

Implementation of on-site velocity boundary conditions for D3Q19 lattice Boltzmann simulations

Martin Hecht¹ and Jens Harting^{1,2}

¹ Institute for Computational Physics, University of Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Germany

² Department of Applied Physics, TU Eindhoven, Den Dolech 2, 5600 Eindhoven, The Netherlands

E-mail: hecht@icp.uni-stuttgart.de and j.harting@tue.nl

Received 6 October 2009

Accepted 21 December 2009

Published 28 January 2010

Online at stacks.iop.org/JSTAT/2010/P01018

[doi:10.1088/1742-5468/2010/01/P01018](https://doi.org/10.1088/1742-5468/2010/01/P01018)

Abstract. On-site boundary conditions are often desired for lattice Boltzmann simulations of fluid flow in complex geometries such as those of porous media or microfluidic devices. The possibility of specifying the exact position of the boundary, independently of other simulation parameters, simplifies the analysis of the system. For practical applications it should allow one to freely specify the direction of the flux, and it should be straightforward to implement in three dimensions. Furthermore, especially for parallelized solvers, it is of great advantage if the boundary condition can be applied locally, involving only information available on the current lattice site. We meet this need by describing in detail how to transfer the approach suggested by Zou and He to a D3Q19 lattice. The boundary condition acts locally, is independent of the details of the relaxation process during collision and contains no artificial slip. In particular, the case of an on-site no-slip boundary condition is naturally included. We test the boundary condition in several setups and confirm that it can be used to accurately model the velocity field up to second order and does not contain any numerical slip.

Keywords: lattice Boltzmann methods

ArXiv ePrint: [0811.4593](https://arxiv.org/abs/0811.4593)

Contents

1. Introduction	2
2. The simulation method	4
3. Boundary conditions	6
4. The general on-site velocity boundary condition	7
5. On-site no-slip boundary condition	10
6. Numerical results	13
7. Conclusion	19
Acknowledgments	19
Appendix	20
References	22

1. Introduction

The lattice Boltzmann method (LBM) is widely used for the simulation of fluid flow [2]. It solves the Boltzmann equation on a discrete lattice and it has been proven that the Navier–Stokes equations can be recovered [3, 4]. The method has been successfully applied to the simulation of flow in porous media [5, 6], colloidal suspensions [7]–[10], liquid–gas phase transitions and multi-component flows [11]–[13], spinodal decomposition [14, 15], and in many more applications.

In spite of the wide range of applications of the LBM there is still little consensus on how to implement boundary conditions in the LBM. For some applications, especially for complex geometries in technical applications, rather simple approaches like on-site bounce back [16] rules are preferred [17], but on the other hand quite complex methods for implementing exact boundary conditions have been proposed [18, 19]. A promising approach to giving velocity boundary conditions by Zou and He [1] for 2D simulations has been generalized to 3D with the restriction to the inflow being perpendicular to the boundary plane by Kutay *et al* [5]. However, to our knowledge, a generalization to 3D with variable inflow direction has not yet been presented. Apart from this restriction, in the terms used in [5] some of the prefactors have to be revised (the correct ones can be found in [20]), but for the application studied by Kutay *et al*, the terms used in their work might be appropriate. However, especially if the influx direction is not aligned with the computational lattice, our slightly more general expressions have to be used. We follow the ideas of Zou and He [1] and derive flux boundary conditions with variable influx direction for a D3Q19 lattice [21] meaning that in three dimensions the velocity space contains 19 discrete vectors. Zou and He [1] have demonstrated a derivation for the D2Q9 model and briefly sketched the application to pressure boundaries in a D3Q15i model, where *i* indicates an incompressible model of equilibrium distribution functions [22]. Already Zou and He have pointed out that besides the basic idea of applying a bounce back rule to the

non-equilibrium part, a further modification is necessary to achieve the correct transverse momentum. A suitable choice for this correction depends on the lattice type. Zou and He give an expression for the D3Q15i lattice for the case of pressure boundaries. In the subsequent publications on D3Q19 lattices [5, 20], which is one of the most commonly used lattice types nowadays, it is assumed that the flux direction is restricted to the direction normal to the boundary plane and that this symmetry is also reflected in the distribution functions on the boundary nodes. In our generalization we drop this restriction and consistently derive the transverse momentum corrections. We investigate the accuracy of this boundary condition and highlight the special case of on-site no-slip boundary conditions included in this approach by simply setting the velocity equal to zero.

Examples for possible applications are microfluidic devices [23], i.e., microscopic channel structures which are specially designed to modify a given flow profile by the roughness or wettability of the walls or the geometry of the channels. One example of such structure is provided by micromixers [24]. In general, in simulating such devices, it is not always possible to align all walls with the Cartesian planes. In these cases one needs a boundary condition which can be used to specify the velocity in an arbitrary direction, depending on the orientation of the channel to be simulated.

One might also think of applications in porous media [5], where flows through discretized samples of stones are simulated. On the boundaries of the microscopic pores, a no-slip condition has to be applied. This is a special case of velocity boundary conditions with the velocity being zero. Since the channels cannot be aligned with the computational lattice, the question arises of how large the error introduced by the discretization is. The on-site velocity boundary conditions put forward in the present paper can be used as a replacement of the bounce back rule for the no-slip condition if the velocity is set to zero. In contrast to the case for the usual bounce back rule the position of the wall is independent of the BGK relaxation time. This fact is of great advantage when analyzing the permeability of a discretized sample.

Although the assumption of the fluid velocity being zero on the boundaries does not hold for several cases in microfluidics, the no-slip condition is highly important for many cases. Therefore, a considerable research effort has been devoted to developing no-slip boundary conditions [2, 19, 25, 26]. Some approaches turned out to contain an artificial slip length depending on various details of the simulation method, whereas other attempts involve non-local calculations like the evaluation of a velocity gradient to extrapolate the flow field beyond the boundary. In contrast to that, the boundary condition that we propose is of local type and allows us to specify the velocity on the node exactly with vanishing slip length.

Further, the boundary condition is of great benefit for hybrid simulations, i.e., simulations in which two simulation methods are coupled to simulate fluid flow [27]–[29]. The main goal of such hybrid simulations is to save computing time. A computationally cheap method is applied to simulate the flow on a more coarse-grained level, whereas in some regions, where for example interactions on the atomistic level are relevant, a different simulation method including more details is applied. The two simulation methods are coupled for example by exchange of mass, momentum and energy between the two domains via their respective boundary conditions. One can think of different setups: an LB simulation can be embedded into a finite element based Navier–Stokes solver and resolve one region in more detail. Another example case is that in an LB simulation

one region is resolved even on the molecular level by means of a molecular dynamics simulation. Practically, in current hybrid simulations the coupling is implemented within an overlapping region [30] of the two simulations, but it would be a great advance if one could manage simply to couple two boundary conditions without any overlap being needed. The possibility of generally determining the velocity on a boundary node in an LB simulation is one step towards this goal.

The remainder of this paper is structured as follows. In section 2 we describe the simulation method in general and introduce our notation for the lattice vectors. Then, we briefly review different boundary conditions from the literature in section 3. After that, we derive and discuss the velocity boundary condition for the D3Q19 lattice in section 4. We separately discuss the special case of the no-slip condition in section 5. Numerical results are presented and discussed in section 6 and finally, we draw a conclusion in section 7 of the current paper.

2. The simulation method

The lattice Boltzmann method is a numerical method for solving the Boltzmann equation (1) on a discrete lattice. The Boltzmann equation describes the dynamics of a gas from a microscopic point of view: in a gas, particles, with velocities \mathbf{v}_i , collide with a certain probability and exchange momentum among themselves. For ideal collisions the total momentum and energy are conserved in the collisions. The Boltzmann equation expresses how the probability $f(\mathbf{x}, \mathbf{v}, t)$ of finding a particle with velocity \mathbf{v} at a position \mathbf{x} and at time t evolves with time:

$$\mathbf{v} \cdot \nabla_{\mathbf{x}} f + \mathbf{F} \cdot \nabla_{\mathbf{p}} f + \frac{\partial f}{\partial t} = \hat{\Omega}(f), \quad (1)$$

where \mathbf{F} denotes an external body force, $\nabla_{\mathbf{x}, \mathbf{p}}$ the gradient in position and momentum space, and $\hat{\Omega}(f)$ denotes the collision operator. Bhatnagar, Gross, and Krook [31] proposed the so-called BGK dynamics, where the collision operator $\hat{\Omega}$ is chosen as a relaxation with a characteristic time τ to the equilibrium distribution $f^{(\text{eq})}(\mathbf{v}, \rho)$:

$$\hat{\Omega}(f) = -\frac{1}{\tau}(f - f^{(\text{eq})}). \quad (2)$$

The equilibrium distribution function for athermal models depends on the local density $\rho(\mathbf{x}, t)$ and the velocity field $\mathbf{v}(\mathbf{x}, t)$. The lattice Boltzmann method [32] discretizes the probability density f in space and time. The discrete Boltzmann equation, which is solved by the LBM, can be rigorously derived from the Boltzmann equation [33]. The discretization, and especially the analytic expression for the equilibrium distribution $f^{(\text{eq})}$, depends on the lattice type. We use a D3Q19 lattice which is a very popular lattice type for 3D LB simulations. On each lattice site 19 values $f_i(\mathbf{x}, t)$ are stored, each of them assigned to a lattice vector \mathbf{c}_i . We use the notation where the vectors \mathbf{c}_i are the i th column vectors of the matrix

$$\mathbf{M} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 0 \end{bmatrix}. \quad (3)$$

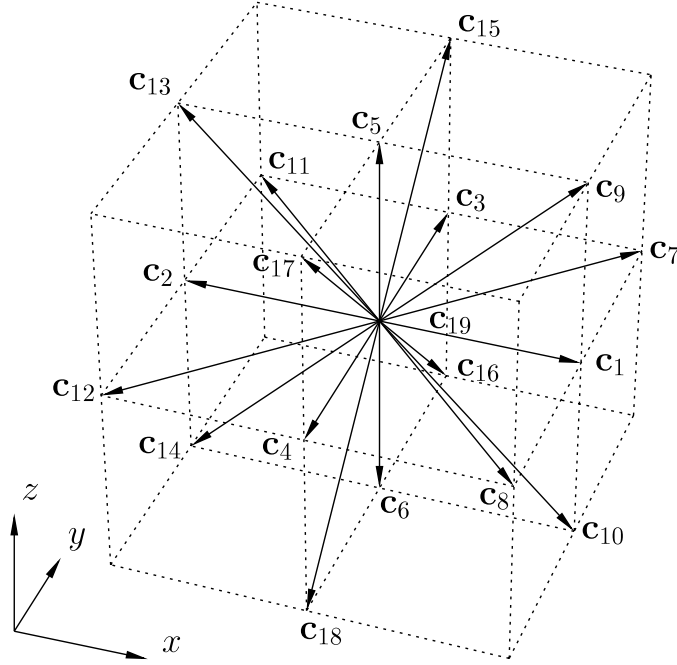


Figure 1. The geometry of the D3Q19 lattice with lattice vectors \mathbf{c}_i as defined in equation (3).

The geometry is shown in figure 1.³ The local density at a lattice point can be obtained by summing all f_i ,

$$\rho(\mathbf{x}, t) = \sum_{i=1}^{19} f_i(\mathbf{x}, t), \quad (4)$$

and the streaming velocity is given by

$$\mathbf{v}(\mathbf{x}, t) = \frac{1}{\rho(\mathbf{x}, t)} \sum_{i=1}^{19} f_i(\mathbf{x}, t) \mathbf{c}_i. \quad (5)$$

We express all quantities in lattice units, i.e., time is measured in units of update intervals and length is measured in units of the lattice constant. For practical applications a suitable mapping to physical units based on a dimensional analysis has to be applied.

In the lattice Boltzmann method two steps are performed in an alternating way.

- (1) The ‘streaming step’: propagate each of the distribution functions f_i to the next lattice site in the direction of its assigned lattice vector \mathbf{c}_i .
- (2) The ‘collision step’: on each lattice site relax the probability functions f_i towards the equilibrium value $f_i^{(\text{eq})}(\mathbf{v}, \rho)$. In BGK dynamics this is according to equation (2).

³ Note that some authors, e.g. in [5], use a different notation in which the vectors \mathbf{c}_8 – \mathbf{c}_{12} are permuted, and \mathbf{c}_{13} and \mathbf{c}_{14} as well as \mathbf{c}_{17} and \mathbf{c}_{18} are exchanged. Sometimes \mathbf{c}_{19} is denoted as \mathbf{c}_0 .

The equilibrium value $f_i^{(\text{eq})}$ is obtained by discretizing the Boltzmann distribution. Several expressions of different order have been proposed, where we use the popular form involving terms in the velocity up to the second order [1], [34]–[37]:

$$f_i^{(\text{eq})}(\rho, \mathbf{v}) = w_i \rho \left[1 + \frac{\mathbf{c}_i \cdot \mathbf{v}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{v})^2}{2c_s^4} - \frac{v^2}{2c_s^2} \right] \quad (6)$$

with the lattice speed of sound $c_s = \frac{1}{\sqrt{3}}$ for the D3Q19 lattice and the lattice weights

$$w_i = \begin{cases} \frac{2}{36}, & i = 1 \dots 6 \\ \frac{1}{36}, & i = 7 \dots 18 \\ \frac{12}{36}, & i = 19. \end{cases} \quad (7)$$

The pressure $p = c_s^2 \rho$ turns out to be proportional to the density and the dynamic shear viscosity is given by [2, 38]

$$\eta = c_s^2 \rho \left(\tau - \frac{1}{2} \right). \quad (8)$$

3. Boundary conditions

On the boundary nodes, the distribution function assigned to vectors \mathbf{c}_i pointing out of the lattice move out of the computational domain in the propagation step, and the ones assigned to the opposing vectors are undetermined because there are no nodes which the distributions could come from. Therefore, on the boundary nodes, special rules have to be applied.

These boundary conditions can be chosen in various manners. Periodic boundaries are realized by propagating the f_i leaving the computational domain on the one boundary to the boundary nodes located on the opposite side of the domain. Closed boundaries are commonly implemented by a so-called mid-grid bounce back rule [2], which means that the distributions f_i pointing out of the domain are copied to f_j , for which $\mathbf{c}_j = -\mathbf{c}_i$, i.e., locally, on each lattice site, the undetermined values are filled with the ones which would stream out of the domain without collision on the boundary node. They enter one time step later into the simulation domain again [39].

However, for many questions in fluid dynamics it is required to determine the pressure or the velocity field at the boundary. The first is known as the Dirichlet boundary condition, and the latter as the Neumann boundary condition. In the Neumann case the flux on the boundary of the domain is fixed, whereas in the Dirichlet case the pressure is given as a boundary condition.

Zou and He [1] have proposed how to implement Dirichlet and Neumann boundary conditions on a D2Q9 lattice and briefly sketched how to apply it for a D3Q15i simulation. Kutay *et al* [5] have transferred this proposal to a D3Q19 lattice. However, their approach is derived under the assumption that the inflow and outflow velocities are always perpendicular to the boundary plane, and oriented along one of the main lattice directions ($\mathbf{c}_i, i = 1 \dots 6$). We generalize this to inflow with arbitrary direction in section 4.

Often more elaborated boundary conditions are applied. Chen *et al* [37] and Ginzbourg and d’Humières [19] suggested extrapolation of the f_i on the first and second layer of the lattice to the nodes outside the domain. These extrapolated values can be thought of as the lattice populations propagating into the domain and arriving on the

boundary nodes in the next streaming step. Inamuro *et al* have introduced a counter-slip to compensate for a numerical slip which occurs when applying on-site bounce back [25]. Skordos came up with an approach where additional differential equations are solved on the boundary nodes to calculate the unknown populations [18]. Ansumali and Karlin have developed a LB no-slip boundary condition from kinetic theory [40], and, more recently, d’Orazio *et al* [41] and Tang *et al* [35] came up with thermal boundary conditions which also involve an extrapolation scheme and bounce back with counter-slip respectively. Ladd and Verberg have developed a boundary condition with a resolution of the position of the wall on a sub-grid level, which is especially required if suspended particles are modeled [9, 42, 43]. Schiller and Dünweg [44] use a reduced set of distribution functions on the boundary nodes. For their reduced D3Q19 model they derive equilibrium distributions and propose a multi-relaxation time dynamics and a special collision operator on the boundary.

Latt *et al* have compared and discussed several of these approaches in [45]. They also include the boundary condition of Zou and He [1] in their discussion. As indicated by Latt *et al*, a generalization of the boundary conditions proposed by Zou and He is still not provided. However, a general local boundary rule which can be applied in a simple way on each node separately would be desirable.

We derive such a boundary condition in section 4. Our generalization of the velocity boundary condition proposed in [1] only involves the distribution functions defined on the local boundary node and allows us by very simple and computationally cheap steps to set the velocity on the node to a distinct vector. The desired value is obtained exactly and we cannot detect any artifacts like a numerical slip length or bends in the velocity profile.

4. The general on-site velocity boundary condition

As mentioned in section 3, we extend the boundary condition of Zou and He [1] to a D3Q19 lattice. We derive the boundary condition for the bottom plane ($z = 0$) in detail and give the results for the other planes in the appendix. They can be derived following the same steps.

The boundary conditions are derived by using the set of equations consisting of equation (4) and the components of equation (5):

$$\rho v_x = f_1 + f_7 + f_8 + f_9 + f_{10} - (f_2 + f_{11} + f_{12} + f_{13} + f_{14}), \quad (9)$$

$$\rho v_y = f_3 + f_7 + f_{11} + f_{15} + f_{16} - (f_4 + f_8 + f_{12} + f_{17} + f_{18}), \quad (10)$$

$$\rho v_z = f_5 + f_9 + f_{13} + f_{15} + f_{17} - (f_6 + f_{10} + f_{14} + f_{16} + f_{18}). \quad (11)$$

Due to the continuity relation $(\partial\rho/\partial t) + \nabla \cdot (\rho\mathbf{v}) = 0$, we are free to specify just three of the four variables (ρ and the three components of \mathbf{v}) on the boundary. If we fix the tangential velocity v_x, v_y on the bottom layer of the lattice, and the density to a given value ρ_0 , the z -component of the inflow velocity v_z can be calculated from equations (11) and (4),

$$v_z = 1 - \frac{1}{\rho_0} [f_1 + f_2 + f_3 + f_4 + f_7 + f_8 + f_{11} + f_{12} + f_{19} + 2(f_6 + f_{10} + f_{14} + f_{16} + f_{18})], \quad (12)$$

where the f_i pointing *out of* the system appear with a prefactor of 2, and all in-plane components appear with weight 1. The components pointing into the system, f_5, f_9, f_{13}, f_{15} , and f_{17} , which are undetermined after the streaming step, do not appear at all. With equation (12) Neumann (or pressure) boundary conditions can be applied by specifying ρ_0 on the boundary and using equation (12) to calculate v_z . If equation (12) is written in the form

$$\rho = \frac{1}{1 - v_z} [f_1 + f_2 + f_3 + f_4 + f_7 + f_8 + f_{11} + f_{12} + f_{19} + 2(f_6 + f_{10} + f_{14} + f_{16} + f_{18})], \quad (13)$$

all three components of the velocity can be specified and equation (13) is used to calculate the density ρ . This is the Dirichlet case, or flux boundary condition. Again, the undetermined populations f_5, f_9, f_{13}, f_{15} , and f_{17} do not enter the calculation.

We have used two out of four equations (equations (9)–(11) and (4)), but we still have to compute the five f_i pointing into the computing domain. Following Zou and He [1] we assume that on the boundary the bounce back condition is still valid for the non-equilibrium part f_i^* of the single-particle distribution f_i :

$$f_i^* = f_i - f_i^{(\text{eq})}. \quad (14)$$

The bounce back condition in the $+z$ -direction (in the direction normal to the boundary) would read as

$$f_5^* = f_5 - f_5^{(\text{eq})} = f_6 - f_6^{(\text{eq})} = f_6^*, \quad (15)$$

which leads, by taking $f_5^{(\text{eq})}$ and $f_6^{(\text{eq})}$ from equation (14), to

$$f_5 = f_6 - w_6 \rho \left[1 - \frac{v_z}{c_s^2} + \frac{v_z^2}{2c_s^4} \right] + w_5 \rho \left[1 + \frac{v_z}{c_s^2} + \frac{v_z^2}{2c_s^4} \right] = f_6 + \frac{2w_5}{c_s^2} \rho v_z = f_6 + \frac{1}{3} \rho v_z. \quad (16)$$

Here we make use of the fact that the distribution functions in equation (6) are approximated by taking only terms up to second order in \mathbf{v} into account. However, this approximation could be applied directly to equation (16) as well. For the derivation of the boundary condition it is needed; otherwise the higher order terms would introduce anisotropic effects in the boundary rule.

Generally, in the collision step (in equation (2)) higher order terms may be taken into account for the bulk, but for the boundary conditions a qualitatively different approach, like a higher order extrapolation scheme, has to be considered when aiming for higher order accuracy.

For the D3Q19 lattice, however, we need two more equations. To keep the symmetry of the problem, we assume bounce back of the non-equilibrium part for all populations f_i . This results in four equations,

$$f_9 = f_{14} + \frac{2w_9}{c_s^2} \rho (v_z + v_x), \quad (17)$$

$$f_{13} = f_{10} + \frac{2w_{13}}{c_s^2} \rho (v_z - v_x), \quad (18)$$

$$f_{15} = f_{18} + \frac{2w_{15}}{c_s^2} \rho (v_z + v_y), \quad (19)$$

$$f_{17} = f_{16} + \frac{2w_{17}}{c_s^2} \rho(v_z - v_y), \quad (20)$$

so the system of equations is overdetermined.

Therefore, following the ansatz of Zou and He [1] for the pressure boundary condition on a D3Q15i lattice, we introduce two new variables N_x^z and N_y^z , the transverse momentum corrections on the z -boundary for distributions propagating in x - and y -directions, respectively. These terms turn out to vanish in equilibrium, but they are non-zero if velocity gradients are present, e.g., when shear flow is imposed by the particular choice of the boundary conditions. It turns out that these expressions appear again in the stress tensor. They reflect the fact that by imposing a transverse velocity component on the boundary, stress is also imposed on the system. The transverse momentum corrections involve the populations propagating in the boundary plane in the update rule of the boundary condition. We add the terms to the right-hand side and assume that the same expression with opposite sign is needed for two of the vectors in the same plane. Our ansatz thus reads as follows:

$$f_9 = f_{14} + \frac{\rho}{6}(v_z + v_x) - N_x^z, \quad (21)$$

$$f_{13} = f_{10} + \frac{\rho}{6}(v_z - v_x) + N_x^z, \quad (22)$$

$$f_{15} = f_{18} + \frac{\rho}{6}(v_z + v_y) - N_y^z, \quad (23)$$

$$f_{17} = f_{16} + \frac{\rho}{6}(v_z - v_y) + N_y^z. \quad (24)$$

The system of equations (21)–(24), together with equations (9) and (10), is now a closed system. By equations (9) and (10) we specify the tangential components of the velocity v_x and v_y , which do *not* need to be equal to zero in our approach. Inserting equations (21)–(24) into (9) and (10) gives exact solutions for N_x^z , and N_y^z , respectively:

$$N_x^z = \frac{1}{2}[f_1 + f_7 + f_8 - (f_2 + f_{11} + f_{12})] - \frac{1}{3}\rho v_x, \quad (25)$$

$$N_y^z = \frac{1}{2}[f_3 + f_7 + f_{11} - (f_4 + f_8 + f_{12})] - \frac{1}{3}\rho v_y. \quad (26)$$

These transverse momentum corrections can be inserted into equations (21)–(24) again, and combining these with equation (16) we find explicit expressions for all unknown populations.

Note that in equations (25) and (26) it is required to sum over all in-plane contributions to the velocity in the x - and y -directions and the weights are consistent with the lattice weights of the f_i appearing in the above expressions. As expected, N_x^z and N_y^z vanish (up to second order which is our precision within this derivation) if we set all f_i to their equilibrium value. The results for the other planes are given in the appendix.

A general form for all boundary planes can be written down by introducing the normal vector on the boundary \mathbf{n} , the tangential vectors $\mathbf{t}_i = \mathbf{c}_i - (\mathbf{c}_i \cdot \mathbf{n})\mathbf{n}$, and the notation f_{-i} denoting the direction to which a population is bounced back $\mathbf{c}_{-i} = -\mathbf{c}_i$. From the populations f_i assigned to a direction \mathbf{c}_i pointing into the wall, the new populations f_{-i}

in the opposite direction can be calculated as

$$f_{-i} = f_i - \frac{\rho}{6} \mathbf{c}_i \cdot \mathbf{v} - \frac{\rho}{3} \mathbf{t}_i \cdot \mathbf{v} + \frac{1}{2} \sum_{j=1}^{19} f_j (\mathbf{t}_i \cdot \mathbf{c}_j) (1 - |\mathbf{c}_j \cdot \mathbf{n}|). \quad (27)$$

Due to the particular choice of equations (21)–(26) or equation (27), it is possible to specify the velocity at an exact value on the lattice site. The rules presented here are independent of the relaxation rate in the collision step, since all calculations involve only the known values of the f_i and equilibrium functions. Relaxation is calculated separately after all unknown f_i are calculated and the macroscopic velocity and density are preserved during collision. There are no restrictions on the orientation of the inflow direction. Furthermore, all calculations are local on each lattice site. Apart from using only terms of first and second order in \mathbf{v} for the equilibrium distributions $f_i^{(\text{eq})}$ in equation (6) no approximations are made. Therefore, we have derived a way to implement explicit local on-site boundary conditions which model the fluid field up to second order in the velocity.

A scheme similar to ours has been proposed by Halliday *et al* [46] for a D2Q9 lattice. These authors construct the unknown distributions locally on each lattice site starting from a Chapman–Enskog analysis. During their derivation they have to choose a set of variables that they consider as free variables. This is similar to the approach of introducing the transverse momentum corrections in order to be able to solve the system of equations. Halliday *et al* find results for the unknown populations which involve the components of the strain rate tensor calculated from the known populations. From this point of view it might be possible to apply a scheme similar to the one proposed by Halliday *et al* to a D3Q19 lattice. However, in three dimensions the systems of equations, in the generality of [46], might become difficult to handle.

Special care has to be taken when connecting the influx and outflux boundary conditions at the corners and edges of the simulation domain with other types of boundary conditions that are applied on other boundary planes. If no-slip boundary conditions are assumed on the x - and y -boundaries, one has to take care that the influx velocity tends to zero at the edges. We discuss the special case of no-slip boundaries as a subset of velocity boundaries in section 5.

5. On-site no-slip boundary condition

The on-site velocity boundary condition proposed in this paper includes an important special case: setting the velocity $\mathbf{v} = 0$ results in a no-slip boundary for non-moving boundaries. Therefore, this boundary condition can also be used as a replacement of the mid-grid bounce back rule. However, even more generally, moving boundaries, e.g., moving shear plates, can be implemented by imposing the wall velocity \mathbf{v} on the boundary nodes. The position of the wall is *exactly* on the lattice nodes. This is in contrast to the case for most no-slip boundaries proposed in the literature, where the wall position is assumed at half the distance between two nodes. However, in many of those approaches the exact position of the wall depends on the BGK relaxation time. This is not the case for our approach.

One of the pillars of the LBM is local mass conservation [45], which should be fulfilled not only in the bulk, but also on closed boundaries. However, some extrapolation schemes

may be less accurate on this point [44], whereas for our on-site approach mass conservation is strictly fulfilled at the closed walls.

The transverse momentum corrections similar to those given in equations (25) and (26) are given in the appendix for each coordinate plane, and both velocity components in each of those planes. They are corrections to the on-site bounce back rule. With these corrections taken into account, the velocity is exactly zero on the node. For edge nodes with both boundaries being implemented as described here, we suggest first applying the bounce back rule for all f_i pointing out of the computational domain and then calculating the transverse momentum correction. On an edge node only one tangential vector along the edge can be used for ensuring the no-slip condition.

Consider for example the edge between the xy -plane and the yz -plane, where the y -axis forms the edge. Contributions in the boundary planes known after bounce back are f_1, f_7 , and f_8 in the xy -plane and f_5, f_{15} , and f_{17} in the yz -plane. However, to ensure the no-slip condition one can define

$$N_y^{xz} = \frac{1}{4}[f_3 - f_4]. \quad (28)$$

The correction to the distributions f_i with $i = 7, 8, 15$, and 17 then is $N_y^{xz} \mathbf{c}_i \cdot \mathbf{t}_i$ which has to be added to the distributions. The prefactor in equation (28) takes into account that the remaining slip velocity after bounce back is distributed among four populations obtained from the bounce back rule. Similar expressions can be written down for each edge. A general expression for the modified bounce back rule is

$$f_{-i} = f_i - \frac{1}{4} \sum_{j=1}^{19} f_j (\mathbf{t}_i \cdot \mathbf{c}_j) (1 - |\mathbf{c}_j \cdot \mathbf{n}^{(1)}|) (1 - |\mathbf{c}_j \cdot \mathbf{n}^{(2)}|), \quad (29)$$

where $\mathbf{n}^{(1)}$ and $\mathbf{n}^{(2)}$ denote the two normal vectors on the two boundary planes meeting at the edge under consideration.

On the edges and corners, apart from the incoming populations, there are so-called ‘buried links’ [26], i.e., lattice vectors \mathbf{c}_i for which the opposing lattice vector \mathbf{c}_{-i} points out of the domain, as well. The two lattice vectors $\mathbf{c}^{(1,2)} = \pm(\mathbf{n}^{(1)} - \mathbf{n}^{(2)})$ make up the buried link on an edge node. In the following, lattice vectors with a subscript, \mathbf{c}_i , denote distinct vectors as defined in equation (3), whereas vectors with a superscript, $\mathbf{c}^{(i)}$, denote vectors which belong to the buried links, and which depend on the normal vectors on the individual boundary planes. The distribution functions assigned to the buried links have to be assigned separately. We choose them such that they contribute to the same density according to their lattice weight:

$$f^{(1,2)} = \frac{1}{22} \sum_{i=1}^{18} f_i (1 - |\mathbf{c}_i \cdot [\mathbf{n}^{(1)} \times \mathbf{n}^{(2)}]|) \cdot \left(1 - \left| \mathbf{c}_i \cdot \left[\frac{\mathbf{n}^{(1)} - \mathbf{n}^{(2)}}{\sqrt{2}} \right] \right| \right). \quad (30)$$

Similarly, the distribution on the resting node is chosen as $f_{19} = (w_{19}/w_7) f^{(1,2)} = 12 f^{(1,2)}$. The weights are always determined by the number of f_i which contribute to the sum and their respective lattice weights w_i according to equation (7). In equation (30) six f_i with lattice weight $\frac{1}{18}$ and ten f_i with weight $\frac{1}{36}$ contribute, which makes up an overall contribution of $\frac{22}{36}$. To reduce this to the desired lattice weight, we have to divide by 22,

and to obtain a value for the resting node, we multiply by 12, because of the lattice weight of the resting node being twelve times larger.

At the edges surrounding the inlet and outlet planes, on the other hand, one needs either pressure or velocity boundary conditions, depending on the boundary type used for the inlet or outlet. For velocity boundaries one has to take care that the velocity profile decays to zero, so the no-slip boundary condition just described can be used on all edges. For pressure boundaries one prescribes a density $\rho = \rho_0$ which we can be used to calculate the distribution assigned to the buried link:

$$\tilde{f}^{(1,2)} = \frac{\rho_0 - 22f^{(1,2)}}{14}, \quad (31)$$

where f_{19} is then calculated as $f_{19} = 12\tilde{f}^{(1,2)}$.

According to Maier *et al* [26] no-slip boundaries cannot be enforced on convex boundary nodes. However, slip along the edge can be reduced by correcting all distributions f_i traveling into the interior of the system by using

$$N_i = \frac{1}{4} \mathbf{c}_i \cdot (\mathbf{n}^{(1)} \times \mathbf{n}^{(2)}) \sum_{j=1}^{19} f_j \mathbf{c}_j \cdot (\mathbf{n}^{(1)} \times \mathbf{n}^{(2)}), \quad (32)$$

which follows the same idea as equations (29) and (21)–(24): momentum in a direction parallel to the surface, which would remain on a node after applying the boundary rule, is removed by modifying those populations that will afterward propagate back into the bulk of the system. In principle, one could split equation (29) into two steps: first, apply bounce back for all populations leaving the system, and then correct the populations traveling away from the edge by the term given in equation (32). For convex edges these are the populations traveling into the bulk, and for concave edges they propagate in the boundary planes. This opens the possibility of implementing all rules in a single procedure, for which the normal vector \mathbf{n} is stored on each lattice site via an integer number. The vector is obtained from the matrix \mathbf{M} defined in equation (3). For values between 1 and 6 equation (27) is applied, for values between 7 and 18 either equation (32) applies or, depending on the values stored on the neighboring nodes, the bounce back rules corrected according to equation (29) may be applied instead. This information, which expresses whether the edge is concave or convex, can be obtained once, when the lattice is generated, and may be stored in the sign of the lattice vector index. The normal vector points into the bulk and indicates the direction of the symmetry plane on the edge nodes.

Finally, on the corner nodes one can define three normal vectors on the boundary planes meeting there, $\mathbf{n}^{(1)}$, $\mathbf{n}^{(2)}$, and $\mathbf{n}^{(3)}$. Like for the buried links, there is a complete plane in which six vectors are located, that only couple to the simulation in the collision step. The buried vectors $\mathbf{c}^{(1\cdots 6)}$ are the ones for which $\mathbf{c}^{(i)} \cdot (\mathbf{n}^{(1)} + \mathbf{n}^{(2)} + \mathbf{n}^{(3)}) = 0$. Since the normal vector on this plane is not contained in the set of vectors for the D3Q19 lattice, additional indices are needed to mark the corner nodes. After bouncing back the known

f_i , the distributions assigned to buried vectors are set to

$$f^{(1\cdots6)} = \frac{1}{18} \sum_{i=1}^{18} f_i \left[1 - \left| \mathbf{c}_i \cdot \left(\frac{\mathbf{n}^{(1)} + \mathbf{n}^{(2)} + \mathbf{n}^{(3)}}{\sqrt{3}} \right) \right| \right] \quad (33)$$

if velocity boundaries are applied, or to $\tilde{f}^{(1\cdots6)} = (\rho_0/18) - f^{(1\cdots6)}$ if pressure boundaries are chosen. f_{19} is set to $12 f^{(1\cdots6)}$ or $12 \tilde{f}^{(1\cdots6)}$, respectively. A correction similar to equation (29) is not necessary on the corner nodes⁴.

In complex geometries there are points in which edges (convex or concave ones) meet planes which are oriented perpendicular to the direction of the edge. There, we propose to use bounce back for those populations which would leave the computational domain and to assign an appropriate value to the resting node, as described for the corner nodes. There are no buried links, because those links are located inside the boundary plane which the edge connects to, so the resting node must be set to $f_{19} = \frac{12}{36} \sum_{i=1}^{18} f_i$. In total there are six planes and four possible orientations of the edges, each of them either convex or concave, making up 48 more cases. However, since there are no buried links and a momentum correction is not necessary either, the rules can be implemented easily in only a few lines of code. In section 6 we show the results of tests of the boundary condition in simple geometries like Poiseuille flow between two plates, where the exact solution is known. As an example for more complex geometries we simulate the flow through a rectangular channel, where also edges and corners are involved.

6. Numerical results

We test our boundary condition by simulating a Poiseuille flow through a tilted channel. The size of the computational domain is $64 \times 8 \times 128$ LB nodes, where the channel has a width of 20 nodes and is tilted by an angle $\alpha = \arctan(\frac{40}{127}) \approx 17.48^\circ$. This angle is chosen such that both ends of the channel intersect the xy -plane at the top and the bottom of the computational domain and that there are two lattice sites on the wall at the left and at the right of the channel at the bottom and the top plane respectively. The flow through our test channel is simulated in three dimensions. However, for convenience, the y -direction is periodic. The walls (only in this test) are implemented as simple bounce back nodes. Here we apply the boundary conditions derived in section 4 as influx and outflux conditions and compare them to other implementations.

We choose this simple test because the analytical solution for the flow field is known and so we can estimate the numerical error. Usually, one would avoid having walls not aligned with the computational lattice because of the staircase like discretization of the walls, which brings in an additional discretization error into the simulation. This discretization error can be avoided by simply aligning the channel with the computational lattice. However, if more complex structures, e.g., Y-channels for applications in microfluidics, are simulated, it may happen that always at least one channel is not aligned with one of the Cartesian directions. A technical workaround, if appropriate boundary conditions are missing, is to simulate a very long channel so that in the first section of the channel, the fluid can relax to a steady flow profile, and only afterward enters

⁴ Values of 19 onwards as vector indices can be used to distinguish the different corner nodes.

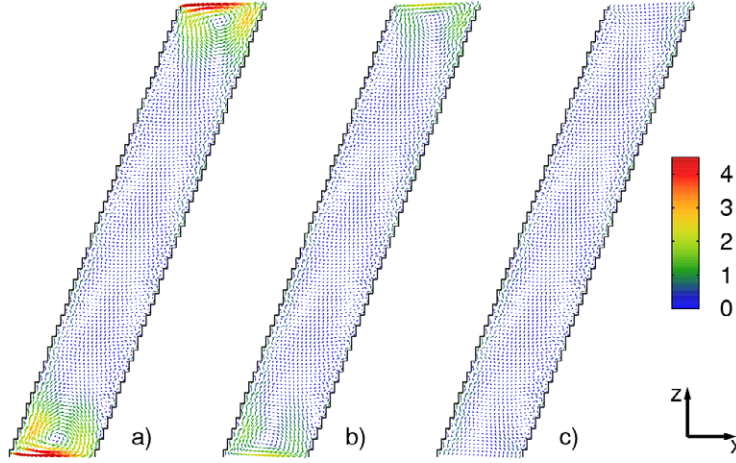


Figure 2. Velocity difference fields for different approaches: inflow and outflow velocity constraints in the direction perpendicular to the boundary plane (a), inflow and outflow velocity tilted and of parabolic shape as in the analytic solution, but with N_x^z and N_y^z set equal to zero (b), and the boundary conditions derived in section 4 with the correct choice for N_x^z and N_y^z (c). The velocity difference vectors are scaled for the drawing by a factor of 1500 and the absolute value of the velocity difference is color coded from blue (small) to red (large).

the actual simulation domain. However, this causes the computational effort to increase substantially.

Knowing the width of the channel, the center of the inlet and outlet, and a given velocity \mathbf{v}_0 on the centerline, one can calculate a Poiseuille flow field inside the channel,

$$\mathbf{v}^P(\mathbf{x}) = \mathbf{v}_0 \left[1 - \left(\frac{x - x_0 - \gamma(z - z_0)}{\Delta x} \right)^2 \right], \quad (34)$$

where x_0 and z_0 denote the center of the simulation space, Δx is the halfwidth of the channel measured along the x -direction, and γ denotes the increment due to the tilting angle, which is related to the components of the velocity by $\gamma = \tan \alpha = (v_z/v_x) = \frac{40}{127}$.

We simulate flow through such a channel and apply different inlet and outlet boundary conditions. We use a relaxation time $\tau = 1$ in all simulations presented here. However, we have checked that the results do not depend on this particular choice. After 5000 time steps a steady flow field is reached. However, to be sure that the simulations have converged, we simulate 20 000 time steps before we evaluate data.

To visualize the difference between simulation and theoretical prediction we subtract the velocity on each lattice node and draw the resulting vector field as in figure 2. The value of the velocity is scaled by a factor of 1500 for drawing the arrows. The colors are assigned the absolute value of the velocity *after* scaling the difference field.

In figure 2(a) we apply the boundary condition used by Kutay *et al* [5] to a case where the restriction of the inflow and outflow velocity parallel to the z -direction introduces an error in the region close to the boundary. Note that in [5], apart from assuming the velocity perpendicular to the boundary, the authors underestimate the transverse components, which may be of no importance in this case. We use the correct coefficients as presented

very recently in [20], but keep the restriction to the inflow perpendicular to the boundary, which has a much larger influence on the flow field. Not only are the first and second layer of nodes close to the boundary affected, but also the boundary condition introduces vortices which have approximately the size of the diameter of the channel. Therefore, the first step in generalizing the boundary condition from [5] to a case where the inflow and outflow velocities have an arbitrary orientation is using equations (17)–(20), as used in the simulation for which the result is shown in figure 2(b). It is obvious that this boundary condition still introduces vortices close to the inflow and outflow. The strength of the vortices is smaller compared to the case shown in figure 2(a). However, the size of the vortices is comparable to the width of the channel here as well. The value of the tangential velocity on the boundary nodes differs from the value that one inserts into the equations. On introducing the transverse momentum corrections N_x^z and N_y^z in equations (21)–(24), the vortices disappear and the velocity takes exactly the value that one specifies with equations (9)–(11) as one can see in figure 2(c). The remaining difference field can be mostly ascribed to the discretization on the lattice. Each single step of the wall discretized to individual steps can be found in the flow profile. However, at the influx and outflux boundaries no additional artifacts can be seen, which demonstrates the strength of our boundary condition. The velocity on the boundary nodes takes exactly the value which we specify, and therefore, no vortices are generated.

The transverse momentum corrections N_x^z and N_y^z could also be understood in terms of a counter-slip similar to that in the approach of Inamuro *et al* [25], but the ansatz used to obtain the unknown populations f_i is different: we assume a bounce back rule for the non-equilibrium part of the distributions and end up with a linear correction to the reflected populations, whereas the authors of [25] construct the unknown distributions based on kinetic theory where the correction appears not on the level of the distribution functions but as a counter-slip on the level of the wall velocity. The values for the density and the velocity inserted into the equilibrium distributions in Inamuro’s method are different from the ones used for bulk nodes. In our approach, however, the boundary nodes are treated similarly to the bulk nodes: the velocity on the boundary node can be calculated by inserting equations (25) and (26) into (21)–(24) and the distributions f_i obtained together with the one from equation (16) and the density from equations (13) into (5). It turns out that the velocity calculated from equation (5) is exactly the one which is imposed at the boundary node by equations (21)–(24).

In all simulations we kept the tilting angle of the channel constant, because the error of our boundary condition is angle independent. We can quantify the quality of the boundary conditions by computing the ratio of the absolute value of the difference field and the calculated velocity field on each node. The values obtained are averaged over the first twenty layers of LB nodes from the boundary:

$$\xi = \int_V \frac{|\mathbf{v}(\mathbf{x}) - \mathbf{v}^P(\mathbf{x})|}{|\mathbf{v}^P(\mathbf{x})|} dV, \quad (35)$$

where the volume V contains those layers of lattice nodes, which are at most a distance of the channel width apart from the boundary of the computational domain. This captures approximately the vortices and provides a measure for the quality of the boundary condition. The results for the different cases shown in figure 2 are listed in

the corresponding table.

Boundary condition	Relative error ξ
On-site velocity (figure 2(c))	0.0996
N_x^z and N_y^z set to zero (figure 2(b))	0.126
$v_x = v_y = 0$ (figure 2(a))	0.175

Good agreement with the expected Poiseuille flow profile (equation (34)) is reflected in small relative errors. Large numbers indicate deviations in the area, where the fluid fields are compared. We ascribe the remaining deviations to the discretization error of the wall and the accompanying uncertainty in the exact wall position in the present case of the staircase like discretization. We check this by increasing the resolution of the simulation by a factor of two. As we expect, the numerical error due to the staircase like discretization decreases roughly by a factor of two to $\xi = 0.051$. This shows that the staircase like discretization introduces a first-order error. Therefore, we need further investigations to see the second-order accuracy of the influx and outflux boundary conditions.

As another test for the flux boundary condition we simulate a straight channel aligned parallel to the computational lattice, again in a $64 \times 8 \times 128$ domain with the same boundary conditions as for the inclined channel. The remaining relative error decreases to 0.002 35, which is typical for Poiseuille flow simulations at this resolution in combination with a mid-grid bounce back rule on the boundary.

We can further measure the quality of our boundary condition in a shear simulation. On a 32^3 lattice we apply periodic boundaries in x - and y -directions and impose a shear velocity of $v_x = \pm 0.02$ with opposite sign on the top and bottom planes. We obtain a linear flow profile within floating point precision. There are no notable jumps between the first and second layers of LB nodes, which confirms that the strain rate tensor $\mathbf{\Pi}$ is set up correctly on the boundary nodes.

In a next step we simulate Poiseuille flow again, but this time we use a 32^3 lattice with periodic boundaries in the y - and z -directions. We apply a body force [14, 47] by adding a force term

$$\Delta \mathbf{v} = \frac{\tau \mathbf{F}}{\rho} \quad (36)$$

to the velocity in equation (6) in the whole simulation volume. The Poiseuille profile we expect is of the form

$$\mathbf{v} = \frac{\mathbf{F}}{2\eta} \left(1 - \left(\frac{x - x_0}{\Delta x} \right)^2 \right), \quad (37)$$

where the viscosity is given by equation (8). The velocity profile found in the simulation together with the expected Poiseuille profile is plotted in figure 3. The parabola contains no fit parameters. The velocity is exactly zero on the boundary nodes, whereas with a simple bounce back a numerical slip can be observed, which results in a velocity of 4×10^{-5} for the same simulation setup without using the transverse momentum corrections N_x^z and N_y^z . We have carried out this test with $\tau = 1$, but the data set presented in figure 3 is obtained with $\tau = 2$ to ensure that our boundary conditions are not restricted to the special case of $\tau = 1$. Apart from the influence of τ on the viscosity (equation (8)), our

Implementation of on-site velocity boundary conditions for D3Q19 lattice Boltzmann simulations

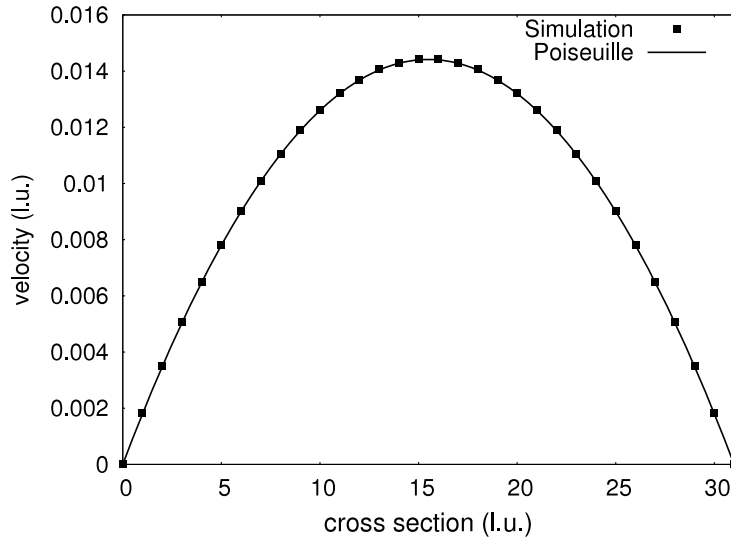


Figure 3. Poiseuille flow between two parallel walls driven by a body force. The simulation data are averaged in each lattice plane parallel to the walls and agree up to floating point precision with the calculated Poiseuille profile.

simulation results are not affected by the relaxation time. In particular, we do not see any τ dependent (numerical) slip.

Also in this second-order test we find that the numerical error is of the size of the floating point precision on the computer. This underlines that our boundary condition reproduces the velocity field up to second order.

Finally, we test our implementation by simulating flow through a square channel. The analytical solution for the velocity of a pressure driven flow in a $b \times b$ square channel is [48]

$$\mathbf{v}(x, y) = -\frac{\nabla p}{2\eta} \left[\frac{b^2}{4} - y^2 - C(x) \right], \quad \text{with} \quad (38)$$

$$C(x) = \frac{8b^2}{\pi^3} \sum_{n=0}^{\infty} (-1)^n \frac{\cosh((2n+1)\pi x/b) \cos((2n+1)\pi y/b)}{(2n+1)^3 \cosh((2n+1)\pi/2)}$$

where $x \in [-b/2, b/2]$ and $y \in [-b/2, b/2]$ are the coordinates in the cross section, with the origin in the center of the channel. The pressure gradient ∇p is imposed by pressure boundaries and the dynamic viscosity is known from equation (8). The infinite sum in equation (38) can be truncated when a given accuracy is reached. We sum up 50 terms and compare this approximation to our numerical results on a 32^3 domain, i.e., $b = 15$ plus one layer of boundary nodes. In figure 4(a) we compare the analytical solution from equation (38) with our simulation results. The velocity in the z -direction is averaged over the y - and z -directions and the averaged value is plotted against the position in the x -direction. A very good agreement of the numerical result with the analytical solution can be seen. For comparison, results for the node based bounce back rule are shown. For this boundary condition it is known that it is only first-order accurate, which can be seen in the kink in the velocity profile close to the boundary nodes. It can however be made second-order accurate by choosing the position of the wall somewhere (depending on

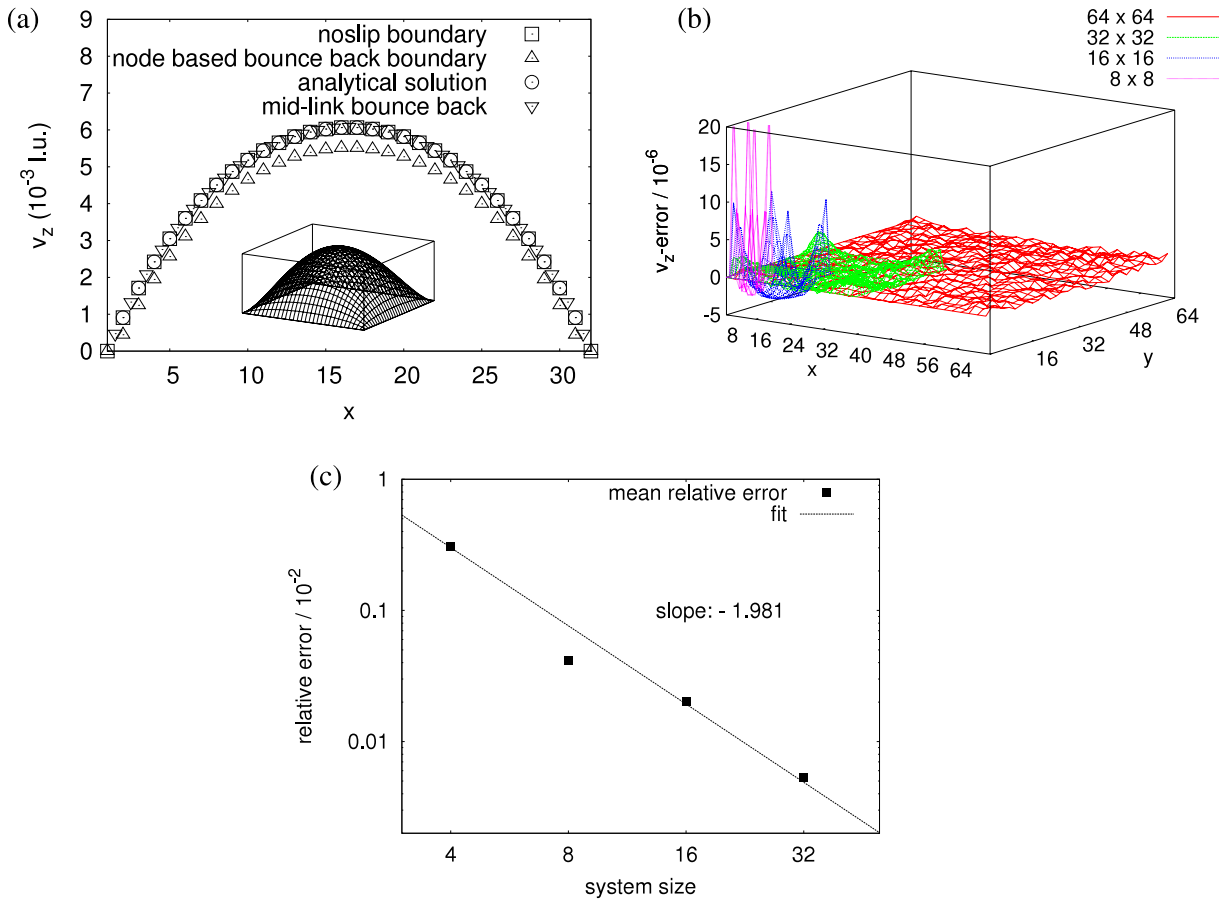


Figure 4. (a) Velocity profile in a square channel averaged along the yz -planes for the no-slip boundary condition (squares), for the node based bounce back rule (triangles), mid-link bounce back (top-down triangles), and the analytical solution. The no-slip boundary condition collapses with the analytical solution, whereas the on-site bounce back boundary condition shows a kink in the profile close to the boundary nodes. The numerical results are obtained on a 32^3 lattice and for the analytical solution the sum in equation (38) is truncated after 50 terms. A 2D profile of the z -velocity in an xy cross section is displayed as an inset. (b) The relative error for system sizes between 8 and 64 nodes: in the corners next to the boundary the largest relative deviations occur. Note that on the boundary nodes, where the truncation error of equation (38) is largest, the deviation between the simulation and approximated analytical solution is negligible. Therefore, the deviation can be taken as a measure for the quality of the numerical result. (c) The averaged error for different lattice resolutions versus the number of lattice nodes in each dimension confirming the second-order accuracy of the no-slip boundary condition.

the BGK relaxation time) in between two nodes, which is known as the mid-grid bounce back [2]. As one can see in figure 4(a), if the wall position is chosen correctly for the bounce back rule, a satisfying accuracy can be achieved, too (top-down triangles). Note that the position of the wall is shifted by half a lattice unit due to the different approach at the wall.

In figure 4(b) the relative error depending on the size of the simulation is studied. There are three different errors involved. The truncation errors of the sum in equation (38) cannot be seen in this figure. If the sum is truncated after just a few terms, the error increases on two of the edges. In the corners of the simulation domain the errors due to the discretization on the lattice remain. This is what determines the accuracy for relatively small simulations. However, if the lattice is refined, this error decreases. It decreases with the square of the lattice constant which is typical for second-order accurate schemes. Another error which dominates for large systems is the floating point precision which is reflected in noisy data in the center of the simulation domain. This error is independent on the lattice constant. In figure 4(b) the relative error is shown for different system sizes. In the corners the error decreases with the system size, whereas the noise in the center is independent of the system size. In figure 4(c) we plot the mean error averaged over the whole system against the number of lattice nodes used for computation in each dimension. The slope of approximately 2 confirms the second-order accuracy of the boundary condition. For the simulations presented in figure 4 pressure boundaries according to equation (12) are used and on the walls and edges we apply no-slip conditions as described in section 5. For this plot we use only the range in which the lattice size dependent error dominates. For 64 lattice nodes in each dimension, the floating point precision in one of our post-processing steps dominates the overall error. Therefore, we only use the smaller systems for this investigation. The inset in figure 4 shows the analytical velocity profile in a cross section perpendicular to the extension of the square channel.

7. Conclusion

We have derived an explicit local on-site flux boundary condition for LB simulations on a D3Q19 lattice. Velocity terms up to second order enter the derivation and this accuracy is also confirmed in the numerical tests. The influx and outflux velocity underlies there being no restrictions in any particular direction. We have demonstrated the numerical accuracy by comparing simulation results for a flow through a tilted channel with the theoretical expectation of a Poiseuille flow. Remaining errors can be assigned to the discretization on the lattice and to rounding errors due to the floating point representation. We have tested the boundary condition in the simulation of Poiseuille flow between two planar walls and in shear flow. In those tests the simulation data fit exactly to the analytical solutions without any slip parameter and independently of the BGK relaxation time. For this test we have used no-slip boundary conditions which are a special case included in the general velocity boundary conditions. Finally, we have tested the boundary condition by simulating the flow through a square channel. The scaling of the numerical error with the lattice resolution again confirms the second-order accuracy.

Acknowledgments

The German Research Foundation (DFG) is acknowledged for financial support (SFB 716 and EAMatWerk). We thank A Narváez, B Dünweg, and U D Schiller for fruitful discussions.

Appendix

Here we give the expressions for the other boundaries not treated explicitly in the text. We start with the top plane where we implement the outflux for our simulations. We obtain

$$\rho = \frac{1}{v_z + 1} [f_1 + f_2 + f_3 + f_4 + f_7 + f_{11} + f_{12} + f_8 + f_{19} + 2(f_5 + f_9 + f_{13} + f_{15} + f_{17})] \quad (\text{A.1})$$

or

$$v_z = -1 + \frac{1}{\rho_0} [f_1 + f_2 + f_3 + f_4 + f_7 + f_{11} + f_{12} + f_8 + f_{19} + 2(f_5 + f_9 + f_{13} + f_{15} + f_{17})] \quad (\text{A.2})$$

with v_z defined in positive z -direction. Here, the undetermined populations after the streaming step are

$$f_6 = f_5 - \frac{1}{3}\rho v_z, \quad (\text{A.3})$$

$$f_{10} = f_{13} + \frac{\rho}{6}(-v_z + v_x) - N_x^z, \quad (\text{A.4})$$

$$f_{14} = f_9 + \frac{\rho}{6}(-v_z - v_x) + N_x^z, \quad (\text{A.5})$$

$$f_{16} = f_{17} + \frac{\rho}{6}(-v_z + v_y) - N_y^z, \quad (\text{A.6})$$

$$f_{18} = f_{15} + \frac{\rho}{6}(-v_z - v_y) + N_y^z \quad (\text{A.7})$$

with N_x^z and N_y^z defined as previously, in equations (25) and (26).

For the left, right, front and back boundaries, which we do not use in this work, one finds the following expressions. For the left ($x = 0$) boundary,

$$\rho = \frac{1}{v_x + 1} [f_3 + f_4 + f_5 + f_6 + f_{15} + f_{16} + f_{17} + f_{18} + f_{19} + 2(f_2 + f_{11} + f_{12} + f_{13} + f_{14})], \quad (\text{A.8})$$

or

$$v_x = -1 + \frac{1}{\rho_0} [f_3 + f_4 + f_5 + f_6 + f_{15} + f_{16} + f_{17} + f_{18} + f_{19} + 2(f_2 + f_{11} + f_{12} + f_{13} + f_{14})], \quad (\text{A.9})$$

and

$$f_1 = f_2 + \frac{1}{3}\rho v_x, \quad (\text{A.10})$$

$$f_8 = f_{11} + \frac{\rho}{6}(v_x - v_y) + N_y^x, \quad (\text{A.11})$$

$$f_7 = f_{12} + \frac{\rho}{6}(v_x + v_y) - N_y^x, \quad (\text{A.12})$$

$$f_9 = f_{14} + \frac{\rho}{6}(v_x + v_z) - N_z^x, \quad (\text{A.13})$$

$$f_{10} = f_{13} + \frac{\rho}{6}(v_x - v_z) + N_z^x \quad (\text{A.14})$$

with

$$N_y^x = \frac{1}{2}[f_3 + f_{15} + f_{16} - (f_4 + f_{17} + f_{18})] - \frac{1}{3}\rho v_y, \quad (\text{A.15})$$

$$N_z^x = \frac{1}{2}[f_5 + f_{11} + f_{15} - (f_6 + f_{16} + f_{18})] - \frac{1}{3}\rho v_z. \quad (\text{A.16})$$

At the right boundary we have

$$\rho = \frac{1}{1 - v_x} [f_3 + f_4 + f_5 + f_6 + f_{15} + f_{16} + f_{17} + f_{18} + f_{19} + 2(f_1 + f_7 + f_8 + f_9 + f_{10})], \quad (\text{A.17})$$

or

$$v_x = 1 - \frac{1}{\rho_0} [f_3 + f_4 + f_5 + f_6 + f_{15} + f_{16} + f_{17} + f_{18} + f_{19} + 2(f_1 + f_7 + f_8 + f_9 + f_{10})], \quad (\text{A.18})$$

and

$$f_2 = f_1 - \frac{1}{3}\rho v_x, \quad (\text{A.19})$$

$$f_{11} = f_8 + \frac{\rho}{6}(-v_x + v_y) - N_y^x, \quad (\text{A.20})$$

$$f_{12} = f_7 + \frac{\rho}{6}(-v_x - v_y) + N_y^x, \quad (\text{A.21})$$

$$f_{14} = f_9 + \frac{\rho}{6}(-v_x - v_z) + N_z^x, \quad (\text{A.22})$$

$$f_{13} = f_{10} + \frac{\rho}{6}(-v_x + v_z) - N_z^x. \quad (\text{A.23})$$

At the front ($y = 0$) boundary, one finds

$$\rho = \frac{1}{1 - v_y} [f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} + 2(f_4 + f_8 + f_{12} + f_{17} + f_{18})], \quad (\text{A.24})$$

or

$$v_y = 1 - \frac{1}{\rho_0} [f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} + 2(f_4 + f_8 + f_{12} + f_{17} + f_{18})], \quad (\text{A.25})$$

and

$$f_3 = f_4 + \frac{1}{3}\rho v_y, \quad (\text{A.26})$$

$$f_7 = f_{12} + \frac{\rho}{6}(v_y + v_x) - N_x^y, \quad (\text{A.27})$$

$$f_{11} = f_8 + \frac{\rho}{6}(v_y - v_x) + N_x^y, \quad (\text{A.28})$$

$$f_{15} = f_{18} + \frac{\rho}{6}(v_y + v_z) - N_z^y, \quad (\text{A.29})$$

$$f_{16} = f_{17} + \frac{\rho}{6}(v_y - v_z) + N_y^z, \quad (\text{A.30})$$

with

$$N_x^y = \frac{1}{2}[f_1 + f_9 + f_{10} - (f_2 + f_{13} + f_{14})] - \frac{1}{3}\rho v_x, \quad (\text{A.31})$$

$$N_z^y = \frac{1}{2}[f_5 + f_9 + f_{13} - (f_6 + f_{10} + f_{14})] - \frac{1}{3}\rho v_z. \quad (\text{A.32})$$

At the back the density is given by

$$\rho = \frac{1}{v_y + 1}[f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} + 2(f_3 + f_7 + f_{11} + f_{15} + f_{16})], \quad (\text{A.33})$$

or the velocity reads

$$v_x = -1 + \frac{1}{\rho_0}[f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} + 2(f_3 + f_7 + f_{11} + f_{15} + f_{16})], \quad (\text{A.34})$$

and the distributions are

$$f_4 = f_3 - \frac{1}{3}\rho v_y, \quad (\text{A.35})$$

$$f_{12} = f_7 + \frac{\rho}{6}(-v_y - v_x) + N_x^y, \quad (\text{A.36})$$

$$f_8 = f_{11} + \frac{\rho}{6}(-v_y + v_x) - N_x^y, \quad (\text{A.37})$$

$$f_{18} = f_{15} + \frac{\rho}{6}(-v_y - v_z) + N_z^y, \quad (\text{A.38})$$

$$f_{17} = f_{16} + \frac{\rho}{6}(-v_y + v_z) - N_y^z. \quad (\text{A.39})$$

References

- [1] Zou Q and He X, 1997 *Phys. Fluids* **9** 1591
- [2] Succi S, 2001 *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond* (Oxford: Oxford University Press)
- [3] Chen H, Chen S and Matthaeus W H, 1992 *Phys. Rev. A* **45** R5339

- [4] Higuera F J, Succi S and Benzi R, 1989 *Europhys. Lett.* **9** 345
- [5] Kutay M E, Aydilek A H and Masad E, 2006 *Comput. Geotech.* **33** 381
- [6] Martys N S and Chen H, 1996 *Phys. Rev. E* **53** 743
- [7] Ladd A J C, 1994 *J. Fluid Mech.* **271** 285
- [8] Ladd A J C, 1994 *J. Fluid Mech.* **271** 311
- [9] Ladd A J C and Verberg R, 2001 *J. Stat. Phys.* **104** 1191
- [10] Harting J, Herrmann H and Ben-Naim E, 2008 *Europhys. Lett.* **83** 30001
- [11] Shan X and Chen H, 1993 *Phys. Rev. E* **47** 1815
- [12] Shan X and Chen H, 1994 *Phys. Rev. E* **49** 2941
- [13] Chen H, Boghosian B M, Coveney P V and Nekovee M, 2000 *Proc. R. Soc. A* **456** 2043
- [14] Harting J, Chin J, Venturoli M and Coveney P V, 2005 *Phil. Trans. R. Soc. A* **363** 1895
- [15] Chin J and Coveney P V, 2002 *Phys. Rev. E* **66** 016303
- [16] Frisch U, d'Humières D, Hasslacher B, Lallemand P, Pomeau Y and Rivet J P, 1987 *Complex Syst.* **1** 649
- [17] Ginzburg I and Steiner K, 2003 *J. Comput. Phys.* **185** 61
- [18] Skordos P A, 1993 *Phys. Rev. E* **48** 4823
- [19] Ginzbourg I and d'Humières D, 1996 *J. Stat. Phys.* **84** 927
- [20] Mattila K, Hyväluoma J and Rossi T, 2009 *J. Stat. Mech.* P06015
- [21] Qian Y, 1997 *Int. J. Mod. Phys. C* **8** 753
- [22] Zou Q, Hou S, Chen S and Doolen G D, 1995 *J. Stat. Phys.* **81** 35
- [23] Karniadakis G, Beskok A and Aluru N, 2005 *Microflows and Nanoflows, Fundamentals and Simulation* (Berlin: Springer)
- [24] Hessel V, Löwe H and Schönfeld F, 2005 *Chem. Eng. Sci.* **60** 2479
- [25] Inamuro T, Yoshino M and Ogino F, 1995 *Phys. Fluids* **7** 2928
- [26] Maier R S, Bernard R S and Grunau D W, 1996 *Phys. Fluids* **8** 1788
- [27] Flekkøy E G, Wagner G and Feder J, 2000 *Europhys. Lett.* **52** 271
- [28] Delgado-Buscalioni R and Coveney P V, 2003 *Phys. Rev. E* **67** 046704
- [29] Delgado-Buscalioni R, Kremer K and Praprotnik M, 2008 *J. Chem. Phys.* **128** 114110
- [30] Dupuis A, Kotsalis E M and Koumoutsakos P, 2007 *Phys. Rev. E* **75** 046704
- [31] Bhatnagar P L, Gross E P and Krook M, 1954 *Phys. Rev.* **94** 511
- [32] McNamara G R and Zanetti G, 1988 *Phys. Rev. Lett.* **61** 2332
- [33] He X and Luo L S, 1997 *Phys. Rev. E* **56** 6811
- [34] Qian Y H, d'Humières D and Lallemand P, 1992 *Europhys. Lett.* **17** 479
- [35] Tang G H, Tao W Q and He Y L, 2005 *Phys. Rev. E* **72** 016703
- [36] Sbragaglia M and Succi S, 2005 *Phys. Fluids* **17** 093602
- [37] Chen S, Martínez D and Mei R, 1996 *Phys. Fluids* **8** 2527
- [38] Hänel D, 2004 *Molekulare Gasdynamik: Einführung in Die Kinetische Theorie der Gase und Lattice-Boltzmann-Methoden* (Berlin: Springer)
- [39] Sukop M C and Thorne D T, 2007 *Lattice Boltzmann Modeling* (Berlin: Springer)
- [40] Ansumali S and Karlin I V, 2002 *Phys. Rev. E* **66** 026311
- [41] d'Orazio A, Succi S and Arrighetti C, 2003 *Phys. Fluids* **15** 2778
- [42] Verberg R and Ladd A J C, 2001 *Phys. Rev. E* **65** 016701
- [43] Chun B and Ladd A J C, 2007 *Phys. Rev. E* **75** 066705
- [44] Schiller U D, 2008 *PhD Thesis* Johannes Gutenberg-Universität Mainz, Germany
<http://www.mpip-mainz.mpg.de/theory.html/Theses>
- [45] Latt J, Chopard B, Malaspinas O, Deville M and Michler A, 2008 *Phys. Rev. E* **77** 056703
- [46] Halliday I, Hammond L A and Care C M, 2002 *J. Phys. A: Math. Gen.* **35** L157
- [47] Guo Z, Zheng C and Shi B, 2002 *Phys. Rev. E* **65** 046308
- [48] Wieghardt K, 1957 *Theoretische Strömungslehre* (Göttingen: Universitätsverlag Göttingen)