underlying grid.

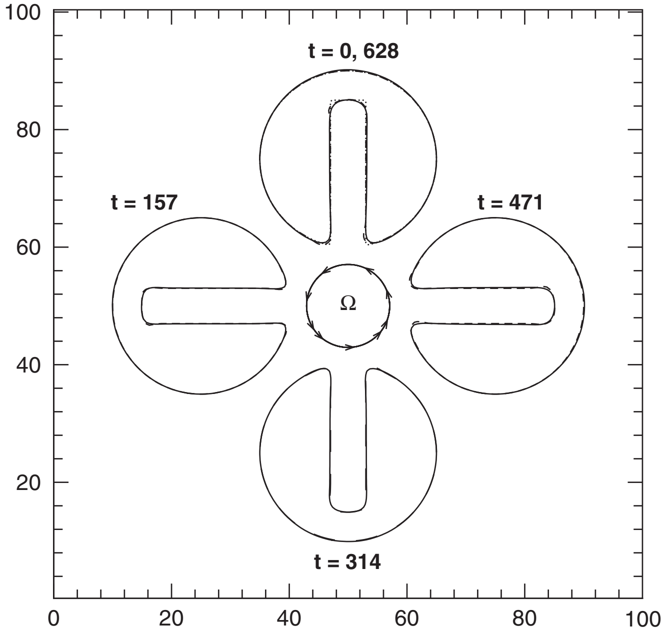

Figure 4. Zalesak's problem (rotation of a notched disk), grid 200 × 200, $\Delta x = \Delta t = 0.5$. Dotted lines, initial position of the slotted disk; solid lines, only the third-order ENO evolution scheme is applied; dashed lines, both the evolution and reinitialization schemes are applied.

4.2.3. Stretching of a circular fluid element. A circular fluid element is placed in a swirling shear flow field within a unit square described by

$$\varphi = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi z)$$

where φ is the stream function. The fluid is stretched into a thin filament by the shearing velocity field. This case provides a challenging test for surface-tracking and surface-capturing methods. Rider and Kothe [44] and Rudman [45] used it to evaluate their VOF methods. In order to make use of the three grids employed previously in the Zalesak problem, the same domain size is adopted here. The circle is initially centred at (50,75) with a radius of 15. The solenoidal velocity field becomes

$$u = -\sin^2\left(\frac{\pi x}{100}\right)\sin\left(\frac{\pi z}{50}\right), \quad w = \sin^2\left(\frac{\pi z}{100}\right)\sin\left(\frac{\pi x}{50}\right) \tag{31}$$

A time step of 0.5 is used for the two grids of 100 × 100, and 0.25 for the grid of 200 × 200. The thickness of interface is set the same as those in the respective gird.

Figure 5 shows the stretching process of the fluid element at $t = 0$, 100, 200 and 300 by the three grids. The circular fluid is torn into a filament by the shearing flow, and becomes thinner with increasing time. There is no appreciable difference by the two grids of 100 × 100 at $t = 100$ and 200, but the non-uniform grid preserves the tail of the filament slightly better than the uniform one. The uniform grid of 200 × 200 preserves the areas of the filaments much

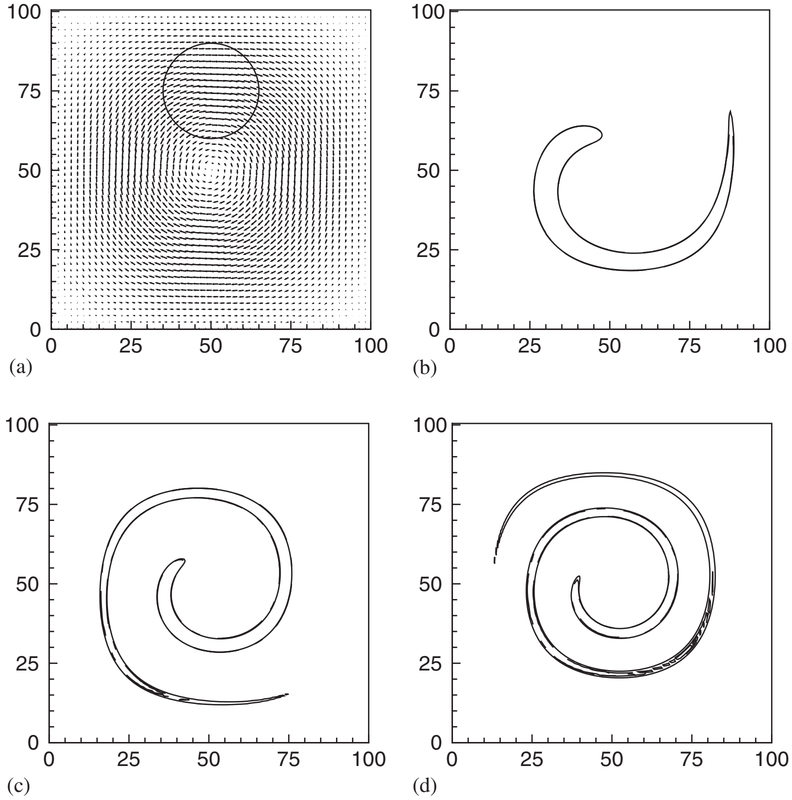

Figure 5. Stretching of a circular fluid element in a swirling deformation flow: (a) the fluid element at the initial state; (b) the stretched fluid element at $t = 100$; (c) the stretched fluid element at $t = 200$; (d) the stretched fluid element at $t = 300$; in (b) (c) and (d), solid lines, uniform grid 200×200; dashed lines, uniform grid 100×100; dashdot lines, non-uniform grid 100×100.

better than the two coarse grids. There is no significant breakup of the filaments, especially with the grid of 200×200, as was found in the results obtained with VOF methods [44, 45] due to the effect of numerical surface tension inherent in the reconstruction procedure.

For the purpose of evaluating the errors of area preserving and the accuracy of interface advection and deformation, the velocity field of Equation (31) is multiplied by $\cos(\pi t/T)$, so that the stretching process is time-reversed according to Leveque [46], where T is the prescribed reversal period. The flow slows down and the fluid is stretched out during $0 < t < T/2$. The flow reverses direction and the fluid shrinks back during $T/2 < t < T$. The fluid element is expected to resume its initial circular shape at $t = T$. We choose two periods, $T = 250$ and 500, to estimate the errors. Figure 6 shows that the fluid element in all the three grids return to the original circular shape with slight deformation after one period of $T = 250$. But for $T = 500$, only the fluid element in the grid of 200×200 nearly returns to the circular shape and those in the other two grids are deformed. This is because the fluid is stretched severely for $T = 500$ and the advection accuracy of the fluid interface degrades if grid is not fine enough. Table I

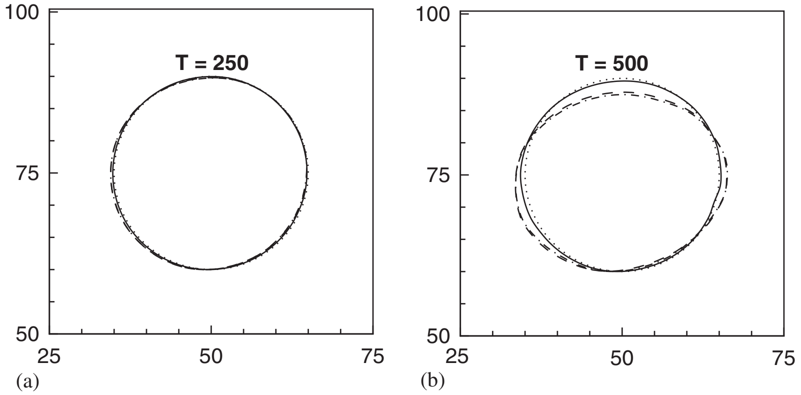

Figure 6. Reversed circular fluid element after one period T: (a) $T=250$; (b) $T=500$; dotted lines, initial contour of the circular fluid; solid lines, uniform grid 200 × 200; dashed lines, uniform grid 100 × 100; dashdot lines, non-uniform grid 100 × 100.

Table I. Area error after one period for a circular fluid in the time-reversed swirling deformation flow.

Grid	% area error ($T=250$)	% area error ($T=500$)
Uniform grid (100 × 100)	0.687	0.09
Non-uniform grid (100 × 100)	0.42	−1.635
Uniform grid (200 × 200)	0.038	1.36

shows the area errors calculated by

$$\varepsilon_A = \frac{A(t) - A(0)}{A(0)}$$

where $A(t) = \int_\Omega H(\phi)\,d\Omega$ is the total area of the fluid element at time t. The total area of the fluid element in all cases is well preserved. Although the area for the case of $T=500$ on the uniform grid of 100 × 100 appears to be preserved better than others, its interface is severely deformed. From this test, we conclude that the current LSM can resolve the stretched interface on the scale of the grid cell without bringing in significant artificial surface tension effects. A similar problem is simulated with the hybrid particle level set method by Enright *et al.* [47]. Very accurate results are observed even with a pretty coarse grid.

4.3. Applications of the coupled system for free surface flows

4.3.1. Travelling solitary wave. Propagation of a solitary wave is a simple and practical free surface problem that has been studied experimentally and numerically. We aim to ascertain whether LSM can predict the viscous damping and run-up on a vertical wall of a travelling solitary wave in a canal, as shown in Figure 7. Here, h is the depth of still water and the subscripts a and w denote air and water, respectively. The channel size is $20h \times 2h$. We set the theoretical wave speed $C_w = \sqrt{gh} = 1.0\,\text{m/s}$, the Reynolds number $\text{Re} = C_w h/\nu_w = 5 \times 10^4$, the viscosity ratio $\nu_a/\nu_w = 15$, and the density ratio $\rho_a/\rho_w = 1.2 \times 10^{-3}$ (these ratios are also

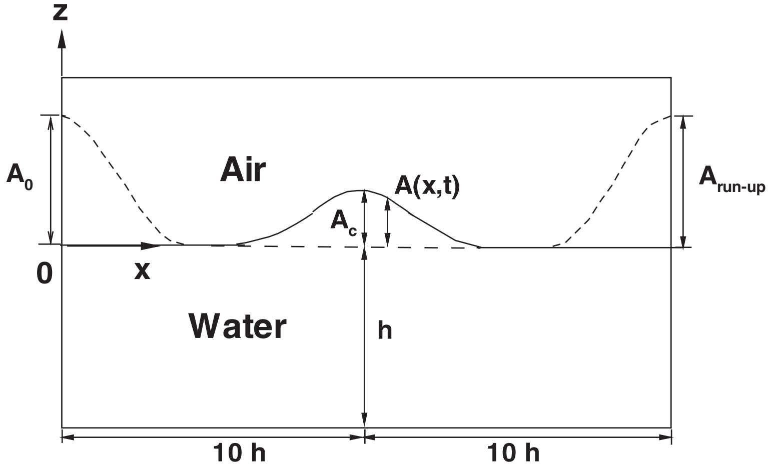

Figure 7. Illustration of the formation and travelling of a solitary wave in an enclosed channel.

employed for all subsequent free surface flow cases). A grid of 200 × 120 is used. The grid distance is uniform in the x direction, and uniform within the range $(0, A_0)$ in the z direction then expands to the top and bottom boundaries. The half-thickness of interface ε is fixed with two grid distance $2\Delta z$. To generate a solitary wave, one can make use of Laitone's analytical approximation [48]. Here we release an initially still water surface with a Boussinesq profile [49] from the left vertical wall which is in hydrostatic balance,

$$A(x,0) = A_0/\cosh^2\left(\frac{\sqrt{3A_0}}{2}x\right)$$

After $t = 6.0$ s, the wave has escaped from the influence of the left wall boundary and may be regarded as a solitary wave. This time is set as the initial time of the solitary wave propagation. Grid points cluster between $0 \leqslant z \leqslant A_0$ with an interval $\Delta z = 0.01$ h to resolve the wave. Because of the large density ratio of air and water, the top boundary condition has negligible effects on the motion of the solitary wave, and therefore no-slip boundary condition is applied for simplicity.

Figure 8(a) shows the travelling train of the solitary wave and its climb at the right vertical wall for the case $A_0/h = 0.4$. We see the slight damping of the wave amplitude due to the viscous effects. The wave speed measured from Figure 8(a) is 1.05 which is close to the theoretical value. Figure 8(b) shows a typical velocity field at $t = 4.0$ s. A vortex centred at the wave top is observed.

To quantify the viscous damping characteristics of the wave, we compute three waves with different initial amplitudes, and compare the results with those predicted by Mei's perturbation theory [50]

$$A_{\max}^{-1/4} = A_{0\max}^{-1/4} + 0.08356\left(\frac{v_{\mathrm{w}}}{C_{\mathrm{w}}^{1/2}h^{3/2}}\right)^{1/2}\frac{C_{\mathrm{w}}t}{h} \tag{32}$$

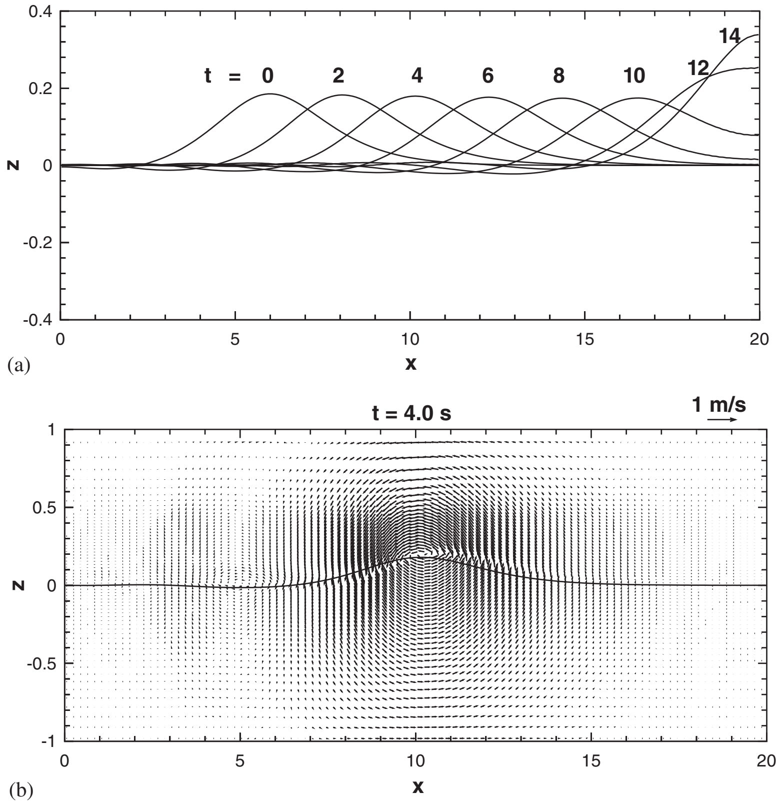

Figure 8. (a) Travelling trains of a solitary wave; (b) a typical velocity field of a solitary wave.

where $A_{\max}$ is the amplitude of the solitary wave, and $A_{0\max}$ is the amplitude at the initial state. Figure 9 shows that the smaller $A_{0\max}/h$ is, the computation agrees better with the perturbation theory. That is because Equation (32) is valid only for $A_{0\max}/h \leqslant 0.1$.

Another quantity for comparison is the wave run-up (the highest point) at the right vertical wall. We compute nine cases with different initial wave amplitudes and measure the run-ups at the right wall boundary. The computational results are compared with the experimental data by Chan *et al.* [51] in Figure 10. A_c in the x axis of the figure is the amplitude of the solitary wave in the middle of the horizontal distance of the computational domain as shown in Figure 7. The agreement between computation and experiment is very good for $A_c/h < 0.3$. After the value 0.3, the experimental data exhibit some scatter. Overall, these results demonstrate that the present LSM can accurately predict the viscous damping characteristics without introducing significant numerical damping effects.

To further quantify the numerical errors, we compute the mass error defined as

$$\varepsilon_M = \frac{M(t) - M(0)}{M(0)}$$

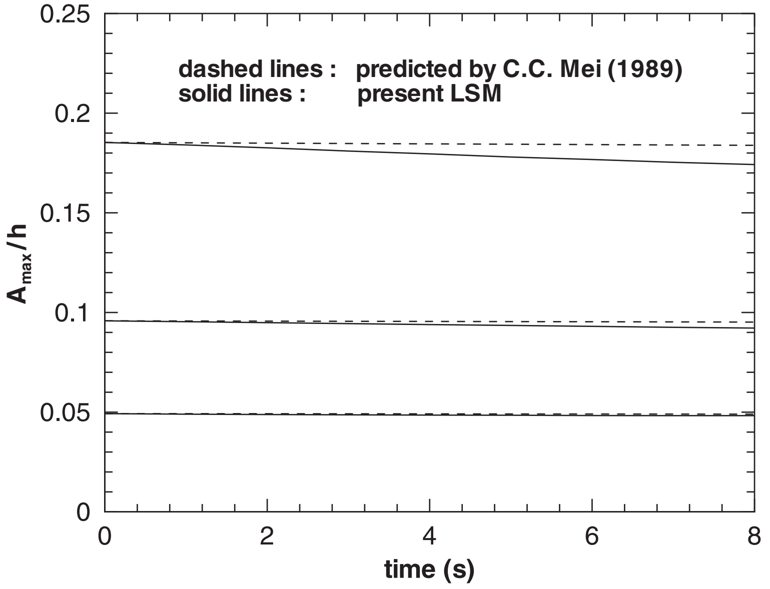

Figure 9. Damping rate of solitary waves.

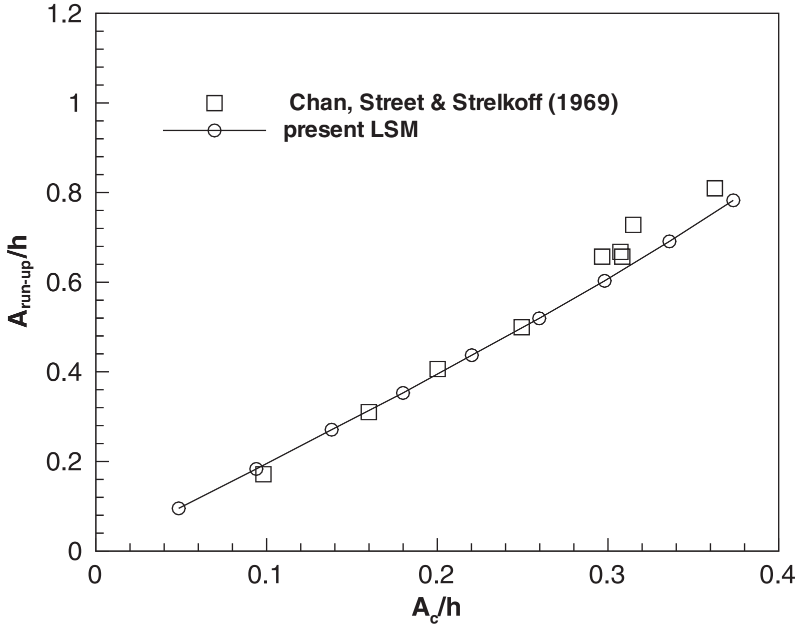

Figure 10. Wave run-up versus incident wave amplitude.

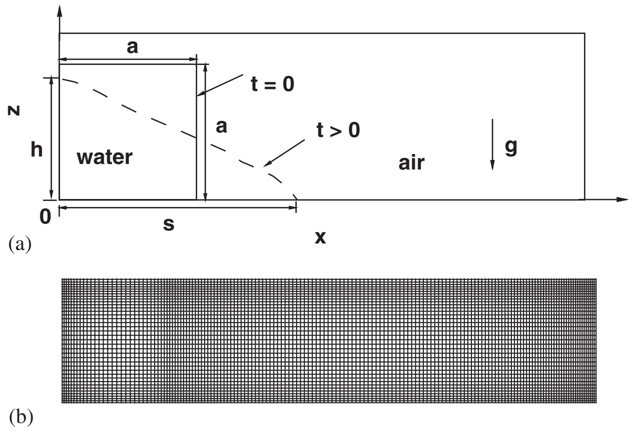

Figure 11. (a) Schematic of two-dimensional broken dam; (b) non-uniform grid for two-dimensional broken dam.

where $M(t) = \int_{\Omega} \rho H(\phi)\,\mathrm{d}\Omega$ is the total mass of water at time t. The numerical mass error is smaller than 0.01% at $t = 20\,\mathrm{s}$ for the case $A_0 = 0.4h$ and is 0.0085% for the case $A_0 = 0.2h$ at the same time. This shows that the present numerical scheme conserves mass quite well.

4.3.2. 2D dam breaking. The collapse of a water column on a rigid horizontal plane is also called a broken-dam problem. It is used to simulate the abrupt failure of a dam, in which an initially blocked still water column starts to spread out after the blocking is removed. It has been experimentally studied in detail by Martin and Joyce [52] to investigate the spreading velocity and the falling rate of water columns. The motion of the water was recorded by cine-photography at about 300 frames per second. One of their cases, a square water column with length $a = 2\frac{1}{4}$ in., is employed here to verify the present LSM. This problem was also studied by Kelecy and Pletcher [9] in their numerical simulation.

The computational domain is $5a \times 1.25a$, the same as that employed by Kelecy and Pletcher. It is sketched in Figure 11(a). s and h denote the surge front position and the remaining height of the water column, respectively. These parameters are used to measure the spreading velocity and the falling rate of the water column. Our numerical experiments are performed in a closed container with wall boundaries. We use a uniform grid of 200×50 and a non-uniform grid of 160×40 with grid points clustered near the left wall, the right wall, the bottom wall and the top and right boundaries of the initial water column as shown in Figure 11(b). The half-thickness of interface ε is fixed with two grid distance $2\Delta x$ for the uniform grid case, which is multiplied by 1.12 for the non-uniform grid case. The still water column is initially in hydrostatic balance. The surface tension effect is examined by keeping or removing the surface tension term in Equation (10). The time is non-dimensionalized by $t_g = \sqrt{a/g}$ in all the plots in 2D and 3D broken dam flows.

Figure 12(a) shows the comparison of the surge fronts between the present computations and Martin and Joyce's experiment. The effect of surface tension on the surge fronts is indistinguishable in the figure. The results with the non-uniform grid deviate from those of the uniform grid only slightly in the final stage but the difference is well within experimental

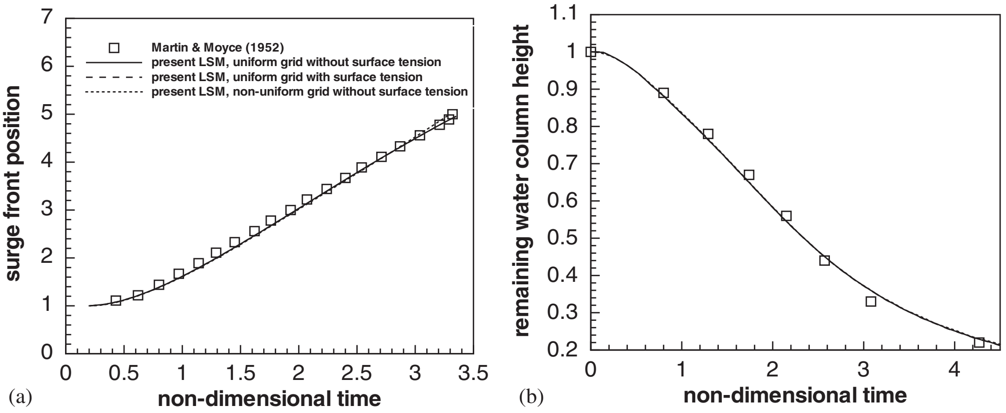

Figure 12. Two-dimensional broken dam: (a) surge front position s versus non-dimensional time; (b) remaining water column height h versus non-dimensional time.

uncertainties. Overall, the water spreading velocity is provided quite accurately by the present LSM. Figure 12(b) shows the comparison of the remaining water column height between computation and experiment. The agreement is excellent. It is seen that surface tension and the differences in the grids do not considerably affect the results.

Snapshots of water positions and the velocity fields in the whole computational domain at the selected times calculated with the uniform grid are displayed in Figure 13. The water column collapses and accelerates toward the air due to the pressure difference between the adjacent water and air along the right boundary of the initial water column. The largest pressure difference is found at the right lower corner of the water column where the water is greatly accelerated and moves rapidly along the bottom wall. Air is entrapped by the water, forming an air bubble, when the surge front is reflected from the right wall and falls into the bottom water pool (Figure 13f), and an elongated thin surge is created by the splashing of the surge front (Figure 13g). The velocity vector fields reveal that a large vortex is formed in the air region in the vicinity of the water surface and accompanies the surface motion at all times. The strongest motion (the largest velocity) always occurs in the air region in the vicinity of the surge front.

Figure 14 shows the time history of the mass errors by the uniform and non-uniform grids. In general, mass is conserved quite well considering the fast-transient surface motion and large topological changes of the free surface. Mass error can be further reduced by employing a finer grid. To study the effect of the thickness of interface on the computational results, five different thicknesses are applied on the uniform grid. Figure 15 shows the effect of the half-thickness of interface ε on the spreading speed of the surge front. It turns out that the results are not very sensitive to the choice of the thickness of interface. But there exists a tendency that a larger interface thickness leads to more deviation from the experimental data. Even though it is not shown here, there is almost no discernible discrepancy on the remaining water column height among these five thickness.

4.3.3. 3D dam breaking. To demonstrate the ability to simulate 3D free surface flows using the present LSM, we consider the breaking of a cubic water column sketched in Figure 16.

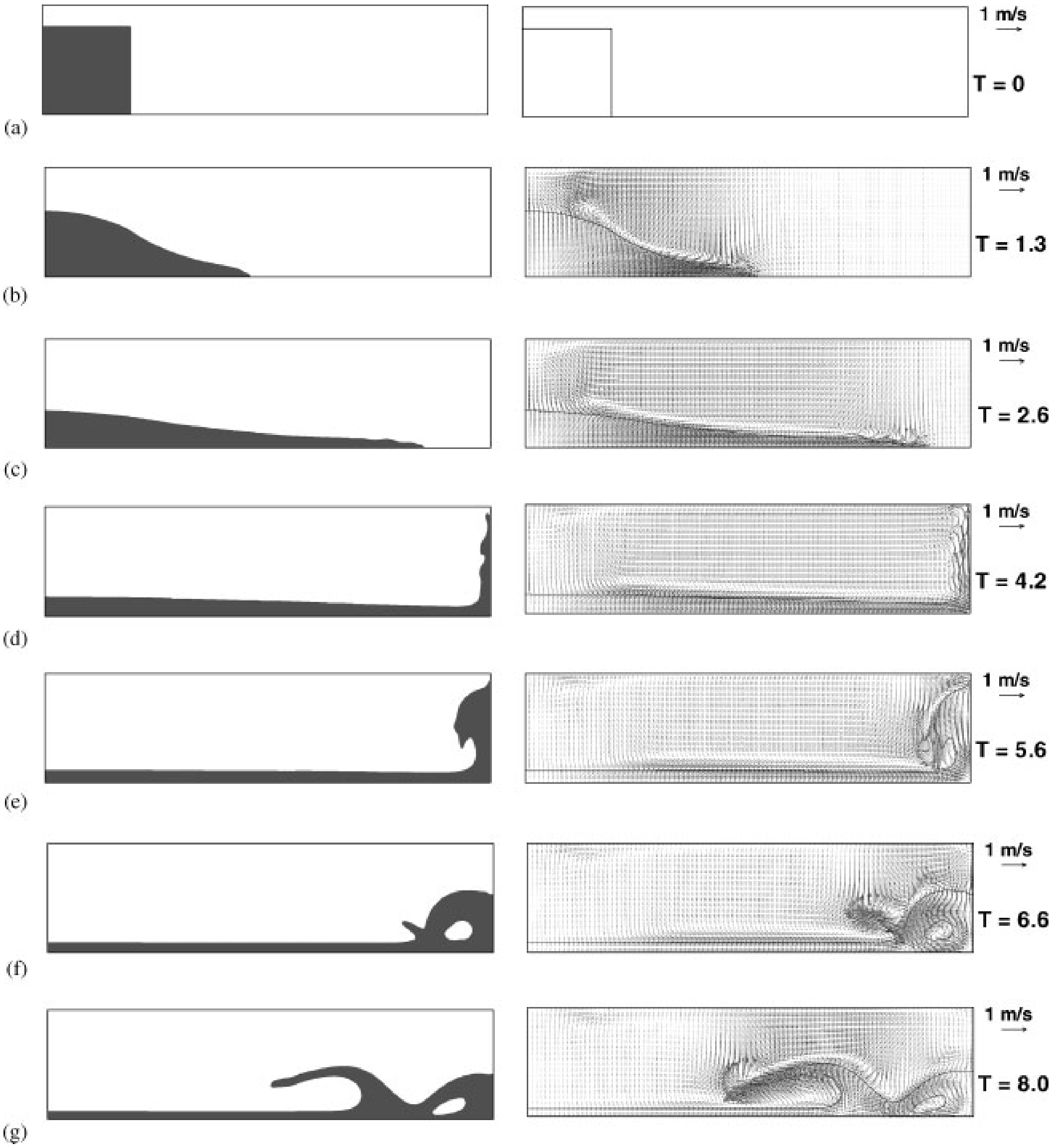

Figure 13. Free surface position (left pictures) and velocity vectors (right pictures) at selected times by the uniform grid; the shadow areas in the left figures represent the water, the lines in the right figures represent the position of free surface.

The computational domain size is $5a \times a \times 1.25a$ in the (x, y, z) directions. A uniform grid of $200 \times 24 \times 50$ is used. The thickness of interface is same as that in the 2D uniform grid case. The top, bottom, left and right boundaries of the computational domain are all prescribed as walls. We apply two types of boundary conditions for the boundaries in the spanwise (y) direction: no-slip and periodic. The former means that the computational domain is an

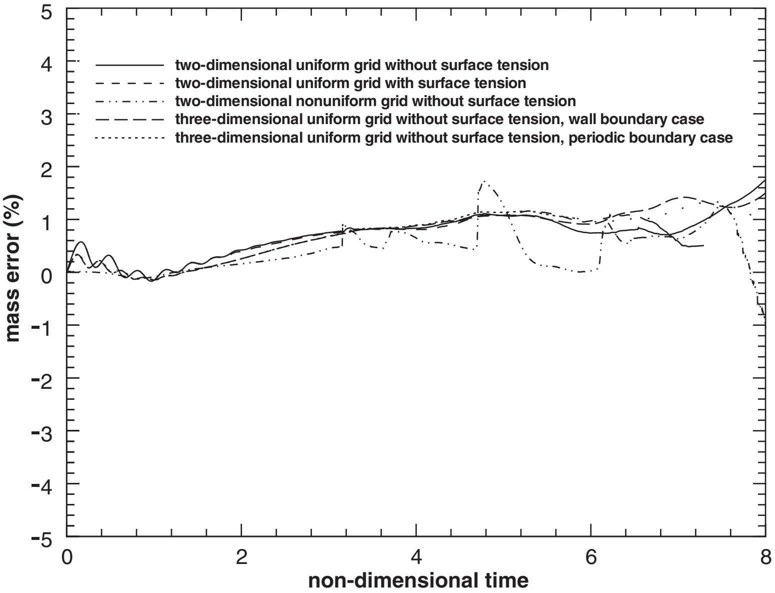

Figure 14. Mass errors of two- and three-dimensional broken dam cases versus non-dimensional time.

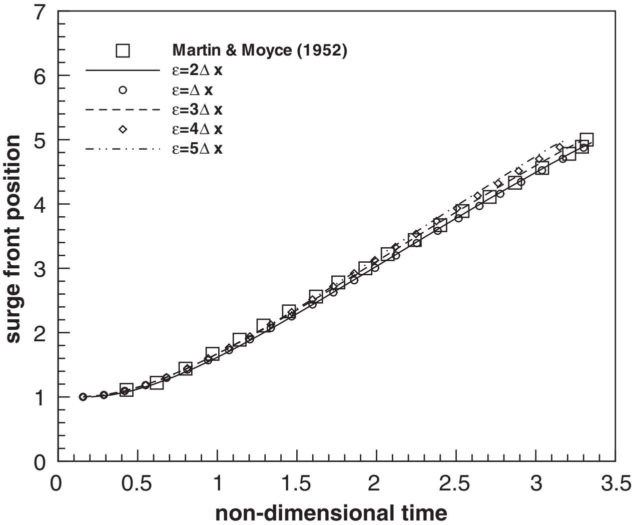

Figure 15. Effects of the thickness of interface on the spreading velocity of surge fronts, where ε is the half-thickness of interface.

enclosed container; the latter implies that the computational domain is a short segment of a wide container in which we would expect essentially 2D flow. These simulations assess the effect of the side boundary on the structure of the surge front and the entrapped air bubble.

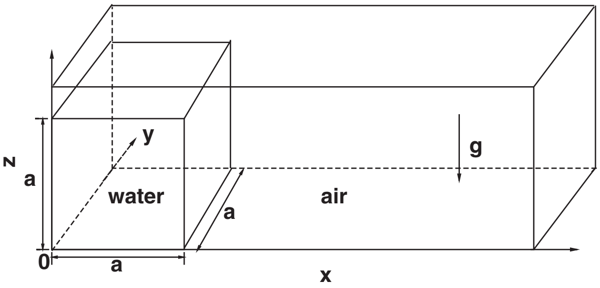

Figure 16. Schematic of three-dimensional broken dam.

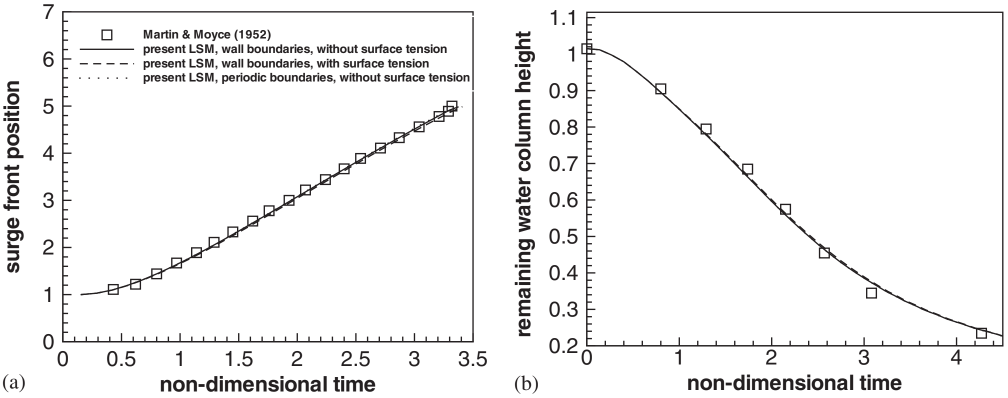

Figure 17. Three-dimensional broken dam: (a) surge front positions versus non-dimensional time; (b) remaining water column height h versus non-dimensional time.

As before, we first examine the spreading velocity and the falling rate of the water column. Figures 17(a) and 17(b) show that the two quantities are in good agreement with experimental data. Note that the surge front and the remaining water column height in these figures are measured in the centre plane in the spanwise direction. There is no obvious difference between the results obtained by different side boundary conditions. As in the 2D cases, surface tension does not have a considerable effect on the results.

Figure 18 shows snapshots of water surface position and the velocity fields in the centre plane in the spanwise direction at the selected times for the case with the periodic boundaries. The free surface shapes are essentially 2D. Figure 19 shows the results with the wall boundaries. The flow does not show any obvious three-dimensionality until at time $T = 8.0$ (Figure 19g). The entrapment of air by the water and formation of an elongated thin surge due to the splashing of the surge front as in the 2D cases are clearly observed in both Figures 18 and 19. Figure 20 shows close-up views of the surge fronts and rear views of the air bubbles observed in Figures 18g and 19g. The surge front in Figure 20(a) is basically 2D because of the periodic boundary conditions for the side walls, while it has a tongue-like shape in Figure 20(b) due to the effects of the wall boundary layers. The entrapped air bubble in Figure 20(c) for the periodic case shows a quasi-cylindrical shape. Interestingly, the entrapped air bubble

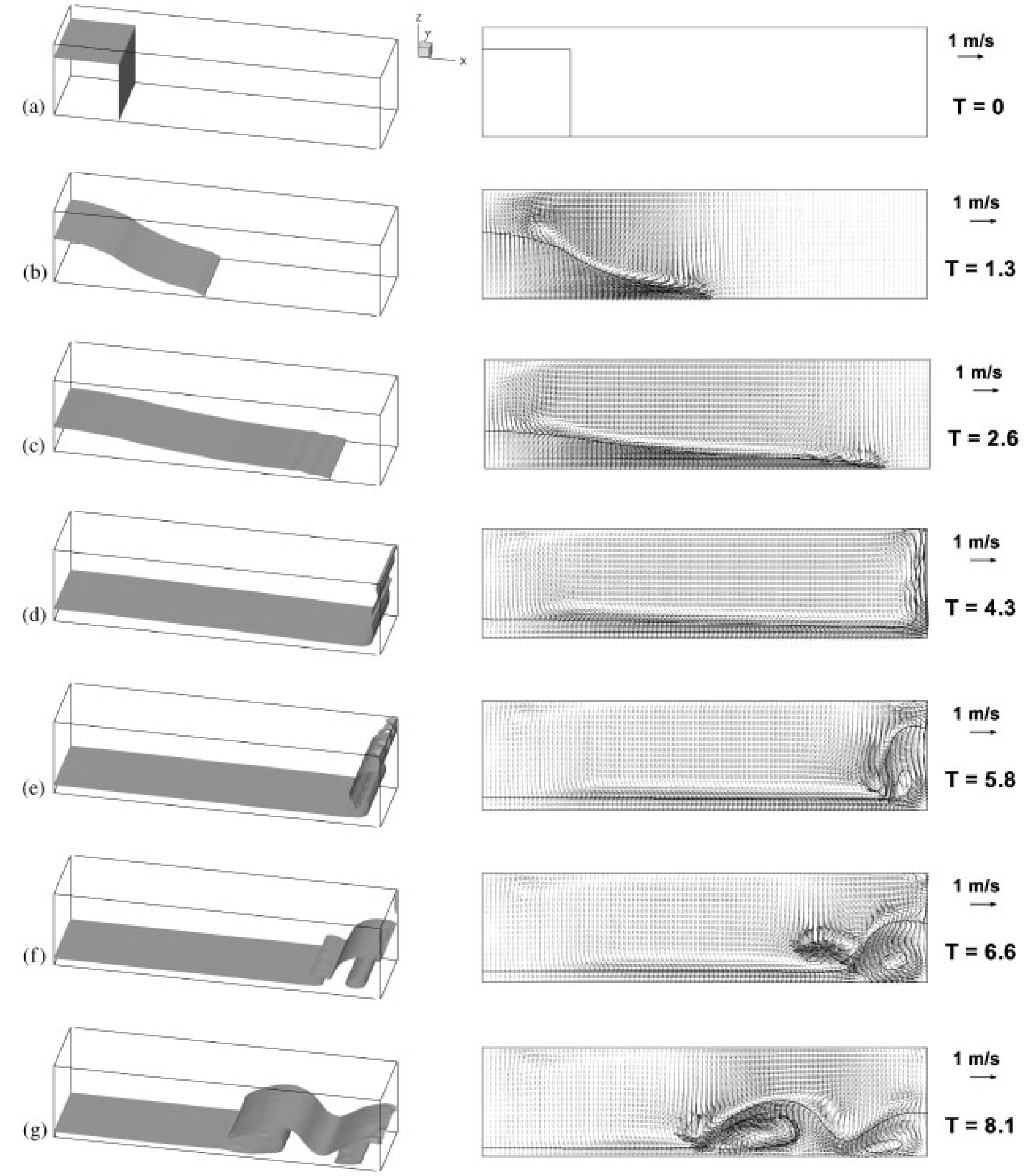

Figure 18. Free surface position (represented by the shallow areas in the left pictures) and velocity vectors in the centre plane of the container (right pictures, lines represent free surface) at selected times for the periodic boundary case without surface tension.

in the wall boundary case shows a symmetric horse-shoe shape in Figure 20(c), and much of the air is concentrated in the central part of the bubble. Despite the three-dimensionality of the free-surface in Figure 19(g), the sliced free surface shape in the centre plane in the spanwise direction (in the right vector picture) is very similar to that in Figure 18g. The mass

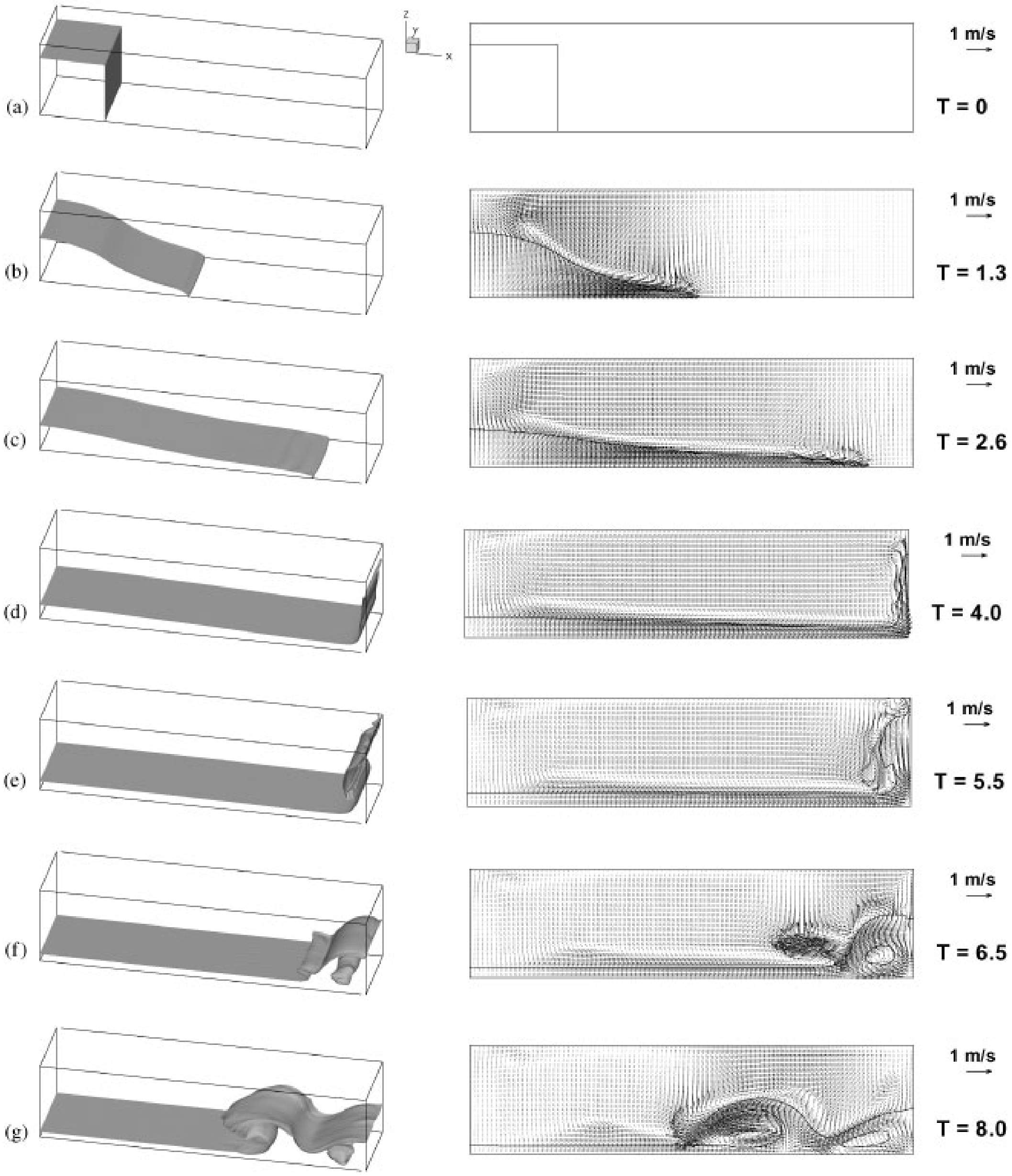

Figure 19. Free surface position (represented by the shallow areas in the left pictures) and velocity vectors in the centre plane of the container (right pictures, lines represent free surface) at selected times for wall boundary case without surface tension.

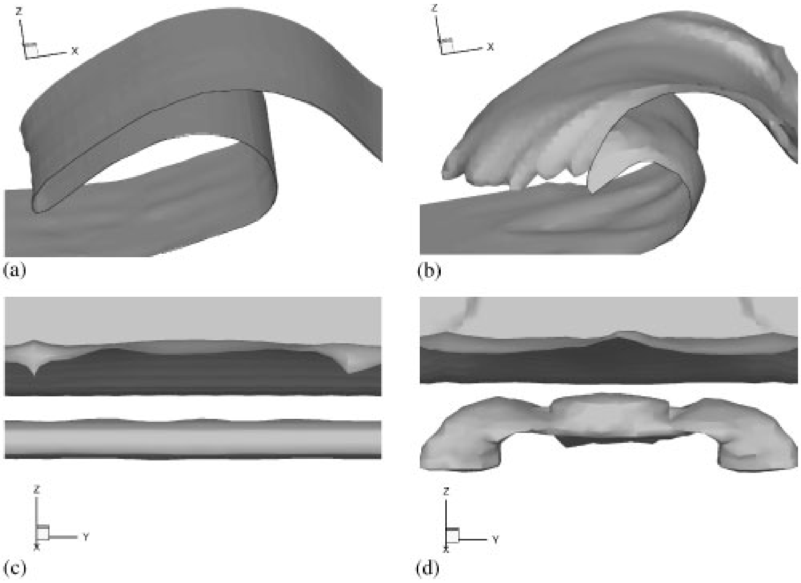

Figure 20. Close-up views of the surge fronts and rear views of entrained air bubbles in Figures 18g and 19g; (a) and (c) for periodic boundaries; (b) and (d) for wall boundaries.

errors for both the 3D cases are also shown in Figure 14. The mass preservation is somewhat better than in the 2D cases.

5. CONCLUSIONS

This paper has presented the coupling of 3D incompressible Navier–Stokes equations with the level set method in a curvilinear co-ordinate system. The equations are discretized by the finite volume method on a non-staggered grid with the four-step fractional step method. It is shown that the splitting error in the four-step fractional step method is second-order in time, the pressure-like variable is also second-order accurate in time to the real pressure, and no special treatment is required for the boundary conditions of the intermediate velocity to maintain consistent time accuracy in the numerical schemes. The four-step fractional step method was tested by calculation for the lid-driven cavity flow. The computational results show excellent agreement with well known benchmark tests. The level set evolution and reinitialization equations were solved with ENO schemes formulated in the curvilinear co-ordinates. They were validated against benchmark tests, namely, the reinitialization of a circle, Zalesak's problem and the stretching of a circular fluid element in a swirling shear flow.

The coupled system was applied to a 2D travelling solitary wave, and 2D and 3D broken dam problems. The computational results show excellent agreement with theoretic predictions and the experimental data. Two types of spanwise boundaries were examined in the 3D broken dam problem, namely the periodic boundaries and the wall boundaries. The former shows essentially 2D flow, while the three-dimensionality shows up in the latter case after some time. Quite complex free surface patterns are revealed by the present results, including the large vortices in the solitary wave case as well as the dam breaking cases, air entrapment in the water, air splashing of the water surge front in the 2D and 3D dam breaking cases. The mass conservation is preserved quite well by the present numerical schemes in all the cases.

ACKNOWLEDGEMENTS

C.L. Lin acknowledges the financial support by ONR N00014-01-1-0262. The authors thank Dr A. Iafrati, Dr M. Sussman and Dr R. Wilson for helpful discussions.

REFERENCES

1. Floryan JM, Rasmussen H. Numerical methods for viscous flows with moving boundaries. *Applied Mechanics Reviews* 1989; **42**:323–341.
2. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian-Eulerian computing method for all speeds. *Journal of Computational Physics* 1974; **14**:227–253.
3. Glimm J, Bryan OM, Menikoff R, Sharp DH. Front tracking applied to Rayleigh–Taylor instability. *SIAM Journal of Scientific and Statistical Computing* 1987; **7**:230–251.
4. Unverdi SO, Tryggvason G. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics* 1992; **100**:25–37.
5. Harlow FH, Welch JE. Numerical study of large-amplitude free surface motions. *Physics of Fluids* 1965; **8**:2182–2189.
6. Noh WF, Woodward P. SLIC (simple line interface calculation). In *Lecture notes in Physics*, Vol. 59. Springer-Verlag: New York, 1976: 330–340.
7. Hirt CW, Nichols BD. Volume of fluid (VOF) methods for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
8. Youngs DL. Time-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*, Morton KW, Baines MJ (eds). Academic: New York, 1982: 273–285.
9. Kelecy FJ, Pletcher RH. The development of a free surface capturing approach for multidimensional free surface flows in closed containers. *Journal of Computational Physics* 1997; **138**:939–980.
10. Jacqmin D. Calculation of two-phase Navier–Stokes flows using phase-field modelling. *Journal of Computational Physics* 1999; **155**:96–127.
11. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed:algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.
12. Osher S, Fedkiw RP. *Level Set Method and Dynamic Implicit Surfaces*. Springer: Berlin, 2002.
13. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.
14. Kang M, Fedkiw RP, Liu XD. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing* 2000; **15**:323–360.
15. Liu XD, Fedkiw RP, Kang M. A boundary condition capturing method for Poisson's equation on irregular domains. *Journal of Computational Physics* 2000; **160**:151–178.
16. Chang YC, Hou TY, Merriman B, Osher S. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics* 1996; **124**:449–464.
17. Foster N, Fedkiw R. Practical animation of liquids. *SIGGRAPH* 2001; 15–22.
18. Iafrati A, Mascio AD, Campana EF. A level set technique applied to unsteady free surface flows. *International Journal for Numerical Methods in Fluids* 2001; **35**:281–297.
19. Sussman M, Smereka P. Axisymmetric free boundary problems. *Journal of Fluid Mechanics* 1997; **341**:269–294.
20. Aslam TD, Bdzil JB, Stewart DS. Level set methods applied to modelling detonation shock dynamics. *Journal of Computational Physics* 1996; **126**:390–409.
21. Smiljanovski V, Moser V, Klein R. A capturing-tracking hybrid scheme for deflagration discontinuities. *Combustion Theory and Modelling* 1997; **1**:183–215.
22. Kim YT, Goldenfeld N, Dantzig J. Computation of dendritic microstructures using a level set method. *Physical Review E* 2000; **62**:2471–2474.
23. Smereka P. Spiral crystal growth. *Physica D* 2000; **138**:282–301.
24. Son G, Dhir VK. Numerical simulation of film boiling near critical pressures with a level set method. *Journal of Heat Transfer* 1998; **120**:183–192.
25. Sethian JA, Adalsteinsson D. An overview of level set methods for etching, deposition, and lithography development. *IEEE Transaction on Semiconductor Manufacturing* 1997; **10**:167–184.

26. Brackbill JU, Kothe DB, Zemach C. A continuum method for modelling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.
27. Zang Y, Street RL, Koseff JR. A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear co-ordinates. *Journal of Computational Physics* 1994; **114**:18–33.
28. Adalsteinsson D, Sethian JA. A fast level set method for propagating interfaces. *Journal of Computational Physics* 1995; **118**:269–277.
29. Sethian JA. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences* 1996; **93**:1591–1595.
30. Tsitsiklis JN. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control* 1995; **40**:1528–1538.
31. Peng D, Merriman B, Osher S, Zhao H, Kang M. A PDE-based fast local level set method. *Journal of Computational Physics* 1999; **155**:410–438.
32. Sussman M, Fatemi E, Smereka P, Osher S. An improved level set method for incompressible two-phase flows. *Computers & Fluids* 1998; **27**:663–680.
33. Perng CY, Street RL. A coupled multigrid-domain-splitting technique for simulating incompressible flows in geometrically complex domains. *International Journal for Numerical Methods in Fluids* 1991; **13**:269–286.
34. Choi H, Moin P. Effects of the computational time step on numerical solutions of turbulent flow. *Journal of Computational Physics* 1993; **113**:1–4.
35. Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics* 1985; **59**:308–323.
36. Perot JB. An analysis of the fractional step method. *Journal of Computational Physics* 1993; **108**:51–58.
37. Brown DL, Cortez R, Minion ML. Accurate projection methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 2001; **168**:464–499.
38. Crandall M, Lions P. Viscosity solutions of Hamilton–Jacobi equations. *Transactions of the American Mathematical Society* 1933; **277**:1–42.
39. Jiang G, Wu C. A high-order WENO finite difference scheme for the equations of ideal magnetohydrodynamics. *Journal of Computational Physics* 1999; **150**:561–594.
40. Harten A, Engquist B, Osher S, Chakravarthy S. Uniformly high-order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics* 1987; **71**:231–303.
41. Shu CW, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics* 1989; **83**:32–78.
42. Ghia U, Ghia KN, Shin CT. High-Re solution for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
43. Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics* 1979; **31**:335–362.
44. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
45. Rudman M. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
46. Leveque RJ. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis* 1996; **33**:627–665.
47. Enright D, Fedkiw RP, Ferziger JH, Mitchell I. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics* 2002; **183**:83–116.
48. Ramaswamy B. Numerical simulation of unsteady viscous free surface flow. *Journal of Computational Physics* 1994; **90**:396–430.
49. Thé JL, Raithby GD, Stubley GD. Surface-adaptive finite-volume method for solving free surface flows. *Numerical Heat Transfer*, *Part B* 1994; **26**:367–380.
50. Mei CC. *The Applied Dynamics of Ocean Surface Waves*. World Scientific: Singapore, 1989.
51. Chan RK, Street RL. A computer study of finite amplitude water waves. *Journal of Computational Physics* 1970; **6**:68–94.
52. Martin JC, Moyce WJ. An experimental study of the collapse of liquid columns on a rigid horizontal plate. *Philosophical Transactions of the Royal Society of London Series A* 1952; **244**:312–324.