GPGPU-based rising bubble simulations using a MRT lattice Boltzmann method coupled with level set interface capturing

Mohammad Amin Safi[a,*], Stefan Turek[a]

[a]*Institute of Applied Mathematics (LSIII), TU Dortmund, Vogelpothsweg 87, D-44221 Dortmund, Germany*

Abstract

A multiphase Lattice Boltzmann (LB) scheme coupled with a level set interface capturing module is used for the simulation of multiphase flows, and in particular, rising bubbles under moderate and high density and viscosity ratios. We make use of consistent time integration and force discretization schemes in particular for pressure forces along with using multiple relaxation time (MRT) form of the collision in the LB equation which enables us to preserve stability and accuracy for high density and critical Eo numbers. We first present the solution for the standard test of a static bubble in order to show the accuracy of the solution with respect to the Laplace law for pressure and also the spurious velocity level. We present quantitative benchmark computations and error analysis for the 2D rising bubble test cases being further validated against high precision finite element solutions in [1]. Furthermore, by applying efficient multi-core and GPGPU implementations outlines, we demonstrate that the desired parallel scaling characteristics of general LBM solutions are well preserved for the proposed coupled computations. The presented implementations are shown to outperform the available GPU-based phase-field LBM solvers in terms of computational time, turning the scheme into a desirable choice for massive multiphase simulations in three dimensions.

Keywords: Rising bubbles, Lattice Boltzmann method, Level set method, High density ratio, GPGPU implementation

1. Introduction

The use of the lattice Boltzmann method (LBM) has long been a major field of interest for the simulation of multiphase flows [2, 3]. The LBM's mesoscopic nature allows for incorporating the so-called intermolecular forces to the right hand side of the LBE causing the two different phases to reach equilibrium in single or multi-component disciplines [4, 5, 6].

Consequently, there has been a number of proposals for schemes to model the relevant intermolecular forces in LBM. Among the most popular diffused interface LB schemes is the Shan-Chen model [4] extensively used in both academic and commercial LBM software packages e. g. PowerFlow [7], OpenLb [8] and Palabos [9]. The model relates the interface force to an intermolecular potential ψ, which defines the non-ideal deviation from LBM's ideal gas equation of state (EOS) for the pressure. While the original EOS of the Shan-Chen model in [4] becomes notoriously unstable at density ratios of almost larger than 20, more robust EOS like

*Corresponding author
Email address: amin.safi@math.tu-dortmund.de (Mohammad Amin Safi)

the R-K or C-S equations allow for density ratios as high as 1000 in single component two phase problems (e. g. water and its vapour), yet at the price of relatively large spurious velocities around the interface [10].

A major improvement to the class of diffused interface LB models is proposed by Lee and Lin [11, 12, 13], where they replaced Shan-Chen's interaction force with the Cahn-Hilliard chemical potential and a pressure evolution LBE is solved for the pressure and momentum instead of density and momentum. Therefore, density profiles and the interface is diffusely captured via solving a second LBE for the order parameter (density). Unlike the Shan-Chen model, they were successful in gaining high density and viscosity ratios and low spurious velocities. However, such a scheme becomes excessively expensive as it requires large number of derivatives for evaluating the chemical potential and density gradient and needs to solve a second LBE (almost as costly as the first one) while the resolution of the interface still strongly depends on that of the underlying mesh for the LBE.

A new trend in the development of multiphase LB models is to confine the LBE to solve for the flow and let a second PDE track the interface in a sharp manner. The idea is motivated from the fact that in single bubble or droplet problems, a sharp realization of the instantaneous position of the interface at high density/viscosity differences and large Eo numbers is of paramount importance. Thommes et al. [14] and later Becker et al. [15] followed this strategy for 3D simulations using a two-fluid approach, where each fluid is solved on its own domain, and the interface forces are imposed explicitly by modifying the bounce-back boundary condition on the curved interface. The interface is then captured through solving level set equation. Finally, the method has to go through an additional re-filling step as the interface changes position in time. However, one might need to construct two different mesh spacings on each fluid side in the case of large viscosity ratios, resulting in more mathematical complexity and higher computational costs [14].

As contrary to the two-fluid approach in [14], Mehravaran and Hannani [16] tried to extend the one-fluid approach, widely used in the Navier-Stokes based solutions, towards LBE, i. e. to solve for the velocity using LBE for a single virtual fluid on the entire domain and treat the interface boundary conditions as local smeared-out force terms. They also proposed to capture the interface by solving the level set equation. Nevertheless, their work does not include any insight regarding the accuracy in the pressure field and the force descretization. Consequently, this preliminary fails to be of practical use under high density and viscosity ratios, as will be discussed in section 5. Furthermore in [16], one could not find validations against the established static or rising bubble solutions and the comparisons are rather restricted to general qualitative ones.

In this paper we follow the general idea of the one-fluid approach, except that we employ a consistent time integration and force discretization in particular with regards to descretizing the pressure gradient term. We show that the accurate realization of the pressure jumps across the interface is the key to a make the model produce reliable solutions in case of high density and viscosity differences as required for multiphase systems e. g. air bubbles in water. Moreover, to keep the scheme stable at very low viscosities as well as to minimize the spurious velocities, we employ the multiple relaxation time (MRT) collision scheme. The accuracy and robustness of the resulting scheme are demonstrated through rigorous error analysis for the static bubble test case, ans well as extensive benchmarking against the finite element solutions for the rising

bubble problems as provided using the FeatFLOW package in [1].

High parallel scalability has always been one of the most remarkable features of almost every LBM solution, making it competitive to the conventional Navier-Stokes solvers in terms of computational time and cost. This necessitates the new-coming variants of LBM to retain such desired computational characteristics. This is the motivation behind the second part of the paper, where efficient GPGPU implementation outlines are considered, in particular with regards to the coupling with the level set module and the associated computational bottleneck that arise for instance in the reinitialization of the level set function.

The rest of the paper is organized as follows; the derivation and discretization of the MRT coupled multiphase LB model is presented in section 2. Section 3 elaborates on the solution of the level set equation as well as different techniques to reinitialize the level set function. Section 4 is dedicated to the implementation outlines for the many core general purpose GPU (GPGPU) parallel processing. The numerical results and validations for the static and rising bubbles are then provided in section 5 followed by the achieved computational performances in section 6. Finally, the paper is closed with conclusions and discussions in section 7.

2. The coupled LB-level set model

2.1. The lattice Boltzmann Equation

The lattice Boltzmann equation describes the collision and propagation of a set of distribution functions in a discretized phase space, i. e. the discrete velocity space. The D2Q9 stencil for two dimensional problems adopted in this work includes 9 microscopic velocities e_k, $k = 0, ..., 8$ [17, 18, 19]. He et. al [20] proposed the following LB equation in the presence of the external force F

$$\frac{\partial f_k}{\partial t} + e_k.\nabla f_k = \sum_j \Lambda_{k,j}(f_j - f_j^{eq}(\rho, \mathbf{u})) + \frac{(e_k - \mathbf{u}) \cdot F}{\rho c_s^2} f_k^{eq} \tag{1}$$

where $c_s = 1/\sqrt{3}$ is the lattice speed of sound, ρ and $\mathbf{u}$ are the fluid density and velocity, respectively. The Maxwell equilibrium distribution f_k^{eq} is defined as

$$f_k^{eq}(\rho, \mathbf{u}) = \omega_k \rho \left[1 + \frac{e_k \cdot \mathbf{u}}{c_s^2} + \frac{[e_k \cdot \mathbf{u}]^2}{2c_s^4} + \frac{[\mathbf{u} \cdot \mathbf{u}]}{2c_s^2}\right]. \tag{2}$$

The collision matrix Λ represents the general relaxation form. For a multiple relaxation time (MRT) collision, the relaxation matrix takes the form of $\Lambda = \mathbb{M}^{-1}\hat{\Lambda}\mathbb{M}$ where $\hat{\Lambda}$ is a diagonal relaxation matrix

$$\hat{\Lambda} = \text{diag}\{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}. \tag{3}$$

and $\mathbb{M}$ is the transformation matrix

$$\mathbb{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & 2 & -1 & 2 & -1 & 2 & -1 & 2 \\ 4 & -2 & 1 & -2 & 1 & -2 & 1 & -2 & 1 \\ 0 & 1 & 1 & 0 & -1 & -1 & -1 & 0 & 1 \\ 0 & -2 & 1 & 0 & -1 & 2 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & -2 & 1 & 0 & -1 & 2 & -1 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \end{bmatrix}. \tag{4}$$

A primary setting suggested for the relaxation parameters is (see [21, 22])

$$\hat{\Lambda} = \mathrm{diag}\{1, s_1, s_2, 1, s_4, 1, s_4, 1/\tau, 1/\tau\}. \tag{5}$$

where τ is the relaxation time and is related to the kinematic viscosity as $\nu = (\tau - 0.5)/3$. The general rule of thumb for s_1, s_2 and s_4 is to keep their values near 1.0. However, since s_1 is associated with the kinetic energy mode, its selection strongly affects the obtained velocity field. We will discuss in section 5 that setting $0.3 < s_1 < 0.5$, gives the optimal results in the problems concerned in this paper.

From equation (1), one could eventually recover the Navier-Stokes equations in the nearly incompressible limit, using the Chapman-Enskog expansion [23]. As a result, the pressure could be recovered as a linear EOS

$$P = \rho c_s^2. \tag{6}$$

2.2. One-fluid multiphase formulation

The one-fluid approach also known as the continuum surface force (CSF) approach for multiphase flow problems consists of solving the Navier-Stokes equations along with the surface tension boundary conditions as local smeared out forces acting on the interface Γ. In that sense, Sussman et al. proposed the following form of the governing equations [24]

$$\rho(\phi)\partial_t \mathbf{u} + \rho(\phi)\mathbf{u}\cdot\nabla\mathbf{u} + \nabla P - \nabla\cdot\left(\mu(\phi)(\nabla\mathbf{u} + \nabla\mathbf{u}^T)\right) = -\sigma\kappa(\phi)\delta_\varepsilon(\phi)\mathbf{n}(\phi) \tag{7}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{8}$$

where σ is the surface tension coefficient, δ is the Dirac delta function and κ and $\mathbf{n}$ are the mean curvature and normal vector to the interface, respectively. Equation (7) is then coupled with the level set equation

for the advection of the level set function ϕ, initially assumed to be a signed distance function, with fluid velocity $\mathbf{u}$

$$\partial_t \phi + \mathbf{u} \cdot \nabla \phi = 0. \tag{9}$$

where $\phi = 0$ indicates the interface Γ. As such, the bulk fluid properties could be related to the individual properties of each phase as functions of the signed distance function

$$\begin{cases} \rho(\phi) = \rho_l H(\phi) + \rho_g (1 - H(\phi)) \\ \mu(\phi) = \mu_l H(\phi) + \mu_g (1 - H(\phi)) \end{cases} \tag{10}$$

where $H(\phi)$ is a regularized Heaviside function such as

$$H(\phi) = \begin{cases} 0 & \phi < -\varepsilon \\ \frac{1}{2}\left[1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi} sin(\frac{\pi\phi}{\varepsilon})\right] & |\phi| \leq -\varepsilon \\ 1 & \phi > \varepsilon \end{cases} \tag{11}$$

and ε is the interface thickness on each side. Consequently, a smoothed realization of the delta function would be of the form [24]

$$\delta(\phi) = \frac{dH}{d\phi} = \begin{cases} 0 & \phi < -\varepsilon \\ \frac{1}{2}\left[\frac{1}{\varepsilon} + cos(\frac{\pi\phi}{\varepsilon})\right] & |\phi| \leq -\varepsilon \\ 0 & \phi > \varepsilon \end{cases} \tag{12}$$

Furthermore, the curvature as well as the normal vector to the interface could be obtained based on the level set function ϕ as

$$\mathbf{n}(\phi) = \frac{\nabla \phi}{|\nabla \phi|} \qquad , \qquad \kappa(\phi) = \nabla \cdot \mathbf{n} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right). \tag{13}$$

A direct mapping of the CSF approach, as described above, to the LBM would lead to severe numerical errors since the simple EOS in equation (6) would break down in case of non-small density changes. Moreover, in the viscous force term in 7 one still has to apply the divergence to the variable viscosity $\mu(\phi)$. In order to remove the unfavourable forms of the variable density and viscosity, equation (7) is divided by $\rho(\phi)$ and a virtual density $\bar{\rho} = 1$ is introduced to the convective terms of the momentum equation [16]. Consequently, the new LBE is similar to equation (1), except that it describes the momentum transport of a single virtual fluid with density $\bar{\rho} = 1$ in the nearly incompressible limit

$$\bar{\rho}(\phi)\partial_t \mathbf{u} + \bar{\rho}(\phi)\mathbf{u} \cdot \nabla \mathbf{u} + \nabla P - \bar{\mu}(\phi)\nabla \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) = \bar{F} \tag{14}$$

where $\bar{\mu} = \mu(\phi)/\rho(\phi)$ is the virtual viscosity. The modified force term $\bar{F}$ is obtained as

$$F = \frac{1}{\rho(\phi)}\left[-\sigma(\phi)\kappa(\phi)\mathbf{n}\delta_\varepsilon + \nabla\mu(\phi) \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \nabla P\left(\rho(\phi) - 1\right) + \mathbf{G}\right]. \tag{15}$$

We note that a more detailed derivation of the above equation could be found in the technical report by Safi and Turek [25] as well as in [16]. One could also see that $\bar{F}$ comes with two new terms; the first one accounts for the jump in the viscose force term, while the second one adds the effect of the pressure gradient ∇P, whose effect becomes noticeable across the interface.

2.3. LBE discretization

2.3.1. Time integration

Starting from equation (1), we use a forward Euler time integration for the collision term along with the Crank-Nicolson scheme for the force term to have [26]

$$f_k(x + e_k\Delta t, t + \Delta t) - f_k(x,t) = -\sum_j \Lambda_{k,j}(f_j(x,t) - f_j^{eq}(x,t)) + \frac{\Delta t}{2} S_k\Big|_{(x,t)} + \frac{\Delta t}{2} S_k\Big|_{(x+e_k\Delta t, t+\Delta t)} \tag{16}$$

where Λ is the generalized relaxation matrix and S_k is defined as

$$S_k = \frac{(e_{k,i} - \mathbf{u}_i)\cdot \bar{F}_i}{\bar{\rho} c_s^2} f_k^{eq}. \tag{17}$$

To render the LBE into an explicit form we use the transformation $g_k = f_k - \frac{\Delta t}{2} S_k$ suggested by Premnath and Abraham [27]

$$g_k = f_k - \frac{\Delta t}{2} S_k \tag{18}$$

$$g_k^{eq} = f_k^{eq} - \frac{\Delta t}{2} S_k \tag{19}$$

to obtain

$$g_k(x + e_k\Delta t, t + \Delta t) = g_k(x,t) - \sum_j \Lambda_{k,j}(g_j(x,t) - g_j^{eq}(x,t)) + (I - \frac{1}{2}\Lambda_{k,j}) S_j(x,t). \tag{20}$$

In the practical implementation, the above equation is carried out in two steps, namely the on-site *collision* step

$$g_k^*(x,t) = g_k(x,t) - \sum_j \Lambda_{k,j}(g_j(x,t) - g_j^{eq}(x,t)) + (I - \frac{1}{2}\Lambda_{k,j}) S_j(x,t) \tag{21}$$

followed by *streaming* to the neighbouring nodes

$$g_k(x + e_k\Delta t, t + \Delta t) = g_k^*(x,t) \tag{22}$$

As a result of the transformation, we recover the fluid pressure and velocity using the zero- and first-order moments of the new distribution function g_k, respectively,

$$\bar{\rho}\mathbf{u}_i = \sum_k e_{k,i} g_k + 0.5\bar{F}_i \qquad , \qquad P = c_s^2\bar{\rho} = c_s^2 \sum_k g_k. \tag{23}$$

2.3.2. Force discretization

Considering equation (15) for the force $\bar{F}$, we need to discretize the derivatives ∇P for the pressure force term, then $\nabla \mathbf{u}$ as in the viscous force term, and finally $\nabla \phi$ for calculating $\mathbf{n}(\phi)$ and $\kappa(\phi) = \nabla \cdot \mathbf{n}$ in the surface tension force, respectively (see equation (13)). The most straightforward, second order accurate approach to obtain the first derivatives (∇a in general) would be to use the central differencing scheme along the main X and Y directions

$$\begin{aligned} \nabla_x a(x,y) &= \frac{a(x+h,y)-a(x-h,y)}{2h} \\ \nabla_y a(x,y) &= \frac{a(x,y+h)-a(x,y-h)}{2h} \end{aligned} \tag{24}$$

where $h = 1$ is the lattice spacing. However, particular care must be taken for the spacial discretization of the pressure term since for high density ratios the changes in pressure become very steep across the interface, leading to large numerical errors and hence incorrect prediction of the interface location. In fact, a closer look at S_k in equation (17) reveals that the term $e_k \cdot \bar{F}$ comprises the directional derivatives of pressure; $e_k \cdot \nabla P$. Moreover, due to the EOS, we have that $O(e_k \cdot \nabla P) \approx O(e_k \cdot \nabla f_k)$, and hence the term $e_k \cdot \nabla P$ contributes to the convection term in the left hand side of the LBE. Consequently, in order to stabilize the effect of the pressure gradient we apply the following averaged scheme proposed in [11] to the pressure which combines the effect of a second order upwinding (w.r.t the $-e_k$ direction) with the central differencing

$$(e_k \cdot \nabla P)^{avg} = \frac{(e_k \cdot \nabla P)^C + (e_k \cdot \nabla P)^U}{2} \tag{25}$$

where the directional central and upwind derivations are calculated as:

$$\begin{aligned} (e_k \cdot \nabla P)^C &= \frac{P(x+e_k\Delta t)-P(x-e_k\Delta t)}{2} \\ (e_k \cdot \nabla P)^U &= \frac{-P(x+2e_k\Delta t)+4P(x+e_k\Delta t)-3P(x)}{2}. \end{aligned} \tag{26}$$

It will be further shown in section 5 that such an averaged directional approach for $e_k \cdot \nabla P$ results in significant improvement over the simple central differencing used in [16] for recovering the correct pressure field and w.r.t. lower spurious velocities in case of high density ratios.

We summarise that using the above discretization scheme in two dimensions, one needs to compute a total of 6 central derivatives for $\nabla \phi$, $\nabla \cdot \mathbf{n}$ and $\nabla \mathbf{u}$, plus 8 directional derivatives for $e_k \cdot \nabla P$. On the other hand, one may only require 8 central differences in case of using the naive central derivatives for all the gradients.

3. Solving the level set equation

In order to solve the level set equation (9) for capturing the interface, we use the following second order Runge-Kutta time integration. Assuming that $\phi(x,t)$ and $\mathbf{u}_(x,t)$ are the values of the level set function and the velocity at node x and time t, we have

$$\phi(x,t^*) = \phi(x,t) - \Delta t \mathbf{u}(x,t)^n \nabla \phi(x,t) \tag{27}$$

which gives the predictive value of $\phi(x, t^*)$, followed by a correction solution

$$\phi(x, t + \Delta t) = \phi(x, t) - \frac{\Delta t}{2} \left(-\mathbf{u}(x, t)\nabla\phi(x, t) + \mathbf{u}(x, t)\nabla\phi(x, t^*)\right). \tag{28}$$

To discretize the convective term $\mathbf{u}\nabla\phi$, we use the weighted essentially non-oscillatory (WENO) scheme as described in [28] which extends the first-order accurate upwind differencing to fifth-order spatial accuracy based on the smoothest possible interpolation for the function ϕ. It is worth noting that although the level set function is generally known to have smooth variations in the computational field, the use of the WENO scheme ensures that the interface is accurately convected in the case that sharp kinks or edges of the bubble might locally disturb the signed distance property of the level set functions which eventually produces steep changes in the value of ϕ. As such local effects could only be moderated to some extend by reinitializing the level set function (see section 3.1), it is reasonable to employ the smoothest approximation for the spacial derivatives of ϕ. We note that at this stage of the work the same computational mesh as for the LBM is used for the discretization of the level set equation, meaning that $\Delta t = 1, \Delta x = 1$. Since the velocities obtained from LBM are in the incompressible range of $u_{max} \ll 1$, one could be sure that the CFL condition of $\Delta t < \Delta x/u$ is always satisfied for this choice of $\Delta t, \Delta x$.

3.1. Level set reinitialization

As the solution ϕ of equation (9) advances in time, the level set function deviates from the signed distance property. Therefore, one needs to reinitialize the level set function to be as close as possible to the signed distance function. To this end, we have employed two schemes in the present work: the PDE-based reinitialization scheme proposed in [24] as the most conventional technique in the level set community, and the brute-force scheme described in [29]. Once a signed distance field is constructed using either of the above schemes, the accuracy of the resulting interface capturing would not show a meaningful difference. Therefore, we only briefly describe the two schemes so as to provide a background for later discussions on the computational performance in section 5.

In the PDE-based method, one solves the following reinitialization equation after solving the level set equation

$$\partial_\theta\phi + W(\phi_0)(|\nabla\phi| - 1) = 0 \tag{29}$$

where θ is an artificial time and the sign function $W(\phi_0)$ is defined as (see [30])

$$W(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \Delta x^2}} \tag{30}$$

and $\Delta x = 1$ in lattice units. Assuming the interface to have a thickness of M lattice cells, a total of M iterations would be sufficient to construct a signed distance field around the interface. One should note that for such a scheme to be effective and accurate, ϕ_0 has to be already close to a signed distance function near the interface, meaning that equation (29) has to be solved M times at the end of each outer iteration.

In contrast, the brute-force method finds the minimum distances ϕ_{dist} from the lattice nodes to a series of reconstructed linear edges which approximate the interface. Eventually, the following smoothed reinitialization is used (see [29])

$$\phi_{new} = \alpha\phi_{dist} + (1-\alpha)\phi_0 \tag{31}$$

where ϕ_0 is the value before reinitialization. Our numerical experiments suggest that using $0.1 \leqslant \alpha \leqslant 0.15$ leads to a minimum loss of mass and best accuracy in capturing the interface. We note that one does not need to perform reinitialization in each time step, since the approximation in (31) is so effective that even a periodic reinitialization suffices. The choice of the length of such intervals is discussed more in section 5.

4. GPGPU considerations

LBM implementation on early general purpose GPUs (GPGPU), e. g. nVIDIA's Tesla generation required a careful programming especially with respect to the memory access pattern and use of shared memory so as to avoid non-coalesced and hence serialized memory transfers of individual threads. Modern Fermi and Kepler generation, on the other hand, not only offer higher memory and instruction throughputs, but also remove most of the coding complexities as for older generations. Efficient techniques for performance optimization tuned for LBM could be found in our previous work [31] as well as the more recent works of Kuznik [32] and Obrecht [33], and thus, are not elaborated in the present paper.

For the current scheme, however, a computational bottleneck is experienced in the reinitialization of the level set function. The commonly used PDE reinitialization method needs to solve equation (29) on the entire domain for M times in each outer LB iteration. In this sense, even using a highly optimized GPU kernel for each solution pass would not prevent ending up in a high computational cost. The brute-fore reinitialization, on the other hand, consists of two costly sub-processes: periodic interface reconstruction, and signed distance calculation. Such computations typically involve out-of-cache memory accesses, frequent branch diverging to find interface edges and finally minimization loops, enforcing substantial compute times, if implemented entirely on the GPU side. In order to improve the efficiency of each step, we propose to carry out the reinitialization process in a hybrid CPU-GPU manner as follows

- At the beginning of each reinitialization step, the values of the level set function for the entire domain are copied from the device (GPU DRAM) back to the host (CPU DRAM).
- The interface edges are then reconstructed on the host using a parallel OpenMP code which goes through a search algorithm to interpolate the coordinates to the interface edges on several subdomains. The coordinates are then reduced into a single array on the host.
- An small array called `EdgeData` of the size of order L (problem's characteristic length in lattice units), containing the interface data is copied to the array `Dev_EdgeData` located on GPU's `constant` memory.

- The array `dev_EdgeData` is then assigned to a GPU kernel which finds the minimum distance from each lattice node (corresponding to one thread) to the interface. Since all threads read the same `constant` memory address of `dev_EdgeData`, a highe memory throughput is guanarntied for this kernel. The kernel eventually updates the values of the level set function based on equation (31).

In fact, the overhead associated with device-host-device memory transfer is negligible provided that the algorithm is called only periodically. In addition, interface reconstruction search and reduction process is carried out efficiently in parallel on the CPU side, since the available cache space for CPU cores is by far larger than that for streaming multiprocessors on GPUs. By keeping the search process running on CPU, one also does not run into complexities in GPGPU programming for saving and reducing the interface edge data, since individual CPU threads write their edge data in separate arrays, which are eventually merged into a single array to be transferred to the device. We will discuss later in section 5.4 that this hybrid parallel algorithm causes an effective saving in the computational time such that the reinitialization process may account up to only 3% of the total simulation time for the problems of interest in this paper.

5. Numerical results

In all the numerical tests in this section we make use of the two numerical discretization approaches described in section 2.3, namely Approach 1 and Approach 2; Approach-1 refers to the discretization analogous to the scheme in [16], i. e. applying simple central differencing for all the gradients including the pressure, whereas Approach-2 uses the directional formulation for the gradient of pressure and the central differencing for the rest of the gradients. We eventually aim to compare our rising bubble simulations against the finite element solutions by Hysing et al., we choose the parameter settings as defined for the two test-cases in [1]. A summary of the configurations converted in LBM units is given in table 1, while the corresponding macroscopic values could be found in [1].

Table 1: Physical parameters and dimensionless numbers for test case 1 and 2, applied to static and rising bubble

test case	ρ_l	ρ_g	μ_l	μ_g	Re	Eo	ρ_l/ρ_g	μ_l/μ_g
Test case 1	10	1	0.1	0.01	35	10	10	10
Test case 2	500	0.5	2	0.02	35	125	1000	100

The values of surface tension coefficient σ as well as gravity $\mathbf{g}$ have to be calculated in accordance to the desired values for the non-dimensional numbers, e. g. Eo and Re numbers

$$Eo = \frac{4\rho_l \mathbf{g}(r_0 L)^2}{\sigma} \tag{32}$$

$$Re = \frac{\rho_1 \sqrt{\mathbf{g}}(2r_0 L)^{3/2}}{\mu_l} \tag{33}$$

where r_0 is the bubble radius and L is the number of lattice cells covering the X dimension of the domain. Additionally, to preserve a consistent accordance with the time measurement in [1], we use the following equation to calculate the macroscopic time

$$T = t\sqrt{\frac{\mathbf{g}}{\mathbf{g}_0 L}} \tag{34}$$

were $\mathbf{g}_0 = 0.98$ is the macroscopic gravity used in [1] and t is the time in LB units (iteration number).

The brute-force method is used as the default reinitialization scheme to produce the numerical results. Then, the level set function ϕ is reinitialized every $N = 5$ time steps for the base mesh of $h = 1/40$. By looking at equations (32) and (34) one infers that $\Delta t \propto h^2$, and thus N increments by a factor of 4 as the lattice is refined by a factor of 2.

5.1. Static bubble

We start our validation tests with the static bubble problem. The bubble has the radius of $r_0 = 0.25$ and an interface thickness of $\varepsilon = 0.05$, placed in the centre of a 1×1 computational domain with periodic boundary conditions on all the entities. We aim to check the solutions against the Laplace law which, for a two dimensional static bubble, describes the pressure difference due to the surface tension effect as

$$P_{in} - P_{out} = \frac{\sigma}{r_0}. \tag{35}$$

We will also examine the magnitudes of the spurious velocities which usually form around the interface as a result of the force descretization error and lack of isotropy [22].

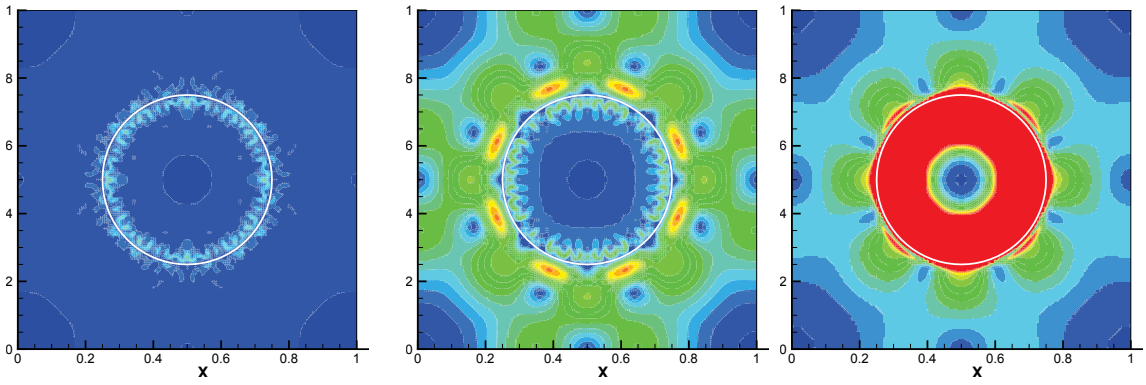
The first test mimics the parameter configuration for test case 1 in Table 1. The simulation is run up to $T = 3$, and perform the computations using both Approach 1 and Approach 2, as well as the MRT-based Shan-Chen diffused interface LB model with eight order force calculations [22]. Table 2 compares the error in the pressure as well as the spurious velocity on different mesh levels. One could see that the maximum of the spurious velocity is in average improved upon the Shan-Chen model by one and two orders of magnitude using Approach 1 and 2 of the coupled LBM-Level set scheme, respectively. Unlike the LB model where the velocity error decreases by refining the lattice, the Shan-Chen model tends to increase the spurious velocities since the actual interface thickness becomes thinner, making the jumps more sharp and hence the force approximation less accurate.

It is also worth nothing that the higher isotropy in Approach 2 has resulted in one order of magnitude lower spurious velocity as compared to Approach 1 (see also figure 1). Nevertheless, the rather low density and viscosity differences in this case introduce no major error in recovering the pressure, and the errors in the Laplace equation are of the same level using both approaches.

Next we increase the density and viscosity ratios to 1000 and 100 respectively in accordance to parameters for the test case 2. As the Shan-Chen model is not capable to incorporate such high density/viscosity differences into its two-component variant, we only present the results using the coupled LBM-level set scheme. As depicted in Table 3, Approach 1 almost fails to recover the correct pressure even up to a lattice

as fine as $h = 1/160$ with a pressure error of 13%, whereas the corresponding error for Approach 2 is less than 1% at nearly one order of magnitude lower spurious velocities.

Figure 2 provides the pressure profile obtained using both discretization approaches on different lattice levels. The simple central differencing approach leads to an overly-smoothed pressure profile which could not reach the correct values of pressure even up to $h = 1/160$ as required by the Laplace law. In contrast, the averaged directional technique of Approach 2 has succeeded in preserving the sharp pressure jump across the interface and the pressure values arrive at the expected levels in a length of almost 2ε.



(a) (b) (c)

Figure 1: Spurious velocity contours measured in macroscopic units for static bubble test 1 using ; (a) Approach 2, $U_{max} = 4.8 \times 10^{-6}$; (b) Approach 1, $U_{max} = 9.8 \times 10^{-6}$; (c) Shan-Chen model, $U_{max} = 3 \times 10^{-4}$

Table 2: pressure and velocity errors for the static bubble test case 1 using different lattice levels

$1/h$	40	80	160	320
Shan-Chen LB Model				
U_{max}	6.5e-5	1.4e-4	3.00e-4	6.24e-4
$\lvert\Delta P - \sigma/r_0\rvert / (\sigma/r_0)$	0.0453	0.0172	0.0017	0.0023
LBM-Level Set, Approach 1				
U_{max}	5.8e-5	3.00e-5	9.8e-6	7.4e-6
$\lvert\Delta P - \sigma/r_0\rvert / (\sigma/r_0)$	0.0108	0.0126	0.0087	0.0058
LBM-Level Set, Approach 2				
U_{max}	1.93e-5	8.2e-6	4.6e-6	2.6e-6
$\lvert\Delta P - \sigma/r_0\rvert / (\sigma/r_0)$	0.0277	0.0147	0.0086	0.0060

5.2. Rising bubble

The performance of the coupled schemes is now evaluated for an unsteady dynamic problem of a single rising bubble. The problem consists of a bubble of radius $r_0 = 0.25$ placed in a rectangular domain of size

Table 3: pressure and velocity errors for the static bubble test 2 using different lattice levels

$1/h$	40	80	160	320
LBM-Level Set, Approach 1				
U_{max}	4.7e-6	5.0e-6	3.0e-6	2.1e-6
$\lvert\Delta P - \sigma/r_0\rvert / (\sigma/r_0)$	0.7520	0.4250	0.1331	0.0249
LBM-Level Set, Approach 2				
U_{max}	5.7e-6	4.1e-6	8.4e-7	3.6e-7
$\lvert\Delta P - \sigma/r_0\rvert / (\sigma/r_0)$	0.1611	0.0211	0.0080	0.0060

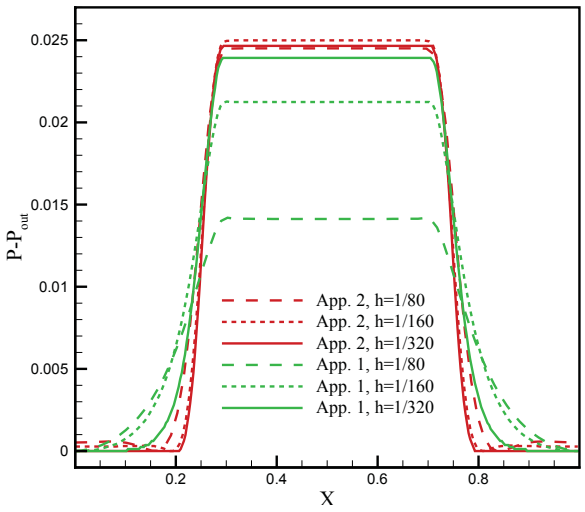


Figure 2: Pressure profile for the static bubble test 2

1×2 as illustrated in figure 3. The no-slip boundary condition is considered for the horizontal walls using the second order bounceback scheme [34], while we impose periodic boundary conditions on the vertical boundaries which, in this particular geometry, implies the slip boundary condition ($\mathbf{u} \cdot \mathbf{n} = 0$) as in [1]. Initially the bubble is stationary and is then allowed to rise by adding the buoyancy force $\mathbf{G} = \mathbf{g}(\rho(\phi) - \rho_l)$ in equation (15), which only acts on the lighter fluid inside the bubble.

In order to provide a quantitative insight into the bubble dynamics, the temporal values of the different bubble metrics are considered as benchmark quantities and will be calculated in time for each test case. Here we use the temporal values of the bubble centroid position y_c, bubble rise velocity $\mathbf{u}_c$ and its circularity $\circlearrowleft$ defined as

$$y_c = \frac{\int_{\Omega_b} y dx}{\int_{\Omega_b} 1 dx} \tag{36}$$

$$\mathbf{u_c} = \frac{\int_{\Omega_b} \mathbf{u}\, dx}{\int_{\Omega_b} 1\, dx} \tag{37}$$

$$\circlearrowleft = \frac{P_a}{P_b} = \frac{\text{perimeter of area-equivalent circle}}{\text{perimeter of bubble}} = \frac{\pi d_a}{P_b} \tag{38}$$

where Ω_b refers to the subspace of the lattice cells with coordinate $\mathbf{x} = (x, y)$ where $\phi(x, y) < 0$, i. e. they fall inside the bubble. One could refer to [1] for more details on the numerical techniques to calculate the above quantities over a structured grid.

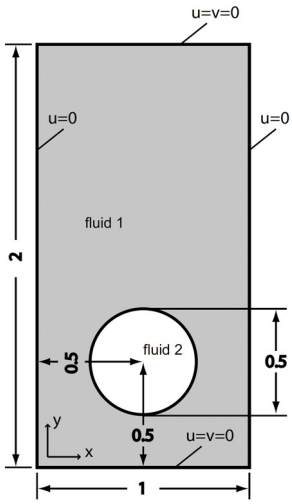
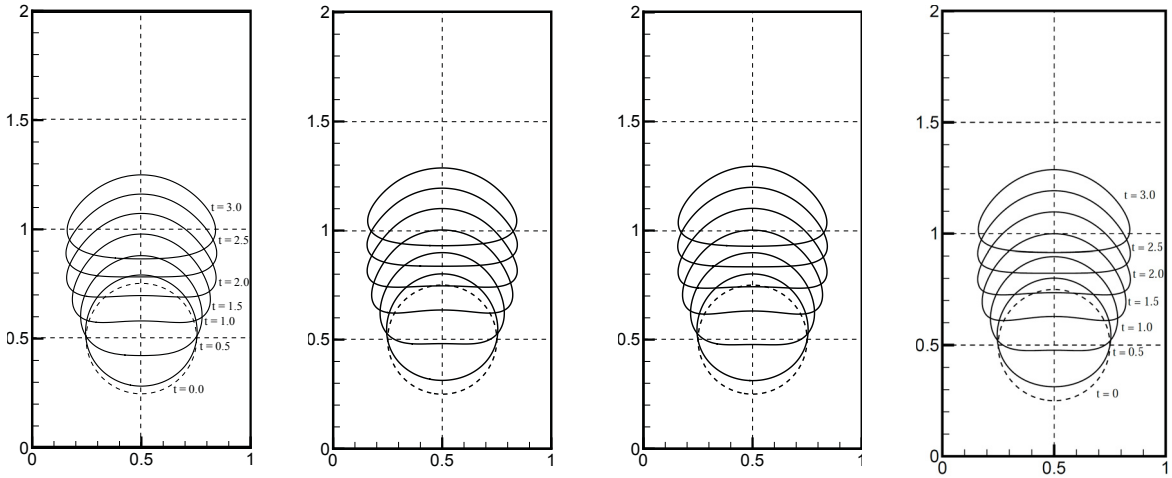


Figure 3: Initial configuration and boundary conditions for the rising bubble problem [1]

5.2.1. Test case 1

In the first test known as *test case 1* in [1], we choose the corresponding parameters as in Table 1 where the density and viscosity ratios are both set to 10. As the time elapses the bubble is expected to gain a stable ellipsoidal shape. An illustration of the time evolution of the bubble shape is depicted in figure 4 for a lattice of $h = 1/160$ using the Shan-Chen LBM, the coupled scheme and the finite element solution obtained by the FeatFLOW package [1]. While the interface obtained by the LBM-Level Set scheme closely resembles that by the FeatFLOW, the Shan-Chen model produces more noticeable discrepancies from the reference shapes.

An illustration of the terminal velocity and pressure contours and the convergence behaviour of the coupled scheme is presented in figure 5 using Approach 2. The convergence trend is further examined in time for the bubble quantities in figure 6. The differences between the curves of $h = 1/80$, $h = 1/160$ and $h = 1/320$ are quite small, while the curve of the coarsest lattice of $h = 1/40$ has clear deviation from the fine grid solutions and experiences difficulty in retaining the ellipsoidal shape of the bubble after $t = 2$ as reflected in the circularity curves. The main reason is that the quality of the interface reconstruction degrades largely at coarser lattices which also provokes intermittent jumps in the circularity curve due to the periodic reinitialization of a weakly reconstructed interface. Table 4 contains the time-integrated errors

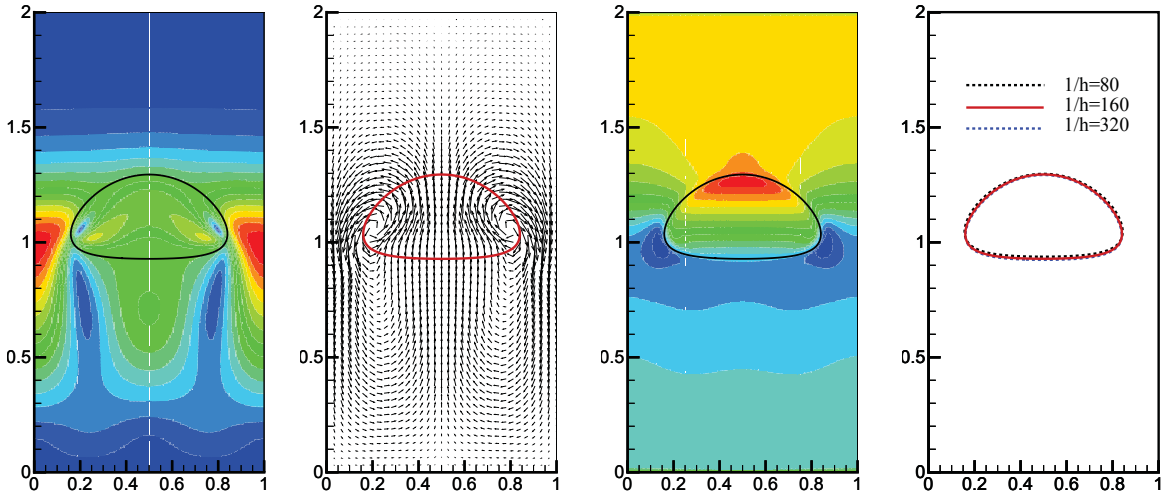


Shan-Chen LBM | LBM-LS App. 1 | LBM-LS App. 2 | FeatFLOW

Figure 4: Time evolution of the bubble shape for the Test Case 1

in the rise velocity in L_2 and L_∞ norm spaces measured relative to the reference lattice of $h = 1/320$. The calculated rates of convergence (ROC) show that the overall accuracy of the coupled scheme in L_2 norm is mostly in between 1.4 and 2 which confirms the second order accuracy in both Approach 1 and 2. The slightly better rate of convergence for Approach 1 is probably due to stronger dependence of the scheme on the lattice resolution as reflected in the larger values of L_2 norm errors for the coarse grid as compared to those from Approach 2.

Finally, figure 7 shows a quantitative comparison between different schemes on the finest level of $h = 1/320$. Again, a very close agreement is discerned between the solutions from the coupled LBM-level set scheme (using both approaches) and those produced by FeatFLOW. Although the Shan-Chen model is able to capture the overall temporal trend, it exhibits apparent deviations especially in the bubble velocity, where the predicted velocities are in average lower than the reference solution. As also pointed out for the static case, Approach 1 is successful in recovering the correct pressure field for this relatively low density and viscosity differences and therefore we did not expect it to introduce any noticeable error in the dynamic test as well.

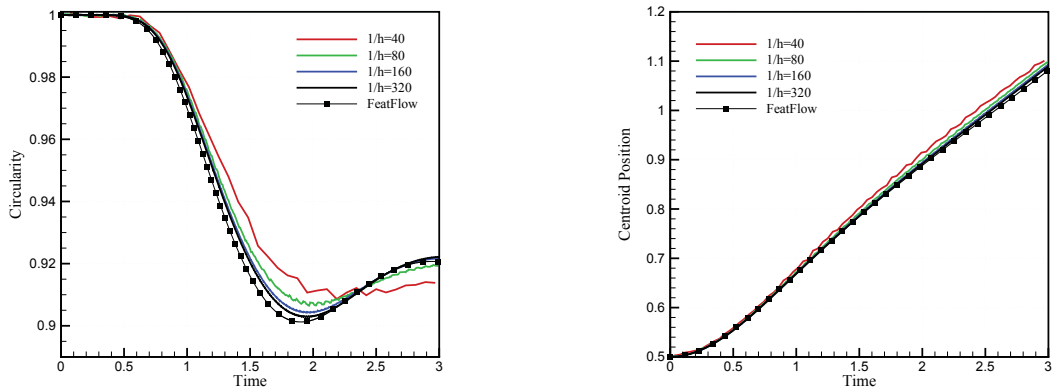


(a) (b) (c) (d)

Figure 5: Rising bubble state at time $T = 3$ for test case 1 for $1/h = 160$: (a) velocity contour (b) velocity vectors (c) pressure contour (d) interface shape for different lattice levels

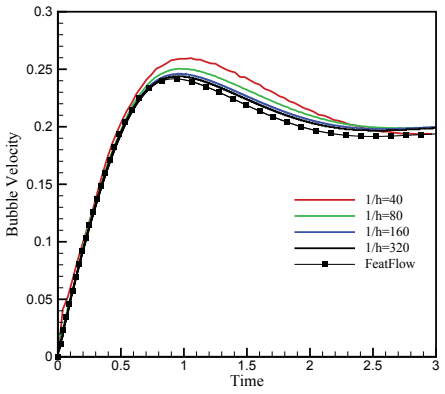
Table 4: Errors in the bubble rise velocity and the rates of convergence (ROC) for the test case 1

LBM-Level Set, Approach 1				
$1/h$	$\| e_2 \|$	ROC_2	$\| e_\infty \|$	ROC_∞
40	0.149206		0.177204	
80	0.053590	1.477261	0.055123	1.684686
160	0.013609	1.977321	0.024643	1.161464
LBM-Level Set, Approach 2				
$1/h$	$\| e_2 \|$	ROC_2	$\| e_\infty \|$	ROC_∞
40	0.051369		0.046389	
80	0.021663	1.245647	0.020337	1.189679
160	0.000692	1.644344	0.000757	1.424072



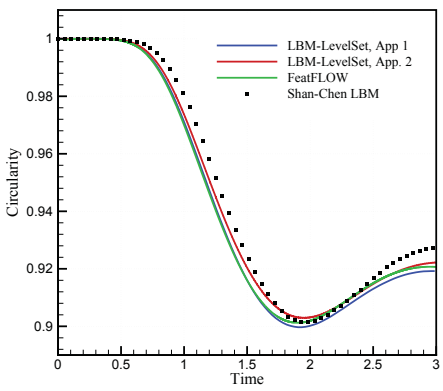
(a)

(b)

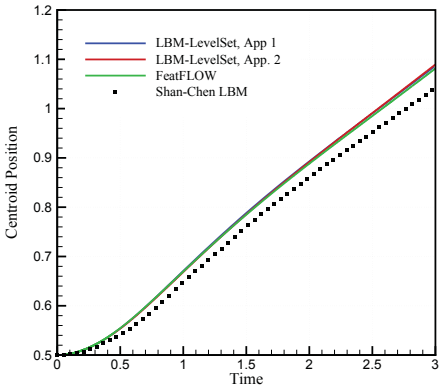


(c)

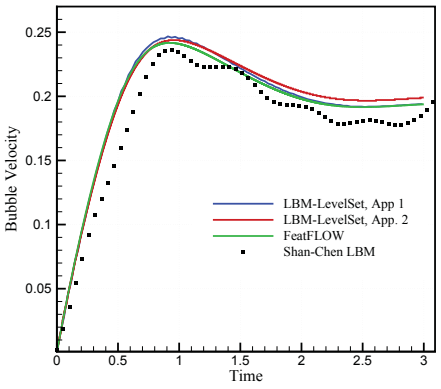
Figure 6: Temporal values of bubble metrics using the coupled LB-level set schem with Approach 2 on different lattice levels for the test case 1, (a) bubble circularity (b) centroid position (c) rise velocity



(a)



(b)



(c)

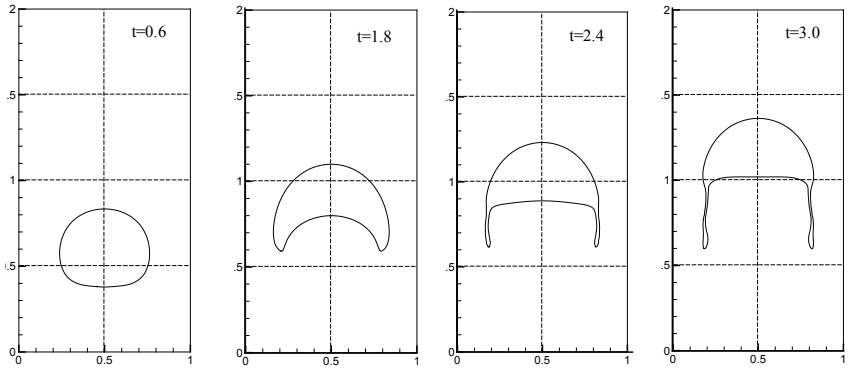
Figure 7: Temporal values of bubble metrics using different numerical schemes for the test case 1, (a) bubble circularity (b) centroid position (c) rise velocity

5.2.2. Test case 2

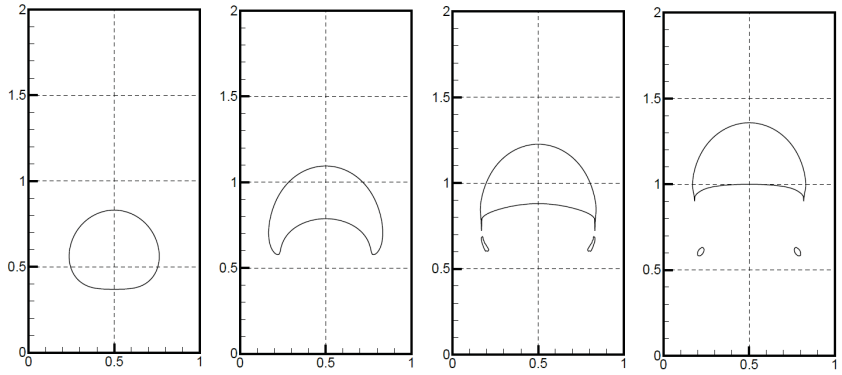
In the second test case we increase the bubble density and viscosity ratios to more demanding values of 1000 and 100 respectively (as in the case of air bubbles in water) while the rest of the parameters are chosen as in table 1. By assuming the same Re number as in the test case 1, the bubble velocity is almost kept constant. However, the increase in the Eo number from 10 in the previous test to 125 means that the surface tension is decreased and the interface is allowed to deform more. According to the bubble shape map in [35], the bubble is expected to exhibit more pronounced deformation and attain the shape of a dimpled cap. As illustrated in figure 8 for a grid of $h = 1/160$, the coupled LB-level set scheme with modifeied descretization of Approach 2 is successful in recovering the time evolution of the bubble shape which closely resembles the one predicted by the FeatFLOW finite element solution in [1]. Figure 9 further depicts the terminal velocity and pressure contour using the same contour levels as for test case 1 in figure 5. It turns out that the trailing vortices expand over a broader range below the bubble as a result of the stretched traling filaments.

A noteworthy observation is that unlike the LBM simulations, the finite element solution predicts the evolution of the trailing filaments into satellite bubbles. We emphasise that such a spin-off phenomena could not be physically verified since level set-based methods do not generally take into account the exact small-scale effects pertained to the chemical potential field in the vicinity of the interface and solutions are partially affected by the specific level set parameterizations in indivudual codes. In fact, there exists no reference solution for this test case after $t = 2$ since other FEM solutions, e. g. the FreeLIFE package [36] do not report such a phenomenon for the same benchmark. This could be examined quantitatively in figure 10 using the converged values of the temporal bubble metrics on a grid resolution of $1/h = 640$. This higher resolution level, applied to all methods, guaranties the accuracy in resolving sharper pressure gradients across the interface as compared to the previous test case. The agreement between the circularity values given by FeatFLOW, FreeLIFE and the coupled LBM-level set breaks up at around $t = 2.3$ when an abrupt jump of the circularity could be clearly spotted in the curve obtained by FeatFLOW. Nevertheless, a fairly good agreement keeps up for the centroid position up to $t = 2.5$. For the rise velocity the agreement stays only up to $t = 1.5$ since the velocity is directly related to the drag force and hence the bubble shape which tends to be slightly more streamlined in the solution by FeatFLOW.

Anothner key observation is that the use of the coupled scheme with naive force discretization of Approach 1 (analogous to the approach in [16]) for this test case results in erroneous predictions of the bubble quantities as well as its shape, as depicted in figure 11 for different lattice levels. This was rather expected from the large truncation errors in pressure gradient approximation emerging in the high density ratio static bubble test. Significant deviations from the reference solutions are thus due to the errors in the pressure field which tend to grow drastically on coarser lattices. In contrary, Approach 2 manifests a smooth convergence behaviour from coarser lattices to finer ones.

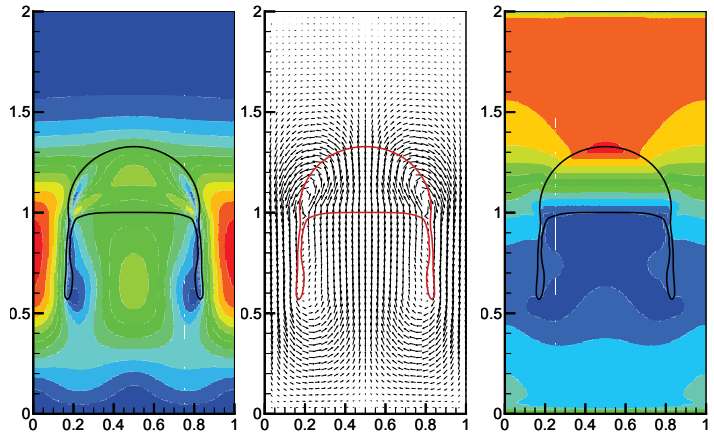


(a)



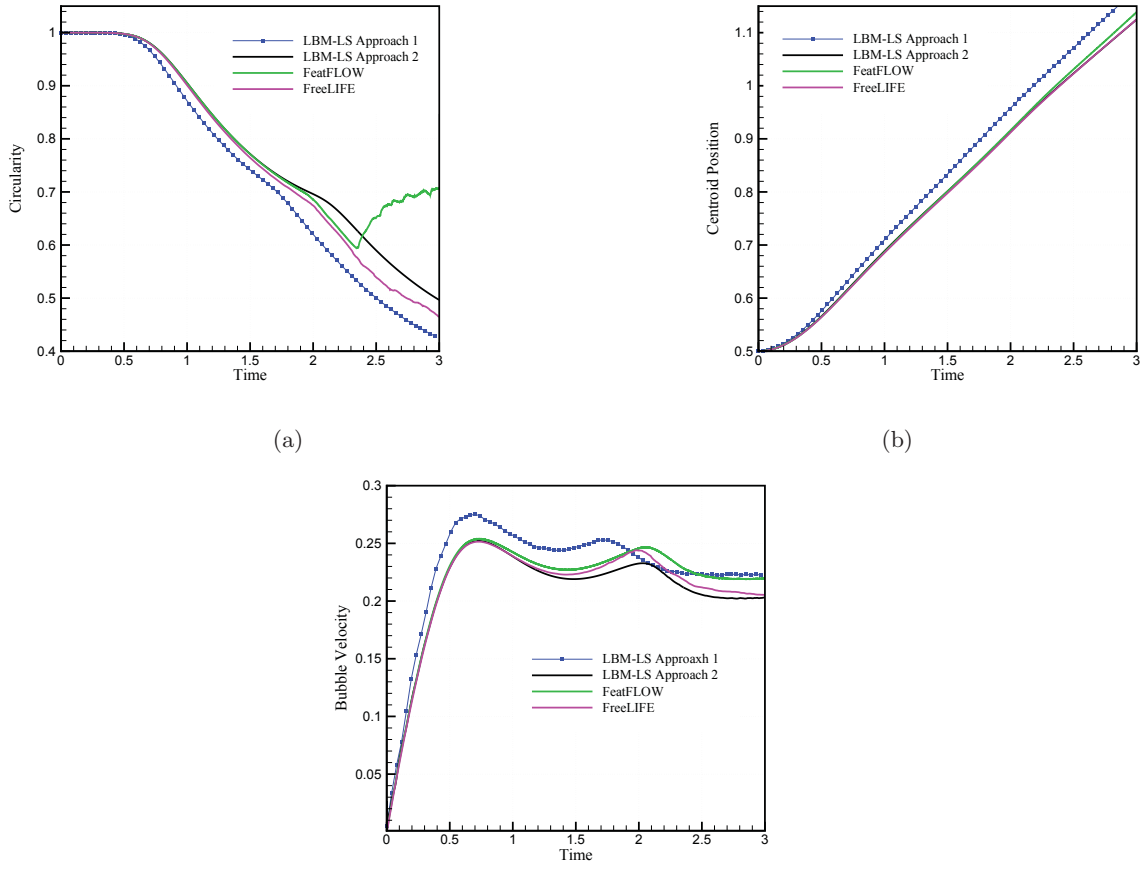
(b)

Figure 8: Time evolution of the bubble in the test case 2 for $h = 1/160$ using (a) LB-level set scheme Approach 2 (b) FeatFLOW



(a) (b) (c)

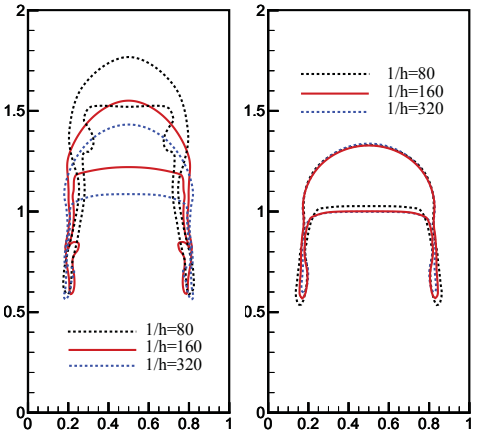
Figure 9: Rising bubble state at time $T = 3$ for test case 2 for $1/h = 160$: (a) velocity contour (b) velocity vectors (c) pressure contour



(a) (b)

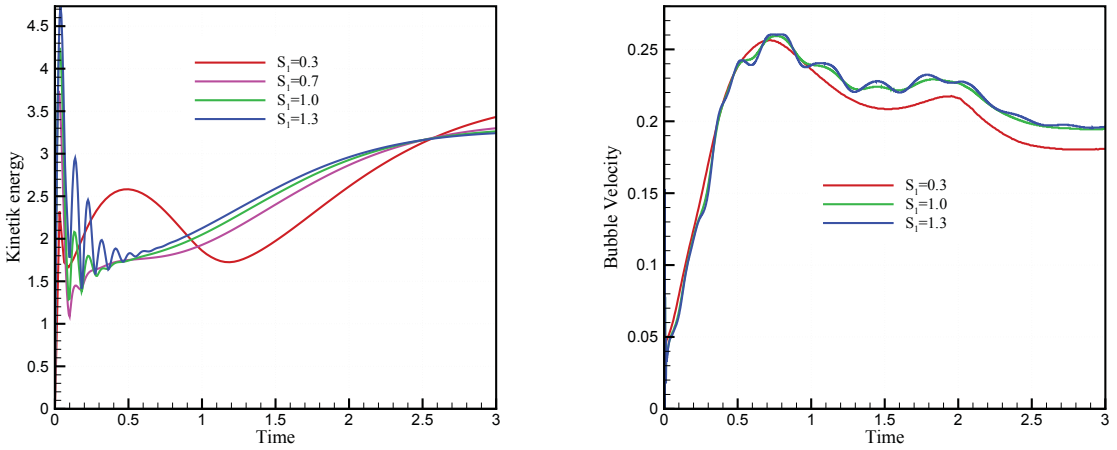
(c)

Figure 10: Temporal values of the bubble quantities using different numerical schemes for test case 2, (a) bubble circularity (b) centroid position (c) rise velocity



(a) (b)

Figure 11: Rising bubble interface at time $T = 3$ for test case 2 using (a) Approach 1 (b)Approach 2



(a) (b)

Figure 12: Effect of choosing different relaxation values s_1 on (a) static bubble kinetic energy (b) the rise velocity of the rising bubble test case 2. In both cases the lattice spacing is $h = 1/160$

5.3. Relaxation time calibration

As previously mentioned in section 2.3.1, one has to take special care in choosing the right relaxation value s_1 for the energy mode η. In order to investigate the effect of different values of s_1 on the numerical results, in figure 12a we have measured the temporal values of the mean kinetic energy $\bar{\eta}$ in the static bubble problem at density and viscosity ratio equal to 1000 and 100 respectively. It is evident from figure 12(a) that by over-relaxing the s_1 values to be larger than 1, the kinetic energy experiences large oscillations before converging to its final level. To dissipate the oscillations one may try to under-relax the energy mode using

$s_1 < 1$ such that the convergence becomes fairly smooth for $s_1 = 0.3$. The observation is also prevalent in the dynamic problem of the rising bubble test case 2 as depicted in figure 12(b) for the bubble rise velocity. It is seen that the rise velocity would not become smooth unless choosing to under-relax the energy moment. Such a phenomenon is believed to become more pronounced and effective in dynamic problems as the Eo number increases, i. e. the inertial effects may dominate the surface tension forces.

5.4. Computational performance

Parallel computations are carried out on a high-end compute machine equipped with an Intel Xeon E5-2670 CPU with 32 processor cores at 2.60GHz and peak memory bandwidth of 51 GB/sec. The machine also benefits from a nVIDIA Kepler K20Xm GPU for GPGPU computing, having 14 streaming multiprocessors (192 processor cores each) and a peak memory bandwidth of 250 GB/sec. For a shared memory parallel implementation of the code on the multi-core CPU, we use OpenMP directives for and split the computational domain to multiple subdomains along the Y axis. The CPU parallel code is based on a 2D version of the optimized 3D code already described in [31]. For GPGPU implementation, we use nVIDIA's `nvcc` compiler for CUDA 6.0 with OpenMP flag of `-Xcompiler -fopenmp` turned on as required for the interface reconstruction parts of the brute-force reinitialization on the CPU. We note that all computations are performed using double precision instructions. This level of precision ensures that the locally extremely small values of the surface tension force are resolved, as their accurate detection is crucial for capturing the exact bubble deformation.

Figure 13 presents the weak scaling of the parallel performance on the multi-core CPU as well as the many-core GPGPU architectures for the test case 2. The conventional million lattice update per second (MLUPS) metric is used for measuring the performance. Two variants of the coupled scheme is implemented; the first one employs the brute-force reinitialization methods, while the second one reinitializes the interface using the PDE-based method. 3.

The highest GPU performance is attained for the code with the brute-force reinitialization, while the code with the conventional PDE-based reinitialization shows an average of 80% lower performance which obviously prevents the coupled LBM computation to reach its expected efficiencies. This impact is also manifested in table 5 which gives a brief survey of the percentage of the total computational time requested by different compute modules for a 320×640 lattice. One could see that the PDE reinitialization asks for nearly the same compute time as the collision and streaming for solving the LB equation, whereas the overhead induced by the brute-force reinitialization account for less than 3% of the simulation time.

Nevertheless, as a result of our efficient GPGPU implementations an overall speed-up of nearly one order of magnitude is achieved over an already optimized parallel CPU code. In terms of computational time, this means that the entire simulation for the second rising bubble test case to reach the steady state on a high resolution lattice of $h = 1/320$ takes less than 30 minutes. To provide a comparison with similar GPGPU implementations of a chemical potential LBM solver, we refer to the recent work by Banari et al. [37] where a best performance of 52 MLUPS for 2D two-fluid Poiseuille flow problem (with no pressure Poison solution) is reported on a Fermi C2070 GPU. In contrast, running the present code on the same device as in [37] shows an average performance of 81 MLUPS. This considerable performance improvement could be partly a result

of the different GPGPU programming techniques used, but to a greater extent a consequence of the more computationally affordable work-flow of the present coupled LBM-level set formulation where one does not need to solve a second costly LBE to find the density profile.

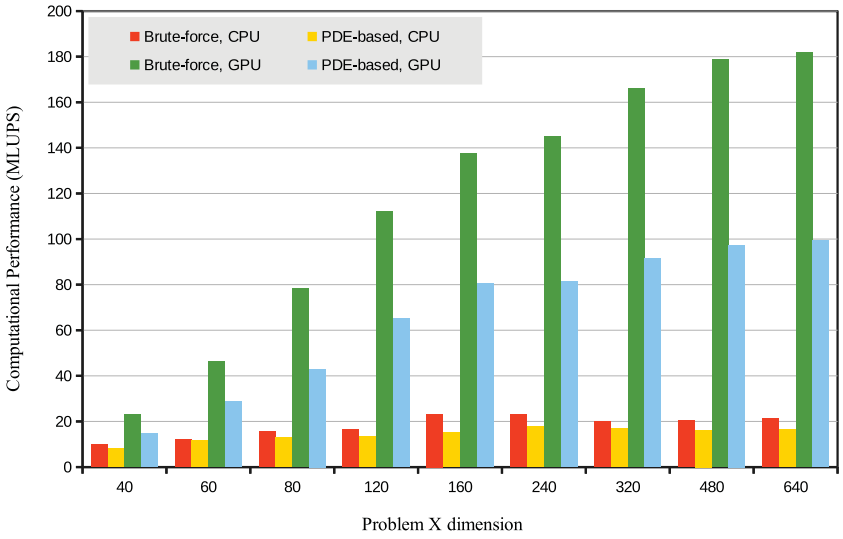


Figure 13: Computational performance for various problem sizes using the two variants of the coupled scheme on CPU and GPGPU

Table 5: Percentage of the total simulation time of different compute modules measured on GPU

Compute module	Brute-force Reinit.	PDE Reinit.
Collision and streaming	73.56	42.8
Update properties	9.02	5.3
Solving level set equation	8.43	5
Calculating normal and curvature	6.69	4.2
level set reinitialization	2.3	42.7

5.4.1. Summary and conclusions

A second order descretization of a coupled LB-level set scheme was introduced for high density and viscosity ratio multiphase flow problems based on the one-fluid formulation of the governing equations. Unlike the previous works on the one-fluid LB model, we showed that by employing a consistent time integration and a stable isotropic pressure force descretization, the coupled LBM-level set scheme is enabled to capture the correct pressure field for static and rising bubble problems with realistic (e. g. air-water) density and viscosity ratios. In particular, qualitative and quantitative benchmarking against the existing established finite element solutions of rising bubble problems proves the accuracy of the improved coupled scheme at both moderate and high density and viscosity ratios as well as for subtle interface deformations associated

with large Eo numbers.

Apart from the accurate numerical predictions, another major significance of the work lies in the efficient implementation of the coupled scheme on GPGPU compute machines. We showed that a combination of the conventional GPGPU optimizations for the LBM part and a hybrid CPU-GPU implementation of the level set module and in particular the brute-force reinitialization technique could end up in substantially shorter computation times as compared to the traditional diffused interface chemical potential LB models. Furthermore, we discussed that the conventional PDE-based reinitialization method, although being effective and affordable for Navier-Stokes solvers which involve large time steps, imposes a destructive performance barrier on the naturally fast LBM computations and is therefore not recommended to be used in combination with LBM simulation.

We finally emphasize that sufficiently short simulation times are of paramount importance for more practical, 3D simulations of large scale bubbly flows. Multiphase systems of this kind often require large grid resolutions to resolve delicate bubble deformations and may therefore become notoriously expensive even for the available parallel LB solvers (see the work by Shu and Yang [38] for instance). As the presented efficient implementations could be directly extended to three dimensions, the authors are highly motivated to perform such simulations while benefiting from fast, but yet compact computational resources offered by GPGPUs. As such, implementations w. r. t. accuracy and performance benchmarking of three dimensional rising bubble simulations are being carried out and are subject to our forthcoming publications.

Acknowledgement

The authors would like to acknowledge the support for this research from the Graduate School of Energy Efficient Production and Logistics of the state of Nordrhein-Westfalen at TU Dortmund, as well as the German Research Foundation (DFG) through the Priority Programme 1648 Software for Exascale Computing (SPPEXA): projects TU 10248-1 and GO 17582-1.

[1] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, International Journal for Numerical Methods in Fluids 60 (11) (2009) 1259–1288.

[2] A. Prosperetti, G. Tryggvason, Computational Methods for Multiphase Flow, First Edition, Cambridge University Press, New York, 2007.

[3] S. Chen, G. Doolen, Lattice boltzmann method for fluid flows, Annual Review of Fluid Mechanics 30 (1998) 329–364.

[4] X. Shan, H. Chen, Lattice boltzmann model for simulating flows with multiple phases and components, Phys. Rev. E 47 (1993) 1815–1819.

[5] X. Shan, Analysis and reduction of the spurious current in a class of multiphase lattice boltzmann models, Phys. Rev. E 73 (2006) 047701.

[6] X. Shan, Pressure tensor calculation in a class of nonideal gas lattice boltzmann models, Phys. Rev. E 77 066702.

[7] Exa powerflow software (2014).
URL `https://www.exa.com/powerflow.html`

[8] Openlb software (2014).
URL `http://optilb.com/openlb/`

[9] Palabos software (2014).
URL `http://www.palabos.org/`

[10] P. Yuan, L. Schaefer, Equations of state in a lattice boltzmann model, Physics of Fluids 18 (4).

[11] T. Lee, C.-L. Lin, A stable discretization of the lattice boltzmann equation for simulation of incompressible two-phase flows at high density ratio, Journal of Computational Physics 206 (1) (2005) 16–47.

[12] T. Lee, P. Fischer, Eliminating parasitic currents in the lattice boltzmann equation method for nonideal gases, Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 74 (4).

[13] T. Lee, P. Fischer, A lattice boltzmann equation method without parasitic currents and its application in droplet coalescence, Vol. 1 SYMPOSIA, 2006, pp. 675–681.

[14] G. Thommes, J. Becker, M. Junk, A. Vaikuntam, D. Kehrwald, A. Klar, K. Steiner, A. Wiegmann, Numerical investigation of a combined lattice boltzmann-level set method for three-dimensional multiphase flow, International Journal of Computational Fluid Dynamics 23 (10) (2009) 687–697.

[15] J. Becker, M. Junk, D. Kehrwald, G. Thmmes, Z. Yang, A combined lattice bgk/level set method for immiscible two-phase flows, Computers and Mathematics with Applications 58 (5) (2009) 950–964.

[16] Simulation of buoyant bubble motion in viscous flows employing lattice boltzmann and level set methods, Scientia Iranica 18 (2) (2011) 231 – 240.

[17] X. He, S. Chen, G. Doolen, A novel thermal model for the lattice boltzmann method in incompressible limit, Journal of Computational Physics 146 (1) (1998) 282–300.

[18] X. He, L.-S. Luo, A priori derivation of the lattice boltzmann equation, Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics 55 (6 SUPPL. A) (1997) R6333–R6336.

[19] L.-S. Luo, Unified theory of lattice boltzmann models for nonideal gases, Physical Review Letters 81 (8) (1998) 1618–1621.

[20] X. He, X. Shan, G. Doolen, Discrete boltzmann equation model for nonideal gases, Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics 57 (1) (1998) R13–R16.

[21] P. Lallemand, L.-S. Luo, Theory of the lattice boltzmann method: Dispersion, dissipation, isotropy, galilean invariance, and stability, Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics 61 (6 B) (2000) 6546–6562.

[22] Z. Yu, L.-S. Fan, Multirelaxation-time interaction-potential-based lattice boltzmann model for two-phase flow, Phys. Rev. E 82 (2010) 046708.

[23] U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, J. P. Rivet, Lattice gas hydrodynamics in two and three dimensions, Complex Syst. 1 (1987) 649–707.

[24] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows, Computers and Fluids 27 (5-6) (1998) 663–680.

[25] A. Safi, S. Turek, Rising bubble simulations using a multiple relaxation time lattice Boltzmann method coupled with level set interface capturing, Tech. rep., Fakultät für Mathematik, TU Dortmund, ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 500 (Jun. 2014).

[26] S. Mukherjee, J. Abraham, A pressure-evolution-based multi-relaxation-time high-density-ratio two-phase lattice-boltzmann model, Computers and Fluids 36 (6) (2007) 1149–1158.

[27] K. Premnath, J. Abraham, Three-dimensional multi-relaxation time (mrt) lattice-boltzmann models for multiphase flow, Journal of Computational Physics 224 (2) (2007) 539–559.

[28] F. R. Osher, S., Level Set Methods and Dynamic Implicit Surfaces, First Edition, Springer, New York, 2003.

[29] S. Turek, O. Mierka, S. Hysing, D. Kuzmin, Numerical study of a high order 3D FEM–level set approach for immiscible flow simulation, in: S. Repin, T. Tiihonen, T. Tuovinen (Eds.), Numerical methods for differential equations, optimization, and technological problems, Computational Methods in Applied Sciences, Vol. 27, Springer, 2012, pp. 65–70.

[30] M. Sussman, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics 114 (1) (1994) 146–159.

[31] A. Safi, M. Ashrafizaadeh, Entropic Lattice Boltzmann simulation of three dimensional binary gas mixture flow in packed beds using graphics processors, Tech. rep., Fakultät für Mathematik, TU Dortmund, ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 504 (Sep. 2014).

[32] F. Kuznik, C. Obrecht, G. Rusaouen, J.-J. Roux, Lbm based flow simulation using gpu computing processor, Computers and Mathematics with Applications 59 (7) (2010) 2380–2392.

[33] C. Obrecht, F. Kuznik, B. Tourancheau, J.-J. Roux, A new approach to the lattice boltzmann method for graphics processing units, Computers and Mathematics with Applications 61 (12) (2011) 3628–3638.

[34] D. Yu, R. Mei, L.-S. Luo, W. Shyy, Viscous flow computations with the method of lattice boltzmann equation, Progress in Aerospace Sciences 39 (5) (2003) 329–367.

[35] G. J. R. Clift, R., M. E. Weber, Bubbles, Drops and Particles, First Edition, Academic Press, New York, 1978.

[36] D. Di Pietro, S. Lo Forte, N. Parolini, Mass preserving finite element implementations of the level set method, Applied Numerical Mathematics 56 (9) (2006) 1179–1195.

[37] A. Banari, C. Janen, S. Grilli, M. Krafczyk, Efficient gpgpu implementation of a lattice boltzmann model for multiphase flows with high density ratios, Computers and Fluids 93 (2014) 1–17.

[38] S. Shu, N. Yang, Direct numerical simulation of bubble dynamics using phase-field model and lattice boltzmann method, Industrial and Engineering Chemistry Research 52 (33) (2013) 11391–11403.