Straight velocity boundaries in the lattice Boltzmann method

Jonas Latt[*] and Bastien Chopard[†]
University of Geneva, Geneva, Switzerland

Orestis Malaspinas, Michel Deville, and Andreas Michler[‡]
Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

(Received 5 January 2008; revised manuscript received 1 April 2008; published 13 May 2008)

Various ways of implementing boundary conditions for the numerical solution of the Navier-Stokes equations by a lattice Boltzmann method are discussed. Five commonly adopted approaches are reviewed, analyzed, and compared, including local and nonlocal methods. The discussion is restricted to velocity Dirichlet boundary conditions, and to straight on-lattice boundaries which are aligned with the horizontal and vertical lattice directions. The boundary conditions are first inspected analytically by applying systematically the results of a multiscale analysis to boundary nodes. This procedure makes it possible to compare boundary conditions on an equal footing, although they were originally derived from very different principles. It is concluded that all five boundary conditions exhibit second-order accuracy, consistent with the accuracy of the lattice Boltzmann method. The five methods are then compared numerically for accuracy and stability through benchmarks of two-dimensional and three-dimensional flows. None of the methods is found to be throughout superior to the others. Instead, the choice of a best boundary condition depends on the flow geometry, and on the desired trade-off between accuracy and stability. From the findings of the benchmarks, the boundary conditions can be classified into two major groups. The first group comprehends boundary conditions that preserve the information streaming from the bulk into boundary nodes and complete the missing information through closure relations. Boundary conditions in this group are found to be exceptionally accurate at low Reynolds number. Boundary conditions of the second group replace all variables on boundary nodes by new values. They exhibit generally much better numerical stability and are therefore dedicated for use in high Reynolds number flows.

DOI: 10.1103/PhysRevE.77.056703 PACS number(s): 47.11.−j, 05.20.Dd

I. INTRODUCTION

The lattice Boltzmann (LB) method has established itself as a tool with growing acceptance for the numerical simulation of fluid flows. However, in spite of several important breakthroughs in the past decades, many aspects of the method have been analyzed only recently. A variety of radically different approaches is consequently found in the literature, and a general consensus on specific implementation issues is often lacking. Even such a common topic as the implementation of boundary conditions is not treated in a unified manner. This state of affairs raises numerous difficulties, from the choice of an appropriate boundary condition to the evaluation of the numerical error. A selection of commonly used boundary conditions is therefore reviewed under a common light and compared in the present paper. The review concentrates on the implementation of Dirichlet velocity boundary conditions, which prescribe a given velocity profile along boundaries of the numerical domain. Furthermore, only straight boundaries are considered that traverse the nodes of the numerical grid, and which are aligned with the main directions of the grid.

Lattice Boltzmann models do not directly simulate the evolution of the flow velocity. Instead, they calculate the dynamics of particle populations which stem from a microscopic description of the fluid. While the macroscopic pressure and velocity fields are easily calculated from the particle populations, the reverse procedure is more contrived. Thus, implementing a velocity condition on straight boundaries boils down to finding a way to translate from macroscopic flow variables to particle populations. This problem has been approached by authors from different viewpoints, some of them based on the kinetic theory of gases, and some of them on a hydrodynamic description of fluids.

Although the numerical scheme of the LB method is derived from microscopic physics, it is able to recover accurate solutions of the macroscopic Navier-Stokes equations. This can be shown in various ways through an asymptotic analysis, in which particle populations are formally related to macroscopic flow variables. An analysis of this type is however not always conducted in the literature for boundary conditions, and only little is known about their hydrodynamic limit. In the present paper, the boundary conditions are therefore inspected with the help of a Chapman-Enskog multiscale analysis (see, e.g., Refs. [1–3]), which relates the lattice Boltzmann model to the equations of a weakly compressible fluid. The results of this analysis are reproduced in Sec. II B. Other approaches could however also be considered for this study. The reader is, for example, pointed to the asymptotic analysis presented in Ref. [4], in which the LB model is directly related to the incompressible Navier-Stokes equations.

The paper starts with an overview of the widely used lattice Boltzmann Bhatnagar-Gross-Krook (BGK) method [1,2,5], which is abbreviated as LBGK. After this, the paper

[*] Current address: Mathematics Department, Tufts University.

[†] bastien.chopard@cui.unige.ch

[‡] Current address: Institute of Aerodynamics and Flow Technology, German Aerospace Center, Braunschweig, Germany.

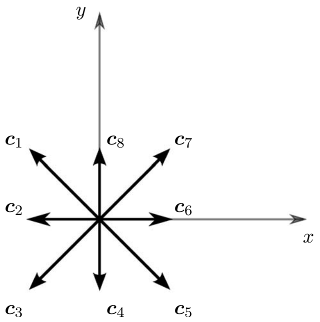

FIG. 1. Lattice vectors for D2Q9. An additional vector with zero components $\boldsymbol{c}_0=(0,0)$ is defined, which denotes a rest particle population.

introduces five different boundary conditions that can be found in the literature. Some well-known properties of the LB method are repeated to make this presentation accessible even to an audience less familiar with this model. A set of two-dimensional (2D) and three-dimensional (3D) benchmarks is finally described, which challenge the LBGK method with each of these boundary conditions as a numerical solver for the incompressible Navier-Stokes equations.

Aiming at a succinct comparison between different boundary conditions, the paper purposely avoids addressing advanced topics, such as the implementation of off-lattice boundaries, or the treatment of complex fluids. In particular, fluid compressibility effects are not analyzed in the benchmark section, although the LBGK method is able to simulate weakly compressible isothermal flows [2,3,16]. Terms related to fluid compressibility are however accounted for in the theoretical discussion of the boundary conditions.

II. THEORETICAL BACKGROUND

A. LBGK method

Lattice Boltzmann simulations are executed on a regular grid. The state of the system consists of q variables $f_i, i=0\cdots q-1$ on each grid node, called particle populations. The topology of a lattice is defined by q lattice vectors $\boldsymbol{c}_i, i=0\cdots q-1$. Thus, a node attributed to the space location $\boldsymbol{r}$ finds its neighbors at the location $\boldsymbol{r}+\boldsymbol{c}_i$. Figure 1 displays these vectors for the D2Q9 lattice, a common candidate for the implementation of 2D simulations [3,6].

The hydrodynamic variables on each node are defined as moments of the particle populations. The particle density ρ stands for the moment of order 0, and the fluid momentum $\rho\boldsymbol{u}$ for the moment of order 1,

$$\rho = \sum_{i=0}^{q-1} f_i \tag{1}$$

and

$$\rho\boldsymbol{u} = \sum_{i=0}^{q-1} \boldsymbol{c}_i f_i. \tag{2}$$

The moment of order 2 yields the tensor $\boldsymbol{\Pi}$, and the moment of order 3 yields the tensor $\boldsymbol{R}$,

$$\boldsymbol{\Pi} = \sum_{i=0}^{q-1} \boldsymbol{c}_i\boldsymbol{c}_i f_i \tag{3}$$

and

$$\boldsymbol{R} = \sum_{i=0}^{q-1} \boldsymbol{c}_i\boldsymbol{c}_i\boldsymbol{c}_i f_i. \tag{4}$$

These moments can be related to hydrodynamic flow variables through a multiscale analysis. In particular, the tensor $\boldsymbol{\Pi}$ is proportional to the strain rate (see Sec. II B). Equations (3) and (4) use a tensor notation without explicit space indexes. In a d-dimensional space, Eq. (3) denotes, for example, d^2 scalar equations, which in full index notation are written as $\Pi_{\alpha\beta}=\Sigma_{i=0}^{q-1}c_{i\alpha}c_{i\beta}f_i$, for $\alpha,\beta=1\cdots d$.

These vectors and tensors, as well as all other expressions in this paper, are written in a system of lattice units. In this system, the time interval between two iteration steps and the lateral or horizontal spacing between two grid nodes are unity. The speed of sound c_s in incompressible fluids is a constant depending on the lattice. An important corollary of this is that the Mach number $\mathrm{Ma}=U/c_s$ of a fluid is proportional to a characteristic fluid velocity U, expressed in lattice units. LB models can be used in a low Mach-number regime to simulate incompressible flows. In that case, the pressure p of the fluid is related to the particle density through the equation of state for an ideal gas,

$$p = c_s^2\rho. \tag{5}$$

A LB iteration takes the system from a discrete time step t to $t+1$ and consists of two steps. The collision step evaluates first a collision operator Ω on each lattice node, and takes the particle populations f_i to their post-collision value f_i'. This operator is local to each node,

$$f_i'(\boldsymbol{r},t) = \Omega_i[f_0(\boldsymbol{r},t), f_1(\boldsymbol{r},t), \ldots, f_{q-1}(\boldsymbol{r},t)]. \tag{6}$$

The collision is followed by a streaming step, which takes the post-collision variables to a neighbor node determined by the corresponding lattice vector:

$$f_i(\boldsymbol{r}+\boldsymbol{c}_i, t+1) = f_i'(\boldsymbol{r},t). \tag{7}$$

LB models possess exact mass and momentum conservation laws which are not subject to numerical approximation. This is expressed by the fact that the precollision populations f_i and their post-collision counterparts f_i' yield the same mass and momentum. Thus,

$$\sum_i \Omega_i = 0 \tag{8}$$

and

$$\sum_i \boldsymbol{c}_i\Omega_i = 0. \tag{9}$$

The common LBGK model [1,2,5] approximates the collision operator as a relaxation of the particle populations to a local equilibrium distribution f_i^{eq}, which depends only on the value of ρ and $\boldsymbol{u}$,

$$\Omega_i = -\omega[f_i - f_i^{\mathrm{eq}}(\rho,\boldsymbol{u})]. \tag{10}$$

The parameter ω determines the inverse time scale of the relaxation process. Through the multiscale analysis in the next section, this parameter can be formally related to the kinematic shear viscosity ν of the fluid,

$$\nu = c_s^2\left(\frac{1}{\omega} - \frac{1}{2}\right). \tag{11}$$

The local equilibrium depends only on the macroscopic variables ρ and $\boldsymbol{u}$. It can be viewed as a truncated expansion of the Maxwell-Boltzmann distribution and is written as

$$f_i^{\mathrm{eq}}(\rho,\boldsymbol{u}) = \left(1 + \frac{1}{c_s^2}\boldsymbol{c}_i\cdot\boldsymbol{u} + \frac{1}{2c_s^4}\boldsymbol{Q}_i{:}\boldsymbol{uu}\right). \tag{12}$$

The scalar product between two vectors is denoted by a single dot $(\cdot)$, and the contraction between two tensors by a colon $(:)$. The tensors $\boldsymbol{Q}_i$ are defined as $\boldsymbol{Q}_i = \boldsymbol{c}_i\boldsymbol{c}_i - c_s^2\boldsymbol{I}$ for $i = 0\cdots q-1$, where $\boldsymbol{I}$ is the identity. The scalars t_i are lattice weights that account for the varying lengths of the lattice vectors $\boldsymbol{c}_i$. Thus, a lattice is fully defined by means of the lattice vectors $\boldsymbol{c}_i$, the lattice weights t_i, and the speed of sound c_s. The benchmarks in this paper make use of the 2D lattice D2Q9 and the 3D lattice D3Q19. On the D2Q9 lattice, with the ordering of indexes used in Fig. 1, the lattice weights are $t_0 = 4/9$, $t_i = 1/9$ for $i \in \{2,4,6,8\}$ and $t_i = 1/36$ for $i \in \{1,3,5,7\}$. The speed of sound on this lattice is defined as $c_s^2 = 1/3$. The lattice-specific constants for the D3Q19 lattice can be found, for example, in Refs. [3,6].

A remarkable property of the equilibrium distribution is that it possesses the same momenta as the particle populations for the conserved variables ρ and $\rho\boldsymbol{u}$,

$$\rho = \sum_{i=0}^{q-1} f_i^{\mathrm{eq}} \tag{13}$$

and

$$\rho\boldsymbol{u} = \sum_{i=0}^{q-1} \boldsymbol{c}_i f_i^{\mathrm{eq}}. \tag{14}$$

B. Multiscale analysis

Through a multiscale Chapman-Enskog analysis of the LBGK model, it can be shown that the macroscopic variables defined in Eqs. (1) and (2) obey the Navier-Stokes equation for a weakly compressible fluid. Only the results of this analysis are shown here, and the reader is referred to Refs. [1–3] for more details. In this analysis, the LB dynamics is developed into a truncated Taylor series in space and time, up to second-order accuracy. Furthermore, to separate physical phenomena happening at different scales, the particle populations are expanded into a power-law series with respect to a small parameter $\varepsilon \ll 1$, $f_i = \sum_{k=0}^{\infty}\varepsilon^k f_i^k$. The two first terms, of order $O(\varepsilon^0)$ and order $O(\varepsilon^1)$, are sufficient to recover asymptotically the dynamics of the Navier-Stokes equation. Thus, the approximation

$$f_i = f_i^{(0)} + \varepsilon f_i^{(1)} + O(\varepsilon^2) \tag{15}$$

is made. The same approximation is applied to develop the time derivative over two scales,

$$\partial_t = \varepsilon\partial_{t1} + \varepsilon^2\partial_{t2} + O(\varepsilon^3). \tag{16}$$

Space derivatives are analyzed at a single scale,

$$\boldsymbol{\nabla} = \varepsilon\boldsymbol{\nabla}_1 + O(\varepsilon^2). \tag{17}$$

The parameter ε is used in analytical developments to formally separate time scales, but it is skipped in the remainder of this paper to enhance readability. The asymptotic and scale separated form of the mass and momentum conservation laws in Eqs. (9) and (8) can be written as follows:

$$\partial_t\rho + \boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}) = 0 \tag{18a}$$

and

$$\partial_t(\rho\boldsymbol{u}) + \boldsymbol{\nabla}\cdot\left(\boldsymbol{\Pi}^{(0)} + \boldsymbol{\Pi}^{(1)} + \frac{1}{2}(\partial_t\boldsymbol{\Pi}^{(0)} + \boldsymbol{\nabla}\cdot\boldsymbol{R}^{(0)})\right) = 0, \tag{18b}$$

where the operator $\boldsymbol{\nabla}$ denotes a divergence on the last index of the involved tensors. The tensors $\boldsymbol{\Pi}^{(0)}$, $\boldsymbol{\Pi}^{(1)}$, and $\boldsymbol{R}^{(0)}$ are velocity moments equivalent to those defined by Eqs. (3) and (4), but evaluated on the components $f_i^{(0)}$ and $f_i^{(1)}$ of the particle populations. The nature of the physics described by the momentum equation, Eq. (18b), depends on the form of these tensors. In the case of the LBGK model, the component $f_i^{(0)}$ of the particle populations is equal to the equilibrium distribution,

$$f_i^{(0)} = f_i^{\mathrm{eq}}(\rho,\boldsymbol{u}). \tag{19}$$

Assuming a low Mach-number regime, Eqs. (18a) and (18b) are then equivalent to the Navier-Stokes equations for a weakly compressible fluid with fixed bulk viscosity. The terms in the multiscale analysis are approximated with finite series of second order, which leads to the conclusion that the LB method is second-order accurate in space and time for the numerical simulation of weakly compressible fluids.

Although this is not the point of view adopted in the present paper, we also point the reader to an alternative theoretical framework in which the LB model is viewed as a discrete analog of the continuum Boltzmann equation with a BGK collision term [7]. This discretization is again second-order accurate, even though the explicit time stepping scheme of Eq. (7) seems to indicate first-order accuracy in time. As it is shown in Ref. [8], the streaming operator of the Boltzmann equation can be approximated by a second-order accurate trapezoidal rule, after which the resulting implicit scheme is recast into the explicit LBGK model by an appropriate change of variables.

It is interesting to point out that the terms of order $O(\varepsilon^0)$ in Eq. (18b) yield the Euler equation, and that the viscous contributions to the dynamics are determined by the $O(\varepsilon^1)$ tensor $\boldsymbol{\Pi}^{(1)}$. During the implementation of a boundary condition, it is not only central that boundary nodes implement an appropriate value of ρ and $\boldsymbol{u}$, which determines the form of the equilibrium distribution, and thus the $O(\varepsilon^0)$ Euler components of the dynamics. As has been pointed out previously among others in Ref. [9], it is also crucial that the first-order tensor $\mathbf{\Pi}^{(1)}$ is implemented correctly to ensure a proper representation of fluid viscosity. The multiscale analysis is helpful to understand the relation between the $O(\epsilon^1)$ terms components of LB models and the macroscopic variables. For the LBGK model, the value of $f_i^{(1)}$ reads as

$$f_i^{(1)} = -\frac{t_i}{c_s^2\omega}\left(\boldsymbol{Q}_i{:}\rho\,\nabla\,\boldsymbol{u} - \boldsymbol{c}_i\,\nabla:\rho\boldsymbol{u}\boldsymbol{u} + \frac{1}{2c_s^2}(\boldsymbol{c}_i\cdot\nabla)(\boldsymbol{Q}_i{:}\rho\boldsymbol{u}\boldsymbol{u})\right). \tag{20}$$

It contains a dominating term proportional to $\boldsymbol{u}$ and two terms scaling as the square of $\boldsymbol{u}$ which are less important at low Mach numbers. They cancel for symmetry reasons during the evaluation of the tensor $\mathbf{\Pi}^{(1)}$,

$$\mathbf{\Pi}^{(1)} = -\frac{2c_s^2}{\omega}\rho\boldsymbol{S}, \tag{21}$$

where $\boldsymbol{S}$ is the strain rate tensor,

$$\boldsymbol{S} = \frac{1}{2}[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T]. \tag{22}$$

As terms with nonlinear velocity components do not contribute to $\mathbf{\Pi}^{(1)}$, it is reasonable to approximate $f_i^{(1)}$ by its linear term only,

$$f_i^{(1)} \approx -\frac{t_i}{c_s^2\omega}\boldsymbol{Q}_i{:}\rho\,\nabla\,\boldsymbol{u}. \tag{23}$$

Indeed, both values of $f_i^{(1)}$ in Eq. (20) and Eq. (23) yield the same stress tensor. They are therefore equivalent in view of their effect on fluid flow, as the asymptotic dynamics in Eq. (9) only depends on $\mathbf{\Pi}^{(1)}$, and not on the details of the first-order particle populations $f_i^{(1)}$.

In summary, the particle populations can be split into two main components which fully determine the hydrodynamic behavior of the model. The term of order $O(\epsilon^0)$ yields the equilibrium distribution, which depends on the macroscopic variables ρ and $\boldsymbol{u}$ via Eq. (12). The term of order $O(\epsilon^1)$ depends additionally on the velocity gradients. For the implementation of a boundary condition it is therefore necessary to possess knowledge of both the macroscopic variables and their gradients.

By introducing the results of the multiscale expansion into Eqs. (18a) and (18b), and by taking the limit of small Mach number, the Navier-Stokes equations for an incompressible fluid are recovered,

$$\nabla\cdot\boldsymbol{u} = 0, \quad \partial_t\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\nabla p + \nu\nabla^2\boldsymbol{u}. \tag{24}$$

These equations are used as a reference in the benchmarks of Sec. V, to verify the quality of the boundary conditions.

III. IMPLEMENTATION OF BOUNDARY CONDITIONS

Prior to the collision step, some particle populations are unknown on boundary nodes, as they are lacking corresponding neighbors. This is illustrated in Fig. 2 for an upper boundary on a D2Q9 lattice. The particle populations on the boundary node are represented schematically by a lattice vector pointing into the direction along which they were propagated during the previous streaming step. Three particle populations, indicated by dashed vectors, are unknown. The role of a boundary condition is to find a substitution for the three missing populations, and potentially also for the remaining six, in a way which is consistent with the dynamics of the model and which leads to the desired macroscopic behavior on the domain boundary.

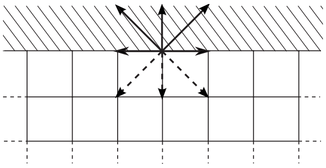

FIG. 2. Boundary node on the top row of a grid. The hatched region, located outside the grid, does not participate in the simulation. Particle populations on the boundary node are represented by their lattice vectors; dashed vectors stand for unknown populations.

When only the unknown populations are substituted, the action of a boundary condition can be viewed as the effect of a fictitious fluid portion located outside the numerical grid. This fluid portion streams the results from an imaginary LB operation into the boundary node and attributes a value to the unknown particle populations. In other types of boundary conditions, the vision of a fictitious LB dynamics is abandoned, and all particle populations are replaced on the boundary node.

Once the substitutions have been made, all boundary conditions described in this paper execute a conventional LBGK collision and streaming step. Boundary nodes implement therefore the same dynamics as bulk nodes, and they are also analyzed in the same theoretical terms. In particular, the results from the multiscale analysis displayed in Sec. II B also apply to boundary nodes. Thus, particle populations on boundaries can be split into an equilibrium component f_i^{eq} and an off-equilibrium contribution $f_i^{(1)}$ according to Eqs. (15) and (19).

An important constraint in determining these values is given by the conservation laws in Eqs. (8) and (9). Exact local mass and momentum conservation are pillars of the LB method, as they ensure among others an exceptional numerical stability. It is therefore desirable that they are respected on boundary nodes as well. This is achieved by making sure that mass and momentum, as computed from Eqs. (1) and (2), yield exactly the desired value. In that case, conservation laws are automatically verified, as it is seen by inserting Eq. (10) into Eqs. (8) and (9), and using the properties of Eqs. (13) and (14). The following sections do therefore not refer to conservation laws, but instead insist on an accurate implementation of Eqs. (1) and (2), without numerical approximation. It should be pointed out for clarity that even though these conservation laws guarantee equal mass in the pre- and post-collision state of a cell, they do not guarantee global mass conservation in the system. It has been shown in Ref. [10] that a simulation may gain or lose mass during time evolution, even with no-slip boundaries. An interpretation of this fact is that the actual mass balance is influenced by how the density is evaluated on boundaries (see Sec. III A).

The first step toward the implementation of a boundary condition is the construction of the equilibrium distribution, defined in Eq. (12), from ρ and $\boldsymbol{u}$. The velocity $\boldsymbol{u}$ is known by definition on velocity boundaries. The value of the particle density ρ is however missing and needs to be computed by a method described in the next section.

A. Evaluation of the density on boundaries

With a Dirichlet velocity boundary condition, the incompressible Navier-Stokes equations always yield a unique solution. No additional boundary condition for the pressure is required. With traditional numerical solvers for the Navier-Stokes equations, this fact is obvious, because the pressure is not treated as an independent variable. Instead, it is viewed as a functional of the velocity field, and may be determined by solving a Poisson equation.

The lattice Boltzmann approach on the other hand uses a quasicompressible method to solve the Navier-Stokes equations. The fluid density ρ is simulated as an independent variable in a regime close to fluid incompressibility and related to the pressure through Eq. (5). As it is shown in this section, the value of the density can however be determined on straight walls from the known particle populations (indicated by solid arrows in Fig. 2) and the value of the velocity. This method works on any 2D and 3D grid and is used by practically all authors of velocity boundary conditions. It is therefore applied throughout in this paper, for all boundary conditions and for all benchmark problems in Sec. V.

The density of a boundary node can be split into three components. The first, ρ_-, is the sum of the unknown particle populations. The second, ρ_+, is the sum of the particle populations opposite to the unknown ones, and the third, ρ_0, sums up the particle populations whose lattice vector is tangential to the boundary or zero. Let $u_\perp$ be the projection of the velocity onto the boundary normal pointing outside of the numerical grid. Then, the particle density of the boundary node is

$$\rho = \rho_- + \rho_+ + \rho_0, \tag{25}$$

and the normal momentum

$$\rho u_\perp = \rho_+ - \rho_-. \tag{26}$$

These equations are combined to obtain

$$\rho = \frac{1}{1+u_\perp}(2\rho_+ + \rho_0), \tag{27}$$

which is independent of the unknown quantity ρ_-. On the D2Q9 lattice of Fig. 1, and on a top grid node shown in Fig. 2, the variables above are defined as follows. The partial densities are $\rho_-=f_3+f_4+f_5$, $\rho_+=f_7+f_8+f_1$, and $\rho_0=f_2+f_6+f_0$. The normal velocity is given by the y component of the boundary velocity, and Eq. (27) becomes

$$\rho = \frac{1}{1+u_y}[2(f_3+f_4+f_5)+f_2+f_6+f_0]. \tag{28}$$

TABLE I. The boundary conditions reviewed in this paper.

Method	Replaces all f_i's	Local	Explicit in 3D
BC1 (Inamuro *et al.*)	No	Yes	No
BC2 (Zou-He)	No	Yes	Yes
BC3 (Regularized)	Yes	Yes	Yes
BC4 (FD)	Yes	No	Yes
BC5 (Nonlinear FD)	Yes	No	Yes

This method can obviously only be applied to straight boundaries. On other types of boundaries, for example, on corner nodes, the locally available information on a cell may be insufficient for the evaluation of the density. In that case, it is common to extrapolate the density from neighboring bulk cells. For the benchmark problems in Sec. V, second-order accurate extrapolation is used to evaluate the pressure in the four corners in 2D problems, and in the eight corners and the 12 edges in 3D problems.

B. Preserving the known particle populations

The two boundary conditions *BC1* and *BC2*, which are introduced in Sec. IV and listed in Table I, compute values for the unknown particle populations only, and leave the others untouched. By doing so, they exploit all the information available from the LB dynamics in the bulk of the simulation. As it is shown in Sec. V, they are rewarded by achieving more accurate results in 2D flows, but lose this advantage in a 3D problem.

As it is pointed out in the introduction to Sec. III, it is important that the values for the density and the velocity are recovered accurately from Eqs. (1) and (2) on boundary nodes. It is clear from Sec. III A that the particle density of a boundary node depends on the known populations and on the value of u_y, but not on the detail of the unknown populations. Therefore, in situations where only the unknown particle populations are replaced, the value of ρ needs not (and cannot) be enforced explicitly. Enforcing the value of the velocity from Eq. (2), on the other hand, yields two equations in 2D and three equations in 3D. Additional closure relations are therefore required to match the number of unknown particle populations, which amount to three on the D2Q9 lattice, and to nine (for D3Q27) or five (for D3Q19 and D3Q15) on 3D lattices.

It was argued in Sec. II B that the hydrodynamic limit of the lattice Boltzmann equation is expressed in terms of the quantities ρ, $\boldsymbol{u}$, and $\boldsymbol{\Pi}^{(1)}$. To help understand the boundary conditions from a hydrodynamic point of view, it is therefore good to reinterpret the closure relations suggested by the authors of boundary conditions as closure relations for the values of the stress tensor $\boldsymbol{\Pi}^{(1)}$. As $\boldsymbol{\Pi}^{(1)}$ is symmetric, this introduces a set of three equations in 2D, and six equations in 3D, which are enforced by Eq. (3). Some of the equation for $\boldsymbol{\Pi}^{(1)}$ [Eq. (3)] are however linearly dependent on those for $\boldsymbol{u}$ [Eq. (2)]. In the 2D example illustrated in Fig. 2, the equation for $\Pi_{xy}^{(1)}$ depends linearly on the equation for u_x, and the equation for $\Pi_{yy}^{(1)}$ depends on the one for u_y. Generally

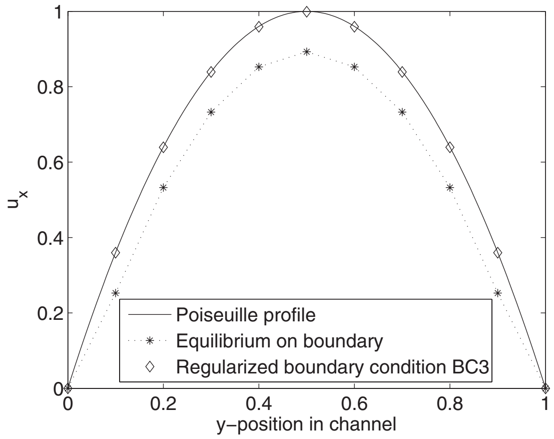

FIG. 3. 2D channel flow, simulated once with an equilibrium distribution on boundaries, and once with the regularized boundary condition *BC3*. The solid line plots the analytic Poiseuille profile. Although only 11 data points are used to resolve the channel width, the result obtained with *BC3* is visually indistinguishable from the Poiseuille profile. With equilibrium boundaries on the other hand, the velocity gradients are ill represented on boundary nodes, which impairs the accuracy of the whole simulation.

speaking, a total of d equations for the tensor components $\Pi^{(1)}_{n\alpha}$, for $\alpha=0\cdots d-1$, are linearly dependent on the equations of u_α. Here, the index n labels the direction of the boundary normal (in the above example, this is the y index). In 2D in particular, the two equations for $\boldsymbol{u}$ and the remaining equation for $\boldsymbol{\Pi}^{(1)}$ yield three closure relations, in the same amount as when a closure relation is introduced for each of the missing particle populations.

C. Replacing all particle populations

The boundary conditions *BC3*, *BC4*, and *BC5* in Table I use the known particle populations only to compute the particle density ρ and, in the case of *BC3*, the stress tensor $\boldsymbol{\Pi}^{(1)}$. After this, they disregard all particle populations and replace their value by new ones, based on the velocity and the computed macroscopic variables. As they treat all particle populations in the same way, they can be formulated generically and are easily implemented on various 2D and 3D lattices. Furthermore, they turn out to be more stable than *BC1* and *BC2*, and can therefore be used at substantially higher Reynolds numbers.

A tempting approach to setting up such velocity boundaries might be to initialize all particle populations, the known and the unknown ones, to their equilibrium distribution with the imposed velocity and the density computed from Eq. (28). It is however clear from the discussion in Sec. II B that this method is inappropriate, as it respects the $O(\varepsilon^0)$ contributions to the dynamics, but neglects the hydrodynamically relevant $O(\varepsilon^1)$ terms. As a result, the quality of the simulation is spoiled, not only close to boundaries, but in the full simulated domain. The expected second-order accuracy can consequently not be reached. To illustrate this issue, Fig. 3 displays the velocity profile obtained for the steady-state solution of a 2D channel flow. The no-slip (zero velocity) walls are implemented once via this equilibrium approach, and once with the obviously more accurate boundary condition *BC3*. An exact visual match between the exact analytical solution and the numerical result is obtained with the accurate boundary condition. With the equilibrium boundary condition on the other hand, the numerical values are far from the analytical curve. Although the velocity has a correct no-slip value on the boundary, the velocity gradient starts out erroneously. This confirms the theoretical considerations of the preceding section, in which the $O(\varepsilon^0)$ components of the particle populations are shown to represent macroscopic variables, and the $O(\varepsilon^1)$ components their gradients.

It is finally noted that one may avoid rewriting the values of all particle populations in an actual implementation to increase numerical efficiency. Once the density and velocity on a boundary node are computed, particle populations which leave the numerical domain (corresponding to the three upward pointing arrows on Fig. 2) have no more influence on the future evolution of the system. This is due to the fact that BGK collision in Eq. (10) does not introduce a coupling between lattice directions, except during the evaluation of macroscopic variables. The value carried by particle populations that leave the domain is therefore irrelevant in the LBGK model, but may still be important if another type of collision is used.

IV. PRESENTATION OF THE REVIEWED BOUNDARY CONDITIONS

The five boundary conditions listed in Table I are analyzed in this paper. The table specifies whether a boundary condition replaces all particle populations by a new value, or only the unknown ones. It also states whether they are local to a given boundary node, or if they need to access neighboring nodes in order to execute their algorithm. The original papers of boundary conditions most often concentrate on 2D implementations. Table I specifies whether there exists a straightforward way to extend them to 3D, or if this requires the solution of an implicit equation.

Boundary condition *BC1* is described in a paper by Inamuro *et al.* [11] in 1995. Boundary condition *BC2* was presented subsequently in 1997 by Zou and He [12]. Both methods modify only the value of unknown particle populations, and both are local. While *BC2* extends naturally to any 3D lattice, 3D formulations of *BC1* are more contrived and require the solution of an implicit equation at each iteration step of the method. Boundary condition *BC3*, formulated by Latt and Chopard [3], replaces all particle populations by a new value. It is local and extends naturally to any 3D lattice. Boundary conditions *BC4* and *BC5* are nonlocal, as they use finite-difference (FD) approximations to evaluate the velocity gradients in Eqs. (20) and (23). They are based on the work by Skordos [13], with two minor modifications that are discussed in Secs. IV D and IV E.

In the remaining part of this section, the five reviewed boundary conditions are introduced within a common framework. A formalism is used in which all particle populations f_i on a boundary node are replaced by alternative populations g_i resulting from the boundary condition algorithm. It is stated explicitly when some of the original f_i's are kept. It is then assumed that a LBGK collision is applied to the populations g_i, followed by a streaming step. Explicit formulas are given for the D2Q9 lattice of Fig. 1 and for a top grid node as in Fig. 2. Furthermore, it is explained how the algorithms can be extended to 3D lattices.

All presented boundary conditions are shown to recover the desired hydrodynamic behavior. According to the discussion in Sec. II B, this amounts to showing that the expected value for the boundary velocity is obtained from Eq. (2), and that the stress tensor calculated from Eq. (3) is consistent with its hydrodynamic value predicted by Eq. (21). These requirements are enforced by construction in the boundary conditions *BC3*, *BC4*, and *BC5*, and are therefore valid for obvious reasons. For the boundary conditions *BC1* and *BC2* however, only the value of the velocity is explicitly enforced. In this paper, the value of the stress tensor is therefore evaluated separately for the D2Q9 lattice, as a proof of validity. For this, the six known particle populations are assumed to obey the value predicted by the multiscale analysis in Eq. (23), and thus, to depend on ρ, $\boldsymbol{u}$, and the gradients of $\boldsymbol{u}$. Based on this assumption, the missing particle populations are evaluated from the algorithm of the boundary condition, and the stress tensor $\boldsymbol{\Pi}^{(1)}$ is computed from Eq. (3).

A. Inamuro *et al.* method (*BC1*)

The boundary condition *BC1* by Inamuro *et al.* uses a reasoning based on the kinetic theory of gases to find an appropriate value for the missing particle populations. It assumes that the missing populations are at a local thermohydrodynamic equilibrium and thus yield the discrete analog of a Maxwell-Boltzmann distribution. This distribution is however centered around a fictitious density ρ' and velocity $\boldsymbol{u}'$, both of which are different from the macroscopic fluid variables ρ and $\boldsymbol{u}$. While ρ' is a free variable, $\boldsymbol{u}'$ is assumed to differ from $\boldsymbol{u}$ only by a "slip velocity" $\boldsymbol{s}$, which is tangential to the wall, $\boldsymbol{u}'=\boldsymbol{u}+\boldsymbol{s}$. These assumptions introduce a set of two unknowns in 2D, and three unknowns in 3D. For an upper boundary on a D2Q9 lattice, they read as

$$g_i = f_i^{\text{eq}}(\rho', u_x + s_x, u_y) \quad \text{for} \quad i = 3,4,5, \tag{29}$$

where f_i^{eq} is the equilibrium distribution described by Eq. (12). As has been emphasized in Sec. III B, the value of the velocity $\boldsymbol{u}$ needs to be enforced on the boundary via Eq. (2). This system of two equations can be solved for the two unknowns ρ' and s_x, with the following result:

$$\rho' = 6\frac{-\rho u_y + f_1 + f_7 + f_8}{1 + 3u_y^2 - 3u_y}, \tag{30}$$

$$s_x = -u_x + 6\frac{\rho u_x + f_1 + f_2 - f_6 - f_7}{\rho'(1-3u_y)}. \tag{31}$$

In these relations, the lattice constants have been replaced by their numerical value for the D2Q9 lattice. The missing populations are now evaluated with the help of Eq. (29), and the other populations are left unchanged,

$$g_i = f_i \quad \text{for} \quad i = 0,1,2,6,7,8. \tag{32}$$

In 3D, the slip velocity $\boldsymbol{s}$ has two independent components. The unknowns thus amount to 3, which matches the number of closure relations introduced by Eq. (2). This system of equations is more difficult to solve than in the 2D case, and to our knowledge, no analytic solution is presently known. For the 3D benchmarks in Sec. V, the lattice Boltzmann code solves these equations numerically on each boundary cell and at every time step with the help of a multivariate Newton-Raphson solver [14].

The velocity $\boldsymbol{u}$ is explicitly enforced by this boundary condition, but nothing is known *a priori* about the value of the stress tensor. The method described in the introduction of Sec. IV is now applied to evaluate $\boldsymbol{\Pi}^{(1)}$ for the D2Q9 lattice, by making use of a computer algebra system. The following values are found:

$$\begin{aligned}\Pi_{xx}^{(1)} = {} & \frac{\rho}{9\omega(1-3u_y)^2(\omega - 2S_{yy} - 3\omega u_y + 3\omega u_y^2)} \\ & \times\Bigg\{ 10S_{xx}S_{yy}(1-3u_y)^2 + 6S_{xy}^2(1-3u_y+3u_y^2) \\ & - \omega(3u_y - 1)\Bigg[5S_{xx}(-1+6u_y-12u_y^2+9u_y^3) \\ & + S_{yy}u_x^2(3-9u_y) + 18u_xS_{xy}\left(\frac{1}{3} - u_y + u_y^2\right)\Bigg]\Bigg\},\end{aligned} \tag{33a}$$

$$\Pi_{yy}^{(1)} = -\frac{2}{3\omega}\rho S_{yy}, \tag{33b}$$

and

$$\Pi_{xy}^{(1)} = -\frac{2}{3\omega}\rho S_{xy}. \tag{33c}$$

The tensor components Π_{yy} and Π_{xy} yield an exact match with the value expected from Eq. (21). As has been pointed out in Sec. III B, this is a direct corollary of the fact that the equation for the velocity, Eq. (2), is satisfied. The component Π_{xx} is the only free parameter on a D2Q9 lattice, and can be viewed as a signature of how the boundary condition influences the dynamics of the fluid. In its present form, Eq. (33) is not very enlightening, and we present therefore two limit values. The first represents the case of a no-slip wall, with vanishing values of $\boldsymbol{u}$,

$$\Pi_{xx}^{(1)} = \rho\frac{-5S_{xx} + \dfrac{2}{\omega}(3S_{xy}^2 + 5S_{xx}S_{yy})}{9\omega - 18S_{yy}} \quad \text{when} \quad \boldsymbol{u} = 0. \tag{34}$$

For the second limit case, $\Pi_{xx}^{(1)}$ is expanded into a finite Taylor series for small values of the components S_{xx}, S_{yy}, and S_{xy}. It is assumed that, in the limit of fluid incompressibility, the Mach number Ma, and thus the velocity $\boldsymbol{u}$ in lattice units, scales like the small parameter ε [15]. The components of the strain rate tensor $\boldsymbol{S}$ exhibit therefore order ε^2 scaling.

From the above Taylor series, only terms of order ε^2 are retained, and terms of the form $S_{\alpha\beta}S_{\gamma\delta}$ or $S_{\alpha\beta}u_\gamma$ are neglected. This leads to the following approximation of Eq. (33):

$$\Pi_{xx}^{(1)} = -\frac{5}{9\omega}\rho S_{xy} + O(\varepsilon^3). \tag{35}$$

This expression is identical to the hydrodynamic tensor value in Eq. (21), with exception of the factor 5/9, which differs from the expected factor 2/3 by 20%. This error has however no important effect in practice, as the Inamuro *et al.* method is found to be equally or even more accurate than the other approaches in low Reynolds applications (see Sec. V). One may however point out that as a consequence of Eq. (35), the trace of $\mathbf{\Pi}^{(1)}$ is nonzero with *BC1*. This is a violation of the continuity equation, because Eqs. (21) and (22) show that the trace of $\mathbf{\Pi}^{(1)}$ is proportional to $\nabla\cdot\boldsymbol{u}$ in the hydrodynamic limit.

B. Zou-He method (*BC2*)

The boundary condition *BC2* by Zou and He is based on an idea referred to by its authors as "applying the bounce-back rule to off-equilibrium parts." This expression can be understood as a reference to a symmetry property of the variables $f_i^{(1)}$ that follows from Eq. (23). Let the opposite opp(i) of a lattice direction i be defined by the relation $\boldsymbol{c}_{\text{opp}(i)} = -\boldsymbol{c}_i$. Then, Eq. (23) is symmetric under the operation of taking the opposite direction,

$$f_i^{(1)} = f_{\text{opp}(i)}^{(1)}. \tag{36}$$

This relation can be used to copy data from known particle populations to the opposite, unknown ones. It may however not be applied blindly to all unknown particle populations, because this would conflict with Eq. (2) for the velocity. On a D2Q9 lattice, for example, there is really only one degree of freedom left on which the boundary condition may have an influence. The choice made in the case of *BC2* was to apply Eq. (36) to the unknown particle population whose lattice vector is parallel to the boundary normal. Thus, for an upper boundary on a D2Q9 lattice, the following relation is applied:

$$g_4^{(1)} = f_8^{(1)}. \tag{37}$$

The remaining two unknown populations $g_3^{(1)}$ and $g_5^{(1)}$ are computed by enforcing the values of the velocity through Eq. (2). On a D2Q9 lattice, the obtained values are

$$g_3^{(1)} = f_7^{(1)} + \frac{1}{2}(f_6^{(1)} - f_2^{(1)}) \tag{38}$$

and

$$g_5^{(1)} = f_1^{(1)} + \frac{1}{2}(f_2^{(1)} - f_6^{(1)}). \tag{39}$$

The value of the unknown particle populations is constructed by adding equilibrium and off-equilibrium parts,

$$g_i = f_i^{\text{eq}}(\rho,\boldsymbol{u}) + g_i^{(1)} \quad \text{for } i = 3,4,5. \tag{40}$$

All other populations are kept as they are,

$$g_i = f_i \quad \text{for } i = 0,1,2,6,7,8. \tag{41}$$

As in the Inamuro *et al.* method *BC1*, the Zou-He method *BC2* enforces the value of the velocity explicitly, but does not specify anything about the stress tensor. The values of $\mathbf{\Pi}^{(1)}$ can again be computed for the D2Q9 lattice, using the procedure described in the introduction of Sec. IV. The resulting expressions match exactly the hydrodynamic values of $\mathbf{\Pi}^{(1)}$ claimed in Eq. (21). This confirms analytically the validity of the Zou-He boundary condition for 2D simulations.

This algorithm cannot be used as it stands in 3D. In that case, Eq. (2) yields three independent equations, but the number of unknowns is larger (five or nine). In that case, the authors of Ref. [12] suggest to proceed as follows. To begin, the "off-equilibrium bounce-back" rule, Eq. (36), is applied to assign a value to all missing particle populations. By doing this, the exact value of the component u_α is recovered for the index α in the direction of the boundary normal. To keep this relation valid, the sum over the value of the unknown particle populations is kept invariant during subsequent operations. The excess of momentum is then evaluated in the remaining directions,

$$\delta_\beta \equiv \sum_i f_i^{\text{neq}} c_{i\beta} \quad \text{for } \beta \neq \alpha. \tag{42}$$

Finally, the values δ_β are redistributed over the unknown particle populations, in order to find an exact match for Eq. (2),

$$g_i^{\text{neq}} = f_i^{\text{neq}} - \sum_{\beta\neq\alpha} \frac{1}{n_\beta} c_{i\beta}\delta_\beta \quad \text{for all unknown } f_i\text{'s}, \tag{43}$$

where n_β is the number of unknown particle populations for which $c_{i\beta}$ is nonzero.

C. Regularized method (*BC3*)

The method [3] by Latt and Chopard replaces all particle populations on a boundary node. It first evaluates the value of $\mathbf{\Pi}^{(1)}$, based on the knowledge of the known particle populations. Then, the equations of the multiscale analysis in Sec. II B are reversed to reconstruct all particle populations from ρ, $\boldsymbol{u}$, and $\mathbf{\Pi}^{(1)}$ in a way consistent with the hydrodynamic limit of the model. With the use of Eq. (21), and by exploiting the symmetry of tensors $\boldsymbol{Q}_i$ and of tensor $\mathbf{\Pi}^{(1)}$, Eq. (23) is rewritten as follows:

$$f_i^{(1)} \approx -\frac{\rho t_i}{c_s^2\omega}\boldsymbol{Q}_i{:}\boldsymbol{S} = \frac{t_i}{2c_s^4}\boldsymbol{Q}_i{:}\mathbf{\Pi}^{(1)}. \tag{44}$$

With this relation, the off-equilibrium part of the particle populations can be reconstructed from the tensor $\mathbf{\Pi}^{(1)}$. This procedure is called a regularization of the particle populations, because it compels the $f_i^{(1)}$ to respect exactly (without numerical error) the symmetry properties of Eq. (21), such as the symmetry explicated in Eq. (36).

The tensor $\mathbf{\Pi}^{(1)}$ is evaluated as follows. First, all unknown particle populations are assumed to take the value obtained by a "bounce-back of off-equilibrium parts," as in Eq. (36). Thus, their value is described as $f_i = f_i^{\text{eq}}(\rho,\boldsymbol{u}) + f_{\text{opp}(i)} - f_{\text{opp}(i)}^{\text{eq}}$. It was emphasized in Sec. IV B that this relation may not be used to define a boundary condition, because it prohibits an exact implementation of Eq. (2). It therefore is used temporarily only, to evaluate the value of $\mathbf{\Pi}^{(1)}$ by means of Eq. (3).

After this, Eq. (44) is used to construct all particle populations,

$$g_i = f_i^{\text{eq}}(\rho,\boldsymbol{u}) + \frac{t_i}{2c_s^4}\boldsymbol{Q}_i\mathbin{:}\mathbf{\Pi}^{(1)} \quad \text{for } i = 0 \cdots q-1. \tag{45}$$

It is obvious that not only ρ and $\boldsymbol{u}$, but also the tensor $\mathbf{\Pi}^{(1)}$ are recovered appropriately by this boundary condition.

D. Finite-difference velocity gradient method (*BC4*)

In the regularized approach of *BC3*, the stress tensor $\mathbf{\Pi}^{(1)}$ is evaluated on the boundary from information locally available on the cell. The boundary condition *BC4* computes $\mathbf{\Pi}^{(1)}$ from Eq. (21) instead, by relating it to the strain rate tensor $\boldsymbol{S}$. The components of $\boldsymbol{S}$ are evaluated by a second-order accurate finite-difference scheme, which accesses the value of the velocity on neighboring grid cells. The algorithm of *BC4* is summarized by the following equation:

$$g_i = f_i^{\text{eq}}(\rho,\boldsymbol{u}) - \frac{\rho t_i}{c_s^2\omega}\boldsymbol{Q}_i\mathbin{:}\boldsymbol{S} \quad \text{for } i = 0 \cdots q-1, \tag{46}$$

where the strain rate tensor $\boldsymbol{S}$ is defined as in Eq. (22). The velocity gradients of $\boldsymbol{S}$ that run along directions parallel to the boundary are evaluated by a symmetric finite-difference scheme. This presumes the knowledge of the velocity on nearest-neighbor cells. Velocity gradients along the boundary normal use nonsymmetric finite differences, which access the velocity on nearest and next-to-nearest neighbors. Note that, instead of evaluating the symmetric tensor $\boldsymbol{S}$, one may alternatively compute all components of the nonsymmetric tensor $\boldsymbol{\nabla u}$ and calculate the population functions from Eq. (23). These two ways of constructing the boundary condition are algebraically equivalent.

This boundary condition was presented in 1993 in a pioneering paper by Skordos [13]. It should be mentioned that Ref. [13] makes a slightly different assumption on the asymptotic value of $\mathbf{\Pi}^{(1)}$, as it uses the following expression in lieu of Eq. (21), in which the density ρ resides inside the spatial derivative:

$$\mathbf{\Pi}^{(1)} = -\frac{c_s^2}{\omega}\{\boldsymbol{\nabla}(\rho\boldsymbol{u}) + [\boldsymbol{\nabla}(\rho\boldsymbol{u})]^T\}. \tag{47}$$

It has however been shown that Eq. (21), which predicts the proper deviatory strain rate for a compressible fluid, can be recovered in a multiscale analysis by taking into account derivatives of nonlinear velocity terms [3,16,17]. The difference between Eqs. (21) and (47), however subtle, may have a noticeable effect in compressible fluids. We decided therefore to depart from the original paper on this point, but emphasize that the credit for this boundary condition should be attributed to Skordos [13]. It should also be mentioned that Ref. [13] discusses both first- and second-order accurate finite-difference approximations to velocity gradients, but only second-order schemes are presented here.

An interesting discussion of boundary conditions with finite-difference approximation to velocity gradients is presented in Ref. [9]. This paper points out that other choices than Eq. (46) also lead to the expected value of the stress tensor, and it proposes different closure schemes that are compatible with the hydrodynamic limit of the model.

E. Nonlinear finite-difference method (*BC5*)

It is shown by multiscale analysis that Eq. (23) is a valid approximation to Eq. (20) within the hydrodynamic scales of the model. It is therefore clear that boundary conditions *BC3* and *BC4*, which are based on this approximation, yield accuracy of second order, consistent with the overall accuracy of the lattice Boltzmann scheme. Beyond this asymptotic estimate, the accuracy may however vary in actual simulations, due to higher-order effects that are not visible in the finite expansion of Sec. II B. For this reason, it is sometimes argued that the full term of Eq. (20) should be implemented on the boundary, to gain accuracy and numerical stability. In order to test this conjecture numerically, a corresponding boundary condition *BC5* is presented in the following. It uses the same principles as *BC4*, but also includes the nonlinear velocity contributions to the particle populations. This approach has not been presented previously in the literature, and serves only as a toy model to experiment with the limits of *BC4*. It is explicated as follows:

$$g_i = f_i^{\text{eq}}(\rho,\boldsymbol{u}) - \frac{t_i}{c_s^2\omega}\left(\boldsymbol{Q}_i\mathbin{:}\rho\,\boldsymbol{\nabla}\,\boldsymbol{u} - \boldsymbol{c}_i\,\boldsymbol{\nabla}\mathbin{:}\rho\boldsymbol{u}\boldsymbol{u} + \frac{1}{2c_s^2}(\boldsymbol{c}_i\cdot\boldsymbol{\nabla})\times(\boldsymbol{Q}_i\mathbin{:}\rho\boldsymbol{u}\boldsymbol{u})\right) \quad \text{for } i = 0 \cdots q-1. \tag{48}$$

For the implementation of this method, the gradients of the components of the tensor $\boldsymbol{uu}$ need to be evaluated via a finite-difference scheme, additionally to the gradients of $\boldsymbol{u}$.

V. BENCHMARKS

This section produces the results of lattice Boltzmann simulations, based on the five reviewed boundary conditions. For some of the simulated problems, analytical solutions are known which describe the steady state of the flow. In that case, the simulation is iterated until a steady state is reached. Then, the numerical result is validated against the analytical solution on every grid point, and the mean discrepancy between these two values is evaluated as a measure of quality for the implemented boundary condition. Other benchmarks focus on the time evolution of the flow. They keep track of the time evolution of a scalar quantity (such as the average energy in Sec. V C), or the occurrence of a particular event (as the maximum value of the enstrophy in Sec. V D). Several 2D benchmarks are presented, implemented on a D2Q9 lattice, and one 3D benchmark on a D3Q19 lattice.

The simulated domains of all presented problems have a rectangular shape. This raises the issue of how to implement corner nodes and, in 3D, edge nodes that lie on the connection between two plane walls. Although some authors of boundary conditions make suggestions on how to treat these cases, some authors do not. To guarantee an equal treatment, the present paper treats corners and edges always in the same way, independently of the boundary condition that is being tested. The finite-difference algorithm of *BC5* was selected to implement these special boundary nodes. This approach is straightforward, because the velocity gradients referred to in Eq. (45) can be evaluated on corners and edges just as everywhere else. It must furthermore be pointed out that Eq. (28) for the particle density cannot be evaluated on corners and edges. In those cases, the density is extrapolated with second-order accuracy from neighboring cells.

In all benchmark problems, a velocity U and a length L are selected that are characteristic for the flow. The dynamics of the flow is then described in a system of dimensionless variables, independent of the numerical grid, in which $U=1$ and $L=1$. The equations of motion depend only on the dimensionless Reynolds number, defined as a function of the kinematic fluid viscosity ν,

$$\mathrm{Re} = \frac{UL}{\nu}. \tag{49}$$

The parameters used for the numerical implementation are described by the number N, which is the number of grid nodes used to resolve the length L, and the velocity U_{lb}, representing the velocity U in a system of lattice units. This is a common choice in LB simulations, because U_{lb} is proportional to the Mach number of the fluid. It can therefore be fine-tuned to make sure the flow is close enough to the limit of incompressibility. For the numerical implementation of the benchmarks, the prescribed velocity field for the initial and/or boundary condition needs to be converted from dimensionless variables to lattice units. This is simply done by multiplying the dimensionless velocity by U_{lb}. From these definitions, the discrete parameters of the simulation are characterized as follows. The grid interval δ_x is given by

$$\delta_x = \frac{1}{N-1}. \tag{50}$$

The time lapse of an iteration δ_t is defined through the relation $U=\delta_x/\delta_t U_{lb}$, which recovers the lattice-independent form of the velocity. Given our choice of $U=1$, this leads to

$$\delta_t = \delta_x U_{lb}. \tag{51}$$

At a given time step, the numerical error is evaluated by computing a l_2-norm of the difference between the simulated velocities $\boldsymbol{u}_{lb}(\boldsymbol{r}_k)$ on grid nodes located at position $\boldsymbol{r}_k$, and the dimensionless analytical solution $\boldsymbol{u}_{\mathrm{analytic}}(\boldsymbol{r}_k)$,

$$\epsilon = \sqrt{\frac{1}{P}\sum_{k=0}^{P-1}\left\|\frac{\boldsymbol{u}_{lb}(\boldsymbol{r}_k)}{U_{lb}} - \boldsymbol{u}_{\mathrm{analytic}}(\boldsymbol{r}_k)\right\|^2}, \tag{52}$$

where the sum runs over all P nodes of the numerical grid.

The parameters of the simulation for a varying resolution N are adapted in order to keep the value of the Reynolds number constant, and to prevent effects of fluid compressibility from interfering with the accuracy of the result. Compressibility errors $\epsilon_{\mathrm{compr}}$ are known to scale as the square Mach number and are therefore estimated by $\epsilon_{\mathrm{compr}}=O(U_{lb}^2)$. This error is required to decrease at least as fast as the discretization error $\epsilon_{\delta_x}=O(1/L_{lb}^2)=O(1/N^2)$. It follows from Eq. (49) that this is achieved by keeping the viscosity ν_{lb} constant when the grid resolution is modified.

The benchmark codes can be found from Ref. [18]. The programs are based on the open source lattice Boltzmann library OpenLB [19], which is publicly available.

A. Steady plane channel flow (2D)

This 2D stationary flow evolves in a straight channel, which extends in the x direction between $x=0$ and $x=l_x$. The walls of the channel are parallel to the x axis, and defined by the equations $y=0$ for the lower wall, and $y=l_y$ for the upper wall. A no-slip condition for the velocity is enforced on these walls. The flow is characterized by a constant pressure drop along the channel. This pressure drop can be obtained by using pressure boundary conditions on the inlet and the outlet. An alternative approach, which is used in this benchmark, is to enforce the velocity profile from the analytical solution of the flow on the inlet and on the outlet. The velocity is parallel to the walls, and the only nonzero component u_x is independent of x, $u_x=u_x(y)$. The height l_y of the channel is taken to represent the characteristic length L. The maximum value of the velocity, which is measured in the middle of the channel, is selected for the characteristic velocity U. The analytic solution to this flow is given by the parabolic Poiseuille profile, which, in dimensionless variables, reads as

$$u_x(y) = 4(y - y^2). \tag{53}$$

The pressure drop amounts to $\partial p/\partial x=-8/\mathrm{Re}$.

The two-dimensional channel flow is for many reasons an inappropriate benchmark, and it is presented here in a historical spirit, to conform with some authors of boundary conditions that use it as a basic test case for their algorithm. One shortcoming of this flow as a benchmark case is that the components S_{xx} and S_{yy} of the strain rate tensor $\boldsymbol{S}$ vanish in the analytical solution. Boundary conditions that replace only unknown particle populations (*BC1* and *BC2* in this paper) are therefore automatically exempt from numerical errors. This follows from the discussion in Sec. III B, where the components Π_{xx} (on horizontal boundaries) and Π_{yy} (on vertical boundaries) of the stress tensor are shown to be the only degrees of freedom for which the boundary condition could possibly be wrong. If the corner nodes are implemented in an appropriate way, these two methods are shown in Refs. [11,12] to solve the 2D channel flow with a precision close to machine accuracy (the best accuracy one can expect to obtain, given the limited precision of floating point values on a computer), independent of the grid resolution. Machine accuracy is however not exhibited in the present benchmark, because we chose to implement corner nodes in a generic

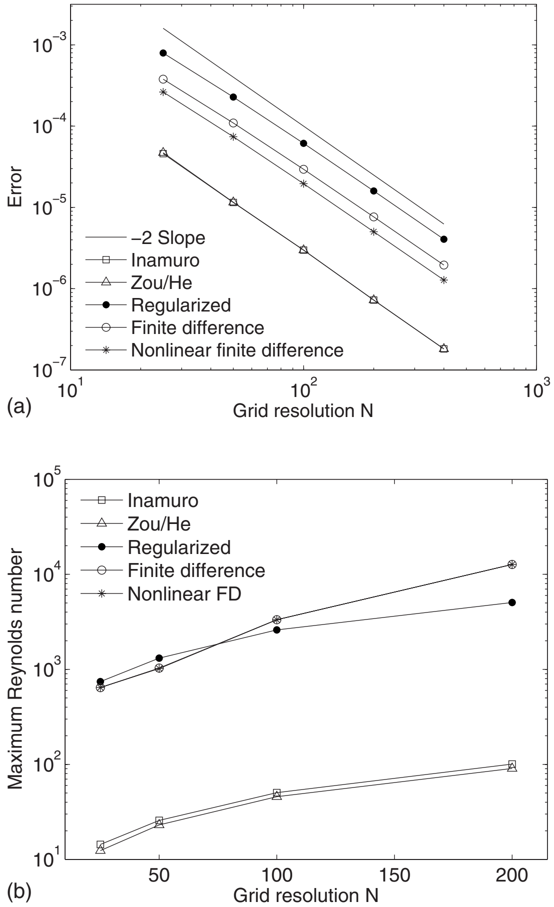

FIG. 4. (a) Numerical accuracy in a 2D channel flow. (b) Numerical stability in a 2D channel flow. The maximum Reynolds number which can be reached before numerical instabilities appear is plotted.

way that works for all boundary conditions and for all types of flows.

In this benchmark, the channel has quadratic shape ($l_x = l_y = 1$), the grid resolution is varied from $N=25$ to $N=400$, and the Reynolds number is Re=10. Compressibility effects are controlled by setting $U_{lb}=0.01$ at a grid resolution $N=50$. As discussed in the preceding paragraph, the velocity U_{lb} is recomputed for different values of N in such a way as to keep the viscosity ν_{lb} constant from one grid to another. Figure 4(a) shows that all reviewed boundary conditions lead to the expected second-order accuracy with respect to grid resolution. Boundary conditions *BC1* and *BC2* are distinctly more accurate, because their overall accuracy is only affected by the error in corner nodes. It is furthermore observed that the nonlinear finite-difference scheme of *BC5* is slightly more accurate than the linear approach of *BC4*. The numerical stability of boundary conditions is explored in Fig. 4(b) by means of the maximum Reynolds number which can be reached before numerical instabilities occur. The most accurate boundary conditions, *BC1* and *BC2* are also the distinctly least stable. Most stable boundary conditions are those based on a nonlocal algorithm, *BC4* and *BC5*.

B. Oscillating plane channel flow (2D)

This laminar channel flow is also known as "Womersley flow" on grounds of the analytical solution proposed by Womersley [20]. It is defined by the same geometry as the steady flow in Sec. V A. The pressure gradient is however not constant, but oscillates in time. Before the reference velocity U can be defined, the equations of the flow are written in an arbitrary system of units, other than the dimensionless one. In this system, the amplitude of the oscillations is denoted by A, and the frequency by ω,

$$\frac{\partial p}{\partial x} = -A\cos(\omega t). \tag{54}$$

In the low-frequency limit $\omega \to 0$, the solution to this flow is defined by a succession of Poiseuille profiles, with oscillating amplitude,

$$u_x(y,t) = \frac{A}{2\nu}(l_y y - y^2)\cos(\omega t). \tag{55}$$

The maximum velocity of the flow, reached in the middle of the channel at time $t=n2\pi/\omega$ for any integer value n, is found to be

$$u_{\max} = \frac{A}{8\nu}L^2. \tag{56}$$

This value $u_{\max}$ is used to define the reference velocity U, as it characterizes the flow reasonably well when oscillations are slow. The reference length L is described by the channel height l_y. As the flow depends on time, it is characterized by two dimensionless parameters, which are the Reynolds number Re defined in Eq. (49), and the Womersley number α, defined as

$$\alpha = \frac{L}{2}\sqrt{\frac{\omega}{\nu}}. \tag{57}$$

In a system of dimensionless variables described by U and L, the time-dependent solution to this flow is given by the Womersley profile [20,21],

$$u_x(y,t) = \mathrm{Re}\left[e^{i\alpha^2/\mathrm{Re}\, t}\frac{8}{i\alpha^2}\left(1 - \frac{\cosh\left[\sqrt{2}(\alpha + i\alpha)\left(y - \frac{1}{2}\right)\right]}{\cosh\left(\frac{\sqrt{2}}{2}(\alpha + i\alpha)\right)} \right)\right], \tag{58}$$

where i is the imaginary unit, and Re means that the real part of the formula needs to be evaluated.

The error ϵ of the numerical result with respect to the analytical solution of Eq. (58) is computed at each time step by evaluation of Eq. (52) over the whole computational domain. The overall error $\bar{\epsilon}$ of the simulation is defined as an average of ϵ over one time period. Only the asymptotic value of $\bar{\epsilon}$, reached after a large number of iterations, is accounted

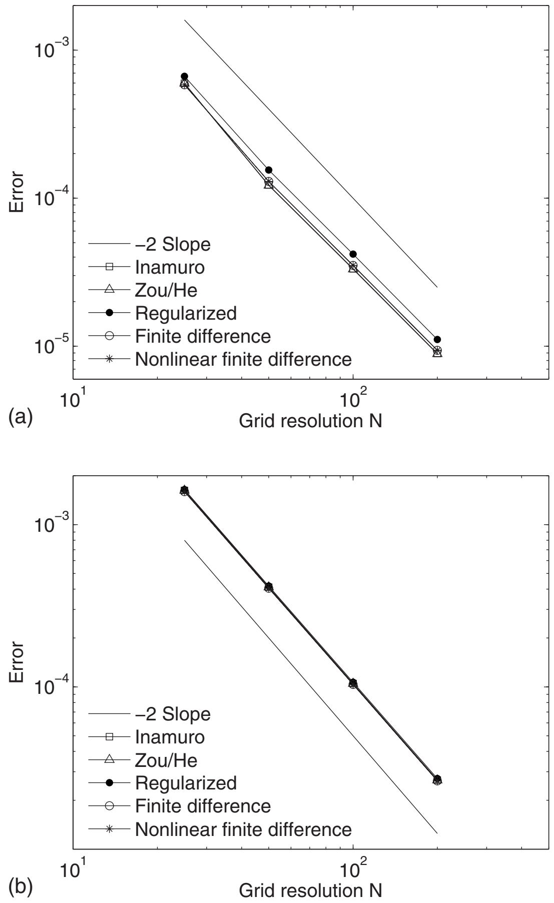

FIG. 5. Numerical accuracy in an oscillating channel. (a) $\alpha=2$ (slow oscillations), (b) $\alpha=5$ (fast oscillations).

for. The initial transient regime of the fluid, and consequently the choice of the initial condition, are irrelevant.

In this simulation, the geometry is quadratic ($l_x=l_y=1$). The Reynolds number is Re=1, and the lattice velocity is set to $U_{lb}=0.01$ at a reference resolution of $N=10$. The numerical results are plotted in Fig. 5(a) for a value of $\alpha=2$ and in Fig. 5(b) for a value of $\alpha=5$. All other parameters are chosen to be the same as in the steady channel flow. Figure 4(a) can also be included in this discussion, as it corresponds to the limit $\alpha=0$ in which the flow does not oscillate. It should be pointed out that, as α increases, inertial effects on the flow dominate over viscous effects. The velocity profile in the bulk becomes progressively independent of the boundary condition. As it can be seen from the simulations [see Fig. 5(b)], the result is practically independent of the chosen boundary condition at a value of $\alpha=5$.

C. Periodic array of vortices (2D)

The following benchmark measures the energy dissipation during time evolution of a laminar flow without external energy input. The initial velocity field consists of an array of counter-rotating vortices [22]. The flow is described by its value inside a periodic box of size $L\times L$. The flow velocity $\boldsymbol{u}_0$ at any time t_0 is derived from its stream function Ψ_0, which is an eigenfunction of the Laplacian operator with eigenvalue λ,

$$\nabla^2\Psi_0=\lambda\Psi_0. \tag{59}$$

The time evolution of this flow is characterized by an exponential decrease of the velocity amplitude,

$$\boldsymbol{u}(t)=\boldsymbol{u}_0 e^{\lambda\nu(t-t_0)} \quad \text{for} \quad t\geq t_0. \tag{60}$$

Among all eigenfunctions of the Laplacian, the following initial stream function Ψ_0 was selected for the benchmark (expressed in dimensionless variables):

$$\Psi_0(x,y)=\frac{1}{2\pi\sqrt{m^2+n^2}}\cos(2\pi mx)\cos(2\pi ny), \tag{61}$$

where x and y are contained in the interval $[0,1]$ and label the two space directions. The velocity field is given by

$$u_{x,0}(x,y)=\frac{\partial\Psi_0}{\partial y}=\frac{-\cos(2\pi mx)\sin(2\pi ny)}{\sqrt{\frac{m^2}{n^2}+1}} \tag{62a}$$

and

$$u_{y,0}(x,y)=-\frac{\partial\Psi_0}{\partial x}=\frac{\sin(2\pi mx)\cos(2\pi ny)}{\sqrt{\frac{n^2}{m^2}+1}}. \tag{62b}$$

It is clear that $\|\boldsymbol{u}_0(x,y)\|\leq 1$ for all values of x and y, which justifies the choice of the reference velocity U. The decay rate λ is found by solving the eigenvalue problem (59),

$$\lambda=-(m^2+n^2). \tag{63}$$

The initial pressure distribution of the flow is evaluated by solving the Poisson equation $\nabla^2 p=-(\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}$:

$$p=-[n^2\cos(4\pi mx)+m^2\cos(4\pi ny)]. \tag{64}$$

The dimensionless pressure p is converted as follows to the density ρ_{lb}, expressed in lattice units, $\rho_{lb}=1+p\delta_x^2/(c_s^2\delta_t^2)$, where the discrete steps δ_x and δ_t are defined as in Eq. (50). This benchmark is formulated as an initial value problem. The initial condition is therefore set up with special care, using an approach described in Ref. [23] and based on an original suggestion in Ref. [13]. The particle populations are first initialized at an equilibrium distribution, based on the desired value of $\boldsymbol{u}$ and p. The off-equilibrium parts of the particle populations are then instantiated with the help of Eq. (23), with numerically computed velocity gradients. Although the flow is periodic in both space directions, nonperiodic boundary conditions are used, and the boundary velocity is explicitly imposed by use of the reviewed boundary conditions. The velocity along the boundaries of this subdomain is set to the analytical value of Eq. (60) at every time step with the help of the five tested boundary conditions.

Figure 6 displays the time evolution of the average energy with different boundary conditions at a resolution $N=8$ and a resolution $N=32$. The Reynolds number is Re=1, and the reference velocity in lattice units is set to $U_{lb}=0.01$ at a grid resolution $N=8$. In an initial regime, the energy decay is in all cases exponential, as predicted by Eq. (60). After reaching a critical value of the time t, the curve starts oscillating and deviates from the theoretical prediction. For the nonlocal

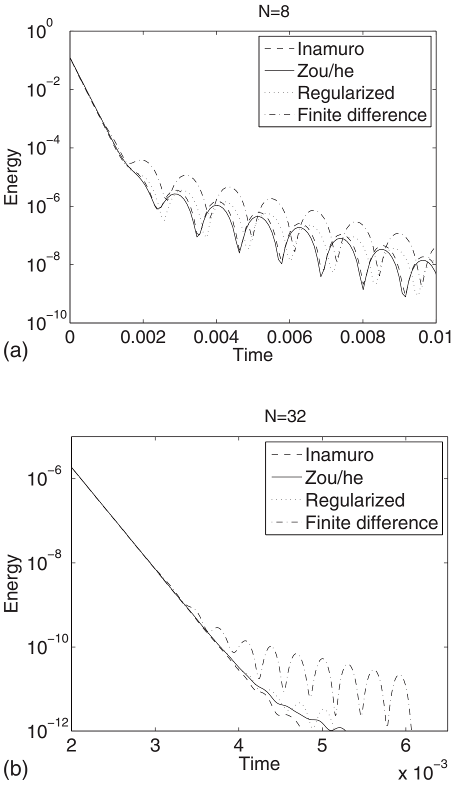

FIG. 6. Time evolution of the average energy in the 2D periodic array of vortices. The curves for the linear and the nonlinear finite-difference methods overlap as far as one can distinguish visually. Therefore, only the linear case is presented. A boundary condition is considered to be of good quality if the point at which the curve oscillates and departs from an exponential decay occurs late.

boundary conditions *BC4* and *BC5*, this deviation occurs earlier than for the local ones. It is therefore concluded that the nonlocal boundary conditions are less accurate for this problem.

D. Dipole-wall collision (2D)

This benchmark, based on Refs. [23,24], analyzes the time evolution of a self-propelled dipole confined within a square box. The box is located in the geometrical domain $[-L,L]\times[-L,L]$ and implements no-slip walls. The initial condition describes two counter-rotating monopoles, one with positive core vorticity at the position (x_1,y_1) and one with negative core vorticity at (x_2,y_2). This is obtained with an initial velocity field $\boldsymbol{u}_0=(u_0,v_0)$ which reads as follows in dimensionless variables:

$$u_0 = -\frac{1}{2}\|\eta_e\|(y-y_1)e^{-(r_1/r_0)^2} + \frac{1}{2}\|\eta_e\|(y-y_2)e^{-(r_2/r_0)^2} \tag{65a}$$

and

$$v_0 = +\frac{1}{2}\|\eta_e\|(x-x_1)e^{-(r_1/r_0)^2} - \frac{1}{2}\|\eta_e\|(x-x_2)e^{-(r_2/r_0)^2}. \tag{65b}$$

Here, the distance to the monopole centers is defined as $r_i=\sqrt{(x-x_i)^2+(y-y_i)^2}$. The parameter r_0 labels the diameter of a monopole and η_e its core vorticity.

The average kinetic energy of this system at a given time is defined by the expression

$$\bar{E}(t) = \frac{1}{2}\int_{-1}^{1}\int_{-1}^{1}\|\boldsymbol{u}\|^2(\boldsymbol{x},t)d^2x, \tag{66}$$

and the average enstrophy by

$$\bar{\Omega}(t) = \frac{1}{2}\int_{-1}^{1}\int_{-1}^{1}\eta^2(\boldsymbol{x},t)d^2x, \tag{67}$$

where $\eta=\partial_x v-\partial_y u$ is the flow vorticity.

The dipole described by Eqs. (65a) and (65b) develops a net momentum directed in the positive x direction and is self-propelled toward the right-hand wall. The collision between the dipole and the no-slip wall is characterized by a turbulent dynamics where the wall acts as a source of small-scale vortices that originate from detached boundary layers. This problem is therefore particularly interesting as a test for the ability of boundary conditions to reproduce the dynamics of boundary layers during collision. For this purpose, the maximum value of the flow enstrophy, which is reached during the dipole-wall collision, is evaluated and retained for a comparison among boundary conditions. As no analytical result for this flow is known, the measured values are compared against benchmark results obtained with a spectral method in Ref. [24].

In the benchmark, the initial core vorticity of the monopoles is fixed to $\eta_e=299.5286$, which leads to an initial average kinetic energy of $\bar{E}(0)=2$. Furthermore, the Reynolds number and the monopole radius are set to Re=625 and $r_0=0.1$. The lattice velocity is set to $U_{lb}=0.01$ at a reference resolution of $N=300$. The monopoles are aligned symmetrically with the box, in such a way that the dipole-wall collision is frontal and takes place in the middle of the wall. The position of the monopole centers is $(x_1,y_1)=(0,0.1)$ and $(x_2,y_2)=(0,-0.1)$. As in Sec. V C, the approach of Ref. [23] is used to set up the initial condition. The initial pressure is evaluated numerically, by solving the Poisson equation with a successive over-relaxation (SOR) scheme, using an algorithm described, e.g., in Ref. [25]. The off-equilibrium parts of the particle populations are then instantiated with the help of Eq. (23), with numerically computed velocity gradients.

The boundary conditions *BC1* and *BC2* could not be used for this problem, because they experienced numerical instabilities at the required Reynolds number (the benchmark values in the literature start at Re=625). These instabilities are due to an inherent limitation of the boundary conditions, and do not originate from the way the initial condition or the boundary condition in corner nodes are treated. This possibility was ruled out by initializing the population functions with an equilibrium distribution, and keeping the corner

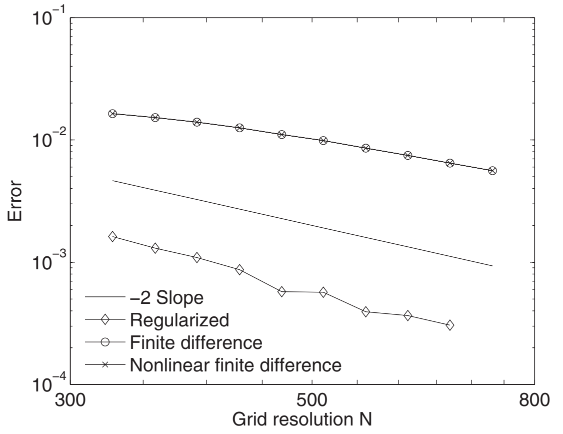

FIG. 7. Accuracy on the value of the enstrophy peak in a 2D dipole-wall collision at Re=625. Boundary conditions *BC1* and *BC2* could not be tested, because they are numerically unstable at this Reynolds number.

nodes at an equilibrium distribution with constant pressure throughout the simulation. Although this approach to setting up initial states and boundaries tends to be exceptionally stable, the simulations using *BC1* and *BC2* were still subject to numerical instabilities.

The numerical results are presented in Fig. 7. Note that a resolution of $N=300$, for example, stands for a total grid size of 600×600, as the size of the system is 2×2 in dimensionless variables. For a Reynolds number Re=625, all three boundary conditions *BC3*, *BC4*, and *BC5*, the error decreases roughly at a second-order rate. The local boundary condition *BC3* is however an order of magnitude more accurate than the other candidates. This is striking, because *BC3* is least accurate in all benchmarks of laminar flows. It is concluded that the local, cell-based approach of *BC3* is particularly well adapted to reproduce the dynamics of a boundary layer. A possible interpretation is that the finite-difference approximation of velocity gradients in *BC4* and *BC5* becomes inaccurate in the presence of a turbulent boundary layer due to small-scale velocity variations.

E. Rectangular steady channel flow (3D)

This problem is a 3D generalization of the channel flow presented in Sec. V A. This time, the channel extends in the z direction. At every fixed value of z, the cross section is rectangular, extending from $x=0$ to $x=L$, and from $y=0$ to $y=\zeta L$, where ζ is the aspect ratio of the cross section. The pressure is independent of the x and the y coordinates, and decreases linearly as a function of z. The velocity has a nonzero component in z direction only, and depends on x and y, $u_z=u_z(x,y)$. As in the 2D case, nonlinear contributions to the Navier-Stokes equation cancel out, and the velocity profile u_z is described as a solution of the following Poisson equation:

$$\nabla^2 u_z \equiv \frac{\partial^2 u_z}{\partial x^2}+\frac{\partial^2 u_z}{\partial y^2}=\frac{\partial p}{\partial z}. \tag{68}$$

As this equation is nonhomogeneous, the general solution is described as the sum of a particular solution u_p and a general solution u_h to the homogeneous problem $\nabla^2 u_z=0$. The particular solution can be taken from the 2D flow in Sec. V A. In dimensionless variables, it reads as

$$u_p(x,y)=4(x-x^2). \tag{69}$$

This settles the choice for the reference velocity U as being the maximum velocity reached in a 2D channel. The pressure drop is the same in the 3D as in the 2D case, $\partial p/\partial z = -8/\mathrm{Re}$. The complete analytic solution to this 3D channel flow is [26]

$$u_z(x,y)=4(x-x^2)+\frac{32}{\pi^3}\sum_{n=0}^{+\infty}\left(\frac{(-1)^n}{(2n+1)^3}\times\cos[(2n+1)\pi x]\frac{\cosh\left((2n+1)\frac{\pi y}{\zeta}\right)}{\cosh\left((2n+1)\frac{\pi}{2}\zeta\right)}\right). \tag{70}$$

Figure 8(a) shows results of numerical accuracy at Re=10. The reference velocity is $U_{lb}=0.01$ at $N=50$, and the aspect ratio is $\zeta=1$. Numerical stability is explored in Fig. 8(b) for a varying grid resolution. The numerical results in this 3D benchmark are similar to those of the 2D application in Sec. V A. The boundary condition *BC1* is again much more accurate than boundary conditions that replace all particle populations. As in the 2D case, all but one component of the strain rate tensor vanish in this flow, and some potential deficiencies of *BC1*, as the one shown in Eq. (35), might not be visible in the benchmark. An unexpected result is that boundary condition *BC2*, which in 2D applications achieves results very similar to those of *BC3*, falls in the same category as the nonlocal boundary conditions *BC4* and *BC5* in this 3D case. Another remarkable difference with respect to 2D results is that the nonlocal boundary conditions achieve unconditional numerical stability in 3D, as soon as the grid resolution exceeds a certain threshold value. At this point, further investigations would be needed to decide if this difference is due to different flow geometries [the analytical solutions in Eqs. (53) and (70) are qualitatively different] or if 3D simulations are inherently more stable than 2D ones for this type of boundary condition.

VI. DISCUSSION AND CONCLUSION

Five boundary conditions for the lattice Boltzmann method have been selected and presented in this paper. As it is concluded from applying the results of a multiscale analysis to boundary nodes, all methods are second-order accurate with respect to the grid resolution. This implies that their error varies asymptotically at the same rate. Beyond this asymptotic estimate, the accuracy experienced in numerical simulation differs however from one boundary condition to another, depending on the flow geometry and the grid resolution. A way to understand these differences might be to take the multiscale analysis to a higher order $O(\epsilon^2)$ and discuss the couplings between higher-order terms and the hy-

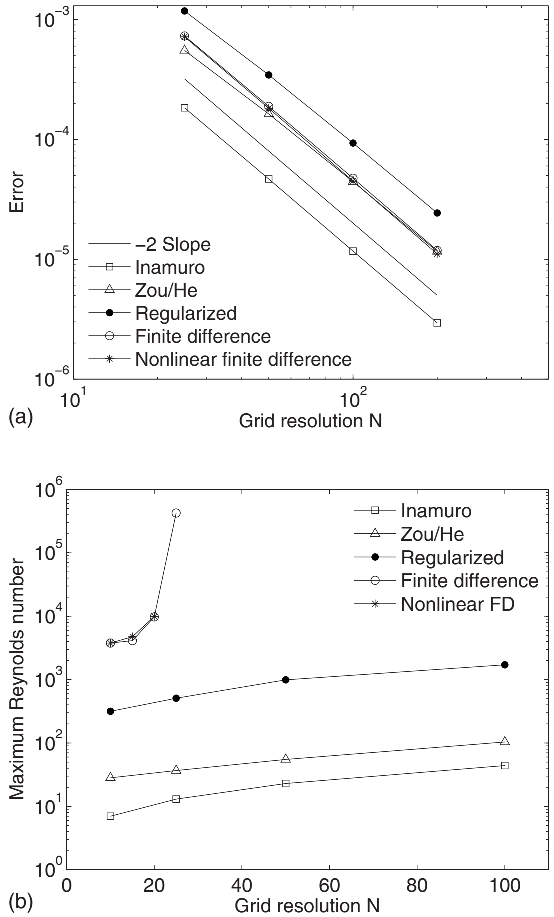

FIG. 8. (a) Numerical accuracy in a 3D channel flow. (b) Numerical stability in a 3D channel flow. The maximum Reynolds number which can be reached before numerical instabilities appear is plotted. Boundary conditions based of finite-difference approximations (*BC4* and *BC5*) are unconditionally stable at high grid resolution. Numerical values are therefore only reported in the unstable regime, up to $N=25$ for *BC4* and up to $N=20$ for *BC5*.

drodynamic scales. This technique is, for example, employed in Ref. [27] to understand the asymptotic low Mach-number behavior of a family of lattice Boltzmann models. Instead of doing this, this paper produces benchmark results for the boundary conditions in many different geometries. Those results are intended to serve as a reference to help select the most appropriate boundary condition for a given lattice Boltzmann simulation. It is emphasized that the results are valid only for straight boundaries which are aligned with the numerical grid. Different conclusions can be expected when off-lattice boundaries are implemented.

In 2D simulations and at low Reynolds numbers, the Inamuro *et al.* and Zou-He boundary conditions, *BC1* and *BC2*, are found to produce the most accurate results. Both of them preserve the known particle populations on a boundary node, a fact which probably explains their excellent benchmark performance. They retrieve a large amount of information from the bulk of the fluid, and manipulate only a few particle populations (3 out of 9). It seems plausible that by doing this, they retain information on higher-order terms which are not visible in the hydrodynamic terms of the multiscale analysis. The Inamuro *et al.* method is also very accurate in the 3D benchmark, but this performance is not reproduced by the Zou-He condition, which in this case compares to nonlocal boundary conditions. The main deficiency of *BC1* and *BC2* is that they are numerically unstable at (even moderately) high Reynolds numbers. They could, for example, not be used to simulate the turbulent dipole-wall collision presented in Sec. V D.

In conclusion, *BC1* and *BC2* are the boundary conditions of choice for the simulation of laminar 2D flows, when high accuracy is important. The Inamuro *et al.* boundary condition *BC1* is also a good candidate for laminar 3D flows, although it is not an explicit method, and thus tends to be complicated to implement. The extension of *BC2* to 3D flow does not seem to make much sense, as it does not exhibit exceptional accuracy, and is less stable than other approaches.

The regularized boundary condition *BC3* uses a hybrid approach, as it is local like *BC1* and *BC2*, but it replaces all particle populations, such as *BC4* and *BC5*. As such, it is less accurate than *BC1* and *BC2* in laminar flows, but it has the ability to reach much higher Reynolds numbers. Furthermore, simulations of a turbulent dipole-wall collision show that it reproduces the dynamics of boundary layers more accurately than all other boundary conditions. Because of its numerical stability, and because it is easy to implement in 2D and 3D applications, *BC3* is a good general-purpose method, and seems to be the best candidate for high Reynolds number flows.

The nonlocal boundary conditions *BC4* and *BC5*, which use a finite-difference scheme to approximate the off-equilibrium part of particle populations, exhibit the best numerical stability. In the laminar 3D benchmark, they are even unconditionally stable on a sufficiently large grid. Their main disadvantage is the nonlocality of their algorithm. This violates the basic principles of the LB method, may substantially increase the complexity of a code, and even be an obstacle to parallelizing the program in a straightforward way. On the other hand, the boundary conditions *BC4* and *BC5* are very general and may adapt well to a larger software project. They are, for example, not bound to the lattice structure, and can be extended to the case of off-lattice boundaries, such as the one described in Ref. [28]. Boundary condition *BC5* takes into account nonlinear velocity terms in off-equilibrium particle populations, which are neglected in *BC4*. Thanks to this, it is slightly more accurate and more stable. The difference is however so small that it does not seem to justify the increased complexity and computational burden.

ACKNOWLEDGMENTS

We thankfully acknowledge the support of the Swiss National Science Foundation (Grants No. FN 200020-1007931, No. PBGe2-117144, and No. FN 200021-107921). We also thank Andreas Malaspinas for inspiring suggestions.

[1] S. Chen, H. Chen, D. Martinez, and W. Matthaeus, Phys. Rev. Lett. **67**, 3776 (1991).

[2] S. Chen and G. D. Doolen, Annu. Rev. Fluid Mech. **30**, 329 (1998).

[3] J. Latt, Ph.D. dissertation, University of Geneva, 2007, http://www.unige.ch/cyberdocuments/theses2007/LattJ/meta.html

[4] M. Junk, A. Klar, and L.-S. Luo, J. Comput. Phys. **210**, 676 (2005).

[5] Y. Qian, D. d'Humières, and P. Lallemand, Europhys. Lett. **17**, 479 (1992).

[6] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond* (Oxford University Press, Oxford, 2001).

[7] X. He and L.-S. Luo, Phys. Rev. E **56**, 6811 (1997).

[8] X. He, X. Shan, and G. D. Doolen, Phys. Rev. E **57**, R13 (1998).

[9] I. Halliday, L. A. Hammond, and C. M. Care, J. Phys. A **35**, L157 (2002).

[10] B. Chopard and A. Dupuis, Int. J. Mod. Phys. B **17**, 103 (2003).

[11] T. Inamuro, M. Yoshina, and F. Ogino, Phys. Fluids **7**, 2928 (1995).

[12] Q. Zou and X. He, Phys. Fluids **9**, 1591 (1997).

[13] P. A. Skordos, Phys. Rev. E **48**, 4823 (1993).

[14] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C++: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 2002).

[15] D. Hänel, *Einfuehrung in die kinetische Theorie der Gase und Lattice-Boltzmann-Methoden* (Springer-Verlag, Berlin, 2004).

[16] P. J. Dellar, Phys. Rev. E **64**, 031203 (2001).

[17] Z. Guo, C. Zheng, and B. Shi, Phys. Rev. E **65**, 046308 (2002).

[18] LB boundary condition benchmarks, http://www.lbmethod.org/bcbench.html

[19] The OpenLB project, http://www.lbmethod.org/openlb/

[20] J. R. Womersley, J. Physiol. **127**, 553 (1955).

[21] J. A. Cosgrove, J. M. Buick, S. J. Tonge, C. G. Munro, C. A. Greated, and D. M. Campbell, J. Phys. A **36**, 2609 (2003).

[22] O. Walsh, in *The Navier-Stokes Equations II—Theory and Numerical Methods*, edited by J. G. Heywood, K. Masuda, R. Rautmann, and S. A. Solonnikov, Lecture Notes in Mathematics, Vol. 1520 (Springer-Verlag, Berlin, 1991), pp. 305–309.

[23] J. Latt and B. Chopard, Int. J. Mod. Phys. C **18**, 619 (2007).

[24] H. J. H. Clercx and C.-H. Bruneau, Comput. Fluids **35**, 245 (2006).

[25] M. Griebel, T. Dornseifer, and T. Neunhoffer, *Numerical Simulation in Fluid Dynamics* (SIAM, New York, 1998).

[26] C.-S. Yih, *Fluid Mechanics* (McGraw-Hill, New York, 1969).

[27] P. J. Dellar, J. Comput. Phys. **190**, 351 (2003).

[28] Z. Guo and C. Zheng, Phys. Fluids **14**, 2007 (2002).