

Implementation of a lattice Boltzmann method for immiscible two-phase flow simulations using the level set method

Lorenz Hufnagel, Daniel Zint

Chair for System-Simulation, Friedrich Alexander Universität Erlangen-Nürnberg
Cauerstr. 11, 91058 Erlangen, Germany
lorenz.hufnagel@fau.de, daniel.zint@fau.de

Abstract. We implemented the lattice Boltzmann method for immiscible multiphase flow simulations and combined it with the level set method according to the paper of G. Thömmes et al. [1]. The level set method is used to compute the movement of the interface between the two phases. The input for the level set method is computed by the lattice Boltzmann method. We verified our implementation with different test cases.

Key words. Lattice Boltzmann method, Level set method, Two-phase, Multiphase

1. Introduction. In engineering the immiscible multiphase flow problem often needs to be considered. For example, bubble dynamics are decisive for the design of chemical reactors [5]. Also, fingering in oil recovery is a common multiphase problem.

In the following we will restrict ourselves to two fluid phases, but the procedure is also applicable for multiple phases. There are different approaches to solve the immiscible two-phase problem. The most spread among these is the colour gradient method of Gunstensen and Rothman [6, 7, 8]. The downside of this method is that it is only applicable for small density and viscosity differences. Another approach is the concept of *interaction potentials* by Shan and Chen [9, 10]. This method actually models miscible fluids and can only approximately describe the behavior of immiscible fluids. Furthermore, there are free surface methods which only take one fluid into account for computation [11, 12, 13]. They work best for large viscosity and density ratios.

In this publication we use a hybrid lattice Boltzmann level set method. The motion of the interface is modeled by the level set method. The input data for the level set method is given by the lattice Boltzmann method (LBM), which is applied independently for each fluid. This method has the advantage that it allows large ranges of viscosity- and density-ratios between the fluid phases, while describing a sharp interface.

The incompressible two-phase flow problem can be described mathematically with the Navier-Stokes equations for two incompressible fluids eqs. (1.1) and (1.2), see [3]. To apply them, we distribute our domain into three parts: Ω_1 and Ω_2 describe the two fluids and $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ is the interface in-between (fig. 1). The material properties in each fluid are constant and the interface is assumed to be sharp. Thus, the overall material properties are discontinuous and the fluids will not mix.

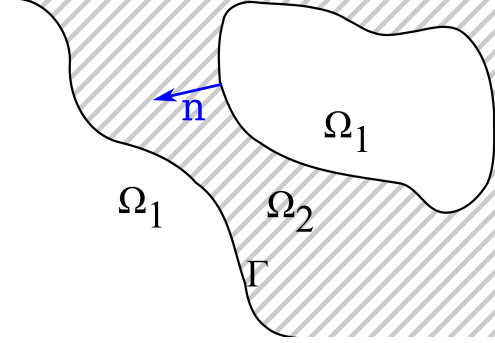


Figure 1. Two fluid domains Ω_i separated by an interface Γ . \vec{n} denotes the surface normal.

The change of momentum within each fluid can be expressed as

$$(1.1) \quad \nabla \cdot \vec{u}_i = 0 \text{ in } \Omega_i,$$

$$(1.2) \quad \frac{\partial \vec{u}_i}{\partial t} + (\vec{u}_i \cdot \nabla) \vec{u}_i = -\frac{1}{\rho_i} \nabla p_i + \nu_i \nabla^2 \vec{u}_i \text{ in } \Omega_i.$$

Here, \vec{u}_i is the velocity, p_i the pressure, ν_i the kinematic viscosity and ρ_i the mass density.

As boundary conditions and initial conditions we use,

$$(1.3) \quad \vec{u}_i(x, t) = 0, \text{ on } \partial\Omega_i \setminus \Gamma,$$

$$(1.4) \quad \vec{u}_i(x, 0) = \vec{u}_{0,i}(x), \text{ in } \Omega_i.$$

At the interface Γ we have a no-slip condition, i.e. continuity of the velocities between the fluid phases and the balance of viscous shear stress with surface tension. Therefore, we get the jump conditions,

$$(1.5) \quad [\vec{u}] = 0,$$

$$(1.6) \quad [2\mu S - p\mathbf{I}] \cdot \vec{n} = 2\sigma\kappa\vec{n}.$$

Here, S is the shear rate tensor $S = \frac{1}{2} (\nabla \vec{u} + \nabla \vec{u}^T)$, σ the surface tension, $\mu = \rho\nu$ the dynamic viscosity and κ the curvature of the interface with respect to its outer normal \vec{n} . Brackets denote the jump of a quantity at the interface,

$$(1.7) \quad [q](\vec{x}) = \lim_{\epsilon \rightarrow 0} (q(\vec{x} - \epsilon\vec{n}) - (q(\vec{x} + \epsilon\vec{n}))) \text{ with } \vec{x} \in \Gamma.$$

In fig. 1 this is the jump from Ω_1 to Ω_2 . For the other direction the sign changes.

This paper is organized as follows: Section 2 gives an introduction to the LBM and the level set method and explains the coupling of these. In section 3 we present our results and finally in section 4 we draw a conclusion.

2. Numerical Methods. The algorithm we describe in this paper uses the Lattice Boltzmann Method (LBM) and the level set method. The LBM covers the description of the velocity and density field for each fluid, as it can be shown to reproduce the Navier-Stokes equations in the asymptotic limit with Chapman-Enskog expansion [16]. The level set method uses a velocity field to model the movement of the interface. We will shortly introduce both methods and show how they can be coupled.

2.1. The Lattice Boltzmann Method. To solve the incompressible Navier-Stokes equations, we use the lattice Boltzmann method. This method describes the evolution of the particle density $f(\vec{x}, \vec{v}, t)$ of the Boltzmann equation in phase space, with (\vec{x}, \vec{v}) as the phase space variables over time. For the LBM algorithm the geometrical domain is discretized with a cartesian grid and the velocity space with a restricted number of velocities adapted to this grid [14, 15, 16, 17]. We use the D2Q9 model in 2D, which has 9 velocity vectors on a plane grid with unit spacing including one zero velocity (fig. 2).

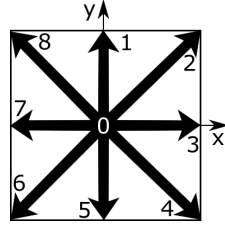


Figure 2. The vectors of one cell in LBM.

The vectors are given by the columns of the matrix,

$$(2.1) \quad c = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 0 & 1 & 1 & 0 & -1 & -1 & -1 & 0 & 1 \end{bmatrix}.$$

The corresponding particle distributions are denoted by $f_i(\vec{x}, t) = f(\vec{x}, \vec{c}_i, t)$. We get the density and velocity from the moments of the particle distribution functions with,

$$(2.2) \quad \rho(\vec{x}, t) = \sum_{i=1}^9 f_i(\vec{x}, t), \quad \vec{u} = \sum_{i=1}^9 f_i(\vec{x}, t) \vec{c}_i,$$

while the pressure is related to the lattice density ρ as $p = \rho \cdot \rho / 3$.

The LBM algorithm alternates between a collision step and propagation step, where for the collision step we use the Bhatnagar-Gross-Krook (BGK) collision operator [18],

$$(2.3) \quad f_i^+(\vec{x} + \vec{c}_i, t + 1) = f_i(\vec{x}, t) - \frac{1}{\tau}(f_i - f_i^{eq}) + G_i.$$

The parameter τ is the relaxation parameter for the collision and controls the kinematic viscosity in lattice units $\nu = \frac{1}{6}(2\tau - 1)$. Note, that we assume the cell spacing Δx and the time step Δt to be unity here. The lattice speed of sound equals to $c_s = \frac{1}{\sqrt{3}}$. G_i is an

additional force, e.g. gravity. In our case we set $G_i = 0$. Furthermore, we use the equilibrium distribution,

$$(2.4) \quad f_i^{eq}(\rho, \vec{u}) = f_i^*(\rho + 3\vec{c}_i \cdot \vec{u} + \frac{9}{2}(\vec{c}_i \cdot \vec{u})^2 - \frac{3}{2}\vec{u} \cdot \vec{u})$$

with the corresponding D2Q9 weight factors,

$$f_i^* = \begin{cases} \frac{4}{9}, & i = 0 \\ \frac{1}{9}, & i = 1, 3, 5, 7 \\ \frac{1}{36}, & i = 2, 4, 6, 8 \end{cases}$$

The propagation step is denoted by,

$$(2.5) \quad f_i(\vec{x} + \vec{c}_i, t + 1) = f_i^+(\vec{x}, t + 1),$$

where two types of boundary conditions are used. To simulate static walls we use bounce-back boundary conditions,

$$(2.6) \quad \tilde{f}_i = f_{i^*}^+(\vec{x}_b, t), \text{ with } \vec{x}_b \in \delta\Omega,$$

where i^* is the index of the opposite direction $\vec{c}_{i^*} = -\vec{c}_i$. To simulate a solid boundary with nonzero tangential velocity, we use moving no-slip boundary conditions,

$$(2.7) \quad \tilde{f}_i = f_{i^*}^+(\vec{x}_b, t) + 6f_{i^*}^* \vec{c}_{i^*} \vec{v}, \text{ with } \vec{x}_b \in \delta\Omega,$$

where \vec{v} is the speed of the moving wall. To model infinitely long domains we use periodic boundary conditions,

$$(2.8) \quad \tilde{f}_i(\vec{x}_a, t) = f_i^+(\vec{x}_b, t),$$

where x_a and x_b are points of opposite boundaries.

One time step in the LBM algorithm is achieved by the following sub-steps:

1. Collision step: $f_i^+ = f_i - \frac{1}{\tau}(f_i - f_i^{eq}) + G_i$
2. Propagation step: $f_i(\vec{x} + \vec{c}_i, t + 1) = f_i^+(\vec{x}, t)$, for $\vec{x} \in \Omega$
3. Boundary treatment: $f_i(\vec{x} + \vec{c}_i, t + 1) = f_i(\vec{x} + \vec{c}_i, t)$, if $\vec{x} \notin \Omega$

We remark that the propagation step is the only non-local operation in the LBM algorithm, as it accesses nearest neighbors. This allows for highly efficient parallel implementations of the LBM.

2.2. The Level Set Method. We deal with the advection of the interface between the two fluids by using the level set method. The method captures the surface that represents the interface with a continuous signed distance function, where the interface is implicitly given as the zero level set.

Let Γ_t be our interface at time t . Within the level set method, Γ_t is the zero isosurface of the level set function φ . Thus, for $\vec{x} \in \Gamma_t$ we have $\varphi(\vec{x}, t) = 0$, in direction of the surface

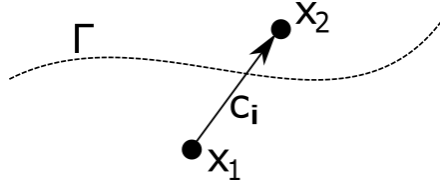


Figure 3. Link between two points in different fluids.

normal we get $\varphi > 0$, and in the other direction we get $\varphi < 0$. For this level set function we get the level set equation introduced by Osher and Sethian in [19] for the evolution of surfaces,

$$(2.9) \quad \frac{\partial \varphi}{\partial t} + \vec{v} \cdot \nabla \varphi = 0,$$

with the velocity $\vec{v} = \vec{u}(\vec{x}(t))$ for $\vec{x}(t) \in \Gamma_t$, which we obtain from the LBM. It is easily visible that the normal of the surface is determined as $\vec{n}(x) = \frac{\nabla \varphi(x)}{\|\nabla \varphi(x)\|}$ and accordingly the curvature as the divergence of the normal $\kappa = \nabla \cdot \nabla \varphi = \Delta \varphi$.

To solve this equation we use the Toolbox of Ian Mitchell [20] for Matlab. This toolbox takes as input the velocity field we get from the lattice Boltzmann method and computes the movement of the interface. The toolbox also provides functions to compute the normal and the curvature of the surface. We will need this information for the coupling of the methods.

2.3. Coupling of LBM and level set method. The lattice Boltzmann method requires a boundary treatment at the interface which implements the macroscopic jump conditions (1.6). For the sake of simplicity, we do not write arrows for every vector, i.e., $\vec{u} \rightarrow u$. One should consider that every position, direction, and velocity is two-dimensional.

For the coupling, we use a bounce back type boundary condition that prescribes the velocity as Dirichlet value as described in [1, p. 1146]. Consider a point $x_2 \in \Omega_2$ that is next to the phase interface, where the link c_i crosses the interface, see fig. 3. We set,

$$(2.10) \quad f_i(x_2, t + 1) = f_i^+(x_2, t) + 6f_i^* c_i \cdot \tilde{u} + R_i.$$

Here, \tilde{u} describes the 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 ensures the jump conditions of the normal stress and corrects the error that is introduced by the first order accurate bounce back treatment above,

$$(2.11) \quad R_i = 6f_i^* \left(q(q - 1)\Lambda_i : [S] - (q - \frac{1}{2})\Lambda_i : S_2 \right),$$

with

$$(2.12) \quad \Lambda_i = c_i \otimes c_i - \frac{1}{2}(c_i \cdot c_i)\mathbf{I}.$$

That is, an orthogonal projection of the column vectors of the (symmetric) deviatoric shear rate jump tensor on the link direction. This projection is weighted with the interface distance and a compensation term for the interpolation error. This error arises, when the interface is not on the exact center between two LBM grid cells. The tensor product is $a \otimes b = ab^T$ and the double contraction $A : B = \text{trace}(AB^T)$. Furthermore, we use the following first-order estimate to obtain the shear rate from the non-equilibrium parts of the distributions, see also [22],

$$(2.13) \quad S_k = -\frac{3}{2\tau} \sum_{i=1}^9 c_i \otimes c_i (f_i - f_i^{eq})(t, x_k).$$

Now we have,

$$(2.14) \quad \Lambda_i : [S] = ([S] : n \otimes n)((n \cdot c_i)^2 - \frac{1}{2}(c_i \cdot c_i)) + 2([S] : n \otimes t)(n \cdot c_i)(t \cdot c_i),$$

which is obtained by a coordinate transformation of (2.11) onto a basis (n, t) of normal and tangential vector to the interface. From the fluid mechanical jump condition (1.6) and the general relation $[\mu S] = \bar{\mu}[S] + [\mu]\bar{S}$ it can be seen,

$$(2.15) \quad [S] : n \otimes n = \frac{1}{2\bar{\mu}}([p] + 2\sigma\kappa) - \frac{[\mu]}{\bar{\mu}}\bar{S} : n \otimes n,$$

$$(2.16) \quad [S] : n \otimes t = -\frac{[\mu]}{\bar{\mu}}\bar{S} : n \otimes t.$$

Here $\bar{\mu} = (\mu_1 + \mu_2)/2$ and $\bar{S} = (S_1 + S_2)/2$ are averaged quantities, κ the interface curvature, and $[\mu] = \mu_2 - \mu_1$ the viscosity jump.

With respect to the given definition of the discrete lattice velocities (2.1, see also [2]), the pressure jump is approximated by,

$$(2.17) \quad [p] \approx \frac{1}{3}(\rho(x_2, t)\varrho_2 - \rho(x_1, t)\varrho_1).$$

and thus, all variables for the boundary conditions are defined.

A refill method is still missing. Due to the fact that the interface moves, some cells change their fluid type. We use an equilibrium/non-equilibrium refill as it is described in [21] consisting of the following four steps:

- 1 Extrapolate density and velocity from interior neighbors.
- 2 Compute the corresponding equilibrium.
- 3 Copy the non-equilibrium part from a direct interior neighbor.
- 4 Reinitialize by adding equilibrium and non-equilibrium parts.

Suppose we choose an inward pointing direction c_i in a boundary point $x \in \Omega_1$, e.g. the direction with the largest angle to the surface normal. Then the density is extrapolated by,

$$(2.18) \quad \tilde{\rho}(x) = 3\rho(x + c_i, t) - 3\rho(x + 2c_i, t) + \rho(x + 3c_i, t).$$

Figure 4 shows the cells we use for the extrapolation. In [1] the velocity is extrapolated

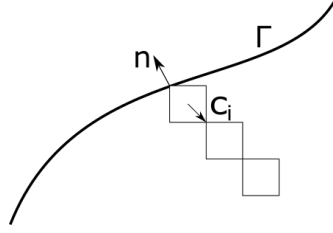


Figure 4. Cells that are used for the refill method.

using the extension velocity of the interface [4]. In contrast, we use the formula given in [23], analogous to the formula for the density,

$$(2.19) \quad \tilde{u}(x) = 3u(x + c_i, t) - 3u(x + 2c_i, t) + u(x + 3c_i, t),$$

which yields similar results, as reported in [23]. We can now compute the equilibrium part $f_i^{eq}(\tilde{\rho}, \tilde{u})$ according to eq. (2.4) and add it to the non-equilibrium part which we copy from a nearest neighbor,

$$f_i(\vec{x}, t + 1) = f_i^{eq}(\tilde{\rho}, \tilde{u}) + f_i^{neq}(x + c_i, t).$$

This completes the refill algorithm.

Finally, the complete coupling of the lattice Boltzmann and the level set method reads (see also fig. 5):

1. Create the initial interface Γ_0 .
2. Run the level set code once to create the surface description for LBM.
3. Run the LBM for a prescribed time interval, using normal, curvature and interface position from the level set
4. Pass the velocity field to the level set code and run it for the same time interval
5. Repeat step 3 and 4 until the end of the simulation.

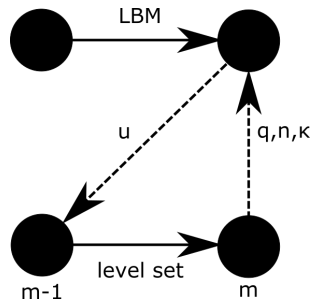


Figure 5. Data exchange between LBM and level set method.

3. Results and Examples. To verify the correctness of our implementation we first built a Couette flow with two stacked fluid layers. Secondly, we used the Young-Laplace experiment for a bubble to validate the implementation of surface tension.

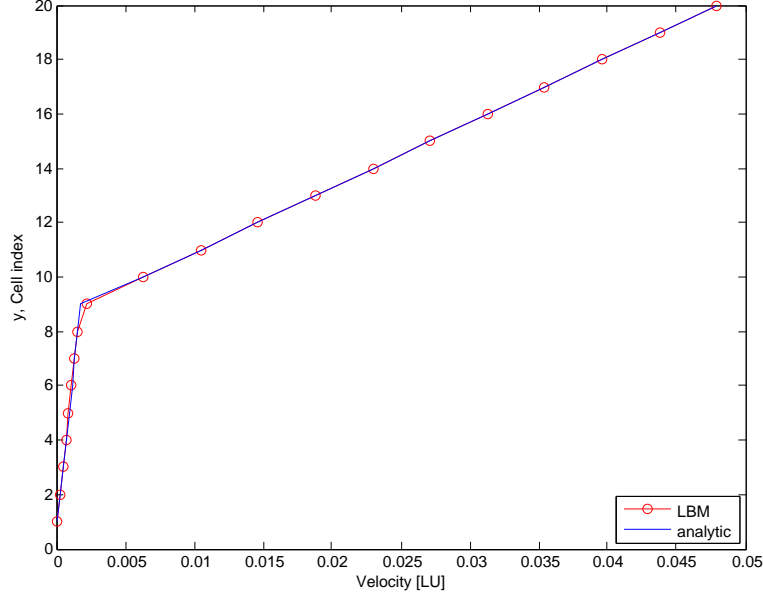


Figure 6. Two horizontal fluid layers Ω_i separated by an interface. The top boundary is moving horizontally, while the lower one is at rest

3.1. Couette channel flow. The Couette channel flow consists of a lower wall $y = 0$ at rest and an upper wall $y = 1$ that is moving tangentially with a prescribed constant velocity. At these walls no-slip boundary conditions hold. Along the x -direction periodic boundary conditions are prescribed. The two fluids layers are oriented such that their interface is parallel to the no-slip walls, and no external pressure gradient is applied. For this flow scenario an analytic solution can be derived from the Navier-Stokes equations, namely they simplify to $\frac{\partial}{\partial y} \left(\eta \frac{\partial u}{\partial y} \right) = 0$. In combination with the no-slip conditions at top and bottom wall this means that the horizontal velocity has a linear profile with a kink at the fluid interface. The respective slopes between both phases are related by $\frac{\partial_y u^{(1)}}{\partial_y u^{(2)}} = \frac{\eta^{(2)}}{\eta^{(1)}}$.

We used identic densities $\rho_1 = \rho_2 = 1$ and 5×20 cells along $x \times y$ direction, as well as a LBM timestep of $\Delta t = 0.0025$. In fig. 6 we show the simulated values and the analytic solution for viscosities $\mu^{(1)} = 1/2$ and $\mu^{(2)} = 10$, where interface was located at $y = 0.42$. One can see that the values agree very well and in fact an overall relative error below 0.01% was attained after 1000 LBM iterations.

Figure 7 shows a second scenario with 5×10 cells along $x \times y$ direction and viscosities $\mu^{(1)} = 1$ and $\mu^{(2)} = 0.2$ with the interface located at $y = 0.75$. Again the numerically approximated values agree very well with the analytic ones and the simulation converged at a relative error of $1.4 \cdot 10^{-4}$ after 2000 LBM iterations. We conclude from this that the implemented LBM boundary conditions successfully model viscosity jumps at the fluid interface. Furthermore,

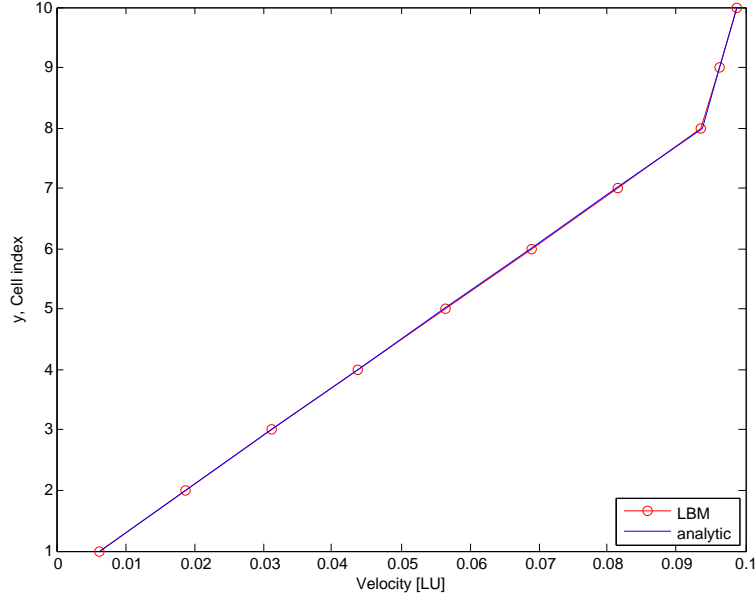


Figure 7. Two fluid domains Ω_i in a Couette flow

we were able to use viscosity ratios greater than 100 : 1 in our tests.

3.2. Young-Laplace experiment for a bubble. The Couette flow scenario does not account for the influence of surface tension as the fluid interface is not curved. If it would get distorted, surface tension would have a stabilizing effect although. Hence, we use the Young-Laplace law for the pressure drop across curved surfaces to validate the implementation of surface tension. Due to surface tension the pressure inside of convex surfaces is generally higher than on the outside and the pressure drop Δp of a bubble of radius r is proportional to the surface tension coefficient σ according to the following relationship,

$$(3.1) \quad \Delta p = \frac{2\sigma}{r} = 2\sigma\kappa.$$

Figure 8 shows the simulation results we obtained for a 40×40 cells grid over a $[0, 1] \times [0, 1]$ -domain with periodic boundaries. At the center, a sphere of radius 0.25 was placed with $\rho_2 = 1.1$ while the exterior contained a fluid with $\rho_1 = 1$. It is visible, how the approximated pressure attains the analytical solution in a oscillatory motion with decaying amplitude, until both values agree. The algorithm needed 1000 LBM-iterations, where every ten iterations a call to the level-set routine was executed. Both fluids had an identical viscosity of $\nu = 1/6$. We noted that the velocity close to the interface contained erroneous components, which lead to a slight distortion of the spherical shape. We explain those fluxes as inaccuracies stemming from the used level set subroutines, according to the analysis shown in [2]. Additionally, we want to state that we were not able to attain stable results for surface tension coefficients $\sigma > O(10^{-3})$. Large density ratios were limited to big viscosity regimes or refined spatial resolution in our tests.

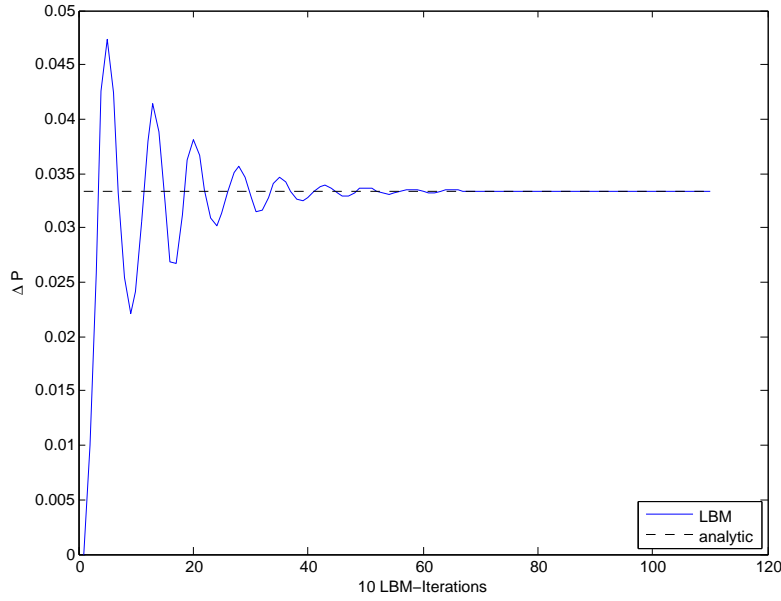


Figure 8. *The pressure drop of a spherical bubble over LBM iterations*

4. Conclusion. We implemented a hybrid lattice Boltzmann method that uses the level set method to model the movement of the interface. The implementation was done in Matlab for a two dimensional case. We adapted the formulas given in [1] to fit in our environment. Finally, we verified our implementation with different test cases.

A more vivid possible application of the algorithm we presented is shown in fig. 9, which shows the transient movement of an oil drop in a lid driven cavity. It is a known problem that level set methods tend to suffer from mass losses [1], but still a very good insight into the detailed flow behavior is possible. Accordingly, a variety of two-phase flow scenarios is possible for simulation with the presented method.

The authors want to thank Ian Mitchell for providing the toolbox for level set methods.

REFERENCES

- [1] G. THÖMMES, J. BECKER, M. JUNK, A.K. VAIKUNTAM, D. KEHRWALD, A. KLAR, K. STEINER, A. WIEGMANN, A lattice Boltzmann method for immiscible multiphase flow simulations using the level set method, *J. Comput. Phys.* 228 (2009), 1139-1156.
- [2] J. BECKER, M. JUNK, D. KEHRWALD, G. THÖMMES, Z. YANG, A combined lattice BGK/level set method for immiscible two-phase flows, *Computers and Mathematics with Applications*, 58(5):950964, 2009.
- [3] J. FERZIGER, M. PERIC., *Computational methods for fluid dynamics*. Springer Science & Business Media, ISBN 978-3540420743 (2012).
- [4] D. ADALSTEINSSON, J.A. SETHIAN, The fast construction of extension velocities in level set methods, *J. Comput. Phys.* 148 (2) (1999) 222.
- [5] G. TRYGGVASON, R. SCARDOVELLI, S. ZALESKI, *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*, Cambridge University Press, ISBN 9780521782401 (2011)

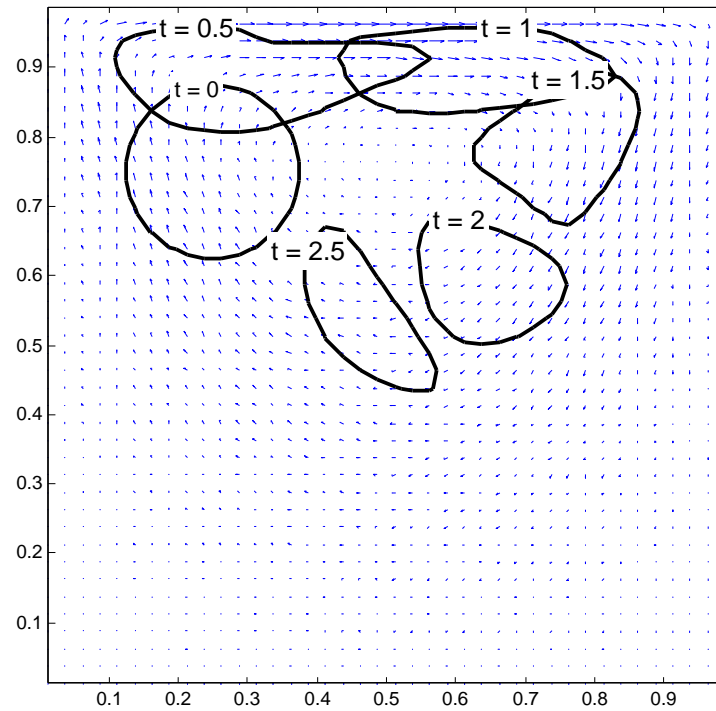


Figure 9. The transient flow and deformation of a bubble in a lid driven cavity. The black lines show the interface at different points of time and the arrows display the velocity field.

- [6] A.K. GUNSTENSEN, D.H. ROTHMAN, S. ZALESKI, G. ZANETTI, Lattice Boltzmann model of immiscible fluids, *Phys. Rev. A* 43 (8) (1991) 4320-4327.
- [7] A.K. GUNSTENSEN, D.H. ROTHMAN, Macroscopic modeling of immiscible flows in three dimensions by a lattice Boltzmann method, *Europhys. Lett.* 18 (2) (1998) 157-161.
- [8] K. SANKARANARAYANAN, I.G. KEVREKIDIES, 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.
- [9] X. SHAN, H. CHEN, Lattice Boltzmann model for simulation flows with multiple phases and components, *Phys. Rev. E* 47 (3) (1999) 567-603.
- [10] 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.
- [11] C. KÖRNER, M. THIES, T. HOFMANN, N. THÜREY, U. RÜDE, Lattice Boltzmann Model for Free Surface Flow for Modeling Foaming, *J. Stat. Phys.* 121 (2005) 179-196.
- [12] R. SCARDOVELLI, S. ZALESKI, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* 31 (1999) 567-603.
- [13] C.W. HIRT, J.P. SHANNON, Free-surface stress conditions for incompressible-flow calculations, *J. Comput. Phys.* 2 (1968) 403-411.
- [14] S. CHEN, G.D. DOOLEN, Lattice Boltzmann method for fluid flows, *Ann. Rev. Fluid Mech.* 30 (1998) 329-364.
- [15] X. HE, L.S. LUO, Lattice Boltzmann model for the incompressible Navier-Stokes equation, *J. Stat. Phys.* 88 (1997) 927-944.
- [16] S. SUCCI, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*, Clarendon Press, Oxford, ISBN 978-0-19-850398-9 (2001).
- [17] D.A. WOLF-GLADOW, *Lattice-Gas Cellular Automata and Lattice Boltzmann Models, An Introduction*

- tion, Springer, (2000).
- [18] P. BHATNAGAR, E. GROSS, M. KROOK, A model for collision processes in gases: 1. Small amplitude processes in charged and neutral one-component systems, *Phys. Rev.* 94 (1954) 511.
 - [19] S. OSHER, J. SETHIAN, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.* 79 (1988) 12-49.
 - [20] I. MITCHELL, A Toolbox of Level Set Methods, August 2015, URL: <http://www.cs.ubc.ca/~mitchell/ToolboxLS/>.
 - [21] A. CAIAZZO, Asymptotic analysis of lattice Boltzmann method for fluid-structure interaction problems, Ph.D. Thesis, Kaiserslautern (Germany) and Pisa (Italy), 2007.
 - [22] T. KRÜGER, F. VARNIK, D. RAABE, Shear stress in lattice Boltzmann simulations, *Phys. Rev. E* 79 (4) (2009) 046704.
 - [23] P. LALLEMAND, L.-S. LUO, Lattice Boltzmann method for moving boundaries, *J. Comput. Phys.* 184 (2003) 406-421.