Proceedings of the ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering
OMAE2013
June 9-14, 2013, Nantes, France

OMAE2013-10102

TWO PHASE FLOW SIMULATION WITH LATTICE BOLTZMANN METHOD: APPLICATION TO WAVE BREAKING

Amir Banari*
Department of ocean Engineering
University of Rhode Island
Narragansett, Rhode Island
Email: amir_ banari@my.uri.edu

Stephan T. Grilli
Department of ocean Engineering
University of Rhode Island
Narragansett, Rhode Island
Email: Grilli@oce.uri.edu

Christian F. Janssen
Inst. M-8, Fluid Dynamics and Ship Theory,
Hamburg University of Technology
Germany
Email: mail@christian-janssen.de

ABSTRACT

A new Lattice Boltzmann method (LBM) is developed to efficiently simulate multiphase flows with high density ratios, in order to study complex air-sea interaction problems, such as wind wave breaking and related sea-spray generation. In this method, which builds and improves on the method proposed earlier by [1], the motion of (diffusive) interfaces between fluids is modeled by solving the convective Cahn-Hilliard equation with the LBM. As in the latter work, we eliminate instabilities resulting from high density ratios by solving an additional Poisson equation for the fluid pressure. The resulting numerical scheme is computationally demanding since this equation must be solved over the entire computational domain, which motivates implementing the method on the massively parallel environment offered by General Purpose Graphical Processing Units (GPGPU), via the nVIDIA CUDA framework. In this paper, we present the equations and numerical methods for the method and the initial validation of the resulting multiphase-LBM for standard benchmark problems such as Poiseuille flow, a rising bubble, and Rayleigh-Taylor instability for two-fluid systems. A good agreement with the reference solutions is achieved in all cases. Finally, the method is applied to simulating an ocean breaking wave in a space periodic domain. In all the presented applications, it is observed that the GPGPU implementation leads to speed-ups of about two orders of magnitude in comparison to a single-core CPU implementation. Although the method is only currently implemented in a two-dimensional (2D) framework, its extension to three-dimensions (3D) should be straightforward, but the need for the efficient GPGPU implementation will become even more drastic in 3D.

1 INTRODUCTION

The numerical simulation of multiphase and multicomponent fluid flows is a challenging task in Computational Fluid Dynamics (CFD), both for conventional macroscopic and mesoscopic methods, such as the Lattice-Boltzmann Method (LBM). In classical CFD methods, multiphase flows are simulated by coupling a Navier-Stokes (NS) equation solver to an interface advection or advection-diffusion equation (e.g., [2]). The former equation is used in combination with a sharp interface model, whereas the latter is mostly used with a diffusive interface model. The interface itself is typically represented by a tracking method (such as the widely used Volume Of Fluid (VOF) method), or an interface capturing method. Most of the interface tracking methods assume a sharp interface, i.e., they consider the phase transition to be clearly defined and thus the interface between two fluids to be infinitely thin. By contrast, the interface capturing methods allow for both sharp or diffusive interface representations, depending on the type of equations solved. An additional challenge, when using

*Address all correspondence to this author.

a sharp interface method, is the accurate computation of the interface curvature and related surface tension forces. This has encouraged many researchers to use diffusive interface methods, in which interface forces can be modeled as a smoothed continuum by distributing them over thin but numerically resolvable layers [3]. Such models have recently attracted much interest, owing to their computational advantages [4, 5]. Because of these various options, when developing and implementing a specific free surface or multiphase CFD model, one has to make a priori decisions regarding using: (i) a sharp or diffusive interface method; (ii) an advection or advection-diffusion equation; and (iii) a tracking or capturing method. For instance, in the two-phase model detailed below, we decided to model the interface motion with Cahn-Hilliard (CH) interface capturing advection-diffusion equation [6], using a scalar order parameter Φ_α ($\alpha = 1, 2$) to identify each phase. The interface between the two phases is then defined as a smooth transition from one to the other value of Φ_α.

Recently, the LBM has matured into a powerful alternative to classical NS solvers, both for simulating single phase, and multiphase/multi-component flows [7, 8, 9]. The LBM is based on the Boltzmann equation, which governs the dynamics of molecular probability distribution functions, from a microscopic scale point of view. The LBM then discretizes the Boltzmann equation on an equidistant lattice, yielding a numerical method for computing macroscopic distribution functions on the lattice. The macroscopic hydrodynamic quantities, such as pressure and velocity, are obtained as low-order moments of these distribution functions. The resulting formulation can be shown to converge towards the solution of the classical governing macroscopic equations [10], such as NS. LBM, however, has several solver-specific advantages, such as a relatively easier implementation, a straightforward treatment of boundary conditions, and data and operator locality, which both yield significantly more efficient parallel implementations than for the more traditional solvers. In this respect, LBM has been shown to take full advantage of the recent advances in General Purpose Graphical Processing Units (GPGPU) [11].

While there have been numerous applications of classical CFD solvers to multiphase flows, whose exhaustive review is outside the scope of this paper, over the past two decades, several noteworthy methods have been developed for simulating multiphase flows in the context of the LBM. These are: [12] color method, the [8] model (SC), the free energy method of [7], and the method of [9].

All of the above methods solved multiphase flows using various approaches but, in all of these, the maximum fluid density ratio achievable in computations was limited by the occurrence of instabilities for high ratio values (typically larger than 10-20). Overcoming this limitation is one of the most challenging current issues in the LBM modeling of mulltiphase flows and the subject of active research. This is also the rationale for developing the method presented in this paper, as we aim at modeling complex flows at an air-water interface, whose density ratio is about 1,000.

To this effect, we recently developed a new LBM, based on improving the approach proposed by [1]. As in the latter work, we eliminate instabilities resulting from high density ratios by solving an additional Poisson equation for the fluid pressure. In our model, however, new equilibrium functions are introduced to retrieve NS equations, with the molecular viscosity being related to the LBM relaxation time similar to classical LBMs. The new corresponding NS equation, however, does not have the undesired terms, which appear in Inamuro et al.'s model and contribute to triggering instabilities for very high density ratios. The location of diffuse interfaces between two fluids is now tracked more accurately, by solving Cahn-Hilliard equation with the LBM using the new equilibrium functions. Details of the surface tension and body forces, and a detailed derivation of the equations can be found in previously published papers [1, 13].

The resulting numerical scheme is computationally demanding, as the Poisson equation must be (iteratively) solved over the entire computational domain. This is efficiently done by implementing the model on a latest generation GPGPU environment, via the nVIDIA CUDA framework. Such GPGPU boards (e.g., nVIDIA Tesla C2070) provide up to 448 cores, 6 GB of main memory, and a double precision computing capability. It has been shown in various publications [14, 11, 15] that LBM methods are especially well suited for such a GPGPU implementation. For all benchmark problems presented later in this paper, the GPU implementation will lead to speed-ups of about two orders of magnitude in comparison to a single-core CPU implementation.

The paper is organized as follows. First we briefly explain the diffusive interface model on the basis of the free energy concept. Then our new model for simulating multiphase flows with very high density ratio is introduced. Finally, the method is validated by comparing numerical results to reference solutions for two-fluid Poiseuille flows, a rising bubble, and the Rayleigh-Taylor instability. Finally, we solve a case of ocean wave braking in a space-periodic domain.

2 Free energy

As mentioned in the introduction, numerical schemes based on a sharp interface representation lead to additional problems in their numerical implementation, compared to diffusive interface models. In particular, sharp

interface models usually require a moving numerical grid, whereas diffusive interface models naturally accommodate fixed grids (such as in the LBM). They also face difficulties for accurately computing the interface curvature and the related surface tension forces. This often leads to the appearance of "parasitic currents" in the numerical solution along the interface. These problems are eliminated when using a diffuse-interface representation, based on the continuous variation of an order parameter (such as fluid density or a function of density), in a way that is physically consistent with microscopic theories of interface processes. Three main types of diffuse-interface models have thus been proposed in the literature: (i) tracking force models [4]; (ii) continuum surface force models [5]; and (iii) phase-field models [3].

In the current work, we use the latter approach, in which the total free energy $\mathcal{F}$ of a two-fluid system is specified to be minimum for the equilibrium interface profile $\phi(\zeta)$. Here, ϕ denotes a continuously varying order parameter (with values ϕ_1 and ϕ_2 referring to fluid 1 and 2 on either side of the interface, respectively; and $\phi_2 > \phi_1$, $\phi \in [\phi_1, \phi_2]$), and ζ is a coordinate normal to the interface. Following [3], the motion of the diffusive interface is modeled, as a function of the order parameter, with the extended Cahn-Hilliard (CH) equation [6], which includes convection terms,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \boldsymbol{u}) = M \nabla^2 \mu_\phi, \tag{1}$$

where the left hand side represents the interface advection, and the right hand side represents the interface diffusion, expressed as a function of a mobility coefficient M and the chemical potential μ_ϕ,

$$\mu_\phi = \beta \Psi' - k \nabla^2 \phi \tag{2}$$

with the bulk free-energy density $\Psi(\phi) = (\phi - \phi_2)^2 (\phi - \phi_1)^2$. The coefficients k and β are related to the surface tension coefficient σ_{12} and interface thickness W as,

$$W = \frac{4}{\phi_1 - \phi_2} \sqrt{\frac{k}{2\beta}} \tag{3}$$

$$\sigma_{12} = \frac{(\phi_1 - \phi_2)^3}{6} \sqrt{2k\beta} \tag{4}$$

A more detailed derivation of the model can be found in [16].

3 Lattice Boltzmann Model

Two-dimensional multiphase (2D) flows are simulated by solving two sets of equations: (i) the NS equations, which provide the flow fields, based on the conservation of mass and momentum; (ii) the extended Cahn-Hilliard Eq. (1), which describes the interface motion. We solve these equations using a new Lattice Boltzmann method (LBM), based on two sets of Lattice Boltzmann particle distribution functions, one for each equation (i) and (ii), for which we find mesoscopic equilibrium distribution functions, which reproduce the desired macroscopic equations. To discretize the LBM equations, we use the D2Q9 set of particle velocities [17], which introduces 9 discrete particle velocities in directions $\mathbf{e}_i$, with,

$$\begin{aligned}
&\boldsymbol{e}_0 = 0; \\
&\boldsymbol{e}_i = c\left(\cos\left((i-1)\pi/4\right), \sin\left((i-1)\pi/4\right)\right), \quad \text{i=1,3,5,7} \\
&\boldsymbol{e}_i = \sqrt{2}c\left(\cos\left((i-1)\pi/4\right), \sin\left((i-1)\pi/4\right)\right), \quad \text{i=2,4,6,8}
\end{aligned} \tag{5}$$

where $c = \Delta x / \Delta t$ is the propagation speed on the lattice, taken as $c = 1$ in this work, with grid spacing Δx and time step Δt. The speed of sound in the D2Q9 lattice is $c_s = c/\sqrt{3}$ [17], and the weighting factors w_i are given as,

$$w_0 = \frac{4}{9}, w_{1,3,5,7} = \frac{1}{9}, w_{2,4,6,8} = \frac{1}{36} \tag{6}$$

3.1 LBM of Navier-Stokes equations

A set of particle distribution functions $g_i(\boldsymbol{x}, t)$ is specified for calculating the flow velocities and pressure, whose time evolution is computed as (assuming a single relaxation time (SRT) formulation, [18])$(i = 0, ..., 8)$,

$$\begin{aligned}
g_i(\boldsymbol{x} + \boldsymbol{e}_i \Delta t, t + \Delta t) = g_i(\boldsymbol{x}, t) - \frac{\Delta t}{\tau_g}\left(g_i(\boldsymbol{x}, t) - g_i^{(eq)}(\boldsymbol{x}, t)\right) \\
- 3 w_i \Delta t \, e_{i\alpha} \sigma_{\alpha\beta}^{visc} \frac{\partial}{\partial x_\beta}\left(\frac{1}{\rho}\right) + 3 w_i e_{i\alpha} B_\alpha \Delta t
\end{aligned} \tag{7}$$

with particle velocities e_i at point $\boldsymbol{x}$ and the time t, relaxation time τ_g, that relates to fluid viscosity as $\tau_g = \nu / c_s^2 + \frac{1}{2}\Delta t$, viscous stress tensor $\sigma_{\alpha\beta}^{visc} = \mu\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha}\right)$, gravitational body force B_α, density ρ, and velocity u. We denote by $g_i^{(eq)}$, the equilibrium state of particle distribution functions,

$$\begin{aligned}
g_i^{eq} = w_i \left\{1 + \frac{e_{i\alpha} u_\alpha}{c_s^2} + \frac{(e_{i\alpha} u_\alpha)^2}{2c_s^4} - \frac{|\boldsymbol{u}|^2}{2c_s^2}\right\} + w_i \frac{k}{\rho} G_{\alpha\beta} e_{i\alpha} e_{i\beta} \\
- v_i \frac{k}{2\rho} |\nabla \phi|^2
\end{aligned} \tag{8}$$

where implicit summations are performed on repeated indices α and β (but not on i). The last two terms in the above equation express the surface tension forces, where v_i is defined as,

$$v_0 = -\frac{5}{3}, v_i = 3w_i (i = 1, 2, ..., 8) \tag{9}$$

and

$$G_{\alpha\beta}(\phi) = \frac{9}{2}\frac{\partial\phi}{\partial x_\alpha}\frac{\partial\phi}{\partial x_\beta} - \frac{9}{4}\frac{\partial\phi}{\partial x_\gamma}\frac{\partial\phi}{\partial x_\gamma}\delta_{\alpha\beta} \tag{10}$$

Note that, compared to the classical LBM models, the fluid density ρ no longer appears in the equilibrium distribution functions $g_i^{(eq)}$. This eliminates potential instabilities caused by high density differences across interfaces and is equivalent to removing the hydrodynamic pressure gradient from the field. As a result, the velocity resulting from this set of equations is not divergence-free and thus has to be corrected. The predicted velocity $\boldsymbol{u}^*$ is computed as the first order moment of g_i's as,

$$\boldsymbol{u}^* = \sum_{i=0}^{b} g_i e_{i\alpha} \tag{11}$$

and is corrected by $\Delta\boldsymbol{u}$, to satisfy the complete momentum equation. Following [1], we state,

$$\Delta\boldsymbol{u} \simeq -\Delta t \frac{\nabla p}{\rho} \quad ; \quad \boldsymbol{u} = \boldsymbol{u}^* + \Delta\boldsymbol{u} \tag{12}$$

Thus, for the corrected velocity field to satisfy continuity equation $\nabla.\boldsymbol{u} = 0$, $\boldsymbol{u}^*$ must satisfy the following Poisson equation,

$$\nabla.\boldsymbol{u}^* = \nabla.(\frac{\Delta t \nabla p}{\rho}) \tag{13}$$

Note, in practice, the collision time τ_g can be considered as an elementary time of collision, so that Δt can be replaced by τ_g in Eq. (13) (see [19]). The Poisson equation (13) can be discretized by various methods. Here, we also solve it in the LBM framework, which yields the following evolution equation for a new set of particle distribution functions h_i $(i = 0, ..., 8)$,

$$h_i(\boldsymbol{x} + \boldsymbol{e}_i\Delta t, t + \Delta t) = h_i(\boldsymbol{x} + t) - \frac{\Delta t}{\tau_h}(h_i - h_i^{(eq)}) + w_i(\nabla.\boldsymbol{u}^*) \tag{14}$$

New equilibrium distribution functions are defined as well, as,

$$h_i^{eq} = w_i p^n(\boldsymbol{x}, t), \tag{15}$$

where n denotes the n-th iteration in the Poisson equation solution. The relaxation time τ_h is related to that of the NS LBM ansatz,

$$\tau_h = \frac{\tau_g}{c_s^2 \rho} + \frac{1}{2}\Delta t, \tag{16}$$

and the pressure is calculated as the zero-th order moment of the particle distribution functions,

$$p^{n+1} = \sum_{i=0}^{b} h_i. \tag{17}$$

This scheme is iteratively run at a given time t until the pressure field converges.

Hence, in this new method, the two previously derived LBM schemes solve: (i) the (pressureless) NS equations for high density ratios, with surface tension forces included in the equilibrium distribution functions; and (ii) a Poisson equation for the correction of velocity fields to account for pressure gradients. By contrast with sharp interface methods, the calculation of interface curvature is not necessary and only the gradients of the phase field parameter ϕ have to be calculated. Applying the Chapman-Enskog expansion [10] to Eq. 7 with the equilibrium distribution function in Eq. 8, these schemes can be shown to converge to the NS equation [16],

$$\partial_t(u_\alpha) + \partial_\beta\left(u_\alpha u_\beta\right) = \frac{k}{\rho}\partial_\beta\left(\frac{\partial\phi}{\partial x_k}\frac{\partial\phi}{\partial x_k}\delta_{\alpha\beta} - \frac{\partial\phi}{\partial x_\alpha}\frac{\partial\phi}{\partial x_\beta}\right) + \partial_\beta\left[\nu\left(\partial_\alpha u_\beta + \partial_\beta u_\alpha\right)\right] + B_\alpha \tag{18}$$

3.2 Lattice Boltzmann scheme for solving Cahn-Hilliard equation

The diffusive interface motion is modeled by the convective Cahn-Hiliard equation,

$$\partial_t(\phi) + \partial_\alpha\left(\phi u_\alpha\right) = M\nabla^2\mu_\phi \tag{19}$$

To solve this equation, we introduce a third probability distribution function, $f_i(x,t)$, whose evolution is again gov-

erned by a standard LBM scheme,

$$f_i(\boldsymbol{x}+\boldsymbol{e}_i\Delta t, t+\Delta t) = f_i(\boldsymbol{x},t) - \frac{\Delta t}{\tau_f}(f_i(\boldsymbol{x},t) - f_i^{(eq)}(\boldsymbol{x},t)) \tag{20}$$

with corresponding equilibrium distribution functions $f_i^{(eq)}$,

$$f_i^{eq} = H_i\phi + v_i\Gamma\mu_\phi + \phi w_i\left\{\frac{e_{i\alpha}u_\alpha}{c_s^2} + \frac{(e_{i\alpha}u_\alpha)^2}{2c_s^4} - \frac{|\boldsymbol{u}|^2}{2c_s^2}\right\}, \tag{21}$$

where Γ is used to control the monility, w_i and v_i according to (6) and (9),

$$H_0 = 1 \quad \text{and} \quad H_{1,2,\ldots,8} = 0 \tag{22}$$

The order parameter ϕ is computed from the first moment of distribution function,

$$\phi = \sum_{i=0}^{b} f_i \tag{23}$$

It can be shown that, with this definition of $f_i^{(eq)}$, the scheme solves the diffusive Cahn-Hilliard equation, for a relaxation time $\tau_f = \frac{1}{2}\Delta t + \frac{M}{\Gamma}$ [16].

Once the order parameter is computed, the space distribution of fluid properties (density ρ, kinematic viscosity ν and dynamic viscosity μ) can be calculated. These properties are assumed to vary smoothly along the phase interface as a function of the order parameter ϕ,

$$\rho(\phi) = \frac{\phi-\phi_2}{\phi_1-\phi_2}(\rho_1-\rho_2)+\rho_2 \tag{24}$$

$$\nu(\phi) = \frac{\phi-\phi_2}{\phi_1-\phi_2}(\nu_1-\nu_2)+\nu_2 \tag{25}$$

$$\mu(\phi) = \frac{\phi-\phi_2}{\phi_1-\phi_2}(\mu_1-\mu_2)+\mu_2 \tag{26}$$

4 Validation

In this section, the new LBM multiphase method is validated by comparing the numerical results to analytical and experimental reference solutions, for three different test cases: a two-fliuid Poiseuille flows, rising bubbles, and the Rayleigh-Taylor instability. The initial analysis of the multiphase- LBM results shows a good agreement with reference solutions for all three test cases. Finally, the method is applied to simulating a breaking ocean wave, in order to demonstrate the applicability of the method to air-sea interaction problems.

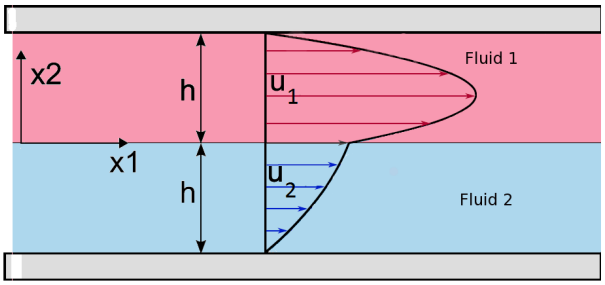


FIGURE 1: A schematic velocity profile of a two-fluid Poiseuille flow between infinite plates

4.1 Two-component Poiseuille flow

The two-fluid Poiseuille flow between two infinite plates is a good test case for validating the method for high viscosity and density ratios. Two immiscible fluids are accelerated by a body force in a rectangular channel and slowed down by viscous drag along the plate surface Fig. 1. At the phase interface, the continuity of fluid velocity and stresses has to be satisfied. Since the interface is planar, its curvature and surface tensions forces are zero. The analytical solution of the Navier-Stokes equations for the velocities u_1 and u_2 in fluid 1 and fluid 2, respectively, is given by,

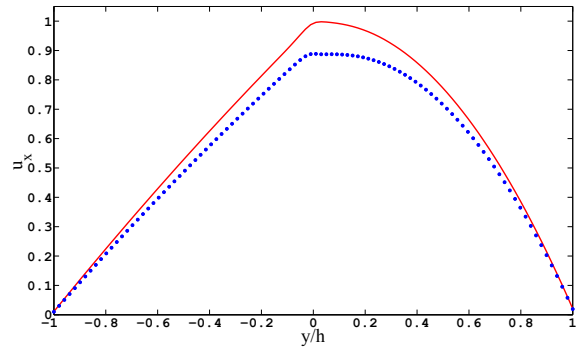
$$u_1 = \frac{g\,h^2\,(\rho_1+\rho_2)}{2\,(\mu_1+\mu_2)} - \frac{x_2\,g\,h\,(\mu_1\,\rho_2-\mu_2\,\rho_1)}{2\,\mu_1\,(\mu_1+\mu_2)} - \frac{g\,\rho_1\,x_2^2}{2\,\mu_1} \tag{27}$$

$$u_2 = \frac{g\,h^2\,(\rho_1+\rho_2)}{2\,(\mu_1+\mu_2)} - \frac{x_2\,g\,h\,(\mu_1\,\rho_2-\mu_2\,\rho_1)}{2\,\mu_2\,(\mu_1+\mu_2)} - \frac{g\,\rho_2\,x_2^2}{2\,\mu_2} \tag{28}$$

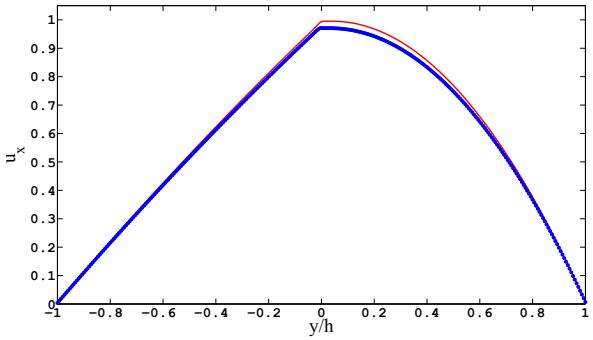
with fluid densities ρ_i, dynamic viscosities μ_i, gravity g and channel height $2h$. The simulations were started from a state of rest, with zero flow velocities in the whole computational domain. A periodic boundary condition in the flow direction and no-slip boundary conditions at the plate surfaces were used. The interface thickness is set to $W = 4$ lattice units. The Reynolds number is $\mathrm{Re} = 100$, based on half of the channel width, the viscosity of the first fluid phase, and the maximum velocity in the channel. The LB Mach number is fixed to $\mathrm{Ma} = 0.01$, and τ_f is set to 1; $\rho_1/\rho_2 = 100$ and the viscosity ratio is $\nu_1/\nu_2 = 0.1$. The simulations were stopped as soon as a steady state was reached. To check the accuracy of the numerical scheme, the following relative L^2-norm error is used,

$$L^2 = \sqrt{\frac{\int_0^{2h}\left[u\,(x_2) - u_{1,2}\,(x_2)\right]^2\,\mathrm{d}x_2}{\int_0^{2h} u_{1,2}^2\,(x_2)\,\mathrm{d}x_2}} \tag{29}$$

where $u_{1,2}$ denotes the analytical solution in fluid 1 or 2, respectively, depending on the value of x_2.



(a) $h = 100$



(b) $h = 300$

FIGURE 2: Non-dimensional velocity profiles ($u_x = u/u_{max}$) for different grid resolutions (Re = 100, $\rho_1/\rho_2 = 100$, $\nu_1/\nu_2 = 0.1$).

Fig. 2 shows steady-state velocity profiles computed for different two grid sizes, as compared to the analytical solution. In previous studies of two-phase Poiseuille flows with LBM multiphase models [20, 21], local oscillations of the fluid velocity near the phase interface were observed, even for low density ratios. Our new LBM multiphase method does not show these oscillations, and accurately reproduces the slope discontinuity in velocity profile. Fig. 3 shows the L^2 errors for different grid configurations; first-order convergence can clearly be observed with decreasing grid size.

4.2 Rising bubble

The transient behavior of bubbles under the influence of gravity has always been a major and demanding test case for validating numerical schemes for two-phase flows. Although the simulation setup in terms of grid initialization and boundary conditions is straightforward, the flow structure simultaneously illustrates the significant effects of

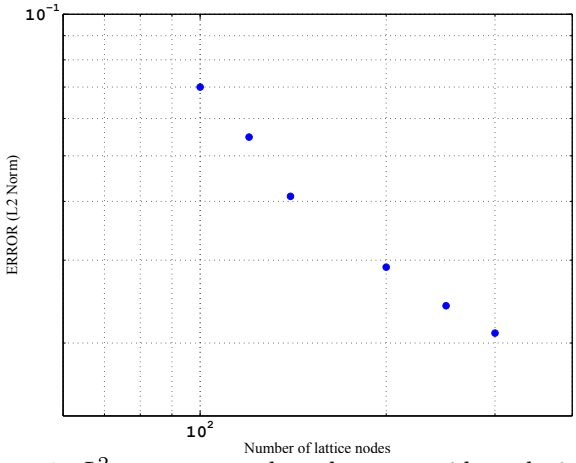


FIGURE 3: L^2-norm error plotted versus grid resolution, in steady-state two-fluid Poiseuille flow

viscosity, buoyancy, and surface tension forces. Several experimental studies on the rising and deformation of single bubbles in a quiescent liquid have been reported in the literature [22,23]. The bubble shapes vary greatly in different flow regimes, as a function of several non-dimensional parameters. Typically the Bond number, the Reynolds number, and the Morton number are used to describe the test case. These are defined as,

$$\mathrm{Bo} = \frac{g\Delta\rho D^2}{\sigma} \tag{30}$$

$$\mathrm{Mo} = \frac{g\mu_1^4}{\sigma^3\rho_1}\left(1 - \frac{\rho_2}{\rho_1}\right) \tag{31}$$

$$\mathrm{Re} = \frac{U_t D}{\nu_1} \tag{32}$$

where D is the bubble diameter, $\Delta\rho = \rho_1 - \rho_2$ the density difference between the two fluids, ρ_1 the density of the heavier fluid, ρ_2 the density of the lighter fluid, g the gravitational acceleration, U_t the terminal velocity of the bubble, and ν_1 and μ_1 are kinematic and dynamic viscosity, respectively, of the heavier fluid.

A sphere with a diameter of $D = 60$ lattice nodes is placed in a computational domain of 256×1024 lattice nodes. Periodic boundary conditions are used on the lateral sides of the domain, a bounce-back scheme boundary condition is applied as on the top and bottom of the domain. The remaining flow parameters are set to $\rho_1 = 6000$ and $\rho_1/\rho_2 = 1000$, $\mu_1/\mu_2 = 1000$, $\tau_f = 1$, $\phi_1 = 0.4$, $\phi_2 = 0.1$. We run the simulation for three different test cases (a) - (c), with different Moand Bo; the specific simulation parameters are given in Table 1. Fig. 4 shows the computed terminal shape of the bubble and velocity fields, compared

TABLE 1: Simulation parameters for the rising bubble test case

Case	ν_1	k	β	g	Bo	Mo
(a)	2×10^{-3}	0.266	1.481	1.85×10^{-11}	0.1	10^{-3}
(b)	2×10^{-3}	0.266	1.481	1.85×10^{-9}	10	10^{-1}
(c)	1×10^{-2}	0.210	1.171	1.46×10^{-8}	100	10^{3}

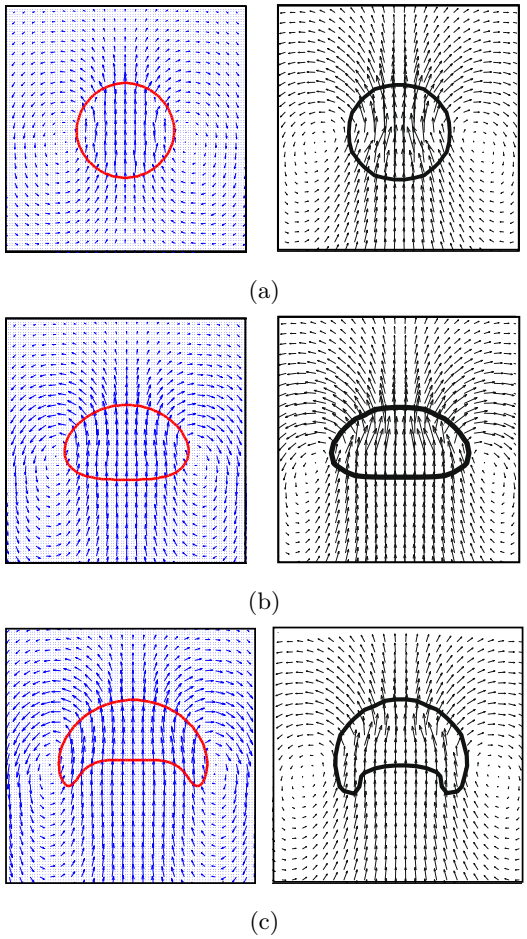
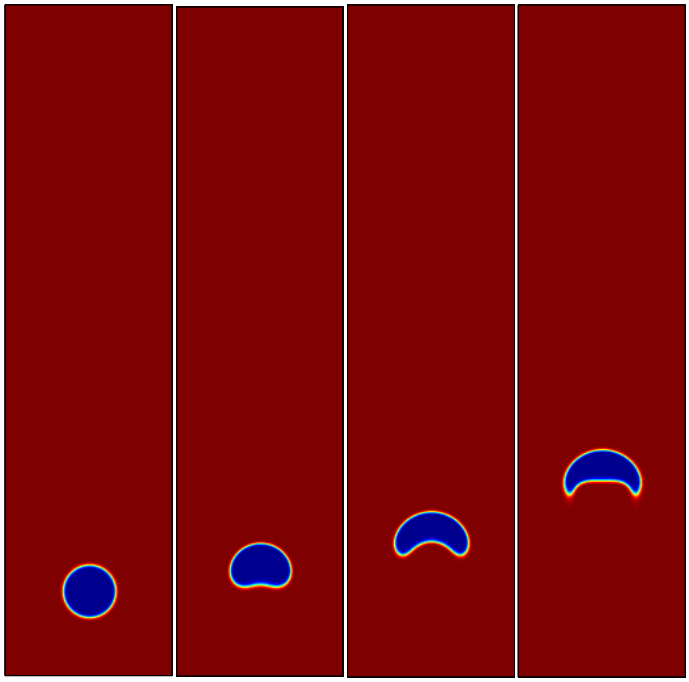


FIGURE 4: Terminal shape and velocity vectors for a bubble rising under buoyancy; comparison of present LBM (red) and [24] (black) results, for three cases: (a) Bo = 0.1, Mo = 0.001; (b) Bo = 10, Mo = 0.1; (c) Bo = 100, Mo = 1000.

to the numerical results of [24], who used a NS-VOF Level set method.

Fig. 5 shows the time evolution of the bubble shape during rising, for a density ratio of 1000. Here, the domain size is 128×512 lattice nodes, and the dimensionless parameters are set to Bo = 100 and Mo = 0.001. During the early stages of the simulation, buoyancy forces are dominant. Hence, the bubble is accelerated and starts rising, causing its shape to change. Eventually, the terminal shape of the bubble is formed, when buoyancy , surface tension and viscous forces are balanced.



(a) $t=0$ (b) $t=10{,}000$ (c) $t=20{,}000$ (d) $t=40{,}000$

FIGURE 5: Time evolution (at selected time steps) of rising bubble shape during the bubble acceleration phase, for $\rho_1/\rho_2=1{,}000$, Bo = 100, and Mo = 0.001.

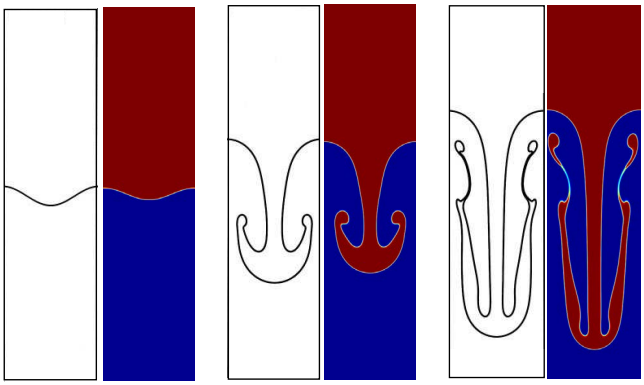
4.3 Rayleigh-Taylor instability

The classic Rayleigh-Taylor instability is used to demonstrate the accuracy of our current model to solve more complicated flows. Here, two immiscible fluids of different densities are placed in a channel, the heavier fluid on top, the lighter one on the bottom. Under the influence of gravity, the heavier fluid will gradually sink into the lighter fluid, which is displaced upwards. The dimensionless numbers that are important in this test case are the Atwood number and the Reynolds number, which are defined as:

$$A=\frac{\rho_1-\rho_2}{\rho_1+\rho_2} \quad \text{and} \quad \mathrm{Re}=\frac{\sqrt{W g}W}{\nu}, \tag{33}$$

where W is the width of the channel, and ρ_1 and ρ_2 are densities of heavy and light fluids, respectively. We set-up this simulation following [9] and use no-slip boundary

conditions at the top and bottom boundaries, and periodic boundary conditions at the lateral sides; gravity is chosen to satisfy $\sqrt{Wg} = 0.04$. The kinematic viscosity for both fluids is the same, and the Atwood and Reynolds numbers are 0.5 and 256, respectively (with accordingly $\rho_1/\rho_2 = 3$). The simulation is carried out on a grid of 256×1024 lattice nodes. Fig. 6 compares the results of our present method to the numerical results of [9] for three selected time steps; a very good agreement can be seen between both methods.



(a) $t' = 0.0$ (b) $t' = 3.0$ (c) $t' = 4.5$

FIGURE 6: Rayleigh-Taylor instability problem for $\rho_1/\rho_2 = 3$, $A = 0.5$, $\mathrm{Re} = 256$. Time evolution of the two-fluid interface for three dimensionless times $t' = t/\sqrt{W/g}$; leftward panels: results of [9]; rightward panels: present results.

4.4 Breaking wave

Previous work [25,26] showed that a periodic sinusoidal wave of large amplitude, with the initial velocities being calculated from linear theory, is not stable and rapidly breaks, since the initial velocity field is not in equilibrium with the initial wave profile. To limit computational time, as in earlier work, the simulation is assumed to be periodic in the flow direction. This characteristic makes such periodic sinusoidal waves a convenient and efficient way of studying wave breaking [2].

The initial wave velocity and interface shape, obtained from linear theory are,

$$\eta = \frac{H}{2}\cos(kx) \tag{34}$$
$$u = \frac{H}{2}\sigma\frac{\cosh k(h+z)}{\sinh(kh)}\cos(kx)$$
$$v = \frac{H}{2}\sigma\frac{\sinh k(h+z)}{\sinh(kh)}\sin(kx)$$

where σ is the wave angular frequency and other wave parameters are shown in Fig. 7.

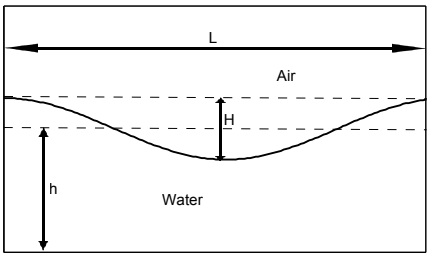


FIGURE 7: Definition sketch for initial interface profile of large amplitude sinusoidal wave.

Fig. 8 shows the time evolution of a high amplitude sinusoidal wave with $H/L = 0.13$ in depth $h/L = 0.25$. We see that the wave it is not stable and rapidly breaks, after traveling one wavelength from initialization.

(a) $t' = 0.0$ (b) $t' = 0.33$

(c) $t' = 1.33$ (d) $t' = 1.66$

(e) $t' = 2.0$ (f) $t' = 2.33$

(g) $t' = 2.66$ (h) $t' = 3.0$

FIGURE 8: Space-periodic wave breaking (Fig. 7). Time evolution of an overturning breaking wave with $H/L = 0.13$ and $h/L = 0.25$ (dimensionless time $t' = t/\sqrt{L/g}$)

The number of grid nodes used in this simulation is 512×256. The nodes update per second (NUPS) for this case is 5×10^5, which is two orders of magnitude faster compare to a single-core CPU implementation.

REFERENCES

[1] Inamuro, T., Ogata, T., Tajima, S., and Konishi, N., 2004. "A lattice boltzmann method for incompressible two-phase flows with large density differences". *Journal of Computational Physics,* ***198***, pp. 628–644.

[2] Lubin, P., Vincent, S., Abadie, S., and Caltagirone, J.-P., 2006. "Three-dimensional large eddy simulation of air entrainment under plunging breaking waves". *Coastal Engineering,* ***53***, pp. 631–655.

[3] Jacqmin, D., 1999. "Calculation of two-phase navier-stokes flows using phase-field modeling". *J. Comput. Phys,* ***155***, pp. 96–127.

[4] Unverdi, S. O., and Tryggvason, G., 1992. "A front-tracking method for viscous, incompressible, multi-fluid flows". *J. Comput. Phys,* ***100***, pp. 25–37.

[5] Brackbill, J., Kothe, D. B., and Zemach, C., 1992. "A continuum method for modeling surface tension". *J. Comput. Phys,* ***100***, pp. 335–354.

[6] Cahn, J., and Hilliard, J., 1958. "Free energy of a nonuniform system". *J. Chem. Phys,* ***28***, pp. 258–267.

[7] Swift, M., Osborn, W., and Yeomans, J., 1995. "Lattice boltzmann simulation of non-ideal fluids". *Phys. Rev,* ***75***, pp. 830–833.

[8] Shan, X., and Chen, H., 1993. "Lattice boltzmann model for simulating flows with multiple phases and components". *Phys. Rev,* ***E47***, pp. 1815–1819.

[9] He, X., Chen, S., and Zhang, R., 1999. "A lattice boltzmann scheme for incompressible multiphase flow and its application in simulation of rayleigh-taylor instability". *J. Comput. Phys,* ***152***, pp. 642–663.

[10] Chapman, S., and Cowling, T., 1970. *The mathematical theory of nonuniform gases.* Cambridge University Press.

[11] Tölke, J., and Krafczyk, M., 2008. "Teraflop computing on a desktop pc with gpus for 3d cfd". *International Journal of Computational Fluid Dynamics,* ***22***, pp. 443–456.

[12] Rothman, D., and Keller, J., 1988. "Immiscible cellular-automaton fluids". *J. Statist. Phys,* ***52***, pp. 1119–1124.

[13] Inamuro, T., Ogata, T., and Ogino, F., 2004. "Numerical simulation of bubble flows by the lattice boltzmann method". *Future Generation Computer Systems 20, 959-964.*

[14] Tölke, J., and Krafczyk, M., 2008. "Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA". *Computing and Visualization in Science,* ***1***, pp. 29–39.

[15] Janßen, C., and Krafczyk, M., 2011. "Free surface flow simulations on GPGPUs using LBM". *Computers and Mathematics with Applications,* ***61***(12), June, pp. 3549–3563.

[16] Banari, A., Janssen, C., and Grilli, S. T., 2012. "An improved lattice boltzmann multiphase model for high density ratios". *to be submitted.*

[17] Qian, Y., d'Humieres, and Lallemand, 1992. "Lattice bgk models for navier stokes equation". *Europhys Lett,* ***17***, pp. 479–84.

[18] Yu, D., Mei, R., Luo, L., and Shyy, W., 2003. "viscous flow computations with the method of lattice boltzmann equation". *Progress in Aerospace Sciences,* ***39***, pp. 329–367.

[19] Sukop, M., and T.Throne, D., 2005. *Lattice Boltzmann Modeling.* Springer.

[20] Gross, M., Moradi, N., Zikos, G., and Varnik, F., 2011. "Shear stress in nonideal fluid lattice boltzmann simulations". *Phys. Rev. E,* ***83***, p. 017701.

[21] Freudiger, S., 2009. "Entwicklung eines parallelen, adaptiven, komponentenbasierten stroemungskerns fuer hierarchische gitter auf basis des lattice-boltzmann-verfahrens". PhD thesis, Technischen Universitaet Carolo-Wilhelmina zu Braunschweig.

[22] Clift, R., Grace, J., and Weber, M., 1978. "Bubbles, drops, and particles". *Academic Press.*

[23] Bhaga, D., and Weber, M., 1981. "Bubbles in viscous liquids: shapes, wakes and velocities". *Fluid Mech,* ***105***, pp. 61–85.

[24] Sun, D., and Tao, W., 2010. "A coupled volume-of-fluid and level set (voset) method for computing incompressible two-phase flows". *International Journal of Heat and Mass Transfer,* ***53***, pp. 645–655.

[25] Longuet-Higgins, M. S., and Cokelet, E. D., 1976. "The deformation of steep surface waves on water. i. a numerical method of computation". *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences,* ***350***, pp. 1–26.

[26] Grilli, S. T., Skourup, and Svendsen, 1989. "An efficient boundary element method for nonlinear water waves". *Engineering Analysis with Boundary Elements,* ***6***, pp. 97–107.