Journal of Computational Physics 228 (2009) 1139–1156

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A lattice Boltzmann method for immiscible multiphase flow simulations using the level set method

G. Thömmes [a], J. Becker [a,*], M. Junk [b], A.K. Vaikuntam [a], D. Kehrwald [a], A. Klar [c], K. Steiner [a], A. Wiegmann [a]

[a] Fraunhofer ITWM, Fraunhofer Platz 1, 67663 Kaiserslautern, Germany
[b] Fachbereich Mathematik und Statistik, Universität Konstanz, 78457 Konstanz, Germany
[c] Fachbereich Mathematik, Technische Universität Kaiserslautern, 67663 Kaiserslautern, Germany

ARTICLE INFO

Article history:
Received 22 August 2007
Received in revised form 29 September 2008
Accepted 12 October 2008
Available online 7 November 2008

MSC:
76P05
76D05
65M06
35B25

Keywords:
Lattice Boltzmann method
Level set method
Free surface
Multiphase

ABSTRACT

We consider the lattice Boltzmann method for immiscible multiphase flow simulations. Classical lattice Boltzmann methods for this problem, e.g. the colour gradient method or the free energy approach, can only be applied when density and viscosity ratios are small. Moreover, they use additional fields defined on the whole domain to describe the different phases and model phase separation by special interactions at each node. In contrast, our approach simulates the flow using a single field and separates the fluid phases by a free moving interface. The scheme is based on the lattice Boltzmann method and uses the level set method to compute the evolution of the interface. To couple the fluid phases, we develop new boundary conditions which realise the macroscopic jump conditions at the interface and incorporate surface tension in the lattice Boltzmann framework. Various simulations are presented to validate the numerical scheme, e.g. two-phase channel flows, the Young–Laplace law for a bubble and viscous fingering in a Hele-Shaw cell. The results show that the method is feasible over a wide range of density and viscosity differences.

1. Introduction

The study of immiscible two-phase flows is an important model problem for free surface flows. Free surface flows with multiple fluid phases appear in a wide range of situations in many areas of applications and industrial processes. For example, bubble dynamics is crucial for the design of chemical reactors and devices for transferring mass or heat between liquid–liquid or gas–liquid mixtures; fingering is important in oil recovery when pressurised water is used to extract viscous crude oil from porous rock deposits; the formation and break-up of liquid–metal jets represents a main feature in metal forming processes; droplet oscillations can be used to measure physical material properties and interfacial dynamics of liquids and gases. Owing to the practical importance of these problems, a large body of literature has been accumulated over the years. We mention [35,30] for an overview of this research field.

* Corresponding author. Tel.: +49 631 31600 4245; fax: +49 631 31600 5245.
E-mail address: becker@itwm.fhg.de (J. Becker).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.10.032

During the last two decades, the lattice Boltzmann method (LBM) has been developed as an alternative numerical scheme for solving the incompressible Navier–Stokes equations. Historically, it originated from lattice gas cellular automata, which simulate the dynamics of fluid particles on a microscopic level based on the Boltzmann equation in a discrete phase space using only a small number of velocities adapted to a regular grid in space. While lattice gas automata deal with individual particles, particle densities (also occupation numbers or particle distributions) are the objects in the LBM model. Together with a number of simplifications this statistical, mesoscopic picture has been a major improvement of LBM which contributed to its competitiveness as a numerical solver. The main advantages attributed to LBM are the ease of implementation (since the nonlinear Navier–Stokes equations are replaced by the semi-linear Boltzmann equation), the simplicity in simulating domains with complex geometry (in particular porous media), and the ease of parallelisation (since only local operations are performed). Chen and Doolen [8], and Succi [39] give a concise and comprehensive summary of the LBM approach, its applications, and the various ramifications and extensions the basic method has undergone over the years.

Numerous combinations of conventional fluid solvers based on finite difference, finite volume or finite element discretisations with numerical methods for free surfaces have been described to simulate multiphase flows. In the LBM framework, most approaches for immiscible multiphase flows are based on the colour gradient method [13], the method of Shan and Chen [33] or the free energy approach [37], and integrate the representation and evolution of the interface in the LBM algorithm. Recently, a hybrid method combining LBM and the front-tracking method has also been proposed [22].

The most simple and wide-spread among these approaches is the *colour gradient* method of Gunstensen and Rothman [14,15,29]. It contains two sets of LBM populations, one for each phase, and models phase separation and interface tension using a *recolouring step*. In each node, the algorithm attempts to separate the two phases as much as possible by redistributing the two sets of populations. Interfaces are implicitly defined by the fluid fraction iso-surface where the content of the two fluids is equal. In general the method is applicable only for small density and viscosity differences and in particular the recolouring step causes grid-dependent artifacts at the interface [21]. A different approach was presented by Shan and Chen [33,34], who introduced the concept of *interaction potentials*. The method in principle models miscible fluids, and immiscible flows can only approximately be described. Swift et al. developed a LBM modification using the *free energy approach* [37,38]. They rely on a second set of populations which describes the fluid fraction and is determined by the free energy of the system. In this case, the lattice Boltzmann approach gives rise to a stress tensor in the Navier–Stokes equations that differs from the classical definition based on transmission conditions at the phase boundary which we use here.

A variety of methods is used for computing free surface motion in areas ranging from mechanical engineering, chemistry and medicine to computer science (see e.g. [26]). These methods are commonly divided into two classes: interface tracking and interface capturing. The former describes surface evolution by tracking individual marker particles moving according to a given velocity field. It can be further subdivided in surface tracking methods, where the set of points constitutes the surface [42], and volume tracking methods, where particles are distributed in space and the surface is reconstructed from the boundary of the point cloud, e.g. in the classical marker-and-cell method [10]. In contrast, the latter class is based on the viewpoint of a marker field and uses a function on a numerical grid that obeys a transport equation with a prescribed advection velocity. In this class, methods of discontinuous type define the surface as the discontinuity set of the field, and methods of continuous type define the surface as a contour surface (or contour line in 2D) of the field.

We apply the continuous interface capturing approach of the level set method, which uses the signed distance to the surface as a marker function in the domain under consideration [31,24]. The evolution of this function over time is governed by a PDE of Hamilton–Jacobi type that is solved numerically by appropriate schemes for hyperbolic equations. The surface is recovered implicitly from the zero level set of the real-valued signed distance function. In the level set framework, topology changes are handled in a natural way. This is an interesting feature when simulating bubble dynamics, where coalescence and break-up of bubbles can be observed in experiments. Furthermore, the level set method can be implemented very efficiently using the narrow-band technique, which stores the distance function only in a small part of the grid around the surface [1]. Other techniques, e.g. the fast marching method for (re)initialisation, further contribute to its computational performance [2].

In the present paper, we develop a hybrid lattice Boltzmann level set method, i.e. an interface capturing method combined with LBM, where boundaries between different fluid phases are represented by sharp interfaces separating the fluids to address the difficulties experienced with the classical LBM approaches. Coupling the level set method with LBM has several advantages. Firstly, the level set method works on a regular, cubic lattice as LBM does, such that numerical quantities can be represented on the same grid thus avoiding errors when transferring data from one grid to the other. Secondly, it combines high accuracy with efficient numerical algorithms and low memory requirements to provide the surface data, e.g. normals and curvatures. Interface evolution is handled independently by the level set method, where the level set function moves according to the flow field of the lattice Boltzmann model. This allows to control fluid flow and interface separately and avoids several drawbacks of previous LB methods for multiphase flows. In particular, the colour gradient method is only applicable when density differences are small and it exhibits grid-dependent perturbations of the interface which cannot be removed in the LBM framework [21]. The method of Chen and Shan suffers, among other things, from mass losses. Mass loss is also an issue in the level set method, but various strategies have been presented in the literature to address this numerical artifact [24,40,36].

To connect the two separate fluid phases in the LBM, we develop a new boundary condition for the populations at the interface. The scheme can be implemented in a similar way as the well-known bounce back boundary conditions at walls. This is achieved by analysing the continuity or jump conditions for the macroscopic quantities velocity, pressure and stress on the kinetic level. In this way surface tension is correctly modelled in the LBM. Moreover, it allows us to simulate flows with large density and viscosity differences, a major goal of our work.

Another hybrid method combining LBM with standard numerical methods for surface computations in [22] presented a lattice Boltzmann front-tracking method which incorporates interface tension via curvature-dependent volume forces along the surface. The method significantly reduces the interface smearing effect of the traditional LBM approaches mentioned above. It successfully simulated the Young–Laplace law and capillary waves with equal viscosity and density. However, our new approach is also able to simulate flow in with high density and viscosity differences. Moreover, the level set method is well suited for topology changes while this is not easily accomplished using particle methods like front-tracking.

The rest of the paper is organised as follows: in section 2 the general setup of the two-phase flow problem is described. Section 3 contains the lattice Boltzmann method and section 4 explains the level set method. In section 5 a new approach for the treatment of the fluid interface within the lattice Boltzmann method is presented. At the interface, suitable LBM boundary conditions have to be prescribed which implement the jump conditions. Moreover, the interface movement requires a reinitialisation of the populations in nodes changing their fluid phase. We also describe how the level set algorithm is coupled with the lattice Boltzmann method. Finally, Section 6 contains a numerical investigation of the approach and a comparison with several benchmark examples. We present channel flow simulations with two immiscible fluid layers and compare with the analytical solutions. The Young–Laplace experiment for the pressure difference inside a bubble is used as a benchmark problem to validate the surface tension of the new scheme. Finally, we investigate viscous fingering experiments in a Hele-Shaw cell. The results show that the new scheme can successfully simulate multiphase flows with complex geometrical features and large viscosity and density differences.

2. The two-phase flow problem

We use the Navier–Stokes equations for two incompressible fluids as a mathematical model for immiscible two-phase flow problems with free surfaces. The equations governing the flow in each phase are presented and the jump conditions coupling the phases at interfaces are discussed in this section.

The flow domain is a bounded, open set Ω in three-dimensional space partitioned into three subsets: Ω_1 and Ω_2 are open subdomains of fluid 1 and 2, respectively, and $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ is the interface (Fig. 1). It is assumed to be a sharp interface where material properties are discontinuous and no mixing in surface layers occurs. Throughout each subdomain material properties are constant. The subdomains and the free surface evolve over time according to the velocity of the flow in Ω.

The evolution of the flow fields is governed by the incompressible Navier–Stokes equations separately in each subdomain Ω_i:

$$\nabla \cdot u^{(i)} = 0, \quad \text{in } \Omega_i, \tag{1a}$$

$$\partial_t u^{(i)} + (u^{(i)} \cdot \nabla) u^{(i)} = -\frac{1}{\varrho^{(i)}} \nabla p^{(i)} + \nu^{(i)} \Delta u^{(i)} + F^{(i)}, \quad \text{in } \Omega_i, \tag{1b}$$

with compatible boundary and initial conditions, for example

$$u^{(i)}(x,t) = 0, \quad \text{on } \partial\Omega_i \setminus \Gamma, \tag{1c}$$

$$u^{(i)}(x,0) = u^{(i)}_{\text{ini}}(x), \quad \text{in } \Omega_i. \tag{1d}$$

Here, $u^{(i)}$ is the velocity, $p^{(i)}$ the pressure, $\nu^{(i)}$ the kinematic viscosity, $\varrho^{(i)}$ the mass density and $F^{(i)}$ an exterior force density per unit mass in subdomain Ω_i.

The system is not completely defined without boundary conditions at the interface Γ. In the presence of viscosity we assume continuity of the velocities; furthermore surface tension balances the jump in normal stresses. Formally this is expressed by the following jump conditions:

$$[u] = 0, \qquad [T]n = 2\sigma\kappa n, \tag{2}$$

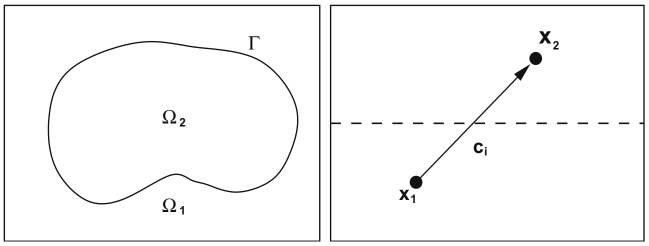

Fig. 1. Left: two-phase partitioning of the domain. Right: link crossing the interface from fluid 1 to fluid 2.

where $T^{(i)} = -p^{(i)}I + 2\mu^{(i)}S^{(i)}$ is the stress tensor with $S^{(i)}_{kl} = \frac{1}{2}\left(\partial_k u^{(i)}_l + \partial_l u^{(i)}_k\right)$, $\mu^{(i)} = \varrho^{(i)}\nu^{(i)}$, σ the surface tension, $\kappa = \frac{1}{2}(\kappa_1 + \kappa_2)$ the mean curvature (for principal curvatures κ_1, κ_2) with respect to the surface normal n. As usual, brackets denote the jump of a quantity, q, across the surface: $[q](x) = \lim_{\epsilon\to 0}(q(x+\epsilon n) - q(x-\epsilon n))$.

Note that the interface and the domains in the model depend on time. The unknowns of the problem are the velocity $u(x,t)$, the pressure $p(x,t)$ and the interface Γ_t. An algorithm for solving two-phase flow problems therefore contains three ingredients:

1. a solver for the flow equations for $u(x,t)$, $p(x,t)$,
2. a scheme for computing the motion of the interface Γ_t, and
3. a method for coupling fluid flow and interface evolution.

In the following chapter we describe the lattice Boltzmann method for fluid flows. In Section 4 we present the level set method for free surface motion and Section 5 addresses the coupling problem by introducing a new LBM boundary condition.

3. The lattice Boltzmann method

We use the lattice Boltzmann method to solve the incompressible Navier–Stokes equations. In this section we give an overview of the method and the specific model used in our implementation. We describe the collision and propagation steps and discuss the implementation of boundary conditions.

The lattice Boltzmann method is based on a kinetic picture of fluid flow and approximates the Boltzmann equation

$$\partial_t f + v \cdot \nabla f = J(f) + G \tag{3}$$

which describes the evolution of the particle density $f(x,v,t)$ in phase space. In Eq. (3), v is the microscopic particle velocity, $J(f)$ denotes the collision operator and G models external forces. In the LBM algorithm, the equation is discretised with a regular grid in space and with a restricted number of velocities adapted to this grid [8,17,39,45]. We use the D3Q15 model in 3D, which has 15 velocity vectors on a cubic grid with unit spacing including one zero velocity. The vectors are given by the columns of the matrix

$$c = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & -1 & 1 & 1 & -1 & 0 & 0 & -1 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 1 & 1 & -1 & 1 & 0 & -1 & 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}. \tag{4}$$

The corresponding particle distributions are denoted by $f_i(x,t) = f(x,c_i,t)$. Density and velocity are recovered from these populations by taking the moments

$$\rho(x,t) = \sum_{i=0}^{14} f_i(x,t), \quad u = \sum_{i=0}^{14} f_i(x,t)c_i. \tag{5}$$

For the collision operator we use the well-known Bhatnagar–Gross–Krook (BGK) approximation [3], such that the discretised evolution equation reads

$$f_i(x + c_i, t+1) = f_i(x,t) - \frac{1}{\tau}(f_i - f_i^{\mathrm{eq}}) + G_i. \tag{6}$$

The parameter τ is the relaxation parameter for the BGK collision operator and controls the kinematic viscosity $\nu = \frac{1}{6}(2\tau - 1)$. Furthermore, the use of the equilibrium distribution

$$f_i^{\mathrm{eq}}(f) \equiv f_i^{\mathrm{eq}}(\rho,u) = f_i^*\left(\rho + 3c_i\cdot u + \frac{9}{2}(c_i\cdot u)^2 - \frac{3}{2}u^2\right) \tag{7}$$

with the corresponding D3Q15 weight factors

$$f_i^* = \begin{cases} \frac{2}{9}, & i = 0, \\ \frac{1}{9}, & i = 1,2,3,8,9,10, \\ \frac{1}{72}, & i = 4,5,6,7,11,12,13,14. \end{cases}$$

is crucial for obtaining the desired Navier–Stokes equations on the macroscopic level. Using these definitions, the odd order moments of f_i^{eq} are zero and for the second and fourth order moments we invoke the relations ($\alpha, \beta, \gamma, \epsilon = 1, 2, 3$)

$$\sum_i f_i^* c_{i\alpha}c_{i\beta} = \frac{1}{3}\delta_{\alpha\beta}, \quad \sum_i f_i^* c_{i\alpha}c_{i\beta}c_{i\gamma}c_{i\epsilon} = \frac{2}{9}(\delta_{\alpha\beta}\delta_{\gamma\epsilon} + \delta_{\alpha\gamma}\delta_{\beta\epsilon} + \delta_{\alpha\epsilon}\delta_{\beta\gamma}),$$

such that it is easily verified that the equilibrium distribution – and hence also the collision operator – preserves density and momentum

$$\sum_i f_i^{\mathrm{eq}} = \rho, \quad \sum_i f_i^{\mathrm{eq}} c_i = u.$$

Provided the initialisation and the boundary conditions are sufficiently regular, a formal asymptotic analysis with respect to the mesh size $\Delta x = h$ and the time step $\Delta t = h^2$ shows the relation between the lattice Boltzmann variables and the Navier–Stokes solution u_{NS}, p_{NS} [18]

$$f_i = f_i^{\mathrm{eq}}(3h^2 p_{NS}/\varrho, h u_{NS}) - 3h^2 \tau f_i^* \Lambda_i : S_{NS} + 3h^2 f_i^* c_i \cdot v_{OT} + O(h^3). \tag{8}$$

Here, Λ_i is defined as

$$\Lambda_i = c_i \otimes c_i - \frac{1}{3}|c_i|^2 I,$$

with the identity matrix I and the tensor product $a \otimes b = ab^T$ of two column vectors. The product A:B of two square matrices is A:B = trace(AB^T). The additional vector field v_{OT} is a solution of a homogeneous linearised Navier–Stokes equation and vanishes if the initial and boundary conditions are suitably constructed.

Computing the moments of the right hand side of Eq. (8), the Navier–Stokes solution (u_{NS}, p_{NS}) is recovered in leading order from

$$\rho = \sum_i f_i = 3h^2 p_{NS}/\varrho + O(h^3),$$

$$u = \sum_i f_i c_i = h u_{NS} + h^2 v_{OT} + O(h^3)$$

so that pressure can be obtained with first order and velocity with second-order accuracy, provided the field v_{OT}, which depends on the initial and boundary conditions, vanishes

$$p_{NS} = \frac{\varrho}{3h^2}\sum_i f_i + O(h), \quad u_{NS} = \frac{1}{h}\sum_i f_i c_i - h v_{OT} + O(h^2).$$

In the interior of the domain, LBM alternates the collision step

$$f_i^+ = f_i - \frac{1}{\tau}(f_i - f_i^{\mathrm{eq}}) + G_i,$$

and the propagation step

$$f_i(x + c_i, t + 1) = f_i^+(x, t + 1),$$

to compute the time evolution of the populations, and solves in this way the incompressible Navier–Stokes equations on the macroscopic level. At the boundary of the domain, however, this procedure cannot be applied since some populations $f_i^+(x,t)$ involved in the propagation step are not defined. More precisely, for boundary points $x_b = x + c_i$ and directions c_i pointing inward, the source point $x = x_b - c_i \notin \Omega$ lies outside the domain. These populations have to be prescribed in such a way that the boundary conditions on the macroscopic level are fulfilled: $f_i(x_b, t+1) = \tilde{f}_i$.

The choice of boundary conditions has an important effect on the accuracy of the numerical solution. For example, the well-known bounce back scheme for no-slip boundary conditions at a wall is second-order accurate with respect to velocity only if the wall is midway between two nodes and first order accurate in all other cases. There is no one-to-one relation between boundary conditions for the Navier–Stokes equations on the macroscopic level and appropriate LBM boundary conditions on the kinetic level. This question has been addressed by many authors who present implementations for different types of fluid dynamical boundary conditions [19,20,4,16,46].

The widely-used bounce back rule is inspired by a simple particle reflection at a wall to implement no-slip boundary conditions ($u = 0$): $\tilde{f}_i = f_{i^*}^+(x_b, t)$, where i^* is the index of the opposite direction $c_{i^*} = -c_i$. We use the more sophisticated Bouzidi boundary condition because it ensures second-order accuracy of the velocity for geometries of arbitrary shape [4]

$$\tilde{f}_i = \begin{cases} 2q f_{i^*}^+(x_b, t) + (1-2q) f_{i^*}^+(x_b + c_i, t), & q < 0, \\ \frac{1}{2q} f_{i^*}^+(x_b, t) + \frac{1-2q}{2q} f_{i^*}^+(x_b + c_i, t), & q \geqslant 0. \end{cases} \tag{9}$$

Here q denotes the normalised distance of the interface from x_b along the link c_i.

We can also apply pressure boundary conditions at the inlet or outlet of a channel. For a flat inlet in the yz-plane, a pressure p_0 in x-direction is imposed by prescribing the density $\rho_0 = 3h^2 p_0/\varrho$ [46]. After defining the ficticious velocity $u_0 = \rho_0 - (f_0 + f_2 + f_3 + f_9 + f_{10} + 2(f_5 + f_8 + f_{11} + f_{13} + f_{14}))$, we set

$$f_1 = f_8 + \frac{2}{3}\rho_0 u_0,$$

$$f_i = f_{i^*} + \frac{1}{12}\rho_0 u_0 - \frac{1}{4}(c_{i2}(f_2 - f_9) + c_{i3}(f_3 - f_{10})), \quad i = 4, 6, 7, 12.$$

At the outlet an analogous procedure can be used. Further suggestions for boundary conditions can be found for example in [19,20,16] and references therein.

Finally, we can add a vector-valued force per unit mass $F(x,t)$ by setting $G_i = 3f_i^* c_i \cdot F$; for example, gravity in z-direction can be included by $G_i = -3f_i^* g c_i \cdot e_z$.

The **LBM algorithm** can be summarised as follows:

1. Collision step: $f_i^+ = f_i - \frac{1}{\tau}(f_i - f_i^{eq}) + G_i$.
2. Propagation step: $f_i(x + c_i, t + 1) = f_i^+(x, t)$, for interior nodes.
3. Boundary conditions: $f_i(x + c_i, t + 1) = \tilde{f}_i(x + c_i, t)$, if $x \notin \Omega$.

4. The level set method

The movement of the interface between different fluid phases is handled in our approach by the level set method. This surface capturing method uses the continuous signed distance function defined on a Eulerian grid and represents the interface by the zero level set of this function. The level set equation of Osher and Sethian is the basis for the approach. In this chapter we present the numerical scheme used to solve this PDE quickly and accurately. Furthermore, we show how the parameters describing the surface can be computed from the level set function.

4.1. The level set equation

Let Γ_t be the orientable free surface at time t. In the level set method, Γ_t is the zero iso-surface of the *level set function* φ: for $x \in \Gamma_t$ we have $\varphi(x, t) = 0$, and $\varphi > 0$ on one side of Γ_t (in the direction of the normal) and $\varphi < 0$ on the other side. When we choose a point x_0 at time t_0 we can follow its path $x(t)$ and its velocity is $v = v(x(t)) = \dot{x}(t)$. For the level set function we can therefore deduce

$$\varphi_t + v \cdot \nabla\varphi = 0. \tag{10}$$

This is the *level set equation* of Osher and Sethian [25] for the evolution of the free surface.

Since the gradient $\nabla\varphi$ is perpendicular to an iso-surface, we know that the gradient is parallel to the normal n: $\nabla\varphi = \|\nabla\varphi\| n$. If the level set function actually is the signed distance to the surface in direction of the normal, we have $\varphi(x(t) + sn, t) = s$ for $s \in \mathbb{R}$, so $\nabla\varphi \cdot n = \varphi_s = 1$, and hence

$$n = \nabla\varphi. \tag{11}$$

The normal therefore coincides with the gradient and, furthermore, since $\kappa = \nabla \cdot n$, also the mean curvature is easily deduced from the signed distance function,

$$\kappa = \Delta\varphi. \tag{12}$$

4.2. Numerical scheme

The computational domain is a rectangular box, as for LBM, where the sides are treated as periodic boundaries. The level set equation of Osher and Sethian (10) is solved in the narrow-band by the Hamilton–Jacobi WENO scheme, which is fifth-order accurate in space. Time integration uses a second-order accurate Runge–Kutta method [24].

In a computer implementation we can significantly reduce memory requirements and computation time by using the *narrow-band technique*. Since we are only interested in the movement of the iso-surface $\{\varphi = 0\}$, it is sufficient to construct the level set function only at points close to the surface. This technique does not store the distant points at all but merely retains several bands of nodes around the current interface (usually 5–10 layers are stored).

The level set function is initialised by the signed distance to a given surface Γ_0. Usually, we start with a triangulation and construct the signed distance function from the triangulation. Of course, it is also possible to start the algorithm with other given initial data φ_0.

The flow field v of the level set Eq. (10) is determined by the external flow field u. In principle, the movement of the iso-surface can be modelled by using $v = u$. The drawback of this approach is that $\varphi(t)$ may gradually lose the signed distance property of φ_0 for times $t > 0$. Therefore, to preserve the signed distance property during the simulation, the velocity field v is constructed using the *constant velocity extensions method* of Adalsteinsson and Sethian [2], i.e.

$$v(t) = u(t) \quad \text{on } \Gamma_t,$$

$$\nabla v(t) \nabla\varphi(t) = 0 \quad \text{in } \Omega.$$

For each grid node adjacent to the zero iso-surface, the nearest point on the surface is determined. Then, the velocity at this projection point is computed from the discrete flow velocity field u, taken from the lattice Boltzmann method, in the neighbourhood by extrapolation using a linear or quadratic polynomial ansatz. Note that u is continuous at the interface while this may not be true for its derivatives. We therefore extrapolate separately on both sides of Γ_t. For the velocity v at the projection point we take the mean value of both extrapolations and store it in the grid node under consideration. Having constructed the velocities adjacent to the surface, the rest of the velocity field in the narrow-band is eventually constructed via the *fast marching method* [1].

After several time steps, the level set $\{\varphi = 0\}$ may have moved in such way that it approaches the boundary of the narrow-band. Also, after some time steps, the function φ may have lost the property of a signed distance function owing to numerical errors. In these cases, it is necessary to adjust the narrow-band and to reinitialise φ. In our code, *reinitialisation* is efficient due to the fast marching method of Adalsteinsson and Sethian [1].

When modelling the flow of a bubble, the mass of the modelled bubble should be preserved. It is well-known that the level set method, due to numerical errors, tends to shrink convex iso-surfaces, i.e. it leads to mass loss. To prevent this, we preserve the volume of, e.g. Ω_1 by correcting the signed distance function φ with the correction term

$$c_\varphi = \frac{V_{exact} - V(\Omega_1)}{S(\Gamma)}, \tag{13}$$

where V_{exact} is the known volume of Ω_1, and $V(\Omega_1)$ and $S(\Gamma)$ are the current volume of part Ω_1 and the area of the interface, respectively. In general, $c_\varphi \ll h$, and the error introduced by the correction is of the same order as the interpolation error (see also [36]).

4.3. Calculation of surface properties

Generally, a higher order finite difference approach is used for computing the surface properties like normal and curvature from the discrete level set function [31]. However, it has been found that estimating surface properties by the finite difference approach can be inaccurate and has a slow rate of convergence with respect to the underlying mesh width. Therefore, we use an alternative method based on *weighted least squares* for estimating the desired surface properties as discussed in [43]. In this approach, depending on the desired order of accuracy, the number of points around the node of interest (the *stencil* in this context) and the degree of a local polynomial model are chosen in advance.

Let $(\bar{x}, \bar{y}, \bar{z})$ be the node; wlog. we can choose the origin (0,0,0). For local coordinates (x,y,z), the mth order local polynomial in $\mathbb{R}^3$ has $l = (m+1)(m+2)(m+3)/6$ coefficients,

$$f(x,y,z) = \sum_{k=0}^{m} \sum_{\substack{p+q+r=k,\\ p,q,r \geqslant 0}} \frac{1}{(p+q+r)!} \hat{c}_{(p,q,r)} x^p y^q z^r. \tag{14}$$

Here, $\hat{c}_0$ is the constant term, $\hat{c}_1, \hat{c}_2, \hat{c}_3$ are the first derivatives, $\hat{c}_4$ to $\hat{c}_9$ are the second derivatives, and the remaining terms are higher order derivatives of the polynomial. For sufficiently smooth functions φ, approximate derivatives to any desired accuracy can be obtained by an appropriate choice of mesh width h and polynomial degree m.

When the polynomial f of degree m with l coefficients approximates φ on $N_r \geqslant l$ stencil points, the coefficients c are computed by minimising

$$\min_c \|Ac - b\|_2^2, \tag{15}$$

where $A \in \mathbb{R}^{N_r \times l}$ is the three-dimensional Vandermonde matrix and $b \in \mathbb{R}^l$ contains the values of φ at the stencil points. The system would be solved in the standard least squares sense by $\hat{c} = (A^T A)^{-1}$ Af. In the weighted least squares approach, $\hat{c}$ is given by $(A^T W^2 A)^{-1} A^T W^2 b$, where W is a diagonal matrix of size N_r. In our approach, we use weights based on φ as in [32], i.e. the diagonal entries are

$$W_{ii} = \frac{h^2}{h^2 + |\varphi_i|}. \tag{16}$$

Here, φ_i denotes the value of the level set function at the ith stencil point.

Once the coefficients $\hat{c}$ are estimated, the next step is to determine where a lattice link intersects the level set. An approximate intersection point $P_i(x_*, y_*, z_*)$ can be found by checking whether there is a sign change of φ in the link's vertices and, if so, solving $f(x,y,z) = 0$. Solving the equation amounts to finding the roots of a polynomial in $\mathbb{R}$. For $m \geqslant 5$, a simple closed solution does not exist. Hence, one forms the companion matrix of the polynomial and gets the unique intersection point from the eigenvalues [43].

The normal at the intersection point, defined as $n = \nabla\varphi / \|\nabla\varphi\|_2$, can be approximated using the local polynomial by setting $n := \nabla f / \|\nabla f\|_2$, where the gradient is evaluated at P_i, $\nabla f = \nabla f(x_*, y_*, z_*)$.

For estimating the curvature, given by $\kappa = \nabla \cdot n$, we would need the Hessian of the level set function. Similar to the estimation of the normal, the Hessian is approximated by the Hessian of the local polynomial $Hf(x_*, y_*, z_*)$. Let t and s be orthonormal vectors to n spanning the tangent plane. For determining the principal curvature we construct the Weingarten matrix a [9] with

$$a = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} = \begin{bmatrix} t_1 & t_2 & t_3 \\ s_1 & s_2 & s_3 \end{bmatrix} Hf \begin{bmatrix} t_1 & s_1 \\ t_2 & s_2 \\ t_3 & s_3 \end{bmatrix}.$$

The eigenvalues of a are given by

$$\kappa_{1,2} = \frac{a_{22}+a_{11}}{2} \pm \sqrt{\frac{(a_{11}-a_{22})^2}{4} + (a_{12})^2}, \tag{17a}$$

and the mean curvature can then be estimated from

$$\kappa = \frac{1}{2}\mathrm{Trace}(\mathbf{a}) = \frac{\kappa_1+\kappa_2}{2} = \frac{a_{22}+a_{11}}{2}. \tag{18}$$

Using this least squares approach, the estimated mean curvature is of order $\mathcal{O}(h^{m-1})$ when the approximating polynomial is of order m.

5. Coupling of LBM and level set method

At the interface, suitable LBM boundary conditions have to be prescribed which implement the jump conditions (2). Moreover, nodes which are crossed by the moving interface change their fluid type such that a reinitialisation of the population is required. Finally the level set algorithm has to be coupled with the lattice Boltzmann method. These three aspects are discussed in the following subsections.

5.1. New boundary conditions at the interface

To ensure the continuity of velocity, $[u] = 0$, across the interface, we use a simple bounce back type Dirichlet condition on each side of the interface. For example, for the point x_2 in Fig. 1, we set

$$f_i(x_2, t+1) = f^+_{i^*}(x_2, t) + 6hf^*_i c_i \cdot \tilde{u} + R_i, \tag{19}$$

where the prescribed velocity $\tilde{u}$ is a linear interpolation of the velocity along the direction c_i, evaluated at the location $\tilde{x} = x_1 + qc_i = x_2 + (q-1)c_i$ on the interface,

$$\tilde{u} = qu(x_2, t) + (1-q)u(x_1, t).$$

The additional term R_i is needed for two reasons: firstly, to ensure the jump conditions of the normal stress and, secondly, to correct the error terms resulting from the bounce back treatment. We set

$$R_i = 6h^2 f^*_i \Lambda_i : A, \quad A = -q(1-q)[S] - (q-1/2)S^{(2)} + O(h). \tag{20}$$

For a practical implementation, the symmetric velocity gradient $S^{(2)}$ at x_2 as well as its jump $[S]$ at the interface point $\tilde{x}$ have to be computed up to first order accuracy from the lattice Boltzmann variables. For the computation of $S^{(k)}$, we choose

$$S^{(k)} = -\frac{3}{2\tau h^2}\sum_i c_i \otimes c_i (f_i - f^{\mathrm{eq}}_i)(t, x_k) + O(h). \tag{21}$$

For the jump $[S]$ we can extract information from the interface condition. First, because the velocity field is assumed to be smooth in each phase and Lipschitz continuous across the interface, the tangential components of the velocity derivative do not jump

$$[(t_1 \cdot \nabla)u] = 0, \quad [(t_2 \cdot \nabla)u] = 0.$$

Expressed in terms of $[S]$, we find

$$[S] : t_1 \otimes t_1 = 0, \quad [S] : t_1 \otimes t_2 = 0, \quad [S] : t_2 \otimes t_1 = 0, \quad [S] : t_2 \otimes t_2 = 0. \tag{22}$$

We remark that (22) expresses four components of the matrix representation of $[S]$ with respect to the local basis (t_1, t_2, n). In fact, any matrix B can be expressed with respect to some orthonormal basis (b_1, b_2, b_3) in the form

$$B = \sum_{k,l=1}^{3} (B : b_k \otimes b_l) b_k \otimes b_l$$

which follows easily by applying the right hand side to a general vector x and using Einstein's summation convention

$$(B : b_k \otimes b_l) b_k \otimes b_l x = (B : b_k \otimes b_l) b_k (x \cdot b_l) = (B : b_l \otimes x) b_k = \mathrm{trace}(Bxb_k^T) b_k = ((Bx) \cdot b_k) b_k = Bx.$$

Moreover, we have for any B and general vectors a, b

$$B : a \otimes b = \mathrm{trace}(Bab^T) = (Ba) \cdot b = a \cdot (B^T b) = B^T : b \otimes a.$$

Noting that $[S] = [S]^T$, we obtain with (22)

$$[S] = ([S] : n \otimes n) n \otimes n + \sum_{k=1}^{2} ([S] : n \otimes t_k)(n \otimes t_k + t_k \otimes n)$$

so that the required product with Λ_i is

$$\Lambda_i : [S] = ([S] : n \otimes n)((n \cdot c_i)^2 - |c_i|^2/3) + \sum_{k=1}^{2} 2([S] : n \otimes t_k)(n \cdot c_i)(t_k \cdot c_i). \tag{23}$$

The remaining three components of $[S]$ with respect to the basis (t_1, t_2, n) are related to the stress conditions at the interface. Using $[T] = -[p]I + 2[\mu S]$, we first note that

$$[\mu S] : n \otimes n = [p] + 2\kappa\sigma, \quad [\mu S] : n \otimes t_k = 0, \ \ k = 1, 2.$$

Then, using the relation $[\mu S] = [\mu]\overline{S} + \bar{\mu}[S]$ with averages $\overline{S} = \frac{1}{2}(S^{(1)} + S^{(2)})$ and $\bar{\mu} = \frac{1}{2}(\mu^{(1)} + \mu^{(2)})$, we find

$$\begin{aligned} [S] : n \otimes n &= \frac{1}{2\bar{\mu}}([p] + 2\sigma\kappa) - \frac{[\mu]}{\bar{\mu}} \overline{S} : n \otimes n, \\ [S] : n \otimes t_k &= -\frac{[\mu]}{\bar{\mu}} \overline{S} : n \otimes t_k. \end{aligned} \tag{24}$$

Algorithm for LBM interface boundary conditions:

The product $\Lambda_i : [S]$ is computed with (23) and (24) using the curvature information, the approximated pressure jump

$$[p] \approx \frac{1}{3h^2}(\rho(x_1, t)\varrho^{(1)} - \rho(x_2, t)\varrho^{(2)})$$

and the averaged symmetric velocity gradient $\overline{S} = (S^{(1)} + S^{(2)})/2$ with $S^{(i)}$ obtained approximately according to (21). Again Using (21), also the required product $\Lambda_i : S^{(2)}$ is approximated. Combined with the pre-factors involving the scaled interface distance q, the quantity R_i in (20) is completely determined in terms of the lattice Boltzmann variables and can be used in (19) in connection with the interpolated velocity $\tilde{u}$.

The consistency of this interface condition can be checked using the asymptotic expansion approach presented in [19]. The basic idea is to insert the truncated expansion (8) into the interface condition and Taylor expand the expression around the interface point $\tilde{x}$. In these expansions, one has to work with the appropriate left and right derivatives because, in general, derivatives jump across the interface. The expansion of f_i and $\tilde{u}$ in (19) at $\tilde{x}$ yields

$$6(1-q)hf_i^* c_i([u_{NS}] + h[v_{OT}]) - 6h^2 f_i^* \Lambda_i : (q(1-q)[S_{NS}] + (q - 1/2)S_{NS}^{(2)}) = R_i + O(h^3)$$

Since R_i contains no first order contributions in h, we conclude that $c_i \cdot [u_{NS}] = 0$ for all directions c_i crossing the interface so that $[u] = 0$. Similarly, the jump condition for the field v_{OT}, which satisfies a homogeneous linearised Navier–Stokes equation, turns out to be $[v_{OT}] = 0$ because R_i contains only quadratic expressions in c_i. Next, the term involving $S_{NS}^{(2)}$ cancels with the associated approximation contained in R_i. The remaining equality between $\Lambda_i : [S_{NS}]$ and the corresponding term in R_i implies the stress conditions.

Remark 1. The derivation of the new interface conditions is not restricted to the choice of the D3Q15 model in our implementation. For simulations in three dimensions, other lattice Boltzmann models, e.g. D3Q19 or D3Q27, could alternatively be used. This affects the number of links intersecting the interface and, hence, the amount of data computed at the interface. For these models the runtime would, in general, slightly increase.

5.2. Refill methods

Since we are dealing with a free-surface problem, fluid nodes can change their type (from fluid 1 to fluid 2, and vice versa) when the interface Γ_t moves. Then the populations have to be reinitialised or refilled. Different procedures for accomplishing this *refill* have been proposed and analysed in [5,6]. We choose the equilibrium/non-equilibrium refill, which produced the best results in these tests. The refill procedure consists of four steps:

1. Interpolate density and velocity from interior neighbours.
2. Compute the corresponding equilibrium.
3. Copy the non-equilibrium part from a direct interior neighbour.
4. Reinitialise by adding equilibrium and non-equilibrium parts.

Suppose we choose an inward pointing direction c_i in a boundary point $x = x_b \in \Omega_1$, e.g. the direction of smallest angle with the surface normal. Then the density is interpolated using the three nearest neighbours in the interior of subdomain Ω_1:

$$\tilde{\rho} = 3\rho(x - c_i, t+1) - 3\rho(x - 2c_i, t+1) + \rho(x - 3c_i, t+1).$$

Interpolation of the velocity uses two neighbours and the velocity u_Γ at the interface point x_Γ, which is taken from the level set method:

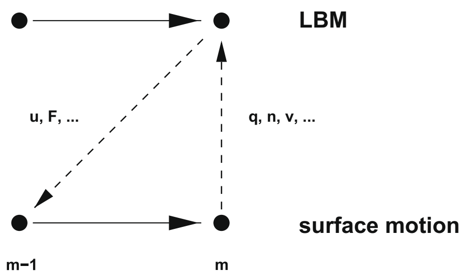

Fig. 2. Data exchange between LBM and the level set method at time step n. LBM passes the current flow field u, while the level set method computes the description for new surface points (e.g. distance q, normal n, velocity v).

$$\tilde{u} = \frac{2}{q^2+3q+2}u_\Gamma + \frac{2q}{q+1}u(x-c_i,t+1) + \frac{2q}{q+2}u(x-2c_i,t+1).$$

Here, $q = \|x_\Gamma - x_1\| / \|c_i\|$ is the distance to the interface normalised by the link length. The new equilibrium $f_i^{\mathrm{eq}}(\tilde{\rho},\tilde{u})$ is then added to the non-equilibrium part copied from the direct neighbour, $f_i^{\mathrm{neq}}(x-c_i,t+1)$. Of course, the same procedure is also applied in the second subdomain Ω_2.

5.3. Coupling of the two methods

A simulation is started by creating the initial interface Γ_0 from a triangulation in stereo lithographic format (STL). The level set code then creates the surface description for LBM: the links intersecting the interface, the normalised distance q along the links, the normal, the tangential vectors t_1, t_2, and the principal curvatures κ_1, κ_2. Moreover, the velocity of the intersection point is passed. With this data the LBM computes the flow field and applies the new boundary condition to couple the two fluid phases. As illustrated in Fig. 2, after a prescribed number of time steps, LBM passes the current velocity field to the level set code. The level set function is conveyed according to this field over the same time interval and passes the new geometry information to LBM. This completes a step of the combined algorithm and the simulation proceeds by alternating LBM and the level set method until termination.

6. Numerical results and examples

In this section we test the lattice Boltzmann method coupled with the level set method in a few different situations. First, channel flows of two immiscible fluid layers with known analytical solution are simulated. Then we verify the correct imple-

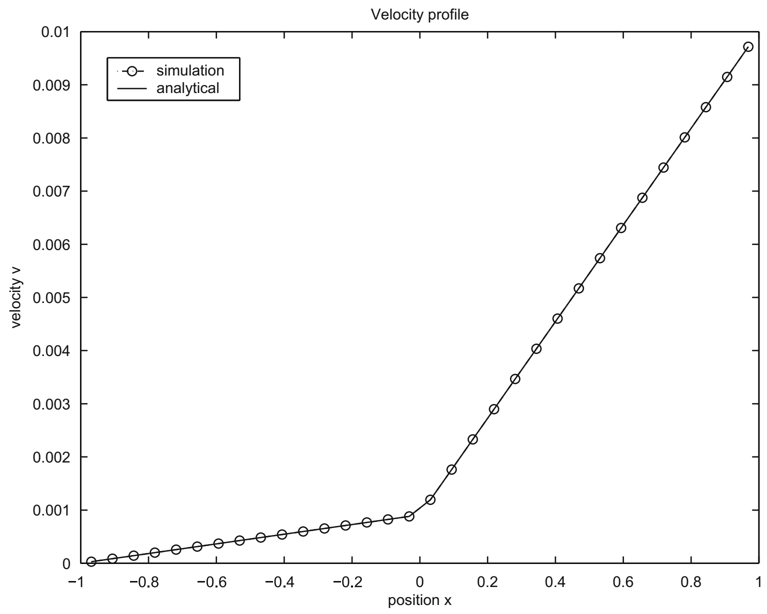

Fig. 3. Velocity profile for a Couette channel flow with viscosities $\mu_1 = \frac{4}{6}$ (left, $x < 0$) and $\mu_2 = \frac{1}{6}$ (right, $x > 0$).

mentation of surface tension in the Young–Laplace bubble experiment. Finally, in a Hele-Shaw cell, interface motion in viscous fingering experiments is investigated.

6.1. Couette and Poiseuille channel flows

We simulated channel flows with two fluid layers in a channel along the x-axis. In this case the analytical solution only depends on the ratio of dynamic viscosities. Furthermore, the presence of surface tension should have a stabilising effect on the interface, which in this setup is subject to viscous stresses, and preserve its flat shape.

First we simulated a Couette flow with fixed lower wall at $y = -1$ and upper wall at $y = 1$ moving with constant velocity $v_0 = 0.16$ in x-direction. At the walls no-slip boundary conditions were implemented using the bounce back method, and periodic boundary conditions were applied at the inflow and outflow. The densities of the fluids were $\rho_1 = \rho_2 = 1$ and the surface tension coefficient was $\sigma = 0.016$. Starting with fluids at rest, after an initial phase the flow should approach a stationary solution. By imposing continuity of the velocity $[v] = 0$ and the shear stress $[\mu\partial_x v] = 0$, a linear velocity profile was derived from the Navier–Stokes equations

$$v(x) = \begin{cases} v_0\left(\frac{\mu_1}{\mu_1+\mu_2}x + \frac{\mu_2}{\mu_1+\mu_2}\right), & x \geqslant 0, \\ v_0\left(\frac{\mu_2}{\mu_1+\mu_2}x + \frac{\mu_2}{\mu_1+\mu_2}\right), & x < 0, \end{cases}$$

where μ_1 denotes the dynamic viscosity in the lower fluid and μ_2 the viscosity in upper fluid. We simulated the flow in a quadratic domain $[-1,1]^2$ in the xy-plane with N nodes in x- and y-direction. In the 3D code we used 8 nodes in z-direction and imposed periodic boundary conditions along this axis. Fig. 3 shows excellent agreement with the analytical solution of the resulting velocity profile on a grid with $N = 32$ nodes across the channel after $T = 1000$ time steps. This is also confirmed by the convergence analysis in Fig. 4. In the figure the relative velocity error in the maximum norm

$$e_\infty = \frac{\max_i |v_i - v(x_i)|}{\max_i |v(x_i)|}$$

is plotted versus the grid spacing $h = \frac{2}{N}$ for $N = 16, 32, 64, 128$. As expected, our numerical experiments also showed that results were independent of σ. Moreover, flows with different density values, in particular ratios up to 1:1000, could be simulated and produced the same results when the kinematic viscosities in the LBM were adjusted accordingly to simulate given dynamic viscosities.

As a second example we considered two-phase Poiseuille flows. Here, both walls ($y = 1$ and $y = -1$) were at rest and the flow was driven by a pressure difference Δp_0. At the walls we used bounce back no-slip boundary conditions and pressure boundary conditions were applied at the inflow and outflow. The densities and the surface tension coefficient were the same

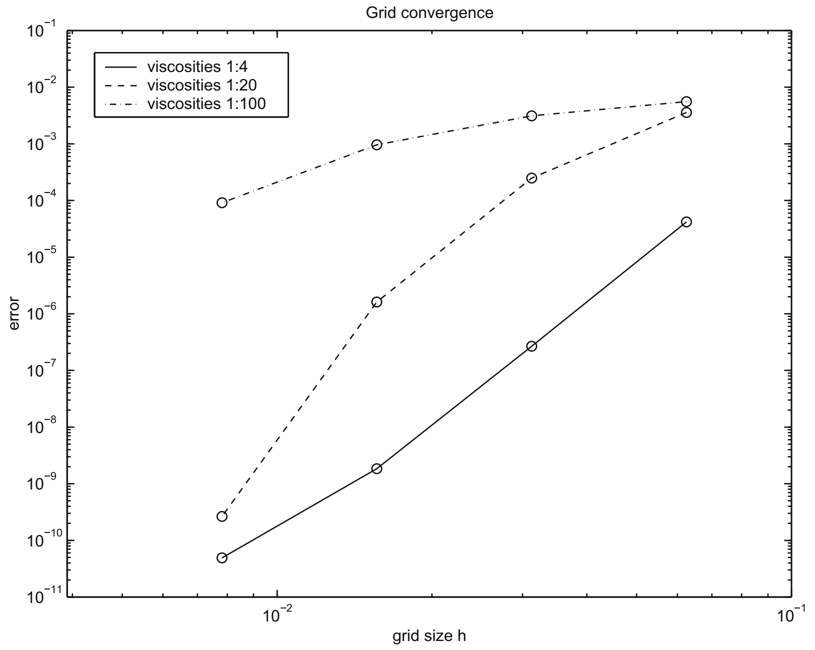

Fig. 4. Grid convergence for Couette channel flows with three different viscosity ratios: 1:4, 1:20, and 1:100.

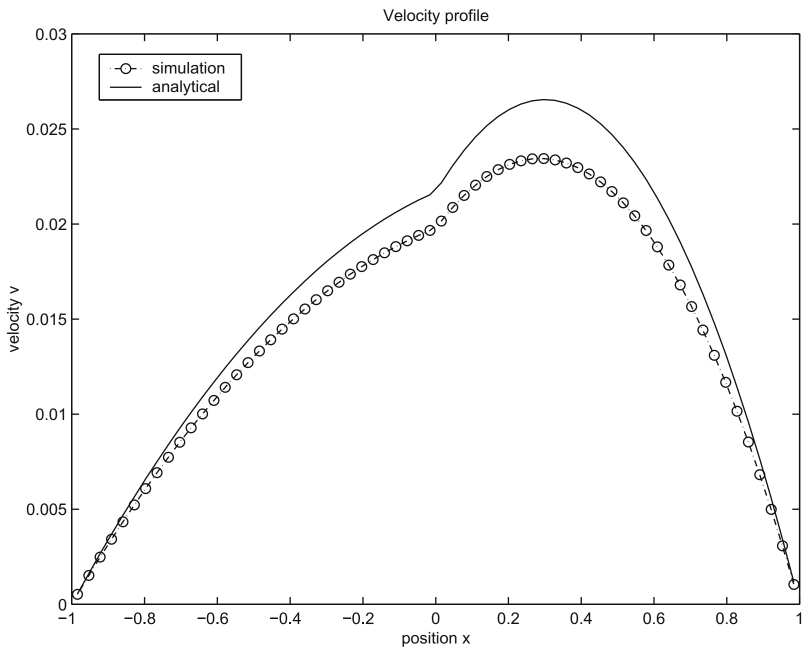

Fig. 5. Velocity profile for a Poiseuille channel flow with viscosities $\mu_1 = 0.4$ (left) and $\mu_2 = 0.1$ (right).

as before. The stationary solution of the Navier–Stokes equations consists of two quadratic velocity profiles matched continuously at the interface

$$v(x) = \begin{cases} \frac{\Delta p_0}{2\mu_2}\left(x^2 - \frac{\mu_1-\mu_2}{\mu_1+\mu_2}x - \frac{2\mu_2}{\mu_1+\mu_2}\right), & x \geqslant 0, \\ \frac{\Delta p_0}{2\mu_1}\left(x^2 - \frac{\mu_1-\mu_2}{\mu_1+\mu_2}x - \frac{2\mu_1}{\mu_1+\mu_2}\right), & x < 0. \end{cases}$$

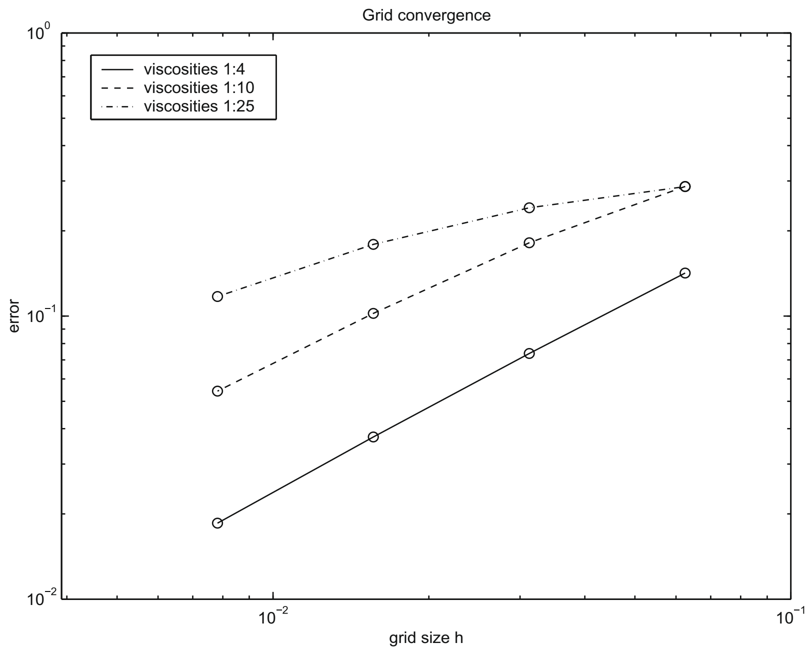

Fig. 6. Grid convergence for three Poiseuille channel flows with different viscosity ratios.

A comparison of the numerical and analytical velocity along the x-direction for a grid with $N = 64$ nodes across the channel after $T = 1000$ time steps is displayed in Fig. 5.

An error analysis (Fig. 6) reveals, apart from convergence, an increase of the absolute error when the viscosity ratio becomes large. Since a similar effect is observed in single phase simulations for growing viscosities, this effect should mainly be attributed to the viscosity dependence of the chosen wall boundary conditions.

6.2. Young–Laplace experiment for a bubble

To validate the implementation of surface tension in the new method we used the Young–Laplace experiment for the pressure difference inside a bubble as a benchmark problem. The pressure p inside a spherical bubble of radius r surrounded by a second fluid is proportional to the surface tension coefficient σ according to the Young–Laplace law

$$p = \frac{2\sigma}{r} = 2\sigma\kappa, \quad \text{mean curvature } \kappa = \frac{1}{2}(\kappa_1 + \kappa_2). \tag{25}$$

We consider a unit cube $[0,1]^3$ containing a bubble of radius $r = 0.25$ at the centre. For the simulations we used an equidistant grid of N nodes in each coordinate direction and periodic boundary conditions at the sides of the cube. The viscosities were chosen equal, $\nu_i = 1/6$, and the densities were $\rho_1 = 1$ and $\rho_2 = 1000$. Simulations were started with zero pressure difference and stopped when the initial fluctuations were reduced sufficiently.

Fig. 7 shows a time series of initial pressure oscillations, which are in general quickly damped until the final value is attained when the viscosities are not exceedingly high. For our choice of viscosity it could be assumed that oscillations had sufficiently subsided when simulations were stopped at $t = 500$ on a grid with $N = 16$ (correspondingly, according to the diffusive scaling, $t = 2000$ for $N = 32$, and $t = 8000$ for $N = 64$). Grid convergence of the pressure error is shown in Fig. 8. We note that the order of convergence is influenced by the order of the curvature reconstruction. This is expected since the Young–Laplace pressure is sensitive to curvature errors produced by the level set method. For the orders 3 and 4 in curvature reconstruction we observe approximately second-order convergence of the pressure.

A numerical artifact observed with many numerical methods is the appearance of spurious currents at the interface. This is true in our case as well. Fig. 9 shows an example for $N = 32$. However, the spurious currents decrease when the grid is refined and convergence in the maximum norm can be established. In particular, for curvature reconstruction of order 3 and 4, first order convergence is achieved.

In [36] it has been verified that the capillary number $Ca = \nu\rho_1 U/\sigma$ corresponding to the velocity U of the spurious currents is constant with respect to the non-dimensional Laplace number $La = r\sigma\rho/\mu^2$. This could also be confirmed in our simulations. We varied the surface tension σ and recorded the maximum norm of the spurious velocities on a grid with $N = 16$. The results using curvature reconstructions of order 2, 3, and 4, respectively, are summarised in Table 1. For higher curvature orders the spurious velocities and hence the capillary number become smaller, but within each of the three sets of simula-

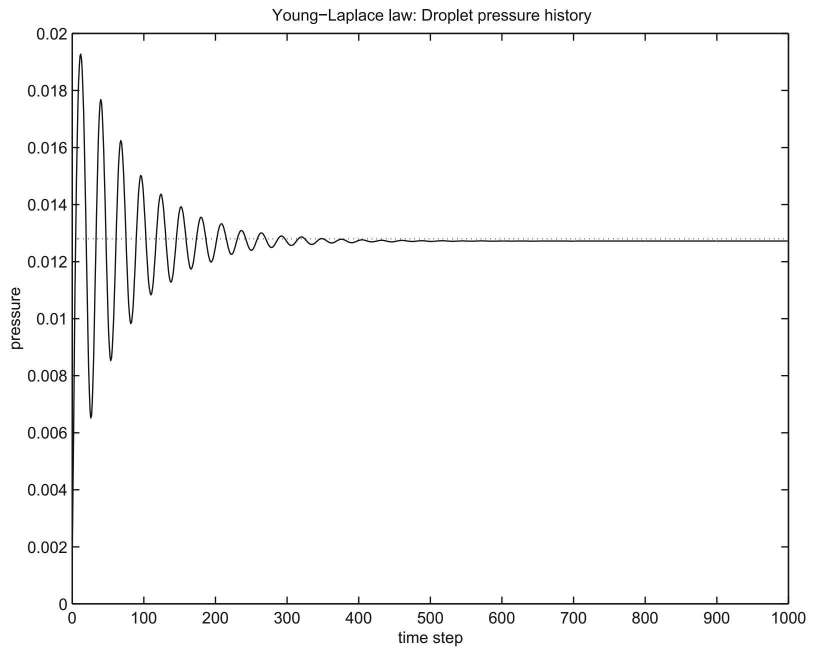

Fig. 7. Pressure history for $N = 32$ after $t = 1000$ time steps.

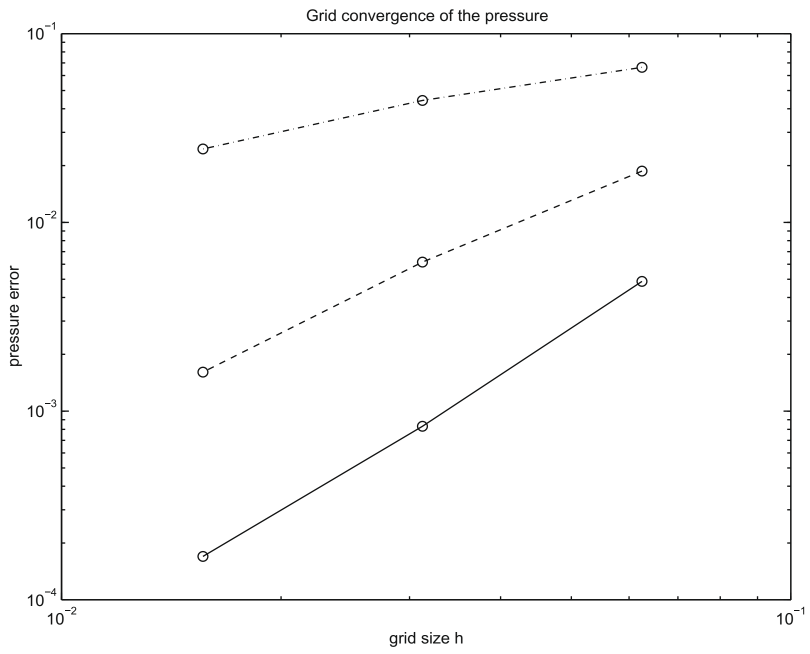

Fig. 8. Grid convergence of the pressure for curvature reconstruction of different orders: order 2 (-·), order 3 (--), order 4 (-).

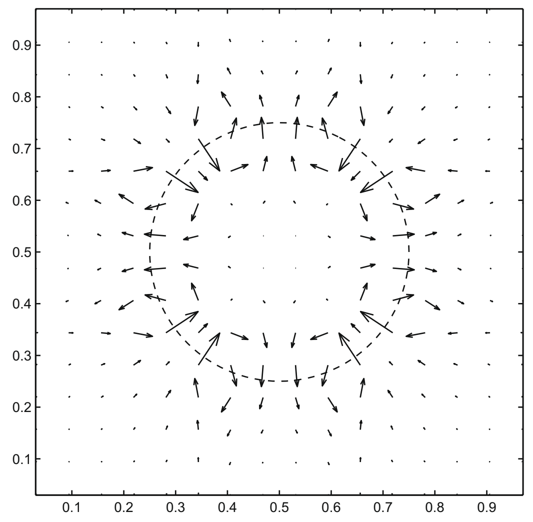
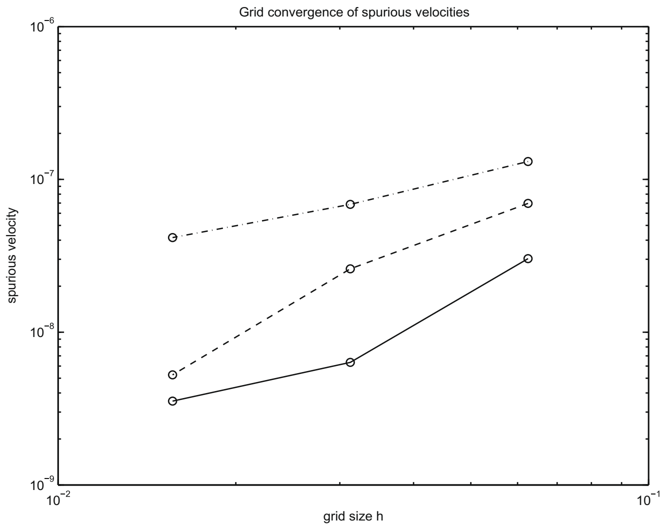

Fig. 9. (Left) Spurious velocities at the interface. (Right) Grid convergence for spurious currents at the interface for surface reconstruction orders 2 (-·), 3 (--) and 4 (-).

Table 1
Independence of *Ca* from σ (and therefore from *La*). Results for three different orders of curvature reconstruction.

σ	Order 2		Order 3		Order 4	
	U	*Ca*	*U*	*Ca*	*U*	*Ca*
10^{-3}	8.1949(−8)	1.3658(−5)	4.3563(−8)	7.2605(−6)	1.8941(−8)	3.1568(−6)
10^{-4}	8.1971(−9)	1.3662(−5)	4.3557(−9)	7.2595(−6)	1.8947(−9)	3.1578(−6)
10^{-5}	8.1974(−10)	1.3662(−5)	4.3557(−10)	7.2595(−6)	1.8947(−10)	3.1578(−6)
10^{-6}	8.1974(−11)	1.3662(−5)	4.3557(−11)	7.2595(−6)	1.8947(−11)	3.1578(−6)

tions *Ca* has approximately the same value. We note that simulations with high density ratios or kinematic viscosity ratios can be successfully performed. However, accuracy deteriorates and spurious currents increase when the ratio attains excessively large values. Moreover, our numerical experiments indicate that the results depend also on internal parameters of the implementation, e.g. the grid refinement of the level set grid with respect to the LBM grid or the order of curvature reconstruction for the calculation of curvature in the level set method. Based on this experience we feel that using third order curvature and a level set grid that coincides with the LBM grid produces the best results. The accuracy of the triangulation used to initialise the level set function turned out not to be of significant importance for the result at the final time.

6.3. Viscous fingering

When a fluid of low viscosity pushes a second fluid of high viscosity owing to a pressure gradient or to gravity, the interface is prone to instabilities which lead to the penetration of the first fluid into the domain of the second in the form of a finger or tongue. A prime example is the use of water to drive oil out of porous rocks (see [41]). An experimental model setup for the study of the so-called *viscous fingering* is the Hele-Shaw cell where the flow domain consists of a narrow gap between two parallel plates. It was first described by Hele-Shaw [11] and the analysis of the fingering instability started with the seminal paper of Saffman and Taylor [28]. Since the governing equations for the gap-averaged velocity are similar to Darcy's law, the Hele-Shaw cell also serves as a 2D model for flow through porous media, where the flow is governed by Darcy's law. The investigation of interface formation in immiscible flows with and without surface tension effects has revealed that this can be used as a generic model problem for analytical studies as well as for benchmarking numerical simulations of free surfaces [7,12,44,23,27].

In our simulations we considered the flow in the narrow gap between the plates and investigated the finger by assuming that the tip can be fitted by an exponential shape as described in [27]. We used a channel $[0,32] \times [-1,1]$ with aspect ratio 16:1 using $16N$ grid points in x-direction, and N in y-direction. At the inlet (outlet), a pressure $p_{\text{in}} = 0.032$ ($p_{\text{out}} = -0.032$) was applied in x-direction to simulate a pressure gradient. At the walls ($y = \pm 1$) we imposed the usual no-slip boundary conditions. In z-direction periodic boundary conditions were applied. An initial interface perturbation in the form of a sine shape, $y(x) = 1 + 0.5\cos(\pi x)$, was prescribed at the beginning. We assumed a driving fluid of viscosity $\nu_1 = 0.1$ and a second fluid of higher viscosity $\nu_2 = 1$. The densities of the fluids were the same, $\rho_1 = \rho_2 = 1$, and a surface tension coefficient $\sigma = 0.016$ was chosen.

Fig. 10 shows the contour of the resulting viscous finger at several time steps. The velocity of the finger tip was $U = 0.032$ and the corresponding capillary number was $Ca = \rho_2 \nu_2\, U/\sigma = 2$. An exponential function is plotted for comparison using an ansatz in the form

$$y(x) = \pm(e^{k(x-x_0))} - \beta),$$

with parameters $\beta = 0.67$, taken from Fig. 7 in [27], and $q = 0.7165$ obtained from the relation

$$2q - \sin 2q + Ca(4q^2 - 4\cos^2 q) = 0,$$

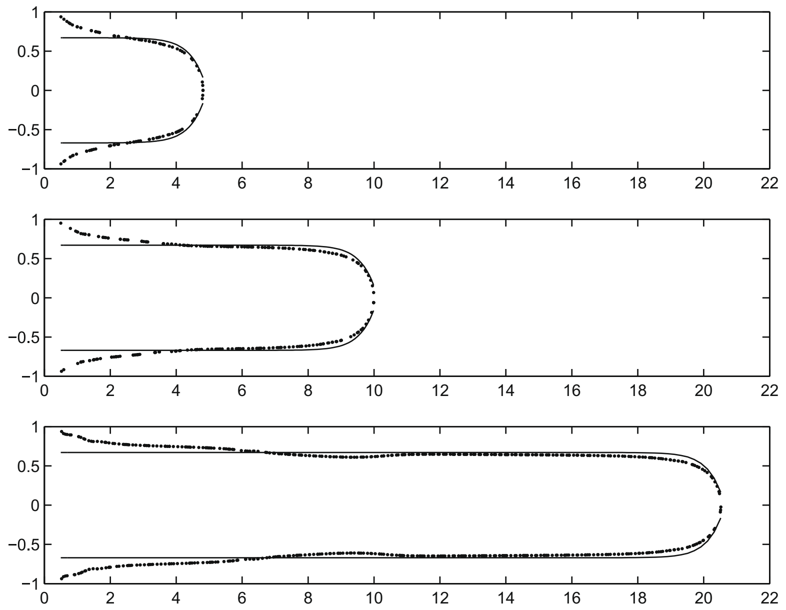

Fig. 10. Viscous fingers at time steps $t = 15{,}000$, 30,000 and 60,000.

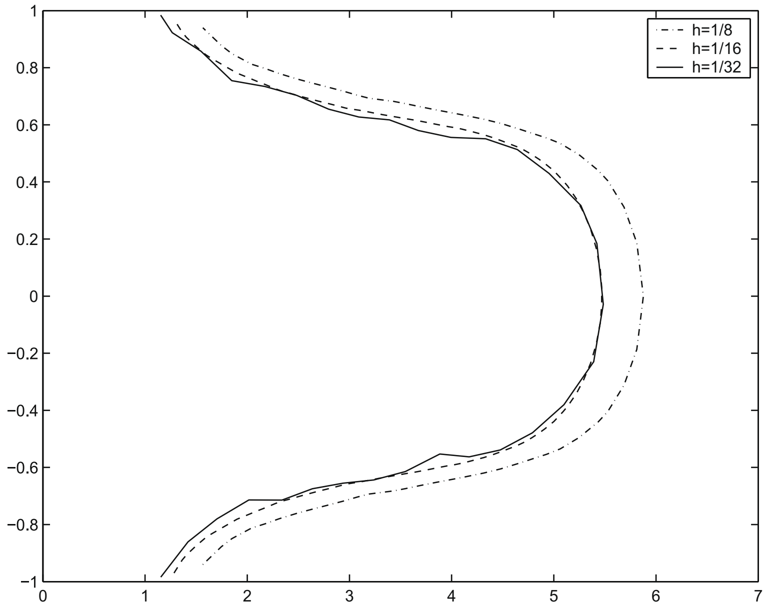

Fig. 11. Grid convergence of the finger shape.

such that $k = q/(1-\beta) = 2.1712$. For x_0 we take the position of the finger tip. In the plots of the finger tips, good agreement with the expected exponential shape is observed. In this simulation the interface experiences strong deformation and stretching. We compared the volume of the two fluid phases over the course of the fingering experiment with the volume influx at the inlet. It revealed that there was excellent agreement of the recorded volume with the predicted volume from the influx with a deviation of less than 1%.

Remark 2. In all the simulations presented here no mass conservation problems were encountered. However, we report that mass loss was observed when studying rising bubbles. In this situation mass correction as described above was necessary. These results will be presented in a subsequent publication.

In the simulation it was furthermore observed that the points on left which touched the walls moved slightly to the right. This numerical artifact was produced by the fact that the wall did not lie on the cell centres but on cell edges. Owing to the domain periodicity in the level set method, the velocity extrapolation did not produce zero velocity of the level set function at the wall. The artifact was corrected in a post-processing step by shifting the finger shape to the left. We note, moreover, that the finger width coincides better with the theoretical value at points further away from the tip. This reflects the fact that the asymptotic analysis leading to the exponential shape assumes an infinitely long finger [27].

To check the correctness of the shape obtained on this rather coarse grid, we computed the results on two finer grids with $N = 32$ and $N = 64$, respectively, and confirmed the grid convergence of the finger shape (Fig. 11).

7. Summary and conclusions

We have developed a lattice Boltzmann method for immiscible two-phase flows which modifies the populations at the interface such that the interface conditions for the macroscopic quantities e.g. velocity u, pressure p and stress S, and surface tension effects are naturally incorporated in the LBM framework. Coupled with the level set method, which handles interface evolution, the new algorithm allows for the computation of flows with complex geometrical features and large viscosity and density differences. Numerical results for typical two-phase flow problems demonstrate the validity of this approach and verify its stability and accuracy.

In contrast to conventional approaches for immiscible multiphase flows and surface tension in the LBM framework, e.g. the colour gradient method of Gunstensen et al. [14,15], or the approach of Shan and Chen [33,34], in the new method, phase boundaries are represented by sharp interfaces separating the different fluid phases. In this way the typical interface smearing effect of these methods is completely avoided. Interfaces are described by the level set function and moved according to the flow in the lattice Boltzmann model using the level set method. In this way several drawbacks of previous methods are overcome. In particular, the colour gradient method is only applicable when density differences are small and it exhibits grid-dependent perturbations of the interface which cannot be removed in the LBM framework [21]. The method of Shan and Chen suffers, among other things, from mass losses. This is also true for the level set method, but various strategies have been presented in the literature which successfully remedy this numerical problem. As an alternative to the mass correction presented here a particle level set method could be used. This will be investigated in future work.

Additionally, the new approach has the computational advantage of smaller memory requirements since only one set of populations is needed in LBM, while in the colour gradient method each phase has its own populations. In the approach of Shan and Chen the potential field has to be stored on the whole 3D grid, whereas we use the memory efficient *narrow-band* technique, which effectively stores and computes the level set function only in a small layer around the interface, for the level set method [1].

In comparison with the hybrid LBM in [22], we mention, in particular, that our new approach is also able to simulate flow in complex geometries with high density and viscosity differences. Moreover, the level set method is well suited for topology changes while this is not easily accomplished using particle methods like front-tracking.

Comparing our method with the free energy method [37,38] we remark, that it is not based on a physical principle of such general applicability as the free energy method. Rather, the starting point of our approach was numerical analysis. We derived a discretisation based explicitly on the governing macroscopic jump conditions for the mesoscopic particle distributions. Hence the conservation laws are fulfilled by construction up to errors due to the discretisation on the finite grid, which is sufficient for a numerical scheme.

Future work in the realm of free-surface multiphase flows will focus on applications of the method to the simulation of Rayleigh–Taylor instability and bubble dynamics including break-up and coalescence of bubbles.

Acknowledgments

M. Junk, K. Steiner and A. Klar acknowledge the support of the Deutsche Forschungsgemeinschaft (Germany) for this work by the grants *JU 440/1-3*, *STE 871/1* and *KL 1105/9*, respectively.

References

[1] D. Adalsteinsson, J.A. Sethian, A fast level set method for propagating interfaces, J. Comput. Phys. 118 (2) (1995) 269–277.
[2] D. Adalsteinsson, J.A. Sethian, The fast construction of extension velocities in level set methods, J. Comput. Phys. 148 (2) (1999) 2–22.
[3] P. Bhatnagar, E. Gross, M. Krook, A model for collision processes in gases: I. Small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (1954) 511.
[4] M. Bouzidi, M. Firdaouss, P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, Phys. Fluids 13 (11) (2001) 3452–3458.
[5] A. Caiazzo, Asymptotic analysis of lattice Boltzmann method for fluid–structure interaction problems, Ph.D. Thesis, Kaiserslautern (Germany) and Pisa (Italy), 2007.
[6] A. Caiazzo, Analysis of lattice Boltzmann nodes initialisation in moving boundary problems, Prog. Comput. Fluid Dyn. 8 (2008) 3–10.
[7] J. Casademunt, Viscous fingering as a paradigm of interfacial pattern formation: recent results and new challenges, Chaos 14 (3) (1997) 809–824.
[8] S. Chen, G.D. Doolenm, Lattice Boltzmann method for fluid flows, Ann. Rev. Fluid Mech. 30 (1998) 329–364.
[9] M.P. Do Carmo, Differential Geometry of Curves and Surfaces, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.
[10] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1992) 2182–2189.
[11] H.J.S. Hele-Shaw, On the motion of a viscous fluid between two parallel plates, Nature 58 (1898) 34–36.
[12] J.M. Guevara-Jordan, J. Glimm, A mixed finite element method for Hele-Shaw cell equations, Comput. Geosci. 1 (1997) 35–58.
[13] A.K. Gunstensen, Lattice-Boltzmann studies of multiphase flow through porous media, Ph.D. Thesis, MIT, 1992.
[14] A.K. Gunstensen, D.H. Rothman, S. Zaleski, G. Zanetti, Lattice Boltzmann model of immiscible fluids, Phys. Rev. A 43 (8) (1991) 4320–4327.
[15] A.K. Gunstensen, D.H. Rothman, Microscopic modeling of immiscible flows in three dimensions by a lattice Boltzmann method, Europhys. Lett. 18 (2) (1998) 157–161.
[16] Z. Guo, C. Zheng, B. Shi, An extrapolation method for boundary conditions in lattice Boltzmann method (sic), Phys. Fluids 14 (6) (2002) 2007–2010.
[17] X. He, L.S. Luo, Lattice Boltzmann model for the incompressible Navier–Stokes equation, J. Stat. Phys. 88 (1997) 927–944.
[18] M. Junk, A. Klar, L. Luo, Asymptotic analysis of the lattice Boltzmann equation, J. Comput. Phys. 210 (2) (2005) 676–704.
[19] M. Junk, Z. Yang, One-point boundary condition for the lattice Boltzmann method, Phys. Rev. E 72 (2005).
[20] M. Junk, Z. Yang, Outflow boundary conditions for the lattice Boltzmann method, Prog. Comput. Fluid Dyn. 8 (2008) 38–48.
[21] D. Kehrwald, Numerical analysis of immiscible lattice BGK, Dissertation, Fachbereich Mathematik, Universität Kaiserslautern, 2002.
[22] P. Lallemand, L.S. Luo, Y. Peng, A lattice Boltzmann front-tracking method for interface dynamics with surface tension in two-dimensions, J. Comput. Phys. 226 (2) (2007) 1367–1384.
[23] J.W. McLean, P.G. Saffman, The effect of surface tension on the shape of fingers in a Hele Shaw cell, J. Fluid Mech. 102 (1980) 455–469.
[24] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, New York, 2003.
[25] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[26] S.J. Osher, G. Tryggvason, Numerical methods for multiphase flows, J. Comput. Phys. (Special Issue) (16) (2001) 249–762.
[27] D.A. Reinelt, P.G. Saffman, The penetration of a finger into a viscous fluid in channel and tube, SIAM J. Sci. Stat. Comput. 6 (3) (1985) 542–561.
[28] P.G. Saffman, Sir G. Taylor, The penetration of a fluid into a porous medium or Hele-Shaw cell containing a more viscous liquid, Proc. Roy. Soc. A 245 (1958) 312–329.
[29] K. Sankaranarayanan, I.G. Kevrekidis, S. Sundaresan, J. Lu, G. Tryggvason, A comparative study of lattice Boltzmann and front-tracking finite-difference methods for bubble simulations, Int. J. Multiphase Flow 29 (1) (2003) 109–116.
[30] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Ann. Rev. Fluid Mech. 31 (1999) 567–603.
[31] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1996.
[32] J.A. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface method, J. Comput. Phys. 163 (2000) 489–528.
[33] X. Shan, H. Chen, Lattice Boltzmann model for simulating flows with multiple phases and components, Phys. Rev. E 47 (3) (1993) 1815.
[34] X. Shan, H. Chen, Simulation of nonideal gases and liquid–gas phase transition by the Lattice Boltzmann equation, Phys. Rev. E 49 (4) (1994) 2941.
[35] W. Shyy, H.S. Udaykumar, M.M. Rao, R.W. Smith, Computational Fluid Dynamics with Moving Boundaries, Taylor & Francis, 1996.
[36] A. Smolianski, Finite-element/level-set/operator-splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces, Int. J. Numer. Methods Fluids 48 (3) (2005) 231–269.
[37] M.R. Swift, W.R. Osborn, J.M. Yeomans, Lattice Boltzmann simulation of nonideal fluids, Phys. Rev. Lett. 75 (1995) 830–833.
[38] M.R. Swift, E. Orlandini, W.R. Osborn, J.M. Yeomans, Lattice Boltzmann simulations of liquid–gas and binary fluid systems, Phys. Rev. E 54 (5) (1996) 5041–5052.
[39] Sauro Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Clarendon Press, Oxford, 2001.

[40] M. Sussmann, A second order coupled level set and volume of fluid for computing growth and collapse of vapor bubbles, J. Comput. Phys. 187 (2003) 110–136.
[41] G. Tryggvason, H. Aref, Numerical experiments on Hele Shaw flow with a sharp interface, J. Fluid Mech. 136 (1983) 1–30.
[42] G. Tryggvason, B. Brunner, A. Esmaeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2) (2001) 708–759.
[43] A.K. Vaikuntam, Numerical Estimation of Surface Parameters by Level Set Methods, PhD Thesis, Tu Kaiserslautem, 2008.
[44] N. Whitaker, Some numerical methods for the Hele-Shaw equations, J. Comput. Phys. 111 (1993) 81–88.
[45] D.A. Wolf-Gladow, Lattice-Gas Cellular Automata and Lattice Boltzmann Models, An Introduction, Springer, 2000.
[46] Q. Zou, X. He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, Phys. Fluids 9 (6) (1997) 1591–1597.