Commun. Comput. Phys.
doi: 10.4208/cicp.211015.040316a

Vol. **20**, No. 2, pp. 301-324
August 2016

Finite Volume Lattice Boltzmann Method for Nearly Incompressible Flows on Arbitrary Unstructured Meshes

Weidong Li[1,2] and Li-Shi Luo[2,1,*]

[1] *Department of Mathematics & Statistics, Old Dominion University,*
Norfolk, Virginia 23529, USA.
[2] *Computational Science Research Center, Beijing 100193, China.*

Received 21 October 2015; Accepted (in revised version) 4 March 2016

Abstract. A genuine finite volume method based on the lattice Boltzmann equation (LBE) for nearly incompressible flows is developed. The proposed finite volume lattice Boltzmann method (FV-LBM) is grid-transparent, *i.e.*, it requires no knowledge of cell topology, thus it can be implemented on arbitrary unstructured meshes for effective and efficient treatment of complex geometries. Due to the linear advection term in the LBE, it is easy to construct multi-dimensional schemes. In addition, inviscid and viscous fluxes are computed in one step in the LBE, as opposed to in two separate steps for the traditional finite-volume discretization of the Navier-Stokes equations. Because of its conservation constraints, the collision term of the kinetic equation can be treated implicitly without linearization or any other approximation, thus the computational efficiency is enhanced. The collision with multiple-relaxation-time (MRT) model is used in the LBE. The developed FV-LBM is of second-order convergence. The proposed FV-LBM is validated with three test cases in two-dimensions: (a) the Poiseuille flow driven by a constant body force; (b) the Blasius boundary layer; and (c) the steady flow past a cylinder at the Reynolds numbers Re = 10, 20, and 40. The results verify the designed accuracy and efficacy of the proposed FV-LBM.

AMS subject classifications: 76M12, 76M28, 76D05, 82C40

PACS: 47.11.Df, 47.11.Qr, 47.10.ad, 51.10.+y

Key words: Finite volume method, lattice Boltzmann equation, arbitrary unstructured mesh, complex geometry, multi-dimensional scheme.

1 Introduction

In its simplest form, the orthodox lattice Boltzmann method (LBM) is associated with uniform Cartesian meshes due to its tightly coupled discretizations of phase space and

---

[*] Corresponding author. *Email addresses:* `lwd_1982.4.8@163.com` (W. D. Li), `lluo@odu.edu`, `lluo@csrc.ac.cn` (L.-S. Luo)

time [1,2]. In spite of its accuracy and algorithmic simplicity [3–7], the capability for the LBM to *accurately* treat flows with complicated geometries is limited. To overcome this limitation, there have been continuous attempts to develop the lattice Boltzmann equation (LBE) based on finite volume (FV) formulation with unstructured meshes [8–17]. The finite-volume method (FVM) for solving the Navier-Stokes equations is a matured technique (cf., *e.g.*, the review [18] and monograph [19]). A key feature of the FVM is its use of *unstructured* meshes to treat complex geometries with ease [18]. With edge-based (2D) or face-based (3D) data structure, cell-centered FVM can be efficiently implemented. The characteristics of the FVM is much determined by its reconstruction step, in which the fluxes are reconstructed at the cell boundaries from the hydrodynamic variables at cell centers, and the hydrodynamic variables are also allowed to be discontinuous at cell boundaries. This reconstruction step distinguishes FVM from other methods, such as finite difference method (FDM) and finite element method (FEM) and so on.

To justify the present work, we begin with a brief survey of existing work on the development of finite-volume lattice Boltzmann method (FV-LBM) for nearly incompressible flows with low Mach numbers. In the work by Nannelli and Succi [8], the fluxes of the distribution functions at cell boundaries are *directly* computed from their values at cell centers through interpolations without the aforementioned reconstruction step. Though this scheme increases geometrical flexibility, it is only implemented on meshes of quadrilateral cells. To remove this limitation, Peng *et al.* [9,10] extend the scheme to a vertex-centered FV-LBM on triangular meshes. However, none of these schemes [8–10] and their variants [11–13,15,17] is *bona fide* FVM for they lack the centerpiece of the FVM — the reconstruction of fluxes at cell boundaries; they are essentially some recasts of some finite difference schemes [13]. These schemes are severely limited in time step size, thus they are computationally inefficient [13].

More recently, Patil and Lakshmisha developed a genuine FV-LBM on triangular unstructured meshes [16]. The reconstruction in this work uses the least-square method to obtain the gradients at cell centroids and Roe's flux-difference splitting scheme to compute the fluxes at cell boundaries. In addition, the techniques of total variation diminishing (TVD) and limiters are used in the scheme. While Roe's flux-difference splitting scheme is effective for high speed flows, it is problematic for nearly incompressible flows with the Mach number $\mathrm{Ma} \ll 1$, because the numerical dissipation is inversely proportional to the local Mach number Ma [20]. Consequently the scheme is too dissipative and thus inaccurate for nearly incompressible flows when the Mach number $\mathrm{Ma} \ll 1$.

In this work we will develop a finite-volume lattice Boltzmann method on arbitrary unstructured meshes with second-order convergence. The gradients at cell centers and the fluxes at cell boundaries are computed by using the least-square method and the low-diffusion Roe scheme, respectively. In addition, the collision term in the LBE is treated implicitly, and the edge-based data structure is adopted in our implementation. These features of the proposed FV-LBM greatly enhance its computational efficiency.

The remainder of the paper is organized as follows. Section 2 describes the finite-volume formulation of the LBE in sufficient details in four parts. Specifically, Section 2.1

reviews the LBE with the multiple-collision-times (MRT) collision model, which is used in our FV-LBE formulation for its numerical stability and efficiency. Section 2.2 discusses the finite-volume discretization of the LBE. Section 2.3 discusses the time marching method and the implicit treatment of the collision term in the LBE. Section 2.4 describes the ghost cell method used to realize the boundary conditions in the proposed FV-LBM. Section 3 presents the numerical results of this work. Three two-dimensional test cases are used to validate the proposed FV-LBM: (a) Poiseuille flow between two parallel plate driven by a constant body force; (b) Blasius boundary layer over a semi-infinite flat plate; and (c) steady flow past a cylinder in free space with the Reynolds number $\mathrm{Re}=10$, 20, and 40. Finally, Section 4 concludes the paper.

2 Formulation of the FV-LBE

In this section, we will first provide a brief introduction of the LBE with the multiple-time-relaxation collision model due to d'Humières [21], which is the governing equation to be solved. We will then proceed to discuss the finite volume spatial discretization of the LBE, and time marching scheme, and the ghost cell method for boundary conditions.

2.1 The lattice Boltzmann equation

Our starting point is the Boltzmann equation with the discrete velocity set $\{\boldsymbol{\xi}_\alpha\}$ and the collision term from the lattice Boltzmann equation (LBE):

$$\partial_t f_\alpha + \boldsymbol{\xi}_\alpha \cdot \boldsymbol{\nabla} f_\alpha = \Omega_\alpha, \tag{2.1}$$

where $f_\alpha := f_\alpha(\boldsymbol{x},t)$ is the distribution function corresponding to the discrete velocity $\boldsymbol{\xi}_\alpha$, and Ω_α is the collision term to be specified latter. In the literature of LBE, the notation DdQq denotes a model with q discrete velocities in d dimensional physical space. To be concrete, in this work we will use D2Q9 model of which the discrete velocities are:

$$\boldsymbol{\xi}_\alpha = \begin{cases} (0,0), & \alpha=0, \\ (\pm1,0)c,\ (0,\pm1)c, & \alpha=1-4, \\ (\pm1,\pm1)c, & \alpha=5-8, \end{cases} \tag{2.2}$$

where $c:=\sqrt{3RT}=1$ is the lattice velocity. It is important to note that because the discrete velocity set $\{\boldsymbol{\xi}_\alpha | \alpha=0,1,\cdots,(q-1)\}$ is a constant one, thus $\boldsymbol{\xi}_\alpha \cdot \boldsymbol{\nabla} f_\alpha = \boldsymbol{\nabla}\cdot(\boldsymbol{\xi}_\alpha f_\alpha)$. Eq. (2.1) can be written more concisely in the following vector form:

$$\partial_t \mathbf{f} + \boldsymbol{\nabla} \cdot \mathbf{J} = \boldsymbol{\Omega}, \tag{2.3}$$

where symbols in bold and bold sans serif fonts denotes a vector in q dimensions and a tensor, and specifically for the D2Q9 model,

$$\mathbf{f}=\begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_8 \end{pmatrix}, \quad \boldsymbol{\Omega}=\begin{pmatrix} \Omega_0 \\ \Omega_1 \\ \vdots \\ \Omega_8 \end{pmatrix}, \quad \mathsf{J}=\begin{pmatrix} \xi_0 f_0 \\ \xi_1 f_1 \\ \vdots \\ \xi_8 f_8 \end{pmatrix}. \tag{2.4}$$

We will use the collision model with multiple relaxation times (MRT) due to d'Humiéres [21], which can be written as

$$\boldsymbol{\Omega}=-\mathsf{M}^{-1}\cdot\mathsf{S}\cdot\left[\mathbf{m}-\mathbf{m}^{(0)}\right], \tag{2.5}$$

where $\mathbf{m}$ and $\mathbf{m}^{(0)}$ are the vectors of the moments and their equilibria, respectively, M is the $q\times q$ transformation matrix which maps the distributions $\mathbf{f}$ the to the moments $\mathbf{m}$, *i.e.*,

$$\mathbf{m}=\mathsf{M}\cdot\mathbf{f}, \quad \mathbf{m}=\mathsf{M}^{-1}\cdot\mathbf{m}, \tag{2.6}$$

and S is the $q\times q$ diagonal matrix of relaxation rates $\{s_\alpha\}$.

There are nine moments in the D2Q9 model:

$$\mathbf{m}=(\rho,e,\varepsilon,j_x,q_x,j_y,q_y,p_{xx},p_{xy})^\dagger, \tag{2.7}$$

where † denotes transpose; ρ is zeroth-order moment of the mass density of the flow, e is the second-order moment related to energy, ε is fourth-order moment related to energy square, $j_x:=\rho u_x$ and $j_y:=\rho u_y$ are the first-order moments corresponding to the x and y components of the flow momentum, respectively, q_x and q_y are the third-order moments related to the x and y components of the energy fluxes, respectively, and p_{xx} and p_{xy} are the second-order moments related to the stresses. The equilibrium moments are

$$m_1^{(0)}:=e^{(0)}=-\rho(2-3u^2), \tag{2.8a}$$
$$m_2^{(0)}:=\varepsilon^{(0)}=\rho(1-3u^2), \tag{2.8b}$$
$$m_{4,6}^{(0)}:=q_{x,y}^{(0)}=-\rho u_{x,y}, \tag{2.8c}$$
$$m_7^{(0)}:=p_{xx}^{(0)}=\rho(u_x^2-u_y^2), \tag{2.8d}$$
$$m_8^{(0)}:=p_{xy}^{(0)}=\rho u_x u_y. \tag{2.8e}$$

Note that the equilibria of the conserved moments, *i.e.*, ρ and $\boldsymbol{j}:=\rho(u_x,u_y):=(j_x,j_y)$, are themselves, and the LBE model used here is athermal, thus the thermal energy, which is related to the moment $m_1\equiv e$, is not a conserved quantity.

According to the (arbitrary) ordering of moments stipulated in Eq. (2.7), the transformation matrix M is:

$$\mathsf{M}=\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix}. \tag{2.9}$$

It is obvious that $\mathsf{M}^{-1}\propto\mathsf{M}^{\dagger}$, because M is orthogonal. The diagonal matrix of relaxation rates, S, is

$$\mathsf{S}=\mathrm{diag}(0,s_e,s_\varepsilon,0,s_q,0,s_q,s_\nu,s_\nu), \tag{2.10}$$

where the zeros correspond to the conserved modes, and the unit of s_α is δt.

With the equilibria given by Eqs. (2.8), the speed of sound in the system is

$$c_s=\frac{1}{\sqrt{3}}c, \tag{2.11}$$

and the shear viscosity ν and the bulk viscosity ζ are

$$\nu=\frac{1}{3s_\nu}c^2, \quad \zeta=\frac{1}{3s_e}c^2. \tag{2.12}$$

Note that because both space and time are continuous in Eq. (2.1), thus the viscosities are proportional to $1/s_k$, and not to $(1/s_k-1/2)$. Also, the unit of the relaxation rates, $\{s_\alpha\}$, is inverse of time. Because the system obeys the equation of state for the ideal gas, thus the pressure is $p=\rho {c_s}^2$.

The (conserved) macroscopic variables can be obtained from the distributions $\{f_\alpha\}$ as the following:

$$\mathsf{Q}:=\begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \end{pmatrix}=\mathsf{C}\cdot\mathbf{f}, \tag{2.13}$$

where C projects the distributions to the conserved moments:

$$\begin{aligned}\mathsf{C}:=&\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \xi_{0x} & \xi_{1x} & \cdots & \xi_{8x} \\ \xi_{0y} & \xi_{1y} & \cdots & \xi_{8y} \end{pmatrix} \\ =&\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{pmatrix},\end{aligned} \tag{2.14}$$

the rows in C are the first, fourth, and sixth rows in the transform matrix M. Because the conserved moments are the collisional invariants, therefore

$$\mathsf{C}\cdot\mathbf{\Omega}=\mathbf{0}. \tag{2.15}$$

2.2 The finite volume discretization of the lattice Boltzmann equation

One of the most distinctive features of the finite-volume method is unstructured mesh. To be concrete, we will restrict this work to two dimensions only. Fig. 1 illustrates two adjacent triangular cells i and j with volumes V_i and V_j, respectively. The interface between the two cells are represented by $\boldsymbol{S}_{ij}:=S_{ij}\hat{\boldsymbol{n}}_{ij}$, where S_{ij} and $\hat{\boldsymbol{n}}_{ij}$ are the surface area and the out-normal unit vector at the interface center $\boldsymbol{R}_{ij}$ with respect to the cell i. Immediately, we have $\boldsymbol{S}_{ji}=-\boldsymbol{S}_{ij}$ with respect to the cell j.

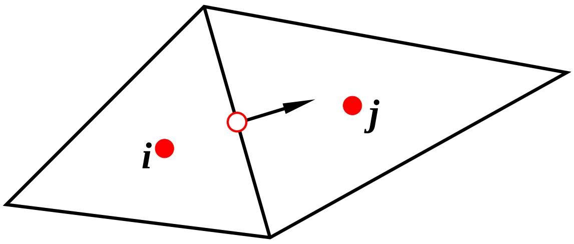

Figure 1: Illustration of two cells and the interface between them. The cell "i" and the cell "j" are on the left and the right side of their interface "ij", respectively. The cell centroids are marked with •. The arrow marks the vector $\hat{\boldsymbol{n}}_{ij}$ sitting at the interface center $\boldsymbol{R}_{ij}$ marked by ∘.

We integrate Eq. (2.3) over the cell i with cell volume V_i and boundary ∂V_i:

$$\partial_t \int_{V_i} \mathbf{f}\,\mathrm{d}V + \oint_{\partial V_i} \mathbf{F}\,\mathrm{d}S = \int_{V_i} \mathbf{\Omega}\,\mathrm{d}V, \tag{2.16}$$

where $\mathbf{F}:=\mathsf{J}\cdot\hat{\boldsymbol{n}}$ is the convective flux, J is defined in Eq. (2.4), and $\hat{\boldsymbol{n}}$ is the unit vector out normal to the surface element $\mathrm{d}S$, to obtain the following equation

$$\partial_t \mathbf{f}_i = -\frac{1}{V_i}\left[\sum_{j=1}^{N_\mathrm{S}} \mathbf{F}_{ij} S_{ij}\right] - \mathbf{\Omega}_i, \tag{2.17}$$

where the variables with subscript i and ij are the quantities averaged over either the cell volume V_i and the cell boundary $\boldsymbol{S}_{ij}$, respectively; and N_S is the total number of the discretized boundary surface elements of the cell i.

To determine the fluxes at a cell interfaces $\boldsymbol{S}_{ij}$, we need to reconstruct the variables at both sides of the interface, denoted by $\mathbf{f}_\mathrm{L}$ and $\mathbf{f}_\mathrm{R}$, to compute the fluxes on both sides of the interface. Before calculating $\mathbf{f}_\mathrm{L}$ and $\mathbf{f}_\mathrm{R}$ with high order (≥ 2), we need to obtain the space derivatives of the distribution functions at the cell center. In this work, to get second order reconstruction, the piecewise linear approximation is adopted in a cell and the inverse distance weighted least-square method is used to compute the gradients of the distribution functions at the cell center. In this method, the derivatives of $\mathbf{f}_i$ are obtained through the following minimization procedure:

$$\min_{\boldsymbol{\nabla}\mathbf{f}_i}\left(\sum_{j=1}^{N_S}\frac{1}{r_{ij}}\left\|\left(\mathbf{f}_j-\mathbf{f}_i\right)-\boldsymbol{r}_{ij}\cdot\boldsymbol{\nabla}\mathbf{f}_i\right\|^2\right), \tag{2.18}$$

where $\boldsymbol{r}_{ij}:=\boldsymbol{r}_j-\boldsymbol{r}_i$, and $r_{ij}=\|\boldsymbol{r}_{ij}\|=\sqrt{(x_j-x_i)^2+(y_j-y_i)^2}$ is the distance between centers of the cell i and cell j. The result of the above minimization procedure, Eq. (2.18), leads to the following equation for $\boldsymbol{\nabla}\mathbf{f}_i$:

$$\mathsf{A}\cdot\boldsymbol{\nabla}\mathbf{f}_i^{\dagger}=\begin{pmatrix}\mathbf{B}_{ix}^{\dagger}\\ \mathbf{B}_{iy}^{\dagger}\end{pmatrix}, \tag{2.19}$$

where A is a $d\times d$ matrix, $\mathbf{B}_{i\alpha}$, $\alpha\in\{x,y\}$, is a q dimensional vector:

$$A_{\alpha\beta}:=\sum_{j=1}^{N_S}\frac{r_{ij\alpha}r_{ij\beta}}{r_{ij}},\quad \mathbf{B}_{i\alpha}:=\sum_{j=1}^{N_S}\frac{r_{ij\alpha}}{r_{ij}}\left(\mathbf{f}_j-\mathbf{f}_i\right), \tag{2.20}$$

and $r_{ij\alpha}$ is the α-th component of $\boldsymbol{r}_{ij}$ and α, $\beta\in\{1,2\}$, *i.e.*, $\boldsymbol{r}_{ij}=(x_{ij},y_{ij})$, $r_{ij1}=x_{ij}$ and $r_{ij2}=y_{ij}$.

Clearly, the elements of the matrix A are determined purely by the coordinates of their centroids, thus they only need to be computed once and to be stored for later use. As for the elements of $\mathbf{B}_{i\alpha}$, $\alpha\in\{x,y\}$, can be computed with the edge-based data structure to enhance computational efficient. After obtaining the gradients $\boldsymbol{\nabla}\mathbf{f}_i$ at the cell i from Eq. (2.18), we can reconstruct $\mathbf{f}_{\rm L}$ and $\mathbf{f}_{\rm R}$. Suppose that the cell i and the cell j are at the left and right to the cell boundary $\boldsymbol{S}_{ij}=S_{ij}\hat{\boldsymbol{n}}_{ij}$ with its center position $\boldsymbol{R}_{ij}:=(x_{ij},y_{ij})$, as illustrated in Fig. 1, then we have:

$$\mathbf{f}_{\rm L}=\mathbf{f}_i+\left(\boldsymbol{R}_{ij}-\boldsymbol{r}_i\right)\cdot\boldsymbol{\nabla}\mathbf{f}_i, \tag{2.21a}$$

$$\mathbf{f}_{\rm R}=\mathbf{f}_j+\left(\boldsymbol{R}_{ji}-\boldsymbol{r}_j\right)\cdot\boldsymbol{\nabla}\mathbf{f}_j, \tag{2.21b}$$

where $\boldsymbol{R}_{ij}=\boldsymbol{R}_{ji}$ is the position vector of the center of the cell boundary $\boldsymbol{S}_{ij}=-\boldsymbol{S}_{ji}$. It should be noted that limiter is not needed in the reconstruction for incompressible flows, of which flow fields are continuous.

With $\mathbf{f}_{\rm L}$ and $\mathbf{f}_{\rm R}$ available, the fluxes at cell interfaces can be computed by the low-diffusion Roe scheme [20], which is given as:

$$\mathbf{F}_{ij}=\frac{1}{2}\left[\mathbf{F}(\mathbf{f}_{\rm L})+\mathbf{F}(\mathbf{f}_{\rm R})-U|\boldsymbol{\xi}_\alpha\cdot\hat{\boldsymbol{n}}_{ij}|_{\max}(\mathbf{f}_{\rm R}-\mathbf{f}_{\rm L})\right], \tag{2.22}$$

where $\boldsymbol{\xi}_\alpha$ is discrete particle velocity, $\mathbf{F}(\mathbf{f}_{\rm L})$ and $\mathbf{F}(\mathbf{f}_{\rm R})$ are flux functions on the left and right sides of cell interface, respectively. $\hat{\boldsymbol{n}}_{ij}$ is the out-normal unit vector with respect to the cell i, and the local characteristic velocity U is defined by

$$U=\max[\min(k\|\boldsymbol{u}\|,1.0),\nu/\Delta x,\epsilon], \tag{2.23}$$

where ϵ is a cut-off parameter and is $1.0\cdot 10^{-5}$ in this work, and k is an adjustable parameter and $k=1.0$ is used in what follows unless otherwise stated.

Generally, one can define the integration of fluxes as

$$\mathbf{R}_i := \sum_{j=1}^{N_{\mathrm{F}}} \mathbf{F}_{ij} S_{ij}. \tag{2.24}$$

With the edge-based data structured [18], the flux integration can be efficiently calculated by two steps as:

1. Initialization of all convective flux integration by zero:

```
Do $i=1$, all cells
    $\mathbf{R}_i=\mathbf{0}$
Enddo
```

2. Looping all surfaces to get convective flux integration:

```
Do $I_{\mathrm{face}}=1$, all interfaces
    $i=$ the cell number for the cells left to the interfaces
    $j=$ the cell number for the cells right to the interfaces
        $\mathbf{R}_i=\mathbf{R}_i+\mathbf{F}_{ij}S_{ij}$
        $\mathbf{R}_j=\mathbf{R}_j-\mathbf{F}_{ij}S_{ij}$
Enddo
```

Note that $\mathbf{F}_{ij}S_{ij}$ is computed only once per time step in the calculations. Obviously the algorithm described above requires no geometric information of a cell, *i.e.*, the shape of a cell, which can be triangular or quadrilateral in two dimensions, thus the present method is grid-transparent and can be easily implemented on arbitrary unstructured meshes.

2.3 The time marching method

To enhance the stability and accelerate the convergence, the collision term is treated implicitly in the proposed scheme. Specifically, integrated with time from t_n to t_{n+1} and with explicit convection term, the Eq. (2.17) becomes

$$\mathbf{f}_i^{n+1}-\mathbf{f}_i^{n} = -\frac{\Delta t}{V_i}\mathbf{R}_i^n + \int_{t_n}^{t_{n+1}} \mathbf{\Omega}_i \mathrm{d}t, \tag{2.25}$$

where $\Delta t := t_{n+1}-t_n$ is the time step size and the superscript n and $n+1$ denote the variables at times t_n and t_{n+1}, respectively. Applying the trapezoidal rule to the integration term of $\mathbf{\Omega}_i$ in the above equation yields:

$$\mathbf{f}_i^{n+1} = \mathbf{f}_i^{n} - \frac{\Delta t}{V_i}\mathbf{R}_i^n + \frac{\Delta t}{2}\left(\mathbf{\Omega}_i^n + \mathbf{\Omega}_i^{n+1}\right). \tag{2.26}$$

Because of $\mathbf{\Omega}_i^{n+1}$, the collision term in the proposed scheme is therefore implicit. The proposed scheme does not need to linearize the collision term $\mathbf{\Omega}_i$, which requires calculations of the Jacobian, because of the conservation of the mass and momentum can be realized by:

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{V_i} \mathsf{C} \cdot \mathbf{R}_i^n, \tag{2.27}$$

where C is the matrix of the collisional invariants defined in Eq. (2.13). After obtaining the (conserved) macro-variables at the time t_{n+1}, we have the equilibrium moments $[\mathbf{m}^{(0)}]_i^{n+1}$ by the definition of $\mathbf{m}^{(0)}$, Eqs. (2.8), so, from Eq. (2.26), the distribution function $\mathbf{f}_i^{n+1}$ can be computed:

$$\begin{aligned} \mathbf{f}_i^{n+1} &= \left(\mathsf{I} - \frac{\Delta t}{2} \tilde{\mathsf{S}} \right)^{-1} \cdot \left[\left(\mathsf{I} + \frac{\Delta t}{2} \tilde{\mathsf{S}} \right) \cdot \mathbf{f}_i^n - \frac{\Delta t}{V_i} \mathbf{R}_i^n - \Delta t \mathsf{M}^{-1} \cdot \mathsf{S} \cdot [\mathbf{m}^{(0)}]_i^{n+1/2} \right] \\ &= \left(\mathsf{I} - \frac{\Delta t}{2} \tilde{\mathsf{S}} \right)^{-1} \cdot \left[\left(\mathsf{I} + \frac{\triangle t}{2} \tilde{\mathsf{S}} \right) \cdot \mathbf{f}_i^n - \frac{\Delta t}{V_i} \mathbf{R}_i^n - \Delta t \tilde{\mathsf{S}} \cdot [\mathbf{f}^{(0)}]_i^{n+1/2} \right], \end{aligned} \tag{2.28}$$

where I is the $q \times q$ identity matrix, $\tilde{\mathsf{S}} := \mathsf{M}^{-1} \cdot \mathsf{S} \cdot \mathsf{M}$, and

$$[\mathbf{m}^{(0)}]_i^{n+1/2} := \frac{1}{2} \left([\mathbf{m}^{(0)}]_i^{n+1} + [\mathbf{m}^{(0)}]_i^n \right), \tag{2.29a}$$

$$[\mathbf{f}^{(0)}]_i^{n+1/2} := \mathsf{M}^{-1} \cdot [\mathbf{m}^{(0)}]_i^{n+1/2}. \tag{2.29b}$$

For steady flows, the local time stepping method is often used to accelerate convergence. The local time-step in cell i is defined as [19]:

$$\Delta t_i = \sigma \frac{V_i}{\Lambda_i}, \tag{2.30}$$

where Λ_i and σ denote the convective spectral radius and the CFL number, respectively. In the case of cell-centered FVM, the convective spectral radius is defined by [19]:

$$\Lambda_i = \max_{\alpha \in \{0,1,\cdots,8\}} \left(\textstyle\sum_{j=1}^{N_F} |\boldsymbol{\xi}_\alpha \cdot \hat{\boldsymbol{n}}_{ij}| \Delta S_{ij} \right). \tag{2.31}$$

2.4 Boundary conditions

We will use the ghost cell method [19] to realize both the Dirichlet and the Neumann boundary conditions for the macro-variables ρ and $\boldsymbol{u}$. Fig. 2 illustrates the ghost cell method. The cell "`abc`" with its centroid located at point i is next to the boundary — its edge "`ab`" is a part of the boundary. The cell "`abd`" with centroid located at point i' is the reflective image of the cell "`abc`" about the boundary segment "`ab`". The point W_{ab} is the intersection between edge ii' and edge "`ab`".

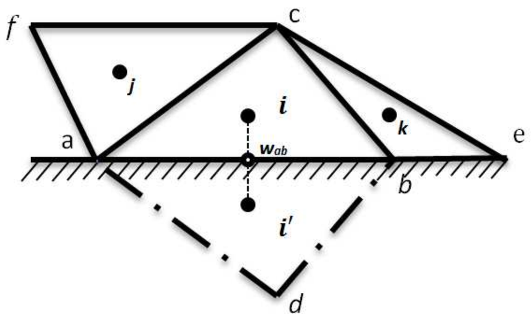

Figure 2: Illustration of the ghost cell method. The boundary cell "abc" and its ghost cell abd with their centroids located at point i and point i', respectively.

Based on the ghost cell method, we can obtain the macro-variables at the cell boundary "ab". For instance, we assume the boundary ab is the wall, on which the velocity at W_{ab} should satisfy the no-slip boundary conditions and the normal gradient of pressure, *i.e.*, $\partial_{\hat{\boldsymbol{n}}} p := \hat{\boldsymbol{n}} \cdot \boldsymbol{\nabla} p = 0$. Therefore, with the central-difference approximation, the macro-variables at i' and W_{ab} can be obtained. Consequently, the equilibrium moments $\mathbf{m}^{(0)}(W_{\mathrm{ab}})$ at the point W_{ab} can also be calculated. To compute the moments $\mathbf{m}(W_{\mathrm{ab}})$ at W_{ab}, we write $\mathbf{m}(W_{\mathrm{ab}})$ in two parts, their equilibria and non-equilibrium part:

$$\mathbf{m} = \mathbf{m}^{(0)} + \delta\mathbf{m}, \tag{2.32}$$

where $\delta\mathbf{m}$ denotes the non-equilibrium part. The non-equilibrium moments can be obtained by exploration [22]. From this scheme, the non-equilibrium part of the moments at i is taken as a good approximation of the counterpart at W_{ab}. Namely, the moments at W_{ab} are given by:

$$\mathbf{m}_i \approx \mathbf{m}^{(0)}(W_{\mathrm{ab}}) + \left(\mathbf{m}_i - \mathbf{m}_i^{(0)}\right). \tag{2.33}$$

Finally, the moments st i' are obtained by the central difference scheme:

$$\mathbf{m}_{i'} = 2\mathbf{m}(W_{\mathrm{ab}}) - \mathbf{m}_i, \tag{2.34}$$

and then, the distribution functions at i' should be given as:

$$\mathbf{f}_{i'} = \mathsf{M}^{-1} \cdot \mathbf{m}_{i'}. \tag{2.35}$$

3 Results and discussions

In this Section, several well-known cases for incompressible flow in two-dimensions are used to validate the proposed finite volume lattice Boltzmann method. Unless otherwise stated, we will fix the CFL number $\sigma = 0.1$, and set $s_e = s_\epsilon = s_\nu$ and $s_q = 1.3 s_\nu$ in what follows.

It should be noted that for the FV-LBE, there is no special (magic) relationship between s_q and s_ν [23–25]. So long as the scheme is stable, the relaxation rate s_q does not have a strong influence on the FV-LB scheme as it does on the original LB scheme [25–27]. In fact, we have tested the values of s_q in the range of $0.5s_\nu \leq s_q < 5.0s_\nu$, and found that the FV-LBE is rather insensitive to the value of s_q, provided that s_q is not too close to zero or much larger than s_ν. This fact may be closely tied to the specific implementation of the boundary conditions. The value of $s_q = 1.3s_\nu$ is chosen empirically based on numerical tests. A thorough analysis of the effect of s_q in the proposed FV-LBE is deferred to future investigation.

3.1 Poiseuille flow

The first case we consider is the incompressible Poiseuille flow between two parallel no-slip walls driven by a constant body force, and with the periodic boundary conditions along the streamwise direction, as illustrated in Fig. 3. This flow can be solved analytically; the streamwise velocity u_x is given by

$$u_x(y) = U_0\left(1 - \frac{4y^2}{H^2}\right), \quad -\frac{H}{2} \leq y \leq +\frac{H}{2}, \tag{3.1}$$

where $U_0 := gH^2/8\nu$ is the maximal velocity along the channel center line, H is the channel width, and g is the constant acceleration due to external force. We will set $H = 10.0$, $U_0 = 0.1$, and $g = 0.004$. Since the analytic solution of velocity profile $u_x(y)$ is a parabola, it is often used to verify the accuracy of a numerical scheme.

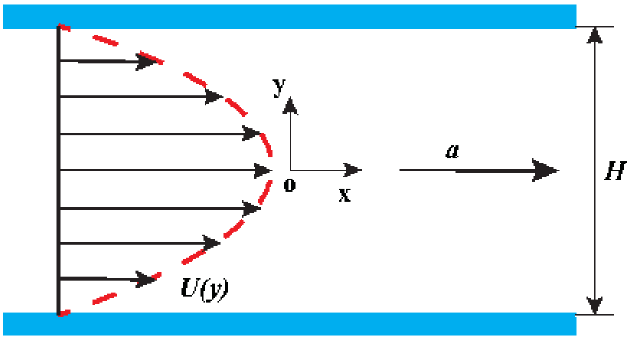

Figure 3: Sketch of Poiseuille flow driven by a constant body force between two parallel plates in 2D.

We use uniform meshes of the size $N_x \times N_y$, with $N_x = 1$, and $N_x = 10$, 20, 40, and 80. Of course, we need one additional layer in x direction as the buffer layer for the periodic boundary conditions along the streamwise direction. Fig. 4 shows the analytic results of $u_x(y)$ given by Eq. (3.1) and the numerical result with $N_y = 80$. Table 1 shows the L_2-normed errors of the velocity $u_x(y)$ and the order of *convergence n* obtained by the Richardson extrapolation.

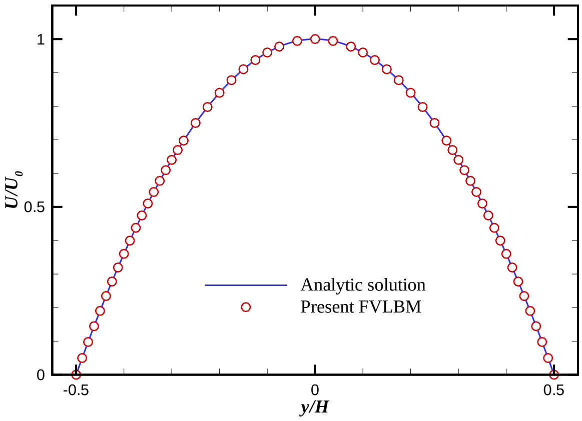

Figure 4: Poiseuille flow. Comparison of the numerical result (circles) of the velocity $u_x(y)$ obtained by the FV-LBM with $N_y = 90$ with the analytic formula of Eq. (3.1) (solid line).

The designed order of *convergence* for the proposed FV-LBM is 2, and it is indeed verified by the results given in Table 1. It should be pointed out that the traditional LBE with the MRT collision model can yield *exact* result for the Poiseuille flow. In addition, the stress is also second-order accurate in the traditional LBE. But, clearly, the necessary interpolations and extrapolations in the proposed FV-LBE method affect its accuracy. The symmetry of the discrete velocity set and a uniform Cartesian mesh lead cancellation of some errors in the LBE. The FV-LBE no longer has this advantage because of its unstructured mesh.

Table 1: Poiseuille flow. The N_y-dependence of the L_2-normed error of the velocity $u_x(y)$ and the order of convergence n obtained by the Richardson extrapolation.

N_y	L_2-normed error	n
10	2.51×10^{-3}	–
20	5.76×10^{-4}	2.13
40	1.38×10^{-4}	2.05
80	3.855×10^{-5}	1.84

3.2 Blasius boundary layer over a semi-infinite flat plate

The second case we consider is the Blasius boundary layer over a semi-infinite flat plate (cf. [28,29]). The self-similar solution for the Blasius boundary layer is well known — the thickness of the boundary layer grows as

$$\delta(x) := \frac{x}{\sqrt{\mathrm{Re}(x)}}, \quad \mathrm{Re}(x) := \frac{U_\infty x}{\nu}, \tag{3.2}$$

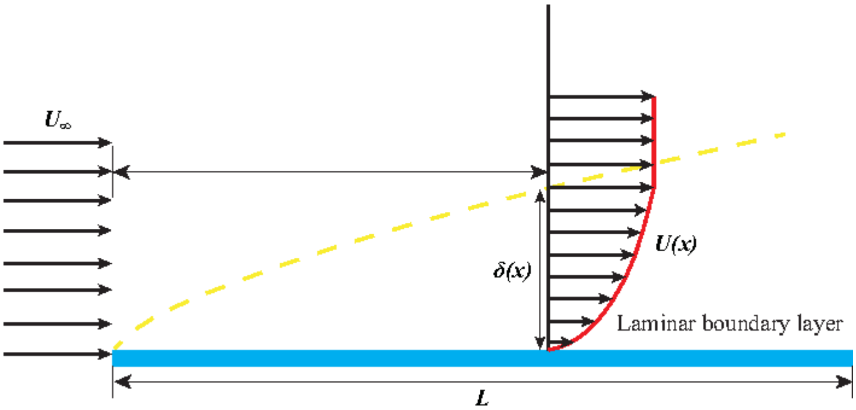

Figure 5: An illustration of the Blasius laminar boundary layer over a semi-infinite plate.

where U_∞ is constant speed of the free-flow, and x is the distance along the plate, as illustrated in Fig. 5. The streamwise velocity u_x is a function of

$$\eta := \frac{y}{\delta(x)}. \tag{3.3}$$

The streamwise velocity $u_x(\eta)$ grows very quickly from zero at the plate ($\eta = 0$) to $0.99U_\infty$ at $\eta \approx 5$. Clearly, the flow is highly anisotropic, *i.e.*, $u_x \gg u_y$ in downstream when the boundary layer has developed. In fact, $u_y(x,y)$ has to be rescaled by

$$V_\infty := \frac{U_\infty}{\sqrt{\mathrm{Re}(x)}}, \tag{3.4}$$

that is, the magnitude of the normal velocity $u_y(x,y)$ is weaker than that of the streamwise velocity $u_x(x,y)$ by about a factor of $\sqrt{\mathrm{Re}(x)}$.

The Reynolds number of the Blasius is defined as

$$\mathrm{Re} = \frac{U_\infty L}{\nu}, \tag{3.5}$$

where U_∞ is the speed of the far-field free-stream flow and L is the length of the flat plate. In our simulations, we set the Reynolds number $\mathrm{Re} = 10^5$ with $L = 100.0$ and $U_\infty = 0.1$, so the flow is nearly incompressible.

Fig. 6 sketches the flow domain used in the calculations with the boundary conditions and the mesh. The domain size is $L_x \times L_y = 122.5 \times 50$. The origin of the coordinate system is set at the tip of the flat plate, so $(x,y) \in [-22.5, 100.0] \times [0,50]$. On the left edge of the domain, the inflow boundary conditions are specified with the constant velocity $\boldsymbol{U} = (U_\infty, 0)$ and the pressure calculated by interpolation from interior of the flow field. On the right edge, the outflow boundary conditions are specified with zero gradients conditions. On the top edge, normally, the far-field boundary conditions should be given. Here, we

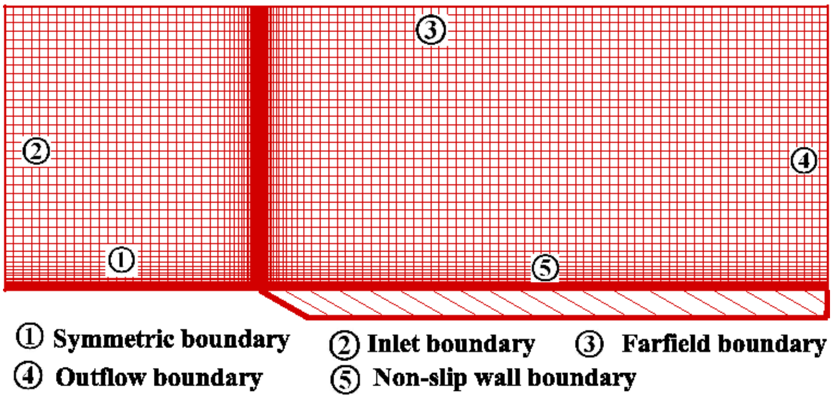

Figure 6: The mesh and boundary conditions configuration of the plate boundary layer case.

simply set it as the inflow boundary conditions as the left edge of the domain. On the bottom edge, the no-slip boundary conditions are imposed along the flat plate located at $y=0$ and $0\le x\le 100$, and the symmetric boundary conditions are given in front of the flat plate at $y=0$ and $-22.5\le x\le 0$. All the boundary conditions are realized by the ghost cell method when appropriate.

The nonuniform meshes to be used in our simulations are specified by three parameters: the minimal grid spacing h_0 at the first layer of mesh at the flat plate, the maximum grid spacing $h_{\max}$, and the the grid spacing stretching factor r. Thus, the grid spacing is stretched geometrically until it reaches the maximum $h_{\max}$, *i.e.*,

$$h_i = h_0 r^i, \quad i\ge 1, \quad \text{and} \quad h_i \le h_{\max},$$

then the grid spacing becomes a constant with the spacing of $h_{\max}$. The value of $h_{\max}$ is set to 1.2 unless otherwise stated. The fine meshes are used near the flat plate in the y direction, and around the tip of the flat plate along the x in both direction. We use five meshes with different mesh parameters h_0, $h_{\max}$, and r, which are given in Table 2.

Table 2: The height of the first layer, h_0/L, and the height growth factor of the meshes for the Blasius flow.

Mesh	h_0/L	r
Mesh-1	1.0×10^{-4}	1.15
Mesh-2	1.0×10^{-4}	1.10
Mesh-3	1.0×10^{-4}	1.05
Mesh-4	5.0×10^{-4}	1.05
Mesh-5	1.0×10^{-3}	1.05

Fig. 7 shows the mesh dependence of the numerical results of the normalized velocities $u_x(\eta)/U_\infty$ and $u_y(\eta)/V_\infty$ at $x=40.0$. With $x=40$, $\mathrm{Re}(x)=4.0\times 10^4$, thus the normal velocity $u_y(\eta)$ is about 200 times weaker than the streamwise velocity $u_x(\eta)$ at the same

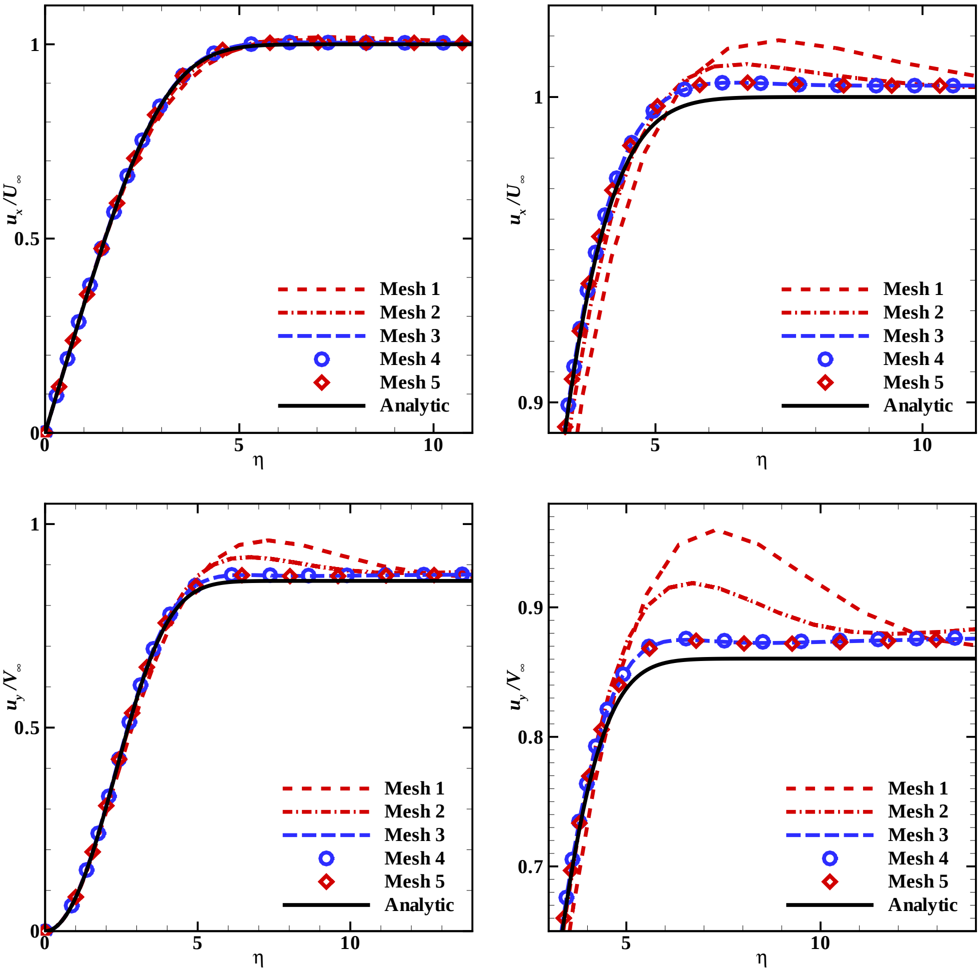

Figure 7: The mesh dependence of the numerical solutions for the Blasius boundary layer. Top row: $u_x(\eta)/U_\infty$, $0\le\eta\le 11$ (left) and $3\le\eta\le 11$ (right). Bottom row: $u_y(\eta)/V_\infty$, $0\le\eta\le 14$ (left) and $3\le\eta\le 14$ (right). $x=40.0$, $\mathrm{Re}=10^5$. The figures right column are used to magnifying differences due to different meshes.

location. Therefore, accurately calculating u_y is far more challenging than accurately calculating u_x. First, we note that all the meshes are adequate to capture the boundary layer near the plate for $\eta<2$. The slopes of both $u_x(\eta)$ and $u_y(\eta)$ at $\eta=0$ are accurately computed with all five meshes, which is important for accurate estimation of the stress at the plate. Second, both $u_x(\eta)$ and $u_y(\eta)$ are computed with about the same accuracy with the meshes, even though $u_y(\eta)$ is much weaker than $u_x(\eta)$ by a factor about 200 $(=\sqrt{\mathrm{Re}})$. This indicates that the FV-LBE scheme is in fact a second-order scheme and the meshes are adequate to resolve the boundary layer, thus the numerical dissipation is well controlled. And third, the discrepancies due to the meshes depend mainly on the stretching factor r but not much on the first grid spacing h_0, as the relative L_2 errors for

Table 3: The relative L_2 errors of $u_x(\eta)/U_\infty$ and $u_y(\eta)/V_\infty$ at $x=40.0$ for the Blasius flow.

mesh	Error in $u_x(\eta)/U_\infty$	Error in $u_y(\eta)/V_\infty$
Mesh 1	1.50372×10^{-2}	5.30702×10^{-2}
Mesh 2	7.05944×10^{-3}	3.30806×10^{-2}
Mesh 3	3.43120×10^{-3}	1.01322×10^{-2}
Mesh 4	3.42317×10^{-3}	1.08417×10^{-2}
Mesh 5	3.45917×10^{-3}	9.48680×10^{-3}

both $u_x(\eta)/U_\infty$ and $u_y(\eta)/V_\infty$ in Table 3. The stretching factor r determines the total number of grid points within the boundary layer beyond the first grid spacing. Furthermore, the stretching factor r determines the number of grid points in the vicinity near the edge of the boundary layer, where $|u_x''(\eta)|$ and $|u_y''(\eta)|$ attain their maxima. In this region, resolution is needed to accurately catch the velocity profiles.

3.3 Steady laminar flow past a circular cylinder in 2D

The third case to be considered in this work is the steady laminar flow past a circular cylinder in a two dimensional bounded domain [30]. In the previous two cases, quadrilateral structured meshes have been used. To demonstrate the mesh transparency of the proposed FV-LBE scheme, hybrid unstructured meshes combining both quadrilateral and triangular elements will be used in this case. Because of the symmetry in transverse direction of the flow, as illustrated in Fig. 8, the computational domain only uses the upper half of the flow domain. The Reynolds number for the flow is defined as:

$$\mathrm{Re}=\frac{U_\infty D}{\nu}, \tag{3.6}$$

where U_∞ is the free flow speed and $D=2R$, R is the radius of the cylinder. Only the steady flows with Re = 10, 20, and 40 are considered here.

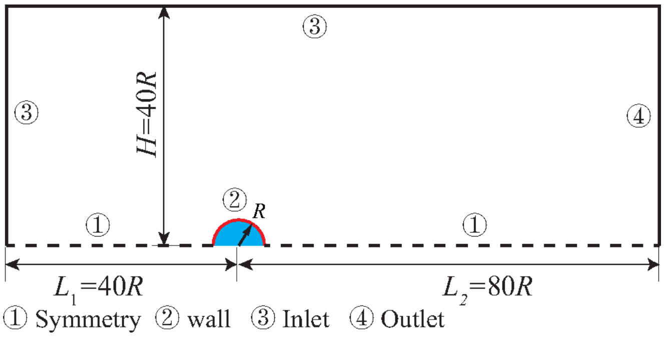

Figure 8: Schematic of the computational domain for the flow past a 2D cylinder.

The size of the computational domain is $L\times H=120R\times 40R$. On the top and left edges of the computational domain, the constant pressure p_∞ and velocity $\boldsymbol{U}=(U_\infty,0)$ are imposed through the equilibrium distribution functions. On the left and top edges, the inlet boundary conditions are used. At the bottom edge, the no-slip boundary conditions are imposed on the surface of the cylinder, and the symmetric boundary conditions are imposed on the two straight segments. All the boundary conditions are realized by the ghost cell method.

Fig. 9 depicts a hybrid mesh used in the calculation. To accurately capture the details of the flow separation near the cylinder, the mesh consists of a body-fitted structured O-mesh of quadrilateral cells, which is fine enough to resolve the boundary layer about the cylinder, and a unstructured mesh of nearly isotropic triangle cells outside the region of the boundary layer, as shown in Fig. 9. Immediately outside of the O-mesh of quadrilateral cells about the cylinder, an unstructured fine mesh of triangular cells covers the flow domain of $(5R+25R)\times 12R$, *i.e.*, $5R$ before and $25R$ behind the cylinder center in the streamwise direction x and $12R$ in spanwise direction y, excluding the area of O-mesh. This fine mesh ensures that the flow field about the rear stagnation point and the entire recirculation zone is well resolved. There are N_θ grid points uniformly placed along the circumference of the semi-circle, *i.e.*, the angular resolution around the cylinder is $\delta\theta=\pi/N_\theta$. We use a number of meshes with different values of $\delta\theta$, the grid size of the first layer along the radial direction, h_0, and the grid stretching factor, r, for the body-fitted mesh around the cylinder in our calculations. The values of $\delta\theta$, h_0 and r for the meshes are given in Table 4. Mesh-1, Mesh-2, and Mesh-3 vary the value of stretching factor, r, whereas Mesh-1 and Mesh-4 vary the size of the first grid, h_0, and Mesh-1, Mesh-5 and Mesh-6 vary the angular resolution $\delta\theta$. These meshes allow us to conduct a comprehen-

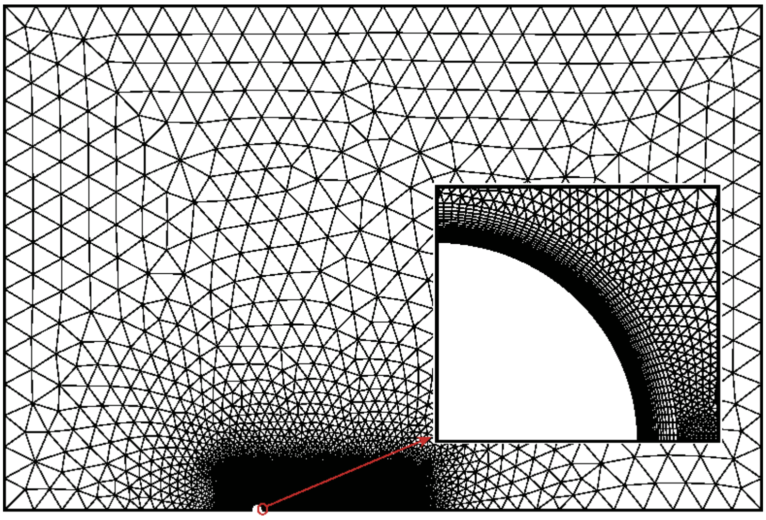

Figure 9: The configuration of the hybrid unstructured mesh for the flow past a 2D cylinder.

Table 4: The height of the first layer, h_0/D, the grid stretching factor, r, for the body-fitted meshes about the cylinder, and the total number of cells, N_{cell}.

Mesh	h_0/D	$\delta\theta$	r	N_{cell}
Mesh-1	1.0×10^{-5}	1.8°	1.10	33 382
Mesh-2	1.0×10^{-5}	1.8°	1.15	16 984
Mesh-3	1.0×10^{-5}	1.8°	1.20	9 880
Mesh-4	5.0×10^{-5}	1.8°	1.10	30 392
Mesh-5	1.0×10^{-5}	0.9°	1.10	54 174
Mesh-6	1.0×10^{-5}	0.6°	1.10	58 644

sive convergence study and to detect the effects of h_0, $\delta\theta$ and r on the interested quantities to be computed.

The quantities to be computed are the drag coefficient C_{d}, the separation length L_{s}, and the separation angle θ_{s} for the stationary symmetric recirculation eddies behind the cylinder, as illustrated in Fig. 10. Fig. 11 depicts the streamlines around the cylinder for $\text{Re}=10$, 20 and 40, obtained with Mesh-6 described in Table 4.

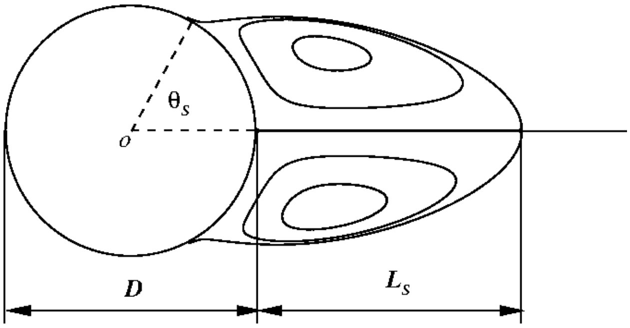

Figure 10: Schematic of the flow past a cylinder with the separation eddies of length L_{s} and angle θ_{s}.

The drag coefficient C_{d} is computed as follows. The force $\boldsymbol{F}$ on the cylinder wall due to the flow field is given by:

$$\boldsymbol{F}=\oint_{\partial\Omega}(-p\mathsf{I}+\boldsymbol{\tau})\cdot\hat{\boldsymbol{n}}\,\mathrm{d}A, \tag{3.7}$$

where p, $\boldsymbol{\tau}$ and $\hat{\boldsymbol{n}}$ denote the pressure, the shear stress tensor and the unit vector out-normal to the cylinder circumference A, respectively. In the LBE, $p=\rho/3$. The shear force tangent to the boundary surface is given by:

$$\boldsymbol{\tau}\cdot\hat{\boldsymbol{n}}=\mu\partial_{\hat{n}}[\boldsymbol{u}-(\boldsymbol{u}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}]:=\mu\partial_{\hat{n}}u_{\hat{t}}. \tag{3.8}$$

Therefore, the drag coefficient C_{d} is given by:

$$C_{\text{d}}=\frac{\hat{\boldsymbol{x}}\cdot\boldsymbol{F}}{\frac{1}{2}\rho_\infty U_\infty^2}, \tag{3.9}$$

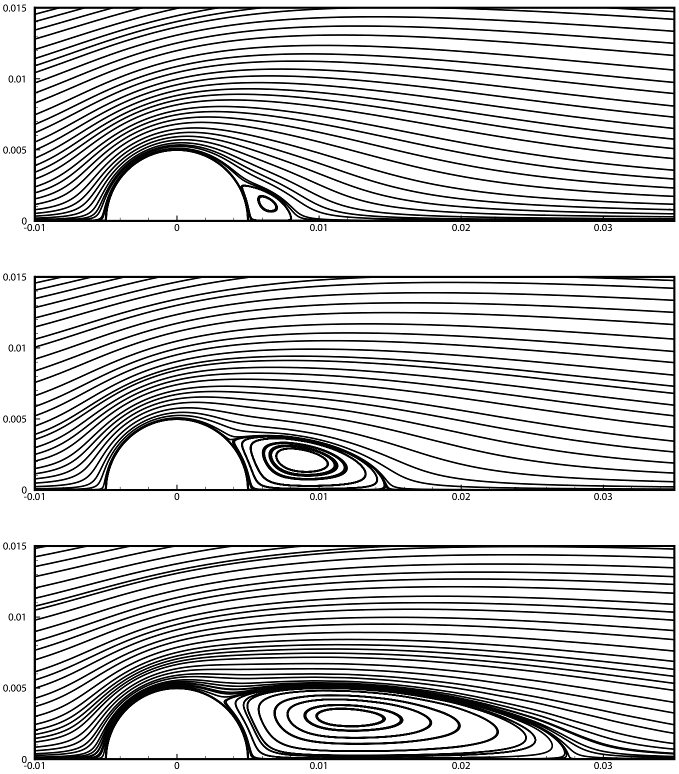

Figure 11: Streamlines of the 2D cylinder flow. From top to bottom: Re = 10, 20 and 40.

where $\hat{\boldsymbol{x}}$ is the unit vector along the x-axis, *i.e.*, the streamwise direction of the flow. Because $\mathbf{m}^{(0)}$ is known at the boundary (cf. Section 2.4, the pressure p is readily available at the boundary. The velocity field $\boldsymbol{u}$ is known at the boundary because of the boundary conditions, thus the shear force $\boldsymbol{\tau}\cdot\hat{\boldsymbol{n}}$ in Eq. (3.8) is approximated by:

$$\boldsymbol{\tau}\cdot\hat{\boldsymbol{n}} \approx \mu \frac{[\boldsymbol{u}-(\boldsymbol{u}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}](\boldsymbol{x}_i)-[\boldsymbol{u}_{\mathrm{w}}-(\boldsymbol{u}_{\mathrm{w}}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}](\boldsymbol{x}_{\mathrm{w}})}{\|\boldsymbol{x}_i-\boldsymbol{x}_{\mathrm{w}}\|}, \tag{3.10}$$

where $\boldsymbol{x}_i$ is the position of a cell centroid next to boundary (cf. Fig. 2), and $\boldsymbol{x}_{\mathrm{w}}$ is a point on the boundary such that $(\boldsymbol{x}_i-\boldsymbol{x}_{\mathrm{w}})$ is parallel to the unit vector $\hat{\boldsymbol{n}}$ out normal to the boundary. The separation length L_{s} and the separation angle θ_{s} are obtained by using the CFD visualization and analysis software `Tecplot`®.

The results for the drag coefficient C_{d}, the normalized separation length L_{s}/R, and the separation angle θ_{s} are tabulated in Table 4. Among the computed quantities, the separation length L_{s}/R appears to be most sensitive to the mesh resolution quantitatively and clearly exhibits a monotonic dependence on the mesh resolution. As the mesh becomes finer with decreasing h, L_{s}/R increases monotonically (cf. the results obtained with Mesh-3, Mesh-2, and Mesh-1 in Table 4). When the grid size of the first layer decreases, h, L_s/R also increases (cf. the results obtained with Mesh-4 and Mesh-1 in Table 4). We observe that the results obtained with two largest meshes with fines grid sizes, *i.e.*, Mesh-1 and Mesh-2, agree with each other for at least two significant digits. The results in Table 4 indicate that the results obtained with the largest mesh and finest grid size, *i.e.*, Mesh-1, may be accurate for two or three significant digits.

We now compare our results with some existing ones [31,32] with convergence studies. We note that, while the steady flow past a cylinder in 2D is a well studied case (cf., *e.g.*, [30]), mesh-independent results are rarely available in literature. The results [31,32] chosen here have been obtained with convergence check. Before we compare our results with the existing ones [31,32], some discussion on the meshes used in the previous studies is in order. In the work by He and Doolen [31], the angular resolution $\delta\theta$ is uniform along the cylinder circumference, whereas in the work of Hejranfar and Ezzatneshan [32], the angular resolution is non-uniform with $0.7668^\circ \lesssim \delta\theta \lesssim 6.744^\circ$, and the finest resolution is about the rear stagnation point of the cylinder. Another detail to be noted is that the first grid spacing in these meshes are not adequate to resolve the boundary layer with the given Reynolds number Re. To properly resolve the boundary layer, it is a common practice to use $h_0 \sim 10^{-5}D$ and to have sufficient grid points within the boundary layer. However, the values of h_0 used in these previous studies [31,32] are far less than adequate, as shown in our results given in Table 5. Also, with a constant stretching factor r, these meshes become rather coarse about the location where the separation length L_s is measured. These factors all affect the accuracy of the results. The finest meshes used in this study ensure that the boundary layer is adequately resolved, the angular resolutions are sufficiently fine, and the recirculation zone is adequately resolved, as shown by the results given in Table 5.

The results of He and Doolen [31] were obtained with the interpolation-supplemented LBE (IS-LBE) [33,34]. The more recent results of Hejranfar and Ezzatneshan [32] were obtained with a high-order compact finite-difference LBE (FD-LBE). These results [31, 32] were obtained with grid-convergence verification. The values of C_d, L_s/R, and θ_s obtained by IS-LBE [31], FD-LBE [32], and the proposed FV-LBE are tabulated in Table 6, along with the values of the parameters pertaining to the meshes used, including the finest grid size along the radial direction, h_0, the finest angular resolution, $\delta\theta$, the total number of grid points $N_{\text{cell}} = N_r \times N_\theta$ or cells used in the calculations, and the normalized diameter of the flow domain D_∞/D for the O-meshes used in the previous studies [31,32].

The differences among the results shown in Table 6 are apparent. Some dissection and discussion regarding certain critical details in the numerics of the previous studies [31,32] are in order. First of all, the flow domain in all these previous studies [31,32] are a punched disk, thus the meshes are structured body-fitted O-meshes with polar coordinates (r,θ), while the flow domain in the present work is a rectangle with a hybrid of a structured body-fitted O-mesh of quadrilateral elements about the cylinder and a unstructured mesh of triangular elements (cf. Fig. 8). There is no *non-uniform local* mesh refinement in the structured O-meshes used in the previous studies [31,32], while in the unstructured meshes used in this study, *non-uniform local* mesh refinements are used to

Table 5: Flow past a cylinder in 2D at Re = 10, 20, and 40. The mesh dependence of the drag coefficient C_d, the normalized separation length L_s/R, and the separation angle θ_s. The results with Mesh-1, which has the finest grid size and the largest number of cells, are used as the reference to compute the relative differences given in parenthesis.

Re	Mesh	C_d	(ΔC_d)	L_s/R	$(\Delta(L_\mathrm{s}/R))$	θ_s	$(\Delta\theta_\mathrm{s})$
10	Mesh-1	3.037	(1.13%)	0.6217	(6.50%)	31.26°	(1.82%)
	Mesh-2	3.036	(1.10%)	0.6075	(8.63%)	31.20°	(2.01%)
	Mesh-3	3.041	(1.27%)	0.5791	(12.9%)	31.23°	(1.92%)
	Mesh-4	3.140	(4.56%)	0.5956	(10.4%)	30.86°	(3.08%)
	Mesh-5	3.011	(0.27%)	0.6654	(0.08%)	31.82°	(0.06%)
	Mesh-6	3.003	—	0.6649	—	31.84°	—
20	Mesh-1	2.132	(0.66%)	1.9504	(4.28%)	44.43°	(0.60%)
	Mesh-2	2.128	(0.47%)	1.9005	(6.73%)	44.45°	(0.56%)
	Mesh-3	2.132	(0.66%)	1.7982	(11.8%)	44.52°	(0.40%)
	Mesh-4	2.168	(2.36%)	1.9184	(5.85%)	44.22°	(1.07%)
	Mesh-5	2.124	(0.28%)	2.0332	(0.22%)	44.60°	(0.22%)
	Mesh-6	2.118	—	2.0376	—	44.70°	—
40	Mesh-1	1.573	(0.31%)	4.5092	(4.11%)	54.37°	(0.51%)
	Mesh-2	1.567	(0.06%)	4.4960	(4.40%)	54.43°	(0.40%)
	Mesh-3	1.570	(0.13%)	4.1289	(12.2%)	54.13°	(0.95%)
	Mesh-4	1.585	(1.08%)	4.4809	(4.72%)	54.41°	(0.44%)
	Mesh-5	1.569	(0.06%)	4.7121	(0.20%)	54.55°	(0.18%)
	Mesh-6	1.568	—	4.7027	—	54.65°	—

Table 6: Flow past a cylinder in 2D with Re = 10, 20, and 40. The drag coefficient C_d, the normalized separation length L_s/R, the separation angle θ_s obtained with IS-LBE [31], FD-LBE [32], and the proposed FV-LBE. The values of the finest grid spacing in radial direction, h_0, the finest angular resolution $\delta\theta$, the number of grid nodes $N_\mathrm{cell} = N_r \times N_\theta$ for the structured O-meshes used in the previous studies [31, 32], or the number of cells used in the present study, and the flow domain size in terms of D_∞/D or $L_x/D \times L_y/D$ are also given. Note that in FV-LBE, N_cell is for one half of the flow domain.

Re	Method	C_d	L_s/R	θ_s	h_0	$\delta\theta$	N_cell	D_∞/D
10	IS-LBE [31]	3.170	0.474	26.89°	0.0378	1.5°	181×241	111.3
	FD-LBE [32]	2.810	0.4891	29.83°	0.0274	0.7668°	51×101	60.0
	FV-LBE	3.003	0.6649	31.84°	1.0×10^{-5}	0.6°	58 644	120×80
20	IS-LBE [31]	2.152	1.842	42.96°	0.0378	1.5°	181×241	111.3
	FD-LBE [32]	2.021	1.8480	43.58°	0.0274	0.7668°	51×101	60.0
	FV-LBE	2.118	2.0376	44.70°	1.0×10^{-5}	0.6°	58 644	120×80
40	IS-LBE [31]	1.499	4.490	52.84°	0.0212	1.2°	226×301	111.3
	FD-LBE [32]	1.515	4.510	51.86°	0.0274	0.7668°	51×101	60.0
	FV-LBE	1.568	4.7027	54.65°	1.0×10^{-5}	0.6°	58 644	120×80

resolve both the boundary layer around the cylinder and the recirculation zone behind the cylinder. Secondly, the finest angular resolution used in the present study is finer that the ones used in the other studies [31,32], and the total number of cells is also far greater than those in the previous studies [31,32]. And thirdly and most crucially, the sizes of the finest grid spacing, h_0, used in the recent studies [31,32] are much larger than what are used in the present work by at least three orders of magnitudes (between *ca.* 2,120 and 3,780). A rough estimate shows that the meshes used in these previous studies [31,32] cannot adequately resolve the boundary layer with the given Reynolds numbers. In addition, our meshes ensure that the entire separation region is adequately resolved with fine enough resolutions, as shown in Fig. 9, and such is not the case in the recent studies [31,32]. Our results in Table 5 clearly show that the values of C_{d}, θ_{s}, and particularly L_{s}/R, depend on the resolution of mesh, and our results do show convergence in terms of both h_0 and $\delta\theta$. In summary, the evidence indicates that the finest mesh we used can resolve the flow adequately and the results obtained with the finest mesh are accurate for at least two or three digits.

4 Conclusion

In this work, a genuine finite volume method is developed to discrete the lattice Boltzmann equation for the nearly incompressible flows with complex boundaries. The accuracy of the proposed FV-LBE is of second-order convergence. The proposed FV-LBE is grid-transparent, thus it independent of the element type and can be easily implemented on arbitrary unstructured meshes, and it naturally includes the flexibility of local grid refinement. In addition, the implicit treatment of the collision term in the proposed FV-LBE greatly enhances the numerical stability hence the efficiency. These features greatly enhance the ability of the proposed FV-LBE to treat complex geometries encountered in realistic engineering applications.

The proposed FV-LBE has been validated with three cases of steady flows in two-dimensions: (a) the Poiseuille flow driven by a constant body force, (b) the Blasius boundary layer over a flat plate, and (c) laminar flow past a cylinder in a bounded domain. The Poiseuille flow is used to verify the order of convergence of the proposal FV-LBE; and the proposed FV-LBE is indeed of second-order convergence. The Blasius boundary layer is used to demonstrate the advantages of using highly anisotropic non-uniform meshes. The proposed FV-LBE is capable to accurately capture the transverse component of the velocity field, which is difficult to achieve without adequate resolution within the boundary layer region. The flow past a cylinder in 2D is used to show the advantages of grid transparency and flexible *arbitrary* and *non-uniform* local mesh refinement implemented in the proposed FV-LBE. The meshes used in this case are hybrid ones with both quadrilateral and triangular cells, and can easily couple structured and unstructured meshes together. The tests demonstrate the potential of the proposed FV-LBE to simulate viscous flows with complex geometries.

The proposed FV-LBE can be extended to three dimensions (3D) easily and this is also an inherent advantage of kinetic methods due to the linearity of the advection term. Development of 3D FV-LBE will be considered in our future work. In addition, fully implicit and high-order techniques will be considered in the future development of the proposed FV-LBE.

Acknowledgments

The authors are grateful to Professor Manfred Krafczyk of The Institute for Computational Modeling in Civil Engineering (iRMB), Technische Universität Braunschweig, Germany, for providing computational resources with which part of this work is carried out. The authors would like to thank an anonymous referee for his/her thoughtful comments which help us improve the manuscript. The support from the Richard F. Berry Jr. Endowment at Old Dominion University is also gratefully acknowledged.

References

[1] X. He and L.-S. Luo. A priori derivation of the lattice Boltzmann equation. *Phys. Rev. E*, 55(6):R6333–R6336, 1997.
[2] X. He and L.-S. Luo. Theory of lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. *Phys. Rev. E*, 56(6):6811–6817, 1997.
[3] D.Z. Yu, R. Mei, L.-S. Luo, and W. Shyy. Viscous flow computations with the method of lattice Boltzmann equation. *Prog. Aerospace Sci.*, 39(5):329–367, 2003.
[4] L.-S. Luo, M. Krafczyk, and W. Shyy. Lattice Boltzmann method for computational fluid dynamics. In R. Blockley and W. Shyy, editors, *Encyclopedia of Arospace Engineering*, chapter 56, pages 651–660. Wiley, New York, 2010.
[5] W.-A. Yong and L.-S. Luo. Accuracy of the viscous stress in the lattice Boltzmann equation with simple boundary conditions. *Phys. Rev. E*, 86:065701R, 2012.
[6] P. J. Dellar. An interpretation and derivation of the lattice Boltzmann method using Strang splitting. *Comput. Math. Appl.*, 65:129–141, 2013.
[7] P. J. Dellar and L.-S. Luo. Lattice Boltzmann methods. In B. Engquist, editor, *Encyclopedia of Applied and Computational Mathematics*, pages 774–778. Springer, Berlin, 2015.
[8] F. Nannelli and S. Succi. The lattice Boltzmann-equation on irregular lattices. *J. Stat. Phys.*, 68(3-4):401–407, 1992.
[9] G.W. Peng, H.W. Xi, C. Duncan, and S.H. Chou. Lattice Boltzmann method on irregular meshes. *Phys. Rev. E*, 58(4):R4124–R4127, 1998.
[10] G.W. Peng, H.W. Xi, C. Duncan, and S.H. Chou. Finite volume scheme for the lattice Boltzmann method on unstructured meshes. *Phys. Rev. E*, 59(4):4675–4682, 1999.
[11] H.W. Xi, G.W. Peng, and S.H. Chou. Finite-volume lattice Boltzmann method. *Phys. Rev. E*, 59(5B):6202–6205, 1999.
[12] H.W. Xi, G.W. Peng, and S.H. Chou. Finite-volume lattice Boltzmann schemes in two and three dimensions. *Phys. Rev. E*, 60(3):3380–3388, 1999.
[13] M. Stiebler, J. Tölke, and M. Krafczyk. An upwind discretization scheme for the finite volume lattice Boltzmann method. *Comput. Fluids*, 35(8-9):814–819, 2006.

[14] F. Dubois and P. Lallemand. On lattice Boltzmann scheme, finite volumes and boundary conditions. *Prog. Comput. Fluid Dym.*, 8(1-4):11–24, 2008.

[15] S. Ubertini and S. Succi. A generalised lattice Boltzmann equation on unstructured grids. *Commun. Comput. Phys.*, 3(2):342–356, FEB 2008.

[16] D. V. Patil and K. N. Lakshmisha. Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh. *J. Comput. Phys.*, 228(14):5262–5279, 2009.

[17] S. E. Razavi, J. Ghasemi, and A. Farzadi. Flux modelling in the finite-volume lattice Boltzmann approach. *Int. J. Comput. Fluid Dyn.*, 23(1):69–77, 2009.

[18] D. J. Mavriplis. Unstructured grid techniques. *Annu. Rev. Fluid Mech.*, 29:473–514, 1997.

[19] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, New York, 2nd edition, 2005.

[20] W.D. Li, M. Kaneda, and K. Suga. A stable, low diffusion up-wind scheme for unstructured finite volume lattice Boltzmann method. In *Proceedings of The 4th Asian Symposium on Computational Heat Transfer and Fluid Flow*, 2013. June 3–6, 2013, Hong Kong, China.

[21] D. d'Humières. Generalized lattice-Boltzmann equations. In B. D. Shizgal and D. P. Weave, editors, *Rarefied Gas Dynamics: Theory and Simulations*, volume 159 of *Prog. Astronaut. Aeronaut.*, pages 450–458, Washington, D.C., 1992. AIAA.

[22] Z.L. Guo, C.G. Zheng, and B.C. Shi. An extrapolation method for boundary conditions in lattice Boltzmann method. *Phys. Fluids*, 14(6):2007–2010, 2002.

[23] I. Ginzbourg. *Boundary Conditions Problems in Lattice Gas Methods for Single and Multiple Phases*. PhD thesis, Universite Paris VI, France, 1994.

[24] I. Ginzbourg and P. M. Adler. Boundary flow condition analysis for the three-dimensional lattice Boltzmann model. *J. Phys. II*, 4(2):191–214, 1994.

[25] I. Ginzburg and D. d'Humières. Multireflection boundary conditions for lattice Boltzmann models. *Phys. Rev. E*, 68(6):066614, 2003.

[26] C. Pan, L.-S. Luo, and C.T. Miller. An evaluation of lattice Boltzmann schemes for porous medium flow simulation. *Comput. Fluids*, 35(8/9):898–909, 2006.

[27] L.-S. Luo, W. Liao, X. Chen, Y. Peng, and W. Zhang. Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations. *Phys. Rev. E*, 83(5):056710, 2011.

[28] H. Schlichting and K. Gersten. *Boundary Layer Theory*. Springer, Berlin, 8th revised and enlarged edition, 2000.

[29] F. M. White. *Viscous Fluid Flow*. McGraw-Hill Higher Education, New York, 3rd edition, 2006.

[30] M. M. Zdravkovich. *Flow Around Circular Cylinders: Volume I: Fundamentals*. Oxford University Press, Oxford, 1997.

[31] X. He and G. D. Doolen. Lattice Boltzmann method on curvilinear coordinates systems: Flow around a circular cylinder. *J. Comput. Phys.*, 134:306–315, 1997.

[32] K. Hejranfar and E. Ezzatneshan. Implementation of a high-order compact finite-difference lattice Boltzmann method in generalized curvilinear coordinates. *J. Comput. Phys.*, 267:28–49, 2014.

[33] X. He, L.-S. Luo, and M. Dembo. Some progress in lattice Boltzmann method: Part I. Nonuniform mesh grids. *J. Comput. Phys.*, 129(2):357–363, 1996.

[34] X. He, L.-S. Luo, and M. Dembo. Some progress in lattice Boltzmann method: Enhancement of Reynolds number in simulations. *Physica A*, 239(1–3):276–285, 1997.