INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **42**:187–210 (DOI: 10.1002/fld.499)

An efficient method for capturing free boundaries in multi-fluid simulations

F. Xiao[1,*,†] and A. Ikebata[2]

[1] *Department of Energy Sciences, Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama, 226-8502, Japan*
[2] *Production Technology Laboratory, TOTO Ltd., 1-1 Nakashima 2-chome, Kokurakita-ku, Kitakyushu-city, 802-8601, Japan*

SUMMARY

An easy-to-use front capturing method is devised by directly solving the transport equation for a volume of fluid (VOF) function. The key to this method is a semi-Lagrangian conservative scheme, namely CIP_CSL3, recently proposed by the author. In the CIP_CSL3 scheme, the first-order derivative of the interpolation polynomial at each cell centre is used to control the shape of the reconstructed profile. We show in the present paper that the first-order derivative, which plays a crucial role in reconstructing the interpolation profile, can also be used to eliminate numerical diffusion. The resulting algorithm can be directly used to compute the VOF-like function and retain the compact thickness of the moving interface in multi-fluid simulations. No surface reconstruction based on the value of VOF function is required in the method, which makes it quite economical and easy to use. The presented method has been tested with various interfacial flows including pure rotation, vortex shearing, multi-vortex deformation and the moving boundaries in real fluid as well. The method gives promising results to all computed problems. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: Multi-fluid flow; numerical method; transport scheme; moving boundary; interface tracking

1. INTRODUCTION

Numerical simulations for multi-fluid flows often require explicit computations for the free interfaces which separate different fluid components. The earliest work in describing and tracking a free surface can be traced back to the MAC (Marker and cell) method [1]. In the MAC method, Lagrangian particles were tracked and predicted according to the fluid velocity

[*] Correspondence to: F. Xiao, Department of Energy Sciences, Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama, 226-8502 Japan.
[†] E-mail: xiao@es.titech.ac.jp

Contract/grant sponsor: Scientific Research of Japan Society for the Promotion of Science; contract/grant number: 13750137.

Received 20 March 2002
Revised 9 January 2003

Copyright © 2003 John Wiley & Sons, Ltd.

field. The position of the interface is located in the cells that contain particles of both fluids. Without re-positioning the particles, the spatial distribution of the particles tends to become highly irregular with some cells over-filled while some others are under-filled in regions where the spatial domain is heavily deformed. Adding extra particles or re-distributing the existing particles might improve the numerical solutions, but a large increase in computational cost prevents the method from being practical, particularly in simulations involving large distorted configurations.

Successive studies for the numerical representation of a moving interface led to another volume tracking concept for incompressible fluid flows, the volume of fluid (VOF) [2, 3]. A VOF method records and predicts the fraction of a 'fluid', which is an indicator of the volume ratio for a specified fluid component and is sometimes called the VOF function. The interface is reconstructed by using line segments (2D) or polygons (3D) which cut through the cells that have a fractional value of the VOF function. There is freedom in choosing the reconstruction technique for the interface, for example, the SLIC (using line segments aligned to grid lines) or the PLIC (using linear approximations [4–8]).

Using a VOF function to identify different fluids appears to be a natural representation of an interface on a fixed grid. Another attractive property of VOF is that it is quite easy to conserve the 'mass' (or equivalently the volume in incompressible fluid) of the fraction function. A conventional VOF method requires an interface reconstruction which is based only on the fraction of fluid in each cell. The reconstruction procedure involves extra arithmetic operations and 'if' logics, and consequently is more computationally intricate, especially in 3D applications.

Another method that shares great popularity for computing a moving free boundary is the level set scheme [9, 10]. A level set function, which is actually defined as the signed distance for any point according to how far away it is located in respect to the interface, gives a well-regulated field for computing the geometric quantities such as the normal vector and the curvature of an interface. However, because there is no rigorous restriction for mass conservation during the process for re-initializing the distance function, the conservation of mass is not guaranteed. So, in many applications where conserving the mass of a fluid appears to be essential, the VOF method is still preferred.

In fact, none of the existing schemes are perfect in all the numerical aspects, like geometric faithfulness and mass conservation. Moreover, both the VOF and the level set require some extra steps to rearrange the field variable after the advection computation. An interface reconstruction is conducted in a VOF method which is based on the SLIC or PLIC approach. Many numerical simulations show that a SLIC reconstruction is usually not adequate in capturing complex interfaces in real applications. A PLIC reconstruction produces a much better numerical result than the SLIC reconstructions, but needs more complex computational efforts, especially in three dimensions. Under the level set method, on the other hand, the re-initialization procedure is required to make the distance function well defined. Thus, there might be a desire for interface capturing schemes which work without the extra treatments besides advection. The aim of this paper is to devise a way which does not require the reconstruction procedure in a VOF-type methodology.

Recalling that for a given velocity field $\mathbf{U}$ any VOF method is designed to solve

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{U} f) - f \nabla \cdot \mathbf{U} = 0 \tag{1}$$

Copyright © 2003 John Wiley & Sons, Ltd.

for a function f valued between 0 and 1. One can be motivated to directly solve the advection equation (1) for a VOF function, which usually shows up as a density function or a colour function in applications, by using a sophisticated advection scheme. If the advection scheme is well-designed, we can expect that the computed solution can be used as the VOF function and needs no interface reconstruction. However, as commented by Rider and Kothe [12], any advanced Eulerian advection scheme has intrinsic numerical diffusion and tends to smear the free interface. Therefore, it is difficult to retain the compact thickness and the geometric faithfulness of the interface by only using an advanced advection equation. Remedies to the numerical diffusions of advection schemes have been devised [13, 14] and applied to multi-fluid simulations [15]. Nevertheless, direct computation of the VOF function with only an advection scheme does not seem to have been widely accepted yet as a practical tracking method for moving boundaries.

This work presents an economic numeric method to compute moving boundaries. The scheme is based on our newly developed transport method namely CIP-CSL3 [16]. The scheme is exactly mass-conservative no matter how the slope limiter is determined. We will show that with an easy anti-diffusion modification in the original scheme, the thickness of the transition layer can be kept compact. Section 2 briefly describes the advection scheme and its multi-dimensional implementation. The anti-diffusion modification that is effective in avoiding the smearing of the VOF function is discussed in Section 3. Various numerical tests which show the effectiveness of the resulting scheme are reported in Section 4, and the paper ends with a short summary in Section 5.

2. THE ADVECTION SCHEME

Yabe *et al.* [17] and Xiao and Yabe [16] have recently suggested a new class of schemes, so called Constrained Interpolation Profile_Conservative Semi-Lagrangian (CIP_CSL) schemes, by using a semi-Lagrangian updating to compute the cell-interface values. Unlike other conventional schemes such as MUSCL [18, 19] and PPM [20], the CIP_CSL schemes are constructed on a profile with a global smoothness of at least C^0 and cause much less dispersion error. In Reference [16], the numerical oscillations were eliminated by introducing a slope limiter at the centre of each mesh cell. The slope, as a parameter in reconstructing the interpolation profile, in fact provides us with large freedom to control the interpolation profile and thus modulate the numerical solutions. Next, we describe the CIP-CSL3 scheme.

The CIP-CSL3 method is originally designed to solve the transport equation

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}(uf) = 0 \tag{2}$$

where t refers to the time, x the spatial co-ordinate, u the characteristic speed and f the transported quantity. It can be directly applied to the advection equation of the VOF function

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}(uf) - f\frac{\partial u}{\partial x} = 0 \tag{3}$$

by simply adding a divergence term.

From the given data $f(x_1), f(x_2), \ldots, f(x_i), \ldots, f(x_N)$ with $x_1 < x_2 < \cdots < x_i < \cdots < x_N$, which denote the numerical solution of Equation (3) $\{f_i^n\}$ over the computational domain

Copyright © 2003 John Wiley & Sons, Ltd.

at the nth time step ($t=t^n$), we make use of a piece-wise cubic polynomial function, like that in the original CIP method [21].

The ith piece of the interpolation function is constructed over upwind stencils. Taking the case of $u \geqslant 0$, a left-bias interpolation can be written as

$$\begin{aligned} F_i^{\mathrm{L}}(x) = f(x_i) + c_{1i}^{\mathrm{L}}(x-x_i) + c_{2i}^{\mathrm{L}}(x-x_i)^2 \\ + c_{3i}^{\mathrm{L}}(x-x_i)^3, \quad \text{for } x \in [x_{i-1}, x_i] \end{aligned} \tag{4}$$

From the continuity conditions of $F_i^{\mathrm{L}}(x)$ at the two ends of the cell, we have

$$F_i^{\mathrm{L}}(x_i) = f^n(x_i) \tag{5}$$

and

$$F_i^{\mathrm{L}}(x_{i-1}) = f^n(x_{i-1}) \tag{6}$$

A constraint for the conservation of cell-integrated average is imposed as

$$\frac{1}{\Delta x_{i-1/2}} \int_{x_{i-1}}^{x_i} F_i^{\mathrm{L}}(x)\,\mathrm{d}x = \bar{f}_{i-1/2}^{n} \tag{7}$$

where $\Delta x_{i-1/2} = x_i - x_{i-1}$.

Another constrained condition for the interpolation construction is imposed on the first-order derivative of $F_i^{\mathrm{L}}(x)$, at the middle point of the cell

$$\frac{\mathrm{d}F_i^{\mathrm{L}}(x)}{\mathrm{d}x} = d_{i-1/2}^n \tag{8}$$

The slope of the interpolation function at the cell centre $d_{i-1/2}^n$ remains as a free parameter to be determined. It is this parameter that provides us a way to modify the interpolation function for reducing numerical diffusion and suppressing numerical oscillation. Some practical candidates for computing $d_{i-1/2}^n$ were given in Reference [16]. The modifications to $d_{i-1/2}^n$ which effectively diminish the numerical diffusions will be discussed in the following section.

In terms of $f_i^n, f_{i-1}^n, d_{i-1/2}^n$ and $\bar{f}_{i-1/2}^n$, the polynomial (4) can be completely determined from (5–8), and the coefficients read

$$\begin{aligned} c_{1i}^{\mathrm{L}} &= -\frac{6}{\Delta x_{i-1/2}}\bar{f}_{i-1/2}^n + \frac{6}{\Delta x_{i-1/2}} f_i^n - 2d_{i-1/2}^n \\ c_{2i}^{\mathrm{L}} &= -\frac{6}{\Delta x_{i-1/2}^2}\bar{f}_{i-1/2}^n + \frac{3}{\Delta x_{i-1/2}^2}(3f_i^n - f_{i-1}^n) - \frac{6}{\Delta x_{i-1/2}} d_{i-1/2}^n \\ c_{3i}^{\mathrm{L}} &= \frac{4}{\Delta x_{i-1/2}^3}(f_i^n - f_{i-1}^n) - \frac{4}{\Delta x_{i-1/2}^2} d_{i-1/2}^n \end{aligned} \tag{9}$$

Analogously, the right-bias F_i^{R} interpolation function can also be derived. Once the interpolation function is determined, the numerical solution of f at time step $n+1$ is updated by a semi-Lagrangian solution as

$$f_i^{n+1} = \begin{cases} F_i^{\mathrm{L}}(x_i - u\Delta t), & \text{if } u > 0 \\ F_i^{\mathrm{R}}(x_i - u\Delta t), & \text{if } u < 0 \end{cases} \tag{10}$$

The cell-integrated average $\bar{f}$ is advanced by a flux form from the conservative relation

$$\bar{f}_{i-1/2}^{n+1} = \bar{f}_{i-1/2}^{n} - (g_i - g_{i-1})/\Delta x_{i-1/2} + \bar{f}_{i-1/2}^{n}(u_{i+1/2} - u_{i-1/2})/\Delta x_i \tag{11}$$

where g_i represents the flux across boundary $x = x_i$ during $t^{n+1} - t^n$ and is computed as

$$g_i = \int_{t^n}^{t^{n+1}} [\min(0,u) F_i^{\mathrm{R}}(x_i - u(t - t^n)) + \max(0,u) F_i^{\mathrm{L}}(x_i - u(t - t^n))]\,\mathrm{d}t \tag{12}$$

The CIP-CSL3 scheme is substantially different from the PPM scheme at the following aspects: (1) the cell-interface value in the CIP-CSL3 scheme is treated as a dependent variable and predicted via a semi-Lagrangian step, whereas the cell-interface value in the PPM is determined by an interpolation based on the cell-integrated average value; (2) to eliminate the numerical oscillations, both the slope limiter and the modification to the cell-interface value are required in PPM. Thus, the cell-interface values are not necessarily continuous at the cell boundaries. In CIP-CSL3, however, an oscillation-less solution can be easily obtained by just modifying the slope at the cell center. The cell-interface values are always continuous. (3) A cubic polynomial is used in the CIP-CSL3 method, whereas the PPM uses a parabolic function. (4) Our numerical experiments show that a CIP-CSL type scheme gives better results in terms of numerical dispersion [11].

The arithmetic operation count of CIP-CSL3 is almost the same as that of PPM. To evaluate the computational efficiency, we have timed the elapsed CPU seconds of CIP-CSL3 and PPM with a 1D advection of a rectangular wave on a 400-cell mesh for 10 000 steps of time integration on a Windows PC with a PentiumIII/850 MHz. The FORTRAN compiler is Compaq/DIGITAL Visual FORTRAN version 6.1. The CPU elapsed time of PPM is 2.95 s and that of the CIP-CSL3 is 3.10 s.

In multi-dimensional implementation, a splitting that is based on the 1D solver is used. The staggered cell edge values that are not aligned along the splitting direction need to be updated according to the provisional time variation of the cell-integrated averages $\bar{f}$. We update the interface values by using a cubic spline function.

Taking the time variation of the cubic spline in terms of f and $\bar{f}$, for example in the x direction, we arrive at

$$\begin{aligned} &\frac{1}{\Delta x_{i-1/2}}\,\delta_t f_{i-1} + 2\left(\frac{1}{\Delta x_{i-1/2}} + \frac{1}{\Delta x_{i+1/2}}\right)\delta_t f_i + \frac{1}{\Delta x_{i+1/2}}\,\delta_t f_{i+1} \\ &\quad = 3\left(\frac{1}{\Delta x_{i-1/2}}\,\delta_t \bar{f}_{i-1/2} + \frac{1}{\Delta x_{i+1/2}}\,\delta_t \bar{f}_{i+1/2}\right) \end{aligned} \tag{13}$$

Equation (13) can be easily solved by a direct elimination method. Another completely explicit formulation can be obtained by using a simple averaging to the values at the two

Copyright © 2003 John Wiley & Sons, Ltd.

ends of the tri-diagonal band in (13) as

$$\delta_t f_{i-1} = \left(\frac{1}{\Delta x_{i-3/2}} + \frac{1}{\Delta x_{i-1/2}}\right)^{-1} \left(\frac{1}{\Delta x_{i-3/2}}\,\delta_t \bar{f}_{i-3/2} + \frac{1}{\Delta x_{i-1/2}}\,\delta_t \bar{f}_{i-1/2}\right) \quad \text{and} \tag{14}$$

$$\delta_t f_{i+1} = \left(\frac{1}{\Delta x_{i+1/2}} + \frac{1}{\Delta x_{i+3/2}}\right)^{-1} \left(\frac{1}{\Delta x_{i+1/2}}\,\delta_t \bar{f}_{i+1/2} + \frac{1}{\Delta x_{i+3/2}}\,\delta_t \bar{f}_{i+3/2}\right) \tag{15}$$

The interface value f_i can be updated immediately after the time variation tendency $\delta_t f_i$ is computed.

3. INTERFACE CAPTURING: ANTI-DIFFUSION MODIFICATION

Tracking or capturing free boundaries plays a key role in simulating multi-fluid flows, and thus has been an active research field for more than a few decades. As mentioned in the introduction, the interface reconstruction of the VOF in three dimensions is not an easy job, while the re-initialization in the level set appears computational expensive. An alternative way that does not involve a reconstruction procedure is desirable.

A noteworthy work on direct computation of VOF function is due to Rudman [22], where the FCT (flux-corrected transport) concept is used to eliminate numerical diffusion and the initial sharpness of the VOF function can be enforced.

Since CSL3 provides a formulation that allows modifying the interpolation profile through the slope $d^n_{i-1/2}$ but without violation of mass conservation, we can easily design an algorithm that maintains sharp transition jumps.

There are many ways to determine the first-order derivative at the cell center $d^n_{i-1/2}$ in CSL3 method. For capturing a VOF function (or a Heaviside function), the desired numerical solution should have at least the following two features: (1) small numerical oscillation; and (2) small numerical smearing. A trade-off between the oscillation reduction and the sharpness enforcement can be easily achieved in the CIP-CSL3 scheme by modifying the slope $d^n_{i-1/2}$.

The numerical oscillation can be effectively suppressed if we approximate $d^n_{i-1/2}$ by using the following formulation [19]

$$\hat{d}_{i-1/2} = \text{minmod}(\tilde{d}_{i-1/2}, 2S_i, 2S_{i-1}) \tag{16}$$

with $S_i = (f^n_{i+1/2} - f^n_{i-1/2})/\Delta x_i$. We simply use the central differencing to approximate $\tilde{d}_{i-1/2}$ in the present paper.

As given in Reference [16], the 'point-value' of the interpolation function at cell centre can be computed as

$$f^n_{i-1/2} = \frac{3}{2}\,\bar{f}^n_{i-1/2} - \frac{1}{4}(f^n_i + f^n_{i-1}) \tag{17}$$

As long as $\hat{d}_{i-1/2}$ is computed, a modification can be made by finally determining $d^n_{i-1/2}$ with a function in form as $\Phi(\beta, \hat{d}_{i-1/2})$. One of the simplest forms is

$$d^n_{i-1/2} = \beta \hat{d}_{i-1/2} \tag{18}$$

Copyright © 2003 John Wiley & Sons, Ltd.

The parameter β functions as a modification to the slope. It is obvious that any β larger than one will enforce the local sharpness of each piece-wise constructed profile. When applied to a VOF function valued between 0 and 1, the following are some candidates for computing β,

$$\text{(i) } \beta = (1 + \alpha |f_i^n - f_{i-1}^n| / \Delta_{i-1/2}) \tag{19}$$

$$\text{(ii) } \beta = \alpha(1 - |\bar{f}_{i-1/2}^n - 0.5|) \tag{20}$$

$$\text{(iii) } \beta = \alpha_1 \mathrm{e}^{-\alpha_2 |\bar{f}_{i-1/2}^n - 0.5|} \tag{21}$$

Our numerical tests show that both (20) and (21) effectively enforce the sharpness of the jump in any Heaviside function in a highly deforming velocity field and are able to build a narrow, sharp transition zone around the surface that is identified with the 0.5 value of the VOF function. The numerical experiments suggested that a value of α between 2.0 and 2.5 in (20), and the values of 2.0 and 1.0 for α_1 and α_2 in (21) give a proper trade-off between reducing numerical diffusion and avoiding numerical oscillation. Expression (20) appears more computationally efficient in real applications. We hereafter refer to the CSL3 scheme with these anti-diffusion modifications as C3_VOF, a shortening of *Constrained interpolated profile_Conservative semi-Lagrangian scheme with Cubic polynomial for VOF*.

Since no surface construction is required in C3_VOF method, it can be more easily implemented in multi-dimensional computation. As discussed in the last section, the C3_VOF with slope modification (20) requires almost the same computational efforts as that of PPM. Comparisons with other popular interface capturing schemes can be made by checking [12] where a good reference of different schemes is given with respect to both the solution quality and computational expense.

The multi-dimensional version of the C3_VOF scheme is constructed through a splitting of the 1D scheme. The multi-dimensional implementation discussed in the last section can be used. The advection equation of the VOF function (1) and (3) includes a divergence-correction to the dimensional splitting based on the 1D scheme, which proves to be necessary in case of deformational velocity field for multi-dimensional applications.

4. NUMERICAL TESTS

We first tested the 1D C3_VOF scheme. A 1D Heaviside function is computed with a β defined by (19). Shown in Figure 1, the thickness of the transition region was kept constant even up to 10 000 steps. With a larger α value, the initial sharpness is better preserved. Furthermore, by a properly chosen α, a transition zone with a desired width, not just a jump of one mesh size, can be constantly reproduced. As a measure of numerical diffusion, we plotted the L_2 errors for different values of α in Figure 2. Compared to the calculation without anti-diffusion modification ($\alpha = 0$), the numerical diffusion has been effectively suppressed in the cases of $\alpha = 0.2$ and $\alpha = 1$. The slopes of the transition jumps remain constant even after a long term computation. Compared to other anti-diffusion or artificial compression techniques, C3_VOF provides a more flexible formulation to control the thickness of the transition layer across a moving interface.

Copyright © 2003 John Wiley & Sons, Ltd.

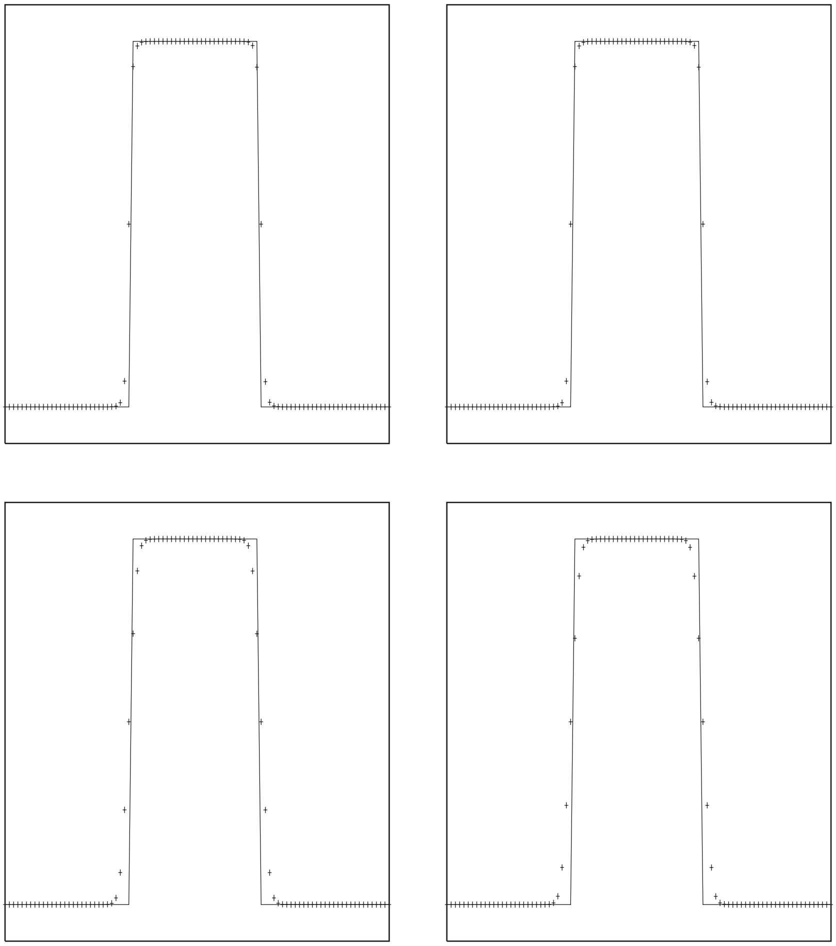

Figure 1. Computed results after 1000 steps (left) and 10 000 steps (right) with $\alpha = 1$ (top) and $\alpha = 0.2$ (bottom).

As a two-dimensional example, the Zalesak's cut-cylinder problem was computed. Figure 3 shows the results after one revolution. For comparison, we include the numerical result of PPM as well. It is clear that the slope modification in C3_VOF works effectively in reducing numerical diffusion and does not cause remarkable geometric distortion. The thickness of the transition layer being less than 4 grid cells is retained. The PPM method, however, is not able to get rid of the numerical diffusion across the interface jump, thus producing a more smeared and wider transition layer. Figure 4 displays the numerical interface computed by C3_VOF, which is identified as the 0.5 contour of the numerical solution, against the exact solution. The numerical solution appears to be accurate and geometrically faithful. We also repeated the rotating solid problem used by Rudman [22]. Shown in Figure 5, C3_VOF gives competitive results to those tested in Rudman [22]. The L_1 error (defined by (9) in Reference [22]) of C3_VOF for this test is 2.57×10^{-2}, which manifests an accuracy higher than the FCT-VOF method proposed in Reference [22] but lower than the PLIC.

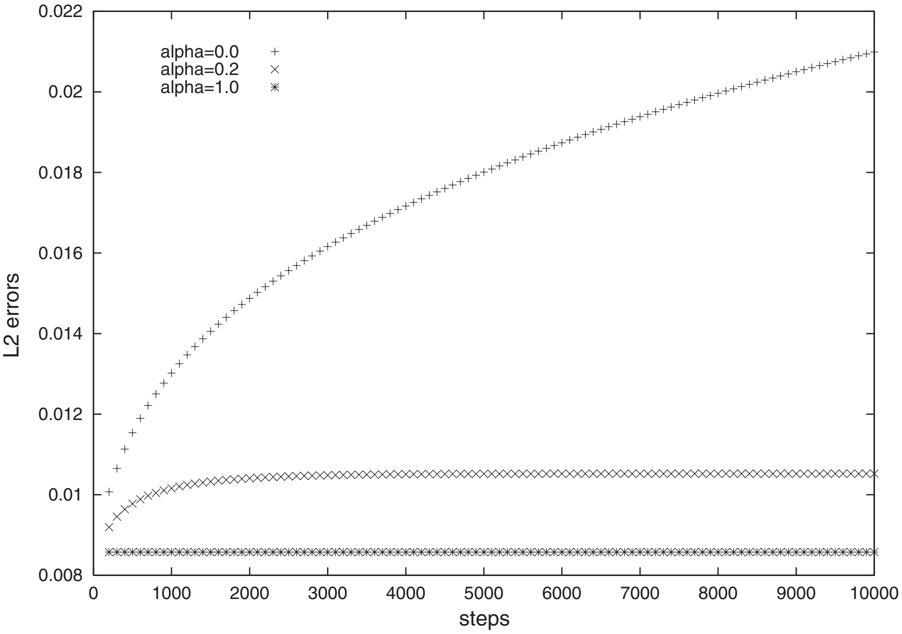

Figure 2. L_2 errors with different anti-diffusion parameter α. The calculation steps are up to 10 000.

We evaluated the convergence rate of C3_VOF by computing a rotating solid cylinder [12] on three gradually refined meshes which divide a unity square domain into 32×32, 64×64 and 128×128 grid points, respectively. The computed interfaces are plotted in Figure 6 against the exact solution. It is observed that the numerical interface has been adequately reproduced even on the 32×32 mesh. The L_1 errors and the mesh-refinement convergence rates are given in Table I. An overall first order convergence rate is obtained. Compared with the data given in Table II of [12], C3_VOF has an accuracy among the practical methods for interface tracking.

As suggested by Rider and Kothe [12, 7], a practical interface capturing method should be tested and verified with not only the translational and rotational velocity field but also with the highly deforming fluid flow. We tested C3_VOF using the velocity fields designed in References [12] and [7].

A complex flow field which leads to stretching and spiraling of the initial shape as used by Rider and Kothe [12, 7] and many others as defined with a stream function

$$\Psi = -\frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y)\cos(\pi t/T) \tag{22}$$

We conducted the calculations with β defined by (20). The computational domain is a unit square, and meshes of different resolutions as 32×32, 64×64 and 128×128 are used. As expected from (22), a reverse velocity field will be generated as $t > T/2$, and the ideal solution at $t = T$ will be identical to the initial configuration. Figures 7–9 show the results at different instants and the comparisons with the exact solution on various grid resolutions. The

Copyright © 2003 John Wiley & Sons, Ltd.

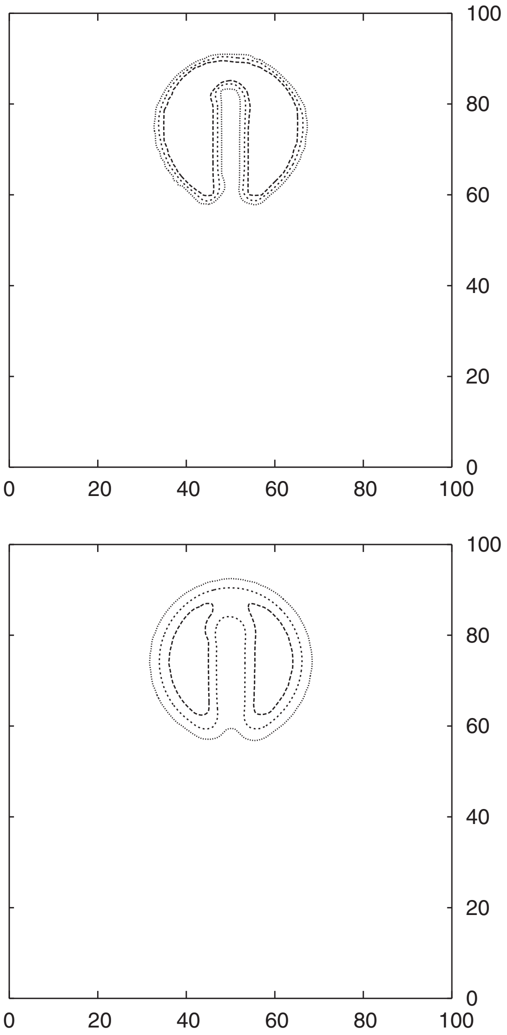

Figure 3. Zalesak's test after one revolution of rotation computed by C3_VOF (top) and PPM (bottom) respectively. Plotted are the contours of 0.05, 0.5 and 0.95.

initial circle has been distorted into a spiral with a stretched thin tail. The numerical solution on the low spatial resolution (on 32×32 mesh) failed in reproducing the detail structures near the stretched tail and the interface suffered numerical smearing, but the overall interface has been reasonably captured when the profile reversed to its initial position. Compared with the schemes proposed by Rider and Kothe [7] and Harvie and Fletcher [8], the present method reproduced the head of the spiral as well, but did not cause the discrete 'globs' (see the result at $t = 3$ in Figures 7 and 14(e) in Reference [7] or Figure 12(c) in Reference [8]) which appear to be commonly associated with the surface construction procedure

Copyright © 2003 John Wiley & Sons, Ltd.

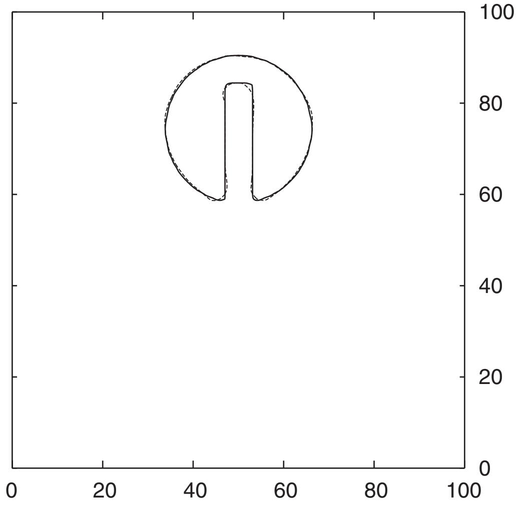

Figure 4. Zalesak's test after one circle of rotation. Displayed are the exact solution (solid line) and the numerical solution computed by C3_VOF (dashed line).

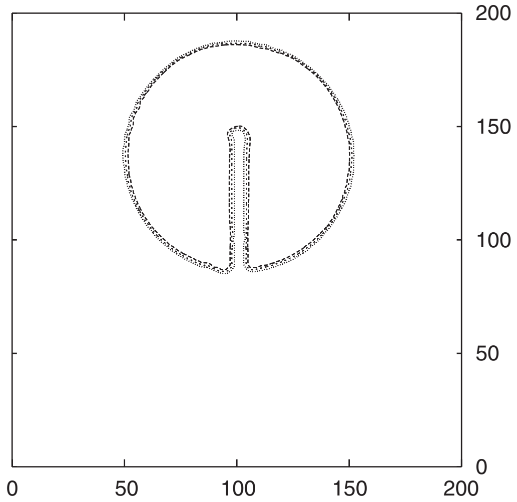

Figure 5. Another Zalesak's test used in Rudman [22]. Plotted are the contours of 0.05, 0.5 and 0.95.

as used in References [7] and [8]. Similar to those in References [7] and [8], the final solution on a 32×32 mesh (Figure 7 with $t = 6$), was not able to accurately simulate the exact solution, nevertheless, the numerical results of the present scheme still look visually acceptable. As shown in Figures 8 and 9, heightening the spatial resolution significantly improved the solution. The simulation of 128×128 computation gives satisfactory results for both the

Copyright © 2003 John Wiley & Sons, Ltd.

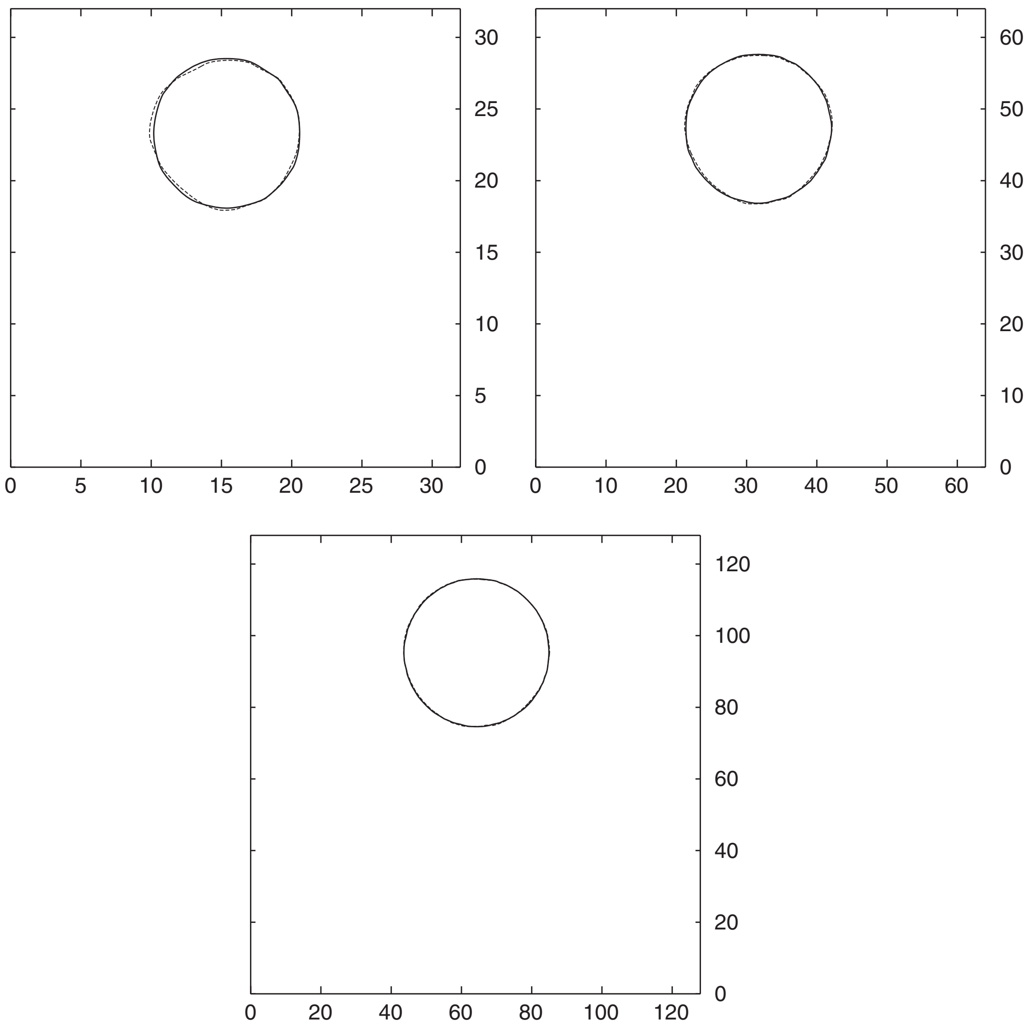

Figure 6. Solid body rotation of a circle on a 32×32 mesh (top), a 64×64 mesh (middle) and a 128×128 mesh (bottom). Displayed are the computed interfaces of the transport bodies (dashed line) and the exact solution (solid-line).

Table I. L_1 errors and convergence rates of C3_VOF on the solid rotation problem.

32×32	Order	64×64	Order	128×128
7.42×10^{-3}	0.924	3.91×10^{-3}	1.07	1.86×10^{-3}

largely distorted configuration ($t = 3$) and the final solution ($t = 6$). The grid refinement convergence rate for this test problem is given in Table II. The convergence rates are higher than first-order.

Copyright © 2003 John Wiley & Sons, Ltd.

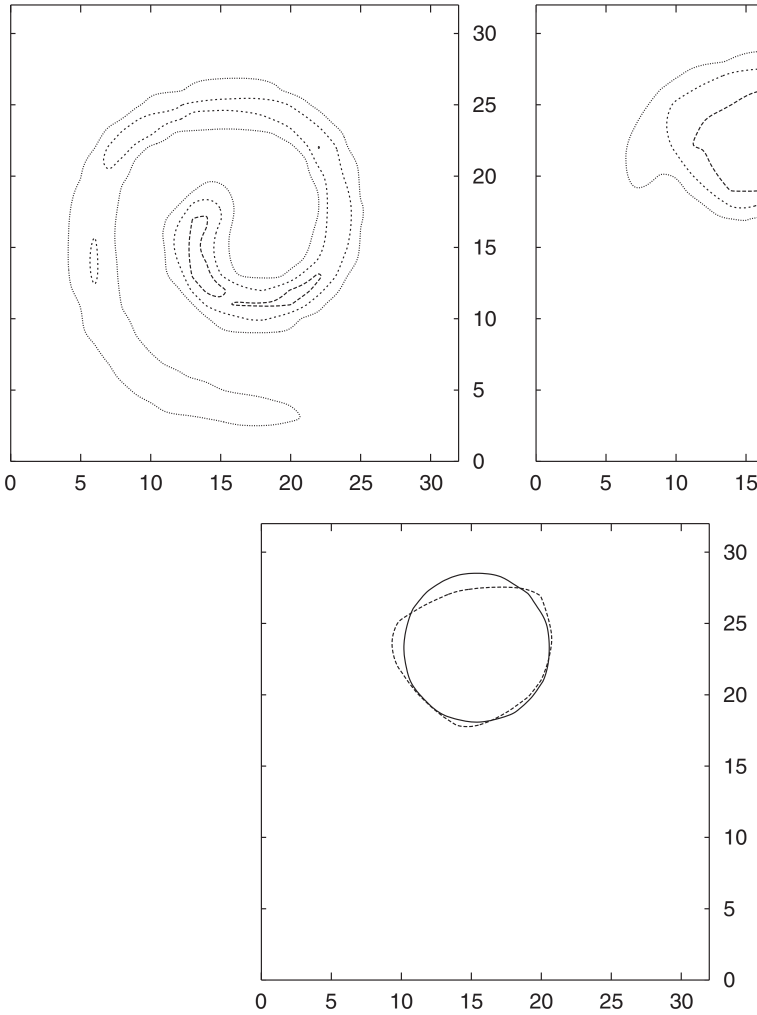

Figure 7. Single-vortex shearing flow test on a 32×32 mesh with $T = 6$. The velocity field reverses at $t = T/2$, and restores the configuration back to its initial state at $t = T$. Displayed are the contours of 0.05, 0.5 and 0.95 of the numerical results at $t = 3$ (top) and at $t = 6$ (middle). The bottom panel shows the difference between the numerical result (dashed line) and the exact solution (solid-line circle).

Another more stringent test is to capture an interface transported by a velocity field defined by

$$\Psi = \frac{1}{4\pi} \sin\left(4\pi\left(x + \frac{1}{2}\right)\right) \cos\left(4\pi\left(x + \frac{1}{2}\right)\right) \cos(\pi t/T) \tag{23}$$

This problem was initially used by Smolarkiewicz [23] to test general advection schemes and suggested by Rider and Kothe [12] for interface tracking methods. As can be seen in

Copyright © 2003 John Wiley & Sons, Ltd.

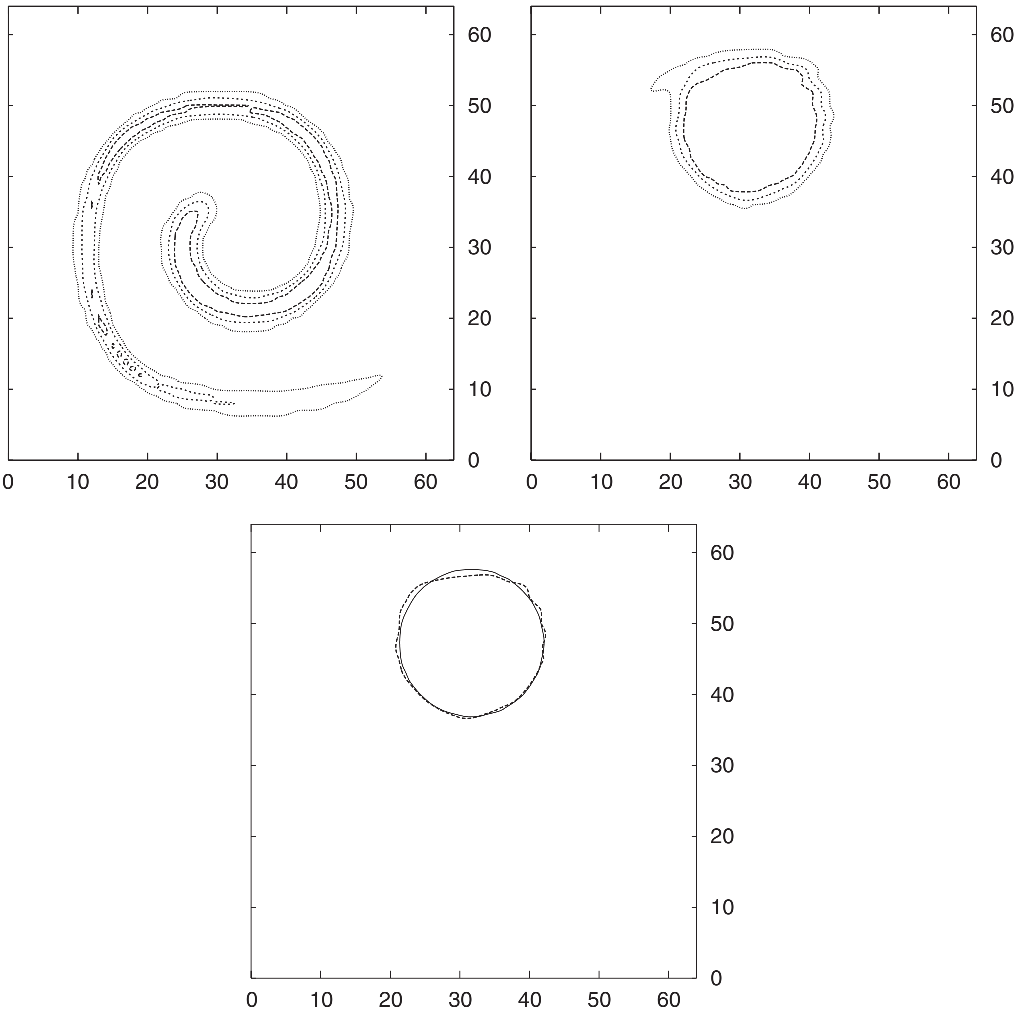

Figure 8. Same as Figure 7, but on a 64 × 64 mesh.

Table II. L_1 errors and convergence rates of C3_VOF on the single vortex problem.

32×32	Order	64×64	Order	128×128
3.22×10^{-2}	1.35	1.26×10^{-2}	1.21	5.42×10^{-3}

Figure 3 of [12], the initial sphere was largely deformed into several segments which appear topologically diverse, such as spiral, thin film and thin bridge. The thin segments usually appear not to be resolvable for most interface tracking methods.

A 2D cylinder as defined in the above example was transported with the velocity field (23). The time required to move the distribution back to its initial state is $T = 2$. A periodic

Copyright © 2003 John Wiley & Sons, Ltd.

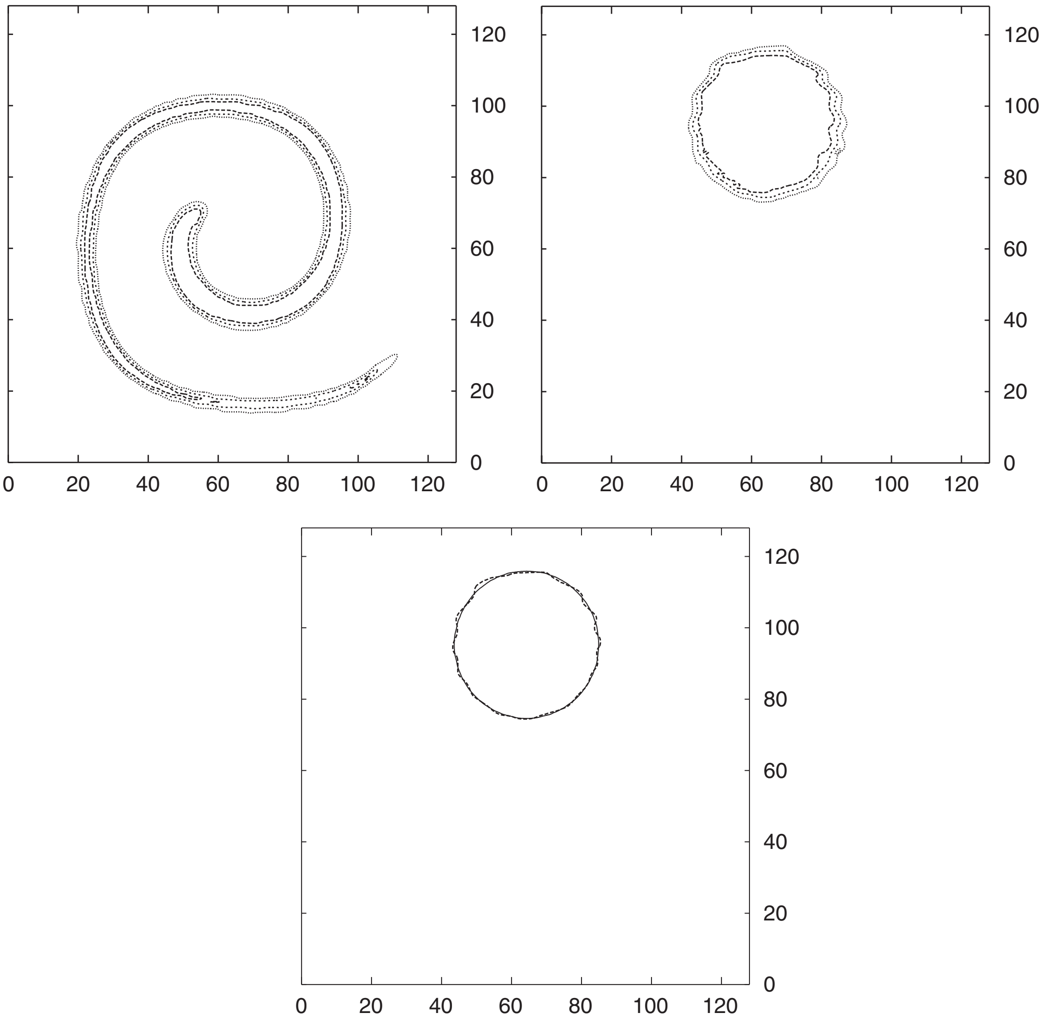

Figure 9. Same as Figure 7, but on a 128 × 128 mesh.

lateral boundary condition is imposed in this computation. Figures 10–12 shows the results at different instants of C3_VOF on 32×32, 64×64 and 128×128 meshes, respectively. The farthest filament that is entrapped by the two topmost vortexes could not be well resolved on 32×32 and 64×64 resolutions, while the computation on 128×128 mesh produced a competitive solution with respect to the results of other interface schemes given in Figure 5 of [12]. The bottom panel in Figure 12 plots the final distribution against the exact solution. The main part of the body has been satisfactorily restored, the entrapped thin filament has been transported back to around its original location and merged with the main part. On the coarse computational grid, it seems to be difficult to restore the highly stretched segment that is too thin to be resolved by the finite spatial resolution. When un-resolvable thin structures develop, we will lose the information regarding the initial structure. Therefore, we cannot

Copyright © 2003 John Wiley & Sons, Ltd.

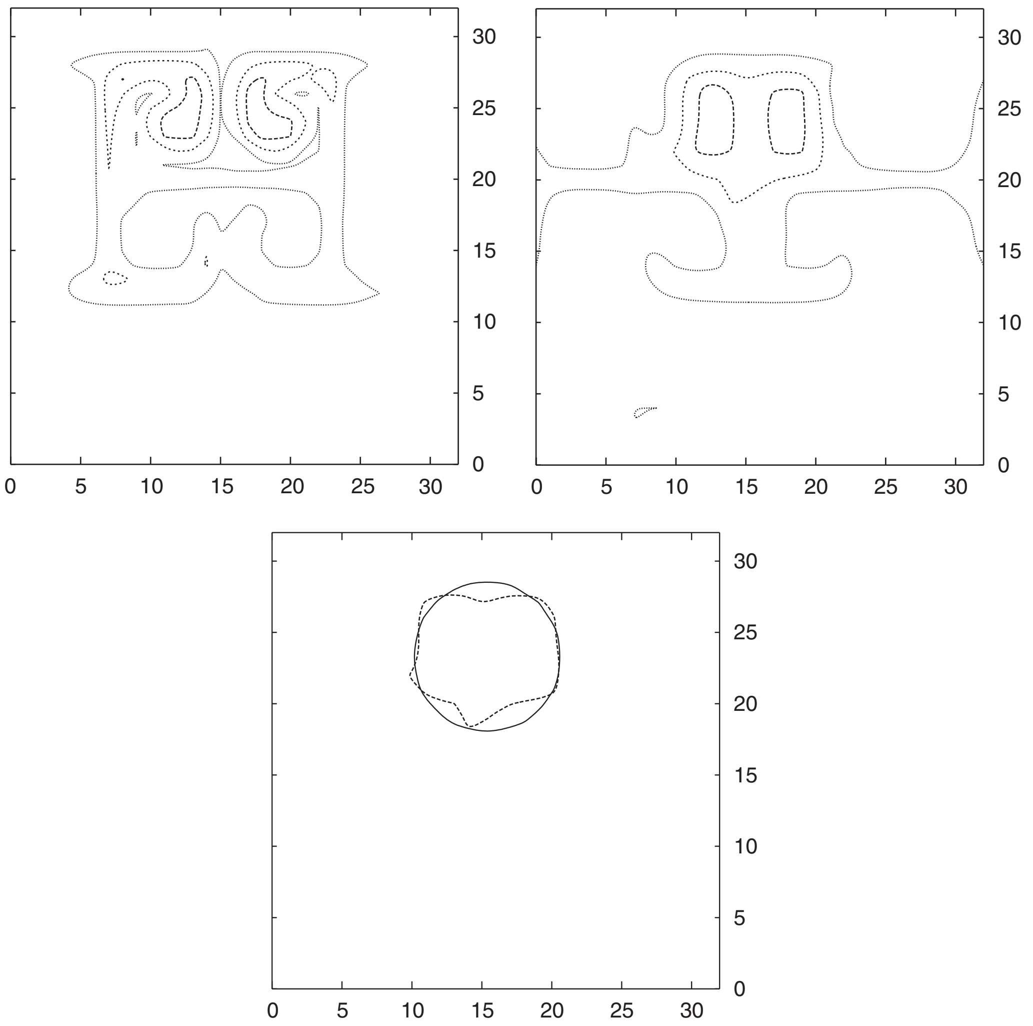

Figure 10. Multi-vortex deformational flow test on a 32×32 mesh with $T = 2$. The velocity field reverses at $t = T/2$, and restores the configuration back to its initial state at $t = T$. Displayed are the contours of 0.05, 0.5 and 0.95 of the numerical results at $t = 1$ (top) and $t = 2$ (middle). The bottom panel shows the difference between the numerical result (dashed line) and the exact solution (solid-line circle).

expect any numerical scheme with a finite resolution to be able to restore the exact initial configuration after an overly distorting transport. Figure 6 in Reference [12] displays the final results of some interface tracking methods. Again, compared to those schemes, the C3_VOF gives promising results. We should also note that compared to other schemes which use PLIC or SLIC for interface reconstruction, the lost information during the C3_VOF advection computation seems to be more easily recovered for this restoration flow test. From the bottom panels of Figures 10–12, we observe that the final interfaces of the deformed 2D cylinder are reasonably well restored back to the exact solution even on a mesh of low resolution.

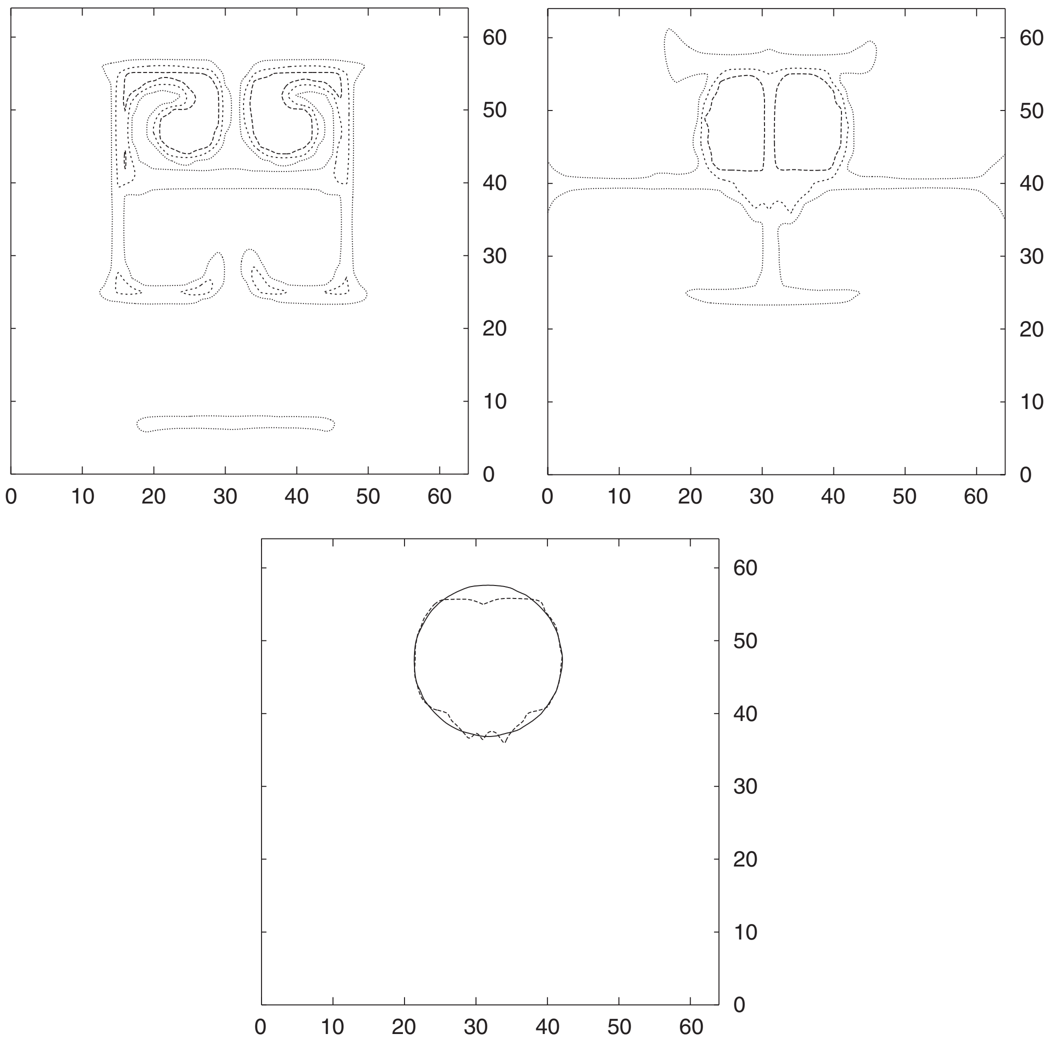

Figure 11. Same as Figure 10, but on a 64 × 64 mesh.

Table III. L_1 errors and convergence rates of C3_VOF on the multi-vortex deformation problem.

32×32	Order	64×64	Order	128×128
4.24×10^{-2}	0.96	2.03×10^{-2}	1.06	1.04×10^{-2}

This reveals to some extent that the C3_VOF can be effectively used as an interface capturing scheme even in a low resolution simulation.

The L_1 errors and the convergence rates for this test problem given in Table III also share a first order accuracy.

Copyright © 2003 John Wiley & Sons, Ltd.

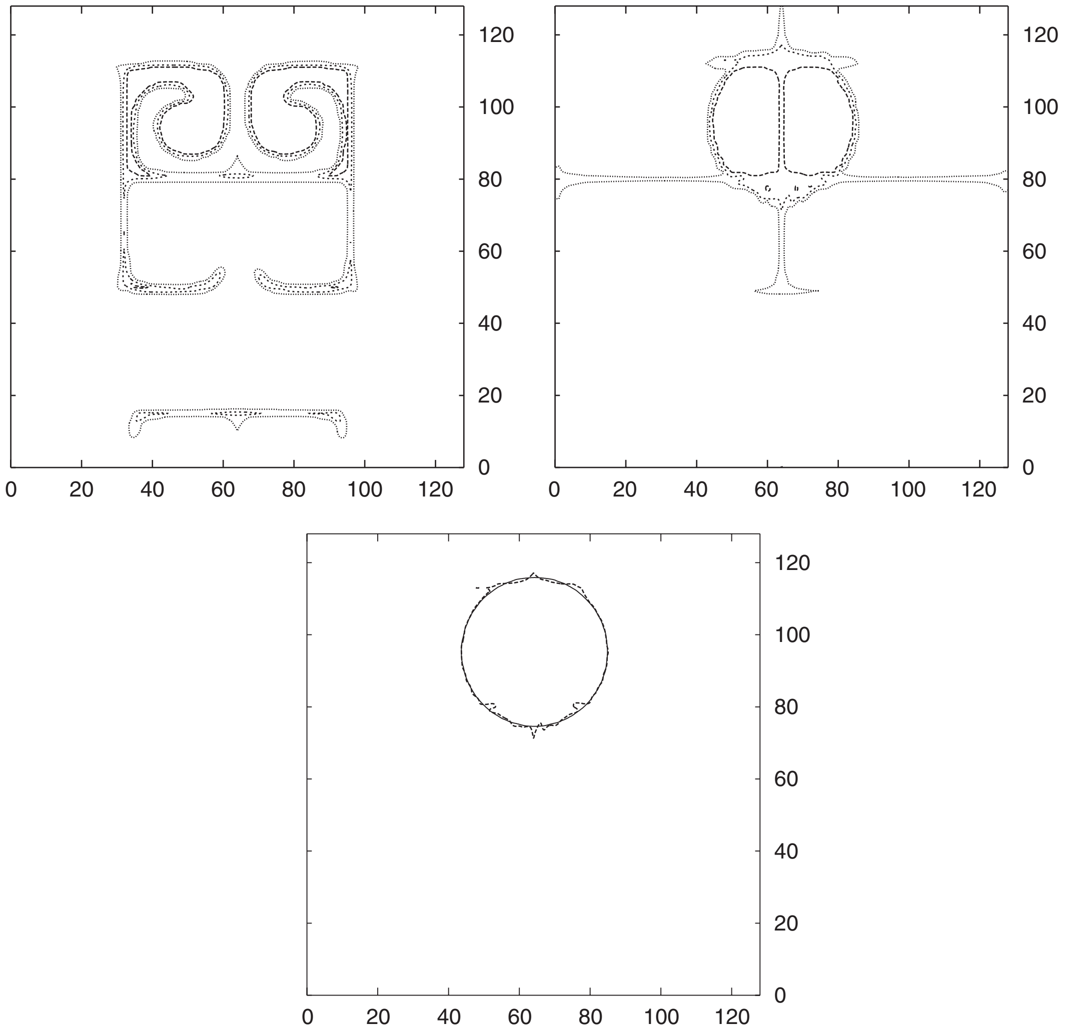

Figure 12. Same as Figure 10, but on a 128×128 mesh.

We have applied the interface tracking method to the simulation of multi-fluid dynamics. The C3_VOF was incorporated in a numerical code for incompressible flow. In the fluid dynamic code, a fractional solution procedure similar to that in References [24] and [25] is used. The advection is computed by the CIP scheme [26]. In the real hydrodynamic simulations reported in this paper, the gravitational acceleration is $g = 9.8\,\mathrm{m/s^2}$. The densities for air and liquid are specified as $\rho_{\mathrm{a}} = 1.1763\,\mathrm{kg/m^3}$ and $\rho_{\mathrm{l}} = 996.62\,\mathrm{kg/m^3}$, respectively. The viscosity coefficients are $\mu_{\mathrm{a}} = 18.62 \times 10^{-6}$ Pa s and $\mu_{\mathrm{l}} = 854.4 \times 10^{-6}$ Pa s. The air/water interface is identified by the 0.5 contour of the VOF function $f(x, y, t)$ computed by C3-VOF scheme, and the density and the viscosity coefficient over the whole computational domain are

Copyright © 2003 John Wiley & Sons, Ltd.

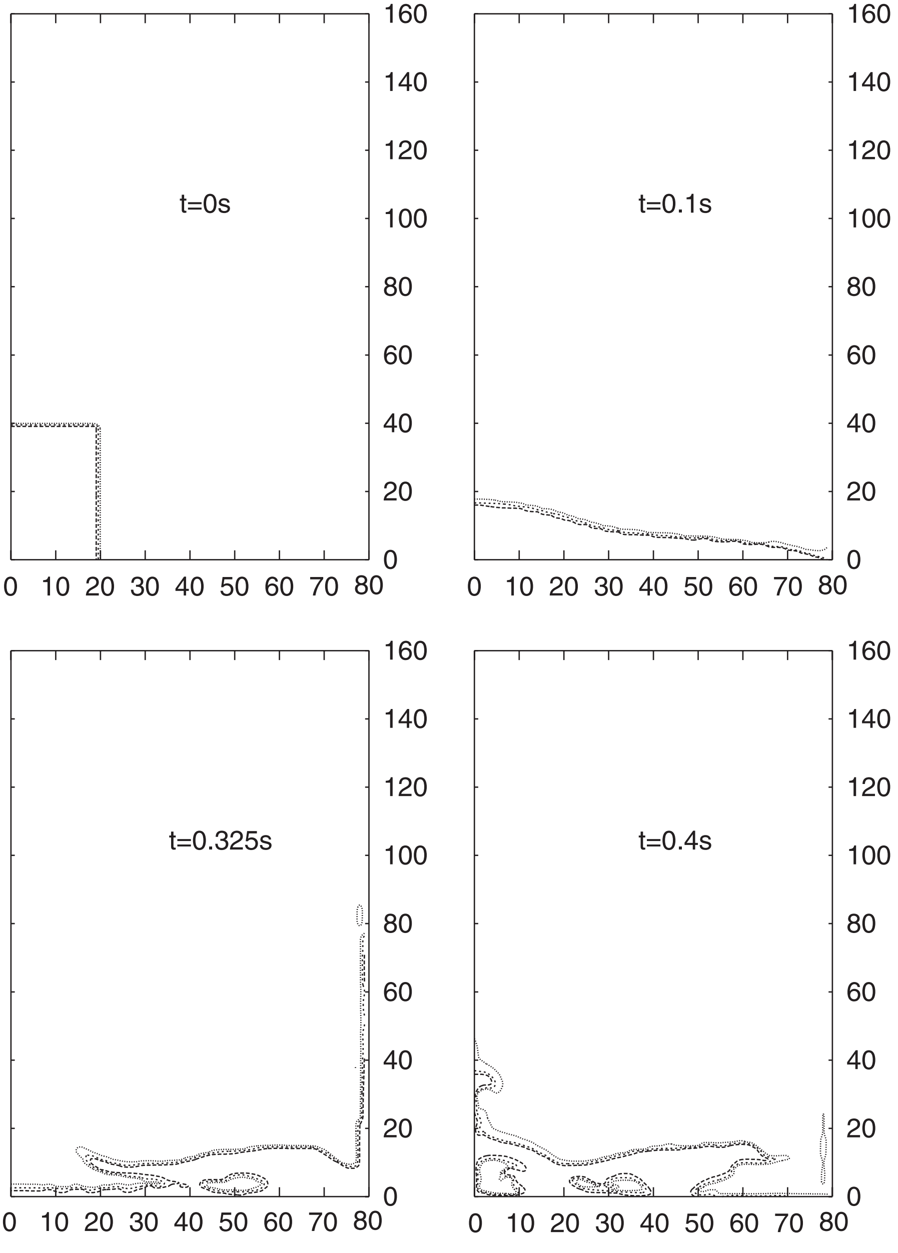

Figure 13. The VOF function of the density current generated by the release of a rectangular water dam. Displayed are the contours of 0.05, 0.5 and 0.95. The interface is computed by the C3_VOF scheme with sharpness enhancement (20).

Copyright © 2003 John Wiley & Sons, Ltd.

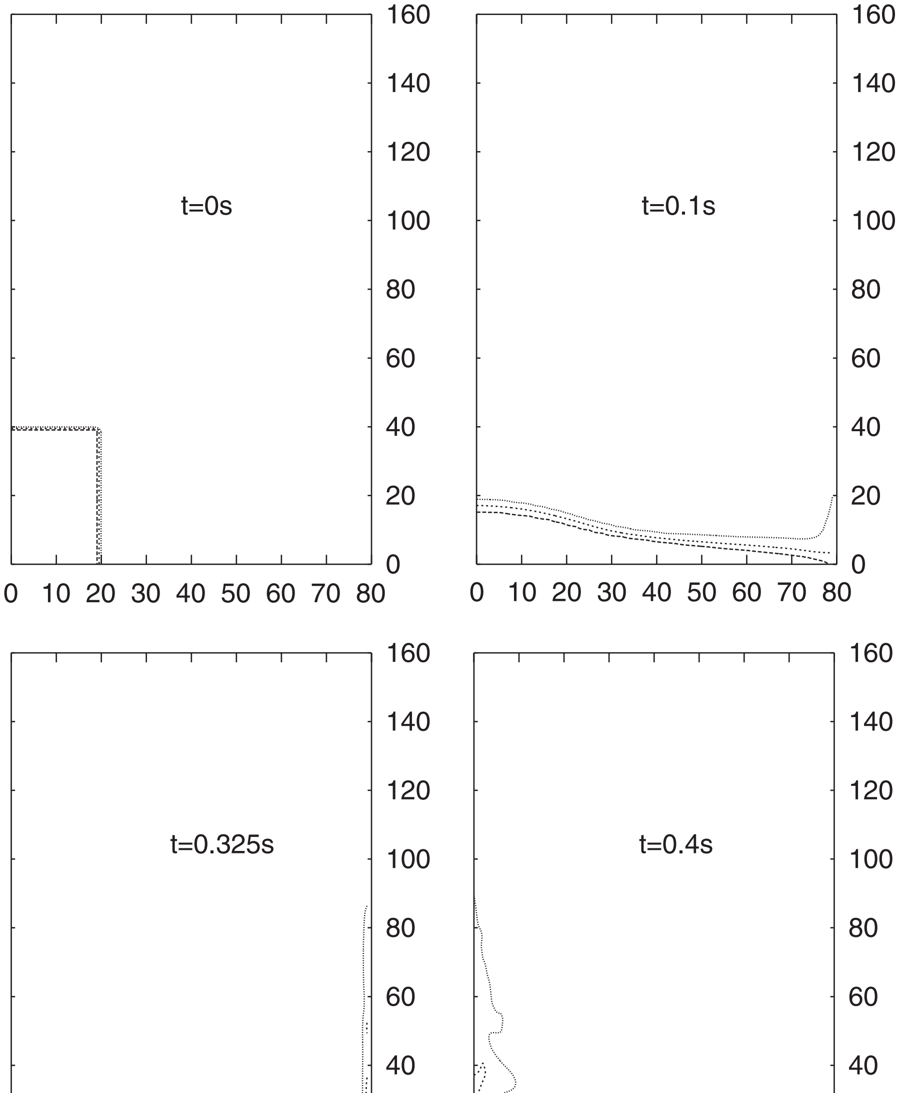

Figure 14. The VOF function of the density current generated by the release of a rectangular water dam. Displayed are the contours of 0.05, 0.5 and 0.95. The interface is computed by the original CSL3 scheme without sharpness enhancement (i.e. $\beta = 1.0$ in (20)).

Copyright © 2003 John Wiley & Sons, Ltd.

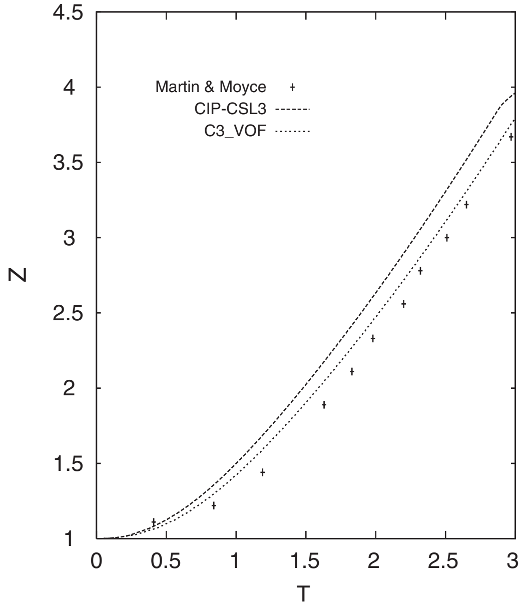

Figure 15. The displacement of the water front. Same as Reference [27], the normalized distance is defined by $Z=z/a$ with z being the real distance of the surge front and a the width of the initial water column. The time is normalized as $T=\sqrt{2g/a}t$, where t is the real time and g the gravitational acceleration. Displayed are the simulation results of C3_VOF with β defined by (20) and the original CIP-CSL3.

defined by

$$\rho(x,y,t)=\bar{f}(x,y,t)\rho_{\mathrm{l}}+[1-\bar{f}(x,y,t)]\rho_{\mathrm{a}} \tag{24}$$

and

$$\mu(x,y,t)=\bar{f}(x,y,t)\mu_{\mathrm{l}}+[1-\bar{f}(x,y,t)]\mu_{\mathrm{a}} \tag{25}$$

In order to validate the volume tracking method presented in this paper, we computed the dam-breaking problem, which was experimentally studied by Martin and Moyce [27] a half century ago and then widely used as a benchmark test for numerical models. A 80×160 mesh, with the grid spacing being uniformly 1 mm, is used. The initial rectangular water column has a width $a=20\,\mathrm{mm}$ and a height of $2a$. Shown in Figure 13, the collapse of the water column creates a complex interfacial flow which is characterized by the breakup and coalescence of liquid, entrained air bubbles and a heavily distorted interface. The C3_VOF method effectively prevented the smearing of the interface and the thickness of the transition layer of the interface was kept compact even after the interface had experienced topological changes. The air bubbles trapped in the water during the latter stage ($t=0.325\,\mathrm{s}$ and $t=0.4\,\mathrm{s}$) were simulated with well defined interfaces. Recall that the numerical diffusion is suppressed in C3_VOF through a modification of the slope which is a free parameter for the interpolation function of the CIP-CSL3 scheme. We repeated the simulation but without the slope modification for the

Copyright © 2003 John Wiley & Sons, Ltd.

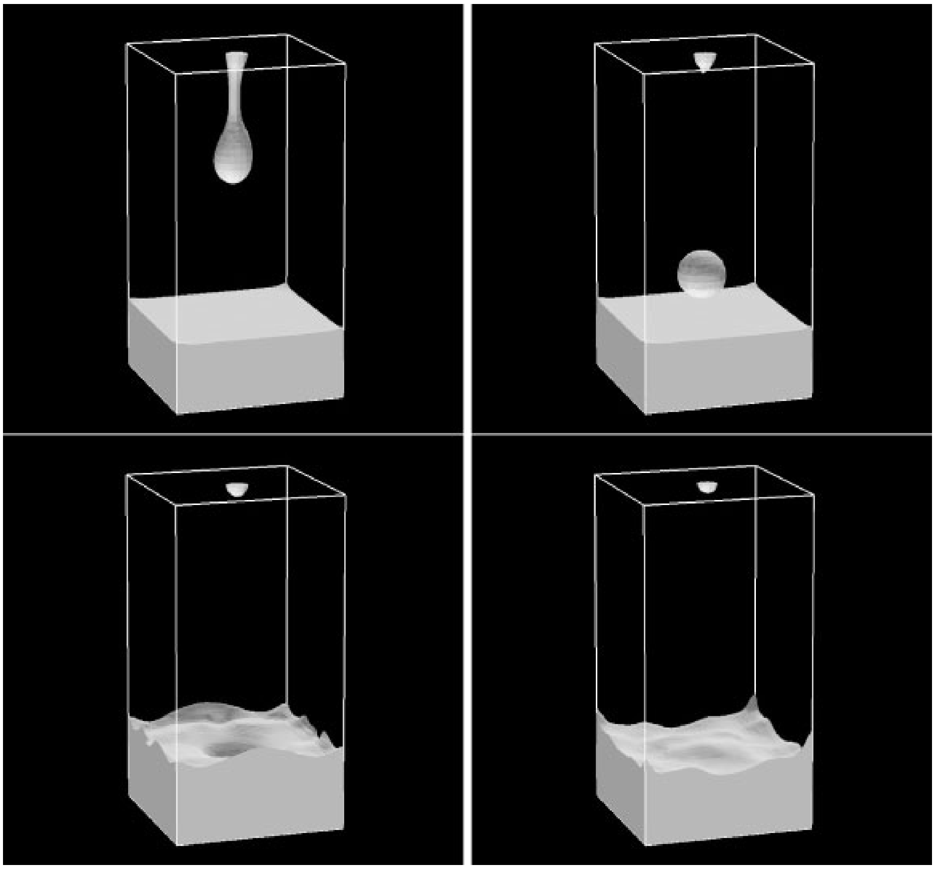

Figure 16. Simulation of the interaction between a falling drop and liquid.

sharpness reinforcement, i.e. the original CIP-CSL3 scheme was used. Figure 14 shows the corresponding results. In this case, the interface was continuously spread out so that the trapped bubbles were not adequately resolved. The bubble around the left-lower corner could not be recognized. We validated the computational model by examining the displacement of the water front along the lower surface. The scaled displacement is plotted in Figure 15. The computational result of the C3_VOF agrees well with the experiment, while the original CIP-CSL3 gave a less accurate front speed.

The 3D implementation of the C3_VOF is straightforward. The interfaces in our 3D computations were also well resolved. Figure 16 shows the snapshots of the interaction between a falling drop and a mass of liquid. The numerical model is similar to the 2D one used in the previous test. The surface tension model is the CSF formulation proposed by Brackbill *et al.* [28]. The surface tension coefficient is $\sigma = 71.69 \times 10^{-3}$. A $30 \times 30 \times 50$ computational grid was used. The drop was initially attached onto the top of a sealed tank. As can be expected, it was torn under the force of gravity, and the major part fell downward to the liquid surface while the remainder adhered to the ceiling due to capillary force. The impact of the falling drop caused a splashing of the liquid. The entire process was reasonably well simulated. The total mass of both the drop and the liquid was conserved in this simulation.

5. CONCLUSIONS

An efficient numerical method to track a free moving boundary is obtained by simply incorporating a slope modification to the CIP_CSL3 scheme. The resulting method, C3_VOF, conserves the mass of the transported quantity, and is able to preserve the compact thickness of the interface, without significant numerical oscillation. Our numerical tests show that the present method gives satisfactory results for even a heavily deforming velocity field. Without surface reconstruction, the C3_VOF appears very computationally efficient and the implementation in 3D is straightforward. By the preliminary application in multi-fluid computation, we can expect the present method to be applicable to a wider spectrum of interfacial flows in real problems.

ACKNOWLEDGEMENTS

This work is supported in part by the grant-in-aid for scientific research of Japan Society for the Promotion of Science (No. 13750137).

REFERENCES

1. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *The Physics of Fluids* 1976; **24**:2182–2189.
2. Noh WF, Woodward P. SLIC (simple line interface method). *Lecture Note in Physics* 1976; **24**:330–340.
3. Hirt CW, Nichols BD. Volume of fluid (VOF)methods for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
4. 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; **24**:273–285.
5. Ashgriz A, Poo JP. FLAIR: Flux line-segment model for advection and interface reconstruction. *Journal of Computational Physics* 1991; **93**:449–468.
6. Puckett EG, Almgren AS, Bell JB, Marcus DL, Rider WJ. A high-order projection method for tracking fluid interface in variable density incompressible flows. *Journal of Computational Physics* 1997; **130**:269–282.
7. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
8. Harvie DJE, Fletcher DF. A new volume of fluid advection algorithm: the stream scheme. *Journal of Computational Physics* 2000; **162**:1–32.
9. Sethian JA. *Level Set Methods and Fast Marching Methods*. Cambridge University Press: Cambridge, 1999.
10. 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.
11. Xiao F, Yabe T, Peng D, Kobayashi H. Conservative and oscillation-less atmospheric transport schemes based on rational functions. *Journal of Geophysical Research* 2002; **107**:4609.
12. Rider WJ, Kothe DB. Stretching and tearing interface tracking methods. *Los Alamos National Laboratory Report, LA-UR-95-1145* (*AIAA-95-1717*), 1995.
13. Harten A. The artificial compression method for computation of shocks and contact discontinuities. I. Single conservation laws. *Communications on Pure and Applied Mathematics* 1977; **30**:611–638.
14. Yang H. An artificial compression method for ENO schemes: The slope modification method. *Journal of Computational Physics* 1990; **89**:125–160.
15. Pan D, Chang CH. The capturing of free surfaces in incompressible multi-fluid flows. *International Journal for Numerical Methods in Fluids* 2000; **30**:203–222.
16. Xiao F, Yabe T. Completely conservative and oscillation-less semi-Lagrangian schemes for advection transportation. *Journal of Computational Physics* 2001; **170**:498–522.
17. Yabe T, Tanaka R, Nakamura T, Xiao F. Exactly conservative semi-Lagrangian scheme (CIP-CSL) in one dimension. *Monthly Weather Review* 2001; **129**:332–344.

Copyright © 2003 John Wiley & Sons, Ltd.

18. van Leer B. Toward the ultimate conservative difference scheme. Part IV: A new approach to numerical convection. *Journal of Computational Physics* 1977; **23**:276–299.
19. van Leer B. Toward the ultimate conservative difference scheme. Part V: A second order sequel to Godunov's method. *Journal of Computational Physics* 1979; **32**:101–136.
20. Collela R, Woodward PR. The piece-wise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics* 1984; **54**:174–201.
21. Yabe T, Aoki T. A universal solver for hyperbolic-equations by cubic-polynomial interpolation. 1. one-dimensional solver. *Computer Physics Communications* 1991; **66**:219–232.
22. Rudman M. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
23. Smolarkiewicz P. The multi-dimensional Crowley advection scheme. *Monthly Weather Review* 1982; **110**:1968–1983.
24. Yabe T, Wang PY. Unified numerical procedure for compressible and incompressible fluid. *Journal of The Physical Society of Japan* 1991; **60**:2105–2108.
25. Yoon SY, Yabe T. The unified simulation for incompressible and compressible flow by the predictor–corrector scheme based on the CIP method. *Computer Physics Communications* 1999; **119**:149–158.
26. Yabe T, Ishikawa T, Wang PY, Aoki T, Kadota Y, Ikeda F. A universal solver for hyperbolic-equations by cubic-polynomial interpolation. 2. Two and three dimensional solvers. *Computer Physics Communications* 1991; **66**:233–242.
27. Martin JC, Moyce WJ. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London*, *Series A* 1952; **244**:312–324.
28. Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.

Copyright © 2003 John Wiley & Sons, Ltd.