2-12-2015

A Unified and Preserved Dirichlet Boundary Treatment for the Cell-Centered Finite Volume Discrete Boltzmann Method

Leitao Chen
University of Pittsburgh, chenl12@erau.edu

Laura A. Schaefer
University of Pittsburgh

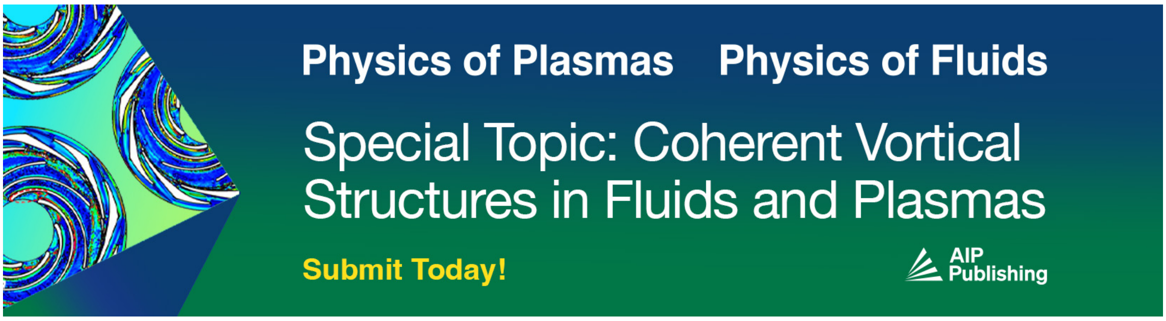
Scholarly Commons Citation
Chen, L., & Schaefer, L. A. (2015). A Unified and Preserved Dirichlet Boundary Treatment for the Cell-Centered Finite Volume Discrete Boltzmann Method. *Physics of Fluids, 27*(2). https://doi.org/10.1063/1.4907782

RESEARCH ARTICLE | FEBRUARY 12 2015

A unified and preserved Dirichlet boundary treatment for the cell-centered finite volume discrete Boltzmann method

Leitao Chen (陈磊涛) ; Laura A. Schaefer

Physics of Fluids 27, 027104 (2015)
https://doi.org/10.1063/1.4907782



PHYSICS OF FLUIDS **27**, 027104 (2015)

A unified and preserved Dirichlet boundary treatment for the cell-centered finite volume discrete Boltzmann method

Leitao Chen (陈磊涛)[a] and Laura A. Schaefer[b]
Mechanical Engineering and Material Science, University of Pittsburgh, Pittsburgh, Pennsylvania 15261, USA

(Received 12 August 2014; accepted 22 January 2015; published online 12 February 2015)

A new boundary treatment is proposed for the finite volume discrete Boltzmann method (FVDBM) that can be used for accurate simulations of curved boundaries and complicated flow conditions. First, a brief review of different boundary treatments for the general Boltzmann simulations is made in order to primarily explain what type of boundary treatment will be developed in this paper for the cell-centered FVDBM. After that, the new boundary treatment along with the cell-centered FVDBM model is developed in detail. Next, the proposed boundary treatment is applied to a series of numerical tests with a detailed discussion of its qualitative and quantitative properties. From the results, it can be concluded that the new boundary treatment is at least first-order accurate for a variety of Dirichlet boundary conditions (BCs). It can handle both the velocity and density Dirichlet BCs in a unified way and further realize some BCs that the conventional lattice Boltzmann model fails to simulate. In addition, such a boundary treatment can incorporate different lattice models without changing its framework, and it can preserve the Dirichlet BCs up to machine accuracy in different situations. © *2015 AIP Publishing LLC.* [http://dx.doi.org/10.1063/1.4907782]

I. INTRODUCTION

The commonly used lattice Boltzmann method (LBM) solves a particle distribution function (PDF) from a Lagrangian point of view and couples the discretization of the particle velocity space (lattice model) and configuration space. Such coupling means that the resulting mesh configuration has to be uniform and square-shaped to form a lattice connection network for streaming for most two dimensional (2D) square lattice models such as D2Q9. Such a Cartesian-like mesh creates challenges when the flow problems involve curved boundaries, which is one of the major reasons that LBM is still not fully competitive with other modern computational fluid dynamics (CFD) tools.[1] For simple curved boundaries, the problem can be solved by applying a curvilinear mesh,[2,3] but such coordinate transformation is impossible for complex boundaries such as that in porous media, which will always have some grid points not located on the boundaries.

Many boundary schemes have been developed to treat curved boundaries, among which one category, an interpolative pseudo-boundary-fitting approach,[4–12] is the most popular. In these schemes, a layer of solid nodes near the boundary but outside of the fluid domain is created. The PDFs that will flow into the fluid nodes from these solid nodes have to be evaluated in order to finish the streaming step. The links that connect the fluid nodes and solid nodes are intersected by the boundaries. The intersection ratio that serves as a boundary-location parameter is used to calculate the PDFs at the solid nodes. However, this pseudo-boundary-fitting technique still has two pitfalls. First, the stability of the scheme is sensitive to the intersection ratio that varies at different locations and is not controllable when the boundary geometry is complex. Second, since the mesh node still does not conform

---

[a] Author to whom correspondence should be addressed. Electronic addresses: lec67@pitt.edu and leitaochen@gmail.com.
[b] Electronic mail: las149@pitt.edu

to the boundary, the actual boundary locations interpreted by the velocity solution may deviate from the boundary locations predefined by the users. As noted by Mei and Luo[5] in their study, the vorticity generation and stress distribution on the boundary are very sensitive to its geometry.

However, the flexibility of the LBM should not be considered to be intrinsically limited by the shortcomings of these curved boundary techniques. It has been observed by many authors[13–16] that the LBM is only a special finite-difference version of the continuous Boltzmann equation, and the coupling between the discretization of the momentum space and the discretization of configuration space is not necessary to recover the Navier-Stokes (NS) equation. This is the theoretical proof that the discrete Boltzmann equation (DBE) can be solved on an arbitrarily shaped mesh. Lately, many Eulerian solvers that directly solve the DBE on an unstructured mesh have been developed, such as the finite element discrete Boltzmann method (FEDBM)[17–20] and the finite volume discrete Boltzmann method (FVDBM).[21–35] (Some authors named their methods as finite volume lattice Boltzmann methods (FVLBM). However, technically speaking, those methods are FVDBM since they also solve the DBE.) With the use of an unstructured mesh, the grid nodes can perfectly fit the boundaries, and then the aforementioned approximation schemes for a curved boundary (and their disadvantages) can be completely avoided.

When the grid nodes are located on the boundaries, the solution scenario is very similar to what is used on the flat boundaries for conventional LBM. However, the boundary schemes developed for the flat boundaries of LBM are not appropriate for FVDBM. The boundary conditions (BCs) for flat surfaces of LBM can basically be put into two categories according to their mechanisms. The first type is for open boundaries, such as an inlet and outlet, where the flow velocity or density is prescribed.[36–38] This type of scheme can be well represented by Zou-He's technique,[37] which basically solves a closed linear system. The unknowns are the PDFs flowing into the fluid domain and the constraints are usually flow velocity or density on the boundaries. Zou-He's scheme is designed for the D2Q9 lattice and could only handle three unknowns, which will break down when the boundary is encountered with an overconstrained or underconstrained scenario. The overconstrained case happens when both velocity and density are defined on the boundary or a lower-order lattice such as D2Q7 is applied. In both cases, the number of constraints is more than the number of unknown PDFs. On the other hand, when a higher-order lattice model is applied, such as D2Q13, the BC will be underconstrained because the number of unknown PDFs flowing inward is more than the number of known constraints. However, it is mentioned in the previous paragraph that the decoupling between the momentum space discretization and configuration space discretization leads to freedom in the choice of mesh. This also provides freedom in the choice of lattice model. More importantly, the nature of such decoupling theoretically guarantees the independency between the mesh and lattice models and therefore makes it possible that any combination of different meshes and lattices can be solved by FVDBM. It would be ideal if the boundary scheme can also maintain this independency. However, none of the previously proposed boundary schemes have been able to use different lattice models regardless of how the configuration space is discretized.

The second type of BC is a wall boundary. The most commonly used technique is the bounce-back (BB) scheme. It has been argued by many authors,[36,38–40] however, that BB is only first-order and will not maintain a non-slip condition on the wall when the relaxation time is large. The halfway BB addresses this issue, but it is still not applicable to FVDBM, since in the unstructured Eulerian solver, there is no rigorous definition of where the halfway-wall should be located. Other flat boundary schemes, such as the second-order scheme for an inclined wall,[41] the interpolation[42] and extrapolation[43] schemes for a flat wall, and the nonreflecting scheme for open boundaries,[44] cannot be adopted by FVDBM simply because they are all developed for the LBM approach on a uniform Lagrangian grid network for streaming while such a network does not exist in the Eulerian FVDBM on an unstructured mesh.

A clearer and more logical way to categorize the physical BCs is by the type of macroscopic variables, density or velocity, and by whether their values (Dirichlet condition) or derivatives (Neumann condition) are prescribed. For example, an open boundary with velocities prescribed and a wall boundary both belong to the Dirichlet velocity boundary. Such a way of classification is also adopted by conventional CFD tools. Therefore, the boundary treatment developed for the FVDBM should inherit the same way to realize different physical BCs. In addition, the study of nonslip velocity, which is only for the wall boundary that is a sub-category of the Dirichlet velocity boundary, should also be extended to other types of velocity boundaries and even density boundaries. However, none of the boundary schemes developed for the LBM has been studied thoroughly in terms of their abilities to preserve the Dirichlet density and velocity boundaries in general.

There are two types of FVDBM in terms of different unstructured meshes: vertex-centered and cell-centered. The major differences are how the control volume (CV) is constructed and where the solution is stored. The method used in this paper is cell-centered not only because it is easier to program and construct the CV (a cell-centered CV has three vertices while the vertex-centered CV has an arbitrary number of vertices) but also provide better accuracy than the vertex-centered mesh.[45] Most FVDBM approaches[22,23,25–30,33,46] in the literature are solved on a vertex-centered mesh. Their boundary schemes, called the co-volume method, do not work for the cell-centered mesh due to the difference in the mesh structure. There are a few FVDBM papers[24,32,34,35] that use a cell-centered mesh but still adopt the Zou-He scheme or something similar on the boundaries. That older scheme works for the unstructured mesh only because a D2Q9 lattice model is applied and will fail when a different lattice is applied. One of the goals of this paper is to develop a boundary scheme that works for all lattice models in a unified way. Guo *et al.*[47] developed a boundary scheme that is completely different from the Zou-He scheme and applied it in their hydro-thermal LBM simulation.[48] In that paper, the total PDF on the boundary is decomposed into two parts: the equilibrium and non-equilibrium. The equilibrium part can be computed from the BCs and the non-equilibrium part is extrapolated from an interior node. This method provides a possibility that all PDFs, no matter how many there are, can be evaluated on the boundary without considering matching the number of unknown PDFs and the number of constraints. However, such a scheme was developed for the conventional LBM and must be modified for the cell-centered FVDBM. Ghasemi and Razavi[31] used Guo's scheme for their cell-centered FVDBM model on a quadrilateral mesh. Unfortunately, they only partially applied this scheme to their thermal model, while still using the Zou-He scheme for their hydrodynamic BC.

Moving beyond this past work, in this paper, a hydrodynamic boundary treatment that is at least first-order is developed for the cell-centered FVDBM on a triangular unstructured mesh. This approach can incorporate different lattice models and mesh structures and realize the Dirichlet velocity and density BCs in a unified way. Such a treatment can also work independently with different internal schemes that solve the DBE. In addition, the proposed boundary treatment can preserve Dirichlet conditions on the boundaries up to machine accuracy independently of different conditions, which include different relaxation times, different lattice models, different mesh resolutions, different shapes of boundaries, and different internal schemes.

II. CELL-CENTERED FINITE VOLUME DISCRETE BOLTZMANN MODEL

A. Velocity discretization

Equation (1) shows the DBE with the Bhatnagar-Gross-Krook (BGK) collision model,[49] which is obtained by discretizing the particle velocity space of the continuous Boltzmann equation with a finite number of velocity components

$$\frac{\partial f_\alpha}{\partial t} + \boldsymbol{e}_\alpha \cdot \boldsymbol{\nabla} f_\alpha = -\frac{1}{\tau}\left(f_\alpha - f_\alpha^{eq}\right) \quad \alpha = 0, 1, \ldots, X, \tag{1}$$

where f_α is the PDF in the αth direction of discretized particle velocity space, and τ is the relaxation time. The lattice velocity and equilibrium PDF are $\boldsymbol{e}_\alpha$ and f_α^{eq}, whose definitions are determined by the type of lattice model applied. In the present work, three two-dimensional lattice models are chosen, D2Q7, D2Q9, and D2Q13.

For the D2Q7 lattice,

$$\boldsymbol{e}_\alpha = \begin{cases} [0,0]\, c & \alpha = 0 \\ \left[\cos\left((\alpha-1)\dfrac{\pi}{3}\right), \sin\left((\alpha-1)\dfrac{\pi}{3}\right)\right] c & \alpha = 1-6 \end{cases}, \tag{2a}$$

$$f_\alpha^{eq} = \omega_\alpha \rho \left[1 + \frac{\boldsymbol{e}_\alpha \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2c_s^2} \right], \tag{2b}$$

$$\omega_\alpha = \begin{cases} \dfrac{1}{2} & \alpha = 0 \\ \dfrac{1}{12} & \alpha = 1-6 \end{cases}. \tag{2c}$$

Then, for D2Q9,

$$\boldsymbol{e}_\alpha = \begin{cases} [0,0]\,c & \alpha = 0 \\ \left[\cos\left((\alpha-1)\dfrac{\pi}{2}\right), \sin\left((\alpha-1)\dfrac{\pi}{2}\right)\right] c & \alpha = 1-4 \\ \left[\cos\left((\alpha-5)\dfrac{\pi}{2}+\dfrac{\pi}{4}\right), \sin\left((\alpha-5)\dfrac{\pi}{2}\right)+\dfrac{\pi}{4}\right] \sqrt{2}c & \alpha = 5-8 \end{cases}, \tag{3a}$$

$$f_\alpha^{eq} = \omega_\alpha \rho \left[1 + \frac{\boldsymbol{e}_\alpha \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2c_s^2} \right], \tag{3b}$$

$$\omega_\alpha = \begin{cases} \dfrac{4}{9} & \alpha = 0 \\ \dfrac{1}{9} & \alpha = 1-4 \\ \dfrac{1}{36} & \alpha = 5-8 \end{cases}. \tag{3c}$$

At last, for D2Q13,

$$\boldsymbol{e}_\alpha = \begin{cases} [0,0]\,c & \alpha = 0 \\ \left[\cos\left((\alpha-1)\dfrac{\pi}{2}\right), \sin\left((\alpha-1)\dfrac{\pi}{2}\right)\right] c & \alpha = 1-4 \\ \left[\cos\left((\alpha-5)\dfrac{\pi}{2}+\dfrac{\pi}{4}\right), \sin\left((\alpha-5)\dfrac{\pi}{2}\right)+\dfrac{\pi}{4}\right] \sqrt{2}c & \alpha = 5-8 \\ \left[\cos\left((\alpha-9)\dfrac{\pi}{2}\right), \sin\left((\alpha-9)\dfrac{\pi}{2}\right)\right] 2c & \alpha = 9-12 \end{cases}, \tag{4a}$$

$$f_\alpha^{eq} = \omega_\alpha \rho \left[1 + \frac{\boldsymbol{e}_\alpha \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2c_s^2} + \frac{(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})^3}{2c_s^6} - \frac{3(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})(\boldsymbol{u}\cdot\boldsymbol{u})}{2c_s^4} \right], \tag{4b}$$

$$\omega_\alpha = \begin{cases} \dfrac{3}{8} & \alpha = 0 \\ \dfrac{1}{12} & \alpha = 1-4 \\ \dfrac{1}{16} & \alpha = 5-8 \\ \dfrac{1}{96} & \alpha = 9-12 \end{cases}, \tag{4c}$$

where c is the constant lattice speed for isothermal flow and usually chosen to be unity; c_s is the speed of sound. For D2Q7,

$$c_s^2 = c^2/4. \tag{2d}$$

For D2Q9,

$$c_s^2 = c^2/3, \tag{3d}$$

and for D2Q13,

$$c_s^2 = c^2/2. \tag{4d}$$

The relation between relaxation time and flow viscosity is defined as

$$\upsilon = \tau c_s^2. \tag{5}$$

By substituting Eqs. (2d), (3d), or (4d) into (5), the viscosity for the different lattice models can be obtained. It can be seen that the speed of sound for each lattice above is different. Therefore, in order to achieve the same viscosity, each different lattice requires a different relaxation time. Meanwhile, the macroscopic flow variables are recovered from the PDF as follows:

$$\begin{bmatrix} \rho \\ \rho \boldsymbol{u} \end{bmatrix} = \sum_{\alpha=0}^{X} \begin{bmatrix} f_\alpha \\ \boldsymbol{e}_\alpha f_\alpha \end{bmatrix}. \tag{6}$$

B. Spatial discretization and cell-centered finite volume formulation

In terms of solution techniques, the finite volume method ignores the type of element: triangle, rectangle, hexagon, or even mixed. In addition, the triangular element is more adaptive to complex boundaries than other polygonal elements, especially when the boundaries are highly curved or have sharp corners. As a result, the triangular mesh is chosen to decompose the configuration space. In order to solve Eq. (1) with the finite volume method, a CV has to be selected to establish the finite volume formulation. In the present work, the CV is chosen to be the triangle itself, with V_1, V_2, and V_3 the three vertices and P the centroid. The solution is stored at point P, not the vertices, so such a scheme is called cell-centered. The counterpart of a cell-centered scheme is a vertex-centered scheme, in which the solution is stored at the vertices and the CV is a polygon that surrounds each vertex. By choosing the cell-centered scheme, the effort for creating polygonal CVs, therefore, can be saved.

In order to initialize the formulation, Eq. (1) is integrated over the CV and the order of the terms is slightly rearranged as

$$\int_{CV} \frac{\partial f_\alpha}{\partial t} dA = \int_{CV} \frac{1}{\tau} \left(f_\alpha^{eq} - f_\alpha\right) dA - \int_{CV} \boldsymbol{e}_\alpha \cdot \boldsymbol{\nabla} f_\alpha dA. \tag{7}$$

The first and second integral can be simplified with the cell-averaged value within the CV, namely,

$$\int_{CV} \frac{\partial f_\alpha}{\partial t} dA = \frac{\partial \overline{f}_\alpha}{\partial t} A_{CV}, \tag{8}$$

$$\int_{CV} \frac{1}{\tau} \left(f_\alpha^{eq} - f_\alpha\right) dA = \frac{1}{\tau} \left(\overline{f}_\alpha^{eq} - \overline{f}_\alpha\right) A_{CV}, \tag{9}$$

where A_{CV} is the area of the CV. Since P is the centroid, it can be assumed that the PDF at the centroid is equal to the averaged PDF

$$\overline{f}_\alpha = f_\alpha (P), \tag{10}$$

$$\overline{f}_\alpha^{eq} = f_\alpha^{eq} (P). \tag{11}$$

Equations (8) and (9) then become

$$\int_{CV} \frac{\partial f_\alpha}{\partial t} dA = \frac{\partial \overline{f}_\alpha}{\partial t} A_{CV} = \frac{\partial f_\alpha (P)}{\partial t} A_{CV}, \tag{12}$$

$$\int_{CV} \frac{1}{\tau} \left(f_\alpha^{eq} - f_\alpha\right) dA = \frac{1}{\tau} \left(\overline{f}_\alpha^{eq} - \overline{f}_\alpha\right) A_{CV} = \frac{1}{\tau} \left[f_\alpha^{eq} (P) - f_\alpha (P)\right] A_{CV}. \tag{13}$$

The last integral in Eq. (7), physically speaking, is actually the net flux of the PDF through the boundaries of the CV according to the Gauss divergence theorem. Since the CV is a triangle, it is equal to the sum of the PDF flux through each of the three faces

$$\int_{CV} \boldsymbol{e}_\alpha \cdot \boldsymbol{\nabla} f_\alpha dA = \sum_{i=1}^{3} F_{\alpha,i}, \tag{14}$$

where $F_{\alpha,i}$ is the PDF flux on ith face of the triangular CV. The cell-centered finite volume formulation of Eq. (1) then becomes

$$T = C - F, \tag{15}$$

where T is the temporal term, F is the flux term, and C is the collision term

$$T = \frac{\partial f_\alpha(P)}{\partial t}, \tag{16}$$

$$C = \frac{1}{\tau}\left[f_\alpha^{eq}(P) - f_\alpha(P)\right], \tag{17}$$

$$F = \frac{1}{A_{CV}} \sum_{i=1}^{3} F_{\alpha,i}. \tag{18}$$

C. Flux calculation

In order to calculate F in Eq. (18), the flux on each face $F_{\alpha,i}$ is needed. Figure 1 shows the stencil for calculating the flux on one of the faces of triangle P, along with its neighbor on that face, triangle Q. Then, the flux calculation on the other two faces can be repeated in the same way.

In Fig. 1, C is the center point of the V_2V_3 face. The dashed line passes through point C and is orthogonal to the face. The unity normal to the face is **n** and $\boldsymbol{e}_\alpha$ is the lattice velocity and also the direction of the local flux. Therefore, the flux through the current face is calculated as follows:

$$F_{\alpha,i} = f_\alpha(C)(\boldsymbol{e}_\alpha \cdot \boldsymbol{n}) L_i, \tag{19}$$

where $f_\alpha(C)$ is the PDF at point C and L_i is the length of the face. It is important that such a flux is advective due to its hyperbolic nature. Therefore, the problem of evaluating $f_\alpha(C)$ is actually a typical 1D Riemann problem of a traveling wave with the dashed line as the axis of wave propagation. By projecting $\boldsymbol{e}_\alpha$ onto the dashed line, we have

$$\widehat{\boldsymbol{e}}_\alpha = \boldsymbol{n}\,|\boldsymbol{e}_\alpha| \cos\beta. \tag{20}$$

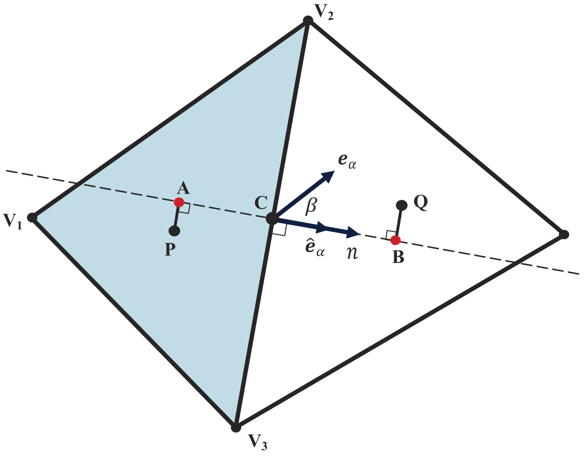


FIG. 1. Two-point stencil for flux calculation.

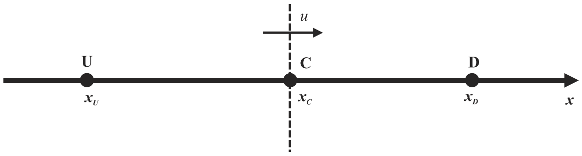


FIG. 2. Riemann problem across the cell interface.

Points A and B are the projected points of centroids P and Q, respectively, onto the dashed line. Therefore, we have the Riemann problem that a wave moves from left to right along the dashed line at the wave speed, u,

$$u = |\widehat{e}_\alpha| . \tag{21}$$

The face center, point C, in Fig. 1 now represents the position of cell interface in Fig. 2. Points U and D are the centers of the upwind and downwind cells, respectively. However, whether point A or B is upwind or downwind depends on the direction of the local lattice velocity. Therefore, we specify the distances

$$\begin{cases} |\boldsymbol{x}_C - \boldsymbol{x}_U| = |\boldsymbol{x}_C - \boldsymbol{x}_A| , & |\boldsymbol{x}_C - \boldsymbol{x}_D| = |\boldsymbol{x}_C - \boldsymbol{x}_B| \quad (\boldsymbol{e}_\alpha \cdot \boldsymbol{n}) > 0 \\ |\boldsymbol{x}_C - \boldsymbol{x}_U| = |\boldsymbol{x}_C - \boldsymbol{x}_B| , & |\boldsymbol{x}_C - \boldsymbol{x}_D| = |\boldsymbol{x}_C - \boldsymbol{x}_A| \quad (\boldsymbol{e}_\alpha \cdot \boldsymbol{n}) \le 0 \end{cases} , \tag{22}$$

while it always holds that

$$|\boldsymbol{x}_U - \boldsymbol{x}_D| = |\boldsymbol{x}_A - \boldsymbol{x}_B| . \tag{23}$$

The PDFs at U and D are also determined by the lattice velocity as

$$\begin{cases} f_\alpha(U) = f_\alpha(A) , & f_\alpha(D) = f_\alpha(B) \quad (\boldsymbol{e}_\alpha \cdot \boldsymbol{n}) > 0 \\ f_\alpha(U) = f_\alpha(B) , & f_\alpha(D) = f_\alpha(A) \quad (\boldsymbol{e}_\alpha \cdot \boldsymbol{n}) \le 0 \end{cases} . \tag{24}$$

In Fig. 1, it can be seen that A is the closest point to P on the dashed line (the same is true for points B and Q). So we can assume

$$f_\alpha(A) = f_\alpha(P) , \quad f_\alpha(B) = f_\alpha(Q). \tag{25}$$

Then, the Lax-Wendroff scheme[50] shows that

$$f_\alpha(C) = f_\alpha(U) + \left(\frac{|\boldsymbol{x}_C - \boldsymbol{x}_U|}{|\boldsymbol{x}_D - \boldsymbol{x}_U|} - \frac{u\Delta t}{2\,|\boldsymbol{x}_D - \boldsymbol{x}_U|} \right) [f_\alpha(D) - f_\alpha(U)] . \tag{26}$$

It can be seen that Eq. (26) degrades to a first-order upwind scheme if the second term on the right hand side is omitted. Even though such a scheme is more complicated, it is chosen over the first-order upwind scheme because the internal scheme must be kept on the second-order to show that the boundary scheme is at least first-order.

Finally, Eq. (26) is used in Eq. (19) to evaluate the PDF flux on each face and then summed over the three faces to yield the net flux for one CV in Eq. (18).

D. Collision term and time marching

The evaluation of collision in Eq. (17) is straightforward, since only the equilibrium PDF must be computed. f_α^{eq} for different lattices (D2Q7, D2Q9, and D2Q13) should be obtained with the according equations from Eqs. (2), (3), or (4).

It has been suggested by Lee[18] that the $\upsilon - \tau$ relation is influenced by the temporal scheme. In order to maintain the relation in Eq. (5), the time-marching scheme must be at least second-order. Here, the second-order Runge-Kutta method (RK2) is applied. From Eqs. (15) and (16), it can be developed that

$$\frac{\partial f_\alpha(P)}{\partial t} = T = C - F. \tag{27}$$

Then, the RK2 scheme gives

$$f_\alpha^{n+1}(P) = f_\alpha^n(P) + \Delta t\left(\frac{1}{2}K_1 + \frac{1}{2}K_2\right), \tag{28}$$

$$K_1 = T_\alpha^n = C_\alpha^n - F_\alpha^n, \tag{29}$$

$$K_2 = T_\alpha^* = C_\alpha^* - F_\alpha^* = \left(C_\alpha - F_\alpha\right)\big|_{f_\alpha^* = f_\alpha^n + \Delta t K_1}. \tag{30}$$

It should be noted that the letter P is omitted in Eqs. (29) and (30) in order to show that the update of the local PDFs is also affected by nearby values due to the advection. In addition, for the sake of computing K_2, along with the PDFs, the density, velocity, and equilibrium PDFs should also be updated.

Two stability criteria set the upper limit for the size of the time step. The first is that Δt should be comparable to the size of the relaxation time in order to capture the correct physics due to collision, which is given as

$$\Delta t < 2\tau. \tag{31}$$

Another criterion is the Courant-Friedrichs-Lewy (CFL) condition. Given the stencil in Fig. 2, the wave should travel no more than half of the upwind cell for one time step, so

$$CFL < 0.5. \tag{32}$$

In order to maintain an absolute stability, the definition of the CFL suggested by Patil[34] is applied and slightly modified to

$$CFL = \frac{\Delta t\left(|\boldsymbol{e}_\alpha|_{\max} + |\boldsymbol{U}|_{\max}\right)}{\frac{A_{\min}}{L_{\min}^x + L_{\min}^y}}, \tag{33}$$

where $|\boldsymbol{U}|_{\max}$ is the magnitude of maximum macroscopic velocity in the computational domain, $A_{\min}$ is the area of the smallest triangle, and $L_{\min}^x$ and $L_{\min}^y$ are the projected lengths of the smallest triangle in the x and y directions.

III. BOUNDARY TREATMENT

In defining the necessary boundary treatment, it can be seen that only the advection term requires BCs since the collision term is localized at the centroids of the CVs and there is no centroid located on the boundaries. For the advection term, the same flux calculation scheme for internal faces developed in Sec. II C is also applied to the faces on the boundaries. This will guarantee the order of accuracy is consistent throughout the entire computational domain and also the smooth transition of the solution between the boundaries and interior region. It can be seen in Fig. 1 that such a treatment requires one neighbor triangle on each side of a face. However, there will be only one neighbor for the face on the boundaries, which demands a special numerical treatment that is able to evaluate the flux on the boundaries and recover the Dirichlet BCs, like velocity and density, both with the desired accuracies. The boundary treatment in this paper consists of two parts. First, a ghost triangle attached to the boundary face but outside of the computational domain is created. Second, the PDF at the centroid of the ghost triangle is evaluated with the physical BCs and information from the interior region. After that, the fluxes through all the faces, including the faces on the boundaries, are computed identically with the method described in Sec. II C regardless of which face is on the boundaries.

A. Evaluation of the PDF for a ghost triangle

In Fig. 3, the triangle P has one of its faces on the boundary with two nodes N_1 and N_2. C is the center point between N_1 and N_2, namely,

$$\left|\boldsymbol{x}_C - \boldsymbol{x}_{N_1}\right| = \left|\boldsymbol{x}_C - \boldsymbol{x}_{N_2}\right|. \tag{34}$$

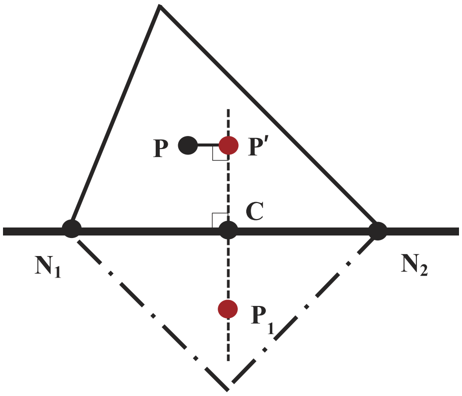


FIG. 3. Ghost triangle on boundary.

The dashed line passes through the point C and is orthogonal to the face. P' is the projected point of P onto the dashed line. The ghost triangle, which is drawn using the dotted-dashed line, is constructed such that its centroid P_1 is on the dashed line and has a distance to the point C such that

$$\left|\boldsymbol{x}_C - \boldsymbol{x}_{P_1}\right| = \left|\boldsymbol{x}_C - \boldsymbol{x}_{P'}\right|. \tag{35}$$

This way of constructing the ghost triangle will ensure that P_1 itself is its projected point onto the dashed line and therefore reduces the error of evaluating the PDF at P_1.

Since point C is equally spaced between P' and P_1, then with linear extrapolation, it holds that

$$f_\alpha\left(P_1\right) = 2f_\alpha\left(C\right) - f_\alpha\left(P'\right). \tag{36}$$

The PDF at P' is evaluated as internal triangles with the same assumption as Eq. (25), so

$$f_\alpha\left(P'\right) = f_\alpha\left(P\right). \tag{37}$$

Point C is also equally spaced between N_1 and N_2, so then with linear interpolation

$$f_\alpha\left(C\right) = \frac{1}{2}\left[f_\alpha\left(N_1\right) + f_\alpha\left(N_2\right)\right]. \tag{38}$$

Combining Eqs. (36)–(38),

$$f_\alpha\left(P_1\right) = f_\alpha\left(N_1\right) + f_\alpha\left(N_2\right) - f_\alpha\left(P\right). \tag{39}$$

Now, the only unknowns are the PDFs at the two boundary nodes N_1 and N_2. Since the PDFs at all boundary nodes will be evaluated in the same way, Subsection III B will illustrate the procedure for one example node on the boundary.

B. Evaluation of the nodal PDF on the boundary

Consider a single boundary node, N, as shown in Fig. 4. P_1, P_2, P_3, and P_4 are the centroids of all of the triangles connected to N, and d_1, d_2, d_3, and d_4 the distances between the corresponding centroids and N. It should be noted that even though there are four connected triangles in Fig. 4, the number of such connected triangles could be arbitrary.

For any variable θ, its value at point N can be extrapolated with a function Ψ such that

$$\theta\left(N\right) = \Psi\left[\theta\left(P\right)\right] = \sum_{j=1}^{M}\frac{\theta\left(P_j\right)}{d_j}\Bigg/\sum_{j=1}^{M}\frac{1}{d_j}, \tag{40}$$

where M is the total number of connected triangles. This extrapolation scheme is fully compatible with different boundary geometries and local mesh structures. It works on both flat and curved boundaries, and it treats each boundary node in the same way without considering how many triangles are connected to it or how each connected triangle is sized or shaped.

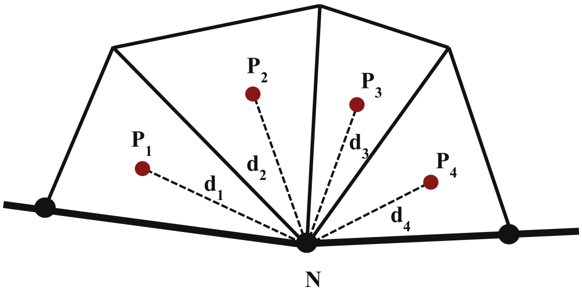


FIG. 4. Boundary node and its connected triangles.

Instead of manipulating the total PDF at the boundary nodes, here it is chosen to decompose the total PDF into an equilibrium part and non-equilibrium part

$$f_\alpha(N) = f_\alpha^{eq}(N) + f_\alpha^{neq}(N). \tag{41}$$

The equilibrium part is obtained by using Eqs. (2), (3), or (4), depending on the type of lattice applied, with given macroscopic variables that are defined in the BCs. Therefore, all of the components of the equilibrium PDFs can be computed once no matter what type of lattice model is applied. In Guo's scheme,[47] $f_\alpha^{neq}(N)$ is computed using extrapolation along the directions of characteristic velocities. Here, we modify it by using Eq. (40), in which θ is replaced by f_α^{neq}. All of the components of non-equilibrium PDFs are computed in the same way, and they are lattice-type-independent. Therefore, the entire scheme can be applied to any type of lattice, including the three lattices used in this paper.

In addition, Eq. (41) can also recover the Dirichlet BCs (velocities and densities) up to machine accuracy. For any location in the flow domain, it holds that

$$\sum_{\alpha=0}^{X}\begin{bmatrix} f_\alpha \\ \boldsymbol{e}_\alpha f_\alpha \end{bmatrix} = \sum_{\alpha=0}^{X}\begin{bmatrix} f_\alpha^{eq} \\ \boldsymbol{e}_\alpha f_\alpha^{eq} \end{bmatrix} = \begin{bmatrix} \rho \\ \rho\boldsymbol{u} \end{bmatrix}. \tag{42}$$

Replacing f_α with $f_\alpha^{eq} + f_\alpha^{neq}$, then Eq. (42) is equivalent to

$$\sum_{\alpha=0}^{X}\begin{bmatrix} f_\alpha^{neq} \\ \boldsymbol{e}_\alpha f_\alpha^{neq} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathit{0} \end{bmatrix}. \tag{43}$$

Performing the recovery of macroscopic variables on Eq. (41),

$$\sum_{\alpha=0}^{X}\begin{bmatrix} f_\alpha(N) \\ \boldsymbol{e}_\alpha f_\alpha(N) \end{bmatrix} = \sum_{\alpha=0}^{X}\begin{bmatrix} f_\alpha^{eq}(N) \\ \boldsymbol{e}_\alpha f_\alpha^{eq}(N) \end{bmatrix} + \sum_{\alpha=0}^{X}\begin{bmatrix} f_\alpha^{neq}(N) \\ \boldsymbol{e}_\alpha f_\alpha^{neq}(N) \end{bmatrix} = \begin{bmatrix} \rho(N) \\ \rho(N)\boldsymbol{u}(N) \end{bmatrix}. \tag{44}$$

Since $f_\alpha^{eq}(N)$ is computed with given BCs, the recovery of density and velocity from $f_\alpha^{eq}(N)$ will simply yield the BCs as

$$\sum_{\alpha=0}^{X}\begin{bmatrix} f_\alpha^{eq}(N) \\ \boldsymbol{e}_\alpha f_\alpha^{eq}(N) \end{bmatrix} = \begin{bmatrix} \rho^{BC}(N) \\ \rho^{BC}(N)\boldsymbol{u}^{BC}(N) \end{bmatrix}. \tag{45}$$

According to Eq. (40), $f_\alpha^{neq}(N)$ is linearly extrapolated using the internal centroids, at which points Eq. (43) is automatically satisfied. So

$$\sum_{\alpha=0}^{X}\begin{bmatrix} f_\alpha^{neq}(N) \\ \boldsymbol{e}_\alpha f_\alpha^{neq}(N) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathit{0} \end{bmatrix}. \tag{46}$$

Substituting Eqs. (45) and (46) into (44),

$$\begin{bmatrix} \rho(N) \\ \rho(N)\boldsymbol{u}(N) \end{bmatrix} = \begin{bmatrix} \rho^{BC}(N) \\ \rho^{BC}(N)\boldsymbol{u}^{BC}(N) \end{bmatrix}. \tag{47}$$

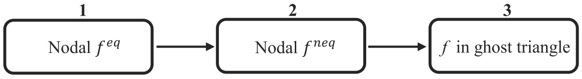


FIG. 5. Realization of proposed boundary treatment.

Therefore, the macroscopic variables at any boundary node are the exact Dirichlet BCs at that node.

C. Realization of different physical boundary conditions

With the developed framework, all commonly used BCs can be realized in a unified way. Fig. 5 shows the three-step procedure to implement the proposed boundary treatment. The first step is to evaluate the equilibrium part of the nodal PDFs on the boundaries, which varies slightly depending on what type of physical BC is present (density or velocity). Then the non-equilibrium part is evaluated exactly in the same way by Eq. (40). Steps 1 and 2 can be switched. The last step is the evaluation of the PDFs for the ghost triangle. For the Dirichlet BCs, Eq. (39) is applied to linearly extrapolate the PDFs for the ghost triangle. Modification is introduced for the non-Dirichlet BCs (periodic and fully developed boundary), which will be explained later.

1. Periodic boundary condition

The faces on the periodic boundaries are treated as internal ones, so the fluxes through those faces are computed with the same method developed in Sec. II C. Neither the PDF evaluation for the ghost triangle nor the nodal PDF evaluation is needed. However, extra effort is required to find the periodic neighbor triangles to play the role of ghost triangles.

2. Velocity boundary condition

The velocity boundaries contain the wall boundaries, including a stationary wall and/or a moving wall, and open velocity boundaries, such as the velocity inlet and velocity outlet. However, when the velocity is prescribed on these boundaries, the density is usually unknown. Here, it is chosen to extrapolate the density at the boundary nodes with Eq. (40). The given velocity and extrapolated density can then be used to compute the equilibrium PDFs. It should be noted that, when the velocity is given on the boundaries, the x and y velocities are not required to be defined at the same time. Sometimes, only the x or y velocity is defined, which will be seen in the simulation in Sec. IV B. Under such a scenario, the velocity component that is not defined on the boundary is extrapolated with Eq. (40).

3. Density boundary condition

A density or pressure boundary usually occurs at the inlet or outlet of flows. However, the velocity there is typically unknown. Equation (40) is used again to extrapolate the velocity from the interior triangles. Then the equilibrium PDFs are calculated with the given density and extrapolated velocity.

4. Other boundary conditions

Sometimes density and velocity might both be defined on the boundary, which will cause a physically overconstrained condition, and therefore, a numerically overconstrained condition for the conventional LBM when D2Q9 lattice or other lower-order model is applied. However, in the current numerical framework, this type of BC will not lead to failure. The step of density or velocity extrapolation can simply omitted and the given density and velocity can then be used to calculate the equilibrium PDFs. However, the extrapolations of non-equilibrium PDFs are still needed.

Finally, there is another type of BC, the fully developed or zero-gradient BC, which is usually applied to the outlet of a long channel flow. For such a BC, Eq. (40) is used to extrapolate the total PDFs at boundary nodes without breaking it into the equilibrium and non-equilibrium parts. This is equivalent to forcing the equality of the PDFs at the boundary and the PDFs of the nearest upstream neighbors to eliminate the gradient on the boundary. Since the gradient of the PDFs is zero on the boundary, which is also true for density and velocity. Additionally, during the step of evaluating PDFs for the ghost triangle, the linear extrapolation given in Eq. (39) should be replaced by $f_\alpha(P_1) = f_\alpha(P)$, which will eliminate the gradient of PDFs between the ghost triangle and its neighbor. As a result, the nodal PDFs on the boundary are not needed for evaluation of the PDFs of the ghost triangle. However, they are still kept for the sake of recovering macroscopic variables on the boundaries.

IV. NUMERICAL RESULTS AND DISCUSSIONS

In Sec. II C for the flux calculation, it is assumed that the PDFs at the projected point are directly copied from the centroid. However, such an evaluation contains an error that depends on the mesh quality. Taking points A and P in Fig. 1, for example, the distance between A and P will become larger if triangle P is badly skewed, and then the error introduced by Eq. (25) will increase. There are basically two strategies to minimize this error. The first is using a linear or higher-order interpolation scheme in the place of the direct copying in order to better evaluate the PDFs at the projected points. The second is to control the mesh quality for the entire computational domain. It can be imagined that the assumption by Eq. (25) contains no errors in the ideal case that all triangles are equilateral. Given that higher-order interpolation is complicated on an unstructured mesh and, more importantly, the major task of this paper is to demonstrate the boundary treatment, here we choose the latter. There are two types of mesh used in this paper, which are shown in Fig. 6.

The first type of mesh consists of all isosceles right triangles (IRT) with identical size and shape. This type of mesh is not able to capture the curved boundary. The mesh consists of many squared blocks that have four triangles in each of them. For the one shown in Fig. 6(a), there are nine blocks in each direction. So, it is a 9 × 9 IRT mesh. The other type of mesh (Fig. 6(b)) is more general with arbitrarily shaped and sized triangles and therefore can achieve the exact body-fitting for complex boundary geometries.

The first difference between these two types of mesh is the local connectivity at each node. For the IRT mesh, eight identical triangles share each interior node and four are connected to each boundary node except for the corner nodes (two triangles). For the general mesh, each node is shared by arbitrary number of arbitrarily shaped and sized triangles, no matter where it is located.

The second difference between Figs. 6(a) and 6(b) is the accuracy associated with the equality assumption stated in Eq. (25). When the two-point stencil for the IRT mesh is constructed as shown

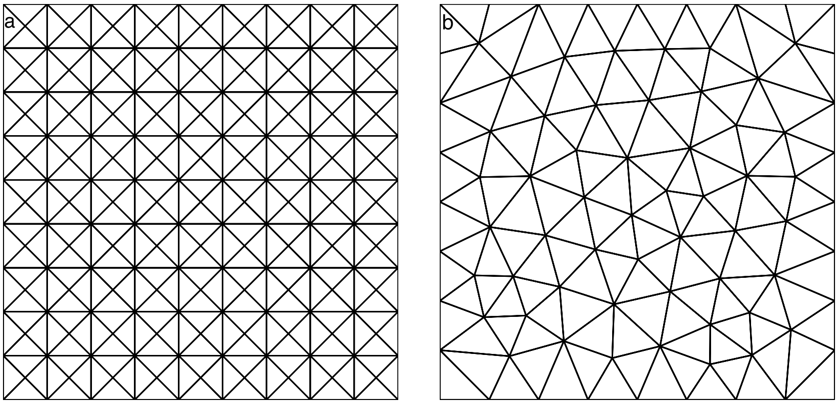


FIG. 6. Two different types of mesh: (a) IRT mesh. (b) General mesh.

in Fig. 1 on each face of a triangle, the projected point for the longest face will be the centroid itself, but for the other two faces, will deviate from the centroid. Therefore, Eq. (25) contains no error on the longest face but errors with consistent magnitude on the other two faces throughout the entire mesh. In contrast, Eq. (25) always generates errors for all three faces of each triangle in the general mesh. Therefore, simulation with the IRT mesh could achieve a better accuracy than the general mesh. In the following numerical tests, the IRT mesh is used for problems with solely flat boundaries to generate most quantitative conclusions about the proposed boundary treatment, while the general mesh is used for the problem with curved boundaries and demonstrates that the boundary treatment developed in this paper is capable of handling complex geometries to the same level as the flat boundaries.

However, both two meshes are completely unstructured and have exactly the same data structure and the finite volume solution techniques. Therefore, the proposed FVDBM scheme and boundary treatment can solve a problem on any triangular mesh in the same way, including the two types of meshes used in this paper and other triangular meshes not mentioned in this paper. Besides the IRT and general meshes, the properties of other types of triangular meshes are not discussed in this paper, since their effects on the solution are not unique to the FVDBM and can be found in many publications.

A. Couette flow

The first modeling case is the Couette flow that has a moving wall on the top and a stationary wall at the bottom. Periodic boundaries are chosen for the left and right boundaries. The analytical solution is very simple and has a linear x velocity and zero y velocity. The moving wall is given a horizontal moving velocity such that the Mach number is Ma = 0.087 with a D2Q9 lattice applied. The relaxation time τ is fixed to maintain Reynolds number as Re = 1. The first simulation is performed on the IRT mesh that has 72 square blocks in the y direction. Figure 7 is the steady-state x velocity along with the analytical solution on the plane that stands in the middle between the left and right boundaries with the velocity and distance normalized.

It can be seen that the numerical solution matches the analytical solution, with the x velocity linearly changing from the bottom wall to the top wall. The grid-convergence study is performed with the aforementioned Re and Ma. Four IRT meshes, with 9, 18, 36, and 72 blocks in the y direction, respectively, are applied in order to have the mesh resolution doubled every time. The L_2 norms of relative errors of x and y velocities for the different meshes are shown in the log-log space in Fig. 8. The slope of each line represents the corresponding order of accuracy. The lines that represent first-order and second-order accuracies are also displayed. For the x velocity, the average slope is 0.6, which is lower than first order. The average slope for the y velocity is 1.2, which is

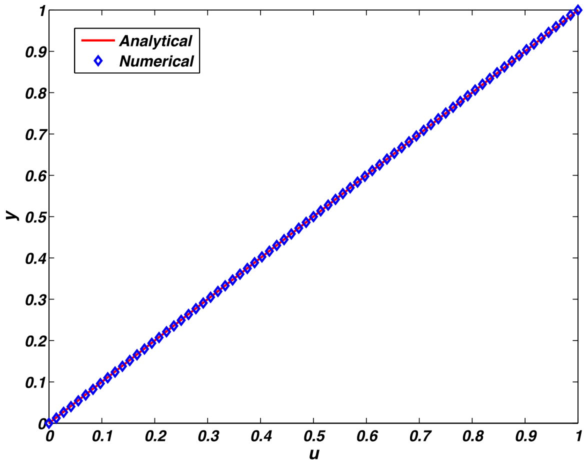


FIG. 7. Solution of Couette flow.

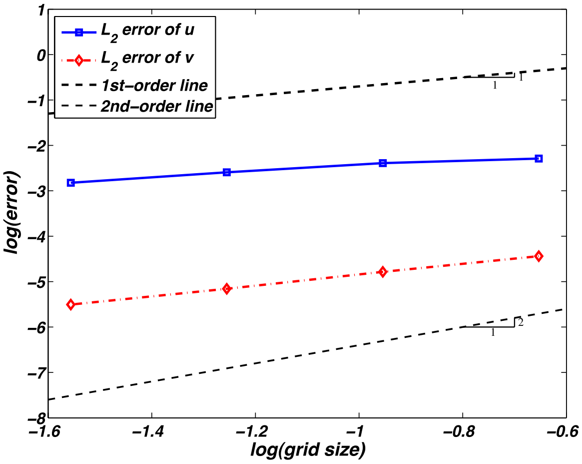


FIG. 8. Grid convergence analysis of Couette flow.

higher than first order. Taking another average between the x and y velocities will give the general order of accuracy for the velocity in the presented Couette flow, which is 0.9. It has been discussed that the assumption by Eq. (25) will introduce a nontrivial error for the IRT mesh. Therefore, a better assumption will definitely yield an order of accuracy higher than 0.9. However, for any simulation, the general order of accuracy is the lowest order of accuracy between the internal scheme and boundary scheme. The improvement of the assumption aforementioned is only related to the internal scheme and has nothing to do with the boundary treatment. Therefore, the actual order of accuracy solely contributed by the boundary treatment would be larger than 0.9. Considering other inevitable errors involved in the simulation but not associated with the boundary treatment (e.g., errors due to discretization of phase space, compressibility error, etc.), it can concluded that the assorted velocity and periodic boundary treatment in the Couette flow is at least first-order accurate with a good level of confidence.

Next, the slip velocities on the top and bottom boundaries are studied. In order to expand the concept of slip velocity to any velocity boundary, a variable called boundary velocity deviation (BVD) is created, which is the difference between prescribed boundary velocity and real velocity on the boundary recovered from the velocity solution. It is possible that mesh refinement will affect the value of BVD. Therefore, the average BVD in the x and y directions on the top and bottom boundaries versus different levels of mesh refinement used in the previous grid convergence study is shown in the log-log space in Fig. 9.

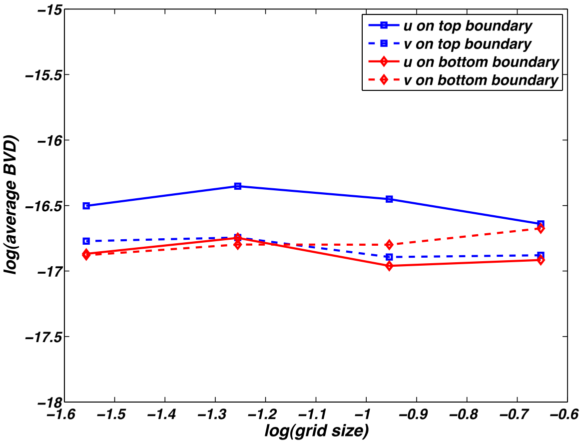


FIG. 9. Average BVDs on top and bottom boundaries vs. the grid size in Couette flow.

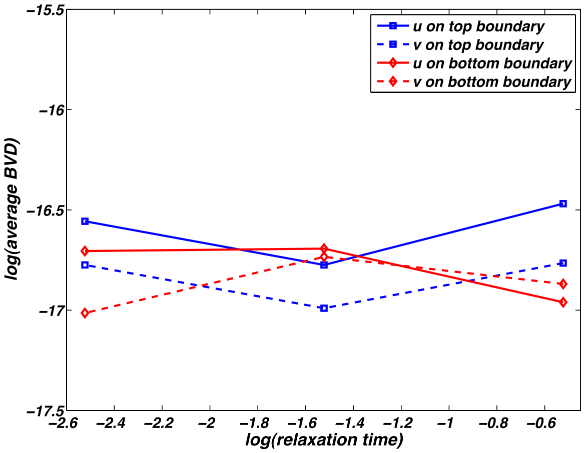


FIG. 10. Average BVDs on top and bottom boundaries vs. the relaxation time in Couette flow.

The first thing that can be observed is that the BVD in the x direction on top boundary is larger than all other BVDs located on zero-velocity boundaries, which is expected since it is the only non-zero velocity definition among all velocity boundaries. Therefore, the larger the velocity defined on the boundary, the larger the BVD. Then, the more important conclusion is that the orders of the magnitudes of all BVDs are independent of grid size and are all lower than 10^{-16}, the machine epsilon.

It is argued by many authors[36,38,40] that, for the conventional LBM simulations, the magnitude of slip velocity is affected by the size of relaxation time. From this assertion, it is expected that one can conclude the effect of relaxation time on BVD for the proposed FVDBM and boundary treatment. A series of simulations is carried out by having three different relaxation times, 0.3, 0.03 and 0.003, respectively. Everything else is kept the same in the model as in the previous simulations, and the same mesh with 18 blocks in the y direction is used. The BVDs on the top and bottoms boundaries of the Couette flow versus different sizes of relaxation time are illustrated in Fig. 10. It can be seen that the proposed boundary treatment is able to prevent BVDs from being regularly influenced by the change of relaxation time. It also can preserve BVDs up to machine accuracy under a wide range of relaxation times, which is extremely helpful when a small relaxation time is needed for the high-Re flows.

B. Poiseuille flow

Poiseuille flow is another typical benchmark problem, since the analytical solutions of velocity and pressure are obtainable. There are several ways to set up the physical BCs for Poiseuille flow. Usually, different densities are defined at the inlet and outlet to drive the flow. However, for the sake of demonstrating that the currently proposed boundary treatment is capable of solving some peculiar cases that the conventional LBM approach fails to solve and also helping to determine the analytical solution at the outlet, a physically overconstrained case is purposely constructed here by adding velocity BCs at the inlet and outlet besides the existing density BCs. The types of physical BCs on each surface of a rectangular domain are listed in Table I, in which the filled or empty circle indicates whether the corresponding physical quantity is prescribed or not, respectively.

The top and bottom boundaries are set with zero velocities in both the x and y directions; the inlet is given a symmetric parabolic velocity profile in the x direction and a constant density, which is higher than the density value defined at the outlet, and zero velocities in the y direction are both given at the inlet and outlet. It should be noted that the density BCs alone are able to reach a steady-state solution, in which the x velocity at the inlet will be known. Therefore, in order to ease the physically overconstrained condition, the x velocity that is going to be defined at the inlet should be carefully chosen. A D2Q9 lattice model is chosen for the simulation with the relaxation

TABLE I. Physical definitions of boundary conditions for Poiseuille flow.

	Inlet (left)	Top wall	Outlet (right)	Bottom wall
Density	●	○	●	○
X velocity	●	●	○	●
Y velocity	●	●	●	●

time $\tau = 0.1$. The density at the inlet and outlet, ρ_{in} and ρ_{out}, is 2 and 1.95, respectively. Under a 0.05 density difference, the symmetric parabolic velocity profile at the inlet, without contradicting the density BCs, is chosen to have its maximum velocity to be 0.0595. Then, the maximum Mach number in the system is roughly 0.1. In order to quantitatively measure the error, the analytical solution at steady state should be obtained in the first place. Specifically, the analytical x velocity profile on the outlet boundary and the density distribution along the centerline of the channel are selected as the benchmarks. For any channel flow with an inlet and outlet of the same size, mass conservation states that

$$\rho_{in}\overline{U}_{in} = \rho_{out}\overline{U}_{out}, \tag{48}$$

where $\overline{U}_{in}$ and $\overline{U}_{out}$ are the average x velocities in the flow direction at the inlet and outlet. For the inlet, since a symmetric parabola of velocity is defined, $\overline{U}_{in}$ is simply two-thirds of the maximum velocity. As aforementioned, ρ_{in} and ρ_{out} are defined by the density BCs. Therefore, $\overline{U}_{out}$ can be calculated by Eq. (48). Then, the analytical velocity profile at the outlet can be constructed by creating another symmetric parabola with its maximum value to be $1.5\overline{U}_{out}$. The analytical density distribution along the centerline is relatively easier to find, which is a straight line connecting ρ_{in} and ρ_{out}. The simulations are performed with the IRT meshes. Figs. 11 and 12 show the velocity profile at the outlet and the density distribution along the centerline of the channel, respectively, along with their analytical counterparts when a steady-state solution is reached on an IRT mesh with 72 blocks in the y direction. It can be seen that the numerical solution matches the analytical solution very well.

A grid-convergence study is also carried out. As with the Couette flow, four IRT meshes with 9, 18, 36, and 72 blocks in the y direction are used. The L_2 norms of the relative errors of x velocity and density with respect to the analytical solution for different meshes are shown in the log-log space in Fig. 13. It can be observed that all errors decrease while the mesh resolution is being doubled. The slope of each line represents the corresponding order of accuracy. The line that represents second-order accuracy is shown in the dashed line in Fig. 13. For velocity, the average slope is 1.86, which is slightly lower than second order. Considering the error introduced by the assumption in Eq. (25) as well as other errors not associated to the boundary treatment, the actual order of

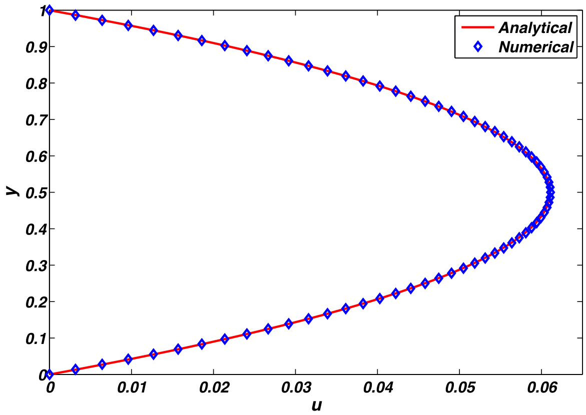


FIG. 11. Velocity profile at the outlet in Poiseuille flow.

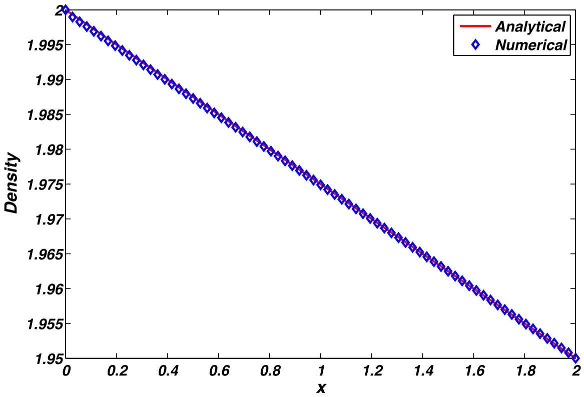


FIG. 12. Density distribution along the centerline in Poiseuille flow.

accuracy for velocity contributed by the boundary treatment would be approximately second order. The same analysis is repeated for density, resulting in an average slope of 2, which is exactly second order. Again with Eq. (25) and other numerical errors being considered, the order of accuracy for the solution of density would be higher than second order when solely considering the contribution of the boundary treatment. So conservatively speaking, the velocity and density boundary treatment demonstrated in the Poiseuille flow is second-order accurate.

By comparing Fig. 13 with Fig. 8, it can be seen that the Poiseuille flow has a higher order of accuracy than the Couette flow. The reason is that the left and right boundaries in Couette flow are periodic. Therefore, the inlet and outlet are considered as internal faces. As has been stated previously, the flux calculations for internal faces are subject to the error due to the assumption in Eq. (25). By contrast, the Poiseuille flow is bounded by the Dirichlet BC on each of its boundaries. Therefore, the accuracy of the solution on the Dirichlet boundaries is no longer affected by the internal flux calculation scheme. In addition, the accurate solution on the Dirichlet boundary will propagate into the internal region and improve the overall accuracy. As a result, although with the same assumption in Eq. (25) for the flux calculation scheme, the Poiseuille flow that has Dirichlet boundaries is able to yield a more accurate result than the Couette flow that has periodic boundaries in the x direction. An approach to increase the order of accuracy of a periodic boundary is to use a more accurate assumption than Eq. (25) for the flux calculation, which, however, is independent of the proposed boundary treatment.

Next, the BVD on the boundaries where the velocity BC is prescribed is studied, which includes the inlet, outlet, and top and bottom boundaries. It should be noted that the proposed boundary treatment is able to preserve any component of a velocity vector that is defined on the boundary,

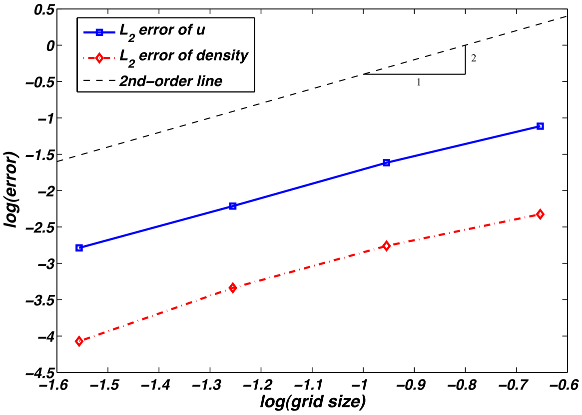


FIG. 13. Grid convergence analysis of different errors in Poiseuille flow.

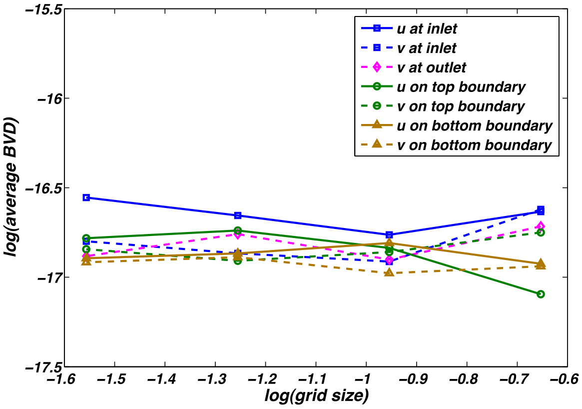


FIG. 14. Average BVDs on different velocity boundaries vs. the grid size in Poiseuille flow.

without being affected by whether other components of the same vector are prescribed or not. The BVD in the x and y directions on each of the velocity boundaries versus different levels of mesh refinement in the log-log coordinate is shown in Fig. 14. First, the BVD in the x direction on the inlet boundary is larger than the BVDs on all other boundaries where the zero velocities are defined, which is similar to the Couette flow. Second, the orders of the magnitudes of all BVDs are below 10^{-16} and independent of grid size. Finally, it should be noted that, although only the y velocity is defined at the outlet while the x velocity is unconstrained, it is still able to preserve the y velocity at the outlet up to machine accuracy.

A similar concept for the density on the boundaries can be created, the boundary density deviation (BDD), and then studied. There are only two boundaries with density prescribed, the inlet and outlet. The average BDDs on the inlet and outlet boundaries versus different levels of mesh refinement in log-log coordinates are shown in Fig. 15. It can be seen that the BDDs at the inlet and outlet are slightly greater than 10^{-16}, and more importantly, independent of mesh resolution. By comparing Figs. 14 and 15, it can be seen that BDDs are basically one order larger than BVDs. This is because the magnitude of density is one order larger than velocities in the solution.

By summarizing the BVD and BDD studies for the Couette and Poiseuille flows, a new way of treating density and velocity on the boundaries can be provided. Instead of viewing density and velocity as two different variables, one a scalar and the other a vector, a new vector can be created that combines these two variables, $\boldsymbol{V} = (\rho, \boldsymbol{U}) = (\rho, u, v)$. Every component in $\boldsymbol{V}$ is required to calculate f_α^{eq}. When some components of $\boldsymbol{V}$ are defined on the boundary, namely, a Dirichlet BC is applied for those components, the rest of $\boldsymbol{V}$ is calculated with the proposed extrapolation scheme, Eq. (40). In addition, the proposed boundary treatment does not put a constraint on whether the Dirichlet condition must be applied or not on any component of $\boldsymbol{V}$. Actually, each component of $\boldsymbol{V}$ is independent from the others. When a Dirichlet BC is applied to any component of $\boldsymbol{V}$ on a

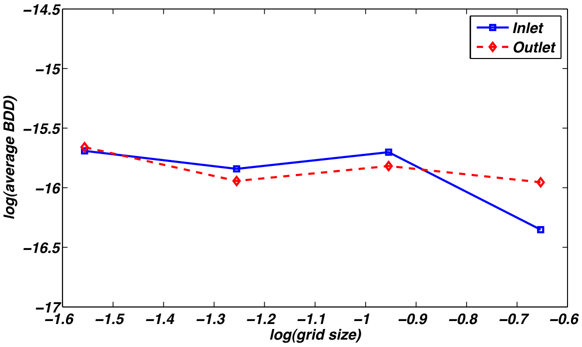


FIG. 15. Average BDDs on inlet and outlet boundaries vs. the grid size in Poiseuille flow.

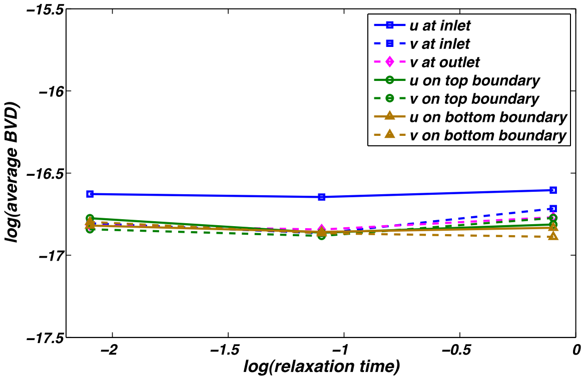


FIG. 16. Average BVDs on different velocity boundaries vs. the relaxation time in Poiseuille flow.

boundary, the value of that component on that boundary can be preserved up to machine accuracy, regardless of whether other components of $\boldsymbol{V}$ are also defined with a Dirichlet condition or extrapolated. Therefore, it can be concluded that the proposed boundary treatment can also solve some underconstrained case that the conventional LBM cannot. A typical example is the same Poiseuille flow problem, but with only density prescribed at the inlet and outlet. Such a case is physically solvable but will nullify the LBM with Zou-He's boundary scheme. Due to the lack of defined velocity in the y direction at the inlet or outlet, a closure cannot be formed. However, in contrast, such a system can be easily solved by the proposed boundary treatment and therefore extends the applications of lattice Boltzmann simulations by making physically solvable cases also numerically solvable.

The effect of relaxation time on BVDs and BDDs for the Poiseuille flow is also studied. Three sizes of relaxation time are chosen for the study, 0.8, 0.08, and 0.008. The same IRT mesh that has 36 blocks in the y direction is used for all simulations. The results of BVDs and BDDs are shown in Figs. 16 and 17, respectively. In Fig. 16, it can be observed that the BVD on each velocity boundary is below 10^{-16} and basically unchanged for different sizes of relaxation time. In Fig. 17 for density, the proposed boundary treatment is also able to preserve BDDs up to machine accuracy for different relaxation times.

C. Lid-driven cavity flow

The third simulation case is the square cavity flow driven by a moving lid. This is also a very important benchmark problem since it is highly nonlinear, caused by the singularities at the two corners of the moving lid. The major task in this part is to demonstrate the proposed FVDBM and

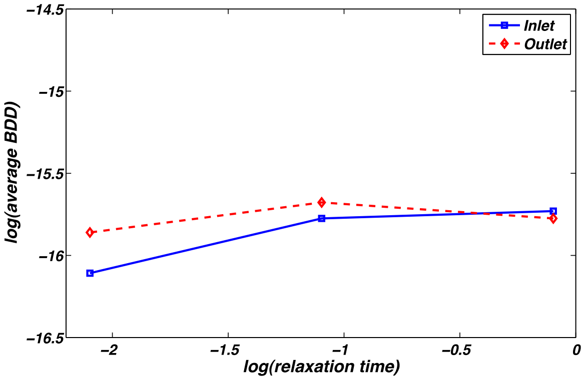


FIG. 17. Average BDDs on inlet and outlet boundaries vs. the relaxation time in Poiseuille flow.

boundary treatment's ability for handling different lattice models. Therefore, all three lattice models introduced in Sec. II will be tested on the same cavity flow where the velocity of the lid is 0.1 and Re = 100. The top boundary is the lid moving from left to right. All other boundaries are walls with zero velocities.

In order to facilitate the discussion of the following numerical results, a phenomenon that is unique to the DBE first should be presented. The DBE (Eq. (1)) is a hyperbolic equation with the collision term acting as a stiff source term and will become stiffer if the relaxation time decreases. In order to capture such a stiff effect correctly, the time step size must be proportional to the relaxation time, which is controlled by Eq. (31). In addition, for any flow simulation, the time step size should be proportional to the grid size, which is due to the CFL condition. Therefore, the grid size and relaxation time are linked together by the time step size. As a result, they should also be proportional in size in order to have a stable and accurate result.

The first simulation was completed with a D2Q7 lattice model on a 35×35 IRT mesh. Its velocity profiles on the vertical and horizontal mid planes are shown in Fig. 18, along with the benchmark solution from Ghia *et al.*[51] Another simulation was carried out on the same model, which has the same velocity of the moving lid, the same domain size and mesh resolution, and the same boundary treatment, but with a D2Q9 lattice. The D2Q9 lattice should yield a much more accurate result than D2Q7 because the nine-velocity lattice model is able to recover the NS equation, while the seven-velocity lattice model does not have enough velocity components to do so. However, the results show that the solutions from D2Q9 are only slightly better than that from D2Q7. It should be noted that in order to achieve the same viscosity and Re, the D2Q9 model requires a smaller relaxation time than the D2Q7 model (see Eqs. (2d), (3d), (4d), and (5)), if the size of the cavity and velocity of the moving lid are kept unchanged. Since the grid size must be proportional to the relaxation time, D2Q7 requires a lower mesh resolution than D2Q9 due to its larger relaxation time. Conversely, the stiffness effect of D2Q7 will be modeled with a smaller error than D2Q9 if the same mesh resolution is applied. Therefore, for the D2Q7 lattice, the error due to insufficient lattice velocity components will be mitigated by the relatively high mesh resolution. (However, a D2Q9 lattice is still chosen over a D2Q7 lattice due to its better accuracy even though the D2Q7 lattice requires 20% less computation time.)

The third simulation was also completed with all parameters maintained at the same values in the previous ones, but with a D2Q13 lattice model. It can be seen that D2Q9 and D2Q13 yield the same solution in Fig. 18. The D2Q13 model should yield more accurate results than D2Q9 since it has a higher-order rotational invariance.[52] However, the results fail to prove this. The reason is the same as for D2Q7 and D2Q9. The D2Q13 model requires a smaller relaxation time than the D2Q9 model in order to have the same viscosity. Therefore, D2Q13 requires a higher mesh resolution than D2Q9. Conversely, a larger error of stiffness will be introduced to D2Q13 than D2Q9 if the same mesh resolution is applied. Therefore, for D2Q13 lattice, the extra accuracy gained by better discretization of velocity space will be offset by relatively poor mesh resolution. In addition, the D2Q13 lattice requires almost 50% more computation time than D2Q9 since it has more PDFs to be solved. As a result, in simulations of flows with low Knudsen numbers, a D2Q9 lattice is a

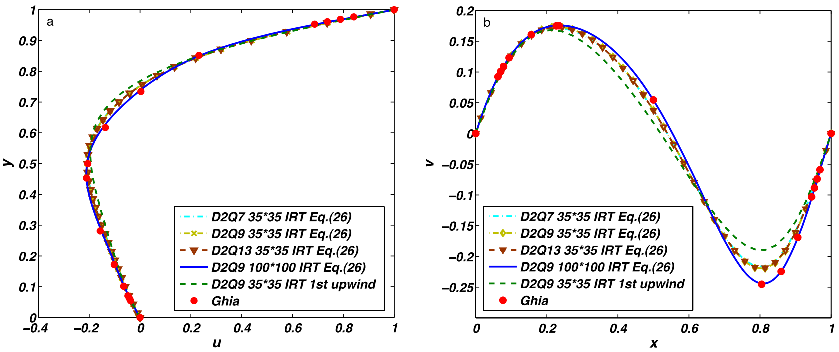


FIG. 18. Velocity profiles on the vertical and horizontal mid planes of Re = 100 lid-driven cavity flow. (a) u vs. y. (b) v vs. x.

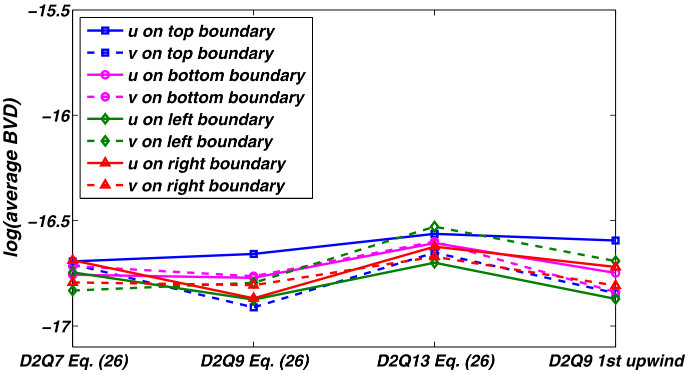


FIG. 19. Average BVDs on each boundary in cavity flow.

better choice than D2Q7 and D2Q13 when considering both accuracy and efficiency. This is why the D2Q9 lattice dominates 2D LBM simulations. A much more accurate solution with a D2Q9 lattice can be obtained with a better refined mesh. The results with the D2Q9 lattice solved on a 100 × 100 IRT mesh are also shown in Fig. 18. It can be seen that the results match Ghia's solution satisfactorily. At last, in order to exhibit the boundary treatment's ability of working with different internal schemes independently, a simulation with the first-order upwind scheme for flux calculation was done on the same model with a D2Q9 lattice. It is well known that a significant numerical viscosity will be introduced by first-order upwind scheme for the hyperbolic problems, which is why it can be seen that in Fig. 18 it generates more flattened velocity profiles than the proposed flux scheme, Eq. (26).

Next, the effect of different lattice models on how the velocities are preserved on the boundaries from the simulations above is studied. The common logarithms of average BVDs in the x and y directions on each boundary versus different lattice models are shown in Fig. 19. The first conclusion, which is the same as that for Couette and Poiseuille flows, is that the BVD in the x direction on the top boundary basically has the largest magnitude since the x velocity on the top boundary is the only nonzero boundary definition. The second conclusion is that, although the D2Q13 lattice has slightly larger BVDs than D2Q7 and D2Q9, the proposed boundary treatment is able to preserve Dirichlet BC up to machine accuracy regardless of what type of lattice model is applied.

The effect of different internal schemes on the BVDs of all boundaries for the same D2Q9 lattice is also studied. It can be seen that, although the first-order upwind scheme yields much larger error than the proposed flux scheme, Eq. (26), for the internal solution (see Fig. 18), the errors of the solutions on the boundaries given by two schemes have the same order of magnitude, both below 10^{-16}. Therefore, the proposed boundary treatment is not only able to work properly with different internal schemes but also able to preserve Dirichlet BC up to machine accuracy when different internal schemes are utilized.

D. Flow over a circular cylinder

The last simulation case is the flow over a circular cylinder, which is included to show that the proposed FVDBM model and boundary treatment are able to handle curved boundaries properly by incorporating the general triangular mesh. The geometric configuration of the computational domain and the generated mesh are shown in Fig. 20, in which D is the diameter of the cylinder. The mesh is generated by a self-developed meshing tool whose major component is triangle,[53] a Delaunay triangulation package.

The physical BCs for each boundary are defined as follows. The left boundary, which is the inlet of the flow, is defined as a velocity boundary by giving an x velocity and zero y velocity. The outlet on the right is considered a fully developed boundary. The top and bottom surfaces are periodic. A D2Q9 lattice is applied for the simulation. The Reynolds number of the flow is kept as Re = 20, and the Mach number at the inlet is Ma = 0.1155.

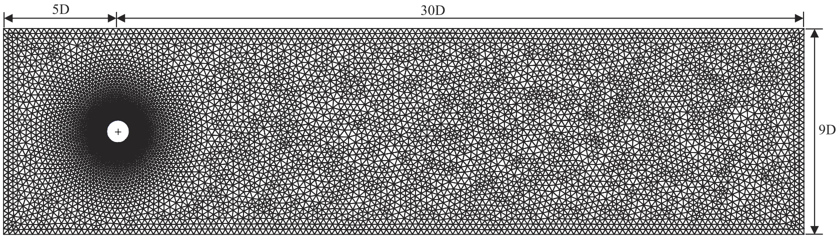


FIG. 20. The configuration and mesh of the flow over a circular cylinder.

Figure 21 shows the distributions of BVDs both in x and y directions on the cylinder surface when a steady-state solution is reached, in which the angle of 0 and 180 represents the front and back stagnation points, respectively. It can be seen that the magnitude of BVDs at any location in the cylinder surface is smaller than 10^{-16}, and therefore, the Dirichlet BC on the circular surface can be preserved up to machine accuracy.

Since the DBE is naturally transient, a steady flow and an unsteady flow are simulated with the same solution techniques. A steady flow is obtained by solving an unsteady flow with sufficient iterations until it reaches a steady state if it has one. Therefore, by calculating the transient average BVDs on the circular cylinder as well as on the inlet boundary during the iterations, which is shown in Fig. 22, it can be concluded that the proposed boundary treatment is also able to preserve the Dirichlet BC up to machine accuracy for unsteady flow both on the flat and curved boundaries.

Another important aspect of CFD is the force evaluation on the boundaries. The total force on any boundary surface is calculated as Eq. (49)

$$\boldsymbol{F} = \int \boldsymbol{S} \cdot \boldsymbol{n} dA, \tag{49}$$

where $\boldsymbol{S}$ is the stress tensor on the boundary surface and $\boldsymbol{n}$ is the unit normal vector pointing into the fluid domain. The stress tensor $\boldsymbol{S}$ is defined in Eq. (50)

$$\boldsymbol{S} = -p\boldsymbol{I} + \rho\upsilon \left[\nabla \boldsymbol{U} + (\nabla \boldsymbol{U})^T\right]. \tag{50}$$

The first term in Eq. (50) is the stress tensor due to pressure p, which is calculated by Eq. (51)

$$p = c_s^2 \rho. \tag{51}$$

Since a D2Q9 lattice model is applied in this simulation, it can be obtained that $p = \rho/3$. The second term in Eq. (50) is the deviatoric stress tensor caused by the flow motion. The pressure stress tensor is easy to calculate, since density can be directly obtained from the solution. However, the deviatoric stress tensor has to be numerically evaluated since the gradient of velocity, $\nabla \boldsymbol{U}$, is not

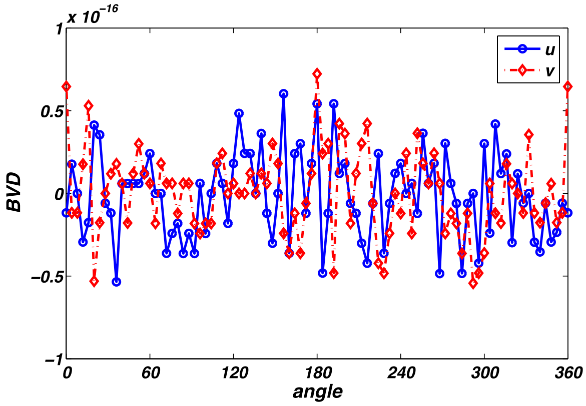


FIG. 21. The BVD distribution on the circular cylinder.

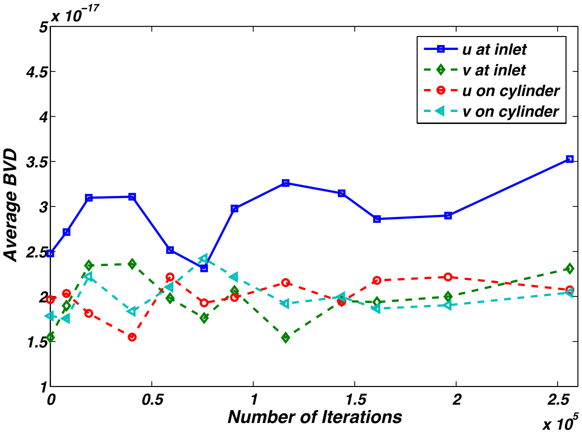


FIG. 22. Average BVDs on the inlet boundary and the circular cylinder, and their histories during iterations.

available in the solution. An extrapolation procedure must be applied since the solution only exists on one side of the boundary. Therefore, the evaluation of the deviatoric stress tensor suffers further numerical error that depends on the order of accuracy of the applied extrapolation scheme. In order to calculate $\nabla \boldsymbol{U}$, two points at different locations have to be selected. Figure 23 shows the two points selected for the extrapolation on one segment of the boundary. The first selected point is C, which is the middle point between boundary nodes N_1 and N_2. Then, the velocity at point C can be evaluated as

$$\boldsymbol{U}_c = \frac{1}{2}\left(\boldsymbol{U}_{N_1} + \boldsymbol{U}_{N_2}\right). \tag{52}$$

The deviatoric stress tensor has two components, the normal stress tensor and shear stress tensor. Both components require that the direction for calculating the gradient must be perpendicular to the boundary. The dashed line that passes the point C and is perpendicular to the boundary is drawn in Fig. 23. The second point must be on the dashed line as well. By using the same method for the flux calculation, the second point is chosen to be the projected point P' of centroid P onto the dashed line. Finally, the gradient of velocity can be evaluated with $\boldsymbol{U}_C$, $\boldsymbol{U}_{P'}$, and the distance between two points. As discussed thoroughly in the previous results, the proposed boundary treatment can exactly recover the Dirichlet BCs. So, $\boldsymbol{U}_C$ is the exact velocity of the boundary regardless of a moving boundary or stationary boundary. As a result, the accuracy of the extrapolation scheme is majorly determined by how accurate the velocity at point P' can be evaluated. Here, it is simply chosen that

$$\boldsymbol{U}_{P'} = \boldsymbol{U}_P. \tag{53}$$

Other higher-order methods are also applicable, but will not be discussed in this paper.

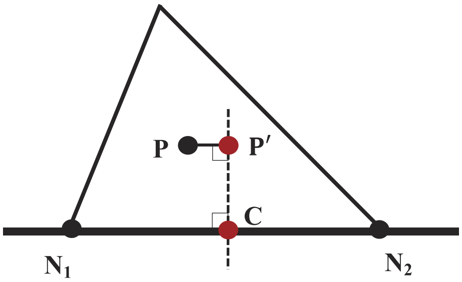


FIG. 23. Calculation of $\nabla \boldsymbol{U}$ on boundary.

TABLE II. Comparison of geometrical and dynamical parameters for flow past a circular cylinder at Re = 20.

Authors	L/a	θ_s	C_D
Coutanceau and Bouard[54]	1.860	44.80	. . .
Dennis and Chang[55]	1.880	43.70	2.045
Nieuwstadt and Keller[56]	1.786	43.37	2.053
Fornberg[57]	1.820	. . .	2.000
Calhoun[58]	1.820	. . .	2.190
Ye *et al.*[59]	1.840	. . .	2.030
He and Doolen[2]	1.842	42.96	2.152
Mei and Shyy[60]	1.804	. . .	. . .
Lee and Lin[18]	1.834	. . .	2.106
Ubertini *et al.*[23]	. . .	. . .	2.090
Zarghami *et al.*[24]	1.820	42.50	2.205
Patil and Lakshmisha[34]	1.884	42.81	1.949
Present	**1.817**	**46.52**	**2.250**

After the total force is evaluated, the drag coefficient on the circular cylinder is calculated as

$$C_D = \frac{F_D}{\rho U^2 a}, \tag{54}$$

where F_D is the component of total force in the flow direction, ρ is the average density of the flow, U is the inlet velocity, and a is the radius of the cylinder. The drag coefficient from the current simulation along with the wake length and separation angle is compared to other published data in Table II. It can be seen that the present results are in the range of the other published data, with slight deviations. The errors mainly originate from two sources. One is from the assumption stated in Eq. (53). Similar to the scheme for the flux calculation, better evaluation at the projected point will improve the accuracy. Another source of error is from the calculation of the velocity gradient on the boundary. It can be seen that, with only two points, the calculated derivative will not located on the boundary, but a point in the flow domain between points P' and C in Fig. 23. The situation can be improved by adding the third point on the dashed line.

V. CONCLUSIONS

In this paper, a self-contained cell-centered FVDBM model that includes the solver and boundary treatment has been developed in detail. Due to its ability to incorporate any type of unstructured triangular mesh, the model can easily solve problems with curved or more complex boundaries. The proposed boundary treatment is developed thoroughly, as well as an analysis and discussion of its properties through different numerical tests. The proposed boundary treatment has the following properties that can help improve the solver's abilities to the possibly fullest extent.

- Different physical BCs, which include the Dirichlet type (density and velocities boundaries) and the non-Dirichlet type (periodic and fully developed boundaries), can be realized in a unified way with minor changes to the three-step procedure (Fig. 5).
- The underconstrained and overconstrained BCs encountered by the conventional LBM can be easily realized by the proposed boundary treatment in a unified way.
- Different lattice models can be incorporated in a unified way.
- The proposed boundary treatment is at least first-order accurate for assorted BCs.
- The proposed boundary treatment can work with different internal schemes.
- The proposed boundary treatment can preserve the Dirichlet condition for any component of the vector $\boldsymbol{V} = (\rho, \boldsymbol{U}) = (\rho, u, v)$ up to machine accuracy regardless of changes made in a number of different factors, as listed below. Therefore, it can provide more accurate evaluations of stress and force on the boundaries.

- The mesh structure on the boundaries (the number of triangles connected to the boundary node, the shapes and sizes of these triangles),
- the geometry of the boundaries (flat or curved),
- the mesh resolution,
- the size of relaxation time,
- the type of lattice applied,
- the type of the internal scheme.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant No. CBET-1233106.

[1] S. Succi, G. Amati, and R. Benzi, "Challenges in lattice Boltzmann computing," J. Stat. Phys. **81**, 5 (1995).
[2] X. He and G. Doolen, "Lattice Boltzmann method on curvilinear coordinates system: Flow around a circular cylinder," J. Comput. Phys. **134**, 306 (1997).
[3] X. He and G. Doolen, "Lattice Boltzmann method on a curvilinear coordinates system: Vortex shedding behind a circular cylinder," Phys. Rev. E **56**, 434 (1997).
[4] O. Filippova and D. Hänel, "Grid refinement for lattice-BGK models," J. Comput. Phys. **147**, 219 (1998).
[5] R. Mei, L. Luo, and W. Shyy, "An accurate curved boundary treatment in the lattice Boltzmann method," J. Comput. Phys. **155**, 307 (1999).
[6] X. Yin and J. Zhang, "An improved bounce-back scheme for complex boundary conditions in lattice Boltzmann method," J. Comput. Phys. **231**, 4295 (2012).
[7] J. Verschaeve and B. Müller, "A curved no-slip boundary condition for the lattice Boltzmann method," J. Comput. Phys. **229**, 6781 (2010).
[8] Q. Chen, X. Zhang, and J. Zhang, "Improved treatments for general boundary conditions in the lattice Boltzmann method for convection-diffusion and heat transfer process," Phys. Rev. E **88**, 033304 (2013).
[9] Z. Guo, C. Zheng, and B. Shi, "An extrapolation method for boundary conditions in lattice Boltzmann method," Phys. Fluids **14**, 2007 (2002).
[10] C. Chang, C. Liu, and C. Lin, "Boundary conditions for lattice Boltzmann simulations with complex geometry flows," Comput. Math. Appl. **58**, 940 (2009).
[11] J. Kang, S. Kang, and Y. Suh, "A dynamic boundary model for implementation of boundary conditions in lattice-Boltzmann method," J. Mech. Sci. Technol. **22**, 1192 (2008).
[12] D. Yu, R. Mei, and W. Shyy, "A unified boundary treatment in lattice Boltzmann method," AIAA Paper No. 2003-0953, 2003.
[13] T. Abe, "Derivation of the lattice Boltzmann method by means of the discrete ordinate method for the Boltzmann equation," J. Comput. Phys. **131**, 241 (1997).
[14] N. Cao, S. Chen, S. Jin, and D. Martínez, "Physical symmetry and lattice symmetry in the lattice Boltzmann method," Phys. Rev. E **55**, R21 (1997).
[15] X. He and L. Luo, "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation," Phys. Rev. E **56**, 6811 (1997).
[16] X. He and L. Luo, "*A priori* derivation of the lattice Boltzmann equation," Phys. Rev. E **55**, R6333 (1997).
[17] T. Lee and C. Lin, "A characteristic Galerkin method for discrete Boltzmann equation," J. Comput. Phys. **171**, 336 (2001).
[18] T. Lee and C. Lin, "An Eulerian description of the streaming process in the lattice Boltzmann equation," J. Comput. Phys. **185**, 445 (2003).
[19] K. E. Wardle and T. Lee, "Finite element lattice Boltzmann simulations of free surface flow in a concentric cylinder," Comput. Math. Appl. **65**, 230 (2013).
[20] Y. Li, E. J. LeBoeuf, and P. K. Basu, "Least-squares finite-element Boltzmann method," Phys. Rev. E **69**, 065701(R) (2004).
[21] H. Chen, "Volumetric formulation of the lattice Boltzmann method for fluid dynamics: Basic concept," Phys. Rev. E **58**, 3955 (1998).
[22] S. Ubertini, G. Bella, and S. Succi, "Lattice Boltzmann method on unstructured grids: Further developments," Phys. Rev. E **68**, 016701 (2003).
[23] S. Ubertini, S. Succi, and G. Bella, "Lattice Boltzmann schemes without coordinates," Philos. Trans. R. Soc., A **362**, 1763 (2004).
[24] A. Zarghami, M. Maghrebi, J. Ghasemi, and S. Ubertini, "Lattice Boltzmann finite volume formulation with improved stability," Commun. Comput. Phys. **12**, 42 (2012).
[25] N. Rossi, S. Ubertini, G. Bella, and S. Succi, "Unstructured lattice Boltzmann method in three dimensions," Int. J. Numer. Methods Fluids **49**, 619 (2005).
[26] G. Peng, H. Xi, and C. Duncan, "Lattice Boltzmann method on irregular meshes," Phys. Rev. E **58**, R4124 (1998).
[27] H. Xi, G. Peng, and S. Chou, "Finite-volume lattice Boltzmann method," Phys. Rev. E **59**, 6202 (1999).
[28] G. Peng, H. Xi, and C. Duncan, "Finite volume scheme for the lattice Boltzmann method on unstructured meshes," Phys. Rev. E **59**, 4675 (1999).
[29] H. Xi, G. Peng, and S. Chou, "Finite-volume lattice Boltzmann schemes in two and three dimensions," Phys. Rev. E **60**, 3380 (1999).
[30] G. Peng, H. Xi, and S. Chou, "On boundary conditions in the finite volume lattice Boltzmann method on unstructured meshes," Int. J. Mod. Phys. C **10**, 1003 (1999).

[31] J. Ghasemi and S. E. Razavi, "On the finite-volume lattice Boltzmann modeling of thermo-hydrodynamics," Comput. Math. Appl. **60**, 1135 (2010).

[32] A. Zarghami, M. J. Maghrebi, and J. Ghasemi, "Finite volume-latticeBoltzmann modeling of viscous flows," Majlesi J. Mech. Eng. **4**, 11 (2011).

[33] M. Stiebler, J. Tölke, and M. Krafczyk, "An upwind discretization scheme for the finite volume lattice Boltzmann method," Comput. Fluids **35**, 814 (2006).

[34] D. V. Patil and K. N. Lakshmisha, "Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh," J. Comput. Phys. **228**, 5262 (2009).

[35] D. V. Patil and K. N. Lakshmisha, "Two-dimensional flow past circular cylinders using finite volume lattice Boltzmann formulation," Int. J. Numer. Methods Fluids **69**, 1149 (2012).

[36] D. R. Noble, S. Chen, J. G. Georgiadis, and R. O. Buckius, "A consistent hydrodynamic boundary condition for the lattice Boltzmann method," Phys. Fluids **7**, 203 (1995).

[37] Q. Zou and X. He, "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model," Phys. Fluids **9**, 1591 (1997).

[38] X. He, Q. Zou, L. Luo, and M. Dembo, "Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model," J. Stat. Phys. **87**, 115 (1997).

[39] D. P. Ziegler, "Boundary conditions for lattice Boltzmann simulations," J. Stat. Phys. **71**, 1171 (1993).

[40] T. Inamuro, M. Yoshino, and F. Ogino, "A nonslip boundary condition for lattice Boltzmann simulations," Phys. Fluids **7**, 2928 (1995).

[41] I. Ginzbourg and D. d'Humières, "Local second-order boundary methods for lattice Boltzmann models," J. Stat. Phys. **84**, 927 (1996).

[42] B. Chun and A. J. C. Ladd, "Interpolated boundary condition for lattice Boltzmann simulations of flows in narrow gaps," Phys. Rev. E **75**, 066705 (2007).

[43] S. Chen and D. Martínez, "On boundary conditions in lattice Boltzmann methods," Phys. Fluids **8**, 2527 (1996).

[44] S. Izquierdo and N. Fueyo, "Characteristic nonreflecting boundary conditions for open boundaries in lattice Boltzmann methods," Phys. Rev. E **78**, 046707 (2008).

[45] D. J. Mavriplis, "Unstructured grid techniques," Annu. Rev. Fluid Mech. **29**, 473 (1997).

[46] Y. T. Chew, C. Shu, and Y. Peng, "On the implementation of boundary conditions in the application of finite volume lattice Boltzmann method," J. Stat. Phys. **107**, 539 (2002).

[47] Z. Guo, C. Zheng, and B. Shi, "Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method," Chin. Phys. **11**, 366 (2002).

[48] Z. Guo, C. Zheng, and B. Shi, "Thermal lattice Boltzmann equation for low Mach number flows: Decoupling model," Phys. Rev. E **75**, 036704 (2007).

[49] P. L. Bhatnagar, E. P. Gross, and M. Krook, "A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems," Phys. Rev. **94**, 511 (1954).

[50] R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, Cambridge, 2002), p. 100.

[51] U. Ghia, K. N. Ghia, and C. T. Shin, "High-re solutions for incompressible flow using the Navier-Stokes equations and multigrid method," J. Comput. Phys. **48**, 387 (1982).

[52] A. Augier, F. Dubois, B. Graille, and P. Lallemand, "On rotational invariance of lattice Boltzmann schemes," Comput. Math. Appl. **67**, 239 (2014).

[53] J. R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and delaunay triangulator," *Applied Computational Geometry: Towards Geometric Engineering*, Lecture Notes in Computer Science Vol. 1148 (Springer-Verlag, Berlin, Germany, 1996), pp. 203–222.

[54] M. Coutanceau and R. Bouard, "Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation: Part 1: Steady flow, part 2: Unsteady flow," J. Fluid Mech. **79**, 231 (1977).

[55] S. C. R. Dennis and G. Z. Chang, "Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100," J. Fluid Mech. **42**, 471 (1970).

[56] F. Nieuwstadt and H. B. Keller, "Viscous flow past circular cylinders," Comput. Fluids **1**, 59 (1973).

[57] B. Fornberg, "A numerical study of steady viscous flow past a circular cylinder," J. Fluid Mech. **98**, 819 (1980).

[58] D. Calhoun, "A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions," J. Comput. Phys. **176**, 231 (2002).

[59] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, "An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries," J. Comput. Phys. **156**, 209 (1999).

[60] R. Mei and W. Shyy, "On the finite difference-based lattice Boltzmann method in curvilinear coordinates," J. Comput. Phys. **143**, 426 (1998).