

# An efficient lattice Boltzmann multiphase model for 3D flows with large density ratios at high Reynolds numbers



Amir Banari<sup>a,\*</sup>, Christian F. Janßen<sup>a,b</sup>, Stephan T. Grilli<sup>a</sup>

<sup>a</sup> Department of Ocean Engineering, University of Rhode Island, USA

<sup>b</sup> Institute for Fluid Dynamics and Ship Theory, Hamburg University of Technology, Germany

## ARTICLE INFO

### Article history:

Received 29 March 2014

Received in revised form 29 September 2014

Accepted 10 October 2014

Available online 1 November 2014

### Keywords:

Lattice Boltzmann method

Multiphase flows

High density ratio

Rising bubble

Drop impact

Breaking wave

## ABSTRACT

We report on the development, implementation and validation of a new Lattice Boltzmann method (LBM) for the numerical simulation of three-dimensional multiphase flows (here with only two components) with both high density ratio and high Reynolds number. This method is based in part on, but aims at achieving a higher computational efficiency than Inamuro et al.'s model (Inamuro et al., 2004). Here, we use a LBM to solve both a pressureless Navier–Stokes equation, in which the implementation of viscous terms is improved, and a pressure Poisson equation (using different distribution functions and a D3Q19 lattice scheme); additionally, we propose a new diffusive interface capturing method, based on the Cahn–Hilliard equation, which is also solved with a LBM. To achieve maximum efficiency, the entire model is implemented and solved on a heavily parallel GPGPU co-processor. The proposed algorithm is applied to several test cases, such as a splashing droplet, a rising bubble, and a braking ocean wave. In all cases, numerical results are found to agree very well with reference data, and/or to converge with the discretization.

© 2014 Elsevier Ltd. All rights reserved.

## 1. Introduction

In recent years, the lattice Boltzmann method (LBM) has become an increasingly attractive, fast, and accurate, alternative modeling method to standard continuum mechanics numerical models, for solving a variety of complex single and multiple-fluid flow problems [1]. Besides its versatility, this is in part due to the LBM's ability to be efficiently parallelized for implementation on General Purpose Graphical Processor Units (GPGPUs). Specifically, it has been shown in various publications [2–4] that LBM methods are especially well-suited for a GPGPU implementation, due to the locality of collision and propagation operators and the explicit nature of the method.

The LBM is based on the Boltzmann equation, which governs the dynamics of molecular probability distribution functions from a microscopic point of view. In the standard LBM implementation, the Boltzmann equation is discretized on an Eulerian mesh, a.k.a. the *lattice*, yielding a numerical method for computing macroscopic distribution functions on the lattice, in which the macroscopic hydrodynamic quantities, such as pressure and velocity, are obtained as low-order moments of these distribution functions [5,1]. To the limit of small time step and grid spacing, the LBM solution can be shown to converge towards the solution of the governing macroscopic equations of continuum mechanics [6]. Hence, with the proper

\* Corresponding author.

E-mail addresses: [amir\\_banari@my.uri.edu](mailto:amir_banari@my.uri.edu) (A. Banari), [mail@christian-janssen.de](mailto:mail@christian-janssen.de) (C.F. Janßen), [grilli@oce.uri.edu](mailto:grilli@oce.uri.edu) (S.T. Grilli).

selection of the LBM collision operator and distribution functions, the LBM solution can be made to converge to that of the Navier–Stokes equations (NS), including in the presence of a free surface (e.g., Janssen et al. [7]). The explicit nature of the method and the linear formulation of advection terms in the LBM collision–propagation equation provide the numerical scheme with several advantages, such as: (i) a relatively easy implementation (as far as uniform grids are concerned); and (ii) the locality of numerical operators, which allows for a more efficient parallel implementation, particularly on GPGPUs, than for more traditional finite volume or finite element algorithms. These characteristics have made the LBM a widely used tool for solving various complex fluid mechanics problems, such as multiphase flows, micro- and nanoscale flows, flows in porous media, and other fluid flow types [8,9,1].

Although many studies of multiphase flows using the LBM have been reported [10,11], most of these had two significant limitations: (i) the maximum density ratio between fluids is typically limited to 5–10, due to the triggering of local instabilities near the fluid-phase interface for larger ratios; (ii) most of the schemes cannot simulate high Reynolds number flows, due to instabilities resulting from the low relaxation times required for high Reynolds numbers (Re). The effects of either one of these problems are sufficient to make simulations unstable, even in the absence of the other problem.

The practical applications that motivated this research deal with air–sea interactions at the ocean surface in high wind conditions, hence with turbulent (i.e., very large Re values around  $10^8$ ) two-fluid flows with a high density ratio (order 1000). Hence, our main goal has been to develop an efficient LBM model that overcame these two limitations.

Several LBM studies of multiphase flows with a high density ratio have been proposed; Zheng et al. [12] proposed an LBM scheme for high density ratio, but in their work they used an artificial density ratio defined as the mean of densities of two fluid system. The deficiencies and limitations of their work have been explained in [13,14]. Two promising concepts were proposed by Lee et al. [8] and Inamuro et al. [9]. Lee et al. [8] used an approach similar to that of He et al. [15], in which they transformed the classical single phase discrete Boltzmann equation, from a mass–momentum to a pressure–momentum formulation. This decreased potential instabilities that could occur due to large fluid density gradients near the phase interface. Also, they split up the intermolecular forces for a non-ideal gas into hydrodynamic pressure, thermodynamic pressure, and surface tension force contributions. They reported that “parasitic currents” at the phase interface affected the numerical results due to the imbalance between thermodynamic pressure and surface tension forces, resulting from truncation errors related to curvature computations. They nearly eliminated this problem by using a thermodynamic identity to recast the intermolecular forcing term from a stress to a potential formulation. Furthermore, to stabilize their numerical scheme for large density ratios, they used different discretization patterns (i.e., central, biased and mixed differences) at different stages of the simulations. With this scheme, they were able to simulate two-phase flows with density ratio up to 1000. However, they could not achieve high Reynolds numbers, because stability issues related to low relaxation times were not addressed, and in their scheme relaxation time was still a function of the Reynolds number. To eliminate the numerical instabilities resulting from high density ratios, Inamuro et al. [9] removed the density from the advection part of the LBM equilibrium distribution functions. This in effect eliminated the pressure gradient from the corresponding macroscopic momentum equation, which thus became a “pressureless” NS equations. To retrieve the complete momentum equations and satisfy mass conservation, they subsequently corrected the velocity field by solving a Poisson equation for the pressure field. In their method, unlike in classical LBMs, the fluid viscosity is no longer related to the relaxation time and hence results stay more stable at high Reynolds numbers. Finally, in Inamuro et al.’s method, viscous effects are modeled by specifying the viscous stress tensor as a body force in the LBM collision operator. Proceeding this way, however, yields additional non-physical terms in the corresponding momentum equation, which decreases the model accuracy. In earlier work [5], we modified Inamuro et al.’s method to solve two-dimensional (2D) two-phase flows with high density ratio, by removing the non-physical terms from the momentum equation and formulating the phase interface tracking equations in a more rigorous way, based on the Cahn–Hilliard equations [16]. Additionally, we efficiently implemented our model for a massively parallel solution on a GPGPU. In doing so, we solved all the governing equations for each fluid, the interface, and the Poisson equation (required for correcting the velocity field) with a LBM scheme, thus achieving an even higher computational efficiency on the GPGPU. Our method, however, only worked for low Reynolds number flows.

In this paper, in light of this earlier work, we develop a new three-dimensional (3D) LBM model, also based on Inamuro et al.’s [9] approach. As before, we introduce new equilibrium distribution functions to both retrieve NS equations and improve the formulation of surface tension and viscous forces. For the interface capturing part, as in [5], we solve the Cahn–Hilliard equation using a LBM scheme with improved equilibrium distribution functions. In this new 3D model, however, we formulate the latter functions to be able to achieve high Reynolds numbers in the applications without suffering from instability problems. The resulting numerical scheme is computationally demanding, as the Poisson equation must be (iteratively) solved for each time step of the solution, in order to obtain the velocity field correction terms. As before, to achieve high computational efficiency, our LBM code is parallelized and implemented on a GPGPU using the nVIDIA CUDA framework. This approach provides access to the latest generation of GPGPU boards, such as the nVIDIA Tesla C2070 that was used in the present work (448 computing cores; 6 GB of memory; and a double precision computing capability).

The paper is organized as follows. We first develop the LBM equations used to solve for multiphase flows with high density ratios and detail their numerical implementation. The method is then validated for a series of applications, by comparing the present numerical results to reference solutions, for the splashing of droplets on a thin fluid layer, for a rising bubble, and for breaking ocean waves. Finally, we draw some conclusions and provide perspectives for future work.

## 2. Governing equations and numerical implementation

### 2.1. Lattice Boltzmann methods

The numerical simulation of multiphase flow problems typically involves solving the macroscopic governing Navier–Stokes (NS) equations within each fluid region, which directly provides hydrodynamic parameters such as velocity and pressure. Here, as discussed in the introduction, we instead use the lattice Boltzmann method (LBM) to solve a mesoscopic problem, whose solution converges to that of NS equations. Additionally, our proposed LBM approach is able to solve two-fluid flows whose density ratio is nearly 1000, typically air and water.

For multiphase flows, a separate model must also be implemented for tracking the motion of interfaces between fluids, which can be represented as sharp or diffused. Here, the latter is assumed and interfaces are tracked by the Cahn–Hilliard advection–diffusion (AD) equation. In the latter, a phase order function is used to smoothly vary fluid properties, such as density and viscosity, across the interface (denoted  $\phi$  and varying between arbitrary values of  $\phi_1$  and  $\phi_2$  set for fluid 1 and 2, respectively, with here  $\phi_1 > \phi_2$ ; see details in Section 2.3).

As in the two-dimensional (2D) model of Banari et al. [5], in our three-dimensional (3D) model, both the NS and AD equations will be solved by two distinct LBM schemes, with separate sets of distributions functions and collision operators. In both cases, the D3Q 19 lattice discretization scheme will be used, which introduces 19 discrete particle velocities in directions  $\mathbf{e}_i$  defined as,

$$\mathbf{e}_i = c \cdot \left\{ \begin{array}{c|cccccccc|cccccccccccc} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{array} \right\} \quad (2.1)$$

where  $c = \Delta x / \Delta t$  denotes the particle propagation speed on the lattice [17], and  $\Delta x$  and  $\Delta t$  are the lattice constant mesh size and time step, respectively.

Details of the LBM equations and schemes are given in the following sections.

### 2.2. Lattice Boltzmann solution of NS equations

The macroscopic continuity and momentum (i.e., NS) equations for incompressible Newtonian fluids read (using the index summation convention),

$$\partial_\alpha u_\alpha = 0 \quad (2.2)$$

$$\partial_t u_\alpha + \partial_\beta (u_\alpha u_\beta) = -\frac{1}{\rho} \partial_\beta (p) + \partial_\beta \left\{ \mu (\partial_\alpha u_\beta + \partial_\beta u_\alpha) \right\} + \frac{k}{\rho} \partial_\beta \left\{ \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \delta_{\alpha\beta} - \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\beta} \right\} \quad (2.3)$$

where the last term in Eq. (2.3) is non-standard and represents surface tension effects at the diffuse interface between both fluids [5] ( $k$  is related to the  $\phi_1$  and  $\phi_2$  and the surface tension coefficient  $\sigma_{12}$ , for fluids 1 and 2, and will be defined in Section 2.3). These equations are solved in a LBM framework using the particle distribution functions  $g_i$ , whose time evolution is governed by,

$$g_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = g_i(\mathbf{x}, t) - \frac{\Delta t}{\tau_g} (g_i(\mathbf{x}, t) - g_i^{(eq)}(\mathbf{x}, t)) + \Delta t (F_i + G_i) \quad (2.4)$$

where  $F_i$  and  $G_i$  represent effects of the viscous and gravitational volume forces, respectively,  $\tau_g$  is a relaxation time, and  $g_i^{eq}$  are the equilibrium distribution functions defined as,

$$\begin{aligned} g_i^{(eq)} = w_i & \left\{ \frac{e_{i\alpha} u_\alpha^*}{c_s^2} + \frac{(e_{i\alpha} u_\alpha^*)^2}{2c_s^4} - \frac{|\mathbf{u}^*|^2}{2c_s^2} + \frac{1}{2c_s^2} \left( \tau_g - \frac{1}{2} \Delta t \right) \left( \frac{\partial u_\beta}{\partial x_\alpha} + \frac{\partial u_\alpha}{\partial x_\beta} \right) e_{i\alpha} e_{i\beta} \right\} \\ & + w_i \frac{k}{\rho} T_{\alpha\beta} e_{i\alpha} e_{i\beta} - v_i \frac{k}{2\rho} |\nabla \phi|^2. \end{aligned} \quad (2.5)$$

where  $c_s = c / \sqrt{3}$ , is the speed of sound and,

$$w_0 = \frac{1}{3}; \quad w_{1,\dots,6} = \frac{1}{18}; \quad w_{7,\dots,18} = \frac{1}{36}, \quad (2.6)$$

$$v_0 = -\frac{2}{c^2}; \quad v_i = \frac{3}{c^2} w_i \quad (i = 1, 2, \dots, 18), \quad (2.7)$$

are LBM weights used for the D3Q 19 scheme, and,

$$T_{\alpha\beta}(\phi) = \frac{9}{2c^4} \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\beta} - \frac{3}{2c^4} \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \delta_{\alpha\beta}. \quad (2.8)$$

expresses surface tension effects at the two-fluid interface, together with the last term in Eq. (2.5). The last term in the curly bracket in Eq. (2.5) removes the dependency of  $\tau_g$  on viscosity, which makes the proposed method extremely stable for the simulation of high Reynolds number flows. For this reason, viscous effects are added back as a volume force in the evolution Eq. (2.4), together with the effect of the gravity force. The volume forces expressing the contributions of viscous effects and gravity force in the LBM governing equation read,

$$F_i = \left(1 - \frac{\Delta t}{2\tau_g}\right) w_i \left[ \frac{\mathbf{e}_i - \mathbf{u}}{c_s^2} + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^4} \mathbf{e}_i \right] \cdot \mathbf{F} \quad (2.9)$$

$$G_i = \left(1 - \frac{\Delta t}{2\tau_g}\right) w_i \left[ \frac{\mathbf{e}_i - \mathbf{u}}{c_s^2} + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^4} \mathbf{e}_i \right] \cdot \mathbf{G} \quad (2.10)$$

with  $\mathbf{G}$  the gravitational acceleration vector, and  $\mathbf{F} = \frac{1}{\rho} \partial_\beta \left( \mu \left( \frac{\partial u_\beta}{\partial x_\alpha} + \frac{\partial u_\alpha}{\partial x_\beta} \right) \right)$  the viscous stress per unit mass. Note that the extended body force formulation of Guo et al. [18] has been used in these equations, since body forces vary in time and space.

The presence of density in the standard equilibrium distribution functions used for solving NS equations is the main source of instability identified in various earlier LBM solutions of multiphase flows with high density ratios [10,19]. Here, to eliminate these instabilities, following Banari et al. [5], density was removed from the first part of the equilibrium functions in Eq. (2.5). This will require adding a correction to the velocity field to satisfy mass conservation, as discussed next. It can be shown by applying a Chapman–Enskog expansion that Eq. (2.4) with the equilibrium distribution functions defined in Eq. (2.5) converges to the NS equations (Eqs. (2.3), (2.2)), without the pressure gradient term (referred to as “pressureless” NS equations). The details of the Chapman–Enskog expansion are shown in Appendix A. The fluid velocity corresponding to these pressureless NS equations is found from Eq. (2.5) as,

$$\mathbf{u}_\alpha^* = \sum_{i=0}^b g_i \mathbf{e}_{i\alpha} + \frac{\Delta t}{2} (\mathbf{F} + \mathbf{G}). \quad (2.11)$$

However, this velocity is not divergence free and hence needs to be corrected. Thus, similar to [9,5], the corrected velocity  $\mathbf{u}$  is obtained as,

$$\mathbf{u} = \mathbf{u}^* + \Delta \mathbf{u} \quad (2.12)$$

with the correction  $\Delta \mathbf{u}$  defined as,

$$\Delta \mathbf{u} \simeq -\Delta t \frac{\nabla p}{\rho} \quad (2.13)$$

as a function of the gradient of the pressure  $p$ , which is obtained by solving the following Poisson equation,

$$\nabla \cdot \left( \frac{\Delta t \nabla p}{\rho} \right) = \nabla \cdot \mathbf{u}^*. \quad (2.14)$$

Similar to [5], this equation is solved iteratively in a LBM framework. For this purpose, an additional set of particle distribution functions  $h_i$  ( $i = 0, \dots, 18$ ) is introduced, whose time evolution is governed by,

$$h_i^n(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = h_i^n(\mathbf{x}, t) - \frac{\Delta t}{\tau_h} (h_i^n - h_i^{(\text{eq},n)}) - \frac{1}{3} w_i (\nabla \cdot \mathbf{u}^*(t)) \quad (2.15)$$

where  $n$  denotes the  $n$ -th iteration in the Poisson equation solution (see [9] for details). In this second LBM scheme, the equilibrium distribution functions are defined as,

$$h_i^{(\text{eq},n)} = \frac{w_i p^n(\mathbf{x}, t)}{\rho_0 c^2} \quad (2.16)$$

with  $\rho_0$  denoting a reference density and the relaxation time  $\tau_h$  being related to the density by,

$$\tau_h = \Delta t \left( \frac{\rho_0 c^2}{\rho c_s^2} + \frac{1}{2} \right). \quad (2.17)$$

Once the Eqs. (2.15)–(2.17) are solved for a given iteration  $n$ , the pressure is obtained as the zeroth order moment of the particle distribution functions as,

$$p^{n+1} = \rho_0 c^2 \sum_{i=0}^b h_i^n. \quad (2.18)$$

This scheme is solved iteratively until convergence is achieved on the pressure for a given time step. Then, the velocity correction is calculated using Eq. (2.13).

As shown in the Chapman–Enskog expansion (Appendix A), the complete set of LBM equations (2.4)–(2.18) solves the complete set of NS equations (2.3). Unlike the classical LBM, however, the relaxation time  $\tau_g$  can be chosen arbitrarily, independently from the Reynolds number, and the scheme is stable for high density ratios between fluids. It should be pointed out that, incorporating viscous forces as volume forces creates additional spurious terms in the macroscopic NS equation. These terms, however, scale with viscosity and hence are negligible for high Reynolds number flows [5].

### 2.3. Lattice Boltzmann scheme for interface capturing

In this 3D model, similar to the 2D model of Banari et al. [5], the position of the interface between the 2 fluids is tracked using the Cahn–Hilliard advection–diffusion equation,

$$\partial_t(\phi) + \partial_\alpha(\phi u_\alpha) = M \nabla^2 \mu_\phi \quad (2.19)$$

where  $M$  is the mobility parameter between two components, and  $\mu_\phi$  is the chemical potential (see details below). Diffuse interface schemes offer some advantages as compared to sharp interface schemes, for which fluid parameters such as density and viscosity vary discontinuously across the interface. Such discontinuities often cause numerical noise, which can potentially trigger instabilities, especially for high density and viscosity ratios. By contrast, in diffuse interface schemes, fluid properties continuously and smoothly vary over a short distance across the interface. Additionally, for sharp interface schemes, the accurate computation of the interface curvature and related surface tension forces is very challenging, whereas surface tension effects are intrinsically included in diffuse interface models. Here, similar to [5], the motion of the diffusive interface is modeled by the Cahn–Hilliard equation,

In the model, the chemical potential is defined as,

$$\mu_\phi = \frac{d\Psi}{d\phi} - k \nabla^2 \phi \quad \text{with } \Psi(\phi) = (\phi - \phi_2)^2(\phi - \phi_1)^2 \quad (2.20)$$

the bulk free-energy density. Coefficients  $k$  and  $\beta$  are related to the surface tension coefficient  $\sigma_{12}$  and interface thickness  $W$  by,

$$W = \frac{4}{\phi_1 - \phi_2} \sqrt{\frac{k}{2\beta}} \quad \text{and} \quad \sigma_{12} = \frac{(\phi_1 - \phi_2)^3}{6} \sqrt{2k\beta}. \quad (2.21)$$

A more detailed derivation and discussion of this model can be found in [5].

Eqs. (2.19)–(2.21), are solved with a LBM using a third set of probability distribution functions,  $f_i(\mathbf{x}, t)$ , whose time evolution is governed by,

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{\Delta t}{\tau_f} (f_i(\mathbf{x}, t) - f_i^{(\text{eq})}(\mathbf{x}, t)). \quad (2.22)$$

The corresponding equilibrium distribution functions  $f_i^{(\text{eq})}$  read,

$$f_i^{(\text{eq})} = H_i \phi + v_i \frac{M}{\tau_f - \frac{1}{2} \Delta t} \mu_\phi + \phi w_i \left\{ \frac{e_{i\alpha} u_\alpha}{c_s^2} + \frac{(e_{i\alpha} u_\alpha)^2}{2c_s^4} - \frac{|\mathbf{u}|^2}{2c_s^2} \right\}, \quad (2.23)$$

with,

$$H_0 = 1 \quad \text{and} \quad H_{1,2,\dots,8} = 0 \quad (2.24)$$

and  $w_i$  and  $v_i$  defined as in Eqs. (2.6) and (2.7). The order parameter  $\phi$  is computed from the first moment of the distribution functions as,

$$\phi = \sum_{i=0}^b f_i. \quad (2.25)$$

Once the order parameter computed, the spatial distribution of fluid properties (density  $\rho$  and dynamic viscosity  $\mu$ ) in the entire domain, and in particular across the interface can be expressed as,

$$\rho(\phi) = \begin{cases} \rho_2 & \phi \leq \phi_2 \\ \frac{\phi - \phi_2}{\phi_1 - \phi_2} (\rho_1 - \rho_2) + \rho_2 & \phi_2 < \phi < \phi_1 \\ \rho_1 & \phi \geq \phi_1. \end{cases} \quad (2.26)$$

Rather than directly expressing the kinematic viscosity as a function of  $\phi$ , it is defined as a function of density as,

$$\mu(\rho) = \frac{\rho - \rho_2}{\rho_1 - \rho_2} (\mu_1 - \mu_2) + \mu_2. \quad (2.27)$$

This was fund by Banari et al. [5] to yield more accurate results.

## 2.4. Numerical implementation and performance

The numerical algorithm for this multi-component LBM model is given in Algorithm 1. The entire model is implemented for a parallel solution on a GPGPU, using the CUDA language in the NVIDIA environment. In the algorithm, after initializing all the fluid properties and LBM variables, these are adjusted through initial iterations performed on the interface shape or the field variables to fulfill the governing equations. Specifically, the initially sharp interface between the two fluids specified at time step zero is smoothed out by solving the advection–diffusion equation, for the interface capturing, for a fixed number of initial iterations (e.g., 5000), while assuming a zero velocity field and adjusting interface thickness based on Eq. (2.21). Once physically relevant initial conditions are achieved, the main computational time loop is started. Note that, in the latter, to increase the efficiency of the computations, the convergence of the iterative solution of the Poisson equation is only checked every 20 iterations.

---

### Algorithm 1 Algorithm for the LBM computation of two-fluid flow fields and phase field interface update

---

Set initial conditions,  $\phi = \phi_i$ ,  $\rho = \rho_i$ ,  $\mu = \mu_i$  and  $p = 0$

Initialize particle distribution functions, i.e. set

$$f_i(\mathbf{x}, t_0) = f_i^{eq}(\mathbf{x}, t_0), g_i(\mathbf{x}, t_0) = g_i^{eq}(\mathbf{x}, t_0), h_i(\mathbf{x}, t_0) = h_i^{eq}(\mathbf{x}, t_0)$$

Run initial iterations to improve the initial conditions for the phase field

**for**  $t < t_{end}$  **do**

    Compute  $f_i(\mathbf{x}, t + \Delta t)$  using Eq. (2.22)

    Compute  $g_i(\mathbf{x}, t + \Delta t)$  using Eq. (2.4),

    Compute  $\phi(\mathbf{x}, t + \Delta t)$  and  $\mathbf{u}^*(\mathbf{x}, t + \Delta t)$  with Eqs. (2.25) and (2.11);

$\rho(\mathbf{x}, t + \Delta t)$  and  $\mu(\mathbf{x}, t + \Delta t)$  are calculated using Eqs. (2.26) and (2.27).

    For every 20 Poisson iterations:

**if**  $\left| \frac{p^{n+1} - p^n}{p^n} \right| > \varepsilon$  **then**

            Compute  $p^{n+1}(\mathbf{x}, t + \Delta t)$  using Eq. (2.15)–(2.18).

**end if**

        Compute  $\mathbf{u}(\mathbf{x}, t + \Delta t)$  using Eq. (2.13).

**end for**

---

The performance of our new 3D LBM multiphase model highly depends on the number of Poisson iterations performed at each time step, which depends upon the problem physics that is tackled. For a given relative error threshold  $\varepsilon$ , we find that the number of iterations required to solve the Poisson equation is initially larger, in the first few time steps of calculations, while the field variables are still dynamically adjusting to the governing equations, but then decreases later on. After this early phase of computations we can evaluate the model efficiency in terms of the usual metric used in LBM models, the “Million of Node Updates Per Second” (MNUPS). On a NVIDIA TESLA 2070 GPGPU (448 CUDA cores), the performance of the current scheme is found to be around 20 MNUPS, using the largest possible grid that can fit in the 6 Gb RAM, with about 40% of the computational time spent for solving the Poisson equation. While this is less than for standard LBM schemes applied to a single fluid, which may achieve over 100 MNUPS, this is still a very respectable performance considering the higher complexity of solving large Reynolds number flows, for two fluids with a high density ratio, while tracking their 3D interface, which may have a very complex geometry in some applications (e.g., ocean breaking waves).

## 3. Validation

A few increasingly complex validation/benchmarking applications of our LBM scheme are presented in the following, to assess the accuracy and convergence of the presented multiphase model and confirm its relevance from investigating new physics in multi-fluid flows, particularly in the context of air–sea interactions, which is the motivation for this research. All simulations are carried out in a dimensionless framework, using non-dimensional lattice variables (here denoted by a prime) scaled on the basis of a length scale  $\lambda$ , time scale  $\tau$  and mass scale  $\varpi$ . Thus, for the mesh parameters,  $\Delta x' = \Delta x/\lambda$ ,  $\Delta t' = \Delta t/\tau$  and  $c' = c\tau/\lambda$ . It is also customary in LBM to assume that  $c' = 1$ , which is akin to having the mesh Courant number be unity. If the length scale is further defined as  $\lambda = \Delta x$ , we then have  $\Delta x' = 1$  and this requires  $\tau = \Delta t$  and  $\Delta t' = 1$  as well. Hence, with these definitions, in lattice variables, we always have  $c' = \Delta x' = \Delta t' = 1$  and  $c'_s = c'/\sqrt{3} = 1/\sqrt{3}$  [17]. Accordingly, the dimensionless relaxation times are defined as,

$$\tau'_g = \frac{\tau_g}{\Delta t} \quad \text{and} \quad \tau'_h = \frac{1}{2} + \frac{\rho'_0}{\rho'} \quad (3.1)$$



for distribution functions  $g_i$  and  $h_i$ , respectively, with  $\rho' = \rho\lambda^3/\varpi$  and  $\rho'_0 = \rho_0\lambda^3/\varpi$ . The non-dimensional reference density is set to  $\rho'_0 = 1$ , which implies that,  $\varpi = \rho_0(\Delta x)^3$  (usually, one will also assume for simplicity,  $\rho_0 = 1 \text{ kg/m}^3$ ) and  $\rho'_i = \rho_i$ , for fluid  $i = 1, 2$ . The non-dimensional surface tension coefficient is further defined as,

$$\sigma'_{12} = \frac{\sigma_{12}\tau^2}{\rho_0\lambda^3} = \frac{\sigma_{12}}{\rho_0 c^2 \Delta x}. \quad (3.2)$$

For the phase order function, we specify  $\phi_1 = 1.0$  and  $\phi_2 = 0.0$  in all cases, and the interface thickness is assumed to span 4 lattice meshes, i.e.,  $W = 4\Delta x$  or  $W' = 4$ , which with Eq. (2.21) yields the interface lattice parameters,

$$\beta' = \frac{3\sigma'_{12}}{(\phi_1 - \phi_2)^4} \quad \text{and} \quad k' = \frac{6\sigma'_{12}}{(\phi_1 - \phi_2)^2}, \quad (3.3)$$

where  $\beta' = \beta/(\rho_0 c^2)$  and  $k' = k/(\rho_0 c^2 \lambda^2)$ . The error threshold for the Poisson equation is set to  $\varepsilon = 10^{-5}$  for all simulations. Additionally, the non-dimensional relaxation time for solving the Cahn–Hilliard equation (2.22) is set to  $\tau'_f = \tau_f/\Delta t = 1$ , and the non-dimensional mobility coefficient,  $M' = M\varpi/(\tau\lambda^3)$  is set to  $M' = 0.001/\beta'$ . It should be pointed out that Lee et al. [11] suggested  $M' = 0.02/\beta'$  to get the most stable results, but such a large mobility would lead to additional diffusion in the fluid. Hence, we selected a lower mobility coefficient to prevent such an unrealistic diffusion. The relaxation time  $\tau'_g$  is set to unity in all cases, because this is a suitable choice to ensure that LBM simulations are numerically stable. In the following, our LBM model will be successively applied to the simulation of: (i) a droplet impacting a thin wet surface; (ii) a bubble rising in a quiescent heavier fluid; and (iii) a breaking ocean wave.

### 3.1. Droplet impact on a thin wet surface

Simulating the impact of a droplet on a wet surface is important for a wide variety of engineering and industrial design problems, such as turbine blades, ink-jet printing, and internal combustion engines. Worthington was the first to study the droplet splash in 1908 [20]. However, more than a century later, this phenomenon is still far from being fully understood, due to the complexity of the problem. The important parameters physical this problem are: (i) the direction of impact (normal or oblique); (ii) the depth of the fluid layer; (iii) the viscosity and density ratios of the two fluids; (iv) whether these are Newtonian or non-Newtonian fluids; (v) the droplet velocity at the time of impact; and (vi) the importance of gravitational effects. Here, for simplicity, we investigate the impact of a single droplet on a thin layer of the same fluid, only up to a short time after the impact. Earlier experimental work [21,22] shows that the impact of a droplet on a thin liquid film may follow two patterns: (i) splashing; and (ii) spreading (deposition). In the spreading pattern, the impact velocity is low and the droplet spreads over the wall, taking the shape of lamella, whereas in the splashing pattern, the impact velocity is high and the lamella eventually takes the shape of a crown consisting of a thin liquid sheets with an unstable free rim at the top, from which numerous small secondary droplets are ejected (Rayleigh–Plateau instability). The important dimensionless numbers relevant for the droplet impact problem are the Weber number and the Reynolds number, which based on, e.g., the properties of fluid 1, read,

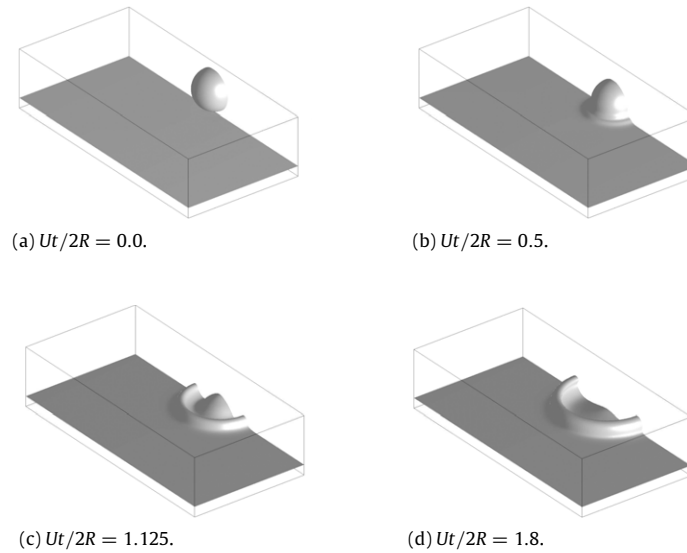
$$\text{We} = \frac{2RU^2\rho_1}{\sigma_{12}} \quad \text{and} \quad \text{Re} = \frac{2RU}{\nu_1} \quad (3.4)$$

where  $U$  is the impact velocity (velocity when the droplet touches the fluid film),  $\rho_1$  and  $\nu_1$  are the density and kinematic viscosity of fluid 1, respectively,  $R$  is the initial radius of the droplet and  $\sigma_{12}$  is the surface tension for the two fluids. Dimensional analysis shows that the gravity force is negligible in the droplet impact and therefore this term ( $G_i$ ) is not kept in the LBM equations for this application.

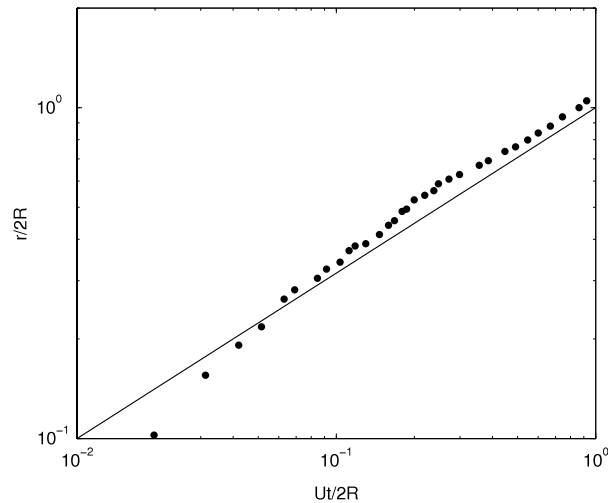
We simulate the fall of a 3D droplet made of fluid 1 in a second lighter fluid (fluid 2), concluded by the droplet impact on a thin layer of fluid 1. Fig. 3.1 shows the computed time evolution of the droplet impact where, owing to symmetry in the  $x$  direction, the solution in only half of the domain was calculated. The grid size is  $128 \times 256 \times 64$  in the  $x$ ,  $y$ , and  $z$  directions, respectively. The density ratio is  $\rho_1/\rho_2 = 1000$  and the viscosity ratio is  $\mu_1/\mu_2 = 100$  (such as for water (1) and air (2)). The Reynolds number is  $\text{Re} = 200$ , with  $U = 0.01\Delta x/\Delta t$  and the Weber number is  $\text{We} = 8000$ .

Earlier work [21,22] indicates that the droplet spreading radius  $r$  should vary with time as  $\sqrt{2RUt}$  during the splashing process (power law). To confirm this, the dimensionless spreading factor ( $r/2R$ ) is plotted versus the non-dimensional time ( $Ut/2R$ ) in Fig. 3.2. Our numerical results are found to follow the power law, except for a short time after the impact, which could result from the impulsive start of the collision, which does not occur in experiments.

A further validation of our numerical results is performed for the impact of a droplet on a thin layer of fluid by comparing with Lee et al.'s [8] numerical results, which were 2D. In our 3D model, a grid with  $1024 \times 512 \times 2$  points is used in the numerical simulations, where the 2 points across the width are aimed at representing 2D conditions. The domain set-up is shown in Fig. 3.3; symmetry boundary conditions are specified on the leftward and rightward boundaries, and no-slip boundary conditions on the top and bottom walls; periodic boundary conditions are specified along the third dimension. As in Lee et al., the density ratio is set to  $\rho_1/\rho_2 = 1000$  and the viscosity ratio is adjusted to  $\mu_1/\mu_2 = 40$ . The Weber



**Fig. 3.1.** Time evolution of the 3D simulation of a droplet impacting a thin layer of fluid, at four different times  $t$ , for  $\rho_1/\rho_2 = 1000$ ,  $\mu_1/\mu_2 = 100$ ,  $Re = 200$  and  $We = 8000$ .



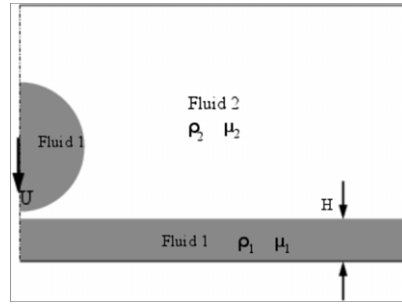
**Fig. 3.2.** Spread factor  $r/2R$  as a function of  $Ut/2R$  in the 3D simulation of a droplet impacting a thin layer of fluid, for  $\rho_1/\rho_2 = 1000$ ,  $\mu_1/\mu_2 = 100$ ,  $Re = 200$  and  $We = 8000$ . The straight line corresponds to the power law  $r = \sqrt{2Ut}R$ .

number is set to  $We = 8000$ , the droplet diameter is  $R = 60\Delta x$ , the layer thickness is such that  $H/R = 0.25$ , and the impact velocity is  $U = 0.005\Delta x/\Delta t$ . Four different simulations were performed for four different Reynolds numbers,  $Re = 20, 100, 500, 2000$ .

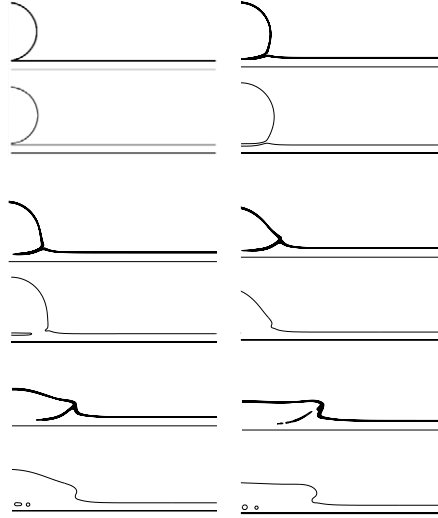
Figs. 3.4–3.6 show the computed time evolution, for a few non-dimensional times  $Ut/2R$  values, of the droplet shape after impact in the first three cases, compared to Lee et al.'s [8] results. Overall, we observe a very good agreement for all three configurations with Lee et al.'s results. In Fig. 3.4, the Reynolds number is low and deposition patterns occur. Viscous forces dominate the initial phase of the impact and the droplet gently spreads on the surface without splashing at all. Immediately after impact, a certain amount of the fluid 1 is trapped within fluid 2, and small bubbles are formed, due to effects of surface tension. The smaller bubbles diffuse into fluid 2 and eventually disappear, after they become smaller than the grid resolution, while the larger ones (Fig. 3.4(e)/(f)) persist and advect with the flow.

In Figs. 3.5 and 3.6 the impact process occurs for higher Reynolds numbers  $Re = 100, 500$ , for which droplet inertia is sufficient to overcome the viscous forces, causing the creation of a finger of fluid 1 into fluid 2. This finger is not stable





**Fig. 3.3.** Domain and parameter definition for the quasi-2D simulation of a droplet impacting a thin fluid layer, following Lee et al. [8].



**Fig. 3.4.** Time evolution in quasi-2D simulations of a droplet impacting a thin fluid film for  $Ut/2R = 0.0, 0.1, 0.2, 0.4, 0.8, 1.6$ .  $Re = 20$ ,  $We = 8000$ ,  $\rho_1/\rho_2 = 1000$  and  $\nu_1/\nu_2 = 40$ ; top: Lee et al.'s results [8], bottom: current LBM model.

and eventually breaks up into smaller droplets. This process, also referred to as Rayleigh–Plateau instability, can be clearly observed in Figs. 3.5 and 3.6(f), whereas this instability was not observed in the work of Lee et al.

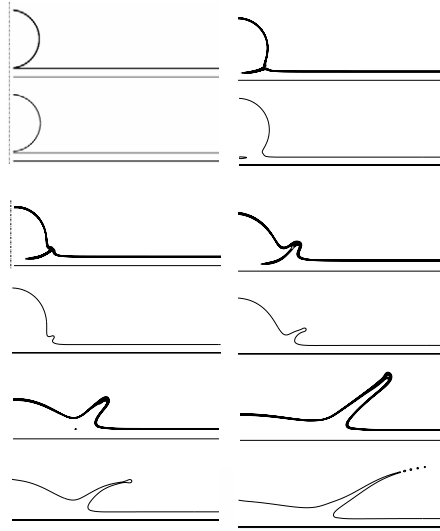
Finally, results for a droplet impacting a fluid layer at an even higher Reynolds number,  $Re = 2000$ , are shown in Fig. 3.7. Note that the method of Lee et al. was not able to go beyond  $Re = 100$  (based on the grid size they used), because of the dependence of relaxation time on viscosity, which has been removed in the current work.

### 3.2. Rising bubble in a quiescent fluid

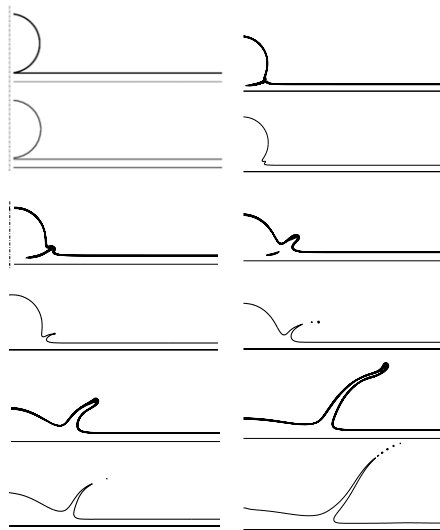
Here, we model the dynamic behavior of a 3D bubble of a light fluid ( $\rho_2, \nu_2$ ) rising in a heavier fluid ( $\rho_1, \nu_1$ ), due to the buoyancy force. Although the LBM simulation setup in terms of grid initialization and boundary conditions is straightforward, the flow structure around the bubble is quite complex and governed by competing effects of viscosity, buoyancy, and surface tension forces. Several experimental studies have been conducted to measure the rise and deformation of single bubbles in a quiescent fluid [23,24]. These showed that the bubble shape greatly varies according to various flow regimes defined by non-dimensional parameters, such as the Bond number  $Bo$  (also known as Eotvos number, the ratio of gravity to surface tension forces), the Reynolds number  $Re$ , and the Morton number  $Mo$ , defined as,

$$Bo = \frac{gD^2}{\sigma_{12}}(\rho_1 - \rho_2); \quad Mo = \frac{g\mu_1^4}{\sigma_{12}^3\rho_1} \left(1 - \frac{\rho_2}{\rho_1}\right); \quad Re = \frac{\rho_1 UD}{\mu_1} \quad (3.5)$$

with the bubble terminal velocity  $U$  and diameter  $D$ , the gravitational acceleration  $g$ , and the dynamic viscosity of the heavier fluid  $\mu_1$ . The terminal shapes of individual rising bubbles were experimentally measured for a broad range of Reynolds and Bond numbers [23], and can be categorized into three shape regimes: (a) spherical, (b) ellipsoidal, (c) spherical cap. In the spherical regime, for small  $Bo$  values, surface tension is dominant and prevents the deformation of the bubble to occur under



**Fig. 3.5.** Time evolution in quasi-2D simulations of a droplet impacting a thin fluid film for  $Ut/2R = 0.0, 0.1, 0.2, 0.4, 0.8, 1.6$ .  $Re = 100$ ,  $We = 8000$ ,  $\rho_1/\rho_2 = 1000$  and  $\nu_1/\nu_2 = 40$ ; top: Lee et al.'s results [8], bottom: current LBM model.

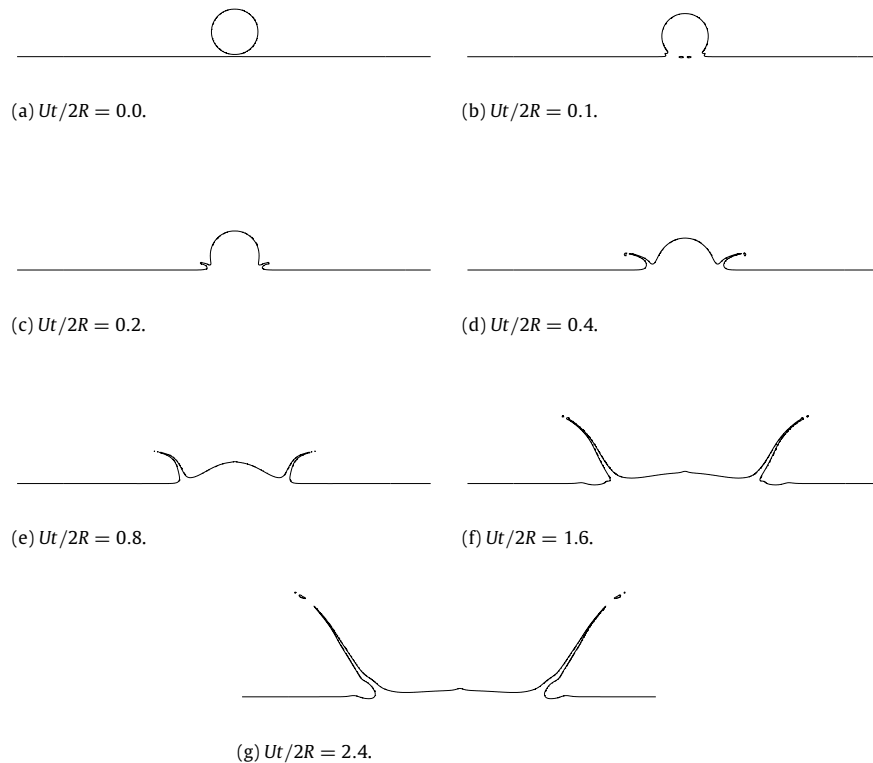


**Fig. 3.6.** Time evolution in quasi-2D simulations of a droplet impacting a thin fluid film for  $Ut/2R = 0.0, 0.1, 0.2, 0.4, 0.8, 1.6$ .  $Re = 500$ ,  $We = 8000$ ,  $\rho_1/\rho_2 = 1000$  and  $\nu_1/\nu_2 = 40$ ; top: Lee et al.'s results [8], bottom: current LBM model.

inertia and viscous forces; consequently, the shape of the bubble remains (almost) spherical during its rise. When increasing the Reynolds and Bond numbers, the contribution of surface tension gradually becomes less important as compared to inertia, and the terminal shape of the bubble becomes ellipsoidal for moderate Reynolds and Bond numbers ( $10 < Re < 500$  and  $10 < Bo < 100$ ), and spherical cap for high Reynolds and Bond numbers.

In our 3D LBM simulations, a spherical bubble of density  $\rho'_2 = 1$  and initial diameter  $D'_0 = 32$  is placed at one bubble diameter above the bottom of a domain of size  $4D' \times 4D' \times 8D'$ , filled with a fluid of density  $\rho'_1 = 1000$  (hence the  $\rho'_1/\rho'_2 = 1000$ ); the fluid viscosity ratio is  $\mu'_1/\mu'_2 = 100$ , which approximates the air–water case. Amaya et al. [25] reported that wall effects are negligible when a domain size of  $4D'$  or larger in the radial direction is used. Also, the vertical domain size has to be large enough to allow the bubble to adjust to its final steady shape during its rise. The required distance of this to occur is different for each case and depends on the Reynolds number, which directly controls the bubble rising velocity. For cases up to  $Re \sim 100$ , a domain of height  $8D'$  was found to be sufficient.

Initially, both fluids are stationary; free slip boundary conditions are prescribed on the lateral sides of the domain and a bounce-back condition on the top and bottom boundaries. Simulations are run for four distinct test cases (A–D) with



**Fig. 3.7.** Time evolution in quasi-2D simulations of a droplet impacting a thin fluid film, for  $Re = 2000$ ,  $We = 8000$ ,  $\rho_1/\rho_2 = 1000$  and  $v_1/v_2 = 40$ , using current LBM model.

**Table 3.1**

Rise of a 3D bubble of a light fluid 2 in a heavy fluid 1 ( $\rho'_1/\rho'_2 = 1000$ ;  $\mu'_1/\mu'_2 = 100$ ). Comparison of terminal Reynolds number between experiments [24] and 3D-LBM numerical simulations.

Test case	Mo	Bo	Simulated Re	Experimental Re	Relative error on Re (%)
A	711	17.7	0.189	0.232	18.53
B	$8.2 \times 10^{-4}$	32.2	55.1	54.6	1.26
C	266	243	7.51	7.77	3.34
D	43.1	339	17.74	18.3	3.06

different  $Mo$  and  $Bo$  values. In Table 3.1, the terminal Reynolds number is compared with experimental data and a reasonable agreement between numerical and experimental results can be observed, but there are larger discrepancies for case A. Similar larger errors for this case were reported in the work of Hua et al. [26], which they interpreted as being due the extremely low rising velocity leading to higher relative errors.

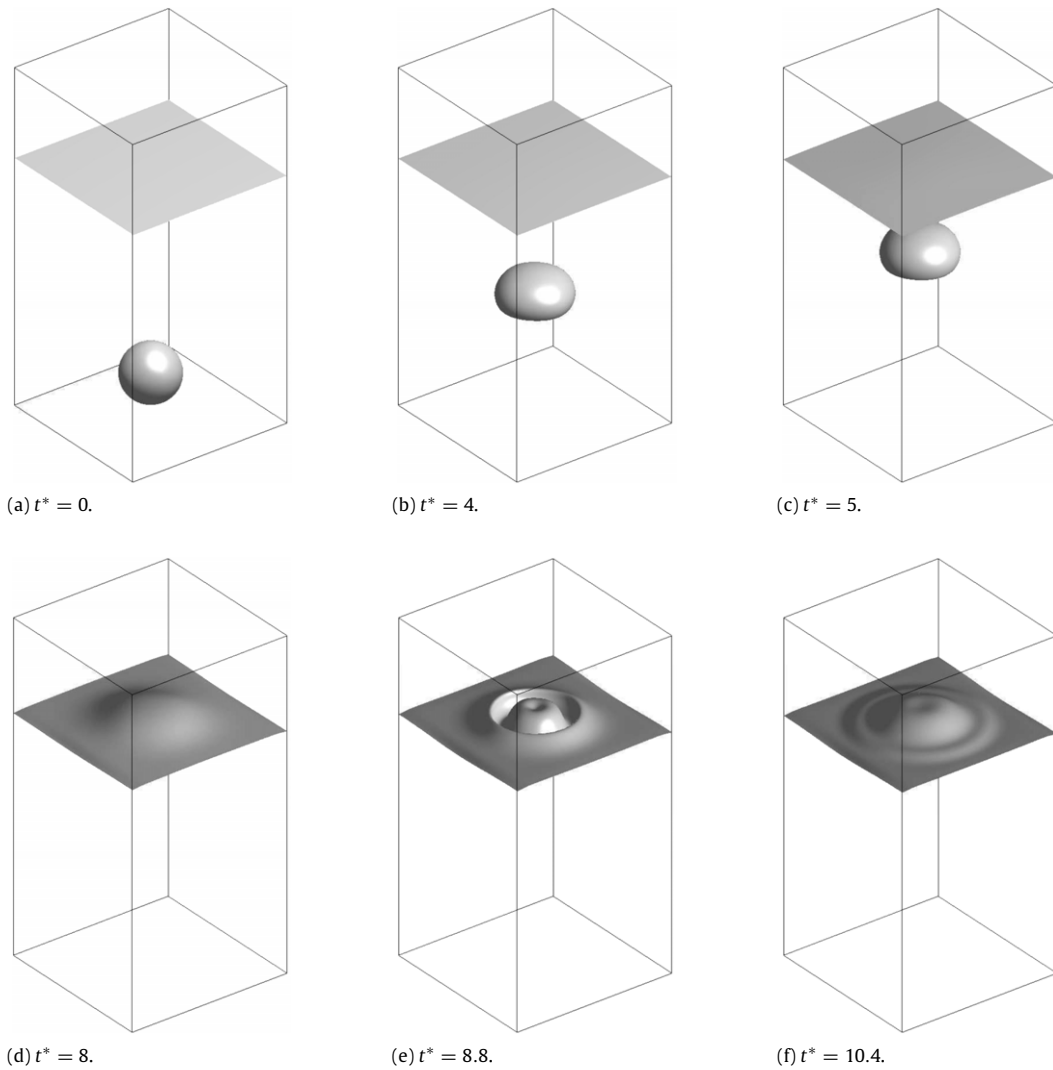
The computed terminal shapes of the bubble for cases (A)–(D) are compared in Table 3.2 to experimental results [24] and to independent numerical results [26]. We see that our computed shapes agree very well with the reference data and that spherical, ellipsoidal, and spherical caps have been formed depending on the case.

Fig. 3.8 further shows the computed time evolution of the bubble shape during its rise, for case (C). During the early stages of the simulation, buoyancy forces are dominant and accelerate the bubble. As the bubble picks up speed, viscous drag gradually changes its shape, bending it downstream. Eventually, the terminal shape of the bubble is formed when buoyancy, surface tension, and viscous force contributions are balanced.

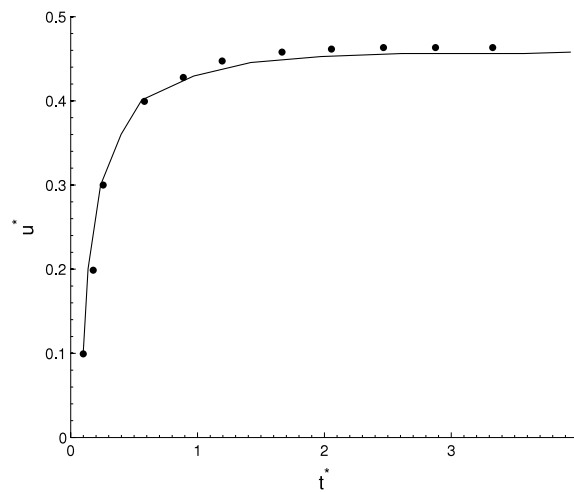
Finally, in Fig. 3.9, the computed bubble rising velocity as a function of time is compared to the numerical results of Hua et al. [26]. The agreement between both sets of numerical results is very good.

### 3.3. Breaking wave

In this last application, we solve the demanding test case of a 3D breaking ocean wave at an air–water interface. Earlier numerical simulations based on fully nonlinear potential flow theory [27,28] showed that a periodic sinusoidal wave of large amplitude, with initial velocities specified from linear wave theory, is not stable in a fully nonlinear model and rapidly overturns and breaks, when used as an initial condition. This type of initialization has thus often been used to rapidly create




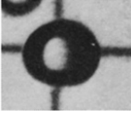
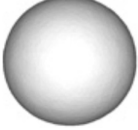


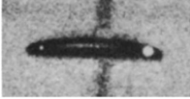


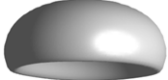
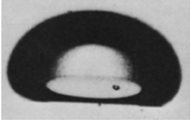


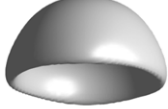

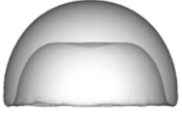

**Fig. 3.8.** Time evolution of a rising bubble of fluid 2 into fluid 1, for case (C), with  $Et = 243$ ,  $Mo = 266$ ,  $t^* = \sqrt{g/D}t$ ,  $\rho'_1/\rho'_2 = 1000$ , and  $\mu'_1/\mu'_2 = 100$ .



**Fig. 3.9.** Rising bubble of fluid 2 into fluid 1: non-dimensional rising velocity  $U^* = u/\sqrt{gD}$  as a function of non-dimensional time  $t^* = \sqrt{g/D}t$ : 3D LBM results (●); experimental results of [26] (—).

**Table 3.2**

Comparison of the computed terminal shapes of a rising bubble of fluid 2 into fluid 1, with the experimental results of [24] and the numerical simulations of Hua et al. [26], for  $\rho'_1/\rho'_2 = 1000$  and  $\mu'_1/\mu'_2 = 100$ .

Test case	Current results	Experimental results	J. Hua et al. results
(A)			
$Bo = 17.7$			
$Mo = 711$			
(B)			
$Bo = 32.2$			
$Mo = 8.2 \times 10^{-4}$			
(C)			
$Bo = 243$			
$Mo = 266$			
(D)			
$Bo = 339$			
$Mo = 43$			

a breaking wave, when validating both single or multiple fluid NS models [29]. To limit computational time, simulations typically assume spatial periodicity in the main flow direction. Hence, this problem is still quasi-2D in the vertical plane, although 3D turbulent flow structures appear in the transverse direction to that of wave propagation.

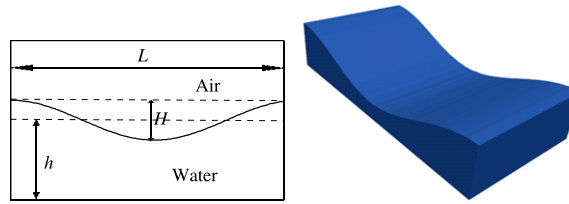
The initial velocity field and interface shape for a 2D linear wave of height  $H$  in depth  $h$ , specified in a 3D domain, are given by [29] (with  $z$  denoting the vertical direction and  $z = 0$  at the undisturbed air–water interface),

$$\begin{aligned}
 \eta &= \frac{H}{2} \cos(kx) \\
 u &= \frac{H}{2} \sigma \frac{\cosh k(h+z)}{\sinh(kh^*)} \cos(kx) \\
 w &= \frac{H}{2} \sigma \frac{\sinh k(h+z)}{\sinh(kh^*)} \sin(kx) \\
 v &= 0
 \end{aligned} \tag{3.6}$$

where  $\sigma = 2\pi/T$  is the wave angular frequency,  $k = 2\pi/L$  is the wave number (Fig. 3.10), and  $h^* = h + \epsilon$ , where  $\epsilon(y) = 0.08y$  is a small linear perturbation of the seafloor geometry, denotes a perturbed depth used in the model for triggering unsymmetrical flows and vortices in the transverse direction to the main flow, thus ensuring that turbulent structures will rapidly develop at high Reynolds numbers [29].

Because the finest LBM mesh resolution (i.e., largest grid) achievable on a single GPGPU in this application will not be fine enough to capture all the turbulent scales, as in [29], we use a Large Eddy Simulation (LES) model as a subgrid scale turbulence model. In LES, a spatial filter is applied to the velocity field, which should be fine enough that the largest turbulent structures of the flow are not filtered out [4]; in LBM-LES,  $\Delta x$  is typically used as the filter length (see Krafczyk et al. [30] for details). In the Smagorinsky LES model, a turbulent eddy viscosity  $\mu_T$  is added to the molecular viscosity  $\mu$  for calculating total forces (viscous and turbulent sub-scale forces) in Eq. (2.9), which is defined as,

$$\mu_T = \rho(C_s \Delta x)^2 \|\mathbf{S}\| \tag{3.7}$$



**Fig. 3.10.** Definition sketch in the vertical plane ( $x, z$ ) for the initial interface profile of a large amplitude sinusoidal wave (left), used to initialize the 3D-LBM domain (right).

with  $C_s$  the Smagorinsky constant and  $\mathbf{S}$  the strain rate tensor being defined by,

$$S_{\alpha\beta} = \frac{1}{2} \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right). \quad (3.8)$$

Since in our LBM scheme, spatial derivatives of the velocity field have already been calculated using Eq. (2.5), there is no need to recalculate them for computing the strain rate tensor, which makes using a LES quite efficient in our model.

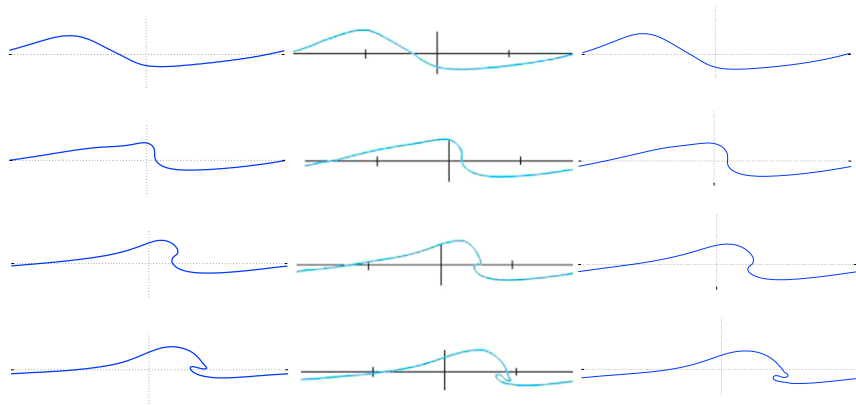
The key dimensionless flow parameters for the periodic breaking wave problem are the Reynolds number,  $Re = \rho_w c L / \mu_w$ , the density ratio,  $\rho_w / \rho_a$ , and the viscosity ratio,  $\mu_w / \mu_a$  (where  $w$  and  $a$  denote water and air respectively). Dimensionless wave parameters are the steepness  $H/L$ , which quantifies wave nonlinearity, and the relative depth,  $h/L$ , which quantifies dispersive effects. In the following application, the fluids characteristics are set to the properties of air and fresh water, i.e.,  $\mu_w / \mu_a = 55$ ,  $\rho_w / \rho_a = 855$  with  $\rho_w = 1000 \text{ kg/m}^3$  and  $\mu_w = 10^{-3} \text{ N s/m}^2$ .

Lubin et al. [29] solved this problem with a 3D-NS-VOF model, using a LES subgrid scheme, in a  $250 \times 25 \times 75$  grid in the  $x, y$ , and  $z$  directions ( $z$  denoting the vertical direction), using a uniform grid size of  $0.0004 \text{ m}$ . In the following we compare results of our 3D-LBM model to the latter study, while assessing their sensitivity to grid resolution. As in Lubin et al., we specify as initial condition a steep sinusoidal wave, with  $H/L = 0.13$  and  $h/L = 0.13$  (i.e., an intermediate depth wave). Based on linear wave theory, the wave celerity is  $c = 0.324 \text{ m/s}$  for a wave period of  $T = 0.308 \text{ s}$  and a wavelength  $L = 0.1 \text{ m}$ , which results in a Reynolds number  $Re = 32,400$ . Note that because of the short wavelength, surface tension effects are expected to play a significant role in the interface dynamics. In the LBM model, free slip boundary conditions are specified on the top and bottom boundaries of the computational domain, and periodic boundary conditions are applied on the four vertical sides (Fig. 3.10).

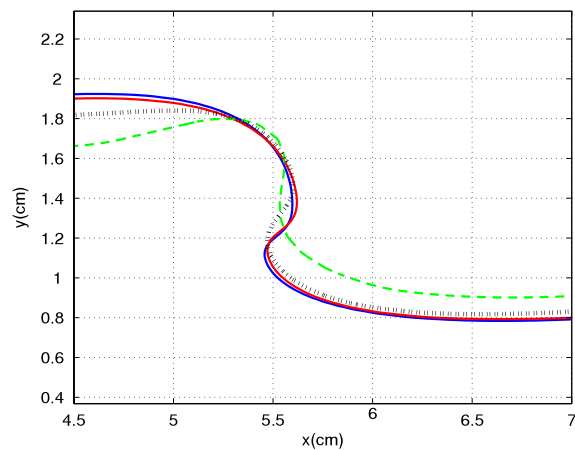
We first solve the problem in a  $256 \times 128 \times 64$  LBM grid (i.e., with  $N = 2.1$  million grid cells; referred to as “grid 1”), which has a resolution in the vertical plane similar to that of Lubin et al. and is wide enough in the lateral direction to ensure fully developed 3D turbulent flow structures. Then, to improve the resolution of breaker jets and better match Lubin et al.’s results in the vertical plane, we use a  $512 \times 64 \times 128$  LBM grid (i.e., with  $N = 4.2$  million grid cells; referred to as “grid 2”). This larger grid, however, is narrower (one-eighth of the length compared to half for grid 1) because of memory limitations on a single GPGPU (owing to the 3 sets of LBM distribution functions and related physical fields that need to be discretized), which will prevent fully turbulent structures from developing in the transverse direction. A convergence study of breaker shape to grid resolution, while maintaining a 4:1 ratio of horizontal to vertical grid size, finally shows that a  $400 \times 100$  resolution in the vertical plane is sufficient to obtain converged breaker shapes. This allows increasing the width to a quarter the length, yielding  $400 \times 100 \times 100$  cells (4 million), for a grid referred to as grid 3.

Fig. 3.11 first shows the time evolution of the free surface profile, before breaking occurs, computed in the middle lateral cross section ( $y = 0$ ; Fig. 3.10), at four times  $t/T = 0.17, 0.3, 0.38$  and  $0.46$ , in our 3D model for both grids 1 and 2, compared to Lubin et al.’s results [29]. As expected, the wave quickly overturns and develops a plunging jet. The agreement of our results with Lubin et al.’s results is good for both grids, although the shape of the plunging jet appears to be in better agreement in the finest resolution grid 2.

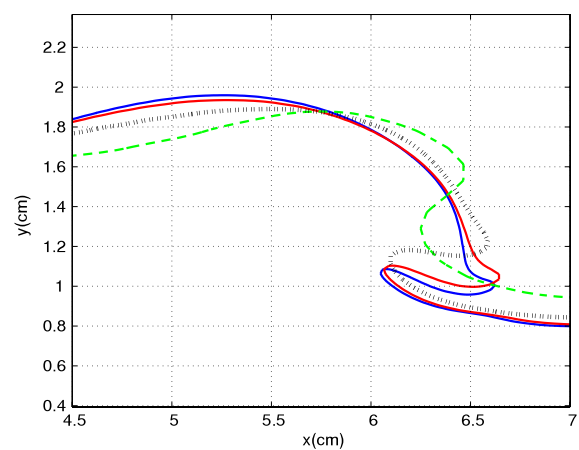
To assess the convergence of LBM results with grid resolution, in Fig. 3.12, we compare results of LBM simulations up to impact of the breaker jet on the free surface, for 6 different grid resolutions, of which the first one is identical to grid 2 and the second one has the same resolution in the vertical plane as grid 3: (1)  $512 \times 64 \times 128$  ( $N = 4.2$  million); (2)  $400 \times 50 \times 100$  ( $N = 2$  million); (3)  $320 \times 40 \times 80$  ( $N = 1$  million); (4)  $256 \times 32 \times 64$  ( $N = 0.5$  million); (5)  $200 \times 25 \times 50$  ( $N = 0.25$  million); and (6)  $128 \times 16 \times 32$  ( $N = 0.066$  million). All these grids have a mesh ratio of 8:1:2, in order to study the convergence of the numerical scheme in similar conditions (i.e., their width is one-eighth their length and their height one quarter their length). Fig. 3.12 compares the computed air–water interface geometry, in the main-cross-section ( $y = 0$ ) of the 3D-LBM domain, for clarity, only for four different resolutions (1, 2, 4 and 6). While large differences are observed between the last 3 coarsest resolutions 2, 4 and 6, these differences become quite small between the two finest resolutions 1 and 2. Hence, both grid 2 and 3, seem to ensure convergence on the breaker shape in the vertical plane. To quantify convergence of the simulations, in the absence of detailed reference data, we compute the  $L_2$  relative error norm of the air–water interface location between results in the 5 coarsest discretizations (2–6) and those in the finest one (1; grid 2). This error norm is plotted in Fig. 3.13 as a function of the resolution, expressed as the number of grid cells in the  $x$  direction. Convergence clearly occurs and we see that it is approximately second-order in the log–log plot.



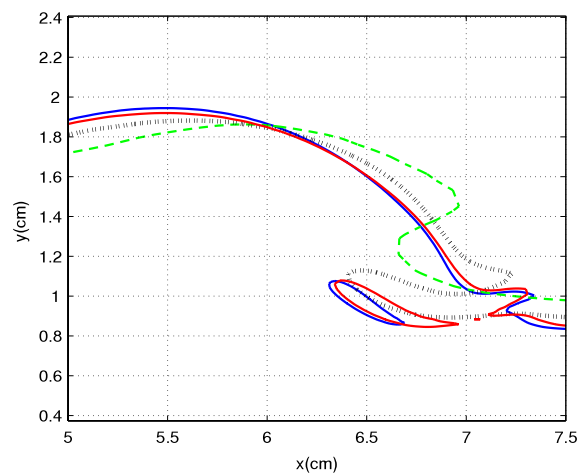
**Fig. 3.11.** Time evolution of a 3D breaking sinusoidal wave with initial characteristics:  $H/L = 0.13$  and  $h/L = 0.13$ ,  $Re = 32,400$ . Comparison of pre-breaking surface profiles in the main vertical cross-section ( $y = 0$ ; Fig. 3.10), at  $t/T = 0.17, 0.3, 0.38$ , and  $0.46$  (top to bottom) in: (left, right) 3D-LBM results in grids 1 and 2, respectively; (middle) Lubin et al.'s 3D-NS-VOF results [29].



(a)  $t = 0.38T$ .



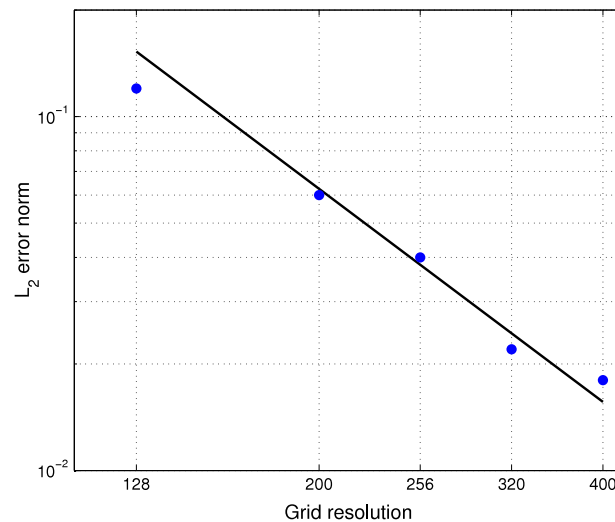
(b)  $t = 0.46T$ .



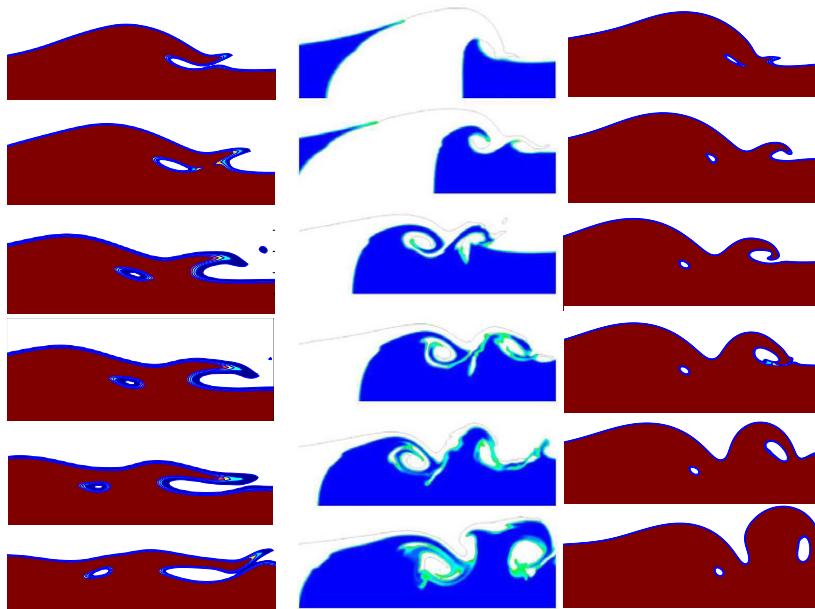
(c)  $t = 0.48T$ .

**Fig. 3.12.** Convergence study for the case of Fig. 3.11. Geometry of the air–water interface in the main cross-section ( $y = 0$ ) for four different resolutions in a grid with dimension ratios 8:4:1 and number of cells in the  $x$  direction: (1; blue; grid 2) 512; (2; red) 400; (4; black) 256; and (6; green) 128. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)





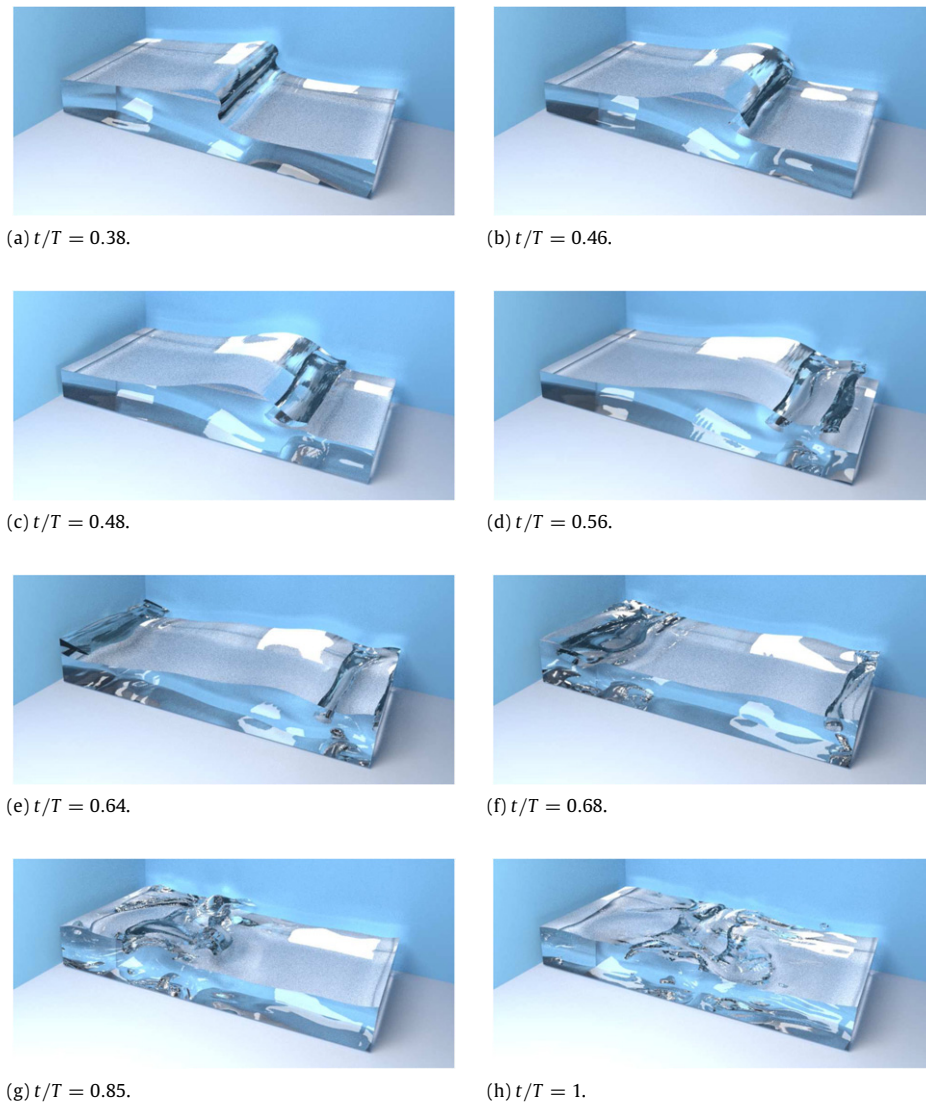
**Fig. 3.13.** Case of Fig. 3.12.  $L_2$  relative error norm, between surface profiles computed for resolution (1) and those for resolutions (2–6), as a function of resolution in the  $x$  direction.



**Fig. 3.14.** Case of Fig. 3.11. Time evolution of a 3D overturning breaking wave in the middle lateral cross section ( $y = 0$ ; Fig. 3.10). Comparison of details of splash-up generation in the main vertical cross-section ( $y = 0$ ; Fig. 3.10) at  $t/T = 0.48, 0.5, 0.52, 0.56, 0.64, 0.68$  (top to bottom) in: (left, right) 3D-LBM results in grids 1 and 2, respectively; (middle) Lubin et al.'s 3D-NS-VOF results [29].

(Note, however, while the proof of convergence is conclusive in such a grid-to-grid comparison, the rate of convergence is only indicative as one would have to use an independent converged solution as a reference to accurately calculate it.)

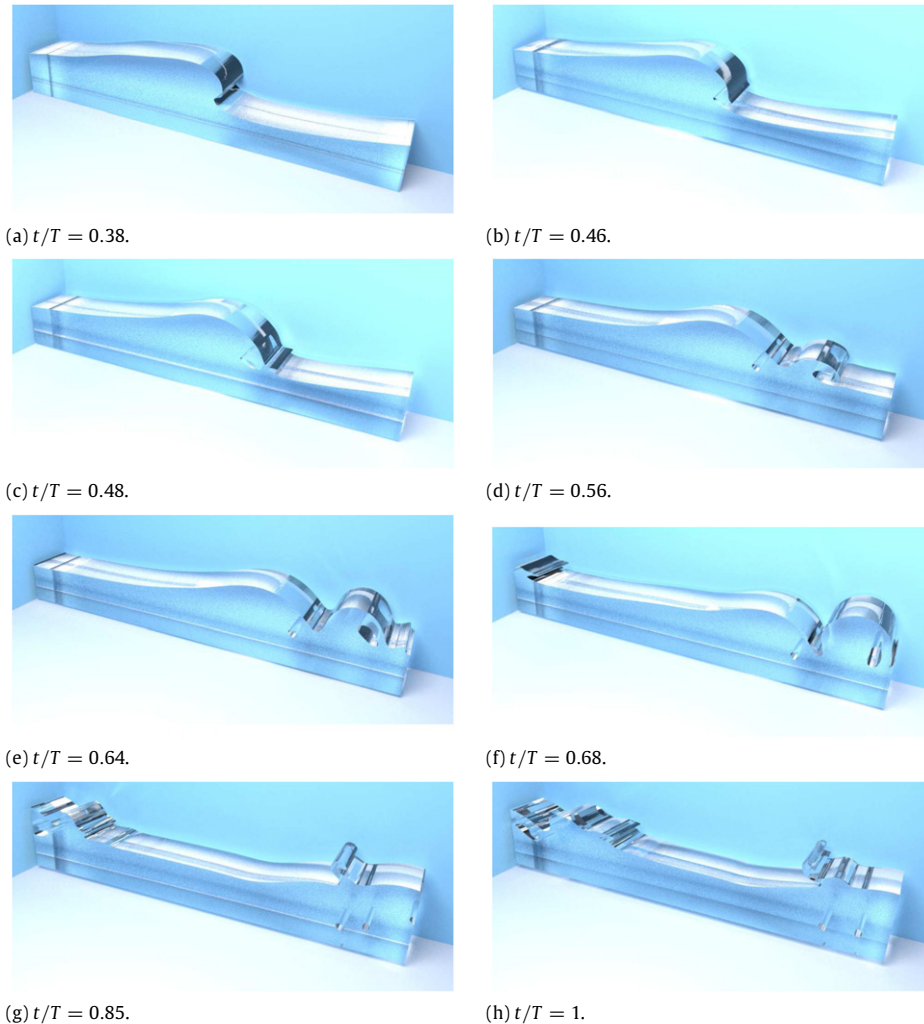
Fig. 3.14 shows details of the splash-up generation after the plunging jet impacts the water surface. Our results for grid 1 and 2 are again compared with those reported by Lubin et al. [29], at 6 different times. (Note that no times were provided in Lubin et al.'s results, so we tried to find similar phases in our computations to perform this qualitative comparison.) In all sets of results, after the initial jet impact, a secondary rebound jet occurs while an air mattress is enclosed in the water. Later on, the rebound jet also impacts the free surface, enclosing an even larger air mattress in the process. While there are differences in the details of the computed free surface shapes, which can be attributed to the different free surface tracking/capturing methods, surface tension force computation, and discretization, overall, our results in both grids are in good qualitative agreement with those of Lubin et al. As for the pre-breaking results, however, we find that the agreement is much better for grid 2 results. The converged shape of the initial plunging jet was also much better modeled in this grid (Fig. 3.12) and, as we see, this applies to the subsequent splash-up and flow structures in the vertical plane during post-breaking processes.



**Fig. 3.15.** Time evolution of a 3D periodic breaking wave for the same case as in Fig. 3.11, for grid 1 with  $256 \times 128 \times 64$  cells in the  $x$ ,  $y$  and  $z$  directions.

Finally, Figs. 3.15, 3.16, and 3.17 show snapshots of 3D-LBM results computed in grids 1, 2 and 3, at eight different times, the first two being the same ( $t/T = 0.38$  and  $0.46$ ) as for the overturning free surface profiles shown in Fig. 3.11, while the last six ( $t/T = 0.48$  to  $t/T = 1$ ) illustrating the wave breaking and post-breaking decay processes up to one wave period, with the 3rd–6th time levels matching some of the times shown in Fig. 3.14. Overall, as expected from previous results in vertical cross-sections, both the wave and breaking processes are much better simulated (and resolved) in grids 2 and 3 than in grid 1. However, as could also have been expected from the much larger width of grid 1 (four times that of grid 2 and twice that of grid 3), for the same length to height ratio 4:1 in all grids, the flow structure becomes much more 3D in grid 1, with larger turbulent structures appearing in the transverse direction towards the end of the simulations. While almost no turbulent structures appear in grid 2, however, grid 3 seems to offer a reasonable compromise, with a good much of breaker shapes in the vertical plane and some turbulent structures in the transverse direction.

More specifically, in the first two snapshots, while the wave is not broken yet and hence is still fairly 2D in all grids, one can already see in grid 1 the appearance of lateral flow perturbations triggered by the perturbation in seafloor bathymetry. Later on, at  $t/T = 0.48$ ,  $0.56$  and  $0.64$ , shortly after the plunging jet has impacted the free surface, as in Fig. 3.14, a sizable splash up jet is generated in all 3 grids, but much more so in grids 2 and 3, which creates a secondary rebound jet, with the enclosure by the plunging water jet of a cylindrical air cushion. At this stage, the flow is still fairly laterally uniform, although some transverse instabilities are clearly starting to grow, particularly in grid 1. Later on, in the last three snapshots, the rebound jet impacts the free surface and the wave gradually dissipates. In grids 1 and 3, the flow at this stage clearly becomes turbulent and 3D, with large lateral variations. We also see sizable pockets of air enclosed in the water at a few locations. In grid 2, however, lateral turbulence does not significantly develop and, likely as a result of this quieter dynamics,



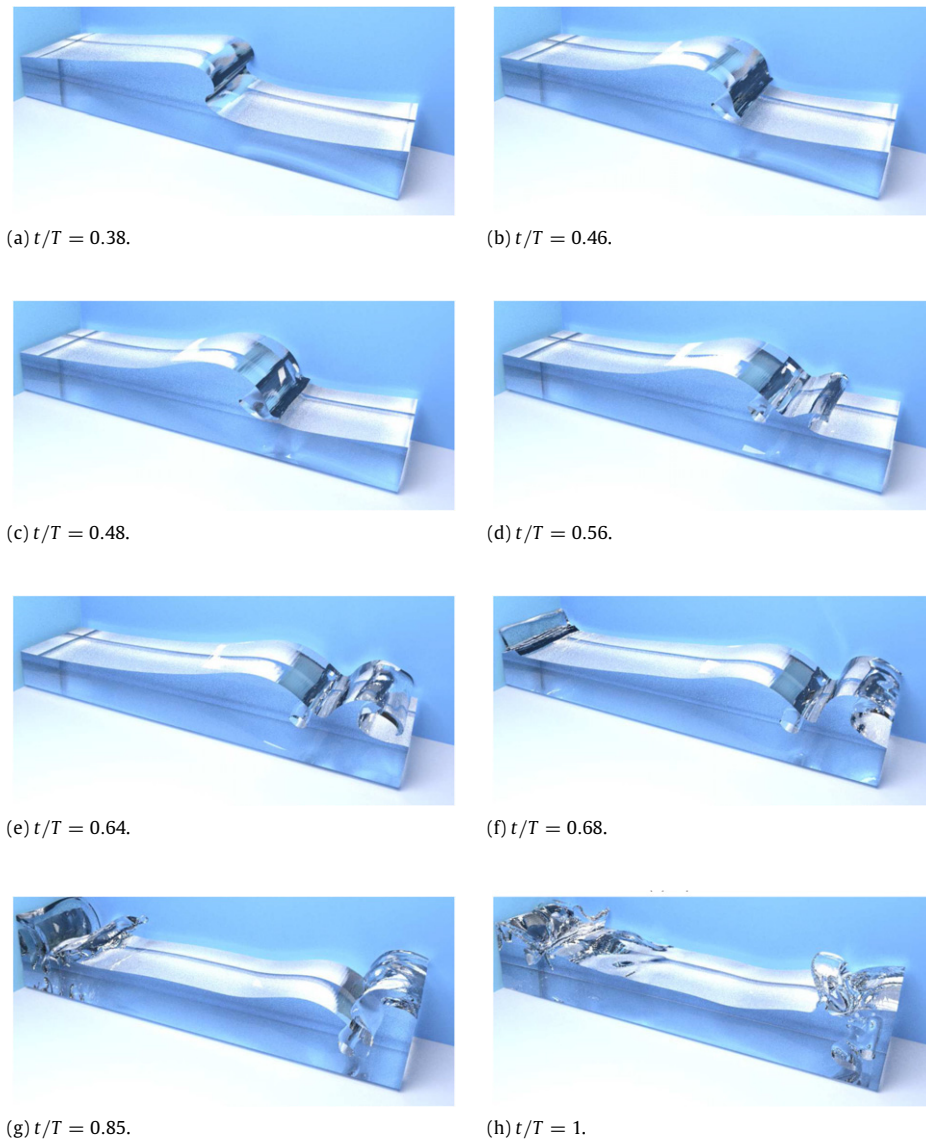
**Fig. 3.16.** Time evolution of a 3D periodic breaking wave for the same case as in Fig. 3.11, for grid 2 with  $512 \times 64 \times 128$ , in the  $x$ ,  $y$  and  $z$  directions.

the wave dissipates more slowly than in grids 1 and 3. Clearly, one would need to use a finer lateral discretization in grid 2, to achieve a fully realistic simulation, with sufficient turbulence.

As indicated above, we find that grid 3 results offer a good compromise between the better resolved breakers of grid 2 and the more intense lateral turbulence of grid 1. Nevertheless, all these simulations still lack the stronger splash up seen in many observations of breaking ocean waves, with many water parcels being ejected in the air and widespread air bubble generation in the water. The likeliest reason for this is that our 3D-LBM discretization is not fine enough to properly resolve such phenomena, due to the memory/grid size limitation when using a single GPGPU. This could only be improved with a significant increase in grid size, which would require implementing our numerical code on a large GPUGPU–CPU cluster, where one could perhaps achieve on the order of 100 million grid cells (i.e., an improvement in resolution of nearly a factor of 3 in each direction, as compared to grid 2). In view of the maximum of 4.2 million cells we were able to achieve here on a single GPGPU, this would require using on the order of 25 GPGPUs. As an indication, recent NS-VOF two-fluid flow simulations by Lubin and Glockner [31], implemented on thousands of CPUs, have shown that very fine flow structures can be resolved in breaking wave simulations, when using tens of millions of grid cells. This extension of our 3D-LBM model to an implementation on large GPGPU clusters, which is hardware already available, will be left out for future work.

#### 4. Conclusions and outlook

In this paper, we presented the development, numerical implementation and validation of an efficient 3D Lattice Boltzmann model (LBM) for the simulation of multiple-fluid flows with high density ratios, at high Reynolds number. This model both extends our earlier 2D model [5] to 3D but, importantly, to high Reynolds number flows. While some of the principles for our method were first proposed by Inamuro et al. [9], we improved the method by using: (i) a 3D D3Q19



**Fig. 3.17.** Time evolution of a 3D periodic breaking wave for the same case as in Fig. 3.11, for grid 3 with  $400 \times 100 \times 100$ , in the  $x$ ,  $y$  and  $z$  directions.

lattice operator; (ii) a more accurate and rigorous formulation of body forces and surface tension terms; and (iii) a new interface capturing method based on solving a convection–diffusion Cahn–Hilliard equation. Additionally, our model relies entirely on the LBM, using three different sets of LBM particle distribution functions for solving (pressureless) Navier–Stokes (NS) equations, a Poisson equation for the pressure field in order to correct the velocity field to satisfy mass conservation, and the Cahn–Hilliard equation. This complete LBM approach makes our numerical scheme particularly well-suited, and thus efficient, for an implementation on heavily parallel GPGPU boards (NVIDIA TESLA 2070). Here, with an implementation in double precision on a single GPGPU, we could achieve up to 20 MNUPS for discretizations of up to 4.2 million grid cells in the breaking wave application. The latter simulation was run in a  $512 \times 64 \times 128$  grid for one wave period, or 101 136 time steps, yielding a total computational time of about 6 h. By contrast, the simulation run in the  $400 \times 100 \times 100$  grid (4 million nodes) had a larger time step and only took 3.5 h to compute one wave period. Inamuro et al. reported that the simulation of two droplets collision with a coarse grid  $192 \times 48 \times 96$  takes about 48 h on an AMD AthlonXP 1800+ PC machine (the Poisson equation also is solved in their method). This computational time comparison shows the capability of the proposed algorithm with the GPU calculation.

Lee et al. did not reported the computational time of their simulations and Unfortunately we are not able to make the efficiency comparison of these two methods. Although in their work they do not solve extra Poisson equation, but they still need to calculate the different types of finite difference (central, biased, mixed) for different variables, which is computationally expensive, to make their simulation stable. Considering that in our work the Poisson equation calculation takes 40%



of the total computing time and we are not limited in low Reynolds number flows, we can use the proposed algorithm as a efficient numerical scheme for simulating multiphase flows with high Reynolds number and high density ratios.

To achieve higher *NUPS* values, the Poisson equation solution should be further accelerated, since it must be performed at each time step and represents a large fraction of the computing time. This could be achieved by using a multigrid-type method, which would require smaller numbers of iterations to compute the velocity field correction terms.

The new 3D LBM model was applied to the simulation of various benchmark problems, for which numerical results were found to agree well with the reference data in most cases; convergence of numerical results with grid size was also assessed, for the demanding application of a breaking ocean wave. This latter case, however, revealed the limitations of the present single GPGPU implementation of the model, due to maximum grid size/resolution that can be achieved. This limitation could be overcome by implementing the model on multiple GPGPUs, by way of a parallel MPI algorithm. Except for a small overhead in computational time, due to communication and data exchange between various GPGPUs and the CPUs, one would expect solving problems in larger grids in nearly the same total time on a GPGPU cluster. For instance, the last application for a breaking wave could be solved in a grid 25 times larger (110 million nodes; improvement by a factor of nearly 3 in resolution) using 25 GPGPUs, perhaps in as little as 7 h of computations (assuming a 15% overhead). This will be left out for further work.

We will also note that the convergence and comparison study made for the 3D breaking wave application seemed to indicate that a finer discretization (perhaps by a factor of 1.5–2 in each direction) is required in the 3D-LBM, to produce results comparable to those of the 3D-NS-VOF model of Lubin et al. [29]. Indeed, in their work, they used a  $250 \times 25 \times 75$  grid while our grids 2 and 3 that matched their results in the vertical plane were  $512 \times 64 \times 128$  and  $400 \times 100 \times 100$ , respectively. Such differences in grid size, however, are to be expected. While they directly discretize and solve the NS equations, we solve a set of discrete Boltzmann equations, whose solution converges to that of the NS equation. In doing so, we discretize the LBM distribution functions and not the physical fields (i.e., velocity and pressure). Nevertheless, we expect that the present efficient GPGPU implementation of our 3D-LBM model, should achieve much smaller computational times, even with larger discretizations, for similar accuracy.

On the basis of the present results, we are confident that the multi-GPGPU implementation in combination with an acceleration of the pressure Poisson equation, would make our numerical model a powerful tool for simulating complex 3D two-fluid flows, in order to explore new physics or solving complex engineering problems, with high fluid density ratios at a high Reynolds number.

## Acknowledgment

The authors wish to acknowledge support from grant OCE-09-27014 of the US National Sciences Foundation (NSF) Physical Oceanography Program.

## Appendix A. Chapman–Enskog expansion to retrieve the Navier–Stokes equation

Here we apply Chapman–Enskog expansion [6] to verify that the mesoscopic evolution Eq. (2.4), with the equilibrium distribution functions defined in Eqs. (2.5)–(2.8), converges to the macroscopic “pressureless” NS equations of motion equation (2.3).

It should be mentioned that the below properties of the D3Q19 lattice scheme have been used in deriving the equations in this section,

$$\sum_{i=0}^b w_i = 1 \quad (\text{A.1})$$

$$\sum_{i=0}^b w_i e_{i\alpha} = 0 \quad (\text{A.2})$$

$$\sum_{i=0}^b w_i e_{i\alpha} e_{i\beta} = c_s^2 \delta_{\alpha\beta} \quad (\text{A.3})$$

$$\sum_{i=0}^b w_i e_{i\alpha} e_{i\beta} e_{i\gamma} = 0 \quad (\text{A.4})$$

$$\sum_{i=0}^b w_i e_{i\alpha} e_{i\beta} e_{i\gamma} e_{i\zeta} = c_s^4 (\delta_{\alpha\beta} \delta_{\gamma\zeta} + \delta_{\alpha\gamma} \delta_{\beta\zeta} + \delta_{\alpha\zeta} \delta_{\beta\gamma}) \quad (\text{A.5})$$

$$\sum_{i=0}^b w_i e_{i\alpha} e_{i\beta} e_{i\gamma} e_{i\zeta} e_{i\eta} = 0. \quad (\text{A.6})$$

The zeroth-, first-, second- and third-order moments of equilibrium distribution functions Eq. (2.5) are calculated as,

$$\sum_{i=0}^b g_i^{(\text{eq})} = \sum_{i=0}^b g_i = 0, \quad (\text{A.7})$$

$$\sum_{i=0}^b g_i^{(\text{eq})} e_{i\alpha} = \sum_{i=0}^b g_i e_{i\alpha} = u_\alpha^*, \quad (\text{A.8})$$

$$\sum_{i=0}^b g_i^{(\text{eq})} e_{i\alpha} e_{i\beta} = \Pi_{\alpha\beta}^{(0)} = \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \delta_{\alpha\beta} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\beta} + u_\alpha^* u_\beta^* + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right) \Delta t \quad (\text{A.9})$$

$$\sum_{i=0}^b g_i e_{i\alpha} e_{i\beta} e_{i\gamma} = c_s^2 (\delta_{\alpha\beta} u_\gamma^* + \delta_{\alpha\gamma} u_\beta^* + \delta_{\gamma\beta} u_\alpha^*). \quad (\text{A.10})$$

The Taylor series expansion in space and time of the left hand side of LB equation (2.4) yields,

$$g_i(\mathbf{x} + e_{i\alpha} \Delta t, t + \Delta t) = g_i(\mathbf{x}, t) + \epsilon (\partial_t + e_{i\alpha} \partial_\alpha) g_i(\mathbf{x}, t) + \frac{1}{2} \epsilon^2 (\partial_t + e_{i\alpha} \partial_\alpha) (\partial_t + e_{i\beta} \partial_\beta) g_i(\mathbf{x}, t) + \mathcal{O}(\epsilon^3) \quad (\text{A.11})$$

with the expansion parameter  $\epsilon = \Delta t$  being small compared to the macroscopic time scales. Then, we expand the particle distribution function  $g_i(\mathbf{x}, t)$  as,

$$g_i = g_i^{(\text{eq})} + \epsilon g_i^{(1)} + \mathcal{O}(\epsilon^2), \quad (\text{A.12})$$

with the following multiple-scale expansion for the time derivative,

$$\partial_t = \partial_{t_0} + \epsilon \partial_{t_1} + \mathcal{O}(\epsilon^2), \quad (\text{A.13})$$

where  $t_0$  denotes the “slow” (macroscopic) time and  $t_1$  is the “fast” time.

Substituting the latter two equations into Eq. (A.11) and sorting out terms in orders of  $\epsilon$ , we obtain,

$$\begin{aligned} \epsilon (\partial_{t_0} + e_{i\alpha} \partial_\alpha) g_i^{(\text{eq})} + \epsilon^2 \left\{ \partial_{t_1} g_i^{(\text{eq})} + (\partial_{t_0} + e_{i\alpha} \partial_\alpha) g_i^{(1)} + \frac{1}{2} (\partial_{t_0} + e_{i\alpha} \partial_\alpha) (\partial_{t_0} + e_{i\beta} \partial_\beta) g_i^{(\text{eq})} \right\} + \mathcal{O}(\epsilon^3) \\ = -\frac{1}{\tau_g^*} (\epsilon g_i^{(1)} + \epsilon^2 g_i^{(2)}) + \mathcal{O}(\epsilon^3) \end{aligned} \quad (\text{A.14})$$

where  $g_i(\mathbf{x}, t)$  has been subtracted and  $\tau_g^* = \tau_g / \Delta t$ . Matching terms of 1st and 2nd orders in  $\epsilon$  in this equation yields,

$$(\partial_{t_0} + e_{i\alpha} \partial_\alpha) g_i^{(\text{eq})} = -\frac{1}{\tau_g^*} g_i^{(1)} \quad (\text{A.15})$$

$$\partial_{t_1} g_i^{(\text{eq})} + (\partial_{t_0} + e_{i\alpha} \partial_\alpha) g_i^{(1)} + \frac{1}{2} (\partial_{t_0} + e_{i\alpha} \partial_\alpha) (\partial_{t_0} + e_{i\beta} \partial_\beta) g_i^{(\text{eq})} = -\frac{1}{\tau_g^*} g_i^{(2)}. \quad (\text{A.16})$$

respectively. Substituting Eq. (A.15) into Eq. (A.16) yields,

$$\partial_{t_1} g_i^{(\text{eq})} + \frac{2\tau_g^* - 1}{2\tau_g^*} (\partial_{t_0} + e_{i\alpha} \partial_\alpha) g_i^{(1)} = -\frac{1}{\tau_g^*} g_i^{(2)}. \quad (\text{A.17})$$

According to the definitions of the zeroth- and first-order moments of  $g_i$  in Eqs. (A.7) and (A.8), we get for  $k = 1, 2, \dots$ ,

$$\sum_{i=0}^b g_i^{(k)} = 0 \quad \text{and} \quad \sum_{i=0}^b g_i^{(k)} e_{i\alpha} = 0. \quad (\text{A.18})$$

Now taking zeroth order moment of Eq. (A.15),

$$\sum_{i=0}^b (\partial_{t_0} + e_{i\alpha} \partial_\alpha) g_i^{(\text{eq})} = -\frac{1}{\tau_g^*} \sum_{i=0}^b g_i^{(1)} \quad (\text{A.19})$$

with the definition of  $g_i^{(\text{eq})}$  in Eq. (A.7), yields,

$$\partial_\alpha u_\alpha^* = 0, \quad (\text{A.20})$$

which is the incompressible mass conservation equation (2.2) (i.e., continuity equation).

Taking the first-order moment of Eq. (A.15) yields,

$$\sum_{i=0}^b e_{i\alpha} (\partial_{t_0} + e_{i\beta} \partial_{\beta}) g_i^{(eq)} = -\frac{1}{\tau_g^*} \sum_{i=0}^b e_{i\alpha} g_i^{(1)} \quad (\text{A.21})$$

$$\partial_{t_0} \sum_{i=0}^b e_{i\alpha} g_i^{(eq)} + \partial_{\beta} \Pi_{\alpha\beta}^{(0)} = 0. \quad (\text{A.22})$$

Using Eqs. (A.7)–(A.9) we obtain,

$$\partial_{t_0} u_{\alpha}^* + \partial_{\beta} \left\{ \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\gamma}} \frac{\partial \phi}{\partial x_{\gamma}} \right) \delta_{\alpha\beta} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\alpha}} \frac{\partial \phi}{\partial x_{\beta}} + u_{\alpha}^* u_{\beta}^* + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \left( \frac{\partial u_{\alpha}}{\partial x_{\beta}} + \frac{\partial u_{\beta}}{\partial x_{\alpha}} \right) \Delta t \right\} = 0. \quad (\text{A.23})$$

Additionally, the first-order moment of Eq. (A.17) reads,

$$\partial_{t_1} u_{\alpha}^* + \frac{2\tau_g^* - 1}{2\tau_g^*} \partial_{\beta} \Pi_{\alpha\beta}^{(1)} = 0, \quad (\text{A.24})$$

with the definition,

$$\Pi_{\alpha\beta}^{(1)} = \sum_{i=0}^b g_i^{(1)} e_{i\alpha} e_{i\beta}. \quad (\text{A.25})$$

Substituting  $g_i^{(1)}$  from Eq. (A.15) into Eq. (A.25), and using Eqs. (A.10) and (A.9), we find,

$$\Pi_{\alpha\beta}^{(1)} = -\tau_g^* \sum_{i=0}^b e_{i\alpha} e_{i\beta} (\partial_{t_0} + e_{i\gamma} \partial_{\gamma}) g_i^{(eq)} = -\tau_g^* \left\{ \partial_{t_0} \sum_{i=0}^b e_{i\alpha} e_{i\beta} g_i^{(eq)} + \partial_{\gamma} \sum_{i=0}^b e_{i\alpha} e_{i\beta} e_{i\gamma} g_i^{(eq)} \right\} \quad (\text{A.26})$$

$$\Pi_{\alpha\beta}^{(1)} = -\tau_g^* \left\{ \partial_{t_0} \Pi_{\alpha\beta}^{(0)} + \partial_{\gamma} c_s^2 (\delta_{\alpha\beta} u_{\gamma}^* + \delta_{\alpha\gamma} u_{\beta}^* + \delta_{\gamma\beta} u_{\alpha}^*) \right\} \quad (\text{A.27})$$

then,

$$\begin{aligned} \Pi_{\alpha\beta}^{(1)} = & -\tau_g^* \left\{ \partial_{t_0} \left[ \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\gamma}} \frac{\partial \phi}{\partial x_{\gamma}} \right) \delta_{\alpha\beta} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\alpha}} \frac{\partial \phi}{\partial x_{\beta}} + u_{\alpha}^* u_{\beta}^* + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \left( \frac{\partial u_{\alpha}}{\partial x_{\beta}} + \frac{\partial u_{\beta}}{\partial x_{\alpha}} \right) \Delta t \right] \right. \\ & \left. + \partial_{\gamma} c_s^2 (\delta_{\alpha\beta} u_{\gamma}^* + \delta_{\alpha\gamma} u_{\beta}^* + \delta_{\gamma\beta} u_{\alpha}^*) \right\} \end{aligned} \quad (\text{A.28})$$

which can be simplified to,

$$\begin{aligned} \Pi_{\alpha\beta}^{(1)} = & -\tau_g^* \left\{ \partial_{t_0} \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\gamma}} \frac{\partial \phi}{\partial x_{\gamma}} \delta_{\alpha\beta} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\alpha}} \frac{\partial \phi}{\partial x_{\beta}} + u_{\alpha}^* u_{\beta}^* + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \left( \frac{\partial u_{\alpha}}{\partial x_{\beta}} + \frac{\partial u_{\beta}}{\partial x_{\alpha}} \right) \Delta t \right) \right. \\ & \left. + c_s^2 (\partial_{\alpha} u_{\beta}^* + \partial_{\beta} u_{\alpha}^*) \right\} \end{aligned} \quad (\text{A.29})$$

where continuity equation (2.2) has been used in the last term. Using chain rule in the term  $\partial_{t_0}(u_{\alpha}^* u_{\beta}^*)$  yields,

$$\partial_{t_0}(u_{\alpha}^* u_{\beta}^*) = u_{\alpha}^* \partial_{t_0}(u_{\beta}^*) + u_{\beta}^* \partial_{t_0}(u_{\alpha}^*) \quad (\text{A.30})$$

with, from Eq. (A.22),

$$\partial_{t_0}(u_{\beta}^*) = -\partial_{\gamma}(\Pi_{\beta\gamma}^0) \quad (\text{A.31})$$

$$\partial_{t_0}(u_{\alpha}^*) = -\partial_{\gamma}(\Pi_{\alpha\gamma}^0) \quad (\text{A.32})$$

yielding after substitution,

$$\partial_{t_0}(u_{\alpha}^* u_{\beta}^*) = -u_{\alpha}^* \partial_{\gamma}(\Pi_{\beta\gamma}^0) - u_{\beta}^* \partial_{\gamma}(\Pi_{\alpha\gamma}^0). \quad (\text{A.33})$$

Therefore,

$$\begin{aligned} \partial_{t_0}(u_{\alpha}^* u_{\beta}^*) = & -u_{\alpha}^* \partial_{\gamma} \left\{ \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\zeta}} \frac{\partial \phi}{\partial x_{\zeta}} \right) \delta_{\beta\gamma} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\beta}} \frac{\partial \phi}{\partial x_{\gamma}} + u_{\beta}^* u_{\gamma}^* + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \Delta t \left( \frac{\partial u_{\beta}}{\partial x_{\gamma}} + \frac{\partial u_{\gamma}}{\partial x_{\beta}} \right) \right\} \\ & - u_{\beta}^* \partial_{\gamma} \left\{ \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\zeta}} \frac{\partial \phi}{\partial x_{\zeta}} \right) \delta_{\alpha\gamma} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_{\alpha}} \frac{\partial \phi}{\partial x_{\gamma}} + u_{\alpha}^* u_{\gamma}^* + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \Delta t \left( \frac{\partial u_{\alpha}}{\partial x_{\gamma}} + \frac{\partial u_{\gamma}}{\partial x_{\alpha}} \right) \right\} \end{aligned} \quad (\text{A.34})$$



then,

$$\begin{aligned} \partial t_0(u_\alpha^* u_\beta^*) &= -u_\alpha^* \partial_\gamma (u_\beta^* u_\gamma^*) - u_\beta^* \partial_\gamma (u_\alpha^* u_\gamma^*) - u_\alpha^* \partial_\gamma \left\{ \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_\zeta} \frac{\partial \phi}{\partial x_\zeta} \right) \delta_{\beta\gamma} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_\beta} \frac{\partial \phi}{\partial x_\gamma} \right. \\ &\quad \left. + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \Delta t \left( \frac{\partial u_\beta}{\partial x_\gamma} + \frac{\partial u_\gamma}{\partial x_\beta} \right) \right\} - u_\beta^* \partial_\gamma \left\{ \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_\zeta} \frac{\partial \phi}{\partial x_\zeta} \right) \delta_{\alpha\gamma} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\gamma} \right. \\ &\quad \left. + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \Delta t \left( \frac{\partial u_\alpha}{\partial x_\gamma} + \frac{\partial u_\gamma}{\partial x_\alpha} \right) \right\}. \end{aligned} \quad (\text{A.35})$$

The first term in the above equation can be replaced by  $-\partial_\gamma (u_\beta^* u_\alpha^* u_\gamma^*)$ . Now, replacing  $\partial t_0(u_\alpha^* u_\beta^*)$  from the above equation into Eq. (A.36) results in,

$$\begin{aligned} \Pi_{\alpha\beta}^{(1)} &= -\tau_g^* \left\{ \partial_{t_0} \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \delta_{\alpha\beta} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\beta} + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \Delta t \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right) \right) - \partial_\gamma (u_\beta^* u_\alpha^* u_\gamma^*) \right. \\ &\quad \left. - u_\alpha^* \partial_\gamma \left( \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_\zeta} \frac{\partial \phi}{\partial x_\zeta} \right) \delta_{\beta\gamma} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_\beta} \frac{\partial \phi}{\partial x_\gamma} + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \Delta t \left( \frac{\partial u_\beta}{\partial x_\gamma} + \frac{\partial u_\gamma}{\partial x_\beta} \right) \right) - u_\beta^* \partial_\gamma \right. \\ &\quad \left. \times \left( \left( \frac{k}{\rho} \frac{\partial \phi}{\partial x_\zeta} \frac{\partial \phi}{\partial x_\zeta} \right) \delta_{\alpha\gamma} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\gamma} + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \Delta t \left( \frac{\partial u_\alpha}{\partial x_\gamma} + \frac{\partial u_\gamma}{\partial x_\alpha} \right) \right) + c_s^2 (\partial_\alpha u_\beta^* + \partial_\beta u_\alpha^*) \right\}. \end{aligned} \quad (\text{A.36})$$

With this definition, Eq. (A.24) becomes,

$$\begin{aligned} \partial_{t_1} u_\alpha^* - \partial_\beta \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right) (\partial_\alpha u_\beta^* + \partial_\beta u_\alpha^*) \right\} &+ \partial_\beta \left\{ \left( \tau_g^* - \frac{1}{2} \right) \partial_\gamma (u_\alpha^* u_\beta^* u_\gamma^*) \right\} \\ &- \partial_\alpha \partial_{t_0} \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \right\} + \partial_\beta \partial_{t_0} \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\beta} \right) \right\} \\ &+ \partial_\beta \partial_{t_0} \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right)^2 \Delta t \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right) \right\} + u_\alpha^* \partial_\beta^2 \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \right\} \\ &- u_\alpha^* \partial_\beta \partial_\gamma \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\beta} \frac{\partial \phi}{\partial x_\gamma} \right) \right\} + u_\alpha^* \partial_\beta \partial_\gamma \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right)^2 \Delta t \left( \frac{\partial u_\gamma}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\gamma} \right) \right\} \\ &+ u_\beta^* \partial_\beta \partial_\alpha \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \right\} - u_\beta^* \partial_\beta \partial_\gamma \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\gamma} \right) \right\} \\ &+ u_\beta^* \partial_\beta \partial_\gamma \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right)^2 \Delta t \left( \frac{\partial u_\gamma}{\partial x_\alpha} + \frac{\partial u_\alpha}{\partial x_\gamma} \right) \right\} = 0. \end{aligned} \quad (\text{A.37})$$

To recover the pressureless NS momentum equation, we multiply Eq. (A.37) by  $\epsilon = \Delta t$  and add it to Eq. (A.23), yielding,

$$\begin{aligned} \partial_t u_\alpha^* + \partial_\beta \left\{ \left( c_s^2 + \frac{k}{\rho} \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \delta_{\alpha\beta} - \frac{k}{\rho} \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\beta} + u_\alpha^* u_\beta^* \right. \\ &\quad \left. + c_s^2 \left( \tau_g^* - \frac{1}{2} \right) \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right) \Delta t \right\} - \partial_\beta \left\{ c_s^2 \Delta t \left( \tau_g^* - \frac{1}{2} \right) (\partial_\alpha u_\beta^* + \partial_\beta u_\alpha^*) \right\} \\ &+ \partial_\beta \left\{ \left( \tau_g^* - \frac{1}{2} \right) \partial_\gamma (u_\alpha^* u_\beta^* u_\gamma^*) \Delta t \right\} - \partial_\alpha \partial_{t_0} \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\} \\ &+ \partial_\beta \partial_{t_0} \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\beta} \right) \Delta t \right\} + \partial_\beta \partial_{t_0} \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right)^2 \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right) \Delta t^2 \right\} \\ &+ u_\alpha^* \partial_\beta^2 \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\} - u_\alpha^* \partial_\beta \partial_\gamma \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\beta} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\} \\ &+ u_\alpha^* \partial_\beta \partial_\gamma \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right)^2 \left( \frac{\partial u_\gamma}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\gamma} \right) \Delta t^2 \right\} + u_\beta^* \partial_\beta \partial_\alpha \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\} \\ &- u_\beta^* \partial_\beta \partial_\gamma \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\} + u_\beta^* \partial_\beta \partial_\gamma \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right)^2 \left( \frac{\partial u_\gamma}{\partial x_\alpha} + \frac{\partial u_\alpha}{\partial x_\gamma} \right) \Delta t^2 \right\} = 0. \end{aligned} \quad (\text{A.38})$$

In the latter equation, the terms including  $\partial_\beta \left\{ c_s^2 \Delta t \left( \tau_g^* - \frac{1}{2} \right) \left( \partial_\alpha u_\beta^* + \partial_\beta u_\alpha^* \right) \right\}$  cancel each other out. The last term  $\left( u_\alpha^* u_\beta^* u_\gamma^* \right)$  is  $\mathcal{O}(\text{Ma}^3)$ , in Mach number and, hence, can be neglected [32]. The terms  $\partial_\alpha \partial_{t_0} \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\}$ ,  $\partial_\beta \partial_{t_0} \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\beta} \right) \Delta t \right\}$ ,  $u_\alpha^* \partial_\beta^2 \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\}$ ,  $u_\alpha^* \partial_\beta \partial_\gamma \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\beta} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\}$ ,  $u_\beta^* \partial_\beta \partial_\alpha \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\}$  and  $u_\beta^* \partial_\beta \partial_\gamma \left\{ \left( \tau_g^* - \frac{1}{2} \right) \frac{k}{\rho} \left( \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\gamma} \right) \Delta t \right\}$  contain the small parameter  $k$  multiplying  $\Delta t$  and thus can be neglected. Finally,  $\partial_\beta \partial_{t_0} \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right)^2 \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right) \Delta t^2 \right\}$ ,  $u_\alpha^* \partial_\beta \partial_\gamma \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right)^2 \left( \frac{\partial u_\gamma}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\gamma} \right) \Delta t^2 \right\}$  and  $u_\beta^* \partial_\beta \partial_\gamma \left\{ c_s^2 \left( \tau_g^* - \frac{1}{2} \right)^2 \left( \frac{\partial u_\gamma}{\partial x_\alpha} + \frac{\partial u_\alpha}{\partial x_\gamma} \right) \Delta t^2 \right\}$  are on the order of  $\Delta t^2$  and can be eliminated.

After accounting for the above simplifications, Eq. (A.38) becomes,

$$\partial_t u_\alpha^* + \partial_\beta (u_\alpha^* u_\beta^*) = \frac{k}{\rho} \partial_\beta \left\{ \frac{\partial \phi}{\partial x_\gamma} \frac{\partial \phi}{\partial x_\gamma} \delta_{\alpha\beta} - \frac{\partial \phi}{\partial x_\alpha} \frac{\partial \phi}{\partial x_\beta} \right\} \quad (\text{A.39})$$

which is identical to Eq. (2.3) without viscous forces, gravitational force and pressure gradient. The effects of viscous and gravitational forces are added as body forces to the LB scheme as proposed in [18].

## Appendix B. Chapman–Enskog expansion for the Cahn–Hilliard equation

By applying a procedure similar to that in Appendix A to Eq. (2.22) we obtain,

$$\epsilon (\partial_{t_0} + e_{i\alpha} \partial_\alpha) f_i^{(\text{eq})} = -\frac{1}{\tau_f^*} \epsilon f_i^{(1)}, \quad (\text{B.1})$$

with  $\tau_f^* = \tau_f / \Delta t$ , and,

$$\partial_{t_1} f_i^{(\text{eq})} - \left( \tau_f^* - \frac{1}{2} \right) (\partial_{t_0} + e_{i\alpha} \partial_\alpha) (\partial_{t_0} + e_{i\beta} \partial_\beta) f_i^{(\text{eq})} = -\frac{1}{\tau_f^*} f_i^{(2)}, \quad (\text{B.2})$$

on the  $\mathcal{O}(\epsilon)$  scale and the  $\mathcal{O}(\epsilon^2)$  scale, respectively.

From the definitions of  $f_i$  and  $f_i^{(\text{eq})}$  we find,

$$\sum_{i=0}^b f_i^{(k)} = 0 \quad \text{and} \quad \sum_{i=0}^b f_i^{(k)} e_\alpha = 0 \quad (k = 1, 2, \dots). \quad (\text{B.3})$$

And the moments of  $f_i^{\text{eq}}$  are,

$$\sum_{i=0}^b f_i = \phi, \quad (\text{B.4})$$

$$\sum_{i=0}^b f_i e_{i\alpha} = \phi u_\alpha, \quad (\text{B.5})$$

$$\sum_{i=0}^b f_i e_{i\alpha} e_{i\beta} = \frac{M \mu_\phi \delta_{\alpha\beta}}{\tau_f^* - \frac{1}{2} \Delta t} + \phi u_\alpha u_\beta. \quad (\text{B.6})$$

Taking the zeroth-order moments of Eqs. (B.1) and (B.2) by using Eqs. (B.4)–(B.6), yields,

$$\partial_{t_0} \phi + \partial_\alpha (\phi u_\alpha) = 0, \quad (\text{B.7})$$

and,

$$\begin{aligned} \partial_{t_1} \phi - \left( \tau_f^* - \frac{1}{2} \right) \left\{ \partial_{t_0} \partial_{t_0} \phi + \partial_\alpha \partial_{t_0} (\phi u_\alpha) \right\} - \left( \tau_f^* - \frac{1}{2} \right) \\ \times \left\{ \partial_\beta \partial_{t_0} (\phi u_\beta) + \partial_\alpha \partial_\beta \left( \frac{M}{\tau_f^* - \frac{1}{2}} \mu_\phi \delta_{\alpha\beta} + \phi u_\alpha u_\beta \right) \right\} = 0. \end{aligned} \quad (\text{B.8})$$

From Eq. (B.7) we have,

$$\partial_{t_0} \partial_{t_0} \phi + \partial_\alpha \partial_{t_0} (\phi u_\alpha) = \partial_{t_0} (\partial_{t_0} \phi + \partial_\alpha (\phi u_\alpha)) = 0, \quad (\text{B.9})$$

and simplify Eq. (B.8) into,

$$\partial_{t_1} \phi = \left( \tau_f^* - \frac{1}{2} \right) \left\{ \partial_\beta \partial_{t_0} (\phi u_\beta) + \partial_\alpha \partial_\beta \left( \frac{M}{\tau_f^* - \frac{1}{2}} \mu_\phi \delta_{\alpha\beta} + \phi u_\alpha u_\beta \right) \right\}, \quad (\text{B.10})$$

or

$$\partial_{t_1} \phi = \left( \tau_f^* - \frac{1}{2} \right) \partial_\beta \left\{ \partial_{t_0} (\phi u_\beta) + \partial_\alpha (\phi u_\alpha u_\beta) \right\} + \partial_\alpha \partial_\alpha M \mu_\phi. \quad (\text{B.11})$$

Multiplying Eq. (B.11) by  $\Delta t$  and adding it to Eq. (B.7), we get,

$$\partial_t \phi + \partial_\alpha (\phi u_\alpha) = \Delta t M \nabla^2 \mu_\phi + \Delta t \left( \tau_f^* - \frac{1}{2} \right) \partial_\beta \left\{ \partial_{t_0} (\phi u_\beta) + \partial_\alpha (\phi u_\alpha u_\beta) \right\}. \quad (\text{B.12})$$

For low Mach number flows, the last term can be neglected, since it has a small magnitude and is multiplied by the small parameter  $\Delta t$ , so that we finally get,

$$\partial_t \phi + \partial_\alpha (\phi u_\alpha) = M \nabla^2 \mu_\phi, \quad (\text{B.13})$$

which corresponds to the Cahn–Hilliard equation with diffusivity coefficient (Mobility)  $M$ .

## References

- [1] D. Yu, R. Mei, L. Luo, W. Shyy, Viscous flow computations with the method of lattice Boltzmann equation, *Prog. Aerosp. Sci.* 39 (2003) 329–367.
- [2] J. Tölke, M. Krafczyk, Implementation of a lattice Boltzmann kernel using the compute unified device architecture developed by nVIDIA, *Comput. Vis. Sci.* 1 (2008) 29–39.
- [3] J. Tölke, M. Krafczyk, TeraFLOP computing on a desktop PC with GPUs for 3D CFD, *Int. J. Comput. Fluid Dyn.* 22 (2008) 443–456.
- [4] C. Janßen, M. Krafczyk, Free surface flow simulations on GPGPUs using LBM, *Comput. Math. Appl.* 61 (12) (2011) 3549–3563.
- [5] A. Banari, C. Janßen, S.T. Grilli, M. Krafczyk, Efficient GPGPU implementation of a lattice Boltzmann model for multiphase flows with high density ratios, *Comput. & Fluids* 93 (2014) 1–17.
- [6] S. Chapman, T. Cowling, *The Mathematical Theory of Nonuniform Gases*, Cambridge University Press, 1970.
- [7] C. Janßen, S.T. Grilli, M. Krafczyk, On enhanced non-linear free surface flow simulations with a hybrid LBM–VOF model, *Comput. Math. Appl.* 65 (2) (2013) 211–229.
- [8] T. Lee, C. Lin, A stable discretization of the lattice Boltzmann equation for simulation of incompressible two-phase flows at high density ratio, *J. Comput. Phys.* 206 (2005) 16–47.
- [9] T. Inamuro, T. Ogata, S. Tajima, N. Konishi, A lattice Boltzmann method for incompressible two-phase flows with large density differences, *J. Comput. Phys.* 198 (2004) 628–644.
- [10] X. Shan, H. Chen, Lattice Boltzmann model for simulating flows with multiple phases and components, *Phys. Rev. E* 47 (1993) 1815–1819.
- [11] T. Lee, Effects of incompressibility on the elimination of parasitic currents in lattice Boltzmann method for binary fluids, *Comput. Math.* 58 (2009) 987–994.
- [12] H.W. Zheng, C. Shu, Y.T. Chew, A lattice Boltzmann model for multiphase flows with large density ratio, *J. Comput. Phys.* 218 (2006) 353–371.
- [13] A. Fakhari, M.H. Rahimian, Phase-field modelling by the method of lattice Boltzmann equations, *Phys. Rev. E* (2010) 036707.
- [14] J.Y. Shao, C. Shu, H.B. Huang, Y.T. Chew, Free-energy-based lattice Boltzmann model for the simulation of multiphase flows with density contrast, *Phys. Rev. E* 89 (2014) 033309. 1–13.
- [15] X. He, S. Chen, R. Zhang, A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability, *J. Comput. Phys.* 152 (1999) 642–663.
- [16] J. Cahn, J. Hilliard, Free energy of a nonuniform system, *J. Chem. Phys.* 28 (1958) 258–267.
- [17] Y. Qian, D. d’Humières, P. Lallemand, Lattice BGK models for Navier–Stokes equation, *Europhys. Lett.* 17 (1992) 479–484.
- [18] Z. Guo, C. Zhen, B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, *Phys. Rev. E* 65 (4) (2002) 046308.1–046308.6.
- [19] M. Swift, W. Osborn, J. Yeomans, Lattice Boltzmann simulation of liquid–gas and binary fluid systems, *Phys. Rev. E* 54 (1996) 5041–5052.
- [20] A.M. Worthington, *A Study of Splashes*, Longmans Green, London, 1908.
- [21] D.A. Weiss, A.L. Yarin, Single drop impact on liquid film: neck distortion, jetting, tiny bubble entrainment, and crown formation, *J. Fluid Mech.* 385 (1999) 229.
- [22] R. Rioboo, C. Bauthier, J. Conti, M. Voue, J. De Coninck, Experimental investigation of splash and crown formation during single drop impact on wetted surfaces, *Exp. Fluids* 35 (2003) 648–652.
- [23] R. Clift, J. Grace, M. Weber, *Bubbles, Drops, and Particles*, Academic Press, 1978.
- [24] D. Bhaga, M. Weber, Bubbles in viscous liquids: shapes, wakes and velocities, *J. Fluid Mech.* 105 (1981) 61–85.
- [25] L. Amaya-Bower, T. Lee, Single bubble rising dynamics for moderate Reynolds number using lattice Boltzmann method, *Comput. & Fluids* 39 (2010) 1191–1207.
- [26] J. Hua, J.F. Stene, P. Lin, Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method, *J. Comput. Phys.* 227 (2008) 3358–3382.
- [27] M.S. Longuet-Higgins, E.D. Cokelet, The deformation of steep surface waves on water: a numerical method of computation, *Proc. R. Soc. A* 350 (1976) 1–26.
- [28] S.T. Grilli, J. Skourup, I.A. Svendsen, An efficient boundary element method for nonlinear water waves, *Eng. Anal. Bound. Elem.* 6 (1989) 97–107.
- [29] P. Lubin, S. Vincent, S. Abadie, J. Caltagirone, Three-dimensional Large Eddy Simulation of air entrainment under plunging breaking waves, *Coastal Eng.* 53 (2006) 631–655.
- [30] M. Krafczyk, J. Tölke, L.S. Luo, Large-Eddy Simulations with a multiple-relaxation-time LBE model, *Internat. J. Modern Phys. B* 17 (01–02) (2003) 33–39.
- [31] P. Lubin, S. Glockner, Detailed numerical investigation of the three-dimensional flow structures under breaking waves, in: *Proc. 7th Conf. on Coast. Dynam.*, Arcachon, France, 2003, pp. 1127–1136.
- [32] X. He, L.S. Luo, Lattice Boltzmann model for the incompressible Navier–Stokes equation, *J. Stat. Phys.* 88 (1997) 927–944.