General velocity, pressure, and initial condition for two-dimensional and three-dimensional lattice Boltzmann simulations

Omid Reza Mohammadipour[*]
Mechanical Engineering Department, Payame Noor University, P.O. Box 19395-3697, Tehran, Iran

Hamid Niazmand
Mechanical Engineering Department, Ferdowsi University of Mashhad, P.O. Box 91775-1111, Azadi Square, Mashhad, Iran

Sauro Succi
Istituto Applicazioni Calcolo, Via dei Taurini 19, 00185 Roma, Italy
and Institute of Applied Computational Science, John A. Paulson School of Engineering and Applied Sciences,
Harvard University, 29 Oxford Street, Cambridge, Massachusetts 02138, USA

(Received 22 September 2016; revised manuscript received 17 January 2017; published 6 March 2017)

In this paper, an alternative approach to implement initial and boundary conditions in the lattice Boltzmann method is presented. The main idea is to approximate the nonequilibrium component of distribution functions as a third-order power series in the lattice velocities and formulate a procedure to determine boundary node distributions by using fluid variables, consistent with such an expansion. The velocity shift associated with the body force effects is included in this scheme, along with an approximation to determine the mass density in complex geometries. Different strategies based on the present scheme are developed to implement velocity and pressure conditions for arbitrarily shaped boundaries, using the D2Q9, D3Q15, D3Q19 and D3Q27 lattices, in two and three space dimensions, respectively. The proposed treatment is tested against several well-established problems, showing second-order spatial accuracy and often improved behavior as compared to various existing methods, with no appreciable computational overhead.

DOI: 10.1103/PhysRevE.95.033301

I. INTRODUCTION

Since its introduction nearly three decades ago [1,2], the lattice Boltzmann (LB) method has attracted considerable attention as an alternative numerical method for simulating a broad variety of complex flows [3]. At variance with mainstream computational fluid dynamics methods, where macroscopic variables such as velocity, pressure, and density are obtained by solving the Navier-Stokes equations, the LB method solves the kinetic equation for particle distribution functions [4]. The macroscopic variables are then obtained by evaluating the hydrodynamic moments of the distribution functions. However, the inverse mapping from the macroscopic variables to the distribution functions is somewhat tricky, especially at boundary nodes, because it requires more unknowns than available from macroscopic hydrodynamics.

The LB boundary condition (BC) treatments that are introduced to supply these unknowns can be divided into two main families, the on-lattice and the so-called off-lattice boundary conditions. For the former, the wall (solid-fluid interface) is located on grid nodes and boundary values are directly assigned to such boundary nodes, whereas for the latter, the wall lies somewhere between grid nodes and boundary values must be adjusted to provide the desired conditions at the wall location.

Most of commonly used boundary conditions, including the original bounceback method [5,6], the nonequilibrium bounceback scheme [7], the local thermohydrodynamic equilibrium assumption for the missing distribution functions [8], the extrapolation scheme [9,10], Grad's approximation [11], and the regularized method [12], can be categorized as on-lattice boundary treatments. Nevertheless, with the aid of suitable interpolation and extrapolation procedures, some of these on-lattice BCs are also amenable to an off-lattice boundary treatment [13–19]. The main advantage of the off-lattice approach is a more accurate geometrical representation of the boundary at a given spatial resolution. The downside is an increase of complexity in the formulation and implementation, along with a potential loss of accuracy due to the interpolation and extrapolation procedures required to transfer information from the bulk to the boundary nodes and vice versa.

Despite the substantial body of remarkable work deployed to improve the accuracy of on-lattice boundary conditions with general geometries (see, for instance, [5,10–12]), some drawbacks still remain. The bounceback rule, as combined with single-relaxation-time formulations of the LB scheme, generally yields a nonzero slip velocity that depends on viscosity [6]. This error exists not only for classical bounceback method, but also for halfway bounceback and even for the linear and quadratic interpolation methods based on the bounceback rule [20]. The nonequilibrium extrapolation scheme [10] and Grad's approximation method [11] fill the unknown distribution functions with appropriately approximated values and preserve the known distributions intact. Since no velocity enforcement procedure is applied in these approaches, the imposed velocity value is close to the desired one at the boundary node, although not exactly the same. The regularized method [12] replaces all (known and unknown) distribution functions at the boundary node by approximated values. This approach enforces the velocity

[*] o.mohammadipour@pnu.ac.ir; o.r.mohammadipour@gmail.com

boundary node exactly to the desired value, but it loses some information due to replacement of the known information with an approximate one. Furthermore, since this BC initializes unknown distributions by applying the bounceback rule on nonequilibrium distributions, to evaluate the stress tensor, it requires also the nonequilibrium distributions moving along opposite directions, which may not be available in some geometries, such as two-dimensional (2D) concave corners or simple edges in 3D geometries.

In addition, there is a wide range of fluid problems in which an external or internal force is involved, whose implementation is best accomplished by shifting the velocity field in the expression of the local equilibria [21]. A common drawback for commonly adopted boundary approaches is that this velocity shift is not considered in the treatment of the boundary information.

The present study introduces an on-lattice BC scheme to implement velocity, pressure, and initial conditions in the LB simulations, based on a formulation for density and unknown distributions in the absence or presence of body forces. It replaces all distribution functions on the boundary node, to enforce the exact desired velocity values, while resorting to a higher-order term in the distribution function, to reduce the information loss during the redistribution step. The present scheme can be applied to any arbitrary geometrical shape (including corners and edges), using the D2Q9, D3Q15, D3Q19, and D3Q27 lattices, without any substantial computational overhead as compared to the existing approaches.

The rest of the paper is organized as follows. In Sec. II the basics of the lattice Boltzmann are reviewed. In Sec. III the boundary treatment is discussed in detail. In Sec. IV different strategies to employ the treatment are summarized and in Sec. V benchmark tests are carried out to assess the accuracy and stability of the scheme. A summary and some conclusions are provided in Sec. VI.

II. LATTICE BOLTZMANN METHOD

Although the lattice Boltzmann equation originated from the lattice gas automata method [22,23], it can be viewed as a special finite-difference form of the continuous Boltzmann kinetic equation [24]. According to the lattice Boltzmann equation, the evolution of the distribution function f for q discrete velocities is given by

$$\frac{\partial f_i(\mathbf{r},t)}{\partial t} + \mathbf{e}_i \cdot \nabla f_i(\mathbf{r},t) = \Omega_i, \quad i = 0,1,\ldots,q-1, \tag{1}$$

where Ω is the collision operator. To solve the distribution function numerically, Eq. (1) is completely discretized with the time step δ_t and lattice space δ_x as

$$f_i(\mathbf{r}+\mathbf{e}_i\delta_t, t+\delta_t) - f_i(\mathbf{r},t) = \Omega_i. \tag{2}$$

In this paper the lattice Bhatnagar-Gross-Krook model [25] is used, which approximates the collision operator as a relaxation of the distribution function f towards a local equilibrium distribution f^{eq},

$$\Omega_i = \frac{1}{\tau}\left[f_i^{\text{eq}}(\mathbf{r},t) - f_i(\mathbf{r},t)\right] + \delta_t G_i, \tag{3}$$

where τ is a dimensionless relaxation time related to the kinetic viscosity of the fluid and G_i is the forcing term. The equilibrium distribution $f^{\text{eq}}$is a low-Mach-number expansion of the Maxwell-Boltzmann distribution and is written as [24]

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

where $\mathbf{u}$ is the macroscopic velocity vector, w_i is the weight factor, ρ is the mass density, $c_s = c/\sqrt{3}$ is the speed of sound, and $c = \delta_x/\delta_t$ is the particle streaming speed. There are many published methods [21,26–28] to deal with body force in the LB method. Here we use one of the most accurate ones introduced by Guo *et al.* [21], where the forcing term is defined as

$$G_i = w_i\left(1 - \frac{1}{2\tau}\right)\left[\frac{\mathbf{e}_i - \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i\cdot\mathbf{u})\mathbf{e}_i}{c_s^4}\right]\cdot\mathbf{F}, \tag{5}$$

where $\mathbf{F}$ is the macroscopic body-force vector. The macroscopic quantities in the lattice Boltzmann method such as mass density, velocity, and pressure are defined as moments of the distribution function

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

$$u_\alpha = \frac{1}{\rho}\left(\sum_i f_i e_{i\alpha} + \frac{\delta_t}{2}F_\alpha\right), \tag{7}$$

$$p = \rho c_s^2. \tag{8}$$

Through the Chapman-Enskog multiscale analysis, the kinematic viscosity ν is related to the dimensionless relaxation time as follows [29]:

$$\upsilon = \left(\tau - \tfrac{1}{2}\right)c_s^2\delta_t. \tag{9}$$

Further information about the LB theory and its applications to complex fluid systems can be found in [3,20,29–33].

III. GENERAL BOUNDARY CONDITION

The evolution of the particle distribution functions using Eq. (2) can be split into two basic steps: collision and streaming. The collision step takes the particle populations f_i to their postcollision value $\tilde{f}_i$ using the following equation:

$$\tilde{f}_i(\mathbf{r},t) = f_i(\mathbf{r},t) + \frac{1}{\tau}\left[f_i^{\text{eq}}(\mathbf{r},t) - f_i(\mathbf{r},t)\right] + \delta_t G_i. \tag{10}$$

The collision is then followed by a streaming step, which takes the postcollision distribution to a neighbor node corresponding to its lattice vector:

$$f_i(\mathbf{r}+\mathbf{e}_i\delta_t, t+\delta_t) = \tilde{f}_i(\mathbf{r},t). \tag{11}$$

On boundary nodes, the streaming step brings some unknown distribution functions into the fluid domain from the solid region. Boundary formulations in the LB method are mainly focused on the calculation of these unknown distribution functions in terms of known ones.

The main idea in this study is to conduct an approximation for the distribution function in general cases (in two and three dimensions), independent of the boundary shape and the number of unknown distribution functions. In this regard, we start our discussion from the Chapman-Enskog multiscale analysis [22] and skip unnecessary details for the sake of simplicity. Based on the Chapman-Enskog multiscale analysis, a distribution function can be expanded as a power-law series with respect to a small quantity $\varepsilon = O(\delta_t)$ as follows:

$$f = \sum_{n=0} \varepsilon^n f^{(n)} = f^{(0)} + \varepsilon f^{(1)} + O(\varepsilon^2). \tag{12}$$

The first-order term of this series is the equilibrium distribution $f^{\rm eq}$ and the following terms are taken as the nonequilibrium part, denoted by $f^{\rm neq}$,

$$f^{\rm neq} = \varepsilon f^{(1)} + O(\varepsilon^2). \tag{13}$$

According to Eq. (4), the equilibrium distribution $f^{\rm eq}$ depends on macroscopic variables ρ and $\mathbf{u}$. However, there is no finite-order expression to determine the nonequilibrium component in terms of the local equilibrium; to this purpose an infinite series in the Chapman-Enskog expansion should be carried out.

Latt *et al.* [12] introduced a symmetric approximation for the nonequilibrium, which is consistent with Grad's approximation, namely [11],

$$f_i^{\rm neq} \approx -\frac{w_i \tau}{c_s^2}\left(\mathbf{e}_i\mathbf{e}_i - c_s^2\mathbf{I}\right) : \rho\nabla\mathbf{u}. \tag{14}$$

Instead, we introduce a general approximation for the nonequilibrium part, based on a fourth-order power series in the lattice velocities:

$$\begin{aligned} f_i^{\rm neq} \approx\ & w_i[A + e_{i\alpha}B_\alpha + e_{i\alpha}e_{i\beta}C_{\alpha\beta} + e_{i\alpha}e_{i\beta}e_{i\gamma}D_{\alpha\beta\gamma} \\ & + e_{i\alpha}e_{i\beta}e_{i\gamma}e_{i\delta}E_{\alpha\beta\gamma\delta}], \end{aligned} \tag{15}$$

where the unknown tensor coefficients $(A,\mathbf{B},\ldots)$ are determined by requiring that the moments of $f^{\rm neq}$ should be consistent with the hydrodynamic equations.

For symmetric lattices such as D2Q9, D3Q15, D3Q19, and D3Q27, the following relations are satisfied:

$$\begin{gathered} \sum_i w_i = 1, \\ \sum_i w_i e_{i\alpha}e_{i\beta} = c_s^2\delta_{\alpha\beta}, \\ \sum_i w_i e_{i\alpha}e_{i\beta}e_{i\gamma}e_{i\delta} = c_s^4(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}); \end{gathered} \tag{16}$$

$$\begin{gathered} \sum_i w_i e_{i\alpha} = 0, \\ \sum_i w_i e_{i\alpha}e_{i\beta}e_{i\gamma} = 0, \\ \sum_i w_i e_{i\alpha}e_{i\beta}e_{i\gamma}e_{i\delta}e_{i\lambda} = 0. \end{gathered} \tag{17}$$

After some calculation using Eqs. (2), (4), (5), (12), (16), and (17), one can obtain the zeroth to second moments of the nonequilibrium distribution functions as follows (see Appendix A for the detailed mathematical derivation):

$$\sum_i f_i^{\rm neq} = 0, \tag{18}$$

$$\sum_i f_i^{\rm neq} e_{i\alpha} = -\frac{\delta_t}{2}F_\alpha, \tag{19}$$

$$\begin{aligned} \sum_i f_i^{\rm neq} e_{i\alpha}e_{i\beta} =\ & -\frac{\delta_t}{2}[u_\alpha F_\beta + u_\beta F_\alpha] \\ & - \tau c_s^2 \delta_t \rho\left(\frac{\partial u_\alpha}{\partial\beta} + \frac{\partial u_\beta}{\partial\alpha}\right). \end{aligned} \tag{20}$$

For a d-dimensional problem, Eqs. (18)–(20) provide 1, d, and $d(d+1)/2$ equations, respectively. These relations fix only the first three tensors in Eq. (15) including A, $\mathbf{B}$, and $\mathbf{C}$, leaving the following tensors unknown, calling for a closure approximation. The simplest such closure is to ignore these tensors by setting them equal to zero, but it is clearly desirable to devise better closures such as to exploit such tensors to minimize the information loss inherent to the redistribution process. For the moment, we keep these tensors in the form of a distribution function denoted by h_i (h standing for higher-order):

$$h_i = w_i(e_{i\alpha}e_{i\beta}e_{i\gamma}D_{\alpha\beta\gamma} + e_{i\alpha}e_{i\beta}e_{i\gamma}e_{i\delta}E_{\alpha\beta\gamma\delta}). \tag{21}$$

Note that as long as the zeroth to second moments of the function h_i is zero, the tensors A, $\mathbf{B}$, and $\mathbf{C}$ can be determined explicitly as functions of the macroscopic properties. In this regard, the approximation of the nonequilibrium part [Eq. (15)] can be rewritten in the following form:

$$\begin{gathered} f_i^{\rm neq} \approx w_i[A + e_{i\delta}B_\delta + e_{i\delta}e_{i\gamma}C_{\delta\gamma}] + h_i \\ \text{where } \sum_i h_i = \sum_i h_i e_{i\alpha} = \sum_i h_i e_{i\alpha}e_{i\beta} = 0. \end{gathered} \tag{22}$$

Substituting Eq. (22) into Eqs. (18)–(20) yields

$$A = -c_s^2 C_{\alpha\alpha}, \tag{23}$$

$$B_\alpha = -\frac{\delta_t}{2c_s^2}F_\alpha, \tag{24}$$

$$C_{\alpha\beta} = -\frac{\delta_t}{4c_s^4}[u_\alpha F_\beta + u_\beta F_\alpha] - \frac{\tau\delta_t\rho}{2c_s^2}\left(\frac{\partial u_\alpha}{\partial\beta} + \frac{\partial u_\beta}{\partial\alpha}\right). \tag{25}$$

So far, the approximation for the nonequilibrium part of distribution function can be expressed as

$$f_i^{\rm neq} \approx w_i\left(e_{i\alpha}e_{i\beta} - c_s^2\delta_{\alpha\beta}\right)C_{\alpha\beta} - \delta_t w_i\frac{e_{i\alpha}F_\alpha}{2c_s^2} + h_i. \tag{26}$$

In this approximation, $C_{\alpha\beta}$ is related to the strain rate tensor and symmetric effects of body forces by Eq. (25). The second term, representing the antisymmetric effects of body forces and a rearrangement of Eq. (26), indicates that the function $h_i$is of $O(\varepsilon^2)$ in the distribution function expansion

$$\begin{aligned} f_i^{\rm neq} &= \varepsilon[f^{(1)}] + O(\varepsilon^2) \\ &\approx \delta_t\left[w_i\left(e_{i\alpha}e_{i\beta} - c_s^2\delta_{\alpha\beta}\right)\frac{C_{\alpha\beta}}{\delta_t} - w_i\frac{e_{i\alpha}F_\alpha}{2c_s^2}\right] + h_i. \end{aligned}$$

For known values of ρ, $C_{\alpha\beta}$, and h_i, unknown distribution functions on the boundary node located at $\mathbf{r}_b$ can be calculated by summing up the equilibrium and nonequilibrium distributions as follows:

$$\begin{aligned} f_i(\mathbf{r}_b,t) = f_i^{\rm eq}(\mathbf{r}_b,t) + w_i\left[\left(e_{i\alpha}e_{i\beta} - c_s^2\delta_{\alpha\beta}\right)C_{\alpha\beta} - \frac{\delta_t}{2c_s^2}e_{i\alpha}F_\alpha\right] \\ + h_i(\mathbf{r}_b,t). \end{aligned} \tag{27}$$

It should be observed that filling the unknown distributions by approximated values [Eq. (27)] does not necessarily lead to a no-slip condition. In fact, in order to enforce the desired velocity on boundary nodes, Eq. (27) must be applied in all, known and unknown, directions. During this redistribution, some known information is lost because known distributions are replaced by approximated values.

This information loss can be formally expressed as follows:

$$f_i^{(\text{loss})} = f_{i(\text{known})} - f_{i(\text{approximated})}.$$

When the approximated values are up to first order in Knudsen number ($f^{\text{neq}} \approx \varepsilon f^{(1)}$) as in Grad's approximation and the regularized method, the information loss is second order in the Knudsen number [$f_i^{(\text{loss})} = O(\varepsilon^2)$]. In our scheme, h_i is included in the final approximation as a higher term in the Knudsen number.

In this regard, the information loss during the redistribution step in the present scheme is expected to be lower than in the regularized method:

$$\begin{aligned} f_i^{(\text{loss})} &= \left[f_i^{(0)} + \varepsilon f_i^{(1)} + O(\varepsilon^2)\right]_{\text{known}} \\ &\quad - \left(f_i^{(0)} + \varepsilon f_i^{(1)} + h_i\right)_{\text{approximated}} = O(\varepsilon^2)_{\text{known}} - h_i. \end{aligned} \tag{28}$$

The exact value of ρ in the definition of f^{eq} and $C_{\alpha\beta}$, and the h_i function, would of course completely avoid any loss of information during redistribution step. Accordingly, by using an approximation value for h_i, one can reduce the information loss (not necessarily to zero, though) for the known distributions. The rest of this section describes how to determine ρ, $C_{\alpha\beta}$, and h_i, one by one.

A. Determination of mass density on boundary nodes

According to Eq. (4), the equilibrium distribution f^{eq} depends on macroscopic variables ρ and $\mathbf{u}$. For no-slip BCs, the density ρ is unknown, due to a lack of information about the boundary nodes. As discussed in previous studies [7,12], the value of the density on straight walls can be determined from the known distribution functions and the boundary velocity values. However, the available information about a boundary node may not be sufficient for the evaluation of the density on more complex geometries.

In these situations, a common approach is to extrapolate the density from the neighboring fluid node [10,12]. Note, however, that this extrapolation enforces a zero-pressure gradient along the extrapolation direction, which is obviously not valid whenever the body force features a nonzero component along the extrapolation direction.

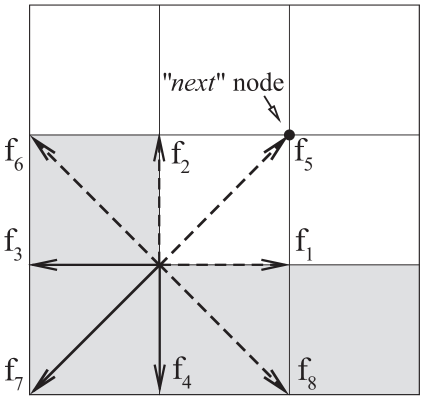

FIG. 1. Layout of distribution functions for a typical boundary node on a wall in the D2Q9 lattice.

In this study, an approximation of mass density is introduced, which is independent of boundary shape, with or without body forces, and can be regarded as an extension of the local density approach described in a previous paper [18]. We first define three sets of indices and one superscript: the known indices I_K, which index lattice directions with known distribution functions; the unknown-known indices I_{UK}, which index directions with unknown distribution functions, where the distributions in the opposite directions are known; the unknown-unknown indices I_{UU}, which index directions with unknown distribution functions, where the distributions in the opposite directions are also unknown; and finally the next superscript *next*, which denotes the fluid node nearest the boundary node. Even though the present boundary scheme is compatible with 2D and 3D lattices, in Fig. 1 we show specifications of the above indices for a simple 2D case, namely, a concave corner boundary node, with five unknown distribution functions entering the fluid domain from the solid region, depicted as dashed vectors in Fig. 1.

For this typical boundary node the three indices are defined as follows:

$$I_K = \{0,3,4,7\}, \quad I_{UK} = \{2,1,5\}, \quad I_{UU} = \{6,8\}.$$

A detailed discussion to determine the *next* node is presented in Appendix C. As long as the boundary shape is fixed, these indices and the *next* location are computed only once for each boundary node during the initialization procedure and then they are available to use in the simulation. By temporarily ignoring the h_i values and performing some algebra, the local density for each boundary node can be calculated as follows (see Appendix B for the detailed mathematical derivation):

$$\rho = \frac{\sum_{i\in I_K} f_i + \sum_{i\in I_{UK}} f_{\text{opp}(i)} + \sum_{i\in I_{UU}} f_i^{(\text{next})} - \frac{\delta_t}{c_s^2}\sum_{i\in I_{UK}} w_i e_{i\alpha} F_\alpha}{1 - \frac{2}{c_s^2}\sum_{i\in I_{UK}} w_i e_{i\alpha} u_\alpha}, \tag{29}$$

where the subscript *opp*(i) denotes the opposite of the i direction.

B. Determination of $C_{\alpha\beta}$ on boundary nodes

According to Eq. (25), the calculation of the **C** tensor requires information about the density, body force, velocity, and velocity gradients. Density, body-force, and velocity values are available by using Eq. (29) and boundary information. Velocity gradients can also be determined by using velocity values of neighboring nodes and a suitable finite-difference scheme. Considering that evaluation of $C_{\alpha\beta}$ in a d-dimensional geometry requires gradients of d velocity components in the d direction, one readily concludes that determination of the **C** tensor by using Eq. (25) is computationally expensive.

Next we present a simpler way to determine this tensor based on the knowledge of the known information. Note that for any node in the computational domain, the $C_{\alpha\beta}$can be calculated as follows:

$$\sum_i f_i e_{i\alpha} e_{i\beta} = \sum_i f_i^{\text{eq}} e_{i\alpha} e_{i\beta} + \sum_i f_i^{\text{neq}} e_{i\alpha} e_{i\beta},$$

$$\sum_i f_i e_{i\alpha} e_{i\beta} = \left(\rho c_s^2 \delta_{\alpha\beta} + \rho u_\alpha u_\beta\right) + \left(2c_s^4 C_{\alpha\beta}\right),$$

$$C_{\alpha\beta} = \frac{\sum_i f_i e_{i\alpha} e_{i\beta} - \rho c_s^2 \delta_{\alpha\beta} - \rho u_\alpha u_\beta}{2c_s^4}. \tag{30}$$

For a boundary node, the density is known by Eq. (29) and the velocity components are given by the no-slip boundary conditions. If one assumes a series of initial approximations for all unknown distribution functions, $C_{\alpha\beta}$ can then be determined by using Eq. (30). Note that, according to Eqs. (16), (17), and (26), the first moment of f^{neq} (which is involved in macroscopic velocity definition) is independent of the **C** tensor:

$$\sum_i f_i^{\text{neq}} e_{i\gamma} = \sum_i w_i e_{i\gamma} \left[\left(e_{i\alpha} e_{i\beta} - c_s^2 \delta_{\alpha\beta}\right) C_{\alpha\beta} + -\frac{\delta_t}{2c_s^2} e_{i\alpha} F_\alpha\right]$$
$$+ \sum_i h_i e_{i\gamma} = -\frac{\delta_t}{2} F_\gamma.$$

Therefore, the **C** tensor does not directly affect the velocity value and vice versa. In this regard, a series of rough approximations for unknown distributions called initial approximations are good enough to determine the $C_{\alpha\beta}$ values even if these approximations do not completely satisfy the no-slip boundary condition. Latt *et al.* [12] suggested bounceback of nonequilibrium parts as initial approximations. There may be two problems with this approach. First, applying the bounceback rule on the nonequilibrium part of unknown distributions leads to a symmetric distribution for the nonequilibrium parts along the bounceback direction, whereas, according to Eq. (26), the nonequilibrium distribution is generally a combination of symmetric and antisymmetric components. Even if the h_i function is ignored, the body-force effects would result in an antisymmetric distribution in the nonequilibrium part. Second, to perform the bounceback rule, the distribution in the opposite direction must be known. This information may not be available in all geometries (e.g., the boundary node depicted in Fig. 1).

Here an approach is introduced for the initial approximation, which applies to any arbitrary geometry and also accounts for antisymmetric effects of body forces:

$$f_i = (1+\tau) f_i^{\text{eq}} - \tau f_i^{\text{eq(next)}}$$
$$- \frac{\delta_t}{2c_s^2} w_i e_{i\alpha} F_\alpha \quad \text{for } i \in I_{UK},\ i \in I_{UU}. \tag{31}$$

After assuming the above approximation for the unknown distribution functions, the **C** tensor can consequently be determined by using Eq. (30).

C. Determination of h_i on boundary nodes

According to Eq. (21), the h_i function can be decomposed into a antisymmetric part denoted by Γ_i and symmetric part denoted by Δ_i:

$$h_i = \Gamma_i + \Delta_i = w_i e_{i\alpha} e_{i\beta} e_{i\gamma} D_{\alpha\beta\gamma} + w_i e_{i\alpha} e_{i\beta} e_{i\gamma} e_{i\delta} E_{\alpha\beta\gamma\delta}. \tag{32}$$

For a fluid node where all distribution functions are known, Γ_i and Δ_i can be determined by using Eqs. (4) and (26) as follows:

$$\Gamma_i = \frac{f_i - f_{\text{opp}(i)}}{2} - \frac{w_i \rho}{c_s^2} u_\alpha e_{i\alpha} + \frac{\delta_t}{2c_s^2} w_i e_{i\alpha} F_\alpha,$$

$$\Delta_i = \frac{f_i + f_{\text{opp}(i)}}{2} - w_i \left[\rho + \left(e_{i\alpha} e_{i\beta} - c_s^2 \delta_{\alpha\beta}\right) \left(\rho \frac{u_\alpha u_\beta}{2c_s^4} + C_{\alpha\beta}\right)\right].$$

It is easy to show that the zeroth to second moments of Γ_i and Δ_i are zero and consequently the h_i as defined in Eq. (32) is compatible with the nonequilibrium approximation in Eq. (22).

Here we choose to extrapolate Γ_i and Δ_i values at boundary nodes from the *next* node:

$$\Gamma_i(\mathbf{r}_b,t) \approx \frac{f_i^{\text{(next)}} - f_{\text{opp}(i)}^{\text{(next)}}}{2} - \frac{w_i \rho^{\text{(next)}}}{c_s^2} u_\alpha^{\text{(next)}} e_{i\alpha}$$
$$+ \frac{\delta_t}{2c_s^2} w_i e_{i\alpha} F_\alpha^{\text{(next)}}, \tag{33}$$

$$\Delta_i(\mathbf{r}_b,t) \approx \frac{f_i^{\text{(next)}} + f_{\text{opp}(i)}^{\text{(next)}}}{2} - w_i \left[\rho^{\text{(next)}} + \left(e_{i\alpha} e_{i\beta} - c_s^2 \delta_{\alpha\beta}\right) \left(\rho^{\text{(next)}} \frac{u_\alpha^{\text{(next)}} u_\beta^{\text{(next)}}}{2c_s^4} + C_{\alpha\beta}^{\text{(next)}}\right)\right]. \tag{34}$$

So far, we approximated the symmetric and antisymmetric parts of the h_i, but it is not clear how this approximated value would affect the accuracy of the present scheme. To this purpose, three different cases, including body-force-driven flow in a straight channel, pressure-driven flow in an inclined channel, and flow between two rotating circular cylinders, are simulated by using the present scheme when the h_i is chosen to be $h_i = 0$, $h_i = \Gamma_i$, and $h_i = \Gamma_i + \Delta_i$, respectively. For now, we focus on h_i effects on overall accuracy and postpone other details until the next section. The corresponding relative errors of the velocity field for these three cases are reported in Fig. 2 as a function of the Reynolds number. For the straight channel, nonzero values of h_i can considerably (about 40%) enhance the accuracy of simulations. Since the h_i distribution

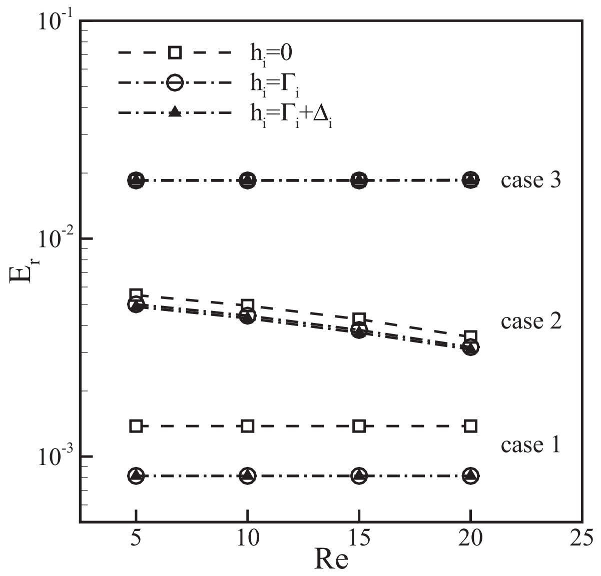

FIG. 2. Relative error in the velocity field as a function of the Reynolds number when h_i is chosen to be $h_i = 0$, $h_i = \Gamma_i$, and $h_i = \Gamma_i + \Delta_i$. Points from bottom to top are numerical results for the body-force-driven flow in a straight channel where the grid resolution along the channel width is $N_R = 20$, pressure-driven flow in an inclined channel where the grid resolution is $N_R = 80$ and the slope of the boundaries is $1/3$, and flow between two rotating circular cylinders where the grid resolution along the inner cylinder radius is $N_R = 20$ and radius ratio is $\beta = 1/2$.

does not affect the nonequilibrium moments in Eqs. (18)–(20), this enhancement can only be associated with a reduction in the information loss during redistribution step, due to the h_i distribution function.

Note that besides the h_i function, the information loss can also be affected by the density value used in the $f^{\mathrm{eq}}$and $C_{\alpha\beta}$ definitions. In other words, any decreases in the accuracy of the density approximation may reduce recovering effects of the h_i function. For example, in the pressure-driven flow in an inclined channel, where the accuracy of the density approximation is affected by the stair-shaped nature of the flow geometry, nonzero values of h_i can enhance the simulation accuracy only about 10%. For more critical conditions such as flow between two rotating cylinders, where the density approximation is affected by the stair-shaped geometry along a potential mass leakage due to centripetal force, the effect of nonzero values of h_i on overall accuracy is not considerable.

Numerical results in Fig. 2 also show that the effects of Δ_i on the overall accuracy in all cases is rather negligible. In this regard, the h_i distribution function can be approximated by its antisymmetric component

$$h_i(\mathbf{r}_b,t) \approx \frac{f_i^{(\mathrm{next})} - f_{\mathrm{opp}(i)}^{(\mathrm{next})}}{2} - \frac{w_i \rho^{(\mathrm{next})}}{c_s^2} u_\alpha^{(\mathrm{next})} e_{i\alpha} + \frac{\delta_t}{2c_s^2} w_i e_{i\alpha} F_\alpha^{(\mathrm{next})}. \qquad (35)$$

The above discussion indicates that the benefits due to the h_i component are flow dependent and hence must be assessed case by case. Routes to further improvement are beyond the scope of the present work.

IV. BOUNDARY CONDITION ALGORITHMS

In this section we introduce various strategies to apply the present scheme as the velocity, pressure, or initial condition in LB simulations.

A. No-slip boundary condition

The implementation of the present scheme as a no-slip BC is summarized as follows.

Step 1. Compute the mass density using Eq. (29).

Step 2. Assume an initial approximation for all unknown distributions using Eq. (31).

Step 3. Determine the **C** tensor using Eq. (30).

Step 4. Determine the h_i function using Eq. (35).

Step 5. Redistribute all distribution functions on boundary nodes using Eq. (27).

B. Pressure boundary condition

Pressure in the lattice Boltzmann method is directly connected to density by Eq. (8). Consequently, for a pressure boundary condition, the mass density is known whereas the velocity components are unknowns. In this situation, the velocity value of the *next* node can be used as an approximation for the boundary velocity:

$$\mathbf{u}(\mathbf{r}_b,t) = \mathbf{u}(\mathbf{r}_{\mathrm{next}},t). \qquad (36)$$

After determining the boundary velocity, the boundary treatment is similar to the no-slip condition. Steps to apply a pressure boundary condition are as follows.

Step 1. Compute the mass density using Eq. (8).

Step 2. Calculate the boundary velocity using Eq. (36).

Step 3. Assume an initial approximation for all unknown distributions using Eq. (31).

Step 4. Determine the **C** tensor using Eq. (30).

Step 5. Determine the h_i function using Eq. (35).

Step 6. Redistribute all distribution functions on boundary nodes using Eq. (27).

C. Initial condition

The accurate initialization of the fluid flow is an important step in numerical simulations, specifically for time-dependent flows. In these situations, the velocity field and density (pressure) distribution are known and the distribution functions at all nodes in the computational domain need to be determined. For this condition the following algorithm is suggested.

Step 1. Calculate the equilibrium distribution functions based on velocity and density values using Eq. (4).

Step 2. Determine the velocity gradients using an appropriate finite-difference scheme.

Step 3. Determine the **C** tensor using Eq. (25).

Step 4. Initialize the distribution function for all nodes in the computational domain using the following equation:

$$f_i(\mathbf{r},t) = f_i^{\mathrm{eq}}(\mathbf{r},t) + w_i\left[\left(e_{i\alpha}e_{i\beta} - c_s^2\delta_{\alpha\beta}\right)C_{\alpha\beta} - \frac{\delta_t}{2c_s^2} e_{i\alpha}F_\alpha\right]. \qquad (37)$$

Note that for an initial-value problem there is no equation to calculate the h_i function in terms of macroscopic quantities. In this regard, the h_i function is ignored in Eq. (37). If the pressure distribution is not available, extra calculations may be required to determine the density field. An example for this situation is discussed in detail in the next section.

D. Curved boundaries

As mentioned before, the present scheme is an on-lattice boundary treatment that approximates the curved boundaries as a series of stair steps. This is a rough approach for curved boundaries, but can nonetheless describe complex geometries by increasing the grid resolution. Of course, this entails higher resolution than curved boundary methods, hence potentially more expensive simulations, even though the global tradeoff between accuracy, efficiency, and simplicity of implementation is ultimately the user's choice. Nevertheless, with the aid of suitable extrapolation schemes, the present boundary treatment could also be used as an off-lattice BC, to track the exact wall location. Combining the present treatment with the extrapolation scheme is beyond the scope of present work.

V. NUMERICAL VERIFICATION

In this section we present the results of benchmark tests we performed in order to assess the accuracy and stability of the boundary treatment outlined above. We considered the following test cases: 2D and 3D pressure-driven flows in straight and inclined channels, flow between two rotating cylinders, and the turbulent dipole-wall collision. The D2Q9 and D3Q27 lattices are employed for 2D and 3D benchmarks, respectively. For the sake of comparison, the nonequilibrium extrapolation method [10], Grad's approximation scheme [11], and the regularized method [12] are also included in the benchmark tests. It must be emphasized that the present scheme does not involve any substantial computational overhead as compared to the above treatments [10–12]. To assess the computational accuracy of the LB solutions, the following relative error measure is defined:

$$E_r = \sqrt{\frac{1}{N}\sum_{N}\left(\frac{R_{\text{analytic}} - R_{\text{num}}}{R_{\text{analytic}}}\right)^2}, \tag{38}$$

where R_{num} and R_{analytic} indicate the numerical and analytic results, respectively, and N is the total number of nodes in the computational domain.

A. Pressure-driven channel flows (in two and three dimensions)

A fully developed pressure-driven channel flow is a classical case to validate no-slip BCs. The flow is characterized by a constant pressure drop along the channel. This pressure drop can be applied by enforcing the velocity profile from the analytic solution of the flow on the inlet and the outlet of the channel, implementing the pressure drop as a body force $F = -\nabla p/\rho$, or using pressure boundary conditions at the inlet and outlet of the channel. Each of these methods has its own benefits to challenge the present boundary treatment, which are discussed below.

Enforcing the analytic solution at the inlet and the outlet of the channel is a reasonable starting point because it provides a chance to compare the current BC with some of well-known boundary treatments that originally did not consider the velocity shift associated with the body-force implementation. In this respect, the method introduced by Zou and He [7], in addition to those in [10–12], is employed in the test case. The geometry is a 2D horizontal straight channel along the x direction, with zero velocity at the walls and an analytic velocity profile at the inlet and outlet for each BC treatment. The analytic velocity distribution of this flow is given by [34]

$$\frac{U}{U_{\max}} = 4\frac{y}{H}\left(1 - \frac{y}{H}\right), \tag{39}$$

where H is the channel height and the maximum value of velocity $U_{\max}$ is defined as $U_{\max} = -\nabla p H^2/8\mu$. In this simulation, the pressure drop is adjusted so as to keep the Reynolds number at a constant value Re = 5.

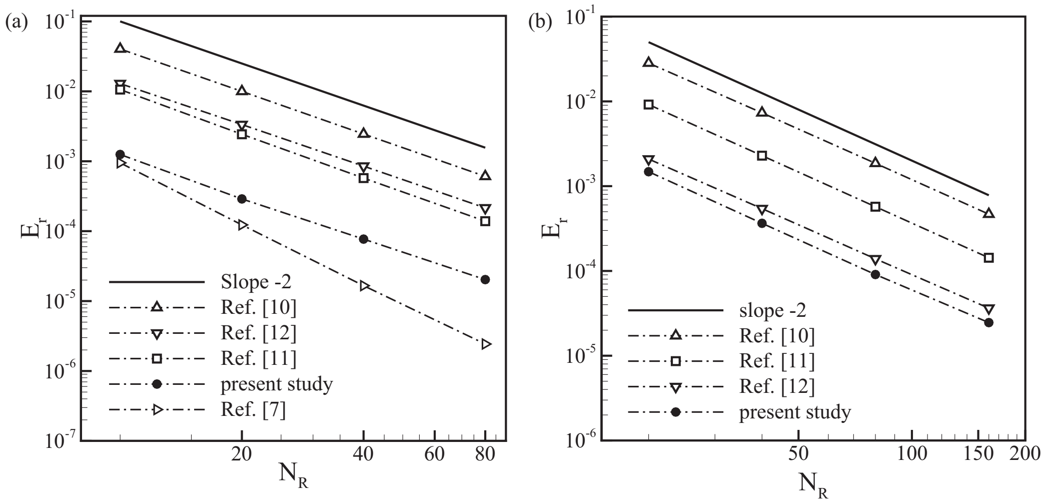

FIG. 3. Relative error in the velocity field as a function of the lattice resolution for straight pressure-driven channel flow at Re = 5 in (a) a two-dimensional geometry and (b) a three-dimensional geometry.

Figure 3(a) shows the profile of the velocity relative error E_r as a function of the lattice resolution N_R for the present boundary treatment, wherein fitting of error data indicates second-order overall accuracy for the present scheme. Also included in this figure are results from other mentioned approaches. As discussed previously, the nonequilibrium extrapolation approach [10] and Grad's approximation [11] do not cater to velocity enforcement and consequently they lead to comparatively less accurate results. The regularized method [12] and the present scheme enforce the desired velocity on the boundary node, but they drop some information during the redistribution step, at variance with the Zou-He method [7], which does not change the known distributions. As a result, the Zou-He method leads to the most accurate results for this test, the present scheme being the second most accurate, likely because of the beneficial effects of the h_i function.

To assess the compatibility of the present scheme with body-force-driven flows, a pressure-driven flow is simulated in a 3D rectangular duct. Again, the channel is horizontal and aligned with the x direction. In this case, the pressure drop is implemented as a body force, the no-slip condition is applied to the walls, and the periodic condition is employed at the inlet and outlet of the channel. The channel has a constant square cross section, extending from $y = -a$ to $y = a$ and from $z = -a$ to $z = a$. This flow exhibits the following analytic solution [35]:

$$\frac{U}{U_{\text{ref}}} = \frac{1}{0.6a^2}\left(a^2 - y^2 - \frac{32a^2}{\pi^3}\sum_{n=1,3,5,\dots}^{\infty}\left[(-1)^{(n-1)/2}\frac{\cosh(n\pi z/2a)}{\cosh(n\pi/2)}\frac{\cosh(n\pi y/2a)}{n^3}\right]\right). \tag{40}$$

In the above, the reference velocity is $U_{\text{ref}} = -0.3a^2\nabla p/\mu$. In this simulation, the flow parameters are adjusted to keep the Reynolds number (based on the reference velocity and hydraulic diameter) constant at Re = 5. The Zou-He method was originally defined for straight walls and flat surfaces on D2Q9 and D3Q15 lattices and consequently it is not included in this test. Since the velocity definition (7) is not directly involved in the nonequilibrium extrapolation method, the formulation of this scheme is not affected by the body-force implementation and is used without any modification. For the regularized method and Grad's approximation, the nonequilibrium approximation (14) is adjusted for the velocity shift in a way similar to the current approach, so as to guarantee an equal treatment. Figure 3(b) presents the velocity profile relative error variations for different lattice resolutions. According to this figure, the present boundary treatment is of second-order accuracy and leads to more accurate results compared to the other tested schemes.

As the last method, we conduct a pressure-driven flow simulation by applying the pressure difference directly to both ends of the channel. As shown in Fig. 4, an inclined 2D channel is modeled in the D2Q9 lattice, where the inclination angle is set to $a/b = 1/3$. Note that upper and lower walls and inlet and outlet ports are modeled as staircase-shaped lines, due to the nature of the Cartesian LB grid. It must be emphasized that no curved boundary treatment is employed in this simulation and the desired velocity or pressure is applied directly to boundary nodes. The velocity vectors are presented at some typical locations along the channel in Fig. 4. From the mentioned boundary treatments [7,10–12], only the nonequilibrium extrapolation approach and the present scheme allow us to implement a pressure boundary condition for an arbitrary shape geometry. Figure 5 shows the velocity profile relative error for the inclined channel flow as a function of the grid resolution. It can be observed that the present BC provides a considerably more accurate prediction of the velocity profile as compared to the nonequilibrium extrapolation scheme.

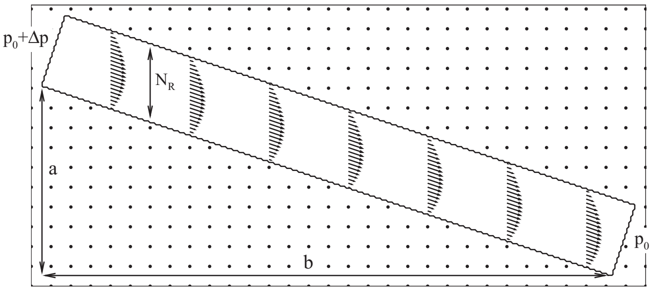

FIG. 4. Geometry and velocity vectors for an inclined flat channel where the grid resolution is $N_R = 40$ and the inclination angle is $a/b = 1/3$.

B. Flow between two rotating circular cylinders

To demonstrate the capability of the present boundary treatment for more complex geometries, the flow between two rotating cylinders is simulated for Reynolds number $\text{Re} = R_i\omega(R_o - R_i)/\nu = 10$. In this simulation, the inner

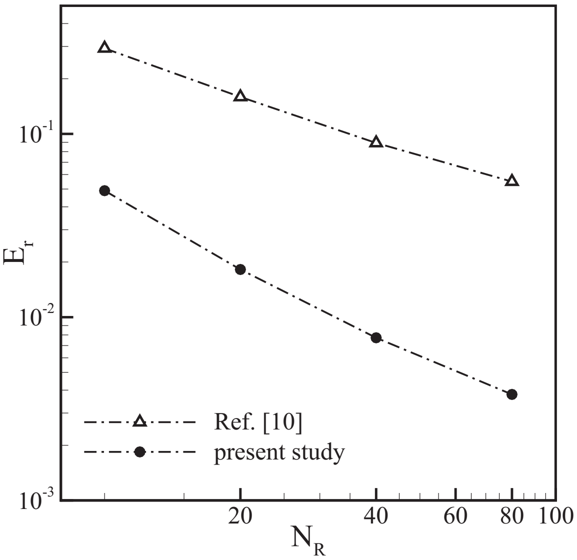

FIG. 5. Relative error in the velocity field as a function of lattice resolution for the inclined pressure-driven channel flow at Re = 5, where the inclination angle is $a/b = 1/3$.

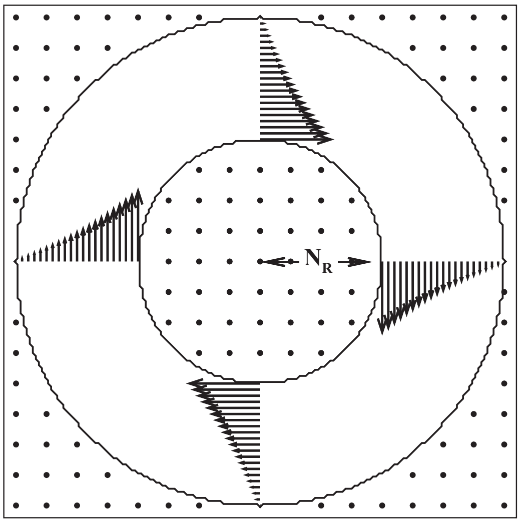

FIG. 6. Geometry and velocity vectors for flow between two rotating circular cylinders where the grid resolution is $N_R = 40$ and the radius ratio is $\beta = 1/2$.

cylinder with radius $R_i = 1$ cm is rotating at a constant angular velocity ω and the outer cylinder with radius $R_o = 2$ cm is kept stationary. This problem is a useful benchmark test to examine the accuracy and efficiency of the proposed boundary treatment in the simulation of moving and stationary boundaries in complex geometries. Note that, similar to the inclined channel case, no extrapolation associated with the curved boundary treatment is employed. The Navier-Stokes equations exhibit the following analytical solution for the transversal velocity [34]:

$$\frac{U_\theta(r)}{U_0} = \frac{\beta}{1-\beta^2}\left(\frac{R_o}{r} - \frac{r}{R_o}\right), \tag{41}$$

where $U_0 = R_i\omega$ is the reference velocity and $\beta = R_i/R_o$. The computational domain is modeled using a square mesh with the radius of the inner cylinder consisting of N_R nodes along lattice rows. A schematic view of this geometry is reported in Fig. 6 for $N_R = 40$. For validation purposes, the velocity profile is compared with the analytic solutions in Fig. 7(a), for different grid resolutions. Also included in this figure are numerical results obtained from the nonequilibrium extrapolation scheme [10] and Grad's approximation [11], and the regularized method [12]. It can be observed that the present boundary condition provides a more accurate velocity prediction as compared to the other schemes. Change in the velocity direction as shown in Fig. 6 leads to a centripetal force normal to boundary surfaces. This situation provides a good challenge to assess the performance of our density approximation [Eq. (29)] in comparison to the density extrapolation, which is the most common approach in dealing with complex geometries. In this regard, a measure of mass leakage is defined as the rate of change in the summation of the mass density over lattice sites in the computational domain after N iterations:

$$\dot{M}_{\text{loss}} = \frac{\sum \rho(x,y)_{t+N\delta_t} - \sum \rho(x,y)_t}{N\delta_t}. \tag{42}$$

Figure 7(b) shows the variation of mass leakage when the Reynolds number increases from 10 to 640 and the grid resolution is kept constant at $N_R = 20$. Since the boundary is not straight in this geometry, Grad's approximation [11], the nonequilibrium extrapolation [10], and the regularized method [12] use extrapolation to obtain the density value from the nearest fluid node. Among these approaches, Grad's approximation [11] shows the best performance in terms of mass leakage, as it keeps the known distributions unchanged. In contrast, the regularized method [12], which approximates both known and unknown distributions, shows considerable mass leakage, especially at high Reynolds numbers. The performances of these two schemes in the simulation of velocity fields are completely different. As shown in Fig. 7(a), the regularized method [12] has better accuracy in the velocity field prediction compared to Grad's approximation [11] because it can enforce the velocity values at the boundary nodes. In other words, Grad's approximation [11] and similarly the nonequilibrium extrapolation method [10] deliver satisfactory

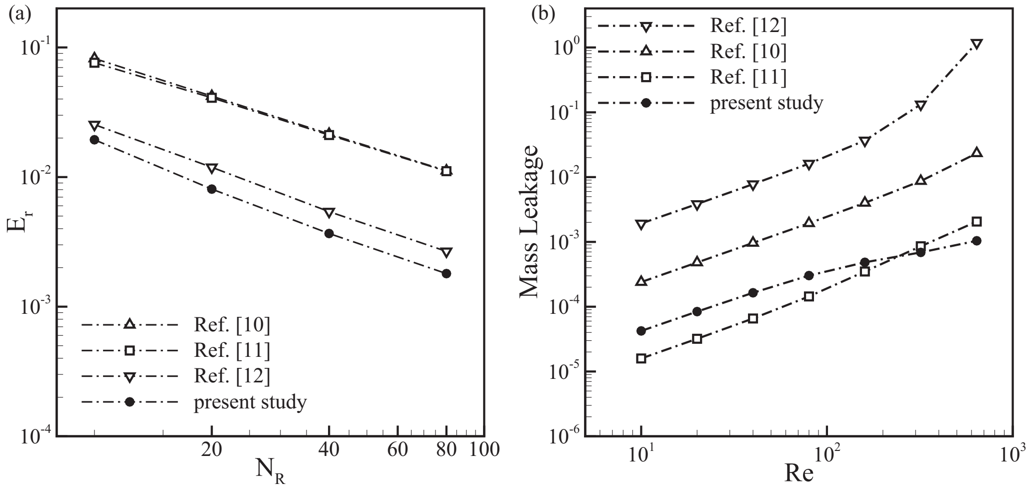

FIG. 7. (a) Comparison of velocity relative errors for different boundary schemes as a function of lattice resolution at Re = 10 for flow between two rotating cylinders. (b) Mass leakage variation as a function of the Reynolds number for a grid resolution of $N_R = 20$ for flow between two rotating cylinders.

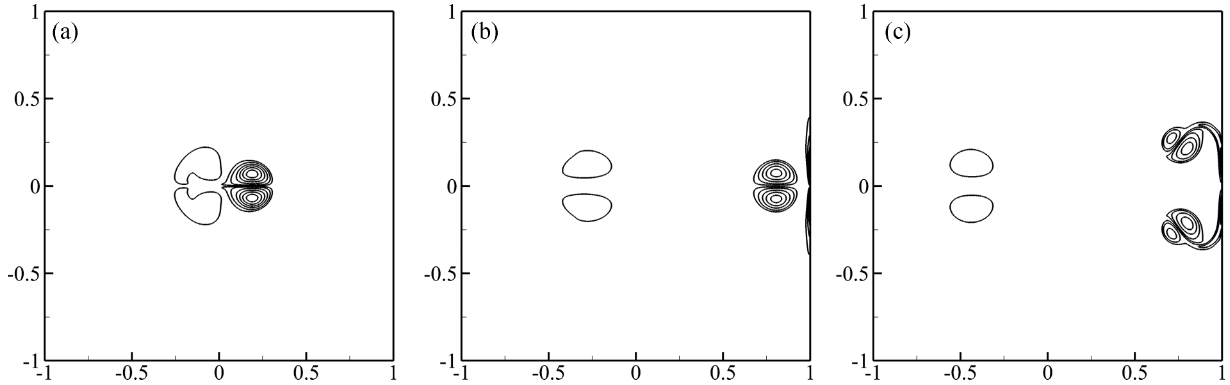

FIG. 8. Sequence of vorticity contour plots showing the flow evolution of a dipole collision for Re = 625 at (a) $t = 0.1$, (b) $t = 0.3$, and (c) $t = 0.5$.

performances in mass leakage control by losing accuracy in the velocity field prediction. However, the present scheme, with the aid of Eq. (29), features an acceptable mass leakage, comparable to Grad's approximation [11] [Fig. 7(b)]. At the same time, it also leads to the most accurate results in velocity field prediction [Fig. 7(a)].

C. Turbulent dipole-wall collision

The complexity of a given flow is not necessarily due to a complex geometry, the collision of a dipole with a flat wall being a good example. A dipole is a system of two counterrotating vortices located next to each other. Due to rotation of these vortices, a forward momentum is induced, leading to a self-propelled motion of the dipole. The benchmark problem considered in this section is the normal collision of a 2D dipole with a no-slip flat wall. The fluid is incompressible and the flow is confined in a square domain $[-1,1] \times [-1,1]$ with the no-slip condition at boundaries. The initial condition is chosen to reproduce two counterrotating monopoles, one with positive core vorticity at the position $(x_1,y_1) = (0,0.1)$ and one with negative core vorticity at $(x_2,y_2) = (0,-0.1)$. This is obtained with the following initial velocity field:

$$\begin{aligned} u_x &= -\tfrac{1}{2}|\omega_e|(y-y_1)\exp[-(r_1/r_0)^2] \\ &\quad + \tfrac{1}{2}|\omega_e|(y-y_2)\exp[-(r_2/r_0)^2], \\ u_y &= -\tfrac{1}{2}|\omega_e|(x-x_1)\exp[-(r_1/r_0)^2] \\ &\quad + \tfrac{1}{2}|\omega_e|(x-x_2)\exp[-(r_2/r_0)^2], \end{aligned} \tag{43}$$

where $r_i = \sqrt{(x-x_i)^2 + (y-y_i)^2}$ is the distance from (x_i,y_i), r_0 is the diameter of a monopole, and ω_e is its initial core vorticity. The value of $\omega_e = 299.5286$ [36] is used with the condition that the total kinetic energy of the system

$$E(t) = \frac{1}{2}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{u}^2(x,t)dx\,dy \tag{44}$$

is initially $E(0) = 2$ for all runs. The diameter of a monopole is set at $r_0 = 0.1$ and the Reynolds number $\mathrm{Re} = UW/\nu$ is $\mathrm{Re} = 625$, where U is a characteristic velocity of the flow and W is the half-width of the domain that is modeled by N_R nodes. The total enstrophy of the dipolar flow field that is monitored in this benchmark is defined as

$$\Omega(t) = \frac{1}{2}\int_{-1}^{+1}\int_{-1}^{+1} \omega^2(x,t)dx\,dy, \tag{45}$$

where $\omega = \partial_x u_y - \partial_y u_x$ is the flow vorticity. The dipole flow is an initial-value problem. The initial value for the velocity field is described by Eq. (43) and can be implemented to the LB approach by the present scheme using Eq. (37).

In this implementation, a second-order finite-difference scheme is used to determine the velocity gradients and consequently the $\mathbf{C}$ tensor is calculated using Eq. (25). The local density, however, is unknown. One way to obtain the local density values is to determine the pressure distribution by solving the following Poisson equation:

$$\nabla^2 p = -\boldsymbol{\nabla} \cdot [\mathbf{u} \cdot (\boldsymbol{\nabla}\mathbf{u})]. \tag{46}$$

After determining the pressure distribution, local densities can be calculated using Eq. (8). There are many numerical schemes available to solve the Poisson equation and here we use the one proposed by Mohammadipour and Niazmand [37] to solve the Poisson equation within the LB approach.

The evolution of the dipolar vortex during its collision with the no-slip wall is shown in Fig. 8 as a contour plot of the vorticity at three given time steps. After releasing the initial monopolar vortices, a secondary vortex flow is constructed around the main dipole. At a later stage, the main dipole travels away along the positive-x direction and the secondary vortex flow left behind combines into another dipole that translates in the opposite direction [Fig. 8(a)]. The secondary dipole is weaker and travels with lower velocity compared to the main dipole Fig. 8(b). As the dipole approaches the wall, small-scale structures are generated in the boundary layer close to the wall, leading to an increase of the vorticity amplitude. After the collision, the dipole halves separate from each other along circular trajectories, preparing for a second collision with the wall, as shown in Fig. 8(c). The time evolution of the enstrophy of the system is shown in Fig. 9(a), where the collision is signaled by a peak in the enstrophy curve. The value of this local maximum will be used as a benchmark to compare the present boundary scheme with results obtained from well-known boundary approaches [7,10–12]. The error

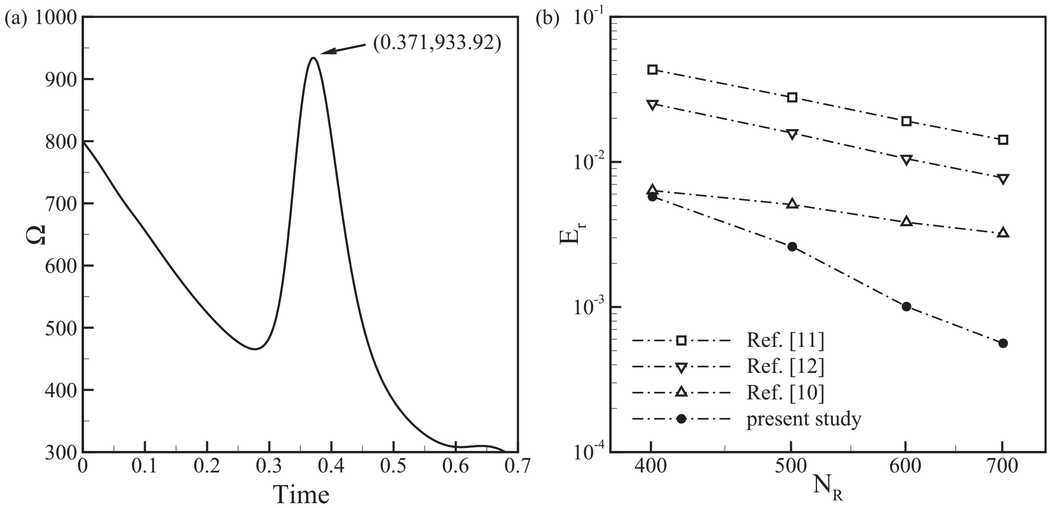

FIG. 9. (a) Enstrophy $\Omega(t)$ for the dipole-wall collision with Re = 625 and grid resolution $N_R = 700$. (b) Numerical accuracy in capturing the maximum enstrophy of the dipole-wall collision as a function of lattice resolution for Re = 625.

in the value of the enstrophy peak is quantified as follows:

$$E_r = \frac{\left|\Omega_{\max}^{\text{ref}} - \Omega_{\max}\right|}{\Omega_{\max}^{\text{ref}}}, \tag{47}$$

where $\Omega_{\max}$ is the value of the enstrophy peak obtained from the lattice Boltzmann simulation and $\Omega_{\max}^{\text{ref}} = 933.4$ is the average value of the enstrophy maximum in previous studies [36,38]. Since the flow in this case is a simple 2D box with four straight walls, it can be easily implemented by using the boundary schemes [7,10–12]. Figure 9(b) illustrates the relative error as a function of grid resolution for the present scheme and the other approaches. The implementation of the Zou-He method [7] shows numerical instabilities and fails to converge with the given grid resolutions. The nonequilibrium extrapolation approach [10] shows a better performance than the regularized method [12] in capturing the collision phenomena. It can be further observed that the present treatment is significantly more accurate than other approaches.

VI. CONCLUSION

An on-lattice boundary condition scheme was proposed to handle pressure and velocity boundary conditions, based on a third-order expansion of the distribution function in the discrete lattice velocities. The basic idea is to decompose the distribution function at the boundary node into equilibrium and nonequilibrium parts and then to approximate each of them with the help of available information from the boundary node and its neighboring node. Schemes for pressure and velocity boundaries were given based on this method, which can be used for arbitrary shape geometries in combination with D2Q9, D3Q15, D3Q19, and D3Q27 lattices. It replaces all distribution functions on the boundary node to enforce the exact desired boundary values while it makes use of higher-order terms of the distribution to minimize the information loss due to the redistribution step.

The scheme was tested against several well-established problems including pressure-driven flow in 2D and 3D straight channels, pressure-driven flow in an inclined channel, laminar flow between two rotating cylinders, and the turbulent dipole-wall collision. The numerical results indicate a competitive performance with existing consolidated boundary treatments, in terms of accuracy, mass leakage, and computational performance. It is therefore hoped that the proposed method offers a further valuable option for the implementation of boundary conditions for in lattice Boltzmann flows and multiscale applications thereof [39].

ACKNOWLEDGMENT

The financial support from Payame Noor University under the Grant No. 1/12/735 is greatly appreciated.

APPENDIX A: DERIVATION OF THE NONEQUILIBRIUM PART MOMENTS

In this appendix we derive the macroscopic approach for the nonequilibrium part of the distribution function. The derivation begins with expansion of the distribution functions around the equilibrium value

$$\begin{aligned} f_i(\mathbf{r},t) &= f_i^{(0)}(\mathbf{r},t) + \varepsilon f_i^{(1)}(\mathbf{r},t) + \cdots \\ &= f_i^{\text{eq}}(\mathbf{r},t) + f_i^{\text{neq}}(\mathbf{r},t) + O(\varepsilon^2), \end{aligned} \tag{A1}$$

where ε is the Knudsen number. Moments of Eq. (A1) in the particle velocity field are calculated as follows:

$$\sum_i f_i = \sum_i f_i^{\text{eq}} + \sum_i f_i^{\text{neq}}, \tag{A2}$$

$$\sum_i f_i e_{i\alpha} = \sum_i f_i^{\text{eq}} e_{i\alpha} + \sum_i f_i^{\text{neq}} e_{i\alpha}. \tag{A3}$$

With the definitions of macroscopic quantities (7) and (8) and the equilibrium part (4), one can easily obtain the following moments:

$$\sum_i f_i^{\text{neq}} = 0, \tag{A4}$$

$$\sum_i f_i^{\text{neq}} e_{i\alpha} = \frac{-\delta_t F_\alpha}{2}. \tag{A5}$$

The calculation of $\sum_i f_i^{\text{neq}} e_{i\alpha} e_{i\beta}$ is somewhat tricky and needs some extra calculations. According to Eq. (2), the lattice Boltzmann evolution equation for the distribution function can be described as follows:

$$f_i(\mathbf{r} + \mathbf{e}_i \delta_t, t + \delta_t) = f_i(\mathbf{r},t) + \frac{1}{\tau}\big[f_i^{\text{eq}}(\mathbf{r},t) - f_i(\mathbf{r},t)\big] + \delta_t G_i. \tag{A6}$$

The left-hand side of Eq. (A6) can be expanded around $(\mathbf{r},t)$ using Taylor series expansion:

$$f_i(\mathbf{r}+\mathbf{e}_i\delta_t, t+\delta_t) = f_i(\mathbf{r},t) + \sum_{n=1}^{\infty} \frac{D^n}{n!} \varepsilon^n \big(f_i^{(0)} + \varepsilon f_i^{(1)} + \cdots\big). \tag{A7}$$

Based on the formal multiscale approximation, known as Chapman-Enskog expansion, the material derivative is expressed as follows:

$$D = \partial_t + e_{i\alpha}\partial_\alpha = \sum_{n=1}^{\infty} \varepsilon^n \partial_{t(n)}. \tag{A8}$$

Substituting Eqs. (A2) and (A3) into Eq. (A1) and collecting coefficients of ε^1 leads to

$$\partial_{t_0} f_i^{(0)} + e_{i\alpha}\partial_\alpha f_i^{(0)} = -\frac{f_i^{(1)}}{\tau} + G_i. \tag{A9}$$

According to Eq. (A9), the nonequilibrium part can be expressed as

$$f_i^{\text{neq}} = \varepsilon f_i^{(1)} = \tau\delta_t\big(G_i - \partial_{t_0} f_i^{(0)} - e_{i\alpha}\partial_\alpha f_i^{(0)}\big). \tag{A10}$$

Note that, to capture the physics, the time step δ_t should be on the order of the Knudsen number ($\delta_t = \varepsilon$). Consequently, $\sum_i f_i^{\text{neq}} e_{i\alpha} e_{i\beta}$ can be calculated as

$$\sum_i f_i^{\text{neq}} e_{i\alpha} e_{i\beta} = \underbrace{\sum_i \tau\delta_t G_i e_{i\alpha} e_{i\beta}}_{\text{term 1}} - \underbrace{\sum_i \tau\delta_t \partial_{t_0} f_i^{(0)} e_{i\alpha} e_{i\beta}}_{\text{term 2}} - \underbrace{\sum_i \tau\delta_t e_{i\alpha} e_{i\beta} e_{i\gamma}\partial_\gamma f_i^{(0)}}_{\text{term 3}}. \tag{A11}$$

There are three terms in Eq. (A11) that are determined below one by one:

$$\begin{aligned}
\underbrace{\sum_i \tau\delta_t G_i e_{i\alpha} e_{i\beta}}_{\text{term 1}} &= \sum_i \tau\delta_t w_i \left(1 - \frac{1}{2\tau}\right)\left[\frac{e_{i\gamma}F_\gamma - u_\delta F_\delta}{c_s^2} + \frac{e_{i\gamma}F_\gamma e_{i\delta}u_\delta}{c_s^4}\right] e_{i\alpha} e_{i\beta},\\
\text{term 1} &= \delta_t\left(\tau - \frac{1}{2}\right)\left[\sum_i w_i \frac{-u_\delta F_\delta e_{i\alpha} e_{i\beta}}{c_s^2} + \sum_i w_i \frac{e_{i\alpha} e_{i\beta} e_{i\gamma} e_{i\delta} F_\gamma u_\delta}{c_s^4}\right],\\
\text{term 1} &= \delta_t\big(\tau - \tfrac{1}{2}\big)[u_\alpha F_\beta + u_\beta F_\alpha],
\end{aligned} \tag{A12}$$

$$\begin{aligned}
\underbrace{\sum_i \tau\delta_t \partial_{t_0} f_i^{(0)} e_{i\alpha} e_{i\beta}}_{\text{term 2}} &= \tau\delta_t \partial_{t_0}\big(\rho c_s^2 \delta_{\alpha\beta} + \rho u_\alpha u_\beta\big),\\
\text{term 2} &= \tau\delta_t\big[c_s^2\delta_{\alpha\beta}(-\partial_\gamma \rho u_\gamma) + u_\alpha \partial_{t_0}\rho u_\beta + u_\beta \partial_{t_0}\rho u_\alpha - u_\alpha u_\beta \partial_{t_0}\rho\big],\\
\text{term 2} &= \tau\delta_t\big\{-c_s^2\delta_{\alpha\beta}\partial_\gamma \rho u_\gamma + u_\alpha\big[F_\beta - \partial_\beta \rho c_s^2 - \partial_\gamma(\rho u_\gamma u_\beta)\big] + u_\beta\big[F_\alpha - \partial_\alpha \rho c_s^2 - \partial_\gamma(\rho u_\gamma u_\alpha)\big]\\
&\quad - u_\alpha u_\beta[-\partial(\partial_\gamma \rho u_\gamma)]\big\},\\
\text{term 2} &= \tau\delta_t\big[-c_s^2\delta_{\alpha\beta}\partial_\gamma \rho u_\gamma + u_\alpha F_\beta - u_\alpha \partial_\beta \rho c_s^2 + u_\beta F_\alpha - u_\beta \partial_\alpha \rho c_s^2 + O(u^3)\big],
\end{aligned} \tag{A13}$$

$$\begin{aligned}
\underbrace{\sum_i \tau\delta_t e_{i\alpha} e_{i\beta} e_{i\gamma}\partial_\gamma f_i^{(0)}}_{\text{term 3}} &= \tau\delta_t c_s^2[\partial_\beta \rho u_\alpha + \partial_\alpha \rho u_\beta + \delta_{\alpha\beta}\partial_\gamma \rho u_\gamma],\\
\text{term 3} &= \tau\delta_t c_s^2[\rho\partial_\beta u_\alpha + u_\alpha\partial_\beta\rho + \rho\partial_\alpha u_\beta + u_\beta\partial_\alpha\rho + \delta_{\alpha\beta}\partial_\gamma \rho u_\gamma].
\end{aligned} \tag{A14}$$

Substituting Eqs. (A12)–(A14) into Eq. (A11) leads to

$$\sum_i f_i^{\text{neq}} e_{i\alpha} e_{i\beta} = -\frac{\delta_t}{2}[u_\alpha F_\beta + u_\beta F_\alpha] - \tau\delta_t \rho c_s^2[\partial_\beta u_\alpha + \partial_\alpha u_\beta]. \tag{A15}$$

APPENDIX B: DERIVATION OF THE LOCAL MASS DENSITY APPROXIMATION

Derivation of the mass density approximation begins with the definition of density in the LB method:

$$\rho = \sum_{i\in I_K} f_i + \sum_{i\in I_{UK}} f_i + \sum_{i\in I_{UU}} f_i = \sum_{i\in I_K} f_i + \sum_{i\in I_{UK}} \big(f_i^{\text{eq}} + f_i^{\text{neq}}\big) + \sum_{i\in I_{UU}} f_i. \tag{B1}$$

We temporarily define a new function as follows:

$$g_i = \frac{f_i^{\mathrm{eq}}}{\rho} = w_i\left[1 + \frac{\mathbf{e}_i\cdot\mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i\cdot\mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2}\right]. \tag{B2}$$

Substituting Eq. (B2) into Eq. (B1) and approximating the I_{UU} distribution functions by the corresponding distribution in the *next* node leads to

$$\rho = \sum_{i\in I_K} f_i + \rho\sum_{i\in I_{UK}} g_i + \sum_{i\in I_{UK}} f_i^{\mathrm{neq}} + \sum_{i\in I_{UU}} f_i^{(\mathrm{next})}. \tag{B3}$$

According to Eq. (26), the nonequilibrium parts of distributions in the I_{UK} directions are related to the distribution function in opposite directions as follows:

$$f_i^{\mathrm{neq}} \approx f_{\mathrm{opp}(i)}^{\mathrm{neq}} - \frac{w_i\delta_t}{c_s^2}e_{i\alpha}F_\alpha + 2h_i. \tag{B4}$$

Substituting Eq. (B4) into Eq. (B3) leads to

$$\rho = \sum_{i\in I_K} f_i + \rho\sum_{i\in I_{UK}} g_i + \sum_{i\in I_{UK}}\left[f_{\mathrm{opp}(i)}^{\mathrm{neq}} - w_i e_{i\alpha}\frac{\delta_t F_\alpha}{c_s^2} + 2h_i\right] + \sum_{i\in I_{UU}} f_i^{(\mathrm{next})},$$

$$\rho = \sum_{i\in I_K} f_i + \rho\sum_{i\in I_{UK}} g_i + \sum_{i\in I_{UK}}\left[\left(f_{\mathrm{opp}(i)} - f_{\mathrm{opp}(i)}^{\mathrm{eq}}\right) - w_i e_{i\alpha}\frac{\delta_t F_\alpha}{c_s^2} + 2h_i\right] + \sum_{i\in I_{UU}} f_i^{(\mathrm{next})}. \tag{B5}$$

By ignoring the h_i, the above equation can be simplified as

$$\rho = \sum_{i\in I_K} f_i + \rho\sum_{i\in I_{UK}} g_i + \sum_{i\in I_{UK}} f_{\mathrm{opp}(i)} - \rho\sum_{i\in I_{UK}} g_{\mathrm{opp}(i)} - \sum_{i\in I_{UK}} w_i e_{i\alpha}\frac{\delta_t F_\alpha}{c_s^2} + \sum_{i\in I_{UU}} f_i^{(\mathrm{next})},$$

$$\rho\left(1 - \sum_{i\in I_{UK}}(g_i - g_{\mathrm{opp}(i)})\right) = \sum_{i\in I_K} f_i + \sum_{i\in I_{UK}} f_{\mathrm{opp}(i)} + \sum_{i\in I_{UU}} f_i^{(\mathrm{next})} - \frac{\delta_t}{c_s^2}\sum_{i\in I_{UK}} w_i e_{i\alpha}F_\alpha. \tag{B6}$$

According to Eq. (B2), the difference between g_i values in two opposite directions is related to velocity as follows:

$$g_i - g_{\mathrm{opp}(i)} = \frac{2w_i u_\alpha e_{i\alpha}}{c_s^2}. \tag{B7}$$

Substituting Eq. (B7) into Eq. (B6) leads to the final approach for mass density:

$$\rho = \frac{\sum_{i\in I_K} f_i + \sum_{i\in I_{UK}} f_{\mathrm{opp}(i)} + \sum_{i\in I_{UU}} f_i^{(\mathrm{next})} - \frac{\delta_t}{c_s^2}\sum_{i\in I_{UK}} w_i e_{i\alpha}F_\alpha}{1 - \frac{2}{c_s^2}\sum_{i\in I_{UK}} w_i u_\alpha e_{i\alpha}}. \tag{B8}$$

APPENDIX C: DETERMINATION OF THE *next* NODE'S LOCATION

In the present study, a superscript $(next)$ is introduced that has a key role in our BC scheme. Here we discuss a simple strategy to specify this parameter in an arbitrary shape boundary in 2D and 3D geometries. The *next* node is the nearest fluid point to the boundary node. Obviously, there is a specific velocity direction in each boundary node that points to this *next* node. We denote this specific velocity direction by $\mathbf{e}_n$ and call it the normal velocity direction. To distinguish $\mathbf{e}_n$ among lattice velocities, we define three categories for lattice nodes. There are some nodes located in the fluid region, which we call fluid nodes. A series of lattice nodes named boundary nodes construct the fluid-solid interface and are subjected to the boundary treatments. There are also some lattice nodes that are located inside the solid region. These nodes that are completely isolated from the computational domain by boundary nodes are solid nodes.

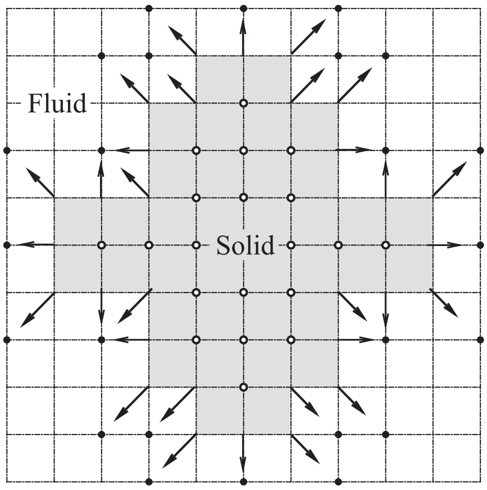

FIG. 10. Specifications of an arbitrary 2D geometry in the D2Q9 lattice.

TABLE I. Lattice parameters and velocity set priority to examine the normal direction rule.

Lattice	Velocity index i	Velocity set e_i	Weighting factor	Priority to check the normal rule
D2Q9	0	(0,0)	4/9	
D2Q9	1,2,3,4	$c(\pm1,0),c(0,\pm1)$	1/9	1
D2Q9	5,6,7,8	$c(\pm1,\pm1)$	1/36	2
D3Q15	0	(0,0,0)	2/9	
D3Q15	$1,2,\ldots,6$	$c(0,0,\pm1),c(0,\pm1,0),c(\pm1,0,0)$	1/9	1
D3Q15	$7,8,\ldots,14$	$c(\pm1,\pm1,\pm1)$	1/72	2
D3Q19	0	(0,0,0)	1/3	
D3Q19	$1,2,\ldots,6$	$c(0,0,\pm1),c(0,\pm1,0),c(\pm1,0,0)$	1/18	1
D3Q19	$7,8,\ldots,18$	$c(0,\pm1,\pm1),c(\pm1,\pm1,0),c(\pm1,\pm1,0)$	1/36	2
D3Q27	0	(0,0,0)	8/27	
D3Q27	$1,2,\ldots,6$	$c(0,0,\pm1),c(0,\pm1,0),c(\pm1,0,0)$	2/27	1
D3Q27	$7,8,\ldots,18$	$c(0,\pm1,\pm1),c(\pm1,\pm1,0),c(\pm1,\pm1,0)$	1/54	2
D3Q27	$19,20,\ldots,26$	$c(\pm1,\pm1,\pm1)$	1/216	3

Figure 10 illustrates boundary configurations for an arbitrary 2D shape in the D2Q9 lattice where solid nodes are depicted by white circles. The solid nodes can be employed as a guide to specify the normal velocity direction $\mathbf{e}_n$ by checking the following rule for all lattice velocities on each boundary node: The velocity direction that connects the boundary node to a fluid node while its opposite direction points to a solid node is the normal velocity direction. This rule is compatible with 2D and 3D lattices and must be examined on velocity sets of the lattice from the lower velocity values to the higher ones. After determining the normal velocity direction $\mathbf{e}_n$, the next node location is $\mathbf{r}_{next} = \mathbf{r}_b + \delta_t \mathbf{e}_n$. In Fig. 10 the normal velocity vector is depicted as a single vector for each boundary node and the corresponding *next* node is illustrated as a black circle. The lattice parameters and the velocity set priority to examine the normal direction rule are listed in Table I.

[1] F. J. Higuera and J. Jiménez, Europhys. Lett. **9**, 663 (1989).

[2] G. R. McNamara and G. Zanetti, Phys. Rev. Lett. **61**, 2332 (1988).

[3] R. Benzi, S. Succi, and M. Vergassola, Phys. Rep. **222**, 145 (1992).

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

[5] D. P. Ziegler, J. Stat. Phys. **71**, 1171 (1993).

[6] X. He, Q. Zou, L.-S. Luo, and M. Dembo, J. Stat. Phys. **87**, 115 (1997).

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

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

[9] S. Chen, D. Martínez, and R. Mei, Phys. Fluids **8**, 2527 (1996).

[10] Z.-L. Guo, C.-G. Zheng, and B.-C. Shi, Chin. Phys. **11**, 366 (2002).

[11] S. Ansumali, S. S. Chikatamarla, and I. V. Karlin, Europhys. Lett. **74**, 215 (2006).

[12] J. Latt, B. Chopard, O. Malaspinas, M. Deville, and A. Michler, Phys. Rev. E **77**, 056703 (2008).

[13] O. Filippova and D. Hänel, Int. J. Mod. Phys. C **9**, 1271 (1998).

[14] R. Mei, L.-S. Luo, and W. Shyy, J. Comput. Phys. **155**, 307 (1999).

[15] M. Bouzidi, M. Firdaouss, and P. Lallemand, Phys. Fluids **13**, 3452 (2001).

[16] D. Yu, R. Mei, L. L.-S. Luo, and W. Shyy, Prog. Aerosp. Sci. **39**, 329 (2003).

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

[18] O. R. Mohammadipoor, H. Niazmand, and S. A. Mirbozorgi, Phys. Rev. E **89**, 013309 (2014).

[19] B. Dorschner, S. S. Chikatamarla, F. Bösch, and I. V. Karlin, J. Comput. Phys. **295**, 340 (2015).

[20] C. K. Aidun, J. R. Clausen, and G. W. Woodruff, Annu. Rev. Fluid Mech. **42**, 439 (2010).

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

[22] D. d'Humires, P. Lallemand, and U. Frisch, Europhys. Lett. **2**, 291 (1986).

[23] F. J. Higuera, S. Succi, and R. Benzi, Europhys. Lett. **9**, 345 (1989).

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

[25] P. L. Bhatnagar, E. P. Gross, and M. Krook, Phys. Rev. **94**, 511 (1954).

[26] Z. G. Feng and E. E. Michaelides, J. Comput. Phys. **195**, 602 (2004).

[27] J. M. Buick and C. A. Greated, Phys. Rev. E **61**, 5307 (2000).

[28] N. S. Martys, X. Shan, and H. Chen, Phys. Rev. E **58**, 6855 (1998).

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

[30] S. Succi, I. V. Karlin, and H. Chen, Rev. Mod. Phys. **74**, 1203 (2002).

[31] F. Higuera and S. Succi, Europhys. Lett. **8**, 517 (1989).

[32] A. A. Mohamad and A. Kuzmin, Int. J. Heat Mass Transfer **53**, 990 (2010).

[33] S. Succi, Europhys. Lett. **109**, 50001 (2015).

[34] I. G. Currie, *Fundamental Mechanics of Fluids*, 3rd ed. (Dekker, New York, 2002), p. 154.

[35] B. Wu, A. Van Hirtum, and X. Y. Luo, Int. J. Appl. Mech. **05**, 1350002 (2013).

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

[37] O. R. Mohammadipour and H. Niazmand, Int. J. Mod. Phys. C **26**, 1550078 (2015).

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

[39] O. Filippova, S. Succi, F. Mazzocco, C. Arrighetti, G. Bella, and D. Hänel, J. Comput. Phys. **170**, 812 (2001).