Journal of Statistical Physics, Vol. 121, Nos. 1/2, October 2005 (© 2005)
DOI: 10.1007/s10955-005-8321-2

Asymptotic Analysis of Lattice Boltzmann Boundary Conditions

Michael Junk[1] **and Zhaoxia Yang**[1]

Received October 12, 2004; accepted June 22, 2005

In this article, we use a general method for the analysis of finite difference schemes to investigate lattice Boltzmann algorithms for Navier–Stokes problems with Dirichlet boundary conditions. Several link based boundary conditions for commonly used lattice Boltzmann BGK models are considered. With our method, the accuracy of the algorithms can be exactly predicted. Moreover, the analytical results can be used to construct new algorithms which is demonstrated with a corrected bounce back rule that requires only local evaluations but still yields second order accuracy for the velocity. The analysis is applicable to general geometries and instationary flows.

KEY WORDS: lattice Boltzmann equation; bounce back rule; asymptotic analysis; incompressible Navier–Stokes equation; Dirichlet boundary conditions.

1. INTRODUCTION

Among the numerical approximation methods for the incompressible Navier–Stokes equation, the lattice Boltzmann method seems to be particular because it relies on an indirect approach to continuum equations via kinetic theory. Mathematically, the kinetic equation is asymptotically connected to the Navier–Stokes system by a *singular limit*. A consequence of this somewhat involved relation is that numerical approximations on the kinetic level influence the approximate Navier–Stokes solution in a way which is difficult to predict. For example, if Dirichlet boundary conditions are to be implemented, one cannot prescribe directly the flow velocity at boundary nodes but one has to set certain variables in the kinetic equation in such a way that the average velocity satisfies the required conditions.

[1]FB Mathematik, Universität des Saarlandes, Postfach 15 11 50, 66041 Saarbrücken, Germany; e-mail: michael.junk@uni-konstanz.de

Typically, the required number of kinetic conditions exceeds the available conditions from the Navier–Stokes problems. This indicates already that the kinetic conditions have to be chosen carefully in order to avoid the appearance of extra conditions on the Navier–Stokes level which would render the problem ill posed (leading to an unwanted behavior on the grid scale like boundary layers, oscillations etc.).

In order to decide whether a specific lattice Boltzmann algorithm leads to the desired Navier–Stokes boundary conditions, we propose a method which is based on a straightforward asymptotic analysis and which is generally applicable to any finite difference method.[1] In fact, such expansions are widely used in the theory of ordinary differential equations and also frequently applied to PDEs in connection with Richardson's extrapolation or deferred correction methods (see ref. 1 for a review). In other words, lattice Boltzmann algorithms can be fully analyzed with standard tools so that the approach is only special because it involves a singular asymptotic limit. But even this aspect is not uncommon. It appears, for example, as central idea in relaxation schemes which originated in the work[2] and which shares several features with lattice Boltzmann algorithms.

With our approach, we divert from the Chapman Enskog analysis which is usually taken as basis for the analysis of lattice Boltzmann schemes (see for example refs. 3–8). Compared to the Chapman Enskog approach which is usually based on the resolution of two time scales (the diffusive and the acoustic time scale), we restrict to the diffusive scaling only. In this way, we can investigate directly the relation between the lattice Boltzmann method and the *incompressible* Navier–Stokes equation which is our main goal (note that with this point of view, acoustic effects are considered as numerical errors of the lattice Boltzmann method).

The technical advantages of the advocated method are twofold. First, only a single time scale expansion is necessary and it is simpler than a two-scale expansion because the expansion coefficients and the corresponding equations depend on one variable fewer. Second, the expansion coefficients depend directly on the aspired solution of the incompressible Navier–Stokes problem and not on the solution of a different problem (the compressible equation). Having this direct dependence, it is straight forward to relate the numerical solution to the exact solution, for example, to obtain error estimates.

An obvious difference between the classical Chapman–Enskog analysis and our approach here is the relation $\Delta t = \Delta x^2$ between the scaled time and space step which deviates from $\Delta t = \Delta x$ in the Chapman–Enskog case. This difference can easily be traced back to the choice of the time scale in the two approaches. To illustrate this point, let us think of a flow through

a channel of length L with typical velocity U. If the grid spacing is δx, we generally assume that $\Delta x = \delta x / L$ is sufficiently small so that the grid resolution is good. In the classical scaling, velocities are measured in terms of a typical particle speed c which gives rise to the time unit $T = L/c$. The relation $\delta x = c\delta t$ between space and time step then translates into $\Delta x = \Delta t$ for the scaled grid parameters $\Delta x = \delta x / L$ and $\Delta t = \delta t / T$. We stress that the time unit T is natural for the investigation of acoustic effects. If low Mach number flows are considered, the typical flow velocity U must be small compared to c. More precisely, one needs $U/c \sim \Delta x$ to approach the incompressible limit for $\Delta x \to 0$. Consequently, the dimensionless time required for a volume of fluid to traverse the channel is proportional to $L/(UT) = \mathcal{O}(1/\Delta x)$ and thus diverges for $\Delta x \to 0$ while the flow velocity tends to zero. Exactly to avoid this technical inconvenience, we divert with our scaling from the classical choice.

Specifically, we measure velocities in units of U and, accordingly, time in units of L/U. In this scaling, the flow velocity is of order one and a volume of fluid needs always the same non-dimensional time to traverse the channel, independent of Δx. The relation $\delta x = c\delta t$ transforms into $\Delta x = c\Delta t / U$ where Δt is now δt scaled by L/U. The low Mach-number assumption $U/c = \Delta x$ then leads to $\Delta t = \Delta x^2$. Note however, that Δt differs in the two approaches by a different scaling of the time step δt.

We remark that the direct asymptotic analysis using the diffusive scaling has first been considered by Sone in ref. 9 and earlier works for the classical Boltzmann equation and has been applied to lattice Boltzmann in refs. 10–13. Note, however, that knowledge about these kinetic methods is *not* required in our approach. In the same spirit, we do not introduce other physical non-dimensional parameters like the Mach and Knudsen number (which would both be proportional to our non-dimensional grid spacing $h = \Delta x$). As far as the treatment of boundary conditions is concerned, we will comment on the connections of our method to the approach in refs. 15, 14, 22 which uses Chapman–Enskog analysis.

The introduction is concluded with an outline of the article. In Section 2 we specify the basic lattice Boltzmann algorithms with bounce back rule which are used to demonstrate our method. The detailed asymptotic analysis is given in Section 3. Finally, we use the information obtained from the asymptotic analysis to improve the bounce back rule and to investigate other existing link-based boundary conditions.[14,15,18,22]

2. THE LATTICE BOLTZMANN ALGORITHM ON DOMAINS WITH BOUNDARIES

We consider the incompressible Navier–Stokes equation on a domain $\Omega \subset \mathbb{R}^d$ with initial and Dirichlet boundary values. Our aim is to find numerical approximations of the fields $\boldsymbol{u}: [0, T] \times \Omega \to \mathbb{R}^d$ and $p: [0, T] \times \Omega \to \mathbb{R}$, which satisfy

$$\nabla \cdot \boldsymbol{u} = 0, \qquad \partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p = \nu \nabla^2 \boldsymbol{u} + \boldsymbol{G}, \qquad \boldsymbol{u}|_{t=0} = \boldsymbol{\psi} \tag{1}$$

with

$$\boldsymbol{u}(t, \boldsymbol{x}) = \boldsymbol{\phi}(t, \boldsymbol{x}), \quad t \in [0, T], \ \boldsymbol{x} \in \partial\Omega \tag{2}$$

where $\boldsymbol{\psi}, \boldsymbol{\phi}$ and $\boldsymbol{G}$ are given functions, functions $\boldsymbol{\psi}: \mathbb{R}^d \to \mathbb{R}^d$ and $\boldsymbol{G}: [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ represent a divergence free initial velocity field and a force term respectively.

We assume (1) and (2) to be in non-dimensional form, i.e. the domain Ω is scaled by a typical length (say the diameter is equal to one) and the data of the problem $(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{G})$ is scaled in such a way that the typical velocity $\boldsymbol{u}$ is also of order one. Defining the time scale using the velocity and space scale as indicated in the introduction, the viscosity parameter ν is actually the inverse of the Reynolds number.

Lattice Boltzmann methods for the problem (1) are based on a simplified microscopic model of the fluid in which particles travel with discrete velocities in the directions $\boldsymbol{c}_0, \ldots, \boldsymbol{c}_b$ over a regular spatial lattice in such a way that their average velocity approximately satisfies (1). The lattice should be compatible with the velocities in the sense that neighboring nodes are connected by vectors $h\boldsymbol{c}_i$ where $h > 0$ is a small dimensionless parameter which regulates the grid resolution. The basic quantities $\hat{f}_i(n, \boldsymbol{j})$ denote the mass densities of particles having discrete velocity $\boldsymbol{c}_i$ at the lattice node labeled with $\boldsymbol{j} \in \mathbb{Z}^d$ (corresponding to $\boldsymbol{x}_{\boldsymbol{j}}(h) = h\boldsymbol{j}$) and time step $n \in \mathbb{N}_0$ (corresponding to $t_n(h) = h^2 n$). Given the particle mass densities $\hat{f}_i$ for the different velocities, the total mass density $\hat{\rho}$ is simply the sum

$$\hat{\rho}(n, \boldsymbol{j}) = \sum_i \hat{f}_i(n, \boldsymbol{j}). \tag{3}$$

Since in our setup, $\hat{\rho}$ will always be close to one (incompressible flow) the values $\hat{f}_i$ approximately form a discrete probability distribution

so that

$$\hat{\boldsymbol{u}}(n, \boldsymbol{j}) = \sum_i \hat{f}_i(n, \boldsymbol{j})\boldsymbol{c}_i \tag{4}$$

is essentially the average particle velocity (in our notation, a hat superscript indicates a discrete function of the grid labels). As a result of our analysis, we will see that the field $\hat{\boldsymbol{u}}$ approximates the scaled Navier–Stokes solution $h\boldsymbol{u}$.

We remark that time and space variables are non-dimensionalized using the same scales as in the Navier–Stokes problem (1). This usage differs slightly from the classical approach but has certain advantages if one wants to investigate the connection of the lattice Boltzmann method to the incompressible Navier–Stokes equation in the limit $h \to 0$ (see also the comments in the introduction).

The evolution of the particles consists of two phases, a transport step and a collision and forcing step. The latter simulates a spatially local particle interaction at the lattice nodes and the effect of the force field on the particles. It has the form

$$\hat{f}_i^{\mathrm{c}}(n, \boldsymbol{j}) = \hat{f}_i(n, \boldsymbol{j}) + C_i[\hat{f}](n, \boldsymbol{j}) + \hat{g}_i(n, \boldsymbol{j}) \tag{5}$$

where the so-called collision operator is of relaxation type

$$C_i[\hat{f}](n, \boldsymbol{j}) = \frac{1}{\tau}(f_i^{\mathrm{eq}}(\hat{\rho}(n, \boldsymbol{j}), \hat{\boldsymbol{u}}(n, \boldsymbol{j})) - \hat{f}_i(n, \boldsymbol{j})), \qquad \tau = \frac{1}{2} + 3\nu$$

and $\hat{\rho}$ and $\hat{\boldsymbol{u}}$ depend on $\hat{f}$ according to (3), and (4). As equilibrium distribution function we use

$$f_i^{\mathrm{eq}}(\rho, \boldsymbol{u}) = [\rho + 3\boldsymbol{u}\cdot\boldsymbol{c}_i + \frac{9}{2}(\boldsymbol{u}\cdot\boldsymbol{c}_i)^2 - \frac{3}{2}|\boldsymbol{u}|^2]f_i^*$$

which works in connection with the D2Q9, D3Q15, D3Q19, and D3Q27 models (for details see ref. 15). In our context, they differ only in the underlying velocity set and the weights $f_i^* = f_i^{\mathrm{eq}}(1, 0)$ defining the equilibrium distribution. The function $\hat{g}_i$ in (5) models the influence of the force term. Its relation to $\boldsymbol{G}$ in (1) is

$$\hat{g}_i(n, \boldsymbol{j}) = 3h^3 f_i^* \boldsymbol{c}_i \cdot \boldsymbol{G}(t_n(h), \boldsymbol{x}_{\boldsymbol{j}}(h)). \tag{6}$$

We remark that only by scaling $\boldsymbol{G}$ with h^3 we can ensure that, in the limit $h \to 0$, the force term acts on the velocity according to Eq. (1). The physical reason is that in a time step of length h^2, the Navier–Stokes velocity $\boldsymbol{u}$ changes proportional to $h^2\boldsymbol{G}$ and since $\hat{\boldsymbol{u}} \approx h\boldsymbol{u}$, the average lattice Boltzmann velocity only changes proportional to $hh^2\boldsymbol{G}$ which explains the factor h^3.

During collision, the original particle distribution function $\hat{f}_i$ transforms into the post-collisional state $\hat{f}_i^c$. After that, the particles simply move undisturbed with their velocities to the neighboring lattice sites. This transport step is described by the update rule

$$\hat{f}_i(n+1, \boldsymbol{j} + \boldsymbol{c}_i) = \hat{f}_i^c(n, \boldsymbol{j}). \tag{7}$$

Since the particles travel a distance h in a single time step of size h^2, the associated particle speed is actually $h/h^2 = 1/h$. This reflects the low Mach number assumption in our scaling where the flow speed is scaled to one so that the quotient between flow and particle speed can only become small if the particle speed diverges.

Obviously the update rule (7) determines $\hat{f}_i(n+1, \boldsymbol{j})$ only at interior nodes $\boldsymbol{x}_{\boldsymbol{j}}(h)$ whose neighbors in all directions are also in the computational domain. The labels of the remaining nodes (so-called boundary nodes) are collected in the set $J_{\mathrm{bdr}}(h)$. They require a modified update rule. For example, if node $\boldsymbol{j} \in J_{\mathrm{bdr}}(h)$ has a missing neighbor in direction $-\boldsymbol{c}_i$, then the population $\hat{f}_i(n+1, \boldsymbol{j})$ cannot be filled with the usual update rule (7). Classically, the bounce back rule is used to define these populations

$$\hat{f}_i(n+1, \boldsymbol{j}) = \hat{f}_{i^*}^b(n, \boldsymbol{j}) = \hat{f}_{i^*}^c(n, \boldsymbol{j}) + 6hf_i^*\boldsymbol{c}_i \cdot \boldsymbol{\phi}(t_n, \boldsymbol{x}_{\boldsymbol{j}i}) \tag{8}$$

which is employed for $\boldsymbol{j} \in J_{\mathrm{bdr}}(h)$ and the incoming velocities with indices in

$$V_{\boldsymbol{j}} = \{i : \boldsymbol{x}_{\boldsymbol{j}} - h\boldsymbol{c}_i \notin \Omega\}.$$

The velocity index i^* is defined by $\boldsymbol{c}_{i^*} = -\boldsymbol{c}_i$. The boundary value $\boldsymbol{\phi}$ is evaluated at the point $\boldsymbol{x}_{\boldsymbol{j}i} = \boldsymbol{x}_{\boldsymbol{j}} - hq_{\boldsymbol{j}i}\boldsymbol{c}_i \in \partial\Omega$ where $q_{\boldsymbol{j}i} \in [0, 1)$ represents the distance to the boundary along direction $\boldsymbol{c}_{i^*} = -\boldsymbol{c}_i$ in units of $|\boldsymbol{c}_i|$ (see Fig. 1).

To start the evolution, we use the initialization proposed in ref. 18.

$$\hat{f}_i(0, \boldsymbol{j}) = f_i^{\mathrm{eq}}(1 + 3h^2 p(0, \boldsymbol{x}_{\boldsymbol{j}}), h\boldsymbol{\psi}(\boldsymbol{x}_{\boldsymbol{j}})) - 3h^2\tau\boldsymbol{c}_i \cdot \nabla(\boldsymbol{c}_i \cdot \boldsymbol{\psi})(\boldsymbol{x}_{\boldsymbol{j}}) \tag{9}$$

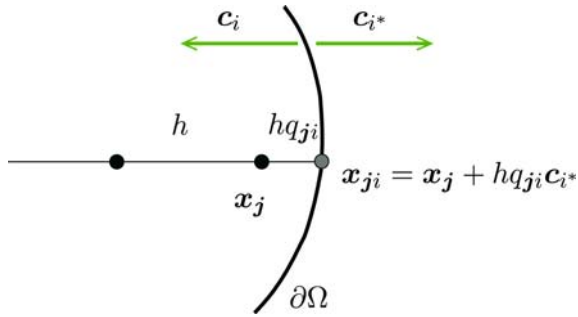

Fig. 1. Intersection of links and boundary give rise to $x_{ji} \in \partial\Omega$ (color online).

where $p(0, \boldsymbol{x})$ is the pressure corresponding to the initial velocity field $\boldsymbol{\psi}$. It is obtained by solving the Poisson equation

$$\Delta p = -\nabla \cdot (\boldsymbol{\psi} \cdot \nabla \boldsymbol{\psi}) + \nabla \cdot \boldsymbol{G}.$$

3. ASYMPTOTIC ANALYSIS

From an abstract point of view, the complete lattice Boltzmann algorithm presented in the previous section constitutes a finite set of equations for the unique determination of the quantities $\hat{f}_i(n, \boldsymbol{j})$. Since these equations depend on an additional small parameter h, it is natural to use tools from asymptotic analysis to understand the behavior of $\hat{f}_i(n, \boldsymbol{j})$. In particular, we want to know how well the average velocity $\hat{\boldsymbol{u}}$ approximates the solution of the Navier–Stokes problem (1). An answer to this question is obtained once we understand how the coefficients in the expansion

$$\hat{\boldsymbol{u}}(n, \boldsymbol{j}) = \boldsymbol{u}_0(t_n, \boldsymbol{x}_j) + h\boldsymbol{u}_1(t_n, \boldsymbol{x}_j) + h^2\boldsymbol{u}_2(t_n, \boldsymbol{x}_j) + \cdots$$

relate to the solution of (1). In fact, it will turn out that $\boldsymbol{u}_0 = 0$ and $\boldsymbol{u}_1$ is the solution of the Navier–Stokes equation (1), while $\boldsymbol{u}_2, \boldsymbol{u}_3, \ldots$, can be viewed as contributions to the velocity error.

More generally, we try to express the lattice Boltzmann variables through a regular expansion

$$\hat{f}_i(n, \boldsymbol{j}) = f_i^{(0)}(t_n, \boldsymbol{x}_j) + hf_i^{(1)}(t_n, \boldsymbol{x}_j) + h^2 f_i^{(2)}(t_n, \boldsymbol{x}_j) + \cdots \tag{10}$$

with smooth and h independent functions $f_i^{(k)}$. Inserting the expansion into all equations of the lattice Boltzmann method, performing Taylor

expansions and equating the expressions appearing in different orders of h separately to zero, a set of conditions on the expansion coefficients arises which can be used to determine the functions $f_i^{(k)}$. The expansion coefficients of the moments

$$\rho_m = \sum_i f_i^{(m)}, \qquad \boldsymbol{u}_m = \sum_i f_i^{(m)} \boldsymbol{c}_i$$

then follow from our knowledge of $f^{(m)}$. The resulting explicit information about the h dependence of density and average velocity $\hat{\rho}$, $\hat{\boldsymbol{u}}$ constitutes the main difference to the Chapman–Enskog analysis where $\hat{\rho}$ and $\hat{\boldsymbol{u}}$ are approximately described through the solutions of differential equations which carry h as a parameter. In particular, with (10) we can derive the relation between the numerical values $\hat{\rho}, \hat{\boldsymbol{u}}$ and the solution of the *incompressible* Navier–Stokes equation while the Chapman–Enskog analysis establishes a link to some intermediate compressible Navier–Stokes equation.

Formally, expansion (10) differs from the Chapman–Enskog expansion in that the mass and velocity averages of the expansion coefficients $f^{(k)}$ with $k \geqslant 1$ are generally non-zero. Also, the Chapman–Enskog expansion usually employs an additional time scale which is relevant for analyzing acoustic effects. In our approach, acoustic effects are also recovered but they typically appear only in the higher order terms (for example, $\boldsymbol{u}_3$ is not divergence free which reflects the weak compressibility of the numerical solution). In cases where the lattice Boltzmann solution contains acoustic effects at leading order (for example, in connection with a poor initialization), our single scale expansion will fail because sound waves imply very large time derivatives in our diffusive scaling which ultimately lead to singular behavior for $h \to 0$.

Hence, expansion (10) can only describe the regular part of the LB solution $\hat{f}_i(n, \boldsymbol{j})$ because of our smoothness assumption on the coefficients. If the numerical solution exhibits fast oscillations (sound waves) or strong gradients at the boundaries (Knudsen layers) the expansion (10) will fail at a certain order because the strong derivatives (proportional to $1/h$ or larger) associated to such effects cannot be represented with smooth coefficients. More precisely, if we find a contradiction in the process of determining the coefficient $f^{(m)}$, for example $f^{(m)}$ turns out to be discontinuous or the condition to determine $f^{(m)}$ has no solution, then we can conclude that $\hat{f}$ can be described by $f^{(0)} + \cdots + h^{m-1} f^{(m-1)}$ *and* that the next order contribution will not be regular. While it is possible to study irregular behavior in detail, for example with the help of a multiscale expansion instead of (10), we should keep in mind that the irregular

effects have to be interpreted as numerical errors as far as the approximation of Eq. (1) is concerned because they are related to physical situations which are not captured in the model (1). Therefore, we will restrict ourselves to the expansion (10) in this article. In fact, we will use the more specific form

$$\begin{aligned} &\hat{f}_i(n, \boldsymbol{j}) = f_i^{(0)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h f_i^{(1)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h^2 f_i^{(2)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + \cdots \\ &f_i^{(0)} = f_i^*, \quad \rho_1 = 0. \end{aligned} \tag{11}$$

This expansion restricts us to lattice Boltzmann solutions which are perturbations of the equilibrium distribution $f_i^* = f_i^{\mathrm{eq}}(1, 0)$ with unit density $\rho_0 = 1$ and zero velocity $\boldsymbol{u}_0 = 0$ (reflecting the low Mach number situation). Together with the second assumption $\rho_1 = 0$ it implies that the density has the form $\hat{\rho} = 1 + h^2 \rho_2 + \cdots$. Both assumptions are compatible with the presented algorithm and can actually be derived (at the expense of some technical arguments, see for example refs. 13, 16).

3.1. Asymptotic Analysis of the Update Rule

Let us start with the update rule of the algorithm. Inserting (11) into the left hand side of the transport step (7), we find expressions of the form

$$f_i^{(k)}(t_{n+1}(h), \boldsymbol{x}_{\boldsymbol{j}+\boldsymbol{c}_i}(h)) = f_i^{(k)}(t_n(h) + h^2, \boldsymbol{x}_{\boldsymbol{j}}(h) + h\boldsymbol{c}_i).$$

Since the functions $f_i^{(k)}$ are assumed to be smooth, we can perform a Taylor expansion around the point $(t_n, \boldsymbol{x}_{\boldsymbol{j}})$. After dropping the argument h of t_n and $\boldsymbol{x}_{\boldsymbol{j}}$ for brevity, we formally obtain an infinite series

$$f_i^{(k)}(t_n + h^2, \boldsymbol{x}_{\boldsymbol{j}} + h\boldsymbol{c}_i) = f_i^{(k)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + \sum_{r=1}^{\infty} h^r D_r(\partial_t, \boldsymbol{c}_i \cdot \nabla) f_i^{(k)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) \tag{12}$$

where $D_r(\theta, \sigma)$ are polynomials like

$$D_1(\theta, \sigma) = \sigma, \quad D_2(\theta, \sigma) = \theta + \sigma^2/2, \quad D_3(\theta, \sigma) = \sigma(\theta + \sigma^2/6), \ldots$$

The general expression is

$$D_r(\theta, \sigma) = \sum_{2a+b=r} \frac{\theta^a \sigma^b}{a!b!}, \qquad r \geqslant 0, \tag{13}$$

where θ replaces the time derivative and σ the directional space derivative $\boldsymbol{c}_i \cdot \nabla$. Altogether, the expansion of the left hand side of (7) is

$$\hat{f}_i(n+1, \boldsymbol{j}+\boldsymbol{c}_i) = \sum_{m \geqslant 0} h^m \left(f_i^{(m)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + \sum_{k=0}^{m-1} D_{m-k}(\partial_t, \boldsymbol{c}_i \cdot \nabla) f_i^{(k)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) \right), \tag{14}$$

Next, we have to expand the right hand side of (7) with the substitution of $\hat{f}_i$ by the expansion (11). Due to the nonlinear term in the collision operator, a mixing of orders occurs. If we denote the moments of $f^{(m)}$ by ρ_m and $\boldsymbol{u}_m$ respectively, we first find from (11) by summation

$$\begin{aligned} \hat{\rho}(n, \boldsymbol{j}) &= 1 + h^2 \rho_2(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + \cdots \\ \hat{\boldsymbol{u}}(n, \boldsymbol{j}) &= h \boldsymbol{u}_1(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h^2 \boldsymbol{u}_2(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + \cdots \end{aligned}$$

Inserting these expansions into f_i^{eq}, we obtain

$$f_i^{\mathrm{eq}}\left(\hat{\rho}(n, \boldsymbol{j}), \hat{\boldsymbol{u}}(n, \boldsymbol{j})\right) = f_i^{\mathrm{eq},(0)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h f_i^{\mathrm{eq},(1)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h^2 f_i^{\mathrm{eq},(2)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + \cdots$$

with $f_i^{\mathrm{eq},(0)} = f_i^*$ and

$$f_i^{\mathrm{eq},(m)} = f_i^* \left\{ \rho_m + 3 \boldsymbol{u}_m \cdot \boldsymbol{c}_i + \frac{9}{2} \sum_{k+l=m} \left[(\boldsymbol{u}_k \cdot \boldsymbol{c}_i)(\boldsymbol{u}_l \cdot \boldsymbol{c}_i) - \frac{1}{3} \boldsymbol{u}_k \cdot \boldsymbol{u}_l \right] \right\}.$$

Thus, the right hand side of (7) turns out to be

$$\sum_{m \geqslant 0} h^m \left[f_i^{(m)} + \frac{1}{\tau}(f_i^{\mathrm{eq},(m)} - f_i^{(m)}) + g_i \delta_{m3} \right](t_n, \boldsymbol{x}_{\boldsymbol{j}}), \tag{15}$$

where δ_{ij} is the Kronecker delta and

$$g_i(t, \boldsymbol{x}) = 3 f_i^* \boldsymbol{c}_i \cdot \boldsymbol{G}(t, \boldsymbol{x}).$$

Equating (14) and (15) we obtain in order m

$$f_i^{(m)} + \sum_{k=0}^{m-1} D_{m-k}(\partial_t, \boldsymbol{c}_i \cdot \nabla) f_i^{(k)} = f_i^{(m)} + \frac{1}{\tau}(f_i^{\mathrm{eq},(m)} - f_i^{(m)}) + g_i \delta_{m3}. \quad (16)$$

If we solve (16) for $f_i^{(m)}$, we find that $f_i^{(m)}$ can be expressed as a sum of $f_i^{\mathrm{eq},(m)}$ and derivatives of lower order coefficients $f_i^{(k)}$. In particular, we can use (16) to successively replace the lower order coefficients by equilibrium coefficients. Eventually, we find

$$f_i^{(m)} = f_i^{\mathrm{eq},(m)} + \sum_{k=0}^{m-1} E_{m-k}(\tau, \partial_t, \boldsymbol{c}_i \cdot \nabla) f_i^{\mathrm{eq},(k)} + F_m(\tau, \partial_t, \boldsymbol{c}_i \cdot \nabla) g_i \quad (17)$$

with recursively defined polynomials E_k and F_k (details are given in Appendix A). Inspecting (17) more closely, we notice that it actually defines $f_i^{(m)}$ once we know the moments $\rho_0, \ldots, \rho_m$ and $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_m$ which make up the equilibrium functions $f_i^{\mathrm{eq},(k)}$. Equations for these moments are obtained by taking corresponding averages of (17). Since

$$\sum_i f_i^{(m)} = \rho_m = \sum_i f_i^{\mathrm{eq},(m)}, \quad \sum_i f_i^{(m)} \boldsymbol{c}_i = \boldsymbol{u}_m = \sum_i f_i^{\mathrm{eq},(m)} \boldsymbol{c}_i$$

we see that (17) gives rise to partial differential equations for $\rho_0, \ldots, \rho_{m-1}$ and $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{m-1}$ because the moments of order m drop out. General expressions for the differential operators in these equations are again given in the Appendix. Here we just list the results related to the relevant orders.

In order $(m=1)$, we find the trivial condition $0=0$ which reflects the fact that our assumption $f_i^{(0)} = f_i^*$ in (11) does not lead to a contradiction. In order $(m=2)$, we obtain

$$\nabla \rho_1 = 0, \quad \nabla \cdot \boldsymbol{u}_1 = 0 \quad (18)$$

which is an incompressibility condition on $\boldsymbol{u}_1$ and $\nabla \rho_1 = 0$ is in accordance with our assumption $\rho_1(t, \boldsymbol{x}) = 0$. The equations for the next order are

$$\partial_t \boldsymbol{u}_1 + \boldsymbol{u}_1 \cdot \nabla \boldsymbol{u}_1 + \frac{1}{3} \nabla \rho_2 = \nu \Delta \boldsymbol{u}_1 + \boldsymbol{G}, \quad \nabla \cdot \boldsymbol{u}_2 = 0. \quad (19)$$

i.e. $\boldsymbol{u}_1$ is a solution of the incompressible Navier–Stokes equation with $\rho_2/3$ as associated pressure. Proceeding to the case $m=4$, the following equations occur

$$\begin{aligned} \partial_t \boldsymbol{u}_2 + \boldsymbol{u}_1 \cdot \nabla \boldsymbol{u}_2 + \boldsymbol{u}_2 \cdot \nabla \boldsymbol{u}_1 + \frac{1}{3}\nabla \rho_3 &= \nu \triangle \boldsymbol{u}_2, \\ \nabla \cdot \boldsymbol{u}_3 &= -\partial_t \rho_2 - \frac{1}{2}\nabla \cdot \boldsymbol{G} \end{aligned} \tag{20}$$

where the divergence condition for $\boldsymbol{u}_3$ has been simplified using the fact that $\boldsymbol{u}_1$ satisfies the Navier–Stokes equation. Together with the second condition of (19), this is a homogeneous generalized Oseen problem for $\boldsymbol{u}_2$ and ρ_3. In particular, if initial and boundary conditions for $\boldsymbol{u}_2$ vanish, the coefficients $\boldsymbol{u}_2$ and ρ_3 are zero. Moreover, we see from the divergence condition that $\boldsymbol{u}_3$ is non-zero whenever the Navier–Stokes pressure is time dependent (reflecting the compressible nature of the lattice Boltzmann algorithm). Keeping in mind that the average velocity has the expansion $\hat{\boldsymbol{u}} = h\boldsymbol{u}_1 + h^2\boldsymbol{u}_2 + h^3\boldsymbol{u}_3 + \cdots$, we conclude that $\hat{\boldsymbol{u}}/h$ yields a second order accurate approximation of the Navier–Stokes velocity if $\boldsymbol{u}_2 = 0$ because $h^2\boldsymbol{u}_3$ then acts as error term added to $\boldsymbol{u}_1$. Similarly, we can access the Navier–Stokes pressure using $(\hat{\rho}-1)/(3h^2)$ to second order accuracy if $\rho_3 = 0$ which also happens if $\boldsymbol{u}_2$ has zero initial and boundary values. We want to stress that these conclusions only apply if the regular expansion (11) correctly describes the lattice Boltzmann solution up to order h^3. In Section 3.3, we show that this assumption is violated when the bounce back rule is used. In this case, additional irregular coefficients need to be added to the expansion which may reduce the accuracy of pressure and velocity predicted in the regular case.

Information about the behavior of the coefficients at $t=0$ and $\partial\Omega$ will be extracted from the initial and boundary part of the LB algorithm. However, before we proceed to this analysis, we summarize our observations and specify the structure of the leading order coefficients using (17) and the detailed form of the operators E_k (see Appendix A),

$$\begin{aligned} f_i^{(0)} &= f_i^*, \\ f_i^{(1)} &= f_i^{\mathrm{eq},(1)} = 3\boldsymbol{c}_i \cdot \boldsymbol{u}_1 f_i^*, \\ f_i^{(2)} &= f_i^{\mathrm{eq},(2)} - \tau \boldsymbol{c}_i \cdot \nabla f_i^{(1)} \\ &= \rho_2 f_i^* + 3\boldsymbol{c}_i \cdot \boldsymbol{u}_2 f_i^* + \frac{3}{2}(3(\boldsymbol{c}_i \cdot \boldsymbol{u}_1)^2 - |\boldsymbol{u}_1|^2) f_i^* - \tau \boldsymbol{c}_i \cdot \nabla f_i^{(1)}. \end{aligned} \tag{21}$$

3.2. Asymptotic Analysis of the Initial Condition

In order to find information about the initial values of the expansion coefficients (21), we insert the expansion (11) into (9). Noting that

$$f_i^{\mathrm{eq}}(1+3h^2p, h\boldsymbol{\psi}) = f_i^*(1+3h\boldsymbol{c}_i\cdot\boldsymbol{\psi}) + h^2\left(3p + 3\boldsymbol{c}_i\cdot\boldsymbol{u}_2 + \frac{3}{2}(3(\boldsymbol{c}_i\cdot\boldsymbol{\psi})^2 - |\boldsymbol{\psi}|^2)\right)$$

a comparison of equal orders yields

$$\boldsymbol{u}_1(0,\boldsymbol{x}) = \boldsymbol{\psi}(\boldsymbol{x}), \quad \boldsymbol{u}_2(0,\boldsymbol{x}) = 0, \quad \rho_2(0,\boldsymbol{x}) = 3p(0,\boldsymbol{x})$$

so that $\boldsymbol{u}_1, \rho_2/3$ satisfies (1).

3.3. Asymptotic Analysis of the Bounce Back Rule

It remains to derive the boundary values of the relevant moments which we obtain by inserting expansion (11) into (8) and later using the specific form (21) of the coefficients.

Expanding around the point $(t_n, \boldsymbol{x}_{ji})$ the computations are very similar to the analysis of the update rule and we can use the same operators D_k. Taking into account that $\boldsymbol{x}_{ji} = \boldsymbol{x}_j + hq_{ji}\boldsymbol{c}_i$, the directional derivative is now $q_{ji}\boldsymbol{c}_i\cdot\nabla$ instead of $\boldsymbol{c}_i\cdot\nabla$. In accordance with (14) we have (defining $D_0(\theta,\sigma) = 1$)

$$\hat{f}_i(n+1, \boldsymbol{j}) = \sum_{m\geqslant 0} h^m \sum_{k=0}^{m} D_{m-k}(\partial_t, q_{ji}\boldsymbol{c}_i\cdot\nabla) f_i^{(k)}(t_n, \boldsymbol{x}_{ji}).$$

The right hand side $\hat{f}_{i^*}^b(n, \boldsymbol{j})$ of (8) has a structure similar to the collision product. Expanding also around $\boldsymbol{x}_{ji} = \boldsymbol{x}_j + hq_{ji}\boldsymbol{c}_i$, we arrive at

$$\begin{aligned}\hat{f}_{i^*}^b(n, \boldsymbol{j}) &= h\beta_i(t_n, \boldsymbol{x}_{ji}) \\ &\quad + \sum_{m\geqslant 0} h^m \sum_{k=0}^{m} D_{m-k}(0, q_{ji}\boldsymbol{c}_i\cdot\nabla) \\ &\quad \times \left[f_{i^*}^{(k)} + \frac{1}{\tau}(f_{i^*}^{eq,(k)} - f_{i^*}^{(k)}) + g_i\delta_{m3}\right](t_n, \boldsymbol{x}_{ji}).\end{aligned}$$

where we have set

$$\beta_i(t,\boldsymbol{x}) = 6f_i^*\boldsymbol{c}_i\cdot\boldsymbol{\phi}(t,\boldsymbol{x}), \quad \boldsymbol{x}\in\partial\Omega.$$

Finally, collecting terms of equal order h^m in the bounce back rule, we get

$$\beta_i(t_n, \boldsymbol{x}_{ji})\delta_{m1} + \sum_{k=0}^{m}\Big(-D_{m-k}(\partial_t, q_{ji}\boldsymbol{c}_i\cdot\nabla)f_i^{(k)} + D_{m-k}(0, q_{ji}\boldsymbol{c}_i\cdot\nabla)$$
$$\times\left[f_{i^*}^{(k)} + \frac{1}{\tau}(f_{i^*}^{\mathrm{eq},(k)} - f_{i^*}^{(k)}) + g_i\delta_{m3}\right]\Big)(t_n, \boldsymbol{x}_{ji}) = 0. \tag{22}$$

With (21), the h^0-contribution again leads to the trivial condition $0=0$ because $f_i^{(0)} = f_i^* = f_i^{\mathrm{eq},(0)}$ and $f_{i^*}^* = f_i^*$. In the first order we have $f_i^{(1)} = f_i^{\mathrm{eq},(1)}$, $f_{i^*}^{(1)} = -f_i^{(1)}$ so that (22) amounts to

$$6f_i^*\boldsymbol{c}_i\cdot(\boldsymbol{\phi}(t_n, \boldsymbol{x}_{ji}) - \boldsymbol{u}_1(t_n, \boldsymbol{x}_{ji})) = 0.$$

From this condition, we can conclude $\boldsymbol{u}_1(t, \boldsymbol{x}) = \boldsymbol{\phi}(t, \boldsymbol{x})$ because at points close to a regular boundary point $\boldsymbol{x}\in\partial\Omega$, the incoming directions include d linearly independent vectors. Only at irregular boundary points with sharp corners we may find less than d independent directions. Altogether, $\boldsymbol{u}_1, \rho_2/3$ turn out to be the solution of the full boundary value problem (1), (2).

Proceeding to order h^2, we find

$$6f_i^*\boldsymbol{c}_i\cdot\boldsymbol{u}_2(t_n, \boldsymbol{x}_{ji}) = f_i^*(6q_{ji} - 3)(\boldsymbol{c}_i\cdot\nabla)\boldsymbol{c}_i\cdot\boldsymbol{u}_1(t_n, \boldsymbol{x}_{ji}). \tag{23}$$

Here it is important to notice that for general geometries the values q_{ji} *cannot* be written as smooth functions of the points $\boldsymbol{x}_{ji}$ because the difference between the values q_{ji} corresponding to neighboring nodes $\boldsymbol{x}_{ji}$ is generally of order one while the distance between the nodes is of order h. Consequently, (23) cannot be satisfied by any *smooth* function $\boldsymbol{u}_2$. According to our observation at the beginning of Section 3, this tells us that the lattice Boltzmann solution $\hat{f}_i$ generally exhibits irregular behavior at the order h^2 because $f_i^{(2)}$ cannot be determined as a smooth function. We thus face an expansion of the form

$$\hat{f}_i(n, \boldsymbol{j}) = f_i^{(0)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + hf_i^{(1)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h^2\hat{\delta}_i(n, \boldsymbol{j}) \tag{24}$$

with a grid function $\hat{\delta}_i(n, \boldsymbol{j})$ in second order. For the averages of $\hat{f}_i$ this implies

$$\begin{aligned}\hat{\rho}(n, \boldsymbol{j}) &= 1 + h^2\sum_i \delta_i(n, \boldsymbol{j})\\ \hat{\boldsymbol{u}}(n, \boldsymbol{j}) &= h\boldsymbol{u}_1(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h^2\sum_i \delta_i(n, \boldsymbol{j})\boldsymbol{c}_i\end{aligned} \tag{25}$$

Since $\delta_i(n, \boldsymbol{j})$ generally has non-zero averages (examples are given in Section 3.4), we conclude that the Navier–Stokes velocity $\boldsymbol{u}_1$ can only be extracted from $\hat{\boldsymbol{u}}$ with first order accuracy and that the pressure is no longer recoverable from $\hat{\rho}$ (inconsistent pressure).

While this describes the general behavior of the bounce back algorithm, the situation may be better in the case of specific geometries where $q_{\boldsymbol{j}i}$ is *constant* (and thus smooth) along connected components of the boundary $\partial\Omega$. For example, in the particular case $q_{\boldsymbol{j}i} = 1/2$ where the boundary is located half a link distance away from the boundary nodes, condition (23) leads to $\boldsymbol{u}_2(t, \boldsymbol{x}) = 0$ at the boundary. As explained above, this implies that the moments $\boldsymbol{u}_2, \rho_3$ of the *regular* part of the expansion vanish. However, by carrying out the expansion to order three, we find that the third order coefficient cannot be described by a smooth function which indicates irregular behavior (the situation is similar to the one explained for general constant $q_{\boldsymbol{j}i}$ below). Since the irregular third order contribution generally has a non-zero average, the pressure will be only first order accurate. More specifically, the expansion of $\hat{f}_i$ has the form

$$\hat{f}_i(n, \boldsymbol{j}) = f_i^{(0)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h f_i^{(1)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h^2 f_i^{(2)}(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h^3 \hat{\eta}_i(n, \boldsymbol{j}).$$

where now irregular behavior is found one order later (this can be checked analytically by carrying out the expansion to order three). Computing the averages, we thus have

$$\begin{aligned}(\hat{\rho}(n, \boldsymbol{j}) - 1)/h^2 &= \rho_2(t_n, \boldsymbol{x}_j) + h \sum_i \eta_i(n, \boldsymbol{j}) \\ \hat{\boldsymbol{u}}(n, \boldsymbol{j})/h &= \boldsymbol{u}_1(t_n, \boldsymbol{x}_{\boldsymbol{j}}) + h^2 \sum_i \eta_i(n, \boldsymbol{j}) \boldsymbol{c}_i\end{aligned}$$

i.e. pressure can be recovered with first order and velocity with second order accuracy in the case $q_{\boldsymbol{j}i} = 1/2$.

However, for other constant values of $q_{\boldsymbol{j}i}$, the situation turns out to be quite different. Despite the fact that the right hand side of (23) is now a smooth function, (23) can typically *not* be satisfied by any smooth $\boldsymbol{u}_2$. This is due to the fact that $\boldsymbol{c}_i \cdot \boldsymbol{u}_2$ is linear in $\boldsymbol{c}_i$ while $(\boldsymbol{c}_i \cdot \nabla)\boldsymbol{c}_i \cdot \boldsymbol{u}_1$ is quadratic which, in general, leads to conflicts if equality is required for a linearly *dependent* set of vectors $\boldsymbol{c}_i$. To give a specific example, we consider a stationary linear flow in a half space $\Omega = (-\infty, 0) \times \mathbb{R}$

$$\boldsymbol{u}(\boldsymbol{x}) = A\boldsymbol{x}, \qquad p(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{x}^{\mathsf{T}} A^2 \boldsymbol{x}, \qquad A = \begin{pmatrix} 4 & 1 \\ 1 & -4 \end{pmatrix}. \tag{26}$$

Using the D2Q9 model, the incoming directions at the boundary $\partial\Omega$ are $(-1,0)^{\mathsf{T}}$, $(-1,1)^{\mathsf{T}}$, and $(-1,-1)^{\mathsf{T}}$. If we assume, for example, $q_{ji}=0$ then the following three conditions on $\boldsymbol{u}_2=(u_2^x,u_2^y)^{\mathsf{T}}$ follow from (23)

$$u_2^x=-2, \qquad u_2^x+u_2^y=-1, \qquad u_2^x-u_2^y=1. \tag{27}$$

However, by adding the second and the third condition, we find $u_2^x=0$ which obviously contradicts the first condition. Consequently, we cannot construct a regular coefficient $f_i^{(2)}$ and the expansion has the form (24) which implies inconsistent pressure and first order accurate velocity.

We conclude with the remark that within the special case of geometries having constant q_{ji} there are some rare cases in which the linear and quadratic $\boldsymbol{c}_i$-dependence in (23) does *not* lead to incompatibilities. One such very special situation is the famous Poiseuille flow in an axis parallel channel which therefore is a rather inadequate test case for the general behavior of boundary algorithms.

In the following section, numerical examples are used to illustrate the theoretically predicted behavior of the lattice Boltzmann algorithm with bounce back rule, i.e. first order accurate velocity and an inconsistent pressure fields for general flows.

3.4. Numerical Tests

We apply the lattice Boltzmann method with bounce back rule to several boundary value problems for which exact solutions are known. The first problem is the stationary linear problem (26) described in the last section which we now restrict to the unit square. The second problem is also a stationary linear problem on the unit square

$$\boldsymbol{u}(\boldsymbol{x})=B\boldsymbol{x}, \qquad p(\boldsymbol{x})=-\frac{1}{2}\boldsymbol{x}^{\mathsf{T}}B^2\boldsymbol{x}, \qquad B=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}. \tag{28}$$

The third problem is the decaying Taylor vortex flow described in ref. 18 and restricted to the unit square with parameters $A=\frac{\pi}{2}$ and $B=\frac{\pi}{2}$. The last two problems are taken from ref. 19. One is the Poiseuille flow driven by the body force in an inclined channel, another is the circular flow in a disk with radius $R=1$. Their exact solutions are given in ref. 19, but the slope of the channel used here is 3/10. The pressure of the circular flow is given by integrated Bessel functions. We calculate it using a second order accurate numerical integration.

For the circular flow, we initialize the velocity to the exact value at time $t=0.5$. For all other test problems, the initial velocity is the exact

value at time $t=0$. The termination time is $T=1$. Boundary conditions are specified by evaluating the exact velocity on the boundary $\partial\Omega$.

The numerical tests are carried out on a sequence of grids with grid size $h \in \{\frac{1}{10}, \frac{1}{20}, \frac{1}{30}, \frac{1}{40}, \frac{1}{50}\}$. The viscosity parameter is fixed as $\nu = 0.1$. The error between exact and numerical values is measured in terms of

$$M_\infty = \max_{0 \leqslant t_n \leqslant T} \max_{\boldsymbol{x}_j \in \Omega} \|\hat{\boldsymbol{M}}(n, \boldsymbol{j}) - \boldsymbol{M}(t_n, \boldsymbol{x}_j)\|_\infty,$$

where, $\hat{\boldsymbol{M}}$ and $\boldsymbol{M}$ represent the numerical and exact values, respectively ($\boldsymbol{M}$ is either pressure or velocity with corresponding $\hat{\boldsymbol{M}}$ given by $(\hat{\rho}-1)/(3h^2)$ and $\hat{\boldsymbol{u}}/h$).

For the flows in the unit square the boundary nodes are located exactly on the boundary, i.e. $q_{ji}=0$ for all boundary nodes. Of course, one would normally take $q_{ji}=1/2$ in this geometry because the bounce back rule performs better in that case as we have seen in the previous section. However, we are interested in the behavior of the bounce back rule for general geometries where it is impossible to enforce $q_{ji}=1/2$ at all nodes. In this sense, $q_{ji}=0$ in connection with the unit square serves as a simple model for more complicated geometries. Clearly, for the other geometries (the circular flow and inclined Poiseuille flow) the boundary nodes inevitably exhibit many different $q_{ji} \in [0, 1)$ no matter how the grids are laid out.

Figure 2 shows the logarithmic error of velocity against the logarithmic grid size, which decreases while the grid becomes finer. The least

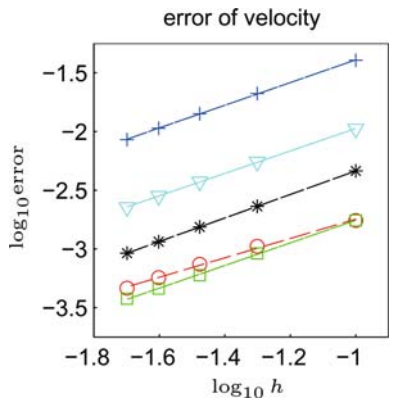

Fig. 2. Logarithmic error of velocity versus $\log_{10} h$ for different test problems using bounce back rule: ($*$) linear flow (26), ($\circ$) linear flow (28), ($+$) Taylor vortex, ($\square$) circular flow and ($\triangledown$) Poiseuille flow. The error is of order h (color online).

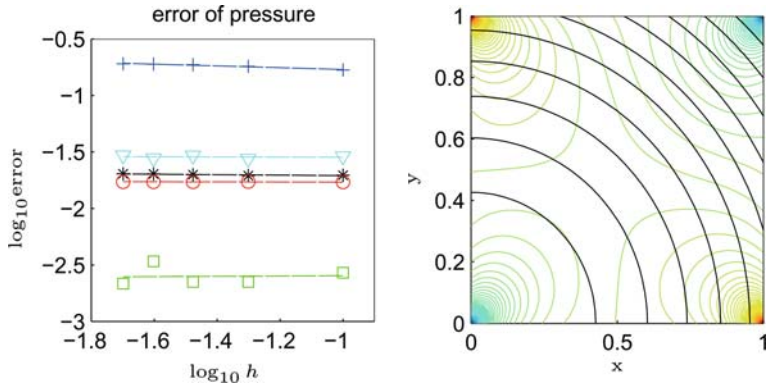

Fig. 3. Left: logarithmic pressure error vs. $\log_{10} h$ for various test problems show that the bounce back rule leads to an inconsistent pressure. Right: pressure contour lines compared with exact solution (black) for the flow (28) (color online).

squares slopes have values around 1, which demonstrates that the bounce back rule can bring out first order accurate velocities in general.

A similar plot for pressure is given in Fig. 3 but here the error increases with decreasing grid size, and the slopes are around zero or even negative, which means the pressure is zero order accurate. The tremendous difference between exact pressure and lattice Boltzmann approximation for the test case (28) is presented in the right plot of Fig. 3. In fact, what we see is essentially the error term $\sum_i \hat{\delta}_i(n, \boldsymbol{j})$ in (25) which is irregular in the corners.

3.5. Modifications of the Basic Algorithm

Since the accuracy of the lattice Boltzmann scheme with bounce back rule is, in general, restricted to first order for velocity and zeroth order for pressure, there is a need to improve the scheme. This is achieved by enforcing $\boldsymbol{u}_2 = 0$ on the boundary. In view of (23) we modify the bounce back rule (8) according to

$$\hat{f}_i(n+1, \boldsymbol{j}) = \hat{f}_i^b(n, \boldsymbol{j}) - h^2 f_i^*(6q_{ji} - 3)(\boldsymbol{c}_i \cdot \nabla)(\boldsymbol{c}_i \cdot \boldsymbol{u}_1(t_n, \boldsymbol{x}_{ji})) \qquad (29)$$

so that the correction term removes the unwanted right hand side of (23). Note, however, that the rule cannot be implemented in that form because $(\boldsymbol{c}_i \cdot \nabla)\boldsymbol{c}_i \cdot \boldsymbol{u}_1$ involves normal derivatives of the Navier–Stokes solution $\boldsymbol{u}_1$ and these derivatives cannot explicitly be written in terms of the boundary data $\boldsymbol{\phi}$. We first have to replace the space derivatives by a suitable

discretization. This discretization is by far not unique so that additional constraints are important in the design like stability and simplicity. Below we present several algorithms which realize approximations of the required derivative terms. The asymptotic analysis of these schemes always follows the same pattern. We have to insert the expansion (11) into the condition, use the structure (21) and check whether $\boldsymbol{u}_2 = 0$ at the boundary. If we write the conditions as additive corrections to the bounce back scheme, we can use the results of Section 3.3 and just analyze the additive correction. For one of the methods, we give an example of this analysis in the appendix.

3.5.1. Finite Difference Approach (FD)

A straightforward idea is to approximate the derivatives in $(\boldsymbol{c}_i \cdot \nabla)\boldsymbol{c}_i \cdot \boldsymbol{u}_1$ by means of finite differences (this has also been used in ref. 18). One choice is

$$\boldsymbol{c}_i \cdot \nabla \boldsymbol{c}_i \cdot \boldsymbol{u}_1(t_n, \boldsymbol{x}_{ji}) \approx \hat{\varphi}_i(n, \boldsymbol{j}) = \begin{cases} \dfrac{\left[\hat{\boldsymbol{u}}(n, \boldsymbol{j} + \boldsymbol{c}_i) - \boldsymbol{\phi}(t_n, \boldsymbol{x}_{ji})\right] \cdot \boldsymbol{c}_i}{(1 + q_{ji})h} & q_{ji} \leqslant \frac{1}{2} \\ \dfrac{\left[\hat{\boldsymbol{u}}(n, \boldsymbol{x}_j) - \boldsymbol{\phi}(t_n, \boldsymbol{x}_{ji})\right] \cdot \boldsymbol{c}_i}{q_{ji}h} & q_{ji} > \frac{1}{2} \end{cases}$$

where the reason to use two different expressions depending on $q_{ji} \leqslant 1/2$ or $q_{ji} > 1/2$ is for the sake of stability. The corresponding boundary algorithm is

$$\hat{f}_i(n+1, \boldsymbol{j}) = \hat{f}_i^b(n, \boldsymbol{j}) - h^2 f_i^*(6q_{ji} - 3)\hat{\varphi}_i(n, \boldsymbol{j})$$

Note that, in general geometries, extra considerations are required if for some boundary node a neighbor is missing in the incoming direction, or in other words, if there exist two opposing incoming directions at the same node. This happens, for example, at corners in 2D and 3D, at edges in 3D, but also along smooth boundaries at nodes where incoming links are almost tangential to the boundary. In such situations the algorithm above is not applicable if $q_{ji} \leqslant 1/2$ so that the corresponding directional derivatives of $\boldsymbol{u}_1$ are missing. However, the derivatives can be computed from the boundary values in these cases by using appropriate finite differences along the boundary. In our test case on the square geometry, for example, we can get the first derivatives of $\boldsymbol{u}_1$ in the corners by taking one-sided derivatives of the boundary values along the edges of the square.

To analyze the proposed FD scheme, we write $\hat{\boldsymbol{u}}$ in the formula for $\hat{\varphi}_i$ as $\sum_i \hat{f}_i \boldsymbol{c}_i$ and insert the expansion. In leading order, we recover the required derivative term and in connection with (23) which takes care of the analysis of the bounce back part $\hat{f}_i^b$, we conclude that $\boldsymbol{u}_2 = 0$ at the boundary. Since $\boldsymbol{u}_2 = 0$ is also compatible with the corner treatment, we conclude that FD leads to first order accurate pressure and second order accurate velocity fields.

We remark that *all* non-local link-based boundary schemes face the same problem as FD at boundary nodes with opposing incoming directions although the problem is rarely addressed in literature. In the examples BFL, MLS below, we use the FD approach for the corners described here (note that simply applying the bounce back rule for the two opposing incoming directions in the corners inevitably reduces the accuracy of the whole method).

3.5.2. Bouzidi's Rule (BFL)

This boundary algorithm has been numerically demonstrated to give second order accurate velocity in ref. 19. Therefore, it is not surprising that BFL can be viewed as a corrected bounce back rule with a particular discretization of the required derivative $(\boldsymbol{c}_i \cdot \nabla)\boldsymbol{c}_i \cdot \boldsymbol{u}_1$. To see this, we write BFL in the form

$$\hat{f}_i(n+1, \boldsymbol{j}) = \hat{f}_{i^*}^b(n, \boldsymbol{j}) + \Delta_i^{\pm}(n, \boldsymbol{j})$$

with

$$\begin{aligned} \Delta_i^-(n, \boldsymbol{j}) &= (1-2q_{\boldsymbol{j}i})\big[\hat{f}_{i^*}^c(n, \boldsymbol{j}+\boldsymbol{c}_i) - \hat{f}_{i^*}^c(n, \boldsymbol{j})\big], \quad q_{\boldsymbol{j}i} \leqslant 1/2, \\ \Delta_i^+(n, \boldsymbol{j}) &= (1-2q_{\boldsymbol{j}i})\big[\hat{f}_i(n+1, \boldsymbol{j}) - \hat{f}_i^c(n, \boldsymbol{j})\big], \quad q_{\boldsymbol{j}i} > 1/2. \end{aligned}$$

With asymptotic analysis we find that the leading order of $\Delta_i^{\pm}(n, \boldsymbol{j})$ is precisely the required term $3f_i^* \boldsymbol{c}_i \cdot \nabla \boldsymbol{c}_i \cdot \boldsymbol{u}_1$. We can thus conclude that the BFL rule gives rise to a second order accurate velocity and a first order accurate pressure (if the nodes with opposing incoming directions are properly treated – see comments for FD).

3.5.3. The Boundary-Fitting Method (FH) and its Improvement (MLS)

The so-called boundary-fitting method (FH) proposed in ref. 22, is based on a linear combination of $\hat{f}_{i^*}^c$ and $f_{i^*}^{\mathrm{eq}}$ at the boundary node without reference to neighboring nodes. As outlined above, we write it as a correction of the bounce back rule which gives rise to

$$\hat{f}_i(n+1, \boldsymbol{j}) = \hat{f}^b_{i^*}(n, \boldsymbol{j}) + \hat{\theta}_i(n, \boldsymbol{j})$$

with

$$\hat{\theta}_i(n, \boldsymbol{j}) = -\chi_{ji}(\hat{f}^c_{i^*}(n, \boldsymbol{j}) - \hat{f}^{\mathrm{eq}}_{i^*}(n, \boldsymbol{j}) + 3 f^*_i \bar{\boldsymbol{v}}_i(n, \boldsymbol{j}) \cdot \boldsymbol{c}_i),$$

where $\bar{\boldsymbol{v}}_i(n, \boldsymbol{j}) = (h\phi(t_n, \boldsymbol{x}_{ji}) - \hat{\boldsymbol{u}}(n, \boldsymbol{j}))/q_{ji}$ for $q_{ji} \geqslant 1/2$ and zero for $q_{ji} < 1/2$. This method has an intrinsic disadvantage because the parameter χ_{ji}

$$\chi_{ji} = \begin{cases} (2q_{ji} - 1)/\tau, & q_{ji} \geqslant \frac{1}{2} \\ (2q_{ji} - 1)/(\tau - 1), & q_{ji} < \frac{1}{2} \end{cases}$$

depends on $1/(\tau - 1)$ which eventually leads to instability when $\tau \approx 1$.

In ref. 15 an improvement (MLS) of FH is achieved by using the next neighbor along the link to calculate $\bar{\boldsymbol{v}}_i(n, \boldsymbol{j}) = \hat{\boldsymbol{u}}(n, \boldsymbol{j} + \boldsymbol{c}_i) - \hat{\boldsymbol{u}}(n, \boldsymbol{j})$ for $q_{ji} < 1/2$. Then χ_{ji} becomes a function of $1/(\tau - 2)$ which enlarges the region of stability but does not overcome the inherent drawback of FH. Moreover, this modification is not defined for opposing incoming directions.

To motivate the boundary conditions FH and MLS, a Chapman–Enskog analysis is used in ref. 15. However, apart from the structural assumptions related to the Chapman–Enskog approach, it requires the additional assumption that the intrinsic time scale of the unsteady flow must be large compared with the advection time on the lattice scale. In our analysis, this assumption is built in from the beginning and we can show the accuracy order without any extra assumption (see Appendix B). An interesting behavior of the method FH is observed in the case when opposing incoming directions are present in the case $q_{ji} = 0$. Then, the accuracy reduces from second to first order for velocity and from first order to inconsistency for pressure. This behavior can be explained with the analysis presented in Appendix B. A simple fix of FH at such nodes is given by applying the method POP_0 (see below) to the opposing incoming directions. With this choice the method FH is still a local method. In our numerical tests, we used this slight modification of FH.

3.5.4. Link-Averaged One Point Approach (POP_θ)

This discretization uses the fact that the coefficient $f^{(2)}_i$ in (21) carries information about the required derivative. In fact

$$\boldsymbol{c}_i \cdot \nabla f^{(1)}_i = 3 f^*_i \boldsymbol{c}_i \cdot \nabla \boldsymbol{c}_i \cdot \boldsymbol{u}_1$$

and from our expansion, we can see that $\boldsymbol{c}_i \cdot \nabla f_i^{(1)}$ can approximately be recovered from the numerical values

$$-\frac{1}{h^2}(\hat{f}_i - \hat{f}_i^c + \hat{g}_i) = \boldsymbol{c}_i \cdot \nabla f_i^{(1)} + \mathcal{O}(h). \tag{30}$$

Instead of using this relation directly, we combine it with an averaging step over all links, which leads to

$$h^2 f_i^*(3-6q_{\boldsymbol{j}i})(\boldsymbol{c}_i \cdot \nabla)\boldsymbol{c}_i \cdot \boldsymbol{u}_1 = -\sum_k K_{ik}(\hat{f}_k - \hat{f}_k^c + \hat{g}_k) + \mathcal{O}(h^3), \tag{31}$$

where the coefficients K_{ik} are defined as

$$K_{ik} = -\frac{3}{2}(3-6q_{\boldsymbol{j}i}) f_i^* \big((\boldsymbol{c}_i \cdot \boldsymbol{c}_k)^2 - |\boldsymbol{c}_i|^2/3 - c_{i\alpha}^2(|\boldsymbol{c}_k|^2 - 1)\big).$$

Here $\alpha \in \{1, \ldots, d\}$ is any index (we take $\alpha = d$). For reasons of stability, it is favorable to evaluate the right hand side of (31) in a semi-implicit form. Eventually, we arrive at the following one-point algorithm at the boundary node $\boldsymbol{x}_{\boldsymbol{j}}$ for which the incoming directions are $i \in V_{\boldsymbol{j}}$:

- compute K_{ik} for all pairs of velocity indices
- select $\theta \in [0, 1]$
- compute $L_{ik} = \delta_{ik} + \theta K_{ik}$ for indices $i, k \in V_{\boldsymbol{j}}$ of the incoming directions
- determine the inverse of L_{ik}
- evaluate $\sigma_k(n, \boldsymbol{j}) = \hat{f}_k^c(n, \boldsymbol{j}) - \hat{g}_k(n, \boldsymbol{j}) - (1-\theta)\hat{f}_k(n, \boldsymbol{j})$ for all directions k
- noting that $\hat{f}_k(n+1, \boldsymbol{j})$ is available for non-incoming directions $k \notin V_{\boldsymbol{j}}$ after the transport step, compute for all $i \in V_{\boldsymbol{j}}$

$$r_i(n, \boldsymbol{j}) = f_{i^*}^b(n, \boldsymbol{j}) - \sum_{k \notin V_j} \theta K_{ik} \hat{f}_k(n+1, \boldsymbol{j}) - \sum_k K_{ik} \hat{\sigma}_k(n, \boldsymbol{j})$$

- determine the required incoming populations $\hat{f}_k(n+1, \boldsymbol{j})$ by solving the linear system (here the inverse of the small matrix L_{ik} is needed)

$$\sum_{k \in V_j} L_{ik} \hat{f}_k(n+1, \boldsymbol{j}) = r_i(n, \boldsymbol{j}), \quad i \in V_j$$

Similar to the original bounce back rule, this algorithm is completely local (no neighbor node needs to be accessed for the evaluation). However, it is not link-based, i.e. the incoming population in direction $\boldsymbol{c}_i$ is computed not only by using information concerning directions $\boldsymbol{c}_i$ and $\boldsymbol{c}_{i*}$, because averaging over all velocities is involved in (31). In fact, this averaging has a stabilizing effect.

Before running the scheme, the inverse of L_{ik} has to be assembled (if $\theta > 0$) for each boundary node (note that the first four steps of the algorithm can be done in a pre-processing step if the boundary is non-moving). Invertibility of the matrix L_{ik} can always be guaranteed with a suitable choice of $\theta \in [0, 1]$. In fact, the parameter θ controls the location of the eigenvalues of θK_{ik} and up to a finite number of choices the spectrum will *not* contain -1. Consequently, $L_{ik} = \delta_{ik} + \theta K_{ik}$ is invertible for all but finitely many choices of θ.

3.5.5. Multi-Reflection Method (MR)

In contrast to the algorithms above, the multi-reflection method MR presented in ref. 14 is not a modification of the bounce back rule, i.e. it does not reduce to bounce back if $q_{ji} = 1/2$ for all nodes. In general, it uses three nodes (i.e. two neighbors) along the incoming directions

$$\begin{aligned}\hat{f}_i(n+1, \boldsymbol{j}) = {} & \kappa_1 \hat{f}^c_{i*}(n, \boldsymbol{j}) + \kappa_0 \hat{f}^c_{i*}(n, \boldsymbol{j} + \boldsymbol{c}_i) + \kappa_{-1} \hat{f}^c_{i*}(n, \boldsymbol{j} + 2\boldsymbol{c}_i) + \bar{\kappa}_{-1} \hat{f}^c_i(n, \boldsymbol{j}) \\ & + \bar{\kappa}_{-2} \hat{f}^c_i(n, \boldsymbol{j} + \boldsymbol{c}_i) + 3 w_i f_i^* h \boldsymbol{\phi}(t_n, \boldsymbol{x}_{ji}) \cdot \boldsymbol{c}_i + 3 f_i^* F_i^{p.c.}(n, \boldsymbol{j}).\end{aligned}$$

In ref. 14, two sets (MR1 and MR2) of parameters κ_1, κ_0, κ_{-1}, $\bar{\kappa}_{-1}$, $\bar{\kappa}_{-2}$, w_i and $F_i^{p.c.}$ are given.

At boundary nodes where only one neighbor is available (i.e. the neighbor $\boldsymbol{j} + 2\boldsymbol{c}_i$ is missing), a modification is suggested in ref. 14 to replace $f^c_{i*}(n, \boldsymbol{j} + 2\boldsymbol{c}_i)$ by $f^c_{i*}(n, \boldsymbol{j} + \boldsymbol{c}_i)$. However, in the case of opposing incoming directions when both neighbors are not available, an algorithm is only given for the case $q_{ji} = 1/2$ when bounce back can be used. In the case of more general q_{ji} we have extended the proposed treatment for one missing neighbor to the case of two missing neighbors, i.e. we replace

$f_{i*}^c(n, \boldsymbol{j}+\boldsymbol{c}_i)$ with $f_{i*}^c(n, \boldsymbol{j})$. Another possibility is to use the FD approach which works well in connection with FD, FH, MLS, and BFL.

According to our asymptotic analysis, when each boundary node has two neighbors along the link, the method MR yields $\boldsymbol{u}_2 = 0$ at the boundary so that second order velocity and first order accurate pressure fields are certainly obtained (if the case of missing neighbors is properly treated). Expanding to higher orders, we even find that MR implies smooth boundary values for $\boldsymbol{u}_3$. In principle, this guarantees a smooth coefficient $f^{(3)}$ so that irregular behavior would only appear at fourth order. This explains the terminology third order *kinetic* accuracy used in ref. 14 from the point of view of the asymptotic expansion approach. However, smoothness of $f^{(3)}$ also requires proper initialization up to third order terms which is difficult for general initial value problems. In fact, the initial value for $\boldsymbol{u}_3$ depends on the initial time derivative of the pressure (see (20)) which is generally not known and has to be determined by solving additional Poisson equations. Secondly, the treatment of the boundary nodes having less than two neighbors may also destroy the smoothness of $f^{(3)}$ so that the pressure is only first order accurate.

If these problems do not appear, like in the case of Poiseuille flow where always two neighbors are available and where the initialization of $\boldsymbol{u}_3$ is easy, the method MR yields indeed second order accuracy for both velocity and pressure (or even the exact pressure which is constant in the case of Poiseuille flow). We could reproduce this result with our code. Also for the stationary linear problem (26) on the unit square, MR1 yields the exact solution if the populations are initialized correctly up to fourth order and if the incoming populations at the corners and the adjacent nodes are prescribed exactly up to order four. If the initial and corner populations are only exact up to third order, we recover both second order pressure and velocity. However, if we do not use the exact populations as boundary values (they are only available in simple cases) but any of the other corner treatments described above, we find unstable solutions (this may be partly due to the fact that we work with the BGK collision operator and not the multiple relaxation time approach). Only the combination with the bounce back rule in the case $q_{ji} = 1/2$ worked satisfactorily.

3.6. Numerical Tests

In the following, we use the decaying Taylor vortex solution with the same parameters as in Section 3.4 to test and compare the boundary schemes. Since the method MR does not yield stable solutions with the available corner treatments for general values q_{ji}, we split the comparison. First we consider those schemes which are modifications of the

bounce back rule for $q=0$ and $q=0.1$ (where q is the common value of q_{ji} at lower and left boundary of the square). In the case $q=1/2$, all these schemes reduce to the bounce back rule and a comparison with MR is now possible because, for $q_{ji}=1/2$, MR works in combination with the bounce back rule at the corners and gives rise to a first order accurate pressure and a second order accurate velocity.

The numerical convergence rates for pressure and velocity are summarized in Table I and the corresponding error plots can be found in Fig. 4.

Tables II and III give an impression on the stability of the different boundary schemes. Here, we compare the maximal error in velocity for a computation on a grid with $h=1/50$ and viscosities ranging from $1/100$ to 10. The letter N is used if no error value could be obtained because of instability.

In the general case $q>0$ (Table IV is a representative case), we can see that FH, POP_1 and BFL have the best stability among the modifications of the bounce back rule where FH is very inaccurate or suffers from instability if $\tau \approx 1$. Only when $q=0$, BFL is slightly more stable than POP_1 which in turn has a slightly smaller error. However, on rectangular domains, the best choice of q with respect to stability is certainly $q=1/2$

Table I. Convergence Rates for the Taylor Vortex Flow from Section 3.4 Computed with Different Boundary Algorithms

	FD	POP_0	POP_1	$POP_{0.7}$	BFL	FH
Pressure	1.011	1.138	0.999	1.172	1.088	1.288
Velocity	1.979	2.008	1.992	2.023	1.975	1.923

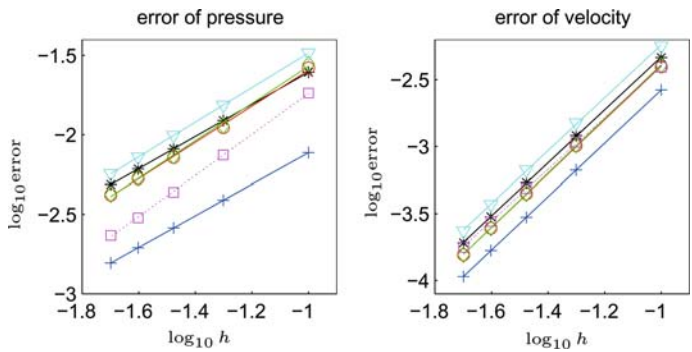

Fig. 4. Double logarithmic error plots of pressure (left) and velocity (right) versus grid size. The boundary schemes are FD (∗), POP_0 (○), POP_1(+), $POP_{0.7}$(◇), BFL(▽), and FH(□) (color online).

Table II. Maximal Velocity Error on a Fixed Grid ($h = 1/50$) with Nodes on the Boundary ($q = 0$) for Varying Viscosity

Viscosity	1/100	1/50	1/10	1/6	1	10
FD	*N*	*N*	0.00019	0.00028	0.00195	*N*
FH	0.00014	0.00015	0.00023	*N*	0.00335	0.15157
POP_0	*N*	0.00006	0.00016	0.00027	0.00278	0.07660
$POP_{0.7}$	*N*	0.00007	0.00015	0.00027	0.00278	0.07651
POP_1	*N*	0.00008	0.00011	0.00027	0.00278	0.07675
BFL	0.00014	0.00015	0.00023	0.00038	0.00272	0.07516

The letter *N* indicates instability.

Table III. Maximal Velocity Error on a Fixed Grid ($h = 1/50$) with Nodes not on the Boundary ($q = 0.1$) for Varying Viscosity

Viscosity	1/100	1/50	1/10	1/6	1	10
FD	*N*	0.00014	0.00019	0.00031	0.00379	*N*
FH	0.00014	0.00015	0.00023	*N*	0.00335	0.15157
POP_0	*N*	*N*	*N*	0.00025	0.00235	0.07480
$POP_{0.7}$	*N*	*N*	0.00015	0.00025	0.00235	0.07510
POP_1	0.00003	0.00003	0.00015	0.00025	0.00235	0.07516
BFL	0.00011	0.00010	0.00021	0.00036	0.00232	0.07425

The letter *N* indicates instability.

Table IV. Maximal Velocity Error on a Fixed Grid ($h = 1/50$) with Nodes Half Way from the Boundary ($q = 1/2$) for Varying Viscosity

Viscosity	0.01	0.02	0.03	0.1	1	10
BB	0.00009	0.00007	0.00006	0.00010	0.00252	0.07457
MR	0.00002	0.00003	0.00004	0.00014	*N*	*N*

The letter *N* indicates instability.

in which case all the considered methods turn into the simple bounce back rule. A comparison with MR for this case is given in Table IV. We have also compared the new method POP_θ with BFL for the circular flow problem. In this case, the geometry coefficients q_{ji} range in the whole interval $[0, 1)$. On two grids ($h = 1/50, 1/100$) with termination time $T = 1$ we have varied the viscosity. Down to $\nu = 0.0001$ both schemes are stable in the sense that they produce bounded solutions.

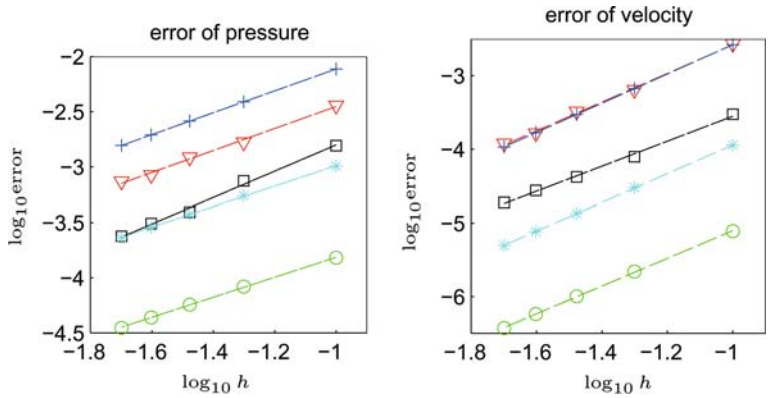

Fig. 5. Double logarithmic error plots of pressure (left) and velocity (right) versus grid size using POP_0: (∗) linear flow (26), (○) linear flow (28), (+) Taylor vortex, (□) circular flow and (▽) Poiseuille flow (color online).

Finally, we have applied the proposed method POP_1 to the set of problems introduced in Section 3.4. While the bounce back algorithm has only low accuracy for these test cases (see Fig. 3), our algorithm shows the predicted second order accurate velocity and first order accurate pressure (see Fig. 5).

4. CONCLUSION

In this article, we have shown that the asymptotic analysis which has been successfully used in the consistency analysis of numerical methods for differential equations[1] can also be applied to lattice Boltzmann schemes. The analysis gives rise to analytic details about the behavior of the numerical solution which allows to access the accuracy of the lattice Boltzmann moments as approximations to the Navier–Stokes fields.

As we have demonstrated, the analysis of conditions apart from the evolution equations (like initial conditions, boundary conditions, but also coupling conditions, etc.) is straightforward. It is clear that the analysis presented here is more general than the one in ref. 20 where different boundary algorithms are investigated for a particular class of flows (Poiseuille flows). We only require smooth solutions of the underlying Navier–Stokes problem but even if the solution exhibits singularities one can analyze the method in a similar way using appropriate tools from asymptotic analysis.

In refs. 14, 21, the analysis is based on the Chapman–Enskog (CE) expansion which establishes (at some order) a connection between the

lattice Boltzmann method and the weakly *compressible* isothermal Navier–Stokes equation. This naturally leads to different expansion coefficients for the LB variables (for example, the coefficients depend on the grid size and they are not specified in terms of the incompressible Navier–Stokes solution as in our expansion) but apart from that the formal steps in the analysis of the boundary conditions and in the construction of new algorithms are very similar: the truncated expansion is inserted into the boundary scheme, Taylor expansions around boundary points are performed and the leading order term is analyzed to derive schemes of higher accuracy. However, the work[14,21] is restricted to stationary flows.

We have also constructed a new local boundary condition (POP_θ) based on the results of our analysis. The numerical tests indicate that it may be an interesting alternative to existing methods.

The same methodology that has been used to analyze and construct boundary conditions which are consistent to Dirichlet conditions on the Navier–Stokes level can also be used to investigate and formulate other conditions like Neumann conditions or normal stress conditions. This will be the subject of future work.

APPENDIX A: DETAILS CONCERNING THE EXPANSION

A.1. Recursively Defined Operators

The polynomials E_k in (17) are defined recursively by $E_0(\tau, \theta, \sigma) = 1$ and

$$E_k(\tau, \theta, \sigma) = \sum_{r=0}^{k-1} (-\tau) D_{k-r}(\theta, \sigma) E_r(\tau, \theta, \sigma), \quad k \geqslant 1,$$

where D_{k-r} are given by (13). The polynomials F_m are simply

$$F_m(\tau, \theta, \sigma) = \tau E_{m-3}(\tau, \theta, \sigma)$$

if we define $E_k = 0$ for $k < 0$. Using these definitions, the equality of (16) and (17) can easily be proved by induction. Explicitly, we have

$$\begin{aligned} E_0(\tau, \theta, \sigma) &= 1, \\ E_1(\tau, \theta, \sigma) &= -\tau\sigma, \end{aligned}$$

$$E_2(\tau,\theta,\sigma) = \tau(\tau-\frac{1}{2})\sigma^2 - \tau\theta,$$
$$E_3(\tau,\theta,\sigma) = -\tau(\tau^2-\tau+\frac{1}{6})\sigma^3 + \tau(2\tau-1)\theta\sigma,$$
$$E_4(\tau,\theta,\sigma) = \tau(\tau^3-\frac{3}{2}\tau^2+\frac{7}{12}\tau-\frac{1}{24})\sigma^4 - \tau(3\tau^2-3\tau+\frac{1}{2})\theta\sigma^2 + \tau(\tau-\frac{1}{2})\theta^2.$$

A.2. Operators in the Moment Equations

The equations for the moments ρ_k, $\boldsymbol{u}_k$ are obtained by taking corresponding Moments of (17). In order to obtain a compact notation, we first rewrite $f_i^{\mathrm{eq},(\mathrm{k})}$ using the symmetric tensor product and the matrix scalar product

$$(\boldsymbol{\alpha}\otimes\boldsymbol{\beta})_{ij} = \frac{1}{2}(\alpha_i\beta_j+\alpha_j\beta_i), \quad A\colon B = \sum_{i,j} A_{ij}B_{ij}$$

leading to

$$f_i^{\mathrm{eq},(k)} = f_i^* \left(\rho_k + 3\boldsymbol{c}_i\cdot\boldsymbol{u}_k + \sum_{r+s=k}(\frac{9}{2}(\boldsymbol{c}_i\otimes\boldsymbol{c}_i-\frac{1}{3}I)\colon\boldsymbol{u}_r\otimes\boldsymbol{u}_s)\right).$$

Inserting this relation into (17) and summing over i after multiplication with 1 resp. $\boldsymbol{c}_i$, we obtain differential equations for the moments which are of the general form

$$\sum_{k=0}^{m-1}\left(a_{m-k}\rho_k + b_{m-k}\cdot\boldsymbol{u}_k + q_{m-k}\colon\sum_{r+s=k}\boldsymbol{u}_r\otimes\boldsymbol{u}_s\right) + \tau b_{m-3}\cdot\boldsymbol{G} = 0,$$

$$\sum_{k=0}^{m-1}\left(A_{m-k}\rho_k + B_{m-k}\boldsymbol{u}_k + Q_{m-k}\sum_{r+s=k}\boldsymbol{u}_r\otimes\boldsymbol{u}_s\right) + \tau B_{m-3}\boldsymbol{G} = 0.$$

In these expressions, the differential operators $a_k(\tau,\partial_t,\nabla)$, $b_k(\tau,\partial_t,\nabla)$, etc. result from weighted averages of the operators $E_k(\tau,\partial_t,\boldsymbol{c}_i\cdot\nabla)$. Specifically, we find the polynomials

$$a_k(\tau,\theta,\boldsymbol{\xi})=\sum_i f_i^* E_k(\tau,\theta,\boldsymbol{c}_i\cdot\boldsymbol{\xi}),\qquad A_k(\tau,\theta,\boldsymbol{\xi})=\sum_i f_i^* E_k(\tau,\theta,\boldsymbol{c}_i\cdot\boldsymbol{\xi})\boldsymbol{c}_i,$$

where θ, $\boldsymbol{\xi}$ have to be replaced by ∂_t, ∇ to obtain the operators. Examples are

$$a_1=0,\quad a_2=\frac{1}{3}\tau(\tau-\frac{1}{2})|\boldsymbol{\xi}|^2-\tau\theta,\quad a_3=0$$

and

$$A_1=-\frac{1}{3}\tau\boldsymbol{\xi},\ A_2=0,\ A_3=\frac{\tau}{3}((-\frac{1}{6}+\tau-\tau^2)|\boldsymbol{\xi}|^2+(2\tau-1)\theta)\boldsymbol{\xi},\ A_4=0.$$

Accordingly, the operators acting on the velocities are given by

$$b_k(\tau,\theta,\boldsymbol{\xi})=\sum_i 3f_i^* E_k(\tau,\theta,\boldsymbol{c}_i\cdot\boldsymbol{\xi})\boldsymbol{c}_i,\quad B_k(\tau,\theta,\boldsymbol{\xi})=\sum_i 3f_i^* E_k(\tau,\theta,\boldsymbol{c}_i\cdot\boldsymbol{\xi})\boldsymbol{c}_i\otimes\boldsymbol{c}_i$$

and relevant examples are

$$b_1=-\tau\boldsymbol{\xi},\quad b_2=0,\quad b_3=\tau((-\frac{1}{6}+\tau-\tau^2)|\boldsymbol{\xi}|^2+(2\tau-1)\theta)\boldsymbol{\xi},\quad b_4=0$$

and

$$B_0=I,\quad B_1=0,\quad B_2=\tau\nu|\boldsymbol{\xi}|^2 I+\tau^2 2\nu\boldsymbol{\xi}\otimes\boldsymbol{\xi}-\tau\theta I,\quad B_3=0.$$

To give an example, we show how B_2 acts on a smooth vector field

$$B_2(\tau,\partial_t,\nabla)\boldsymbol{u}=\tau\nu\Delta\boldsymbol{u}+\tau^2 2\nu\nabla\nabla\cdot\boldsymbol{u}-\tau\partial_t\boldsymbol{u}$$

which are exactly the linear $\boldsymbol{u}$ derivatives in the Navier–Stokes equation (note that $\nabla\nabla\cdot\boldsymbol{u}=0$ for divergence free fields).

Finally, the averaged operator for the quadratic term is a second rank tensor

$$q_k(\tau,\theta,\boldsymbol{\xi})=\sum_i\frac{9}{2}f_i^* E_k(\tau,\theta,\boldsymbol{c}_i\cdot\boldsymbol{\xi})(\boldsymbol{c}_i\otimes\boldsymbol{c}_i-\frac{1}{3}I)$$

and the velocity-weighted average leads to a third order tensor

$$Q_k^{\alpha\beta\gamma}(\tau,\theta,\boldsymbol{\xi})=\sum_i \frac{9}{2} f_i^* E_k(\tau,\theta,\boldsymbol{c}_i\cdot\boldsymbol{\xi})(c_{i\beta}c_{i\gamma}-\frac{1}{3}\delta_{\beta\gamma})c_{i\alpha}$$

where the tensor product $Q_k \boldsymbol{u}_r \otimes \boldsymbol{u}_s$ abbreviates summation over the last two indices. Relevant examples are

$$q_1=0, \quad q_2=\tau(\tau-\frac{1}{2})\boldsymbol{\xi}\otimes\boldsymbol{\xi}, \quad q_3=0$$

and

$$Q_1^{\alpha\beta\gamma}(\tau,\theta,\boldsymbol{\xi})=-\frac{1}{2}\tau(\xi_\beta\delta_{\alpha\gamma}+\xi_\gamma\delta_{\alpha\beta}), \quad Q_2=0.$$

5. APPENDIX B: ASYMPTOTIC ANALYSIS OF THE BOUNDARY FITTING METHOD FH

A detailed description of the method is given in Section 3.5. To analyze the algorithm, we insert the regular expansion with the coefficients (21) into the expression for $\hat{\theta}_i(n,\boldsymbol{j})$ and perform a Taylor expansion around the node $(t_n,\boldsymbol{x}_{ji})$. This gives rise to

$$\begin{aligned}\hat{\theta}_i(n,\boldsymbol{j}) &= -\chi_{ji}\left[\left(1-\frac{1}{\tau}\right)(\hat{f}_{i^*}-f_{i^*}^{\mathrm{eq}})(n,\boldsymbol{j})+3f_i^*\bar{\boldsymbol{v}}_i(n,\boldsymbol{j})\cdot\boldsymbol{c}_i\right]\\ &= -\chi_{ji}\left(1-\frac{1}{\tau}\right)\sum_{p=0}\sum_{m=0} h^{m+p}D_p(0,q_{ji}\boldsymbol{c}_i\cdot\nabla)(f_{i^*}^{(m)}-f_{i^*}^{\mathrm{eq},(\mathrm{m})})(t_n,\boldsymbol{x}_{ji})\\ &\quad -3\chi_{ji}f_i^*(h\boldsymbol{\phi}(t_n,\boldsymbol{x}_{ji}))-\sum_{p=0}\sum_{m=0} h^{m+p}D_p(0,q_{ji}\boldsymbol{c}_i\cdot\nabla)\boldsymbol{u}_m(t_n,\boldsymbol{x}_{ji})\cdot\boldsymbol{c}_i,\end{aligned}$$

and up to third order explicitly

$$\hat{\theta}_i=-3h\chi_{ji}f_i^*/q_{ji}(\boldsymbol{\phi}-\boldsymbol{u}_1)\cdot\boldsymbol{c}_i+h^2(1-2q_{ji})(\boldsymbol{c}_i\cdot\nabla)f_i^{(1)}+\mathcal{O}(h^3)$$

Combined with the expansion (22) of the bounce back part in the FH condition, we get

$$h\left(6-3\frac{\chi_{ji}}{q_{ji}}\right)f_i^*/q_{ji}(\boldsymbol{\phi}-\boldsymbol{u}_1)\cdot\boldsymbol{c}_i-6h^2f_i^*\boldsymbol{u}_2\cdot\boldsymbol{c}_i=\mathcal{O}(h^3)$$

from which we conclude $\boldsymbol{u}_1 = \boldsymbol{\phi}$ and $\boldsymbol{u}_2 = 0$ on $\partial\Omega$.

However, in the case $\boldsymbol{q}_{ji} = 0$ the method FH introduces additional conditions at nodes with two opposing incoming directions which generally cannot be satisfied with a smooth coefficient $f_i^{(2)}$. Thus our regular expansion breaks down at second order which indicates that pressure will be inconsistent and velocity only first order accurate.

To derive the additional condition, we consider opposing incoming directions $\boldsymbol{c}_i$ and $\boldsymbol{c}_{i*} = -\boldsymbol{c}_i$ in the case $q_{ji} = 0$

$$\hat{f}_i(n+1, \boldsymbol{j}) = \hat{f}_{i*}(n, \boldsymbol{j}) + 6 f_i^* h \boldsymbol{\phi} \cdot \boldsymbol{c}_i, \quad \hat{f}_{i*}(n+1, \boldsymbol{j}) = \hat{f}_i(n, \boldsymbol{j}) + 6 f_i^* h \boldsymbol{\phi} \cdot \boldsymbol{c}_{i*},$$

Adding the equations, we obtain

$$\hat{f}_i(n+1, \boldsymbol{j}) + \hat{f}_{i*}(n+1, \boldsymbol{j}) = \hat{f}_i(n, \boldsymbol{j}) + \hat{f}_{i*}(n, \boldsymbol{j})$$

so that the quantity $\hat{f}_i(n, \boldsymbol{j}) + \hat{f}_{i*}(n, \boldsymbol{j}) = C$ is constant in time. Inserting the expansion into this relation and expanding around $(t_n, \boldsymbol{x}_{ji})$, we find in second order

$$\frac{\partial}{\partial t}[\rho_2 + \frac{3}{2}(3(\boldsymbol{u}_1 \cdot \boldsymbol{c})^2 - |\boldsymbol{u}_1|^2) - \tau(\boldsymbol{c}_i \cdot \nabla)(\boldsymbol{u}_1 \cdot \boldsymbol{c}_i)] = 0$$

which is, in general, incompatible with the fact that $\rho_2/3, \boldsymbol{u}_1$ solve the Navier–Stokes problem (1).

ACKNOWLEDGEMENTS

The support by the Deutsche Forschungsgemeinschaft (DFG) through the grant Ju440/1-2 is gratefully acknowledged.

REFERENCES

1. M. Junk and Z. Yang, Asymptotic analysis of finite difference methods, *Appl. Math. Comp.* (in press).
2. S. Jin and Z. Xin, The relaxation schemes for systems of conservation laws in arbitrary space dimensions, *Comm. Pure Appl. Math.* **48**:235–276 (1995).
3. U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J. Rivet, Lattice gas hydrodynamics in two and three dimensions, *Complex Sys.* **1**:649–707 (1987).
4. R. Benzi, S. Succi, and M. Vergassola, The lattice-Boltzmann equation: Theory and applications, *Phys. Rep.* **222**:145–197 (1992).
5. H. Chen, S. Chen, and W. Matthaeus, Recovery of the Navier–Stokes equations using a Lattice-gas Boltzmann method, *Phys. Rev. A* **45**:5339–5342 (1992).
6. S. Chen and G. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* **30**:329–364 (1998).

7. X. He and L.-S. Luo, A priori derivation of the lattice Boltzmann equation, *Phys. Rev*. **E 55**:6333–6336 (1997).
8. X. He and L.-S. Luo, Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, *Phys. Rev*. **E 56**:6811–6817 (1997).
9. Y. Sone, Asymptotic theory of a steady flow of a rarefied gas past bodies for small Knudsen numbers, in Advances in Kinetic Theory and Continuum Mechanics, Proceedings of a Symposium Held in Honour of Henri Cabannes, R. Gatignol and J. Soubbaramayer, eds. (Springer, Berlin, 1990), pp. 19–31.
10. T. Inamuro, M. Yoshino, and F. Ogino, Accuracy of the lattice Boltzmann method for small Knudsen number with finite Reynolds number, *Phys. Fluids* **9**:3535–3542 (1997).
11. M. Junk and A. Klar, Discretizations for the incompressible Navier–Stokes equations based on the lattice-Boltzmann method, *SIAM J. Sci. Comput*. **22**:1–19 (2000).
12. M. Junk, A finite difference interpretation of the lattice Boltzmann method, *Numer. Methods Partial Differ. Eq.* **17**:383–402 (2001).
13. M. Junk, A. Klar, and L.-S. Luo, Asymptotic analysis of the lattice Boltzmann equation, preprint, submitted for publication.
14. I. Ginzbourg and d'Humières,. The multireflection boundary conditions for lattice Boltzmann models, *Phys. Rev*. E **68**:066614 (2003).
15. R. Mei, L.-S. Luo, and W. Shyy, An accurate curved boundary treatment in the lattice Boltzmann method, *J. Comput. Phys*. **155**:307–330 (1999).
16. M. Junk and W.-A. Yong, Rigorous Navier–Stokes limit of the lattice Boltzmann equation, *Asymp. Anal.* **35**:165–184 (2003).
17. R. Mei, L.-S. Luo, and D. d'Humieres, Initializations of lbe simulations, Preprint.
18. P. Skordos, Initial and boundary conditions for the Lattice Boltzmann method, *Phys. Rev*. **E 48**:4823–4842 (1993).
19. M. Bouzidi, M. Firdaouss, and P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, *Phys. Fluids* **13**:3452–3459 (2001).
20. M. Rohde, D. Kandhai, J. J. Derksen, and H. E. A. V. den Akker, Improved bounce-back methods for no-slip walls in lattice-boltzmann schemes: Theory and simulations, *Phys. Rev*. E **67**:066703 (2003).
21. I. Ginzbourg and d'Humières, Local second-order boundary methods for lattice Boltzmann models, *J. Stat. Phys*. **84**:5/6 (1996).
22. O. Filippova and D. Hänel, Grid refinement for lattice-BGK models, *J. Comp. Phys*. **147**:219–228 (1998).