INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:1653–1671
Published online 23 August 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1563

Three-dimensional lattice Boltzmann interface capturing method for incompressible flows

H. W. Zheng[1], C. Shu[1,*,†], Y. T. Chew[1] and J. H. Sun[2]

[1]*Department of Mechanical Engineering, National University of Singapore, Singapore 119260, Singapore*
[2]*Department of Ergonomics and Environmental Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, People's Republic of China*

SUMMARY

This paper presents a new three-dimensional lattice Boltzmann interface capturing method for incompressible flows following the work of Zheng *et al*. (*Phys. Rev. E* 2005; **72**:056705). As shown in the paper, the fourth rank isotropic property of the lattice tensor is not needed for interface capturing. As a result, a new D3Q7 (D3 means three dimensional, Q7 means seven velocity bits) lattice velocity model and its associated equilibrium distribution functions are proposed in the paper. The proposed model is validated by comparing its numerical results with those of an existing lattice Boltzmann interface capturing model (*J. Comput. Phys.* 2004; **198**:628–644) and three-dimensional direction split flux-corrected transport method (*Int. J. Numer. Meth. Fluids* 1997; **24**:671–691). Numerical results showed that the present model performs better than the existing methods in capturing the interface. It greatly improves the computational efficiency and saves at least half of the memory as compared to other lattice Boltzmann interface capturing models. Copyright © 2007 John Wiley & Sons, Ltd.

Received 24 February 2007; Revised 17 May 2007; Accepted 27 May 2007

KEY WORDS: interface capturing; three dimensional; lattice Boltzmann; incompressible flow

1. INTRODUCTION

The dynamics of multiphase flows has many practical applications in engineering, such as in liquid–vapor flow and oil–water flow. Due to complexity of the problem, the experimental study is usually very difficult. On the other hand, as the development of computers and numerical algorithms, the numerical study of multiphase flows plays a more and more important role. In general, most of multiphase flows are three dimensional and different phases may coalesce and detach from each

*Correspondence to: C. Shu, Department of Mechanical Engineering, National University of Singapore, Singapore 119260, Singapore.
†E-mail: mpeshuc@nus.edu.sg

other. Thus, accurate capturing or tracking of the three-dimensional interface is very important for multiphase flow applications.

Most of the existing methods use the Eulerian frame to track or capture the interface since they do not need to move the mesh and are easy to treat coalesce or breakup among different phases. The exact locations of the interface are determined by solving the transport equations. The main differences of various methods lie on the indication function and difference schemes for discretization of the convection term. For example, the volume of fluid (VOF) method uses the volume fraction as an indication function. To discretize the convection term, most of the VOF methods follow the original donor acceptor scheme [1]. This scheme combines the first-order upwind and downwind fluxes to ensure stability and minimize the diffusion [2]. The interface at each step is reconstructed from the volume fraction. However, the reconstruction of the interface may not be unique and depend on the reconstruction scheme. To improve the performance of the interface tracking, some higher-order reconstruction schemes (e.g. [3, 4]) were developed. It is indicated that most of these schemes are very complicated and not easy to be extended to the three-dimensional case. In contrast, the level set method (LSM) [5] utilizes a level set function to indicate the interface. One of its advantages is that the level set function varies smoothly across the interface while the volume fraction is discontinuous (step function). As compared to the volume fraction, it is just an indicator and has no physical meaning. Thus, the function does not need to keep conservation. The drawback of LSM is that, when large topological changes occur around the interface, it requires a re-initialization procedure to keep the distance property. This may violate mass conservation for each phase or component [6]. Its extension to the three-dimensional case is also not easy.

Apart from the VOF and LSM, another efficient method for capturing the interface is the phase field method (or called diffuse interface method). It uses the convective Cahn–Hilliard equation [7], which has an explicit anti-diffusivity term, to capture the interface. The anti-diffusivity leads to thin, well-defined interfaces. A major advantage of the phase field method is that no special convective algorithms are required for the interface equation. Calculations have been carried out successfully even using the standard central difference schemes. Its main problem is that some additional efforts need to be put in the approximation of the fourth-order derivatives in the diffusion term. This difficulty can be overcome in the context of lattice Boltzmann method. The inclusion of the diffusion effect in the collision term is natural in the lattice Boltzmann method. Currently, there are many attempts in this aspect. However, most of them cannot accurately recover the Cahn–Hilliard equation. For example, the original free-energy-based LBM [8] recovers the Cahn–Hilliard equation with some additional terms. Inamuro *et al*. [9] recovered a similar equation where the physical background is not so clear, and there is one additional time derivative term which may violate the Galilean invariance. He *et al*. [10] recovered the Cahn–Hilliard equation with some additional terms. Besides, the mobility in their method is related to the density. Lee and Lin [11] used the same method as that of He *et al*. [10] but replaced the order parameter by the density. The numerical discretization of the convection term is introduced to enhance the stability. To improve the lattice Boltzmann method for interface capturing, Zheng *et al*. [12] recently proposed a new method which can recover the Cahn–Hilliard equation without additional terms and can keep the Galilean invariance. In this method, the modified lattice Boltzmann equation [13] is adopted to remove the time derivative related term. In addition, it does not require the fourth-rank lattice tensor of the discrete velocity model to be isotropic. Thus, the D2Q5 discrete velocity model (D2 means two dimensional, Q5 means five velocities) can be used in the two-dimensional applications. This greatly reduces the computational effort. In this paper, we follow the same idea of

DOI: 10.1002/fld

Zheng *et al.* [12] to propose a three-dimensional lattice Boltzmann interface capturing method. In the method, a new D3Q7 lattice velocity model and its associated equilibrium distribution functions are presented. As compared to the conventional D3Q15 and D3Q19 models, the use of D3Q7 model can greatly improve the computational efficiency in the three-dimensional applications.

The paper is organized as follows. Section 2 proposes the new interface capturing method for the three-dimensional case. It recovers the convective Cahn–Hilliard equation to the second order of accuracy without additional terms. The proposed method will then be verified and applied to some test cases in Section 3. The numerical results will be compared with those of the existing lattice Boltzmann interface capturing methods and the direction split FCT-based VOF method. Finally, the conclusion will be drawn in Section 4.

2. METHODOLOGY

As stated in the Introduction, LBM can be applied to solve the convective Cahn–Hilliard equation [7] which involves not only the convective term but also the diffusion term. However, most of the existing lattice Boltzmann methods cannot completely recover the corresponding LBE to the convective Cahn–Hilliard equation shown as follows:

$$\partial_t \phi + \nabla \cdot (\phi \mathbf{u}) = \theta_M \nabla^2 \mu_\phi \tag{1}$$

where θ_M is called the mobility and μ_ϕ is the chemical potential.

In this paper, we intend to propose a new three-dimensional lattice Boltzmann model which can recover the Cahn–Hilliard equation without any additional terms by following the work of Zheng *et al.* [12]. To achieve this, the modified lattice Boltzmann equation [13] is adopted:

$$f_i(\mathbf{x} + \mathbf{e}_i\delta, t + \delta) = f_i(\mathbf{x}, t) + (1 - q)[f_i(\mathbf{x} + \mathbf{e}_i\delta, t) - f_i(\mathbf{x}, t)] + \Omega_i \tag{2}$$

with BGK approximation [14] of the collision term,

$$\Omega_i = \frac{f_i^{(0)}(\mathbf{x}, t) - f_i(\mathbf{x}, t)}{\tau_\phi} \tag{3}$$

where f_i is the distribution function, τ_ϕ is the dimensionless single relaxation time, $\mathbf{e}_i$ is the lattice velocity, and q is the constant coefficient.

The macroscopic variable (order parameter) ϕ is evaluated by

$$\phi = \sum_i f_i \tag{4}$$

By using Taylor series expansion truncated to the second-order term, Equation (2) can be rewritten as

$$\begin{aligned} &\delta(\partial_t + \mathbf{e}_i \cdot \nabla) f_i + \tfrac{1}{2}\delta^2(\partial_t + \mathbf{e}_i \cdot \nabla)^2 f_i \\ &\quad + (q - 1)[\delta(\mathbf{e}_i \cdot \nabla) f_i + \tfrac{1}{2}\delta^2(\mathbf{e}_i \cdot \nabla)^2 f_i] + O(\delta^3) = \Omega_i \end{aligned} \tag{5}$$

DOI: 10.1002/fld

Furthermore, by applying the Chapman–Enskog expansion to Equation (5),

$$\begin{aligned} f_i &\approx f_i^{(0)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} \\ \partial_t &\approx \varepsilon \partial_{t^0} + \varepsilon^2 \partial_{t^1} \\ \partial_x &\approx \varepsilon \partial_{x^1} \\ \Omega &\approx \Omega^{(0)} + \varepsilon \Omega^{(1)} + \varepsilon^2 \Omega^{(2)} \end{aligned} \tag{6}$$

where the physical meaning of ε is the Knudsen number, we can show that if the distribution function satisfies the following conservation laws

$$\sum_i f_i^{(0)} = \phi \tag{7}$$

$$\sum_i f_i^{(0)} e_{i\alpha} \equiv \frac{1}{q} \phi u_\alpha \tag{8}$$

$$\sum_i f_i^{(0)} e_{i\alpha} e_{i\beta} = \Gamma \mu_\phi \delta_{\alpha\beta} \tag{9}$$

and the coefficient q is set to be

$$q = \frac{1}{\tau_\phi + 0.5} \tag{10}$$

then Equation (2) can recover the Cahn–Hilliard equation with the second order of accuracy

$$\partial_t \phi + \nabla \cdot (\phi \mathbf{u}) - \theta_M \nabla^2 \mu_\phi + O(\delta^2) = 0 \tag{11}$$

where the mobility is defined as

$$\theta_M = q(\tau_\phi q - \tfrac{1}{2}) \delta \Gamma \tag{12}$$

As shown in Equation (12), Γ is a parameter, which is proportional to mobility. The change of Γ will lead to the change of mobility. The chemical potential in Equation (9) is related to the free energy density functional which is taken as

$$F = \int \mathrm{d}V \left\{ a(\phi^2 - \phi^{*2}) + \frac{\kappa}{2} (\nabla \phi)^2 \right\} \tag{13}$$

where ϕ^* is a constant which is related to the equilibrium state. By minimizing the free energy functional, the chemical potential is calculated from the Euler–Lagrange equation [12]:

$$\mu_\phi = a(4\phi^3 - 4\phi^{*2}\phi) - \kappa \nabla^2 \phi \tag{14}$$

To compute the second-order derivative term in the above equation, we apply the Taylor series expansion:

$$\phi(\mathbf{x} + \mathbf{e}_i \delta) - \phi(\mathbf{x}) = \mathbf{e}_i \delta \cdot \nabla \phi(\mathbf{x}) + \tfrac{1}{2} (\mathbf{e}_i \delta \cdot \nabla)^2 \phi(\mathbf{x}) + \tfrac{1}{6} (\mathbf{e}_i \delta \cdot \nabla)^3 \phi(\mathbf{x}) + O(\delta^3) \tag{15}$$

DOI: 10.1002/fld

Summation of the above equation gives

$$\sum_i [\phi(\mathbf{x}+\mathbf{e}_i\delta)-\phi(\mathbf{x})] = \delta\partial_\beta\phi(\mathbf{x})\sum_i e_{i\beta} + \frac{1}{2}\delta^2\partial_\alpha\partial_\beta\phi(\mathbf{x})\sum_i (e_{i\alpha}e_{i\beta}) + \frac{1}{6}\delta^3\partial_\alpha\partial_\beta\partial_\gamma\phi(\mathbf{x})\sum_i e_{i\alpha}e_{i\beta}e_{i\gamma} + O(\delta^3) \tag{16}$$

Clearly, if the first-, second- and third-rank lattice tensors are isotropic, the second-order accuracy of $\nabla^2\phi$ can be achieved by the following approximation:

$$\nabla^2\phi \approx \frac{2}{(\delta)^2\sum_i (e_{i\beta})^2}\sum_i [\phi(\mathbf{x}+\mathbf{e}_i\delta)-\phi(\mathbf{x})] \tag{17}$$

There is no velocity-related terms in Equation (9). This shows that it does not require the fourth-rank lattice tensor to be isotropic as needed by the conventional lattice Boltzmann methods. Based on this analysis and Equation (17), we can draw a conclusion that the sufficient condition for a suitable lattice velocity model is that the first-, second- and third-rank lattice tensors are isotropic. For the three-dimensional case, the minimum number of lattice velocities that satisfies these conditions is 7 and the corresponding lattice velocity model is D3Q7 (D3 means three dimensional, Q7 means seven velocity). Thus, D3Q7 is used in this paper. The discrete velocities of this model can be taken as

$$\mathbf{e}_1 = (0,0,0)^{\mathrm{T}}, \quad \mathbf{e}_{2,5} = (\pm 1,0,0)^{\mathrm{T}}, \quad \mathbf{e}_{3,6} = (0,\pm 1,0)^{\mathrm{T}}, \quad \mathbf{e}_{4,7} = (0,0,\pm 1)^{\mathrm{T}} \tag{18}$$

We can obtain the lattice tensors of this model as

$$\begin{aligned} \sum_i e_{i\alpha} &= 0 \\ \sum_i e_{i\alpha}e_{i\beta} &= 2\delta_{\alpha\beta} \\ \sum_i e_{i\alpha}e_{i\beta}e_{i\gamma} &= 0 \end{aligned} \tag{19}$$

It could be easily verified that the first-, second- and third-rank lattice tensors are isotropic. Thus, the second-order derivative term in Equation (14) can be approximated by Equation (17) with the second order of accuracy. Apart from the lattice velocity model, a lattice Boltzmann interface capturing method should also satisfy the conservation laws given by Equations (7)–(9). To satisfy all the conservation laws, we need to define a suitable equilibrium distribution function.

According to Equations (7)–(9), the equilibrium distribution function in Equation (3) can be chosen as

$$f_i^{(0)} = A_i + B_i\phi + C_i\phi\mathbf{e}_i\cdot\mathbf{u} \tag{20}$$

The coefficients are taken as

$$A_1 = -D\Gamma\mu_\phi, \quad A_i = \frac{1}{2}\Gamma\mu_\phi (i\neq 1), \quad B_1 = 1, \quad B_i = 0\ (i\neq 1), \quad C_i = \frac{1}{2q} \tag{21}$$

where D is the dimension.

DOI: 10.1002/fld

Here, we would like to highlight the efficiency of our method in three-dimensional cases. As we all know, each equilibrium distribution function is calculated in each time step. Thus, the computational effort of the lattice Boltzmann method is approximately proportional to [NX · NY · NZ · kp · istep], where NX, NY and NZ are the number of mesh points in the x, y and z direction, respectively, kp is the number of lattice velocities in the velocity model and istep is the iteration number. As kp is only 7 in our model as compared to 15 or even 19 other models, the efficiency of our method is greatly improved. Apart from the improvement of efficiency, the memory requirement is also reduced due to the use of fewer velocities in the velocity model.

To validate the proposed method, we compare our results with those of direction split flux-corrected transport (FCT) VOF method [2]. The basic idea of one-dimensional FCT involves the predicator step using a lower-order monotonic (and hence diffusive) advection scheme and a corrector step with an anti-diffusive flux to correct the numerical diffusion resulting from the low-order scheme. To extend it to multi-dimensions, there are two methods [2]. They are Zalesak's fully multi-dimensional FCT algorithm [15] and a direction split FCT. In practice, the direction split FCT–VOF gives superior results. Thus, the direction split FCT (DSFCT) is adopted in this paper. Here, we follow the idea of Rudman to give a simple extension from two dimensions to three dimensions. It involves sweeping the entire mesh in each direction with the one-dimensional algorithm followed by an updating step of the intermediate volume fraction. For example, in the x-sweep time step, the predictor is

$$C^*_{i,j,k} = C^n_{i,j,k} - \frac{\delta t}{\delta x}(F^{L(n)}_{i+1/2} - F^{L(n)}_{i-1/2}) \tag{22}$$

where $C^n_{i,j,k}$ and $F^{L(n)}_{i+1/2}$ are the volume fraction and the lower-order flux in time level n. δt and δx are the time step and mesh spacing.

Then, a corrector is applied,

$$C^{n+1/3}_{i,j,k} = \frac{\delta V^n_{i,j,k}}{\delta V^{n+1/3}_{i,j,k}}\left[C^*_{i,j,k} - \frac{\delta t}{\delta x}(A^{C(n)}_{i+1/2} - A^{C(n)}_{i-1/2})\right] \tag{23}$$

where $A^{C(n)}_{i+1/2}$ and $\delta V^n_{i,j,k}$ are the anti-diffusion term and the effective volume. The effective volume is expressed as

$$\delta V^{n+1/3}_{i,j,k} = \delta V^n_{i,j,k} - \delta t \delta V^n_{i,j,k}\frac{(u_{i+1/2,j,k} - u_{i-1/2,j,k})}{\delta x} \tag{24}$$

with

$$\delta V^n_{i,j,k} = \delta x \times \delta y \times \delta z \tag{25}$$

where δx, δy and δz are the spatial spacing in each direction. The order of these three direction sweeps is interchanged each time step to avoid the introduction of systematic error.

3. RESULTS AND DISCUSSION

As shown in Equation (18), the particle velocity in the LBM computation is usually taken as 1. So, the time interval δ in the LBM equals to the spatial mesh spacing. In the present study, the

DOI: 10.1002/fld

computational domain is regular and the uniform mesh is used with $\Delta x = \Delta y = \Delta z = 1$. So, δ in this work is always set as 1. This means that the lattice unit is always kept as 1 and the lattice time is actually the time steps.

3.1. Verification

The profile along the normal direction of the interface [12] can be expressed as

$$\phi = \phi^* \tanh(\pm 2\zeta / W) \tag{26}$$

where the sign in the bracket is dependent on the initial condition of the problem, ζ is the coordinate which is perpendicular to the interface, and the origin is at the interface with interface thickness:

$$W = \frac{\sqrt{2\kappa/a}}{\phi^*} \tag{27}$$

In this section, we use a stable sphere case to verify Equation (26). Initially, a sphere is located at the center of domain with the mesh size of $60 \times 60 \times 60$. The order parameter is set to be $-0.9\phi^*$ inside the sphere and $0.9\phi^*$ elsewhere ($\phi^* = 1$). Periodic boundary conditions are employed at all boundaries. The mobility can be regarded as a counterpart of *Pe* number for convection–diffusion equation. Thus it will influence the stability of the scheme. As the mobility is related to Γ, the Γ value is critical to the stability. Very small value of Γ may lead to unstable computation. In this work, the parameters are set to be $\kappa = 0.002$, $\Gamma = 0.4$, $a = 0.001$ and $\tau_\phi = 0.7$.

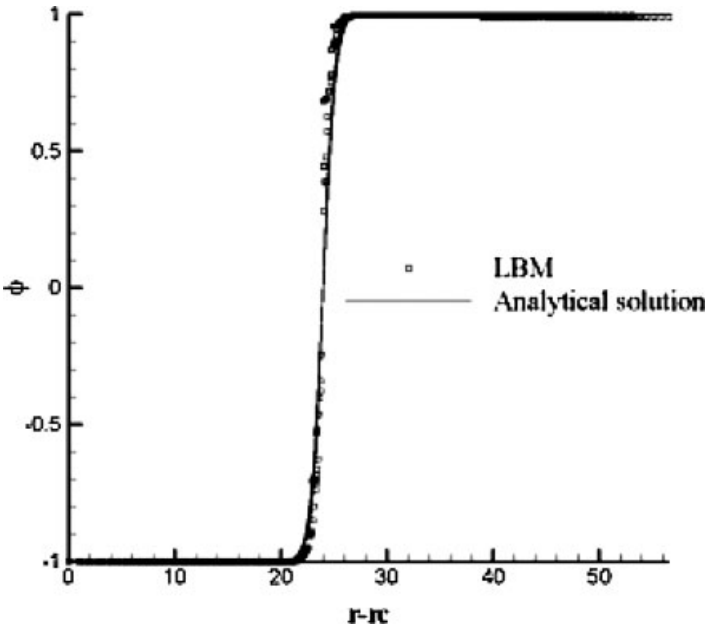

Figure 1. The interface profile.

DOI: 10.1002/fld

For this problem, the analytical interface profile can be expressed as

$$\phi = \phi^* \tanh\left[2\left(\sqrt{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2}-R\right)\Big/ W\right] \tag{28}$$

where r_c is the coordinate of the center of the sphere, and R is the radius of the sphere. We draw the order parameter at all lattice points as a function of the distance from the center point as shown in Figure 1. It shows that the order parameter is independent on the direction. Besides, the numerical results are also compared with the analytical profile described by Equation (28). As illustrated in Figure 1, the numerical order parameter profile (denoted by the symbol) agrees with the analytic solution (the solid line) very well. Several cases are used to further evaluate our method as follows.

3.2. Solid body rotation

In this section, we discuss about a designed solid body that is a combination of a spherical cap with a cross slot as illustrated in Figure 2(a). In order to observe it clearly, we draw three different figures as shown in Figure 2(b)–(d) viewing from three different directions. The computational domain is a box with $0 \leqslant i, j, k \leqslant 120$ ($\mathrm{NX} = \mathrm{NY} = \mathrm{NZ} = 120$). The radius of the sphere is 48 and the width of the bottom cross is 18 in lattice unit in both the x and y directions. The periodic boundary condition is employed at all boundaries. The parameters are chosen the same as those in Section 3.1. Initially, the solid body is located at the center of the box. The order parameter is set to be

$$\phi(x, y, z) = \begin{cases} \phi^* \tanh d & \text{if } z \geqslant 0.5 \\ -\phi^* & \text{if } z < 0.5 \text{ and } |x - x_c| \leqslant 0.075 \\ & \quad \text{and } d \leqslant 0.5 \text{ and } |y - y_c| \leqslant 0.075 \\ \phi^* & \text{else} \end{cases} \tag{29}$$

where d is defined as

$$d = \sqrt{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2} - R \tag{30}$$

The objective of this section is to test whether the surfaces of the solid body can keep their shapes under the rotational velocity field, which is taken in the lattice Boltzmann context as

$$u = 0, \quad v = -u_0\pi\left(\frac{k}{\mathrm{NZ}} - 0.5\right), \quad w = u_0\pi\left(\frac{j}{\mathrm{NY}} - 0.5\right) \tag{31}$$

It is obvious that this velocity field is a rotational field around the x-axis. As shown in the two-dimensional calculation [12], the scale parameter u_0 is a non-dimensional coefficient and is taken as 0.02 in the present computation to ensure that the computation is in the low Mach number range.

Firstly, we compare the efficiency of our method with other lattice Boltzmann methods such as the approach of Inamuro *et al.* [9]. The total physical time, which is defined in Equation (33), is remained as 2 s for all the cases. As stated in Section 2, the computational time is approximately in the order of $O[\mathrm{NX} \cdot \mathrm{NY} \cdot \mathrm{NZ} \cdot \mathrm{kp} \cdot \mathrm{istep}]$. We only vary the mesh size to compare the computational time between the present method and the approach of Inamuro *et al.* [9] on the PC with 2 GB RAM and 2.80 GHz CPU. The computational time for the two methods is listed in Table I. It

DOI: 10.1002/fld

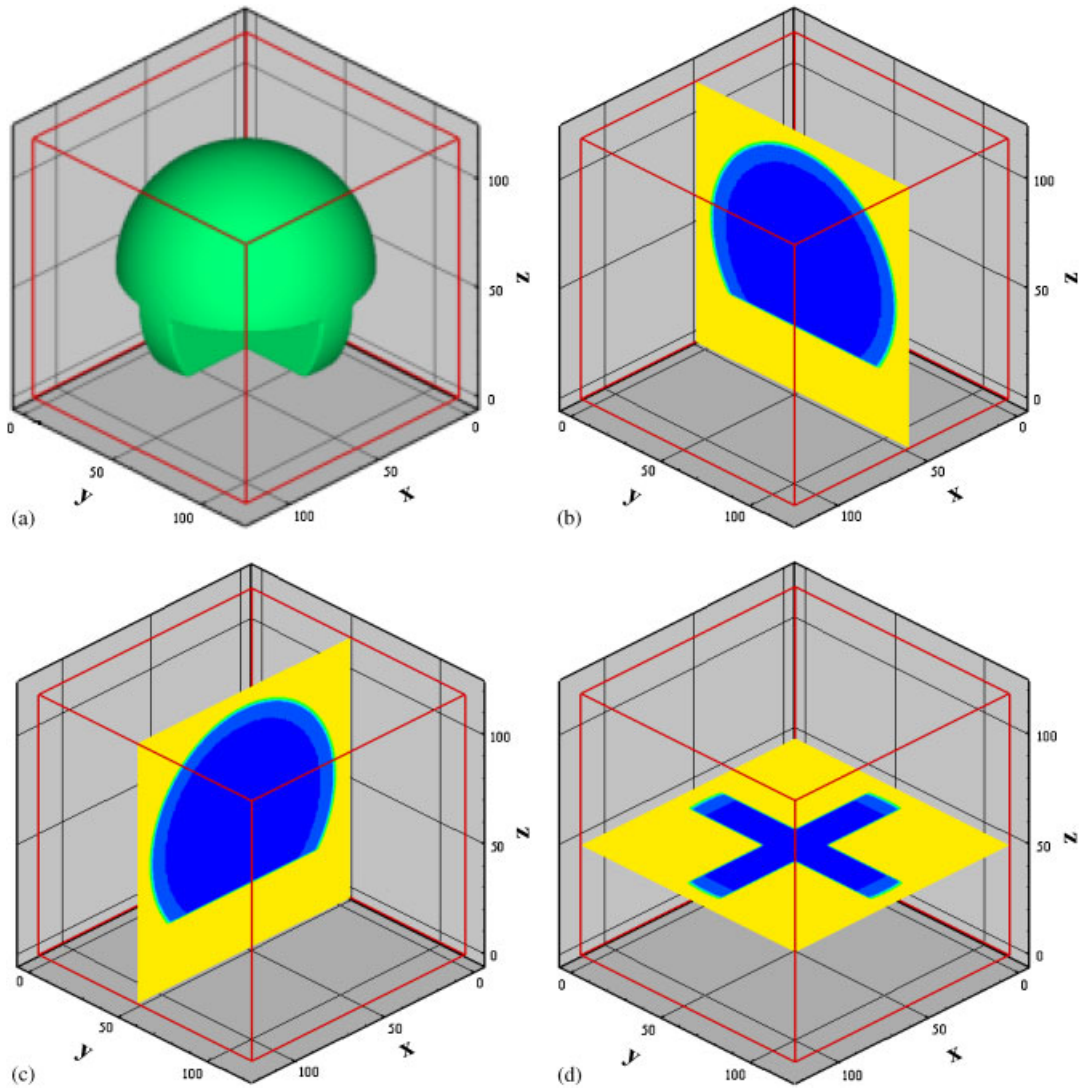

Figure 2. The initial configuration for the rotation case.

Table I. Computational time needed by the present method and the approach of Inamuro *et al.* [9].

Cases	Mesh size	Computational time (s)		Ratio
		Present method	Inamuro *et al.* [9] approach	
1	$40 \times 40 \times 40$	695	1548	2.227
2	$120 \times 120 \times 120$	55 742	124 873	2.24

DOI: 10.1002/fld

can be seen that the time needed by the present method is less than half of the time needed by the approach of Inamuro *et al.* [9]. As the mesh size increases, the ratio of computational time of present method to that of Inamuro *et al.* [9] increases from 2.22 to 2.24. This implies that the method of Inamuro *et al.* [9] will cost more than two times of computational effort than the present method. The reason may be due to the fact that 15 lattice velocities are used in the approach of Inamuro *et al.* [9] while only seven lattice velocities are used in the present method. In addition, the approach of Inamuro *et al.* [9] needs an approximation of both the first and second derivatives in the equilibrium distribution functions.

To evaluate the present method, we also compare our results with those of DSFCT VOF. The corresponding physical velocity field (the unit is m/s) is taken as

$$u^{\text{phys}} = 0, \quad v^{\text{phys}} = -\pi(z^{\text{phys}} - 0.5), \quad w^{\text{phys}} = \pi(y^{\text{phys}} - 0.5) \tag{32a}$$

and the physical domain (the unit is meter) is chosen as

$$x^{\text{phys}}, y^{\text{phys}}, z^{\text{phys}} \in [0, 1] \tag{32b}$$

To make a fair comparison, we need to get the relationship between the physical time t^{phys} and the lattice time t. To achieve this, we need to obtain the reference length L_{ref} and the reference velocity U_{ref} from the physical frame to the lattice Boltzmann frame first. By comparing Equation (32a) with Equation (31), we can easily obtain U_{ref} as $U_{\text{ref}} = 1/u_0$. On the other hand, since the mesh spacing in the lattice Boltzmann frame is always taken as 1, its computational domain is $0 \leqslant x \leqslant \text{NX}$, $0 \leqslant y \leqslant \text{NY}$, $0 \leqslant z \leqslant \text{NZ}$. As shown in Equation (32b), the physical domain is a unit box. So, the length scale L_{ref} is $L_{\text{ref}} = 1/\text{NX}$ for the case of $\text{NX} = \text{NY} = \text{NZ}$. The physical time can be computed by

$$t^{\text{phys}} = \frac{L_{\text{ref}}}{U_{\text{ref}}} t = \frac{u_0}{\text{NX}} t \tag{33}$$

With $u_0 = 0.02$ and $\text{NX} = 120$, it is clear that 6000 time steps (lattice time) in the present computation correspond to 1 s of physical time. Numerical results of DSFCT VOF and present method at 2 s (physical time) are shown, respectively, in Figures 3 and 4. For this problem, there is an analytical solution for the interface position and the direction of the bottom cross. That is, the direction of the bottom cross should go to a certain angle and the sphere should keep its shape as the initial state after one rotation. From Figures 3 and 4, it can be easily observed that the direction of the bottom cross obtained by the two methods agrees well with the analytical one. However, there are some differences for the details of interface. As can be seen from Figure 3, many slots appear around the surface of the bottom cross in the results of the DSFCT VOF method. Besides, the shape is not very smooth at the spherical surface. This indicates that DSFCT VOF gives a poor interface tracking. In contrast, Figure 4 shows that the present method generates accurate results except for some convex circular slots appeared in the links between the spherical cap and the bottom cross. Thus, we can conclude that the present method performs better than the DSFCT VOF method for this case.

3.3. Elongation

As shown in Equation (31) of Section 3.2, the solid body rotation does not induce topological deformation because the velocity gradient in each direction is zero for the case. In this section, we try to introduce a non-zero velocity gradient in each direction. That is, the interface evolution

DOI: 10.1002/fld

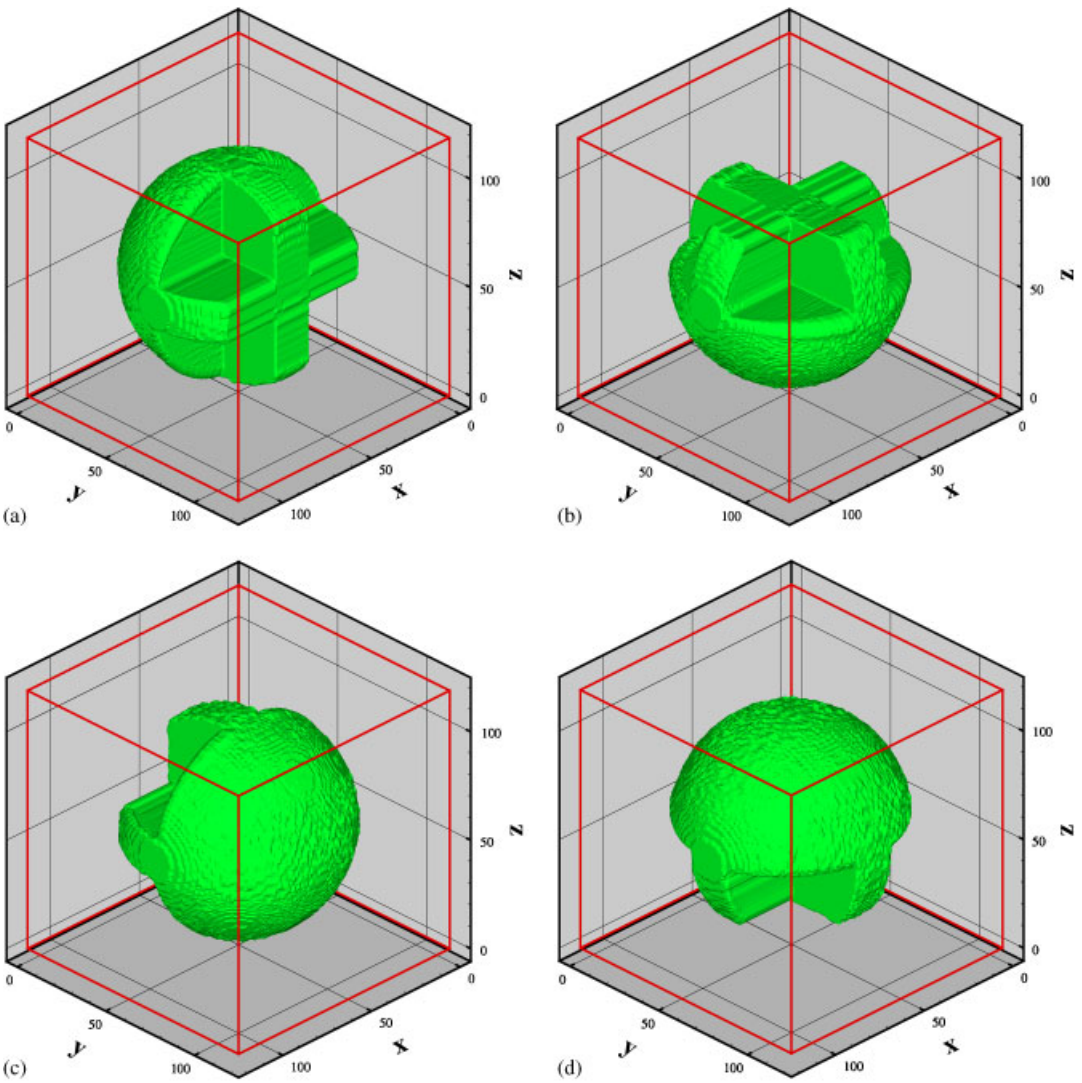

Figure 3. The results of the direction split FCT VOF method for the rotation case.

under elongation velocity field is considered:

$$u = u_0\pi\left(\frac{i}{\mathrm{NX}} - 0.5\right), \quad v = u_0\pi\left(\frac{j}{\mathrm{NY}} - 0.5\right), \quad w = -2u_0\pi\left(\frac{k}{\mathrm{NZ}} - 0.5\right) \tag{34}$$

where u_0 is the scaled parameter and is taken as 0.02, NX is the number of mesh points in the x direction. Like the previous example, the scaled parameter u_0 is used in the present computation to ensure that the computation is in the low Mach number range. Equation (34) indicates that the sphere is stretched under one direction and contracted in other two directions. That is,

DOI: 10.1002/fld

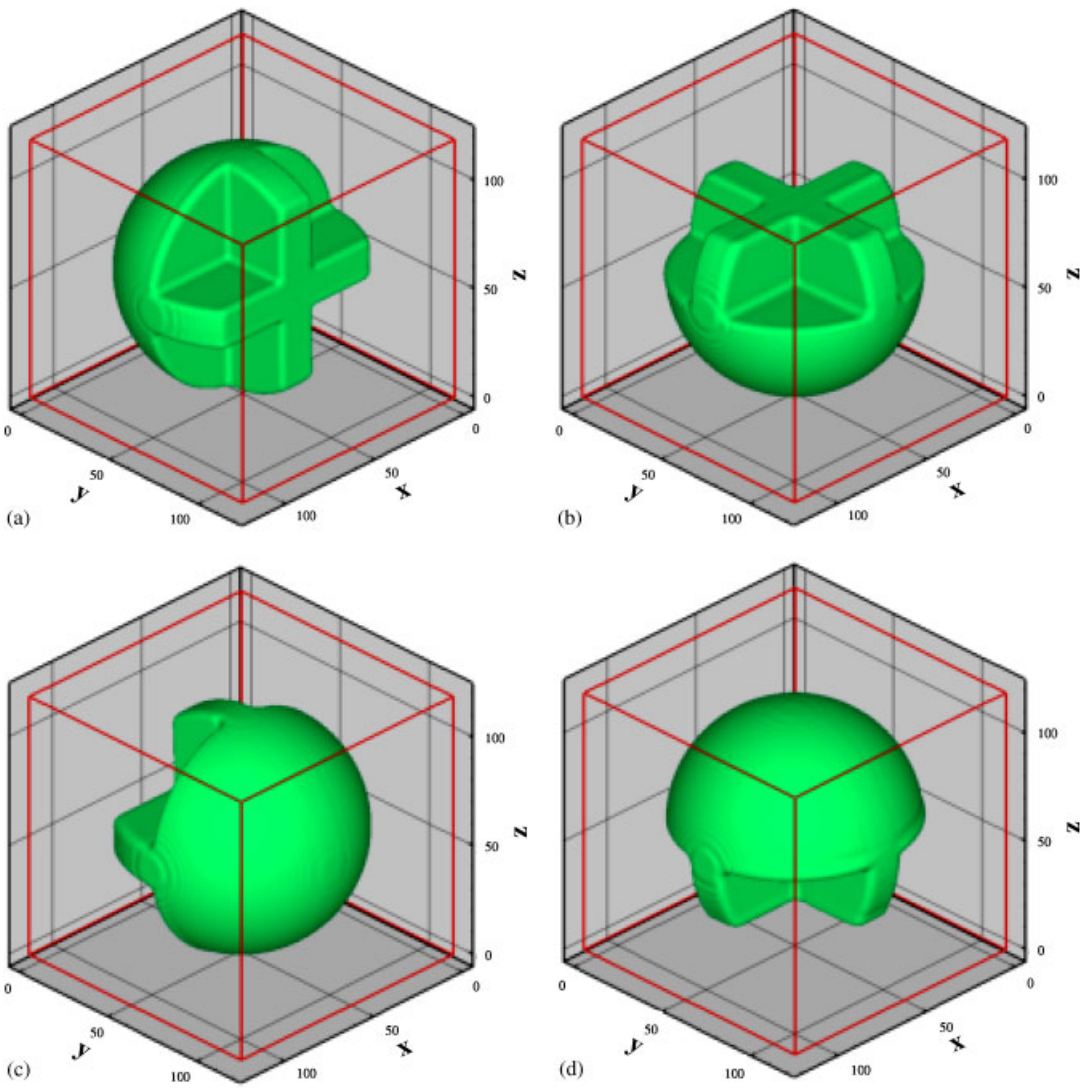

Figure 4. The results of present method for the rotation case.

unlike the rotational case, there is a preferred direction in the velocity field. The objective of this study is to check whether the shape will return to the original one under a certain topological deformation.

The computational domain is a box with $0 \leqslant i, j, k \leqslant 120$. Initially, a sphere is located at the center of the box as shown in Figure 5(a). The order parameter is set to be ϕ^* inside the sphere and $-\phi^*$ elsewhere, and

$$\phi = \phi^* \tanh d \tag{35}$$

DOI: 10.1002/fld

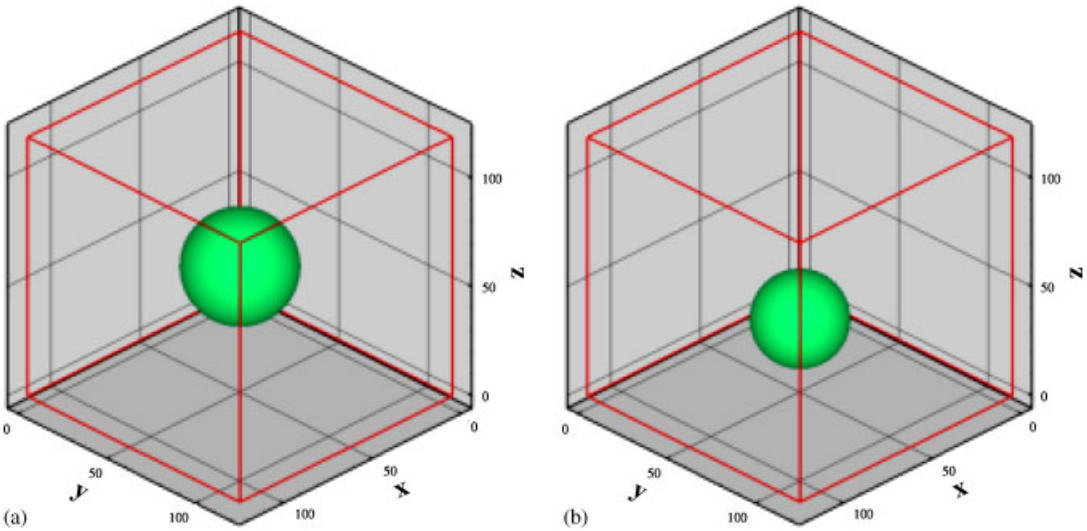

Figure 5. The initial conditions: (a) elongation case and (b) shear flow case.

where d is the same as that shown in Equation (30). The radius of the sphere is 24 in lattice unit. The natural boundary condition is employed at all boundaries. The parameters are chosen the same as those in Section 3.1.

When the physical velocity field (the unit is m/s) is taken as

$$u^{\text{phys}} = \pi(x^{\text{phys}} - 0.5), \quad v^{\text{phys}} = \pi(y^{\text{phys}} - 0.5), \quad w^{\text{phys}} = -2\pi(z^{\text{phys}} - 0.5) \tag{36}$$

and the physical domain (the unit is m) is chosen as

$$x^{\text{phys}}, y^{\text{phys}}, z^{\text{phys}} \in [0, 1]$$

we can get a similar relationship between the physical time t^{phys} and the lattice time t (NX = NY = NZ) as

$$t^{\text{phys}} = \frac{u_0}{\text{NX}} t \tag{37}$$

With $u_0 = 0.02$ and NX = 120, it is clear that 6000 time steps in the present computation correspond to 1 s in other methods. To evaluate our method, we also compare our results with those of DSFCT VOF method. After one cycle of contraction in the z direction and elongation in the other two directions, we adjust the elongation and contraction direction by changing the sign of the velocity field and continue to run the code for another cycle. We try to see whether the sphere would return to its initial state after a complete cycle. From Figures 6(a) and (b) and 7(a) and (b), we can observe that the shape of the body changes from the sphere to a flat disc in the contraction cycle in the z direction. In contrast, during the expansion process in the z direction, the flat disc would return to the sphere as shown in Figures 6(c) and (d) and 7(c) and (d). Besides, the process is reversible. This could be true as we can observe that the shapes in Figures 6(a) and 7(a) agree well with those in Figures 6(c) and 7(c). On the other hand, although the present method and the DSFCT VOF approach give the similar results, there are still some differences. From Figure 6, we

DOI: 10.1002/fld

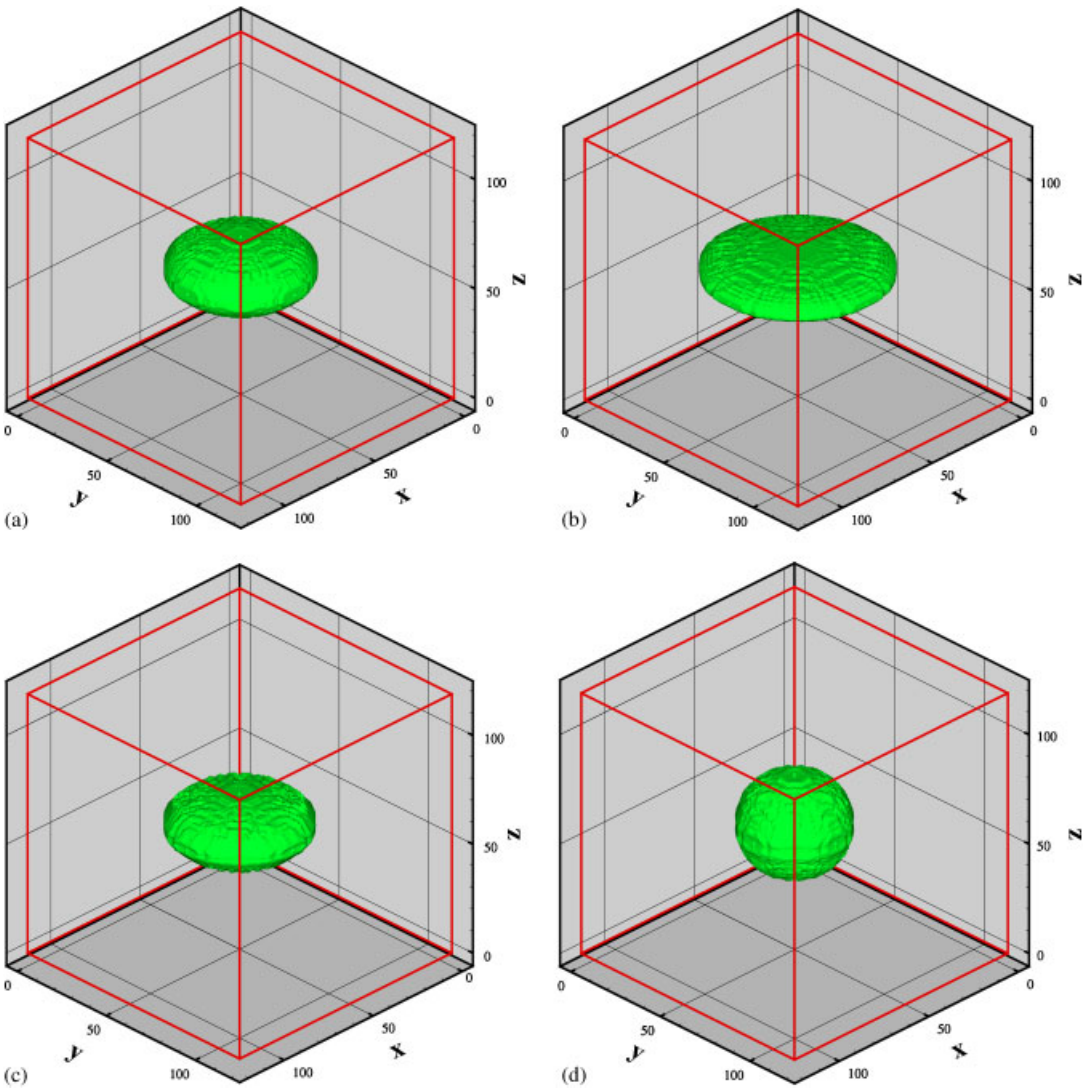

Figure 6. The results of the direction split FCT VOF method for the elongation case.

can easily observe that there are many noises shown in the results of DSFCT VOF method. That is, the shapes are not as smooth as those of the present method. This indicates that DSFCT VOF gives a poor interface tracking. In this sense, we can say that the present method performs better than the VOF method for this test case.

3.4. Shear flow

In this section, we consider a more realistic case, a sphere under the shear flow. As compared to the elongational velocity field, the velocity gradient in each direction is not a constant in this case.

DOI: 10.1002/fld

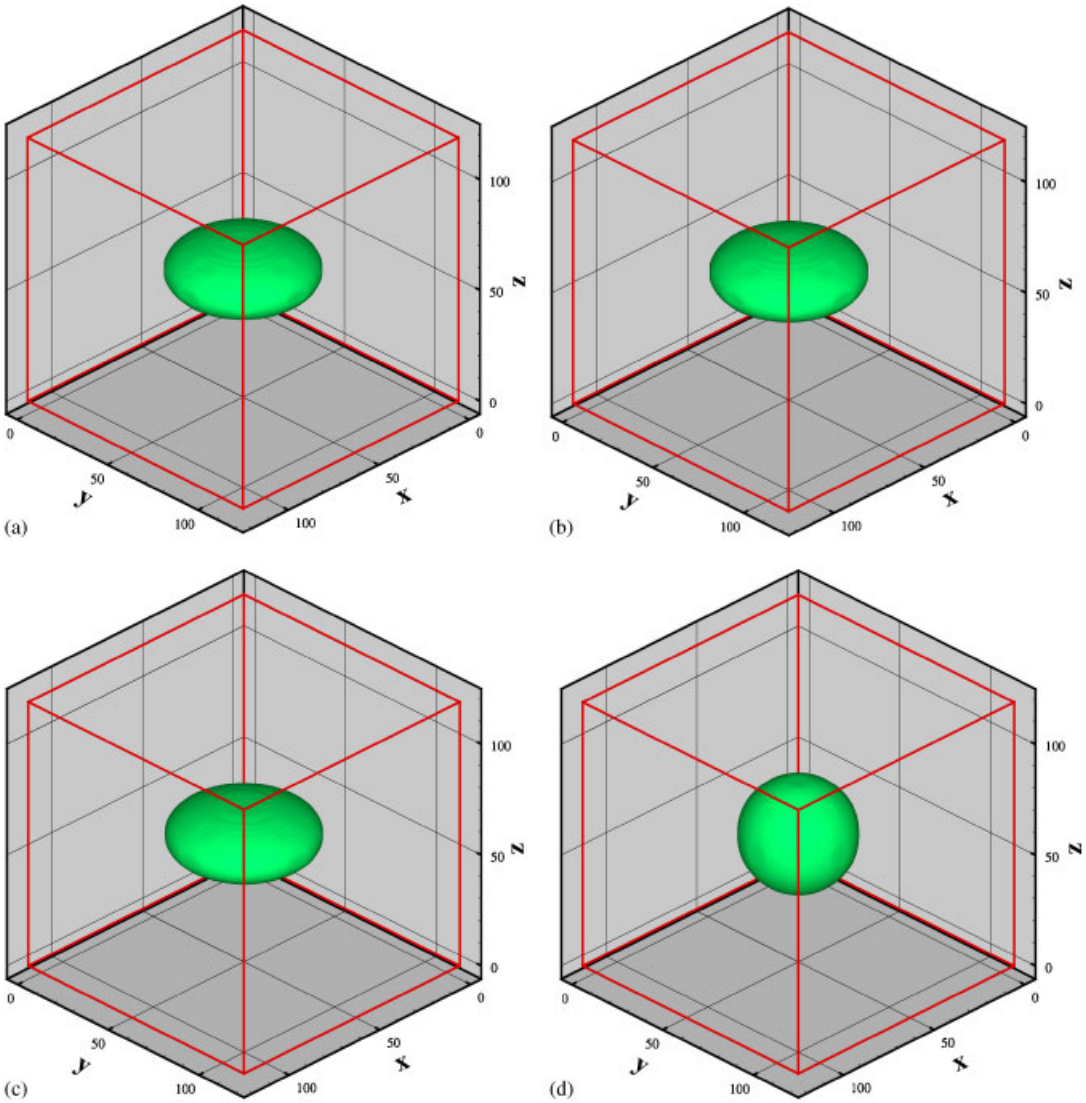

Figure 7. The results of present method for the elongation case.

Again, the computational domain is taken as a box with $0 \leqslant i, j, k \leqslant 120$. Initially, the sphere with the radius of 20 is located at the bottom part of the box. The center of the sphere is (60, 60, 36) as shown in Figure 5(b). Periodic boundary conditions are employed at all boundaries. The parameters are also chosen the same as those in Section 3.3, and the order parameter is set to the same as given by Equation (35).

DOI: 10.1002/fld

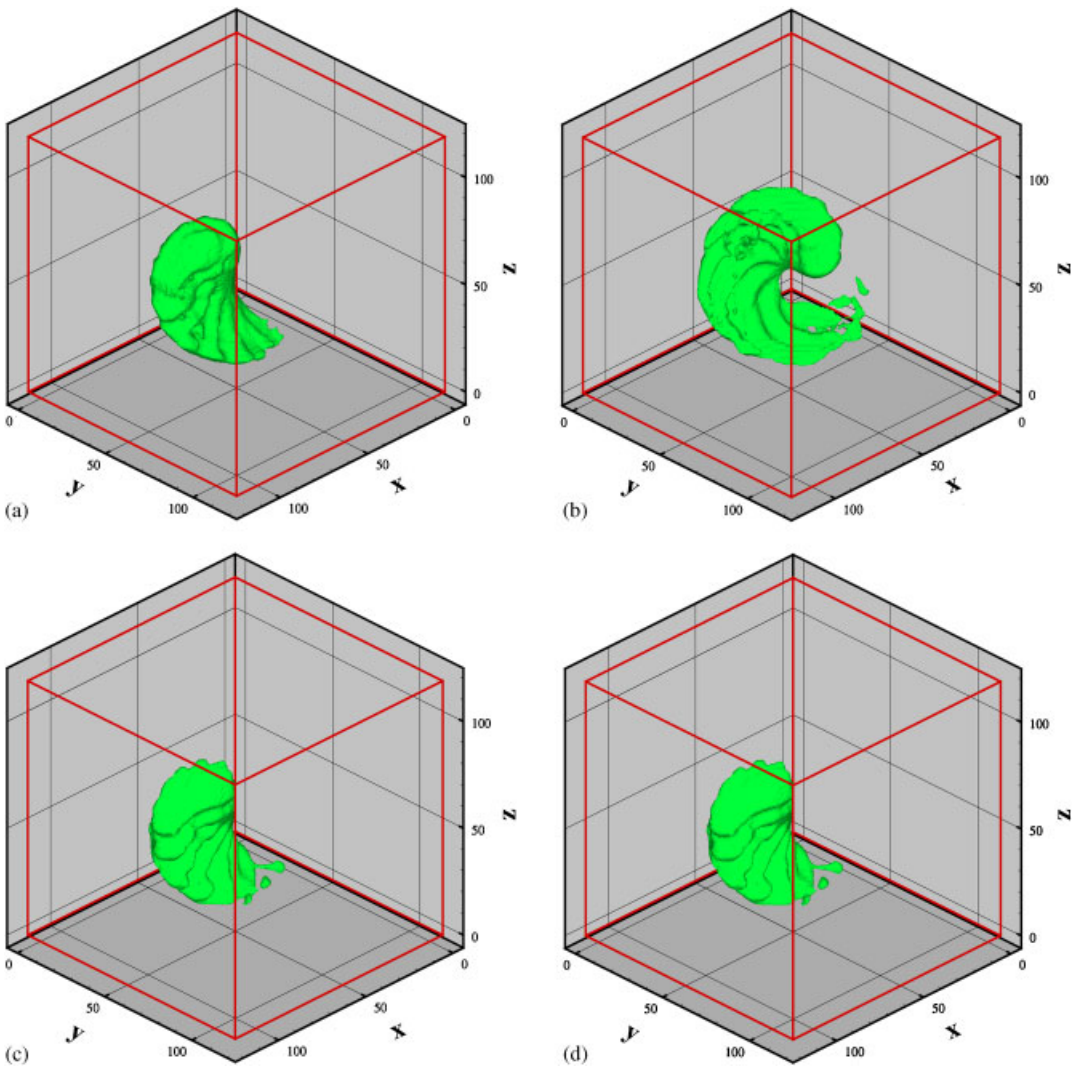

Figure 8. The results of the direction split FCT VOF method for the shear flow case.

During the process before one rotation cycle T, the shear velocity in the LBM computation is given as

$$u = u_0 \pi \cos\left[\pi\left(\frac{i}{\mathrm{NX}} - 0.5\right)\right]\left\{\sin\left[\pi\left(\frac{j}{\mathrm{NY}} - 0.5\right)\right] - \sin\left[\pi\left(\frac{k}{\mathrm{NZ}} - 0.5\right)\right]\right\}$$

$$v = -u_0 \pi \cos\left[\pi\left(\frac{j}{\mathrm{NY}} - 0.5\right)\right]\left\{\sin\left[\pi\left(\frac{i}{\mathrm{NX}} - 0.5\right)\right] - \sin\left[\pi\left(\frac{k}{\mathrm{NZ}} - 0.5\right)\right]\right\} \quad (38)$$

DOI: 10.1002/fld

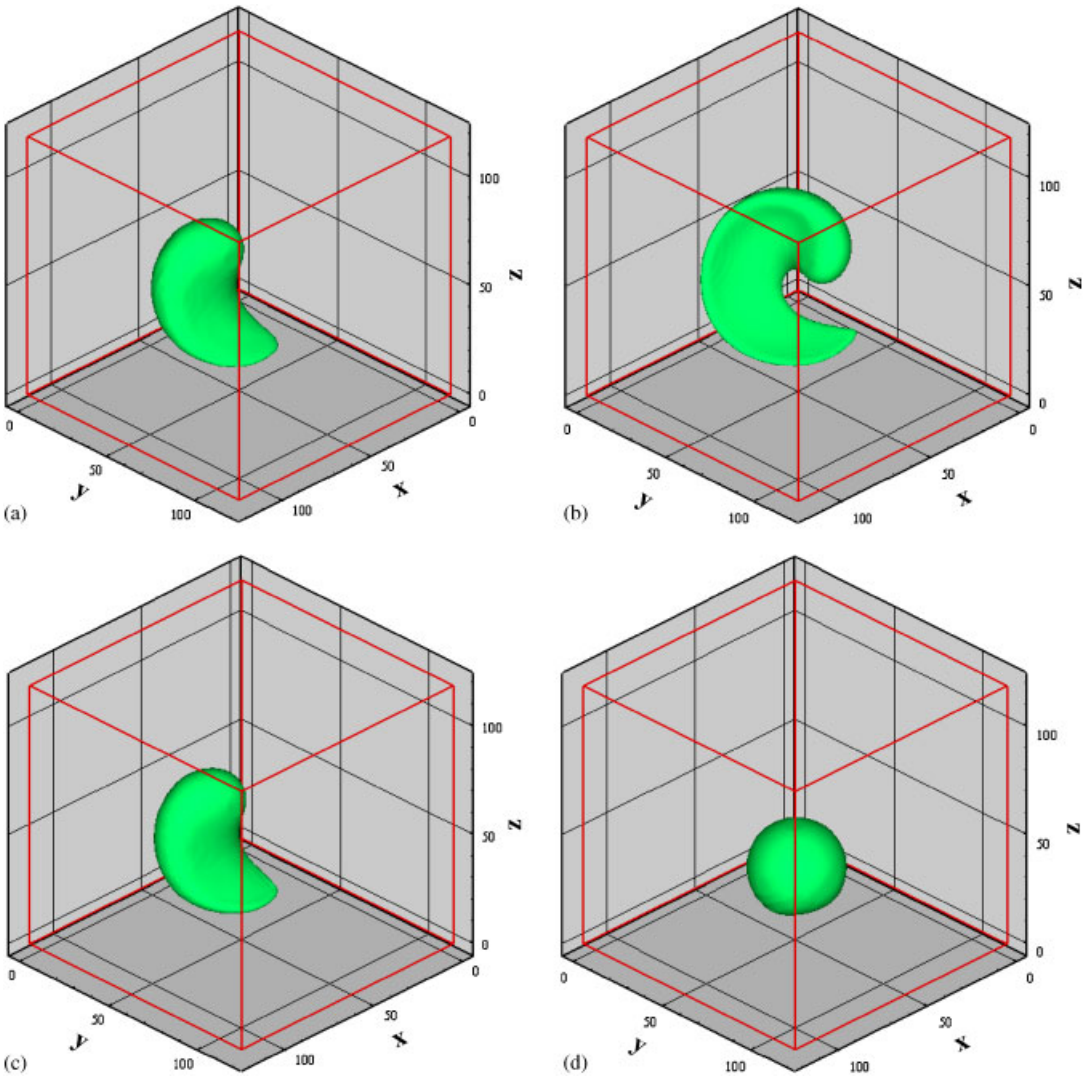

Figure 9. The results of present method for the shear flow case.

$$w = -u_0 \pi \cos\left[\pi\left(\frac{k}{\mathrm{NZ}} - 0.5\right)\right]\left\{\sin\left[\pi\left(\frac{j}{\mathrm{NY}} - 0.5\right)\right] - \sin\left[\pi\left(\frac{i}{\mathrm{NX}} - 0.5\right)\right]\right\}$$

When the physical velocity field (the unit is m/s) is taken as

$$\begin{aligned} u^{\mathrm{phys}} &= \pi \cos[\pi(x^{\mathrm{phys}} - 0.5)]\{\sin[\pi(y^{\mathrm{phys}} - 0.5)] - \sin[\pi(z^{\mathrm{phys}} - 0.5)]\} \\ v^{\mathrm{phys}} &= -\pi \cos[\pi(y^{\mathrm{phys}} - 0.5)]\{\sin[\pi(x^{\mathrm{phys}} - 0.5)] - \sin[\pi(z^{\mathrm{phys}} - 0.5)]\} \\ w^{\mathrm{phys}} &= -\pi \cos[\pi(z^{\mathrm{phys}} - 0.5)]\{\sin[\pi(y^{\mathrm{phys}} - 0.5)] - \sin[\pi(x^{\mathrm{phys}} - 0.5)]\} \end{aligned} \tag{39}$$

DOI: 10.1002/fld

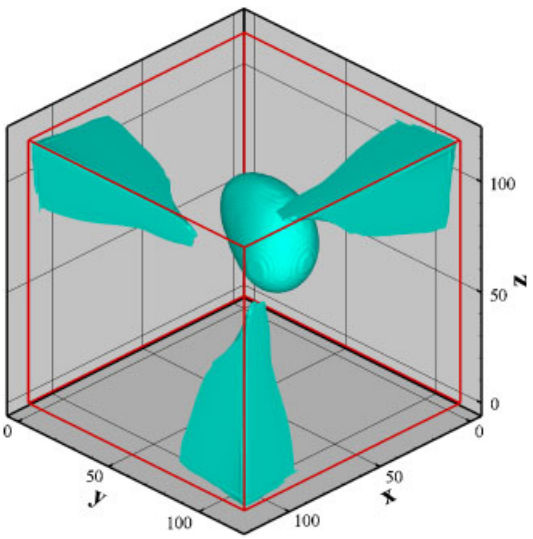

Figure 10. The results from the method of Inamuro *et al.* [9] for the shear flow case.

and the physical domain (the unit is m) is

$$x^{\text{phys}}, y^{\text{phys}}, z^{\text{phys}} \in [0, 1]$$

we can obtain a similar relationship between the physical time t^{phys} and the lattice time t (NX = NY = NZ) as

$$t^{\text{phys}} = \frac{u_0}{\text{NX}} t \tag{40}$$

In the present computation, $u_0 = 0.01$ and NX = NY = NZ = 120. Thus, 12 000 time steps correspond to 1 s in other methods. After one rotation cycle, we adjust the rotation direction by changing the sign of the velocity field and continue to run the code for another cycle T. We are interested to see whether the shape of sphere would return to its initial configuration and whether the process is reversible.

To validate the present method, the DSFCT VOF and the approach of Inamuro *et al.* [9] are also applied to solve this problem. The results of DSFCT VOF are displayed in Figure 8. From Figure 8, it is obvious that the DSFCT VOF method shows quantities of slots at the interface and jetsam near the interface. We can also easily observe that there are some breakups in Figure 8(c). Besides, the shape in Figure 8(c) does not agree with that in Figure 8(a). This is different from the case in Section 3.3, where the shape in the backward evolution process is similar to the forward one. The present results are shown in Figure 9. Clearly, the shape in Figure 9(c) agrees well with that in Figure 9(a). This implies that DSFCT VOF method may not be suitable to the cases with large deformation. Furthermore, as shown in Figure 8(d), there are many disturbances generated by DSFCT VOF method. The shape is greatly deformed after the whole process and does not recover the initial configuration. In contrast, the sphere returns to its original shape and position with a little distortion as shown in Figure 9(d) by the present method. This implies that the present method performs very well even for the case where large deformation occurs. The result from the method

DOI: 10.1002/fld

of Inamuro *et al*. [9] at 3000 time steps is shown in Figure 10. We can observe clearly from the figure that some unphysical disturbances appear near the boundary even though the computation only marches 3000 time steps. If we continue to do the simulation, the disturbances will expand until the sphere and the disturbances coalesce together eventually. By comparing the results in Figures 9 and 10, we can conclude that the present method performs much better than the approach of Inamuro *et al.* [9] for the deformation of interface under shear flow.

4. CONCLUSIONS

A new three-dimensional lattice Boltzmann interface capturing method is proposed in this paper. It does not require interface reconstruction as needed by most of the traditional methods. The computational time cost by the present method is about half of the time needed by other lattice Boltzmann interface capturing methods. We can conclude that the present method has a better efficiency than most of the existing lattice Boltzmann methods since it only uses seven lattice velocities while other methods use at least 15 lattice velocities. Numerical results show that the present method can capture accurate position of the interface. For example, it shows good performance under elongation and shear flow with stretching and tearing. For all the test cases considered, the present method has a better performance than the DSFCT VOF method and the approach of Inamuro *et al.* [9].

REFERENCES

1. Hirt CW, Nichols BD. Volume of fluid (VOF) methods for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
2. Rudman M. Volume tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
3. Youngs DL. Time-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*, Morton KW, Baines MJ (eds). Academic Press: New York, 1982; 273–285.
4. Ashgriz N, Poo JY. FLAIR: flux line-segment model for advection and interface reconstruction. *Journal of Computational Physics* 1991; **93**(2):449–468.
5. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**(1):12–49.
6. Kothe D, Juric D, Mosso SJ, Lally B. Numerical recipes for mold filling simulation. In *Modeling of Casting, Welding and Advanced Solidification Processes VIII*, Thomas BG, Beckermann C (eds). TMS: New York, 1998; 17–28.
7. Cahn JW, Hilliard JE. Free energy of a nonuniform system. I. Interfacial energy. *Journal of Chemical Physics* 1958; **28**(2):258–267.
8. Swift MR, Osborn WR, Yeomans JM. Lattice Boltzmann simulation of liquid–gas and binary fluid systems. *Physical Review E* 1996; **54**:5041–5052.
9. Inamuro T, Ogata T, Tajima S, Konishi N. A lattice Boltzmann method for incompressible two-phase flows with large density differences. *Journal of Computational Physics* 2004; **198**:628–644.
10. He X, Chen S, Zhang R. A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability. *Journal of Computational Physics* 1999; **152**(2):642–663.
11. Lee T, Lin C-L. A stable discretization of the lattice Boltzmann equation for simulation of incompressible two-phase flows at high density ratio. *Journal of Computational Physics* 2005; **206**:16–47.
12. Zheng HW, Shu C, Chew YT. Lattice Boltzmann interface capturing method for incompressible flow. *Physical Review E* 2005; **72**:056705.
13. Lamura A, Succi S. A lattice Boltzmann for disordered fluids. *International Journal of Modern Physics B* 2003; **17**:145.
14. Bhatnagar P, Gross EP, Krook MK. A model for collision processes in gases. *Physical Review* 1954; **94**:511–525.
15. Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics* 1979; **31**:335–362.

DOI: 10.1002/fld