CHALMERS

Chalmers Publication Library

A Lattice Boltzmann Method for the Advection-Diffusion Equation with Neumann Boundary Conditions

This document has been downloaded from Chalmers Publication Library (CPL). It is the author´s version of a work that was accepted for publication in:

Communications in Computational Physics (ISSN: 1815-2406)

Citation for the published paper:
Gebäck, T. ; Heintz, A. (2014) "A Lattice Boltzmann Method for the Advection-Diffusion Equation with Neumann Boundary Conditions". Communications in Computational Physics, vol. 15(2), pp. 487-505.

http://dx.doi.org/10.4208/cicp.161112.230713a

Downloaded from: http://publications.lib.chalmers.se/publication/185208

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted.
The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

Commun. Comput. Phys.
doi: 10.4208/cicp.161112.230713a

Vol. **15**, No. 2, pp. 487-505
February 2014

A Lattice Boltzmann Method for the Advection-Diffusion Equation with Neumann Boundary Conditions

Tobias Gebäck[1],* and Alexei Heintz[1]

[1] *Department of Mathematical Sciences, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden.*

Received 16 November 2012; Accepted (in revised version) 23 July 2013

Communicated by Kazuo Aoki

Available online 27 September 2013

Abstract. In this paper, we study a lattice Boltzmann method for the advection-diffusion equation with Neumann boundary conditions on general boundaries. A novel mass conservative scheme is introduced for implementing such boundary conditions, and is analyzed both theoretically and numerically.

Second order convergence is predicted by the theoretical analysis, and numerical investigations show that the convergence is at or close to the predicted rate. The numerical investigations include time-dependent problems and a steady-state diffusion problem for computation of effective diffusion coefficients.

AMS subject classifications: 76R50, 76M28, 65N75

Key words: Lattice Boltzmann, diffusion, advection-diffusion, Neumann boundary condition.

1 Introduction

The lattice Boltzmann method (LBM) has received much attention for flow simulation since its introduction in the 1990's [2,4,9]. Although much less discussed, it is well known that the LBM can be applied to diffusion and advection-diffusion equations as well, see e.g. [7,13,14,16]. Applications include solute transport in porous media [13], dissolution phenomena [17], dispersion [18,19] and comparisons to NMR experiments [10].

To be precise, the problem we study in this paper is the advection-diffusion equation with isotropic diffusion

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{U})=D\Delta\rho \quad \text{in } \Omega\times[0,T], \tag{1.1a}$$

*Corresponding author. *Email addresses:* `tobias.geback@chalmers.se` (T. Gebäck), `heintz@chalmers.se` (A. Heintz)

©2014 Global-Science Press

$$\rho|_{t=0}=\rho_0 \quad \text{in } \Omega, \tag{1.1b}$$

$$\partial_n\rho=0 \quad \text{on } \partial\Omega\times[0,T], \tag{1.1c}$$

where ρ is the density, $\mathbf{U}$ is a given flow velocity and D is a scalar diffusion coefficient. All the analysis will be valid for this equation, although for the numerical results, we will set $\mathbf{U}=0$, which yields the isotropic diffusion equation with Neumann (zero normal flux) boundary conditions.

When applying the lattice Boltzmann method, macroscopic variables such as ρ in (1.1) are obtained as moments in velocity space of a distribution function f. The evolution of f is described by the lattice Boltzmann equation (LBE)

$$f_i(t+\Delta t,\mathbf{x}+\mathbf{c}_i\Delta t)-f_i(t,\mathbf{x})=J_i(f,f^{(eq)}), \qquad i=0,\cdots,q-1, \tag{1.2}$$

where q is the number of discrete velocities used, $\{\mathbf{c_i}\}_{i=0}^{q-1}$ is the set of velocities, and $J_i(f,f^{(eq)})$, $i=0,\cdots,q-1$ are collision operators describing the relaxation of f towards an equilibrium distribution $f^{(eq)}$. The LBM is applied using a time-splitting approach, first applying a collision step to compute the right hand side of (1.2) and then a streaming step to move the distribution according the velocities $\mathbf{c}_i$, as prescribed by the left hand side of (1.2).

The way boundary conditions in the lattice Boltzmann method are applied is slightly different compared to other numerical methods. Since the LBM simulates the time evolution of a velocity distribution, and macroscopic quantities like density are only obtained as moments of this distribution, boundary conditions have to determine the whole particle distribution, and not only the density or flux that is determined by the macroscopic boundary condition (such as a Dirichlet or Neumann boundary condition). This has to be done in a consistent way in order not to reduce accuracy and/or stability for the solution to the macroscopic equation.

Neumann and other flux boundary conditions for the diffusion equation have not been much studied in the LBM framework, despite the fact that they present certain challenges, as will be seen below. In some cases [11], the bounce-back boundary condition has been applied, which is erroneous as it prescribes not only zero normal flux, but also zero tangential flux. In other cases [13, 17], the Neumann boundary condition has been correctly applied, but only for grid-aligned boundaries, where it is rather easily implemented through specular reflection of the velocity distribution, see [7].

For general boundaries, we have found only two attempts at a proper implementation of flux boundary conditions. Yoshida et al. [18] introduced corrections to the bounce-back rule in order to get a correct total flux through a surface, when the (non-zero) flux was prescribed at the boundary. However, they noted that the tangential flux was incorrect near the surface, as is to be expected when the bounce-back rule is used. The issue of the tangential flux was addressed by Ginzburg [7], and it was noted that it is essential for the flux boundary conditions that there are no restrictions put on the tangential flux, at least to leading orders. The method presented in [7] is appealing, as it consists of only local modifications of the bounce-back rule. However, because of this, it is unclear what the result will be for curved boundaries, especially regarding the exact position of the surface.

We have chosen a different implementation, relying on a layer of ghost points immediately outside the boundary, where the distribution function is specified through macroscopic variables interpolated inside the domain and copied or mirrored to the outside. Although this is a more complicated procedure, involving interpolation, it is in our opinion more flexible with respect to the surface geometry. Another important aspect is the exact conservation of mass, which is implemented through the specification of the second order moment tensor in the ghost nodes.

In this paper, we first present an asymptotic analysis of the lattice Boltzmann advection-diffusion equation in Section 2. In Section 3, we present our implementation of the Neumann boundary condition and prove that it indeed satisfies the Neumann boundary condition to second order. Numerical results can be found in Section 4, and a discussion of the results in Section 5.

2 Asymptotic analysis of the lattice Boltzmann method for the advection-diffusion equation

In this section, we perform an asymptotic analysis in line with the analysis of the solution to the Navier-Stokes equations presented by Junk et al. [12]. It is based on the diffusive scaling $\Delta t \sim (\Delta x)^2$, as this seems to be the appropriate scaling for the diffusion equation (although the Chapman-Enskog analysis presented elsewhere, see e.g. [6,10], also yields correct results). See [12] for a discussion of the differences between the two approaches.

In analogy with [12], if $\mathbb{V} = \{\mathbf{c}_0, \cdots, \mathbf{c}_{q-1}\} \subset \mathbb{R}^d$ is the set of d-dimensional discrete velocities, we introduce the space $\mathcal{F}$ of real valued functions $f : \mathbb{V} \to \mathbb{R}$. The scalar product on $\mathcal{F}$ is defined by

$$\langle f, g \rangle = \sum_{i=0}^{q-1} f(\mathbf{c}_i) g(\mathbf{c}_i), \qquad f, g \in \mathcal{F}.$$

We also introduce the velocity multiplication operators $V_\alpha : \mathcal{F} \to \mathcal{F}$, defined by $(V_\alpha f)(\mathbf{v}) = v_\alpha f(\mathbf{v})$, $\alpha = 1, \cdots, d$, the vector operator $\mathbf{V} = (V_1, \cdots, V_d)^T$ and the matrix operator $\mathbf{V} \otimes \mathbf{V}$ defined by $(\mathbf{V} \otimes \mathbf{V})_{\alpha\beta} = V_\alpha V_\beta$, $\alpha, \beta = 1, \cdots, d$. Finally, we define the contraction product between two matrices as

$$\mathbf{A} : \mathbf{B} = \sum_{\alpha,\beta=1}^{d} \mathbf{A}_{\alpha\beta} \mathbf{B}_{\alpha\beta}.$$

With this coordinate-free notation, we can describe the moments of the velocity distribution f as

$$\rho=\langle 1,f\rangle, \tag{2.1a}$$
$$\mathbf{j}=\langle 1,\mathbf{V}f\rangle, \tag{2.1b}$$
$$\Pi=\langle 1,\mathbf{V}\otimes\mathbf{V}f\rangle. \tag{2.1c}$$

Finally, we introduce the weights for a given velocity model through the function $f^*\in\mathcal{F}$, which satisfies

$$\langle 1,f^*\rangle=1, \tag{2.2a}$$
$$\langle 1,\mathbf{V}f^*\rangle=0, \tag{2.2b}$$
$$\langle 1,\mathbf{V}\otimes\mathbf{V}f^*\rangle=c_s^2 I, \tag{2.2c}$$

where c_s^2 is the speed of sound associated with the velocity model, and I is the identity matrix.

The lattice Boltzmann equation (LBE) with a relaxation type collision operator is now

$$f(n+1,j+\mathbf{c},\mathbf{c})-f(n,j,\mathbf{c})=A\left[f^{(eq)}(f)-f\right](n,j,\mathbf{c}), \tag{2.3}$$

where $A:\mathcal{F}\to\mathcal{F}$ is a linear mapping, and $f^{(eq)}(f):\mathcal{F}\to\mathcal{F}$ is the equilibrium distribution.

The analysis of the LBE (2.3) for advection-diffusion is similar to the Navier-Stokes case, with the main difference being that the equilibrium distribution now consists of only a linear part, namely

$$f^{(eq)}(f)=f^{L(eq)}(\rho,\rho\mathbf{U}), \tag{2.4a}$$
$$f^{L(eq)}(\rho,\mathbf{j})=(\rho+c_s^{-2}\mathbf{c}\cdot\mathbf{j})f^*, \tag{2.4b}$$

where $\rho=\langle 1,f\rangle$ and $\mathbf{U}$ is the given advection velocity. The absence of a quadratic part simplifies the analysis, but also requires a modification of some of the assumptions on the operator A compared to the Navier-Stokes case.

The assumptions on the collision operator A are:

(i) $\langle Af,g\rangle=\langle f,Ag\rangle$, $\forall f,g\in\mathcal{F}$;

(ii) A is positive semi-definite;

(iii) The even and odd functions form invariant subspaces of A;

(iv) $\mathrm{Ker}A=\mathrm{span}(\{1\})$;

(v) $A[(\mathbf{c}\cdot\mathbf{a})f^*]=\frac{c_s^2}{D^*}(\mathbf{c}\cdot\mathbf{a})f^*,\forall a\in\mathbb{R}^d$.

Here, D^* is a diffusion coefficient, which is related to the diffusion coefficient D in (1.1) (see below). Note that choosing $A=\frac{1}{\tau}P$, where P is the orthogonal projection onto the complement of the kernel of A, and $\tau=D^*/c_s^2$, gives the BGK collision operator.

In order to analyze the LBE (2.3), we expand the left hand side of Eq. (2.3) in the small parameter (grid spacing) ϵ, observing the diffusive scaling $\Delta t = \epsilon^2 = (\Delta x)^2$. We also expand f as $f = \sum_{m=0}^{\infty} \epsilon^m f^{(m)}$ and obtain

$$f^{(m)}(t_n+\epsilon^2,\mathbf{x}_j+\epsilon\mathbf{c})-f^{(m)}(t_n,\mathbf{x}_j)=\sum_{r=0}^{\infty}\epsilon^r D_r(\partial_t,\mathbf{c}\cdot\nabla)f^{(m)}(t_n,\mathbf{x}_j), \tag{2.5}$$

where

$$\begin{aligned} &D_r(\theta,\sigma)=\sum_{2a+b=r}\frac{\theta^a\sigma^b}{a!b!}, \quad r\geq 1,\\ &D_0(\theta,\sigma)=0, \quad D_1(\theta,\sigma)=\sigma, \quad D_2(\theta,\sigma)=\theta+\sigma^2/2. \end{aligned}$$

Furthermore, we set $\rho^{(k)}=\langle 1,f^{(k)}\rangle$ and $\mathbf{j}^{(k)}=\langle 1,\mathbf{V}f^{(k)}\rangle$ for $k=0,1,\cdots$.

Since the operator on the right-hand side of (2.3) is linear and acts locally in time and space, we may directly insert the expansion for f and collect terms of equal order of ϵ. In agreement with the low Mach-number assumption, we assume that

$$\mathbf{U}=\epsilon\mathbf{U}^{(1)}.$$

We then get for order ϵ^{k+2}, $k\geq -2$:

$$\sum_{m+r=k+2} D_r(\partial_t,\mathbf{c}\cdot\nabla)f^{(m)}=A\left[f^{L(eq)}(\rho^{(k+2)},\rho^{(k+1)}\mathbf{U}^{(1)})-f^{(k+2)}\right]. \tag{2.6}$$

Setting

$$L^{(k+2)}=-\sum_{\substack{m+r=k+2\\ m<k}} D_r(\partial_t,\mathbf{c}\cdot\nabla)f^{(m)},$$

we get

$$\sum_{m+r=k+2} D_r(\partial_t,\mathbf{c}\cdot\nabla)f^{(m)}=\partial_t f^{(k)}+(\mathbf{c}\cdot\nabla)f^{(k+1)}+\frac{1}{2}(\mathbf{c}\cdot\nabla)^2 f^{(k)}-L^{(k+2)}.$$

Note that $L^{(0)}=L^{(1)}=L^{(2)}=0$ and that $L^{(s)}$ is an odd function if s is odd, and an even function if s is even.

Since the kernel of A is generated by constants, we require for the solvability of (2.6) with respect to $f^{(k+2)}$ that the left hand side of (2.6) is orthogonal to constants, which gives us the solvability condition

$$\partial_t\rho^{(k)}+\nabla\cdot\mathbf{j}^{(k+1)}+\frac{1}{2}\nabla\otimes\nabla:\Pi^{(k)}=\langle 1,L^{(k+2)}\rangle, \tag{2.7}$$

and by applying the inverse $A^{\dagger}$ defined on the image of A to (2.6), we obtain

$$\begin{aligned} f^{(k+2)}=&f^{L(eq)}(\rho^{(k+2)},\rho^{(k+1)}\mathbf{U}^{(1)})\\ &-A^{\dagger}\left[\partial_t f^{(k)}+(\mathbf{c}\cdot\nabla)f^{(k+1)}+\frac{1}{2}(\mathbf{c}\cdot\nabla)^2 f^{(k)}\right]+A^{\dagger}L^{(k+2)}. \end{aligned} \tag{2.8}$$

We now wish to show that the 0-order moment $\rho^{(0)}=\langle 1,f^{(0)}\rangle$ satisfies the advection-diffusion equation. First, setting $k=-2$ in (2.8) yields $f^{(0)}=\rho^{(0)}f^*$, which implies that $\mathbf{j}^{(0)}=\langle 1,\mathbf{c}f^{(0)}\rangle=0$. Then, setting $k=-1$, we get

$$f^{(1)}=(\rho^{(1)}+c_s^{-2}(\mathbf{c}\cdot\mathbf{U}^{(1)})\rho^{(0)})f^*-A^{\dagger}(\mathbf{c}\cdot\nabla)f^{(0)}. \tag{2.9}$$

Taking first moments here and using property (v) for the collision operator A, we get

$$\mathbf{j}^{(1)}=\rho^{(0)}\mathbf{U}^{(1)}-D^*\nabla\rho^{(0)}. \tag{2.10}$$

With $k=0$ in (2.8), we get

$$f^{(2)}=(\rho^{(2)}+c_s^{-2}(\mathbf{c}\cdot\mathbf{U}^{(1)})\rho^{(1)})f^*-A^{\dagger}\left(\partial_t f^{(0)}+(\mathbf{c}\cdot\nabla)f^{(1)}+\frac{1}{2}(\mathbf{c}\cdot\nabla)^2f^{(0)}\right),$$

and thus

$$\mathbf{j}^{(2)}=\rho^{(1)}\mathbf{U}^{(1)}-D^*\nabla\rho^{(1)}. \tag{2.11}$$

Noting that $\Pi^{(k)}=c_s^2\rho^{(k)}I$, $k=0,1$, the solvability condition (2.7) gives

$$\begin{aligned}
k=-1:&\quad \nabla\cdot\mathbf{j}^{(0)}=0;\\
k=0:&\quad \partial_t\rho^{(0)}+\nabla\cdot\mathbf{j}^{(1)}+\frac{c_s^2}{2}\Delta\rho^{(0)}=0;\\
k=1:&\quad \partial_t\rho^{(1)}+\nabla\cdot\mathbf{j}^{(2)}+\frac{c_s^2}{2}\Delta\rho^{(1)}=\langle 1,L^{(3)}\rangle=0.
\end{aligned}$$

Combining this (for $k=0$) with (2.10), we get the advection-diffusion equation for $\rho^{(0)}$,

$$\partial_t\rho^{(0)}+\nabla\cdot(\rho^{(0)}\mathbf{U}^{(1)})-\left(D^*-\frac{c_s^2}{2}\right)\Delta\rho^{(0)}=0. \tag{2.12}$$

Comparing with the advection-diffusion equation (1.1), we see that we get the diffusion coefficient $D=(D^*-c_s^2/2)$.

For $\rho^{(1)}$, using (2.11), the resulting equation is identical,

$$\partial_t\rho^{(1)}+\nabla\cdot(\rho^{(1)}\mathbf{U}^{(1)})-\left(D^*-\frac{c_s^2}{2}\right)\Delta\rho^{(1)}=0. \tag{2.13}$$

However, the initial conditions differ in that $\rho^{(0)}|_{t=0}=\rho_0$ and $\rho^{(1)}|_{t=0}=0$, provided that the initial conditions for f are properly set. Thus, if the boundary conditions we set for f imply appropriate boundary conditions for $\rho^{(1)}$ (e.g. zero Neumann or zero Dirichlet boundary conditions) it can be ensured that the unique solution to (2.13) is $\rho^{(1)}\equiv 0$, and thus that

$$\rho=\rho^{(0)}+\epsilon^2\rho^{(2)}+\mathcal{O}(\epsilon^3), \tag{2.14a}$$

$$\mathbf{j}=\epsilon\mathbf{j}^{(1)}+\epsilon^3\mathbf{j}^{(3)}+\mathcal{O}(\epsilon^4). \tag{2.14b}$$

Therefore, in the next section, we will perform an asymptotic analysis of the boundary conditions in order to show that this is the case.

3 Neumann boundary conditions

3.1 Description of the scheme

To apply boundary conditions, we set the distribution function on *ghost points*, which are lattice points outside the domain, but with one or more velocity vectors $\mathbf{c}_i$ intersecting the boundary and thus ending up inside the domain after the streaming step. The distribution function on a ghost point is determined by using data from a *mirror point* inside the domain at the same distance from the boundary in the normal direction as the ghost point, see Fig. 1. The values of macroscopic variables are determined at the mirror point by interpolation, and these values are used to set the correct distribution function on the ghost point in order to get the desired boundary condition. Collisions are performed at the ghost points just as if they were inside the domain. Note, however, that it is only strictly necessary to assign values to the distribution function in the velocity directions intersecting the boundary since all moments are known from the interpolation procedure and the post-collision distribution can be computed directly.

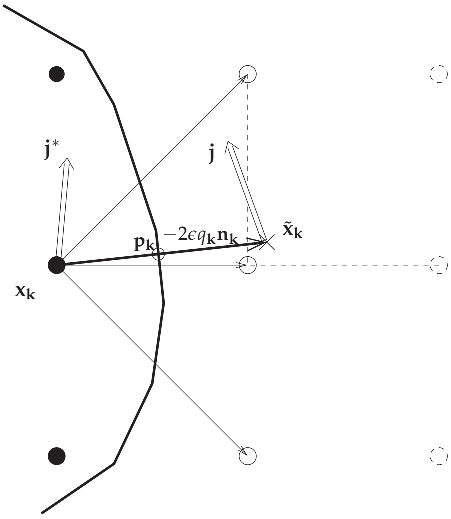
$\mathbf{j}^*$
$\mathbf{j}$
$-2\epsilon q_\mathbf{k}\mathbf{n}_\mathbf{k}$
$\tilde{\mathbf{x}}_\mathbf{k}$
$\mathbf{p}_\mathbf{k}$
$\mathbf{x}_\mathbf{k}$

Figure 1: A schematic picture near the boundary of the domain. Ghost points are marked with solid circles (●) and domain lattice points by empty circles (○). Some discrete velocity vectors intersecting the boundary are shown, as well as the vector $-2\epsilon q_\mathbf{k}\mathbf{n}_\mathbf{k}$ and the intersection $\mathbf{p}_\mathbf{k}$ (marked by a small circle (∘)). The mirror point is marked by ×, and lattice points used for interpolation are connected by dashed lines. The bold arrows show the mirroring of the flux vector $\mathbf{j}$ to $\mathbf{j}^*$.

We denote the location of the ghost point by $\mathbf{x}_\mathbf{k}$ and the mirror point by $\tilde{\mathbf{x}}_\mathbf{k}$. The intersection with the boundary occurs at $\mathbf{p}_\mathbf{k} = \mathbf{x}_\mathbf{k} - \epsilon q_\mathbf{k}\mathbf{n}_\mathbf{k}$, where $\mathbf{n}_\mathbf{k}$ is the outward unit normal at $\mathbf{p}_\mathbf{k}$ and $\epsilon q_\mathbf{k}$ is the distance in the normal direction. Then, the mirror point is defined through $\tilde{\mathbf{x}}_\mathbf{k} = \mathbf{x}_\mathbf{k} - 2\epsilon q_\mathbf{k}\mathbf{n}_\mathbf{k}$. We assume that the grid spacing ϵ is small enough compared to the radius of curvature of the boundary so that there is only one intersection point $\mathbf{p}_\mathbf{k}$ for each ghost point.

The basic idea of the boundary condition is to compute the solution at the mirror point $\tilde{\mathbf{x}}_\mathbf{k}$ and assign its mirror reflection to the ghost point $\mathbf{x}_\mathbf{k}$. We separate two cases: a) a simpler scheme without mass conservation and b) a scheme with an additional term which makes the boundary conditions mass conserving.

3.1.1 Non-mass conserving scheme

In the non-mass conserving scheme, we specify the moments of order 0 (the density ρ^*) and 1 (the flux $\mathbf{j}^*$) and assign the distribution function

$$f(t_n,\mathbf{x}_\mathbf{k}) = f^{L(eq)}(\rho^*,\mathbf{j}^*) = \rho^*(t_n,\tilde{\mathbf{x}}_\mathbf{k})f^* + c_s^{-2}(\mathbf{c}\cdot\mathbf{j}^*(t_n,\tilde{\mathbf{x}}_\mathbf{k}))f^*. \tag{3.1}$$

To achieve the correct mirroring across the boundary, we choose

$$\rho^* = \rho, \tag{3.2a}$$

$$\mathbf{j}^* = \mathbf{j} - 2(\mathbf{j}\cdot\mathbf{n}_\mathbf{k})\mathbf{n}_\mathbf{k}, \tag{3.2b}$$

where ρ^* and $\mathbf{j}^*$ are regarded as functions of t and $\mathbf{x}$.

3.1.2 Mass conserving scheme

In order to obtain a mass conserving scheme, we add a second moment contribution to (3.1) to compensate for the difference between outgoing and incoming populations in the next streaming step. To this end, we assign the distribution function

$$f(t_n,\mathbf{x}_\mathbf{k}) = f^{L(eq)}(\rho^*,\mathbf{j}^*) + \frac{1}{2c_s^4}\left(|\mathbf{c}|^2 - dc_s^2\right)\pi^*(t_n,\mathbf{x}_\mathbf{k})f^*, \tag{3.3}$$

where d is the number of space dimensions and π^* is a scalar function to be determined. Note that with the additional term, the second order moment tensor of f is $\Pi^* = c_s^2\rho^* I + \pi^* I$, while the zeroth and first order moments are unaffected by the additional term and are equal to ρ^* and $\mathbf{j}^*$, respectively.

In order to conserve mass, we choose

$$\pi^*(t_n,\mathbf{x}_\mathbf{k}) = 2c_s^4 \frac{\sum_{i\in\mathcal{O}}\left(\tilde{f}_i(t_{n+1},\mathbf{x}_\mathbf{k}-\mathbf{c}_i) - \tilde{f}_{\bar{\imath}}^{L(eq)}(\rho^*,\mathbf{j}^*)\right)}{(1+\lambda)\sum_{i\in\mathcal{O}} f_{\bar{\imath}}^*(|\mathbf{c}_{\bar{\imath}}|^2 - dc_s^2)}, \tag{3.4}$$

where $\mathcal{O}$ is the set of outgoing directions, which intersect the boundary and end up on the ghost point $\mathbf{x}_\mathbf{k}$, $\bar{\imath}$ is such that $\mathbf{c}_{\bar{\imath}} = -\mathbf{c}_i$ and $\tilde{f}$ is the post-collision distribution function. Finally, λ is the eigenvalue for the collision operator corresponding to the vector $\left(|\mathbf{c}|^2 - dc_s^2\right)f^*$. For the BGK operator, $\lambda = -1/\tau$, for the two-relaxation-time (TRT) operator, λ is the even eigenvalue λ_e, while for the multiple-relaxation-time (MRT) operator, a slightly different expression would be required, which is not difficult to derive.

The choice of π^* ensures exact conservation of mass on the individual ghost point level, after the collision step, while not affecting the moments of order 0 and 1 which enter in the asymptotic expansions in the next section. The expression (3.4) is derived through equating the sum of all outgoing post-collision components (the first term in the nominator) to the sum of all incoming post-collision components determined from (3.3). Note that this is only possible if $\lambda \neq -1$.

3.2 Asymptotic analysis of boundary conditions

Now, we can express the values of the distribution function $f(t_{n+1},\mathbf{x_k})$ in two ways by using Taylor expansions at the boundary point $\mathbf{p_k}$ at time t_n. First, a simple expansion yields (with $D_0(\theta,\sigma)=1$)

$$f(t_{n+1},\mathbf{x_k}) = \sum_{m=0}^{\infty} \epsilon^m \sum_{l=0}^{m} D_{m-l}(\partial_t, q_\mathbf{k}\mathbf{n_k}\cdot\nabla) f^{(l)}(t_n,\mathbf{p_k}). \tag{3.5}$$

On the other hand, we set the distribution function at $\mathbf{x_k}$ according to (3.3) with (3.2a)-(3.2b) and (3.4) (note that $\pi^*=0$ gives the non-conserving scheme). Thus,

$$\begin{aligned} f(t_{n+1},\mathbf{x_k}) = f^* \sum_{m=0}^{\infty} \epsilon^m D_m(\partial_t, -q_\mathbf{k}\mathbf{n_k}\cdot\nabla)\left(\rho^*(t_n,\mathbf{p_k}) + c_s^{-2}\mathbf{j}^*(t_n,\mathbf{p_k})\cdot\mathbf{c}\right) \\ + \frac{1}{2c_s^4}\left(|\mathbf{c}|^2 - dc_s^2\right)\pi^*(t_n,\mathbf{x_k}) f^*. \end{aligned} \tag{3.6}$$

Equating (3.5) and (3.6), and inserting the expressions for ρ^* and $\mathbf{j}^*$ from (3.2a)-(3.2b), we can study the accuracy of the boundary condition.

If we also take into account that we interpolate the values of ρ and $\mathbf{j}$ at $\tilde{\mathbf{x}}_\mathbf{k}$ to compute ρ^* and $\mathbf{j}^*$, we need to replace them by

$$\rho^* = \sum_{k=0}^{\infty} \epsilon^k \Big(\rho^{*(k)} + \sum_{l=p_\rho}^{\infty} \epsilon^l E_\rho^{(k,l)}\Big),$$

$$\mathbf{j}^* = \sum_{k=0}^{\infty} \epsilon^k \Big(\mathbf{j}^{*(k)} + \sum_{l=p_\mathbf{j}}^{\infty} \epsilon^l E_\mathbf{j}^{(k,l)}\Big),$$

where $E_\rho^{(k,l)}$ is the interpolation error term of order l for $\rho^{(k)}$, and p_ρ and $p_\mathbf{j}$ give the order of the interpolations (e.g. $p=2$ for linear interpolation). We will assume below that $p_\rho \geq 2$ and $p_\mathbf{j} \geq 1$.

Now, setting (3.5) and (3.6) equal and equating terms of the same order of ϵ, $m=0$ yields

$$f^{(0)} = \Big(\rho^{(0)} + \frac{1}{2c_s^4}\left(|\mathbf{c}|^2 - dc_s^2\right)\pi^{*(0)}\Big) f^*,$$

which, taking moments of order 0 and 1, implies just $\rho^{(0)} = \rho^{(0)}$ and $\mathbf{j}^{(0)} = 0$. For $m=1$, we get

$$q_{\mathbf{k}}(\mathbf{n}_{\mathbf{k}}\cdot\nabla)f^{(0)}+f^{(1)}=f^*\left(\rho^{(1)}-q_{\mathbf{k}}(\mathbf{n}_{\mathbf{k}}\cdot\nabla)\rho^{(0)}\right)+c_s^{-2}f^*\left(\mathbf{j}^{(1)}-2(\mathbf{j}^{(1)}\cdot\mathbf{n}_{\mathbf{k}})\mathbf{n}_{\mathbf{k}}\right)\cdot\mathbf{c}$$
$$+\frac{1}{2c_s^4}\left(|\mathbf{c}|^2-dc_s^2\right)\left(\pi^{*(0)}-q_{\mathbf{k}}(\mathbf{n}_{\mathbf{k}}\cdot\nabla)\pi^{*(1)}\right)f^*.$$

Computing the moments of orders 0 and 1, we now obtain

$$2q_{\mathbf{k}}(\mathbf{n}_{\mathbf{k}}\cdot\nabla)\rho^{(0)}=0, \tag{3.7a}$$
$$2(\mathbf{j}^{(1)}\cdot\mathbf{n}_{\mathbf{k}})\mathbf{n}_{\mathbf{k}}=0. \tag{3.7b}$$

Given that the advection velocity $\mathbf{U}^{(1)}\cdot\mathbf{n}_{\mathbf{k}}=0$ at the boundary, these two equations both give us back the Neumann boundary condition $\mathbf{j}\cdot\mathbf{n}=0$ in the lowest order, since $\mathbf{j}^{(1)}=-D\nabla\rho^{(0)}$. Note that there is no condition imposed on the flux in the tangential directions at the boundary.

In order to determine the accuracy of the boundary condition, we now need to proceed to higher orders. For $m=2$, the first two moments yield

$$2q_{\mathbf{k}}(\mathbf{n}_{\mathbf{k}}\cdot\nabla)\rho^{(1)}=E_\rho^{(0,2)}, \tag{3.8a}$$
$$2q_{\mathbf{k}}(\mathbf{n}_{\mathbf{k}}\cdot\nabla)\left(\mathbf{j}^{(1)}-(\mathbf{j}^{(1)}\cdot\mathbf{n}_{\mathbf{k}})\mathbf{n}_{\mathbf{k}}\right)-2(\mathbf{j}^{(2)}\cdot\mathbf{n}_{\mathbf{k}})\mathbf{n}_{\mathbf{k}}=-E_{\mathbf{j}}^{(1,1)}. \tag{3.8b}$$

If $E_\rho^{(0,2)}=0$, the first of these equations shows that $\rho^{(1)}$ satisfies a Neumann boundary condition. From (2.13) in Section 2, we know that $\rho^{(1)}$ satisfies an advection-diffusion equation with $\rho^{(1)}\big|_{t=0}=0$. Since the unique solution to this problem with a Neumann boundary condition is identically zero, we obtain $\rho^{(1)}\equiv 0$. However, if $E_\rho^{(0,2)}\neq 0$ (i.e. we use linear interpolation), this need not be the case, and we may have $\rho^{(1)}\neq 0$ and thus only a first order accurate solution.

If $E_{\mathbf{j}}^{(1,1)}=0$, taking the scalar product of (3.8b) with $\mathbf{n}_{\mathbf{k}}$ yields $\mathbf{j}^{(2)}\cdot\mathbf{n}_{\mathbf{k}}=0$, which again gives a Neumann boundary condition for $\rho^{(1)}$ (since $\mathbf{U}^{(2)}=0$), indicating that the choice of flux in (3.2b) is correct. Thus, at least linear interpolation of the flux is necessary.

We conclude that if interpolation errors are of order ϵ^3 for ρ, and of order ϵ^2 for $\mathbf{j}$, then the zeroth moment of the distribution function is

$$\rho=\rho^{(0)}+\epsilon^2\rho^{(2)}+\mathcal{O}(\epsilon^3), \tag{3.9}$$

where $\rho^{(0)}$ satisfies the advection-diffusion equation (1.1) with a Neumann boundary condition.

3.3 Interpolation

In order to achieve interpolation errors of order ϵ^3 for the density and ϵ^2 for the flux, as was shown to be required in the previous section, we make use of the fact that in the

Lattice Boltzmann method the gradient $\nabla\rho$ is available through the flux $\mathbf{j}$ computed as the first moment of the distribution function f, cf. (2.10) and (2.11). This enables us to compute a quadratic interpolating polynomial for approximation of ρ using only 4 grid points in 3D (and 3 in 2D). As an illustration, the grid points used for interpolation in the example in Fig. 1 are connected by the dashed lines.

This method of interpolation is referred to as *compact interpolation* and the 2D-case for the Navier-Stokes equations is analyzed under the term *bubble functions* in [5]. However, as the resulting formulas are quite different in the case of the diffusion equation, and in 3D, we present the interpolation procedure here.

Given a mirror point $\tilde{\mathbf{x}}_\mathbf{k}$ to which we want to interpolate the density and flux from the nearby grid point (see Fig. 1), we choose the closest grid point $\mathbf{x}_0\in\Omega$. In addition, we choose d adjacent grid points, $\mathbf{x}_{\mathbf{i}_\alpha}=\mathbf{x}_0+i_\alpha\mathbf{e}_\alpha\epsilon$, $\alpha=1,\cdots,d$, where $i_\alpha\in\{1,-1\}$, $\alpha=1,\cdots,d$ are chosen so that all $\mathbf{x}_{\mathbf{i}_\alpha}\in\Omega$, and so that the distance $|\tilde{\mathbf{x}}_\mathbf{k}-\mathbf{x}_{\mathbf{i}_\alpha}|$ is minimized. This ensures that we avoid extrapolation whenever we can, although extrapolation can in general not be avoided, depending on the location of the boundary. Also, depending on the geometry, there is no guarantee that such $\mathbf{x}_{\mathbf{i}_\alpha}$ exist, and we will discuss this below.

In order to interpolate the density, we introduce the local polynomial

$$\tilde{\rho}(\xi)=a_0+\sum_{\alpha=1}^{d}a_\alpha\xi_\alpha+\sum_{\alpha,\beta=1}^{d}a_{\alpha\beta}\xi_\alpha\xi_\beta \tag{3.10}$$

with $\xi=\epsilon^{-1}(\mathbf{x}-\mathbf{x}_0)$. Introducing the notation

$$\rho^0=\rho(\mathbf{x}_0),\quad \rho^0_\alpha=\frac{\partial\rho}{\partial x_\alpha}(\mathbf{x}_0),\quad \rho^{\mathbf{i}_\beta}=\rho(\mathbf{x}_{\mathbf{i}_\beta}),\quad \rho^{\mathbf{i}_\beta}_\alpha=\frac{\partial\rho}{\partial x_\alpha}\rho(\mathbf{x}_{\mathbf{i}_\beta}),$$

for $\alpha,\beta=1,\cdots,d$, and equating density values and derivatives at the chosen grid points, we get the following $(d+1)^2$ equations:

$$\rho^0=a_0, \tag{3.11a}$$

$$\rho^0_\alpha=a_\alpha, \tag{3.11b}$$

$$\rho^{\mathbf{i}_\beta}=a_0+a_\beta i_\beta+a_{\beta\beta}, \tag{3.11c}$$

$$\rho^{\mathbf{i}_\beta}_\alpha=a_\alpha+i_\beta(a_{\alpha\beta}+a_{\beta\alpha}) \tag{3.11d}$$

for $\alpha,\beta=1,\cdots,d$. Since the number of unknown coefficients in (3.10) is only $1+d+d^2=(d+1)^2-d$, we need to choose which equations to use. The simplest choice is to use (3.11a) and (3.11b) to compute a_0 and a_α, then use (3.11c) to compute $a_{\alpha\alpha}$, and finally use (3.11d) to compute $a_{\alpha\beta}$, $\alpha\neq\beta$, where we choose $a_{\alpha\beta}=a_{\beta\alpha}=\frac{1}{2}(a_{\alpha\beta}+a_{\beta\alpha})$. Thus, we do not make use of Eq. (3.11d) with $\alpha=\beta$ and do not need the value $\rho^{\mathbf{i}_\alpha}_\alpha$, $\alpha=1,\cdots,d$.

Now, with the coefficients of the polynomial (3.10) determined as above, it may be evaluated at the mirror point $\tilde{\xi}=\epsilon^{-1}(\tilde{\mathbf{x}}_\mathbf{k}-\mathbf{x}_0)$ to yield an approximate value of the density

$\rho(\tilde{\mathbf{x}}_{\mathbf{k}})$. Writing this explicitly, we get

$$\tilde{\rho}(\tilde{\xi}) = \rho^0 + \sum_{\alpha=1}^{d} \rho^0_\alpha \tilde{\xi}_\alpha + \sum_{\alpha=1}^{d} (\rho^{\mathbf{i}_\alpha} - \rho^0 - \rho^0_\alpha i_\alpha)\tilde{\xi}^2_\alpha + \frac{1}{2} \sum_{\substack{\alpha,\beta=1 \\ \alpha\neq\beta}}^{d} i_\beta (\rho^{\mathbf{i}_\beta}_\alpha - \rho^0_\alpha)\tilde{\xi}_\alpha \tilde{\xi}_\beta. \tag{3.12}$$

This explicit expression has the advantage that we can directly compute the contribution of each of the values at the grid points to the total value of $\tilde{\rho}(\tilde{\mathbf{x}}_{\mathbf{k}})$. This is of importance to reduce the amount of communication in parallel computations where the grid points may reside on different computer nodes.

For the flux, standard linear interpolation between the same grid points will be sufficient to achieve the $\mathcal{O}(\epsilon^2)$-error required by the analysis. The rate of convergence is guaranteed in the following proposition.

Proposition 3.1. The interpolation procedure described above yields interpolation errors of orders $\mathcal{O}(\epsilon^3)$ for ρ and $\mathcal{O}(\epsilon^2)$ for $\mathbf{j}$.

Proof. Expanding $\tilde{\rho}$ in a Taylor series at $\xi = 0$, we see that the polynomial coefficients coincide up to and including order ϵ^2, giving an $\mathcal{O}(\epsilon^3)$-error. The result for the flux is standard and is proved similarly. □

Finally, we consider the situation when we can not find interpolation points $\mathbf{x}_{\mathbf{i}_\alpha}$ inside the domain, because different parts of the boundary are located too closely. One solution in this case is to increase the grid resolution, but when this is impossible it may in many situations be acceptable to use the bounce-back rule as fall-back. This is true especially when the geometry has uncertainties as in the case when it is generated from experimental data. Using the bounce-back rule will introduce errors since the tangential flux will be zero, but the effect on the global solution will be limited as long as the bounce-back grid points are few and isolated.

4 Numerical results

In this section, we study three numerical examples in 3D. In the first two, the exact solution is an eigenfunction of the Laplacian inside a sphere, the first being spherically symmetric and the second having an angular dependence which introduces a tangential flux at the boundary. The third example is computing the effective diffusion constant for a periodic array of spheres.

The computations were performed using the BGK collision model [2, 3], using the relaxation parameter $\tau = 1.1$ unless otherwise noted. In all cases, the sphere geometry was implemented by an exact sphere, and the intersection points $\mathbf{p}_{\mathbf{k}}$ were chosen as the closest point on the spherical surface for each ghost point $\mathbf{x}_{\mathbf{k}}$. The distribution function was initialized as

$$f(0,\mathbf{x}_{\mathbf{k}}) = f^{L(eq)}(\rho_0(\mathbf{x}_{\mathbf{k}}), \mathbf{j}_0(\mathbf{x}_{\mathbf{k}})),$$

where $\rho_0 = \psi_0$ and $\mathbf{j}_0 = -D^*\nabla\psi_0$ in accordance with (2.10) without advection velocity. Here, ψ_0 is the given initial condition. See [1] for a discussion on initial conditions. Results are presented for the D3Q7 velocity model with 7 discrete velocities and the D3Q19-model with 19 discrete velocities.

Results using a bounce-back boundary scheme are included for comparison. The bounce-back scheme used is a "bounce-back on the link" scheme, where outgoing velocities are returned to the domain grid point in the same time-step. This scheme was used because it has exact mass conservation.

4.1 Symmetric solution inside a sphere

Let B_R denote the ball of radius R in $\mathbb{R}^3$. We study the PDE

$$\begin{cases} \partial_t\psi = D\Delta\psi & \text{in } B_R, \\ \psi|_{t=0} = \sqrt{\frac{2}{\pi\mu}}\cdot\frac{1}{r}\sin\mu r & \text{in } B_R, \\ \partial_n\psi = 0 & \text{on } \partial B_R. \end{cases} \tag{4.1}$$

If μ solves the equation $\tan(\mu R) = \mu R$, then the analytical solution to (4.1) is (in spherical coordinates)

$$\psi(t,r) = \exp\left(-D\mu^2 t\right)\sqrt{\frac{2}{\pi\mu}}\frac{1}{r}\sin\mu r. \tag{4.2}$$

Here, we choose μ to be the smallest positive solution to $\tan(\mu R) = \mu R$, that is $\mu \approx 4.49341/R$. Note that the tangential flux is zero in this example because of the spherical symmetry. For the numerical solution, we choose $R = 0.4$ and immerse the sphere in a cubical grid with side-length 1 and lattice spacing ϵ.

Fig. 2 shows the dependence of the errors on the grid size ϵ^{-1} for the solution at time $t = 0.15$. The slopes of lines fitted to the log-log data are shown in Table 1. The data confirms $\mathcal{O}(\epsilon^2)$-convergence for the density and $\mathcal{O}(\epsilon^3)$-convergence for the flux as predicted in Section 3.2 for the mass conserving scheme and for the bounce-back scheme. The non-mass conserving scheme has lower convergence rate because mass is lost at the boundary

Table 1: Slopes of lines fitted to the log-log data in Fig. 2, indicating the rate of convergence.

Model	Density	Flux
D3Q7, with mass conservation	-2.29	-2.95
D3Q19, with mass conservation	-2.27	-3.71
D3Q7, no mass conservation	-1.47	-2.73
D3Q19, no mass conservation	-0.93	-2.67
D3Q7, bounce-back	-2.28	-3.09
D3Q19, bounce-back	-2.27	-3.09

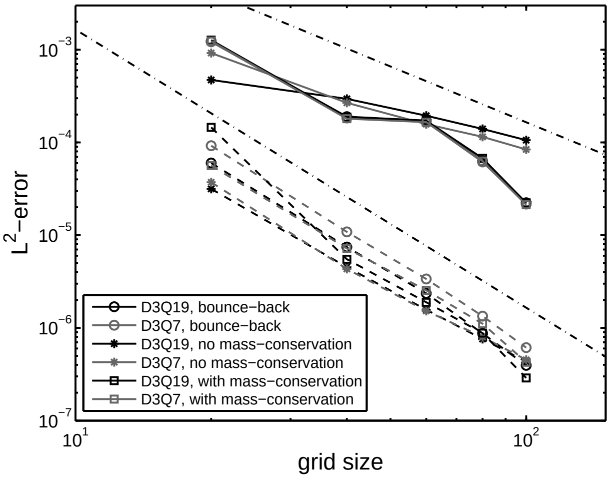

Figure 2: L^2-errors for density (solid lines) and flux (dashed lines) for the diffusion problem (4.1), for the D3Q7 model (grey) and the D3Q19 model (black) with bounce-back (○), mass conserving Neumann (□) and non-mass conserving Neumann (∗) boundary conditions. Note that the mass conserving scheme and the bounce-back scheme give almost identical results for the density. The dash-dotted lines indicate slope −2 (top) and slope −3 (bottom).

at each time-step. The bounce-back scheme performs well because the tangential flux for the exact solution is zero, which is what the bounce-back scheme produces.

4.2 Non-symmetric solution inside a sphere

Let again B_R denote the ball of radius R in $\mathbb{R}^3$. To investigate a non-symmetric problem, we now study the PDE

$$\begin{cases} \partial_t \psi = D\Delta\psi & \text{in } B_R \times [0,T], \\ \psi|_{t=0} = \psi_0(r,\theta,\phi) & \text{in } B_R, \\ \partial_n \psi = 0 & \text{on } \partial B_R \times [0,T], \end{cases} \tag{4.3}$$

with

$$\psi_0(r,\theta,\phi) = \sqrt{\frac{2}{\pi\mu}} \cdot \left(\frac{1}{\mu r} \sin\mu r - \cos\mu r \right) \cdot \frac{\sin\theta\cos\phi}{r}, \tag{4.4}$$

in spherical coordinates (r,θ,ϕ).

If μ solves the equation $\tan(\mu R) = -2\mu R/((\mu R)^2 - 2)$, then the analytical solution to (4.3) is

$$\psi(t,x,r) = \exp\left(-D\mu^2 t\right)\psi_0(r,\theta,\phi). \tag{4.5}$$

Here, we choose μ to be the smallest positive solution to the equation $\tan(\mu R) = -2\mu R/((\mu R)^2 - 2)$, that is $\mu \approx 2.08158/R$. Again, for the numerical solution, we choose $R = 0.4$ and immerse the sphere in a cubical grid with side-length 1.

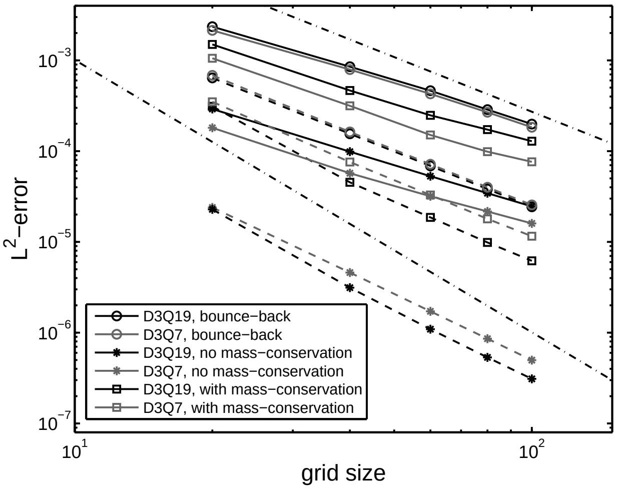

Figure 3: L^2-errors for density (solid lines) and flux (dashed lines) for the diffusion problem (4.3), for the D3Q7 model (grey) and the D3Q19 model (black) with bounce-back (○), mass conserving Neumann (□) and non-mass conserving Neumann (∗) boundary conditions. The dash-dotted lines indicate slope -2 (top) and slope -3 (bottom).

The numerical results at time $t = 0.15$ compared to the analytical solution are shown in Fig. 3, and the slopes of lines fitted to the log-log data are given in Table 2. The convergence is here approximately $\mathcal{O}(\epsilon^{2.5})$ for the flux and $\mathcal{O}(\epsilon^{1.5})$ for the density, i.e. almost but not quite what is predicted by the analysis in Section 3.2. The non-mass conservative scheme has smaller errors, because tangential fluxes near the boundary are most correct with this scheme, while mass conservation is not as important in this example.

Table 2: Slopes of lines fitted to the log-log data in Fig. 3, indicating the rate of convergence.

Model	Density	Flux
D3Q7, with mass conservation	-1.66	-2.11
D3Q19, with mass conservation	-1.53	-2.42
D3Q7, no mass conservation	-1.50	-2.67
D3Q19, no mass conservation	-1.53	-2.40
D3Q7, bounce-back	-1.52	-2.04
D3Q19, bounce-back	-1.53	-2.03

4.3 Effective diffusion coefficient for a periodic array of spheres

Our final numerical example is the computation of effective diffusion coefficient for a periodic array of spheres. This problem has a semi-analytical solution, which can be found in [15].

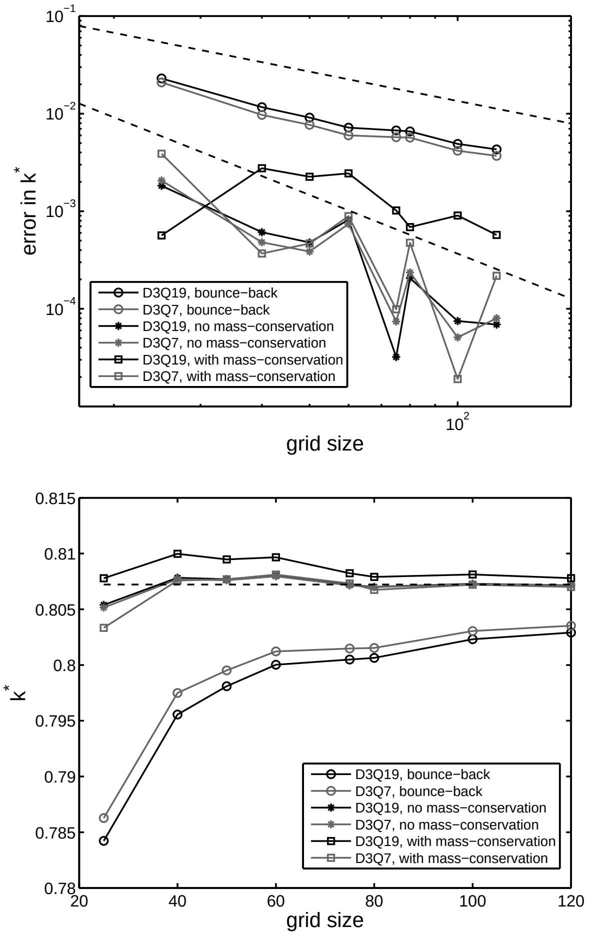

Figure 4: Computed values of k^* for diffusion in a periodic array of spheres for the D3Q7 model (grey) and the D3Q19 model (black) with bounce-back (○), mass-conserving (□) and non-mass conserving ($*$) boundary conditions for different grid sizes N compared to the analytical value. At the top is a log-log plot for the absolute errors in k^*. The slopes of lines fitted to the data are -1.0 for the bounce-back boundary condition, -1.54 for the D3Q19 model with mass conservation (excluding the value for $N=25$), and about -2.3 for the others. The slopes of the dashed lines are -1 and -2. The somewhat irregular convergence behavior is partly explained by the plot at the bottom, which shows the values of k^*, where the computed values for some models oscillate around the analytical value (the dashed line).

In our setup, we apply Dirichlet boundary conditions for the density in the x-direction, $\rho|_{x=0}=\rho_1$ and $\rho|_{x=L_x-1}=\rho_2$. After solving for a steady-state solution, we compute the average flux in the x-direction, denoted $\bar{\mathbf{j}}$, over a plane perpendicular to that direction. The effective diffusion constant D_{eff} is then computed from Fick's law

$$\bar{\mathbf{j}}_x = -D_{\text{eff}} \frac{\rho_2 - \rho_1}{L_x}, \tag{4.6}$$

where L_x is the size of the domain in the x-direction. Finally the *geometry factor* k^* is

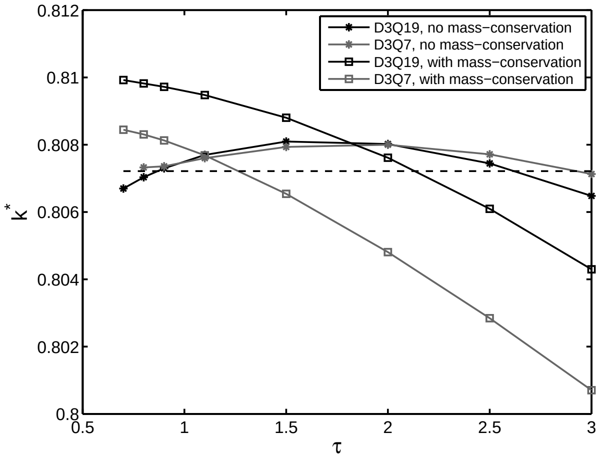

Figure 5: The computed values of $k^* = D_{\text{eff}}/D$ for diffusion in a periodic array of spheres using different values of the relaxation parameter τ (right). Results for the D3Q7 model (red) and the D3Q19 model (blue) with mass conserving (□) and non-mass conserving ($*$) Neumann boundary conditions are compared to the analytical result (dashed black line).

computed as

$$k^* = D_{\text{eff}}/D, \tag{4.7}$$

where D is the free diffusion constant appearing in (1.1). From (4.6), we see that if $\mathbf{j}$ is accurate to $\mathcal{O}(\epsilon^3)$, the convergence of k^* should be $\mathcal{O}(\epsilon^2)$ (since $L_x \propto \epsilon^{-1}$).

Since the analytical result is for an array of spheres that is periodic in all directions, we compute the solution in a domain containing 5 identical spheres after each other to mimic periodicity in the x-direction, and compute $\bar{\mathbf{j}}_x$ near the central sphere in order to minimize boundary effects. Numerical tests with more spheres in line indicate that the errors due to this approximation are small enough not to influence the results appreciably. Periodic boundary conditions are applied in the y- and z-directions.

In Fig. 4, we show the dependence of the error in k^* on the lattice spacing ϵ. For our boundary conditions, the convergence is around the predicted $\mathcal{O}(\epsilon^2)$, while the bounce-back boundary condition performs poorly and has linear convergence.

In Fig. 5, the dependence of the computed value of k^* on the relaxation parameter τ in the BGK collision model is presented. The value of k^* depends strongly on τ for both the D3Q7 and the D3Q19 model with mass conserving boundary condition, but less strongly for non-mass conserving boundary conditions.

5 Conclusions

We have presented a novel boundary condition for the lattice Boltzmann diffusion equation, which implements the Neumann boundary condition for the advection-diffusion equation. An important property of our boundary condition is the possibility of exact mass conservation. The analysis shows that second order accuracy should be obtained for the density, and third order accuracy for the flux. In the numerical examples, this rate of convergence is obtained in case the tangential flux on the boundary is zero (see Section 4.1). Otherwise the convergence rates reduce to approximately $\mathcal{O}(\epsilon^{2.5})$ for the flux and $\mathcal{O}(\epsilon^{1.5})$ for the density (see Section 4.2). As expected, the bounce-back boundary conditions do not reproduce the correct tangential flux and thus do not implement the Neumann boundary conditions correctly when the tangential flux is non-zero.

For computation of effective diffusion coefficients, our scheme shows a clear advantage over the bounce-back scheme, as is to be expected since the bounce-back scheme does not implement the correct Neumann boundary condition. For a more complicated geometry, the effect is expected to be even larger. The D3Q19 model produces slightly worse results with mass conserving boundary conditions, which may be due to that the assignment of a second moment at ghost points introduces a perturbation of the equations. Possibly, assigning a higher-order moment would remedy this situation.

The different velocity models give very similar results. Our results indicate that there is no reason to use models with a larger number of velocities than 7, as this increases the computational time but does not increase the accuracy in our tests. The only limitation of the D3Q7 model is if the diffusion coefficient tensor has non-zero off-diagonal entries. In that case it is shown in [6] that more discrete velocities are needed. We note that our boundary conditions could be applied to the case of general diffusion tensors with only minor modifications.

It should also be noted that in a domain with pure Neumann boundary conditions, where the total mass is supposed to be conserved (as in Section 4.1) the exact conservation of mass is essential for the accuracy of the scheme. If the mass is not exactly conserved, the numerical solution will gradually lose mass and deviate substantially from the exact solution. When conservation of mass is not as essential, as in Sections 4.2 and 4.3, the non-mass conserving scheme is more accurate due to more accurate tangential fluxes at the boundary.

It would be interesting to compare our results to results for the same problems using the boundary condition suggested by Ginzburg [7]. This will be the subject of future work. Further lines of future work are the use of two relaxation time collision models [6,8] to eliminate the dependence on the relaxation parameter in Fig. 5, as well as investigations on how triangulated surface geometries influence the accuracy of the solution. Finally, an attempt at modifying the mass-conserving scheme by assigning higher-order moments in ghost nodes will be made.

Acknowledgments

This work was financed by the VINN Excellence center SuMo Biomaterials, supported by Vinnova. This support is gratefully acknowledged.

References

[1] A. Caiazzo. Analysis of lattice boltzmann initialization routines. J. Stat. Phys., 121(1-2):37–48, 2005.

[2] H. Chen, S. Chen, and W. H. Matthaeus. Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. Phys. Rev. A, 45:R5339–R5342, 1992.

[3] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. Ann. Rev. Fluid Mech., 30:329–364, 1998.

[4] S. Y. Chen, Z. Wang, X. Shan, and G. D. Doolen. Lattice Boltzmann computational fluid dynamics in three dimensions. J. Stat. Phys., 68(3-4):379–400, 1992. Advanced Research Workshop on Lattice Gas Automata (Nice, 1991).

[5] M. Geier, A. Greiner, and J. G. Korvink. Bubble functions for the lattice Boltzmann method and their application to grid refinement. Eur. Phys. J.–Special Topics, 171:173–179, 2009.

[6] I. Ginzburg. Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation. Adv. Water Resour., 28(11):1171–1195, 2005.

[7] I. Ginzburg. Generic boundary conditions for lattice Boltzmann models and their application to advection and anisotropic dispersion equations. Adv. Water Resour., 28(11):1196–1216, 2005.

[8] I. Ginzburg, F. Verhaeghe, and D. d'Humieres. Two-relaxation-time lattice Boltzmann scheme: About parametrization, velocity, pressure and mixed boundary conditions. Commun. Comp. Phys., 3(2):427–478, 2008.

[9] F. J. Higuera, S. Succi, and R. Benzi. Lattice gas dynamics with enhanced collisions. Europhys. Lett., 9(4):345, 1989.

[10] A. Hiorth, U. H. A. Lad, S. Evje, and S. M. Skjaeveland. A lattice Boltzmann-BGK algorithm for a diffusion equation with Robin boundary condition-application to NMR relaxation. Int. J. Numer. Meth. Fluids, 59(4):405–421, 2009.

[11] H.-B. Huang, X.-Y. Lu, and M. C. Sukop. Numerical study of lattice Boltzmann methods for a convection-diffusion equation coupled with Navier-Stokes equations. J. Phys. A, 44(5):055001, 2011.

[12] M. Junk, A. Klar, and L.-S. Luo. Asymptotic analysis of the lattice Boltzmann equation. J. Comp. Phys., 210:676–704, 2005.

[13] Q. Kang. Simulation of dissolution and precipitation in porous media. J. Geophys. Res., 108(B10):1–10, 2003.

[14] K. Moriyama and T. Inamuro. Lattice boltzmann simulations of water transport from the gas diffusion layer to the gas channel in pefc. Commun. Comp. Phys., 9:1206–1218, 2011.

[15] A. S. Sangani and A. Acrivos. The effective conductivity of a periodic array of spheres. Proc. R. Soc. A, 386(1791):263–275, 1983.

[16] M. Stiebler, J. Tölke, and M. Krafczyk. Advection-diffusion lattice Boltzmann scheme for hierarchical grids. Comput. Math. Appl., 55(7):1576–1584, 2008.

[17] F. Verhaeghe, S. Arnout, B. Blanpain, and P. Wollants. Lattice-Boltzmann modeling of dissolution phenomena. Phys. Rev. E, 73(3):036316, 2006.

[18] H. Yoshida and M. Nagaoka. Multiple-relaxation-time lattice Boltzmann model for the convection and anisotropic diffusion equation. J. Comp. Phys., 229(20):7774–7795, 2010.

[19] X. X. Zhang, J. W. Crawford, A. G. Bengough, and I. M. Young. On boundary conditions in the lattice Boltzmann model for advection and anisotropic dispersion equation. Adv. Water Resour., 25(6):601–609, 2002.