Journal of Computational Physics 229 (2010) 6781–6803

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A curved no-slip boundary condition for the lattice Boltzmann method

Joris C.G. Verschaeve *, Bernhard Müller

Department of Energy and Process Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

ARTICLE INFO

Article history:
Received 26 June 2009
Received in revised form 28 April 2010
Accepted 20 May 2010
Available online 26 May 2010

Keywords:
Curved boundary
Lattice Boltzmann
Flow around a circular cylinder

ABSTRACT

We present a generalization of the no-slip boundary condition by Lätt et al. [J. Lätt, B. Chopard, O. Malaspinas, M. Deville, A. Michler, Straight velocity boundaries in the lattice Boltzmann method, Physical Review E 77 (5) (2008) 056703] from straight to curved geometries for the lattice Boltzmann Bhatnager–Gross–Krook method (LBGK). The boundary condition is based on a reconstruction of the populations from the density, velocity and rate of strain. For curved boundaries, the reconstruction reduces the question of accuracy to a technical issue of interpolation. We present a method of interpolation allowing a very accurate representation of the curved boundary. The resulting boundary condition is verified for three different test cases: Taylor–Couette flow in-between rotating cylinders, laminar flow around a cylinder and flow past an impulsively started cylinder, demonstrating its second order accuracy and low error constant. The present boundary is stable for relaxation frequencies close to two.

1. Introduction

The lattice Boltzmann method has become increasingly popular for computing fluid flows [30,33,14]. It interpretes a fluid rather as an ensemble of particles than as continuous matter. The description of these particles is done by a particle distribution function which is continuous in space and time and depends in addition on the particle velocities. The particles can travel from one grid node to the other, where they undergo collisions with each other. These two principles are materialized in the streaming and collision steps of the lattice Boltzmann method. Macroscopic quantities, such as density or momentum, are then recovered as statistical moments of the particle distribution function, the central object of the lattice Boltzmann method. From a numerical point of view, the lattice Boltzmann method is a second order scheme in space and time, if compressibility effects are negligible.

As mentioned in [20], boundary conditions are still an open issue. Rather than focusing on the general velocity boundary condition, the present discussion treats a particular case, namely the no-slip condition at curved walls, meaning that the viscous fluid has the same velocity at the wall as the wall itself.

The difficulty of boundary conditions in lattice Boltzmann methods is to find a formulation for the particle distribution functions leaving into the bulk fluid domain, but which do not stem from the fluid domain.

Since the lattice Boltzmann method is only defined for equidistant Cartesian grids, boundary conditions were first introduced for straight walls. These boundary conditions can be divided into two major families, the bounce back [10] and the so-called wet boundary conditions [13]. For the former, boundary nodes, nodes where the boundary conditions are applied, lie outside of the fluid domain, whereas for the latter, they are on the boundary but still part of the fluid domain. Therefore, the former ones only formulate a closure for the unknown populations, whereas the latter ones still apply a collision step before streaming similar to a bulk node. However, if the boundary is not aligned with the grid lines, a staircase like approximation

* Corresponding author. Tel.: +47 73593590.
E-mail address: joris.verschaeve@ntnu.no (J.C.G. Verschaeve).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.05.022

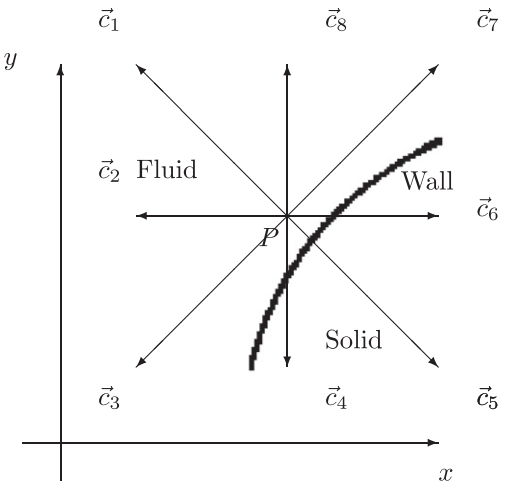

Fig. 1. The D2Q9-lattice node P and its position near the wall splitting the computational domain into a fluid part and a solid part.

of the boundary would lead to a scheme of only first order accuracy in space [10]. For a curved boundary, cf. Fig. 1, the nodes of the lattice can be divided into nodes lying inside the fluid domain (fluid nodes) and nodes lying outside the fluid domain. In order to avoid any confusion, we will call the latter ones solid nodes representing the region outside the fluid domain as a solid body. The boundary itself will then be referred to as the wall, in analogy to a no-slip boundary at a solid wall. Several attempts have been made to formulate a boundary closure allowing to increase the accuracy from first to second order. A generalization of the bounce back boundary condition for straight walls was presented in Refs. [1,18]. Keeping in mind that when using the bounce back boundary condition for a straight wall the exact position of the wall is somewhere in the middle between the bounce back node and the next fluid node, they formulate a scheme interpolating the populations link wise such that a bounce back collision around the intersection of the link in question and the wall is reproduced.

The idea of the bounce back scheme is also reflected in the boundary treatment of Chen et al. [3,21]. In this approach the curved boundary is approximated by facets on which a reflection of the particles occurs. To increase the accuracy of this scheme a correction term including the velocity gradient is introduced as a source term to the bounced back particle distribution functions. In addition, Junk and Yang presented another interesting curved boundary treatment based on a correction of the bounce back condition in [16].

Another family of curved boundary conditions is the boundary fitting method proposed by Fillipova and Hänel [9] and improved by Mei et al. [22]. This method uses an extrapolation of the velocity at the neighboring solid nodes and makes use of the populations coming from fluid nodes in order to find the outgoing unknown populations at the boundary nodes.

An extrapolation of the velocity to the neighboring solid nodes is also used in the method of Guo et al. [12], which they consider to be an extension of the method by Chen et al. [5]. The unknown populations are split into their equilibrium and nonequilibrium parts and each part is computed independently, thereby using an extrapolation of the velocity on the solid node for the equilibrium part. The unknown density and nonequilibrium parts of the populations are then taken from the neighboring fluid nodes along the link.

An entirely different approach is chosen by the immersed boundary method. This method originates from the works of Peskin [24] for macroscopic solvers, but has been reformulated for lattice Boltzmann methods by many authors [8,28,7]. The main characteristic of this method is the use of a local force accounting for the effect of the wall on the fluid. In order to determine this local force, Dupuis et al. [7] need to compute the velocity at the fluid nodes next to the boundary node one time step ahead and interpolate then the desired velocity at the boundary node. The forcing term is introduced as a source term during collision.

The boundary treatment in the present discussion can be applied both for interpolation and extrapolation of the velocity at the boundary node. Since it formulates a closure for the populations and applies a collision step afterwards it can be thought of as being a wet boundary condition. It is based on a well founded, very general formalism derived by Lätt [19,20] who uses an earlier variant by Skordos [29]. This formulation has also been mentioned by Junk in [15]. The strategy of the present curved boundary treatment is to accurately inter- or extrapolate macroscopic quantities at the boundary nodes and to reconstruct the populations by Lätt's reconstruction formalism [19], before applying a usual collision and streaming steps, cf. Section 2. Therefore, it can be understood as a generalization of the finite difference boundary condition for straight walls of Lätt et al. [20].

The present curved boundary treatment reduces the issue of accuracy and stability to a rather technical matter of interpolation for which methods are derived, cf. Section 3, after having introduced basic notions of the lattice Boltzmann method in Section 2. The benchmark tests, Section 4, are chosen in such a way as to allow comparisons with other methods. The first benchmark test is steady Taylor–Couette flow between two cylinders, with the interior cylinder rotating at a constant angular speed and the exterior cylinder at rest, cf. Section 4.2. A second benchmark test is taken from the Priority Research

Program workshop *Benchmark Computations of Laminar Flow Around a Cylinder* [27] consisting of two parts: steady laminar flow around a cylinder confined in-between channel walls at a Reynolds number of Re = 20 and time-periodic flow around the same cylinder at a Reynolds number of Re = 100, cf. Section 4.3. As an initial-boundary value problem, the flow around an impulsively started cylinder was chosen as a third benchmark test case in Section 4.4. Finally, the present discussion is concluded in Section 5.

2. Lattice Boltzmann BGK

In this section we outline the main elements of the lattice Boltzmann method necessary for the present discussion. We treat only the lattice Boltzmann Bhatnager–Gross–Krook (LBGK) version of the lattice Boltzmann method. For a more detailed introduction to the lattice Boltzmann method, and LBGK in particular, we refer to [30,33,14]. The central object of the lattice Boltzmann method is the 'particle' distribution function $f(t,\vec{x},\vec{c})$ which is a function of time t, the position $\vec{x}$ and the particle velocity $\vec{c}$. It corresponds to the probability of finding a particle at a given position $\vec{x}$ with a given velocity $\vec{c}$. In its discrete form, the velocities a particle can travel at are limited to a very small number $\vec{c}_i,\ i=0,\ldots,q-1$. For instance in the D2Q9 lattice, the lattice used throughout this discussion, cf. Fig. 1 and Table 1, a particle can only have one out of nine discrete velocities. These velocities are not arbitrary but chosen in such a way that the system is isotropic and that the velocities are interconnecting the grid nodes of the computational domain. The computational grid and the ensemble of velocities are called a lattice. In addition, lattice weights t_i, $i=0,\ldots,q-1$, are associated to all discrete velocities $\vec{c}_i$, accounting for the difference in length of the lattice vectors. For the D2Q9 lattice, the lattice weights t_i and lattice velocities $\vec{c}_i$ are defined in Table 1. In a broader sense we will use the word 'link' to designate the direction defined by a lattice velocity. For example the spatial diagonal in the first quadrant of the two-dimensional Cartesian coordinate system can be defined by the link $\vec{d}=\frac{1}{\sqrt{2}}\vec{c}_7$ and the origin as the position vector, i.e. we refer to the link as a direction vector (taking only $q-1$ discrete values) of a line in two dimensions. The term 'link wise' is thus equivalent to saying 'according to each of the spatial directions defined by the lattice vectors $\vec{c}_i$ with $i=1,\ldots,q-1$'.

A fundamental condition of the lattice Boltzmann method is that a particle travels exactly the distance Δx, the grid spacing, during one time step Δt. Therefore the particle speed, or lattice constant c, is given by:

$$c=\frac{\Delta x}{\Delta t}.$$

For a number of lattices, the speed of sound c_s is related to the lattice constant by

$$c_s=\frac{c}{\sqrt{3}}.$$

Usually the system of units is chosen such that:

$$\Delta x=\Delta t=c=1.$$

When referring to the grid spacing or time step in its actual physical units we use a lower case δ instead. For example, if we choose the characteristic length L to be 1 and we use $N+1$ points to resolve this length, we have $\delta x=1/N$.

With the discrete set of particle velocities, the distribution function $f(t,\vec{x},\vec{c})$ can then be written as follows:

$$f(t,\vec{x},\vec{c})=\begin{cases} f_0(t,\vec{x}) & \text{if } \vec{c}=\vec{c}_0,\\ \vdots & \vdots\\ f_i(t,\vec{x}) & \text{if } \vec{c}=\vec{c}_i,\\ \vdots & \vdots\\ f_{q-1}(t,\vec{x}) & \text{if } \vec{c}=\vec{c}_{q-1},\\ 0 & \text{else}\end{cases}$$

where f_i is called a population. This leads to q equations, the so-called lattice Boltzmann equation, taking the following form:

$$f_i(t+1,\vec{x}+\vec{c}_i)=f_i(t,\vec{x})+\Omega_i, \tag{1}$$

where Ω_i is the collision operator accounting for the effect of the collisions particles undergo at the lattice nodes. Eq. (1) can be divided into two steps: collision and streaming. If we denote the populations before collision with the superscript *in* and the populations after collision with the superscript *out*, Eq. (1) can be rewritten as follows:

Table 1
The lattice velocities $\vec{c}_i$ and lattice weights t_i of the D2Q9 lattice.

i	0	1	2	3	4	5	6	7	8
t_i	4/9	1/36	1/9	1/36	1/9	1/36	1/9	1/36	1/9
$\vec{c}_i$	(0,0)	(−1,1)	(−1,0)	(−1,−1)	(0,−1)	(1,−1)	(1,0)	(1,1)	(0,1)

$$f_i^{out}(t,\vec{x}) = f_i^{in}(t,\vec{x}) + \Omega_i, \tag{2}$$

$$f_i^{in}(t+1,\vec{x}+\vec{c}_i) = f_i^{out}(t,\vec{x}). \tag{3}$$

The collision step is local whereas the streaming step is nonlocal and corresponds to the propagation of particles after collision to neighboring nodes. Sometimes we will also use the superscript *ps* for post streamed populations. That is to say, after all populations have been streamed to the neighboring nodes, we apply a so-called repopulation or reconstruction step changing the post streamed populations into *in* populations. For bulk nodes the repopulation step is identical to the identity operation in LBGK, but at boundary nodes this is not necessarily the case. For notational convenience, the superscript *in* will often be dropped. During collision, mass and momentum are conserved. The collision operator is usually defined in terms of a relaxation of the populations f_i towards an equilibrium distribution f_i^{eq}, the Bhatnager–Gross–Krook (BGK) operator:

$$\Omega_i = -\omega(f_i - f_i^{eq}), \tag{4}$$

where ω is called the relaxation frequency. The most common choice for the equilibrium function is a low Mach number expansion of the Maxwell distribution:

$$f_i^{eq}(\rho,\vec{u}) = t_i\rho\left(1 + \frac{1}{c_s^2}\vec{c}_i\cdot\vec{u} + \frac{1}{2c_s^4}\mathbf{Q}_i : \vec{u}\vec{u}\right), \tag{5}$$

where the tensor $\mathbf{Q}_i$ is defined as follows:

$$\mathbf{Q}_i = \vec{c}_i\vec{c}_i - c_s^2\mathbf{I},$$

where $\mathbf{I}$ is the unit tensor. Macroscopic quantities such as density ρ or momentum density $\rho\vec{u}$ are given by the statistical moments of the distribution function, i.e.

- density:

$$\rho = \sum_i f_i, \tag{6}$$

- momentum density:

$$\vec{j} = \rho\vec{u} = \sum_i \vec{c}_i f_i, \tag{7}$$

- second order tensor:

$$\mathbf{\Pi} = \sum_i \vec{c}_i\vec{c}_i f_i. \tag{8}$$

The standard procedure to derive the governing equations for these macroscopic quantities is called the Chapman–Enskog multi-scale analysis [33]. The central idea of the Chapman–Enskog multi-scale analysis is that the populations can be developed into an asymptotic expansion of a small quantity ϵ which is identified as the Knudsen number, the ratio of the mean free path to the reference length [33,14,4]:

$$f_i = f_i^{(0)} + \epsilon f_i^{(1)} + \cdots$$

The zeroth order term of this series is taken to be the equilibrium distribution. The difference $f_i^{neq} = f_i - f_i^{eq}$ is usually referred to as the nonequilibrium part. It is to second order equal to $\epsilon f_i^{(1)}$. In order to conserve mass during collision the sum of the nonequilibrium parts has to be zero:

$$\sum_i f_i^{neq} = 0. \tag{9}$$

Similarly, the conservation of momentum requires that:

$$\sum_i \vec{c}_i f_i^{neq} = \vec{0} \tag{10}$$

The multi-scale analysis leads to an expression for the first order term of the populations:

$$f_i^{(1)} = -\frac{t_i}{c_s^2\omega}\left(\mathbf{Q}_i : \rho\nabla\vec{u} - \vec{c}_i\nabla : \rho\vec{u}\vec{u} + \frac{1}{2c_s^2}(\vec{c}_i\cdot\nabla)(\mathbf{Q}_i : \rho\vec{u}\vec{u})\right). \tag{11}$$

A result of the multi-scale analysis is that macroscopic quantities obey the incompressible Navier–Stokes equations for small Mach and Knudsen numbers [33,14]:

$$\nabla\cdot\vec{u} = 0, \tag{12}$$

$$\partial_t\rho\vec{u} + \nabla\cdot(\rho\vec{u}\vec{u}) = -\nabla p + \nabla\cdot\boldsymbol{\tau}, \tag{13}$$

where the pressure p is given by the equation of state of an ideal gas:

$$p = c_s^2 \rho.$$

The viscous stress tensor $\boldsymbol{\tau}$ is defined as:

$$\boldsymbol{\tau} = 2\nu\rho\mathbf{S}.$$

The kinematic viscosity ν can then be related to the relaxation frequency ω:

$$\nu = c_s^2\left(\frac{1}{\omega} - \frac{1}{2}\right).$$

The rate of strain tensor $\mathbf{S} = (\nabla\vec{u} + (\nabla\vec{u})^T)/2$ is related to the second order moment of the first order part of the populations $f_i^{(1)}$:

$$\Pi^{(1)} = \sum_i \vec{c}_i\vec{c}_i f_i^{(1)} = -\frac{2c_s^2}{\omega}\rho\mathbf{S}. \tag{14}$$

We remark that Eqs. (12) and (13) are solved by the scheme up to an error composed of three contributions:

- The spatial error scales like δx^2, where δx is the grid spacing.
- The temporal error scales like δt^2, where δt is the time step.
- The compressibility error scales like Ma^2, where Ma is the Mach number.

Since the Mach number Ma scales like $\delta t/\delta x$, the time step needs to be scaled like $\delta t \propto \delta x^2$ in oder to recover the incompressible Navier–Stokes equations up to second order in space. In the Chapman–Enskog expansion only $\Pi^{(1)}$ enters into the derivation of the hydrodynamic equations, instead of $f_i^{(1)}$ alone. This fact has been used by Lätt [19] to develop a method reconstructing the populations from the pure knowledge of ρ, $\vec{u}$ and $\mathbf{S}$ at the node, by first computing these quantities and then the nonequilibrium part of the populations by:

$$f_i^{neq} \approx -\frac{\rho t_i}{c_s^2\omega}\mathbf{Q}_i : \mathbf{S}, \tag{15}$$

and finally reconstructing the populations by using:

$$f_i^{in} = f_i^{eq} + f_i^{neq}. \tag{16}$$

This procedure is called the regularized lattice Boltzmann method [19]. Lätt's reconstruction method will be used for the no-slip boundary condition derived in Section 3. At the bulk fluid nodes, the usual LBGK collision step is performed.

3. A curved no-slip boundary condition for LBGK

We restrict ourselves to the two-dimensional case, more precisely the D2Q9 lattice, cf. Fig. 1 and Table 1. As for straight walls the main problem of boundary conditions for LBGK lies in the fact that at the nodes close to the boundary, as depicted in Fig. 1, some of the populations are unknown after the streaming step. However, for the curved case, there exist different choices on how to define the boundary nodes. One could for example define the boundary nodes to be the nodes lying in the fluid domain but having a link intersected by the wall, as depicted in Fig. 2. Then we distinguish between bulk fluid nodes (black circles) and boundary fluid nodes (gray circles) at which we need to find closures for the unknown populations after streaming. The solid nodes do not play a role for this configuration and are deprived of any dynamics. Since the boundary nodes are in-between the fluid points at which $\vec{u}$ is known after streaming and the wall at which we impose a boundary condition for $\vec{u}$, this choice of boundary nodes is therefore associated with an *interpolation* of $\vec{u}$ as we shall see later on. We shall call this configuration *configuration 1* in the following. For the boundary node P in question, cf. Fig. 2, the unknown populations are then f_1, f_2 and f_8, since they do not have any fluid nodes from which they are streamed in. We classify the populations at a boundary node in such a way that we shall call the indices, whose lattice vectors are pointing towards bulk fluid nodes fluid indices and we use the letter $\mathcal{F}$ to denote the ensemble of them. In our example of node P in Fig. 2, we have:

$$\mathcal{F} = \{1\}. \tag{17}$$

The indices i of the vectors $\vec{c}_i$ pointing towards itself or other boundary fluid nodes define the elements of the set $\mathcal{B}$ of boundary indices. In our example for point P in Fig. 2, we have

$$\mathcal{B} = \{0, 2, 3, 7, 8\}. \tag{18}$$

Finally, the vectors directed towards solid nodes will form the ensemble of solid indices $\mathcal{S}$, here $\mathcal{S} = \{4, 5, 6\}$. The ensemble of all indices $\mathcal{Q}$ can then be written as:

$$\mathcal{Q} = \mathcal{F} \cup \mathcal{B} \cup \mathcal{S}. \tag{19}$$

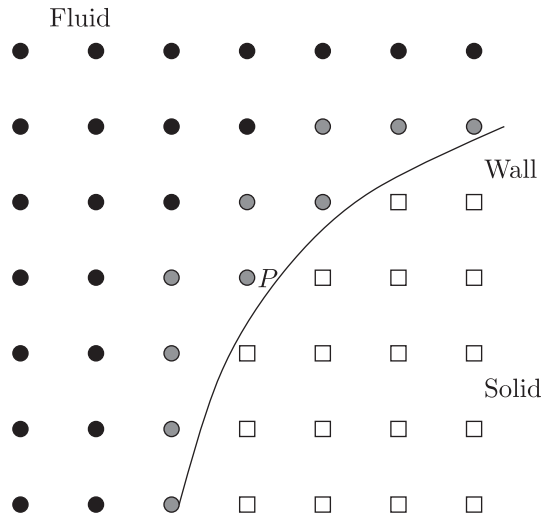

Fig. 2. Position of the wall, dividing the domain into a fluid and a solid domain. The nodes of the computational domain lie either in the fluid domain (circles) or in the solid domain (squares). In this case we choose to define the boundary nodes (gray) to be the fluid nodes having a link intersected by the wall (configuration 1). Black circles: Bulk fluid nodes; Gray circles: Boundary fluid nodes; White squares: Solid nodes; P is the boundary node of interest.

The superscript *opp* is used to denote the ensemble of indices of the opposite vectors. Therefore, the ensemble of indices of the unknown populations (the unknown indices) is:

$$\mathcal{S}^{opp} = \{1, 2, 8\}.$$

Instead of interpolating the velocity, most groups, such as Guo et al. [12] and Fillipova and Hänel [9], cf. Section 1, use an extrapolation of $\vec{u}$ at the boundary solid nodes, cf. Fig. 3. In this case the boundary nodes are defined by lying in the solid domain and having a link intersected by the wall. This configuration will be denoted *configuration 2*. For the boundary node *P* of interest, cf. Fig. 3, the above ensembles are then:

$$\mathcal{F} = \{1, 2, 8\}, \quad \mathcal{S} = \{5\}, \quad \mathcal{B} = \{0, 3, 4, 6, 7\}.$$

The present strategy is to define the boundary nodes to be the nodes being adjacent to the wall with a minimum distance to it, as shown in Fig. 4. This is done by comparing the nodes of configuration 1 and configuration 2 to each other and selecting first those nodes from both ensembles which have all links intersected by the wall with a distance smaller than half the distance between two nodes of the different ensembles. Then the gaps between these nodes with minimum distance need to be filled by looking at the link between alternative nodes (one in the fluid region and the other in the solid region) and choosing

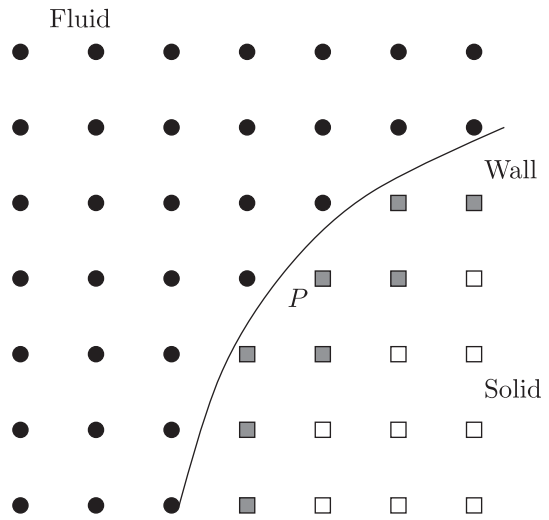

Fig. 3. Same as Fig. 2 but the boundary nodes (gray) are now defined to be the solid nodes having a link intersected by the wall (configuration 2). Black circles: Fluid nodes; Gray squares: Boundary solid nodes; White squares: Bulk solid nodes; P is the boundary node of interest.

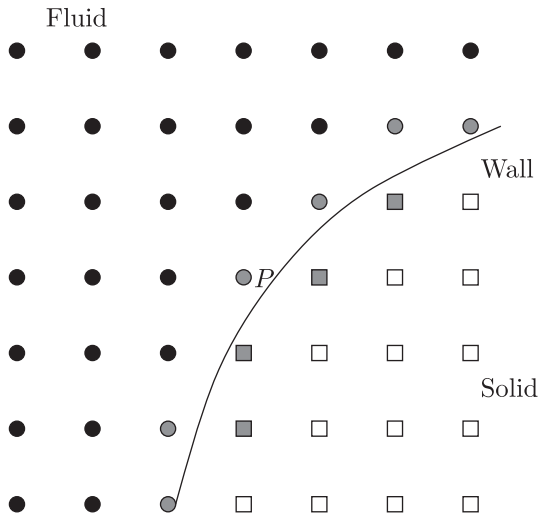

Fig. 4. Same as Figs. 2 and 3 but this time the boundary nodes (gray) are the solid and fluid nodes with the minimum distance to the intersection with the wall on the intersected link (configuration 3). Black circles: Bulk fluid nodes; Gray circles: Boundary fluid nodes; Gray squares: Boundary solid nodes; White squares: Bulk solid nodes; P is the boundary node of interest.

the node with the smaller distance to the intersection between link and boundary, giving us a row of boundary nodes such that all fluid nodes are separated from the solid nodes. In this case we have to *interpolate* $\vec{u}$ for the nodes lying in the fluid domain (the boundary fluid nodes, gray circles) and to *extrapolate* $\vec{u}$ at the nodes lying in the solid region (the boundary solid nodes, gray squares). We shall call this arrangement *configuration 3*. The ensembles of the indices at the boundary node P of interest, cf. Fig. 4, are in this case defined in the following way:

$$\mathcal{F} = \{1,2,3,8\}, \quad \mathcal{S} = \{5\}, \quad \mathcal{B} = \{0,4,6,7\}.$$

Opposed to straight walls, the ensembles $\mathcal{F}$, $\mathcal{B}$ and $\mathcal{S}$ are different for each boundary node. Furthermore, the boundary nodes are not all situated directly at the wall. Therefore, they have a non-zero velocity which complicates the situation additionally, since this velocity is unknown a priori.

3.1. Derivation

The basic equations of the regularized lattice Boltzmann method [19], (5), (15) and (16), were proven to have a very wide applicability for the lattice Boltzmann method. Apart from the regularized lattice Boltzmann method, revealing increased stability, these equations were used by Lätt to develop a grid refinement treatment [19] and boundary conditions for straight walls [20]. The fundamental property of this reconstruction of particle distribution functions f_i^{in} is the fact that only macroscopic quantities are needed to accurately specify f_i^{in}. These macroscopic quantities are the fluid density ρ, the fluid velocity $\vec{u}$ and the rate of strain tensor $\mathbf{S}$. If these quantities at the nodes close to a curved boundary, the boundary nodes, cf. Figs. 2–4, were known for each time step, the boundary could be very accurately represented by injecting these into the reconstruction formulae (5), (15) and (16). The accuracy of our method depends therefore primarily on the accuracy of the approximation of ρ, $\vec{u}$ and $\mathbf{S}$. Once we have found sufficiently accurate approximations, we use Eqs. (5), (15) and (16) to determine all particle distribution functions f_i^{in} at the boundary nodes. Thereafter, we apply the usual collision step (2) followed by the streaming step (3). Therefore, this method belongs to the so-called wet boundary conditions and can be understood as a generalization of Lätt et al.'s finite difference boundary condition for straight walls [20] for curved boundaries. The bulk solid nodes are inactive nodes which do not enter the computation.

In the following, we shall derive schemes to accurately predict ρ, $\vec{u}$ and $\mathbf{S}$.

(a) Approximation of $\vec{u}$ by interpolation along links
After streaming, the velocity $\vec{u}$ can be computed by formulae (6) and (7). In case of a no-slip boundary condition the velocity is given on the wall (Dirichlet boundary condition). However, at the boundary nodes, independent of the configuration chosen, the velocity is unknown, since we do not have populations coming from bulk solid nodes in order to apply formulae (6) and (7). Therefore, we need to interpolate or extrapolate the velocity at the boundary node using the velocity at the bulk fluid nodes close to the boundary node along a link, e.g. H_1 and H_2, cf. Fig. 5. Opposed to most groups, cf. Section 1, we do not interpolate or extrapolate the velocity link wise, but choose the index among the fluid indices whose associated link forms the smallest angle with the wall normal. Thus, we determine the link $\vec{d}$ by:

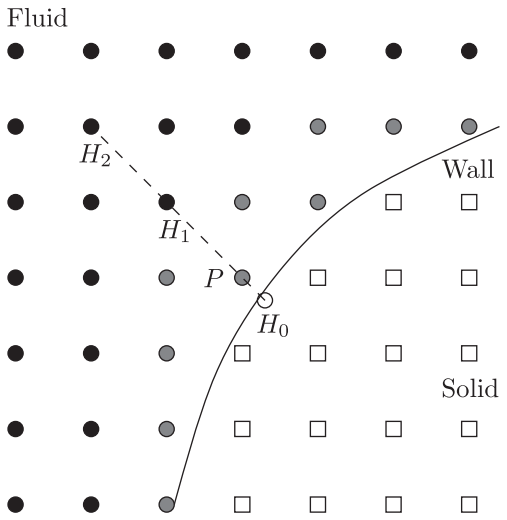

Fig. 5. Interpolation along the link between grid points. The points H_0, H_1 and H_2 are used for interpolation at the boundary node P of interest, symbols cf. Fig. 2.

$$\vec{d} = \frac{\vec{c}_j}{\|\vec{c}_j\|} \text{ where } j \text{ is the index for which } \frac{\vec{c}_i \cdot \vec{n}}{\|\vec{c}_i\|} \text{ is maximum}, \; i \in \mathcal{F}, \tag{20}$$

where $\vec{n}$ is the wall normal through the boundary node P, cf. Fig. 5. We are sure that $\vec{d}$ exists, since the ensemble $\mathcal{F}$ cannot be empty by definition of the boundary node. In the rare case of having two maxima, we choose the smallest index.

The interpolation/extrapolation of u at P is straightforward using quadratic Lagrangian interpolation polynomials [26,25]. First, we compute the distances d_P, d_{H_1} and d_{H_2} of the nodes P, H_1 and H_2 to the origin H_0, the intersection of the link with the wall, cf. Fig. 5. We define d_P to be positive in case of an interpolation, i.e. P is a boundary fluid node, or negative in case of an extrapolation, i.e. P is a boundary solid node. Since the nodes H_1 and H_2 are bulk fluid nodes, cf. Fig. 5, we can compute the velocity $\vec{u}_1$, resp. $\vec{u}_2$ after streaming at these nodes. The velocity $\vec{u}_0$ at the point H_0 is given (Dirichlet Boundary condition). An estimate of $\vec{u}$ at the boundary node P can thus be found by evaluating the interpolation polynomial at d_P:

$$\vec{u}(d_P) = \vec{u}_0 l_0(d_P) + \vec{u}_1 l_1(d_P) + \vec{u}_2 l_2(d_P), \tag{21}$$

where $l_i(d_P)$, $i = 0,1,2$ are the Lagrangian interpolation polynomials of the distances $d_{H_0}(=0)$, d_{H_1} and d_{H_2} evaluated at d_P:

$$l_0(d_P) = \frac{(d_P - d_{H_1})(d_P - d_{H_2})}{d_{H_1} d_{H_2}}, \tag{22}$$

$$l_1(d_P) = \frac{d_P(d_P - d_{H_2})}{d_{H_1}(d_{H_1} - d_{H_2})}, \tag{23}$$

$$l_2(d_P) = \frac{d_P(d_P - d_{H_1})}{d_{H_2}(d_{H_2} - d_{H_1})}. \tag{24}$$

These quantities can be computed once and for all in the preprocessing step.

(b) Approximation of **S** by finite difference stencils along the grid lines
In addition to the velocity $\vec{u}$, we approximate of the rate of strain tensor **S** at the boundary node P in the following way: Once the velocity is known at every boundary node by the procedure discussed above, we use second order finite difference stencils along the grid axes to determine $\partial u_i/\partial j$, $i,j = x,y$, cf. Fig. 6. Since the velocity at the node P enters the computation of $\nabla\vec{u}$, it is important to accurately compute $\vec{u}$ at P by the above method. The nodes which are used for the finite difference stencil can be fluid nodes, boundary nodes or intersections of the grid lines with the wall, cf. Fig. 6 giving rise to either central or one sided finite difference stencils depending on the position of the boundary node at the wall.

(c) Approximation of ρ locally
Opposed to the approximation of $\vec{u}$ and **S** described above, a purely local method can be derived to gain an approximation of ρ at the boundary node P. As already mentioned in Section 1, the difficulty of boundary conditions for the lattice Boltzmann method consists in certain populations at the boundary node, coming from bulk solid nodes, being

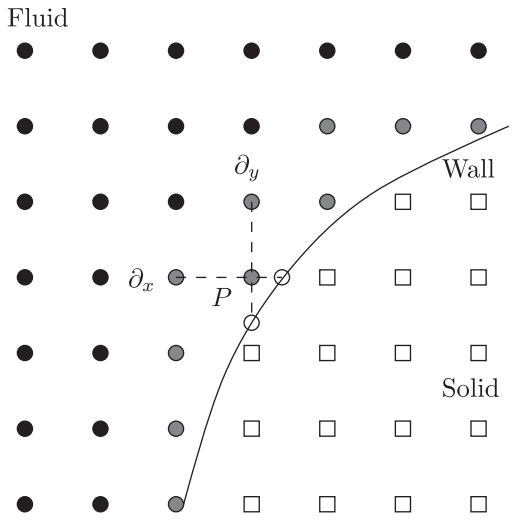

Fig. 6. Computation of the rate of strain **S** at the boundary node P of interest. The points on the horizontal line are used for the computation of $\partial_x\vec{u}$ and those on the vertical line for $\partial_y\vec{u}$, symbols cf. Fig. 2.

unknown. Other populations stemming from the bulk fluid or boundary nodes are known but so far we did not make any use of this information to determine the unknown populations. Instead, we chose to abandon the idea of finding the unknown populations as a function of the known ones and decided to reconstruct all populations, the known and the unknown ones, using Lätt's reconstruction formalism [19]. Nevertheless, we can extract an approximation of ρ from the populations streamed in. From formulae (5) and (15), we observe that the density is separable in (16), such that we can write:

$$f_i^{in} = \rho g_i^{in} = \rho\left(g_i^{eq}(\vec{u}) + g_i^{neq}(\mathbf{S})\right), \tag{25}$$

where g_i^{eq} is defined as

$$g_i^{eq}(\vec{u}) = f_i^{eq}(1,\vec{u}),$$

and g_i^{neq}as

$$g_i^{neq}(\mathbf{S}) = f_i^{neq}(1,\mathbf{S}).$$

We remark that g_i^{eq} and g_i^{neq} can be computed as soon as we know $\vec{u}$ and **S** at the boundary node. We denote the ensemble of the known indices by $\mathcal{K}$. This ensemble contains all indices of the populations which are known after streaming. A possible choice for $\mathcal{K}$ would be the opposite ensembles of the fluid and boundary indices:

$$\mathcal{K} = \mathcal{F}^{opp} \cup \mathcal{B}^{opp},$$

since these populations do not stem from solid nodes. Another choice would be to choose only the populations stemming from fluid nodes:

$$\mathcal{K} = \mathcal{F}^{opp}.$$

In Ref. [32], it is shown that at the boundary the populations coming from boundary nodes can lead to numerical instabilities. Therefore, we choose in analogy $\mathcal{K} = \mathcal{F}^{opp}$. The strategy of a local formulation of ρ is then to approximate ρ by imposing the conditions:

$$f_i^{ps} = \rho g_i^{in}, \quad i \in \mathcal{K}, \tag{26}$$

where f_i^{ps} is the value of the population i after streaming and before replacing it by ρg_i^{in}. If the cardinality of $\mathcal{K}$ is larger than 1, the system of equations, composed by (26), is overdetermined for ρ. A solution of (26) can be found in terms of least squares:

$$\rho = \frac{\sum_{i\in\mathcal{K}} f_i^{ps} g_i^{in}}{\sum_{j\in\mathcal{K}} \left(g_j^{in}\right)^2}. \tag{27}$$

On the other hand, we might also formulate a solution of (26) in terms of a weighted average by

$$\rho = \frac{1}{|\mathcal{K}|}\sum_{i\in\mathcal{K}} \frac{f_i^{ps}}{g_i^{in}}, \tag{28}$$

where $|\mathcal{K}|$ is the cardinality of $\mathcal{K}$. Both ways of approximating ρ, Eqs. (27) and (28), share the shortcoming of not coinciding with the definition of the density, Eq. (6), if $\mathcal{K} = \mathcal{Q}$, i.e. in the case of a bulk fluid node.
Another density approximation can be written by summing over all known populations and dividing by the corresponding sum of the g_i^{in}:

$$\rho = \frac{\sum_{i\in\mathcal{K}} f_i^{ps}}{\sum_{i\in\mathcal{K}} g_i^{eq} + g_i^{neq}}. \tag{29}$$

For $\mathcal{K} = \mathcal{Q}$, this approximation coincides with the definition of the density, Eq. (6). For a straight wall, we see that the resulting approximation, Eq. (29), coincides with the approximation of the density for the no-slip boundary condition presented in [32], when choosing $\mathcal{K} = \mathcal{F}^{opp}$.
All three approximations of ρ, Eqs. (27)–(29) do not conserve mass, in the sense that the sum of populations streamed out from the boundary nodes onto the bulk fluid nodes after collision does not necessarily equal the sum of populations streamed in from the bulk fluid nodes after streaming. This mass conservation condition was first formulated by Chopard and Dupuis [6] for a boundary node at a straight wall and can be generalized to a curved boundary node by:

$$m^{out} = m^{ps}, \tag{30}$$

where m^{ps} is the mass which is streamed in from the bulk fluid nodes onto the boundary nodes after streaming:

$$m^{ps} = \sum_{i\in\mathcal{F}^{opp}} f_i^{ps}. \tag{31}$$

The mass streamed out after collision m^{out} is given by:

$$m^{out} = \sum_{i\in\mathcal{F}} f_i^{out} \tag{32}$$

$$= \sum_{i\in\mathcal{F}} (1-\omega) f_i^{in} + \omega f_i^{eq} \tag{33}$$

$$= \rho \sum_{i\in\mathcal{F}} (1-\omega) g_i^{neq} + g_i^{eq}, \tag{34}$$

where we have used Eq. (25) for the last step. The mass conservation condition (30) leads to another possible approximation of ρ:

$$\rho = \frac{m^{ps}}{\sum_{i\in\mathcal{F}} (1-\omega) g_i^{neq} + g_i^{eq}}. \tag{35}$$

Eq. (35) coincides with the definition of the density, Eq. (6) for $\mathcal{F} = \mathcal{Q}$. In Ref. [32] it is shown that for a straight wall the mass conservation condition (30) leads to imposing:

$$\frac{\partial u_n}{\partial n} = 0, \tag{36}$$

where u_n is the normal velocity at the wall. This is physically justified for an incompressible fluid [23]. However, for a curved boundary the ensemble of fluid indices $\mathcal{F}$ is different for each boundary node, such that imposing (30) on a curved boundary node does not necessarily lead to imposing (36) at this node. Imposing (30) by using the density approximation (35) might therefore induce unphysical artefacts.
We remark that we found a local approximation of ρ. Nevertheless, it is always possible to find a nonlocal approximation of ρ by extrapolation from bulk fluid nodes. In addition, we have a Neumann boundary condition for the pressure p at the wall [11]:

$$\frac{\partial p}{\partial n} = \mu \left(\frac{\partial^2}{\partial \tau^2} + \frac{\partial^2}{\partial n^2} \right) u_n, \tag{37}$$

where τ denotes the tangential variable. For higher Reynolds numbers this boundary condition can be simplified to:

$$\frac{\partial p}{\partial n} = 0. \tag{38}$$

Since we are using the link the most normal to the boundary for the interpolation or extrapolation of $\vec{u}$, we might as well take the density from the bulk fluid node at this link as a fairly accurate approximation of ρ, given Eq. (38).

To conclude this section, we sum up the algorithm:

1. Compute the velocity $\vec{u}$ at each boundary node by means of interpolation/extrapolation.
2. Compute the rate of strain tensor $\mathbf{S}$ at each boundary node by finite difference stencils along the grid lines using the velocities at the boundary nodes computed previously.

3. Compute the density ρ at each boundary node locally by using the values of $\vec{u}$ and $\mathbf{S}$ computed before.
4. Reconstruct the populations at each boundary node by using formulae (5), (15) and (16).
5. Apply the collision step at all nodes.
6. Apply the streaming step at all nodes.

4. Verification

In this section, we present the results of benchmark tests we performed in order to assess the accuracy and stability of the no-slip boundary condition derived above. We considered the following test cases: steady Taylor–Couette flow in-between two circular cylinders in Section 4.2, flow around a circular cylinder confined in-between channel walls in Section 4.3, and flow around an impulsively started circular cylinder, Section 4.4. For the verifications, we used the definition of the numerical error outlined in the following Section 4.1.

4.1. Numerical error

The $\mathcal{L}_2$ norm of the error can be estimated by a power k of the grid size and a constant C. The power k is called the order of convergence, because it indicates the order by which the error is reduced with grid refinement. For a steady flow system, the resolution is only spatial. In case of a lattice Boltzmann method, the grid spacings δx and δy are uniform in both spatial directions. Therefore, the spatial resolution is only given by one parameter N, the number of sections into which the reference length is divided by the grid points.

$$\delta x = \delta y = \frac{1}{N}. \tag{39}$$

In two dimensions, having a reference solution of the velocity $\vec{u}_{ref}$, we can define the velocity error as follows:

$$\epsilon_N = \sqrt{\frac{1}{N_{err}} \sum_{\vec{x} \in N_{err}} |\vec{u}_{num}(\vec{x}) - \vec{u}_{ref}(\vec{x})|^2} \leqslant C \frac{1}{N^k}, \tag{40}$$

where N_{err} is the number/ensemble of points belonging to the fluid domain. The corresponding pressure error is defined in a similar way. When dealing with an integral quantity, such as the drag coefficient C_D, we define the error as:

$$\epsilon_N = |C_{D,num} - C_{D,ref}|. \tag{41}$$

The grid refinement is done by keeping the viscosity constant, which implies that we have to choose the time step δt to be proportional to the square of the grid spacing δx [33]:

$$\delta t \propto \delta x^2 = \frac{1}{N^2}. \tag{42}$$

In this way, the compressibility error is also reduced with second order accuracy, since it scales with the square of the Mach number. The Mach number itself scales with $\delta t / \delta x$.

4.2. Verification part 1

The aim of this section is to verify the accuracy of the present boundary condition for a laminar Taylor–Couette flow [2] in-between two circular cylinders, where the interior cylinder with radius r_1 is rotating at a constant angular speed Ω and the exterior cylinder with radius r_2, $r_2 > r_1$, is at rest. There exists an analytical solution to the Navier–Stokes equations for this flow. The velocity $\vec{u}$ is given by

$$\vec{u}(r) = \frac{u_0 \beta}{1 - \beta^2} \left(\frac{r_2}{r} - \frac{r}{r_2} \right) \vec{e}_\theta,$$

where $u_0 = r_1 \Omega$ is the velocity of the inner wall and $\beta = r_1 / r_2$. Determining the pressure constant by imposing $p(r_2) = 0$, we get the following pressure profile:

$$\frac{p}{\rho u_0^2} = \left(\frac{\beta}{1 - \beta^2} \right)^2 \frac{1}{2} \left(\frac{r^2}{r_2^2} - \frac{r_2^2}{r^2} - 4 \ln \frac{r}{r_2} \right).$$

The Reynolds number $\mathrm{Re} = \frac{(r_2 - r_1) u_0}{\nu}$ is fixed to 10 and $\beta = 0.5$ for all simulations. Subsequently, we impose a homogenous Dirichlet boundary condition $\vec{u} = 0$ for the velocity at r_2 and an inhomogeneous Dirichlet boundary condition $\vec{u} = u_0 \vec{e}_\theta$ at r_1. In case of the pressure, we have a homogeneous Dirichlet boundary condition at r_2.

Numerically, this problem is solved on a square lattice, where the boundary conditions at r_1 and r_2 are imposed by the procedure derived in the previous section, except for the density at the exterior cylinder r_2 which is not determined locally but by the same interpolation/extrapolation procedure as for the velocity, cf. Section 3.

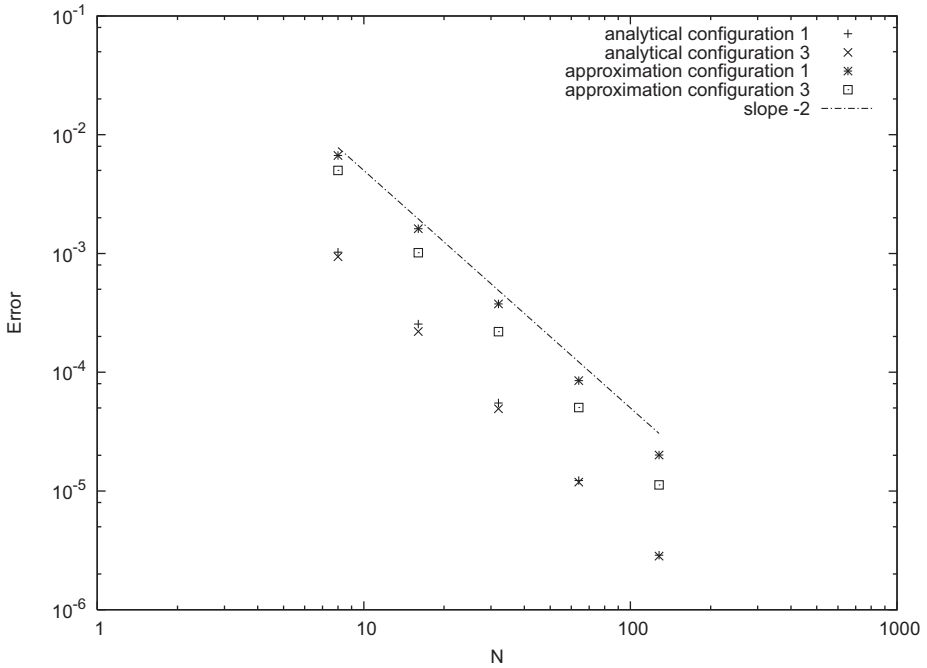

Fig. 7. Error of the velocity field for the Taylor–Couette flow experiment using both the analytical solution for ρ, $\vec{u}$ and **S** and the numerical approximation, cf. Section 3, on different configurations of the boundary nodes.

The first question we want to find an answer to is, which of the configurations presented in Section 3 will give the most accurate results. Since we are using interpolation/extrapolation for the velocity, we might expect configuration 3 to be the most accurate one, since it minimizes the distances to the wall.

Second, what are the actual error contributions of our approximation schemes for ρ, $\vec{u}$ and **S**? For this benchmark test an analytical solution exists, such that we can use the exact values of ρ, $\vec{u}$ and **S** for formulae (5) and (15) to reconstruct the populations at the boundary nodes. This gives us the possibility to measure the error contributions of the approximation schemes derived in Section 3.

Finally, the third question concerns the stability properties of the present curved no-slip boundary condition. It is well known [19] that the regularized Boltzmann method reveals increased stability. But what about the influence of the interpolation and extrapolation schemes?

4.2.1. Results

In order to answer to the first question, we performed simulations first using the analytical solution for ρ, $\vec{u}$ and **S** injected into formulae (5) and (15) on configuration 1 and configuration 3, and, second, using the approximation schemes derived in Section 3 with the density approximated by (29) on configuration 1 and configuration 3. The relaxation frequency was chosen to be $\omega = 1.66667$. The error of the velocity field, computed by formula (40),[1] for different resolutions is displayed in Fig. 7, the corresponding error of the pressure field in Fig. 8. Concerning the velocity field, we see all simulations are of second order accuracy both with the analytical values of ρ, $\vec{u}$ and **S**, and with the computed values by the numerical approximation schemes, cf. Section 3, and as well on configuration 1 as on configuration 3. In addition, when using the analytical results the difference in accuracy is marginal between configuration 1 and configuration 3. However, when approximating ρ, $\vec{u}$ and **S** numerically by the schemes derived in Section 3, we see a difference in accuracy. As presumed above, the simulation on configuration 3 has a lower error constant c, around two times smaller than the simulation on configuration 1. The pressure field, in contrast, is not resolved with second order accuracy, cf. Fig. 8. When using Eqs. (5), (15) and (16), we solve the pressure field only with first order accuracy even with the exact solutions for ρ, $\vec{u}$ and **S**. Since an interpolation or extrapolation cannot be better than the exact solution, the simulations using the approximation schemes, Section 3, inherit the first order accuracy of the pressure field. For coarse resolutions, however, we see an error decrease faster than first order for the simulations using the approximation schemes. This might have its origin in the fact that the error contribution of the approximation schemes to the pressure field decreases with higher than first order and that for finer resolutions the error is dominated by the first order

[1] The ensemble of nodes N_{err} is here defined to be all the nodes between the interior and exterior cylinder, not the entire square domain. The number of nodes is thus not given by $16N^2$. It needs to be computed for each resolution but it is approximately given by the surface in-between the cylinders $3\pi N^2$. Depending on which number is used for Eq. (40), the error constant will differ.

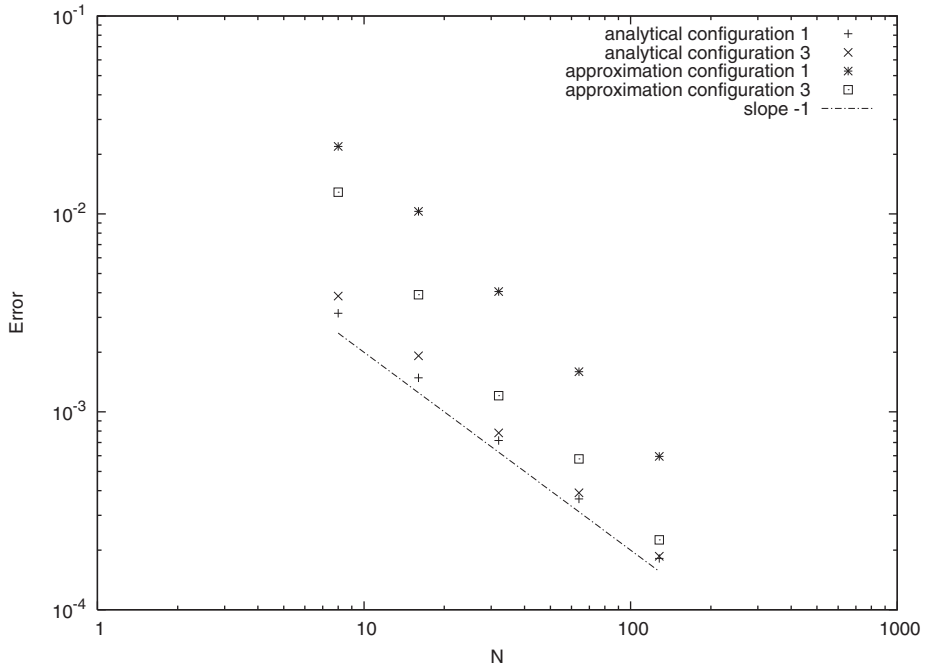

Fig. 8. Error of the pressure field for the Taylor–Couette flow experiment using both the analytical solution for ρ, $\vec{u}$ and **S** and the numerical approximation, cf. Section 3, on different configurations of the boundary nodes.

error contribution of the reconstruction, formulae (5) and (15). Also for the pressure, the simulation on configuration 3 is more accurate than on configuration 1, when using the approximation schemes for ρ, $\vec{u}$ and **S**, with Eq. (29) for ρ, cf. Fig. 8.

The results in Figs. 7 and 8 give also an answer to our second question concerning the error contribution of the present approximation schemes. The present approximation schemes do not deteriorate the second order accuracy of the velocity field and the first order accuracy of the pressure field for this benchmark test. However, they introduce error contributions likewise decreasing with second order accuracy, cf. Fig. 7. The error contribution by the approximation schemes, with Eq. (29) for ρ, concerning the velocity field, can thus be estimated by an increased error coefficient, by 8 times for the simulation on configuration 1 and by 4 times for the simulation on configuration 3.

In order to investigate the behavior of the error on the velocity and pressure field for the different approximation formulae of ρ, Eqs. (27)–(29) and (35), we performed a series of simulations using the approximation schemes for $\vec{u}$ and **S** on configuration 3, cf. Figs. 9 and 10. Clearly, the simulation using (35) for ρ fails to converge for the velocity and for the pressure field. As mentioned above, imposing (30) on a boundary node might lead to unphysical conditions for a curved wall. When using Eqs. (27)–(29) for ρ, we observe that the error for the velocity field is comparable, cf. Fig. 9. However, for the pressure field we see that the approximation of ρ by Eq. (29), gives the most accurate results, cf. Fig. 10. This can be explained by the fact that Eq. (29) coincides with the definition of ρ for a bulk fluid node, opposed to Eqs. (27) and (28), and gives therefore a better approximation of ρ when the ensemble of populations used to compute ρ varies from node to node.

If numerical instabilities occur in numerical simulations, they often originate at the boundaries. Therefore, stability of boundary conditions is a crucial issue in CFD. Stability for LBGK is primarily determined by the value of the relaxation frequency ω. For a value of ω close to two, simulations tend in general to become unstable. The present benchmark test is a good opportunity to test the present no-slip boundary condition, since no additional boundary conditions at which instabilities could occur are used. We performed four series of simulations with four different relaxation frequencies, $\omega_0 = 1.66667$, $\omega_1 = 1.96078$, $\omega_2 = 1.99601$ and $\omega_3 = 1.99960$ using the present approximation schemes on configuration 3. In order to keep the Reynolds number constant at 10, the time step has been decreased accordingly. The results are displayed in Fig. 11. For higher ω, we did not compute the error for all possible resolutions, since the time step on finer grids had to be decreased according to Eq. (42), leading to total simulation times exceeding our computational resources. However, this does not affect the following conclusions about the stability properties of the method. For all tested values of ω, we see that the simulations stay stable. This is not surprising, since the interpolation and extrapolation schemes do not depend on ω. Therefore, we can say that the present curved no-slip boundary condition is stable for relaxation frequencies close to two. From Fig. 11, we observe more accurate results for increasing ω, this might be due to a reduction of the error contributions by the time discretization and the compressibility effects, since we were using smaller time steps in order to keep the Reynolds number fixed.

The geometry, the relaxation frequency and the Reynolds number used for the simulations have been chosen to exactly match the ones of Guo et al. in [12]. We performed therefore the same numerical experiment using their boundary condition.

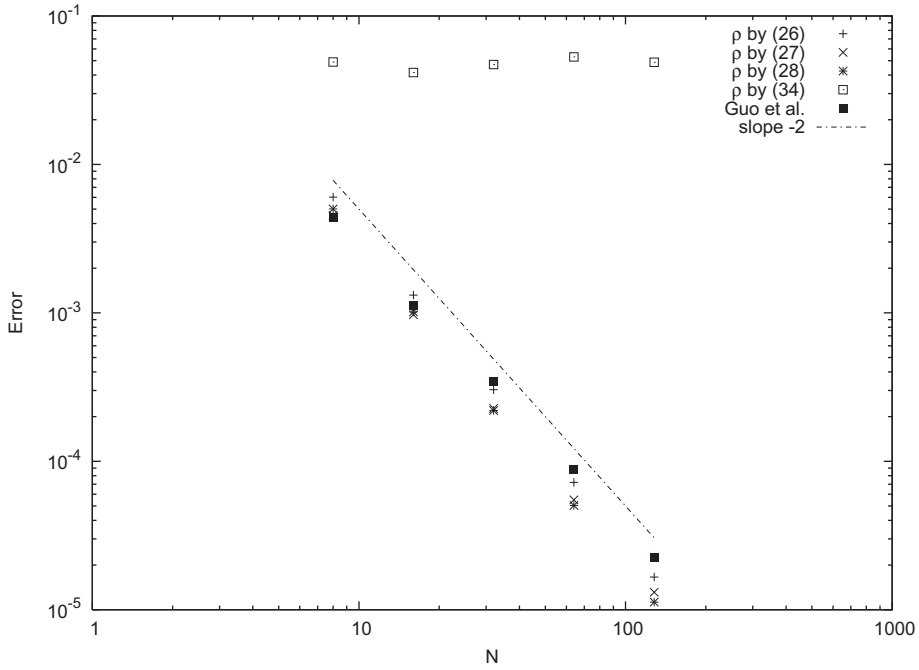

Fig. 9. Error of the velocity field for the Taylor–Couette flow experiment. Comparison of simulations using different approximations of ρ, Eqs. (27)–(29) and (35) combined with the numerical approximations of $\vec{u}$ and **S** on configuration 3, in addition to a simulation using the method of Guo et al. [12].

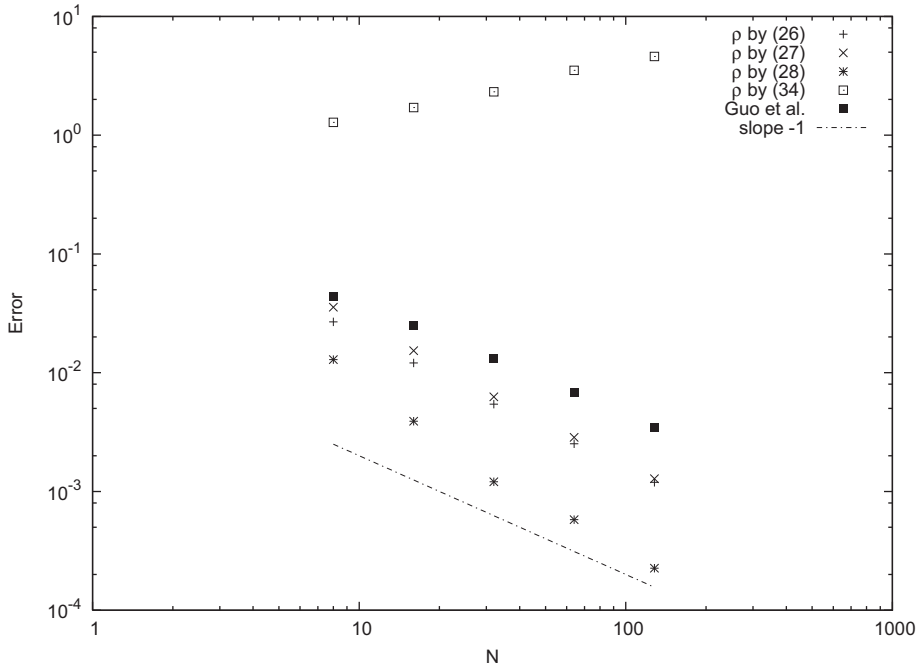

Fig. 10. Error of the pressure field for the Taylor–Couette flow experiment. Comparison of simulations using different approximations of ρ, Eqs. (27)–(29) and (35) combined with the numerical approximations of $\vec{u}$ and **S** on configuration 3, in addition to a simulation using the method of Guo et al. [12].

In order to fix the pressure at the outer cylinder wall we used the analytical value of ρ instead of using the value of ρ from the neighboring bulk fluid nodes at the outer wall. We checked that its effect on the velocity field is negligible compared to a simulation using ρ from the neighboring bulk fluid nodes. The error on the velocity and pressure field can be seen in Figs. 9 and 10, showing that the error on the velocity field is comparable to the error using the present boundary condition, whereas

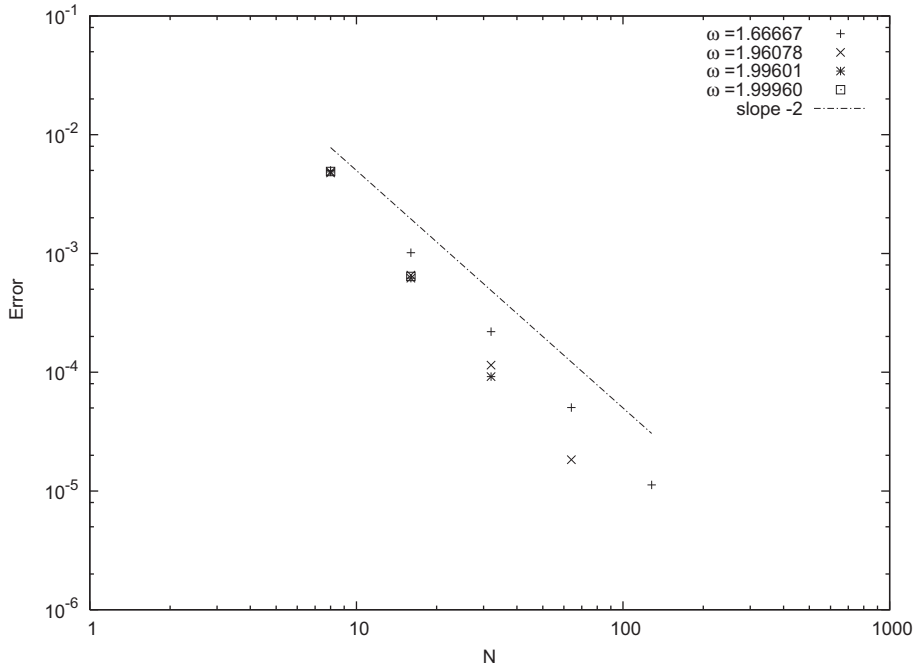

Fig. 11. Error of the velocity field for the Taylor–Couette flow experiment using the numerical approximation of ρ, $\vec{u}$ and **S**, cf. Section 3, on configuration 3 for different values of the relaxation frequency ω.

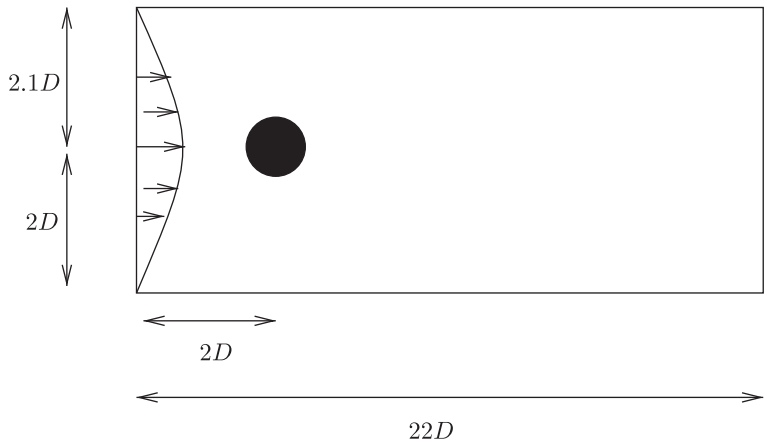

Fig. 12. Geometry of the benchmark test case in Section 4.3.

the error on the pressure field using the boundary condition of Guo et al. [12] is larger than the error using the present boundary condition.

In the following, all simulations use Eq. (29) to approximate ρ at the boundary nodes, since it is the most accurate one.

4.3. Verification part 2

The latter verification proved the second order accuracy of the velocity field for a simple steady flow calculation. We need, however, to verify the present boundary condition also for more complex phenomena related to no-slip boundary conditions, like stagnation and separation points. To this end, the following benchmark test has been chosen from the Priority Research Program workshop *Benchmark Computations of Laminar Flow Around a Cylinder* [27]. It consists of two parts: steady laminar flow around a circular cylinder at a Reynolds number of Re = 20 and time-periodic flow around the same cylinder at a Reynolds number of Re = 100. The cylinder is confined in-between channel walls, as can be seen from Fig. 12. It is placed somewhat off the centerline in order to generate an asymmetric flow field. For both cases, a parabolic inflow condition is

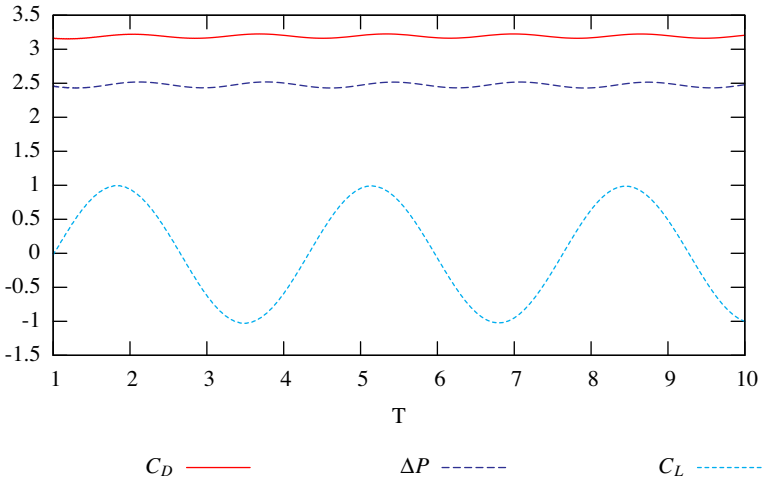

Fig. 13. History of the pressure difference ΔP, the drag coefficient C_D and the lift coefficient C_L for the time-periodic flow around a circular cylinder at Re = 100.

Table 2
Values of the drag coefficient C_D, the lift coefficient C_L, the recirculation length L_a and the pressure difference ΔP for the flow around a circular cylinder at Re = 20 for different resolutions N. Values are displayed for the present boundary condition and the boundary condition of [9]. The lower and upper bounds determined by [27] are given as indication for the tolerance intervals.

	N	C_D	C_L	L_a	ΔP
Lower/upper bound		5.5700/5.5900	0.0104/0.0110	0.0842/0.0852	0.1172/0.1176
Present	10	5.90759	0.00859	0.08574	0.11482
Present	20	5.63092	0.00931	0.08680	0.11188
Present	40	5.59373	0.01036	0.08495	0.11728
Present	60	5.58671	0.01049	0.08474	0.11698
Fillipova and Hänel [9]	40	5.52	0.011	0.08	0.116

imposed and the outflow is a constant pressure outflow boundary. The Reynolds number is based on the mean velocity U of the inflow and the diameter D of the cylinder. We fix $D = 1$ and $U = 1$ and change only the viscosity in order to vary the Reynolds number. For a Reynolds number of 20, the flow is still steady and develops a recirculation bubble behind the cylinder, whereas for Re = 100, the flow is in fact laminar but unsteady. It separates at the cylinder and causes a periodic shedding of vortices behind the cylinder, the so-called von Kármán vortex street. The points at which the flow separates at the cylinder exhibit a periodic movement on the cylinder walls. However, instead of directly measuring them, we focus on the integral quantities defined below, which were determined by the organizers of the workshop [27]. These quantities depend directly on the accurate computation of the boundary layer. Therefore, they allow a quantitative evaluation of the numerical method. For the same reason, we renounce to give contour plots of any quantity, like vorticity, in the computational domain, since these plots do not give any new insights on the numerical method. As a result of this workshop, upper and lower bounds have been formulated for the quantities of interest, meaning that the grid independent solution should be inside the tolerance interval defined by these upper and lower bounds. The quantities of interest are for the Re = 20 case: Drag coefficient C_D, lift coefficient C_L, length of recirculation zone L_a and the pressure difference ΔP between the front point $(1.5D, 2D)$ and the end point $(2.5D, 2D)$. For the Re = 100 case, the following quantities should be computed: maximum drag coefficient C_{Dmax}, maximum lift coefficient C_{Lmax}, Strouhal number St and the above pressure difference $\Delta P(t^*)$ at t^*, where t^* is the point of time in the middle of two maxima of the lift coefficient. Fig. 13 shows the periodic evolution in time of the pressure difference ΔP, the drag coefficient C_D and the lift coefficient $C_L$for the Re = 100 case.

This benchmark test has been computed by a number of different groups, all using different numerical methods, among others LBGK by Fillipova and Hänel in connection with their boundary fitting method presented in [9].[2]

4.3.1. Results

For both cases, Re = 20 ($\omega = 1.84502$) and Re = 100 ($\omega = 1.96696$), we performed simulations with four different resolutions $N = 10, 20, 40, 60$. At the channel walls, the stable version of the no-slip boundary conditions presented in [32] has been used. The inflow and outflow boundary have been imposed using Lätt's reconstruction formulae (5), (15) and (16), where we

[2] Data for comparisons has been taken from their more recent article [9] and not from the workshop results [27].

have used the analytical Poiseuille profile for $\vec{u}$ and the analytical rate of strain tensor **S** for both, the inflow and the outflow boundary condition. The density at the outflow has been fixed to 1 and at the inflow the density was determined using formula (29). Thereby, we avoided spurious reflecting waves coming from the outflow. Simulations with the constant pressure boundary condition of Lätt et al. [20] at the outflow proved that the effect of the Poiseuille velocity profile at the outflow is negligible on the flow in vicinity of the cylinder.

The computed values for the above quantities are displayed in Table 2 (Re = 20) and Table 3 (Re = 100). An error, formula (41), has been determined for these quantities by using the arithmetic means of the lower and upper bounds [27] as reference values. Although the midpoint of the tolerance interval determined by the lower and upper bounds is a rather crude approximation of the reference value, the error decrease of each quantity as a function of the resolution can clearly be illustrated, cf. Fig. 14 (Re = 20) and Fig. 15 (Re = 100).

From Tables 2 and 3, we see that for a resolution of $N = 60$ all quantities, apart from the pressure difference ΔP for the Re = 20 case, have converged, meaning that they are in the tolerance interval defined by the lower and upper bounds. The pressure difference ΔP for Re = 20 is only marginally smaller than the lower bound. From the previous benchmark test, Section 4.2, we know that the pressure is critical since it is only converging with first order accuracy. Nevertheless another reason might be that the tolerance for the pressure difference for the Re = 20 case is much stricter than for the Re = 100 case, i.e. 0.3% and 1.6%, respectively. This might explain also why the pressure difference converged for the Re = 100 case, whereas it did not for the Re = 20 case.

Table 3
Values of the maximum of the drag coefficient C_{Dmax}, the maximum of the lift coefficient C_{Lmax}, the Strouhal number Stand the pressure difference ΔP for the time-periodic flow around a circular cylinder at Re = 100 for different resolutions N. Values are displayed for the present boundary condition and the boundary condition of [9]. The lower and upper bounds determined by [27] are given as indications of the tolerance intervals.

	N	C_{Dmax}	C_{Lmax}	St	ΔP
Lower/upper bound		3.2200/3.2400	0.9900/1.0100	0.2950/0.3050	2.4600/2.500
Present	10	3.57813	0.92627	0.29075	2.48703
Present	20	3.24817	1.03697	0.30560	2.40584
Present	40	3.22256	0.98572	0.30238	2.44601
Present	60	3.23284	0.99421	0.30263	2.47727
Fillipova and Hänel [9]	60	–	–	0.3	–

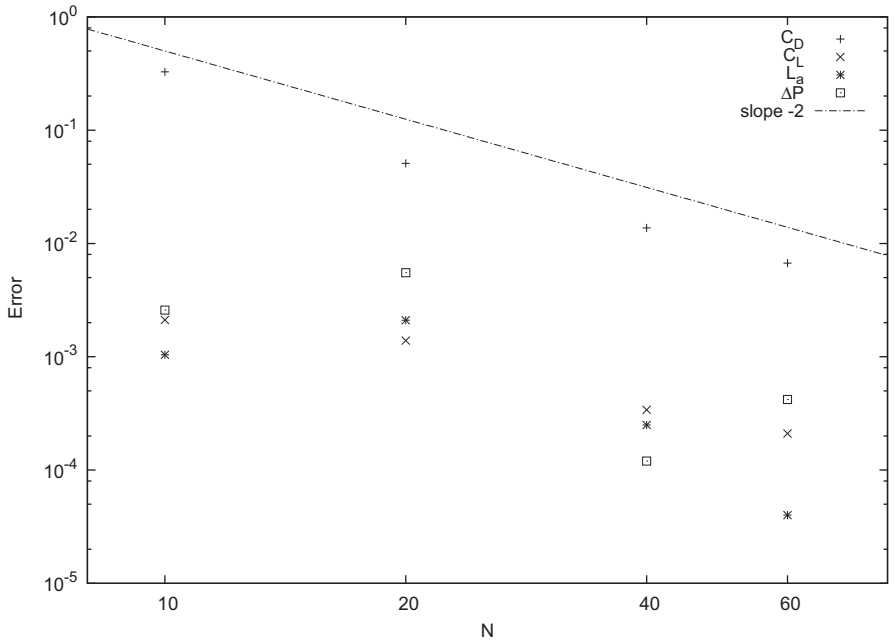

Fig. 14. Error of the drag coefficient C_D, the lift coefficient C_L, the recirculation length L_a and the pressure difference ΔP for the steady laminar flow around a circular cylinder at Re = 20. For the reference values we take the arithmetic means of the tolerance intervals, i.e. $C_{Dref} = 5.58$, $C_{Lref} = 0.0107$, $L_{aref} = 0.0847$ and $\Delta P_{ref} = 0.1174$.

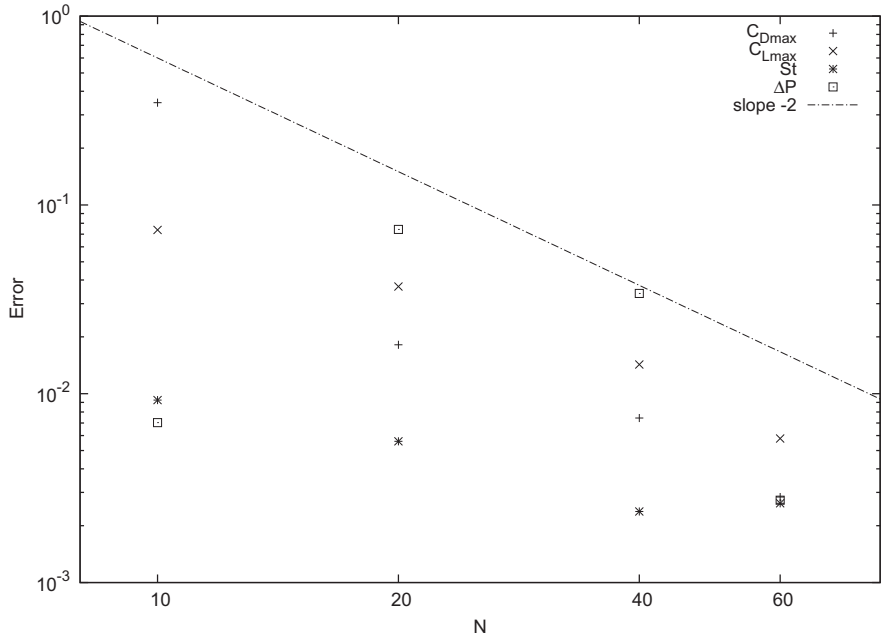

Fig. 15. Error of the maximum value of the drag coefficient C_{Dmax}, the maximum value of the lift coefficient C_{Lmax}, the Strouhal number St and the pressure difference ΔP for the time-periodic flow around a circular cylinder at Re = 100. For the reference values we take the arithmetic means of the tolerance intervals, i.e. $C_{Dref} = 3.23$, $C_{Lref} = 1.0$, $St_{ref} = 0.3$ and $\Delta P_{ref} = 2.48$.

For the Re = 20 case, cf. Fig. 14, the decrease of the error of the drag and lift coefficient seems to follow the second order slope. However, this is not true for the error of the other quantities, i.e., length of the recirculation zone and pressure difference. From Fig. 15, it is difficult to say that the reduction of the error for the Re = 100 case is second order. This might be due to the fact that the quantities are already too close to the reference values. Although we cannot tell from Figs. 14 and 15 whether the computed values converge strictly with second order accuracy, they show a solid reduction of the numerical error (apart from the pressure) verifying the present curved boundary treatment for this benchmark test by producing results in the expected ranges, cf. Tables 2 and 3.

This benchmark test was computed by Fillipova and Hänel with their boundary fitting method [9]. Their computed values are displayed in Tables 2 and 3. Although they do not present all values and in addition use coarser grids in the domain further away from the cylinder, the present curved boundary treatment seems to be more accurate when comparing the values for $N = 40$ of the present boundary treatment and theirs for Re = 20, cf. Table 2. Unfortunately, Mei et al. [22] do not solve this benchmark test, such that a comparison with the improved version of Fillipova and Hänel's boundary fitting method is still an open issue.

4.4. Verification part 3

The geometry of this benchmark test case, the flow around an impulsively started circular cylinder, consists of a rectangular domain where the upper and lower boundaries are chosen to be periodic and the velocity profile at the inlet is constant instead of parabolic. The outflow is a constant pressure boundary. The circular cylinder is put on the center line when compared to the benchmark test above, cf. Section 4.3. Opposed to the above case we are not interested in steady or time-periodic solutions but focus on an initial-boundary value problem. The Reynolds number in this case is based on the velocity of the cylinder U, the diameter D and the kinematic viscosity of the fluid. The characteristic time, however, is defined to be $D/(2U)$ [17]. Instead of moving the cylinder, the reference frame is placed on the moving cylinder, such that the cylinder is at rest but the flow is now moving towards the cylinder with a characteristic velocity U. This shifts the problem of formulating boundary conditions with an arbitrary movement (which is not treated in the present discussion due to its major difficulties [30]) to the problem of finding an adequate initial condition to simulate the starting cylinder. Most often the velocity field surrounding the cylinder is initialized with the potential flow solution [17]:

$$u_r = U\left(1 - \frac{D^2}{4r^2}\right)\cos\theta \quad u_\theta = -U\left(1 + \frac{D^2}{4r^2}\right)\sin\theta, \tag{43}$$

and the viscosity is switched on at $T = 0$. Integral and other quantities, such as drag and recirculation length, are then monitored for specific times. This benchmark test case was also solved by Li et al. [21] and Dupuis et al. [7] to verify their curved

boundary treatments for LBGK, cf. Section 1. Dupuis et al. [7] derived two different versions of their method, the direct forcing approach and the interpolating force approach, but we plot only results for the latter one for comparison, since it seemed to give more accurate results.

4.4.1. Results

We considered two cases, the flows at Reynolds numbers of Re = 40 and of Re = 200, also solved by other groups [17,21,7] to compare our results with those.

Since the lattice Boltzmann method solves the LBGK equation for a particle distribution function, we used Lätt's procedure [19] to initialize the populations f_i using the macroscopic quantities p, $\vec{u}$ and $\mathbf{S}$. Thus, we computed the pressure p via Bernoulli's equation and the rate of strain tensor $\mathbf{S}$ via differentiation of the velocity field, Eq. (43). These quantities were then used to initialize the populations f_i by using Eqs. (5), (15) and (16).

For the Re = 40 case, the domain is a square with side length $100D$ and the cylinder is located at the center. This geometry was chosen in order to minimize the effect of the artificial boundaries and to match the geometry of Li et al. [21] and Dupuis et al. [7]. For the Re = 200 case, the domain is a rectangle with a side length $45D$ in x-direction and $20D$ in y-direction. The cylinder is placed at $(15D, 10D)$. The same geometry was used by Dupuis et al. [7] for this case.

For the Re = 40 case ($\omega = 1.69492$, $N = 20$), the evolution of the drag coefficient is displayed in Fig. 16, while the growth in time of the recirculation length can be seen in Fig. 17. Concerning the recirculation length, we observe a good agreement between the results obtained by Dupuis et al. [7], Li et al. [21] and the present curved boundary treatment. However, for the drag coefficient discrepancies are observable for early times $T < 1$. Especially for very early times $T < 0.08$, we observe spurious oscillations of the drag coefficient for the present method, cf. Fig. 16. Unfortunately, Li et al. [21] do not present data for the drag coefficient for times $T < 0.08$. Dupuis et al.'s interpolating force approach does not display such oscillations. Both Li et al. and Dupuis et al. do not explicitly mention the appearance of such oscillations for $T < 0.08$ [7,21]. For $0.08 < T < 1$ the evolution of the drag coefficient is slightly different for all methods presented in Fig. 16, while for times $T > 1$ the evolution of the drag coefficient agrees well for all lattice Boltzmann methods. Koumoutsakos and Leonard, who did not use a lattice Boltzmann method but instead solved the incompressible Navier–Stokes equation with a vortex method [17], obtained a slightly different evolution of the drag coefficient for $T > 1$, cf. Fig. 16. Li et al. showed that for times $T < 1$ the physics is very sensitive to the Mach number of the flow [21]. In addition, one might expect that the simulation is very dependent on the spatial and temporal discretization, the procedure of imposing the initial condition and last but not least the boundary treatment used for the cylinder, since the problem contains a discontinuity at $T = 0$, when the viscosity is suddenly switched on. This is also supported by the fact that for five different methods (including the direct forcing approach of Dupuis et al. [7]), five different evolutions of the drag coefficient are obtained for $T < 1$. Therefore, it is difficult to draw any conclusions about the quality of the curved boundary treatment based on the evolution of the drag coefficient for $T < 1$. However, it seems that Dupuis et al.'s interpolating force approach gives the smoothest evolution of the drag coefficient for early times, apart from the method of Koumoutsakos and Leonard which is based on the incompressible Navier–Stokes equations and therefore not prone to compressibility effects.

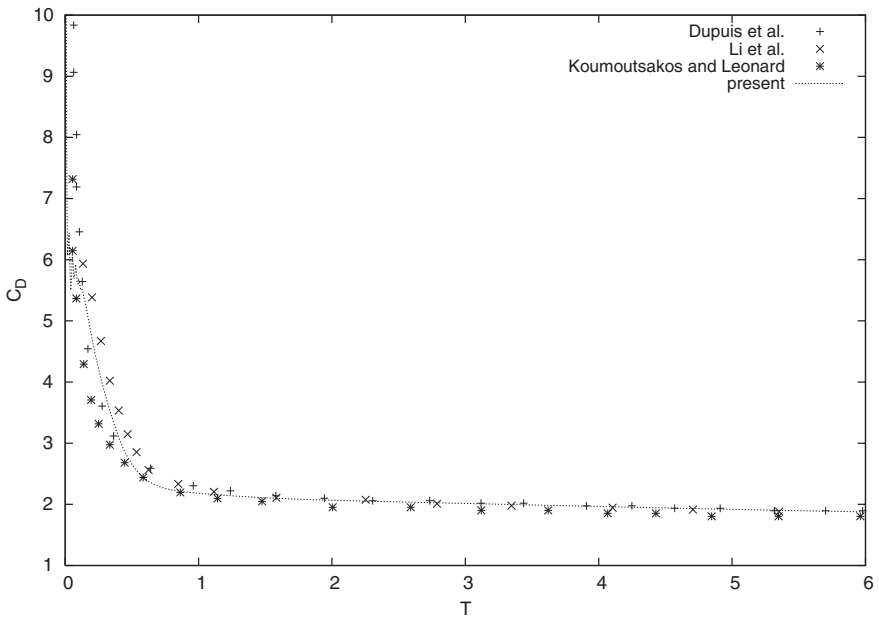

Fig. 16. Evolution of the drag coefficient C_D for the impulsively started cylinder at Re = 40. The present results are compared to the results of Koumoutsakos and Leonard [17], Dupuis et al. [7] and Li et al. [21].

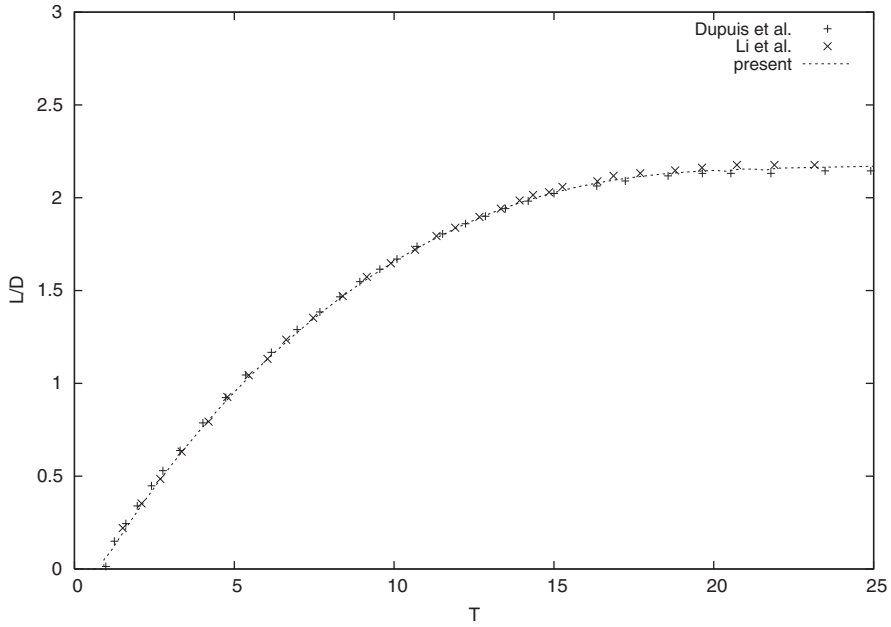

Fig. 17. Evolution of the recirculation length for the impulsively started cylinder at Re = 40. The present results are compared to the results of Dupuis et al. [7] and Li et al. [21].

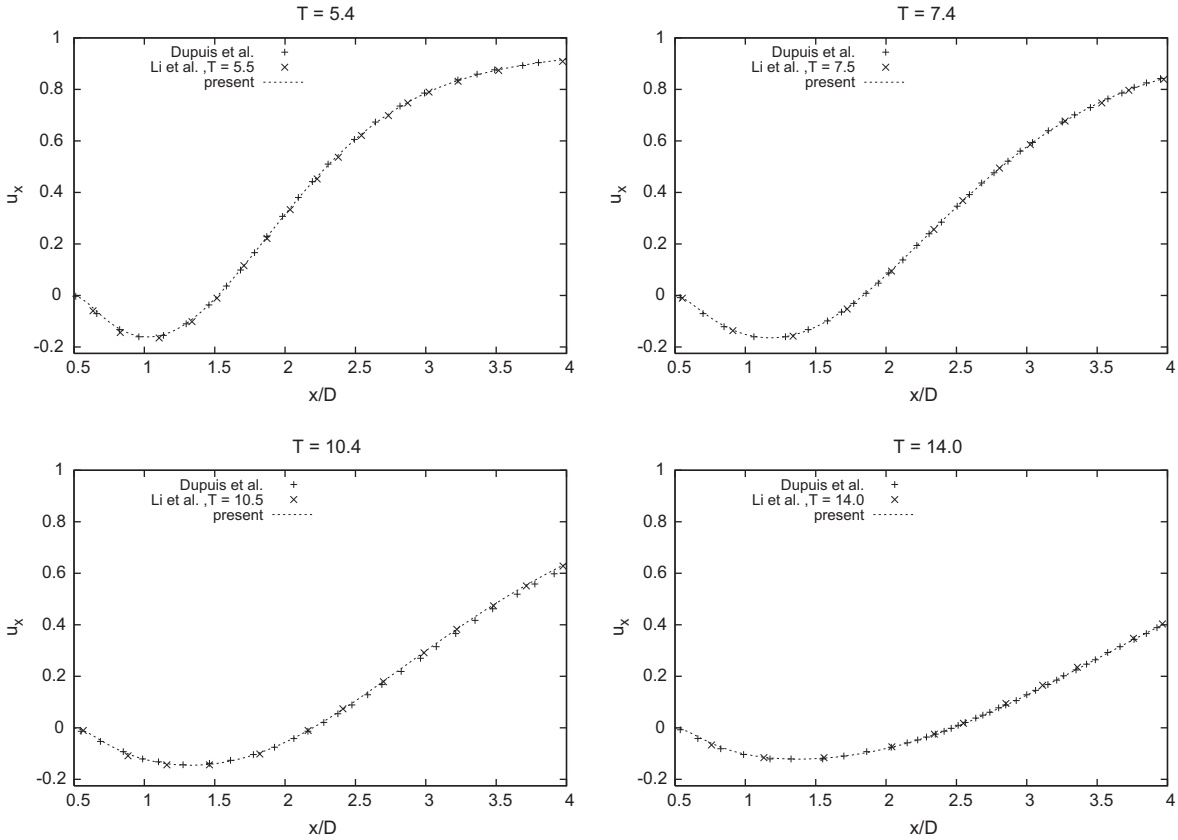

Fig. 18. Instantaneous velocity profiles of the *x*-component behind the impulsively started cylinder at Re = 40 for four different times $T = 5.4, 7.4, 10.4, 14.0$. The present results are compared to the results of Dupuis et al. [7] and Li et al. [21].

Since the simulations are difficult to compare to each other for early times, $T < 1$, this benchmark test includes comparisons of the x-velocity profile along the x-axis behind the cylinder for $T = 5.4, 7.4, 10.4, 14.0$. These profiles are displayed in Fig. 18. Although Li et al. presented their profiles for slightly different times, the profiles computed by all three methods agree very well for all four points in time.

Concerning the resolution, we point out that Li et al. used 40 points along the diameter, whereas Dupuis et al. and the present simulations resolved this length with 20 points.

The second part of this benchmark test, the impulsively started cylinder at Re = 200 ($\omega = 1.86567$, $N = 40$), aims at checking whether the flow around the stagnation point is qualitatively correct by the simulations. As already mentioned, the geometry of the present simulation is the same as for Dupuis et al.'s simulation and instantaneous vorticity contour plots were taken at the same points in time $T = 1, 3, 5$ as by Dupuis et al. [7]. These contour plots are displayed in Fig. 19. For all three points in time, we observe that the contour lines are correctly deflected near the stagnation point at the front end of the cylinder. When leaving the cylinder at the front and bending towards the rear, these contour lines present tiny

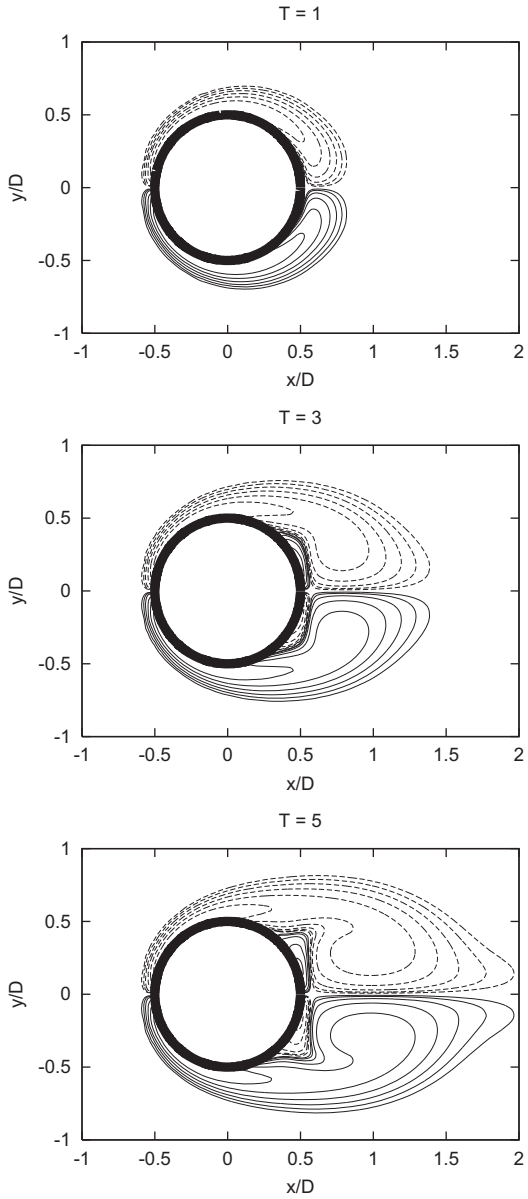

Fig. 19. Instantaneous vorticity contours for the impulsively started cylinder at Re = 200 for three different times $T = 1, 3, 5$ using the present curved boundary treatment. The contour lines are at $\pm 0.25, 0.5, 1, 2, 4, 8$ with the negative contours being dashed.

oscillations. We checked the velocity field used to compute the vorticity field in this region but could not observe any anomalies. Therefore, we tend to believe that these tiny oscillations are due to the finite differencing of the velocity field, since because of the geometry we are constantly switching stencils in order to compute the vorticity. In general the present contour lines are very smooth in this critical region in comparison with Dupuis et al.'s interpolating force approach, cf. Fig. 19 and Fig. 7 in [7]. Apart from this, the evolution of the vorticity field around the cylinder with the present curved boundary treatment agrees very well with the results of Dupuis et al. for the direct forcing approach and the interpolating force approach, cf. Fig. 19 and Fig. 7 in [7].

5. Conclusions

We derived a simple curved boundary treatment of the no-slip condition at walls for the lattice Boltzmann method. This treatment is based on the well founded reconstruction formalism by Lätt [19] and can therefore be understood as a generalization of his finite difference boundary condition for straight walls [20]. The main error contribution could thus be identified to stem from the numerical approximation schemes to compute ρ, $\vec{u}$ and $\mathbf{S}$ which we derived in Section 3. We could improve these schemes by a special choice of the boundary nodes. The verification of the present curved boundary treatment was done in three steps. First, we tested the present boundary condition for Taylor-Couette flow, for which an analytical solution exists, and proved its second order accuracy for the velocity field. Second, by means of the steady and unsteady flow around a circular cylinder, we verified that we could reach accurate results with the present boundary treatment. The third test, the flow around an impulsively started circular cylinder, allowed us to make comparisons to other curved boundary treatments for LBGK. Apart from early times where the solutions in the literature deviate from each other, good agreement with all reference solutions was found at later times.

The present curved boundary treatment reveals increased accuracy and is stable for relaxation frequencies close to two.

The reconstruction formalism by Lätt [19] allows for more general boundary conditions such as inlets or outlets. However, an application of the present curved boundary treatment to inlet or outlet boundary conditions still needs to be verified. Although we did not assume the system to be two-dimensional for the derivation of the present curved boundary treatment in Section 3, a thorough verification of the three-dimensional case is still necessary.

Acknowledgments

The coding has been done using the open source lattice Boltzmann library OpenLB [31].

References

[1] M. Bouzidi, M. Firdaouss, P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, Physics of Fluids 13 (11) (2001) 3452–3459.
[2] S. Chandrasekhar, Hydrodynamic and Hydromagnetic Stability, Dover Publications, Inc., New York, 1961.
[3] H. Chen, C. Teixeira, K. Molving, Realization of fluid boundary conditions via discrete Boltzmann dynamics, International Journal of Modern Physics C 9 (1998) 1281–1292.
[4] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Annual Review of Fluid Mechanics 30 (1998) 329–364.
[5] S. Chen, D. Martinez, R. Mei, On boundary conditions in lattice Boltzmann methods, Physics of Fluids 8 (9) (1996) 2517–2536.
[6] B. Chopard, A. Dupuis, A mass conserving boundary condition for lattice Boltzmann models, International Journal of Modern Physics B 17 (2003) 103–107.
[7] A. Dupuis, P. Chatelain, P. Koumoutsakos, An immersed boundary-lattice-Boltzmann method for the simulation of the flow past an impulsively started cylinder, Journal of Computational Physics 227 (2008) 4486–4498.
[8] Z.-G. Feng, E.E. Michaelides, The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems, Journal of Computational Physics 195 (2004) 602–628.
[9] O. Filippova, D. Hänel, Boundary fitting and local grid refinement for lattice-BGK models, International Journal of Modern Physics C 9 (1998) 1271–1279.
[10] I. Ginzbourg, P.M. Adler, Boundary flow condition analysis for the three-dimensional lattice Boltzmann model, Journal de Physique 4 (1994) 191–214.
[11] P.M. Gresho, R.L. Sani, On pressure boundary conditions for the incompressible Navier–Stokes equations, International Journal for Numerical Methods in Fluids 7 (1987) 1111–1145.
[12] Z. Guo, C. Zheng, B. Shi, An extrapolation method for boundary conditions in lattice Boltzmann method, Physics of Fluids 14 (6) (2002) 2007–2010.
[13] I. Halliday, L.A. Hammond, C.M. Care, Enhanced closure scheme for lattice Boltzmann equation hydrodynamics, Journal of Physics A 35 (2002) 157–166.
[14] D. Hänel, Einführung in die Kinetische Theorie der Gase und Lattice-Boltzmann-Methoden, Springer-Verlag, Berlin, 2004.
[15] M. Junk, A finite difference interpretation of the lattice Boltzmann method, Kaiserslautern University, 1999, preprint.
[16] M. Junk, Z. Yang, One-point boundary condition for the lattice Boltzmann method, Physical Review E 72 (6) (2005) 066701.
[17] P. Koumoutsakos, A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using vortex methods, Journal of Fluid Mechanics 296 (1995) 1–38.
[18] P. Lallemand, L.-S. Luo, Lattice Boltzmann method for moving boundaries, Journal of Computational Physics 184 (2003) 406–421.
[19] J. Lätt, Hydrodynamic Limit of Lattice Boltzmann Equations, Ph.D. thesis, Université de Genève, 2007.
[20] J. Lätt, B. Chopard, O. Malaspinas, M. Deville, A. Michler, Straight velocity boundaries in the lattice Boltzmann method, Physical Review E 77 (5) (2008) 056703.
[21] Y. Li, R. Shock, R. Zhang, H. Chen, Numerical study of flow past an impulsively started cylinder by the lattice-Boltzmann method, Journal of Fluid Mechanics 519 (2004) 273–300.
[22] R. Mei, L.-S. Luo, W. Shyy, An accurate curved boundary treatment in the lattice Boltzmann method, Journal of Computational Physics 155 (1999) 307–330.
[23] S.A. Orszag, M. Israeli, M.O. Deville, Boundary conditions for incompressible flows, Journal of Scientific Computing 1 (1) (1986) 75–111.
[24] C.S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (1977) 220.
[25] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recepies in C++, second ed., Cambridge University Press, 2002.

[26] A. Quarteroni, R. Sacco, F. Saleri, Numerische Mathmatik, vol. 2, Springer-Verlag, Berlin, Heidelberg, 2002.
[27] M. Schäfer, S. Turek, Notes on Numerical Fluid Mechanics, Chapter Benchmark Computations of Laminar Flow Around a Cylinder, Vieweg Verlag, Braunschweig, 1996.
[28] C. Shu, N. Liu, Y. Chew, A novel immersed boundary velocity correction-lattice Boltzmann method and its application to simulate flow past a circular cylinder, Journal of Computational Physics 226 (2007) 1607–1622.
[29] P.A. Skordos, Initial and boundary conditions for the lattice Boltzmann method, Physical Review E 48 (6) (1993) 4823–4842.
[30] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford Science Publications, 2001.
[31] The OpenLB project, 2006. <http://www.lbmethod.org/openlb/>.
[32] J.C.G. Verschaeve, Analysis of the lattice Boltzmann Bhatnager–Gross–Krook no-slip boundary condition: ways to improve accuracy and stability, Physical Review E 80 (2009) 036703.
[33] D.A. Wolf-Gladrow, Lattice-Gas Cellular Automata and Lattice Boltzmann Models, Springer, Berlin, 2000.