

## Highlights

### **A flux bounce-back scheme for the filtered Spectral Element Lattice Boltzmann Method**

Chunheng Zhao, Saumil S. Patel, Hai Lu Lin, Misun Min, Taehun Lee

- Spectral Element method
- Lattice Boltzmann method
- Flux bounce-back scheme
- Turbulent pipe flow

# A flux bounce-back scheme for the filtered Spectral Element Lattice Boltzmann Method

Chunheng Zhao<sup>a</sup>, Saumil S. Patel<sup>b</sup>, Hai Lu Lin<sup>a</sup>, Misun Min<sup>c</sup>, Taehun Lee<sup>a</sup>

<sup>a</sup>*Department of Mechanical Engineering, City College of New York, New York, NY 10031, USA*

<sup>b</sup>*Computational Science Division, Argonne National Laboratory, Lemont, IL 60439, USA*

<sup>c</sup>*Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, USA*

---

## Abstract

We develop a spectral element lattice Boltzmann method (SELBM) with the flux bounce-back (FBB) scheme, to enable accurate simulations of single-phase fluid dynamics in unstructured mesh. We adopt an Eulerian description of the streaming process in place of the perfect shift in the regular LBM. The spectral element method is used to spatially discretize the convective term, while the strong stability-preserving Runge-Kutta (SSPRK) method is used for time integration. To increase stability, we investigate the use of an explicit filter, particularly in the context of the sensitive double shear layer problem. The results indicate that by using the high-order polynomial, we can effectively eliminate the small vortices around the neck region. We introduce the flux bounce-back scheme to enable the current scheme to handle complex boundaries. The proposed scheme and flux boundary method are validated through benchmark simulations, including the unsteady Couette flow and the planar Poiseuille flow. Further validation is provided through the Taylor-Green vortex problem, demonstrating the accuracy and convergence of the scheme for isotropic turbulence. Finally, we consider a fully developed turbulent flow within a cylindrical pipe and correctly predict the turbulent boundary layer profile.

*Keywords:* spectral element method, lattice Boltzmann method, unstructured mesh, flux bounce-back scheme

*PACS:* 0000, 111 1

*2000 MSC:* 0000, 1111

---

## 1. Introduction

The Lattice Boltzmann Method (LBM) is widely used as a numerical solver for fluid dynamics in single- and multi-phase flows [1]. Instead of directly solving the macroscopic governing equations (i.e. Navier-Stokes equations), LBM solves the evolution of the particle distribution function and recovers density and momentum through moments of the particle distribution functions [2]. It is computationally efficient due to a local collision function and a perfect shift streaming function [3, 4]. These features enable massive parallelization, making LBM particularly well suited for modern high-performance computing architectures [5]. The perfect-shift approach to the streaming step in the LBM ensures that the distribution function propagates along the characteristic direction with a constant speed, effectively eliminating numerical errors in the advection step. As a result, the primary source of error in LBM arises from the collision step, which maintains second-order spatial accuracy. Through Chapman-Enskog analysis, the Navier-Stokes equations can be recovered and directly derived from LBM [6, 7, 8]. Although the LBM provides a relatively straightforward alternative to directly solving the Navier-Stokes equations, several critical aspects need careful consideration. Typically, LBM is implemented on a structured mesh [9], or lattice grid, where the spatial interval  $\Delta x$  and the time step  $\Delta t$  are set to one lattice unit, and the Courant-Friedrichs-Lowy (CFL) number is fixed at 1, implying a uniform grid-spacing. In certain scenarios, unstructured meshes are indispensable. For example, boundary layer meshes are required to accurately capture the turbulent boundary layer profile [10]. Using a regular LBM scheme in such cases would require excessive grid refinement, leading to a significant waste of computational resources. Other examples arise in simulations involving complex geometries, such as fluid flow in porous media or pebble bed reactors [11, 12].

The LBM on unstructured meshes can be broadly categorized into finite volume (FV) LBM [13, 14, 15, 16] and finite element (FE) LBM [9, 17, 18, 19]. In unstructured meshes, wall normals generally vary across adjacent boundary surfaces, making it difficult to clearly define the incoming and outgoing directions. As a result, identifying the correct bounce-back directions for boundary conditions in unstructured meshes becomes particularly challenging [20, 21, 22]. Moreover, the perfect shift in the streaming used on structured meshes cannot be directly applied to unstructured meshes, which may introduce additional stability and accuracy issues. In finite volume

LBM, modified bounce-back schemes [16] and distance-based bounce-back approaches [23] have been proposed to address complex boundaries. However, most finite volume formulations directly discretize the discrete Boltzmann equation [24], which may induce stability concerns at high Reynolds numbers. Finite element LBM provides a more natural framework for constructing high-order streaming operators, such as those based on spectral elements. For the discontinuous Galerkin (DG) LBM, flux-based boundary conditions are applied at the boundary element interfaces [17]. For the continuous Galerkin LBM [9], flux boundary conditions combining with a bounce-back scheme can be imposed. However, further investigation is needed to provide a physical interpretation for the Lax-Wendroff-type boundaries [9, 18], which involve high-order streamline diffusion. The DG-LBM has demonstrated the ability to produce stable solutions using the flux bounce-back (FBB) scheme. However, this approach comes with a trade-off: the DG method inherently introduces additional dissipation at the element interface to enhance stability at high Reynolds numbers. Building on previous research, we have developed a spectral element Lattice Boltzmann method based on a continuous Galerkin formulation. Unlike the Lax-Wendroff streaming method, our approach employs a Strong Stability-Preserving (SSP) Runge-Kutta scheme to minimize numerical dissipation. Additionally, a non-dissipative filter is applied to the distribution functions, effectively preserving both density and momentum conservation as well as maintaining numerical stability.

The following sections present the proposed scheme and its validation through benchmark simulations. Section 2 describes the methodology and formulation of the scheme. Section 3 provides validation results through a series of benchmark tests, including the stability test, double shear layer, boundary condition verification, unsteady Couette flow, and planar Poiseuille flow. Additional validations are performed for turbulent flows using the Taylor Green vortex and the Poiseuille pipe flow.

## 2. Methodology

### 2.1. Lattice Boltzmann method

We start with the discrete Boltzmann equation of the density distribution function  $f_\alpha$  for the  $\alpha$ -direction, with an external forcing term  $F_\alpha$ :

$$\left( \frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla \right) f_\alpha = -\frac{1}{\lambda} (f_\alpha - f_\alpha^{eq}) + F_\alpha, \quad (1)$$

where  $f_\alpha^{eq}$  denotes the equilibrium distribution function given as:

$$f_\alpha^{eq} = t_\alpha \rho \left( 1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c_s^4} - \frac{|\mathbf{u}|^2}{2c_s^2} \right). \quad (2)$$

In Equations (1) and (2), the weights  $t_\alpha$  are associated with the equilibrium distribution function. The vector  $\mathbf{e}_\alpha$  represents the discrete velocity vector for  $\alpha$ -direction. The parameter  $\lambda$  represents the relaxation time, which is connected to the kinematic viscosity  $\nu$  through the equation  $\nu = \lambda c_s^2$ , where  $c_s$  denotes the speed of sound. For this study, we chose the *D2Q9* and *D3Q19* models, and the parameters mentioned above are given in [25, 26]. The external forcing term can be expressed as:

$$F_\alpha = t_\alpha \left( \frac{\mathbf{e}_\alpha - \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})\mathbf{e}_\alpha}{c_s^4} \right) \cdot \mathbf{F}_\alpha, \quad (3)$$

which can be expressed as a combination of a leading-order term and a higher-order terms [27]. The leading-order forcing term  $F_\alpha^*$  can be expressed as:

$$F_\alpha^* = t_\alpha \left( \frac{\mathbf{e}_\alpha}{c_s^2} \right) \cdot \mathbf{F}_\alpha, \quad (4)$$

and the higher-order forcing term  $F_\alpha^{**}$  is then:

$$F_\alpha^{**} = t_\alpha \left( \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})\mathbf{e}_\alpha}{c_s^4} - \frac{\mathbf{u}}{c_s^2} \right) \cdot \mathbf{F}_\alpha. \quad (5)$$

Here, the body force is represented by  $\mathbf{F}_\alpha$ . We then integrate Eq. (1) along the characteristic direction and apply the trapezoidal rule. In addition, we introduce the modified equilibrium distribution function as follows [28].

$$\bar{f}_\alpha^{eq} = f_\alpha^{eq} - 0.5\Delta t F_\alpha^*, \quad (6)$$

and the modified distribution:

$$\bar{f}_\alpha = f_\alpha + \frac{f_\alpha - f_\alpha^{eq}}{2\tau} - 0.5\Delta t F_\alpha^*, \quad (7)$$

where  $\tau = \lambda/\Delta t$  denotes the dimensionless relaxation time. The discrete Boltzmann equation can be then solved by a local collision function and

a non-local streaming function with the force splitting method [27]. The collision is conducted as follows:

$$f_\alpha^* = \bar{f}_\alpha - \frac{1}{\tau + 0.5}(\bar{f}_\alpha - \bar{f}_\alpha^{eq}) + \Delta t F_\alpha^{**}, \quad (8)$$

which is followed by the substitution:

$$\bar{f}_\alpha(\mathbf{x}, t - \Delta t) = f_\alpha^*(\mathbf{x}, t - \Delta t), \quad (9)$$

and the streaming is given as:

$$\left( \frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla \right) \bar{f}_\alpha = F_\alpha^*. \quad (10)$$

Finally, the density and momentum can be recovered by the distribution function by:

$$\rho = \sum_\alpha \bar{f}_\alpha, \quad (11)$$

$$\rho \mathbf{u} = \sum_\alpha \mathbf{e}_\alpha \bar{f}_\alpha. \quad (12)$$

## 2.2. Spectral Element Continuous Galerkin method

By the force splitting method [27], we can incorporate the higher-order forcing term  $F_\alpha^{**}$  into the collision operator to preserve the conserved moments, i.e.,  $\sum_\alpha F_\alpha^{**} = \sum_\alpha \mathbf{e}_\alpha F_\alpha^* = 0$ , while the leading-order forcing term  $F_\alpha^*$  is applied to the streaming step. We express the streaming process as a pure advection equation of the particle distribution function based on an unstructured mesh shown in Eq. (10), instead of the perfect shift. The weak form of Eq. (10) in a given element domain  $\Omega_e$ , can be obtained by multiplying the advection equation Eq. (10) by a test function  $\phi$ :

$$\left( \frac{\partial \bar{f}_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla \bar{f}_\alpha - F_\alpha^*, \phi \right)_{\Omega_e} = 0. \quad (13)$$

We can express the above equation in the following matrix form:

$$\mathbf{M} \frac{d\bar{\mathbf{f}}_\alpha}{dt} = -\mathbf{C}_\alpha \bar{\mathbf{f}}_\alpha + \mathbf{M} \mathbf{F}_\alpha^*, \quad (14)$$

where  $\mathbf{M}$  and  $\mathbf{C}_\alpha$  are the mass and convection matrices respectively, whose entries are given as:

$$M_{ij} = \int_{\Omega_e} \phi_i \phi_j \, d\Omega, \quad (15)$$

$$C_{\alpha_{ij}} = \sum_{k=1}^d \left( e_{\alpha, x_k} \int_{\Omega_e} \phi_i \partial_{x_k} \phi_j \, d\Omega \right). \quad (16)$$

The particle distribution function and the forcing terms defined on the nodes become vectors or one-dimensional arrays for sequential storage, shown as  $\bar{\mathbf{f}}_\alpha$  and  $\mathbf{F}_\alpha^*$ . We employ a tensor product basis of the 1D Legendre–Lagrange interpolation polynomials given as:

$$l_i(\xi) = \frac{-1}{N(N+1)} \frac{(1-\xi^2)L'_N(\xi)}{(\xi - \xi_i)L_N(\xi_i)} \quad (17)$$

where  $\xi_i$  are the Gauss-Lobatto-Legendre (GLL) quadrature nodes. Based on GLL quadrature, we can obtain the formulation for lumped diagonal mass matrix in 3D as:

$$\mathbf{M}[\hat{i}, \hat{i}] = \phi_i^2 J(\mathbf{r}) \rho_i \rho_j \rho_k, \quad \hat{i} = 1 + i + (N+1)j + (N+1)^2 k \quad (18)$$

where

$$\rho_i = \frac{2}{N(N+1)} \frac{1}{[L_N(\xi_i)]^2}$$

represents the specific weights, and  $J(\mathbf{r})$  denotes the Jacobian determinant according to the reference domain where:

$$J(\mathbf{r}) = \det \frac{\partial x_i}{\partial r_i} \quad (19)$$

The convection matrix  $\mathbf{C}_\alpha$  can be expressed as:

$$\mathbf{C}_\alpha = \underline{\phi}^T \mathbf{e}_\alpha \underline{\mathbf{D}} \underline{\phi}, \quad (20)$$

where

$$\underline{\mathbf{D}} = \begin{cases} (\mathbf{D}_x, \mathbf{D}_y)^T, & d = 2; \\ (\mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_z)^T, & d = 3, \end{cases} \quad (21)$$

is the derivative matrix composed of  $d$  matrices and  $\mathbf{D}_x$  is composed of the one-dimensional derivative matrix  $\hat{\mathbf{D}}_x$  related to GLL points. When we consider a two-dimensional case,

$$\begin{aligned}\mathbf{D}_x &= \mathbf{I} \otimes \hat{\mathbf{D}}_x, \\ \mathbf{D}_y &= \hat{\mathbf{D}}_y \otimes \mathbf{I},\end{aligned}\tag{22}$$

where  $\mathbf{I}$  is the identity matrix, and the operator  $\otimes$  denotes the tensor product [29]. For a three-dimensional simulation,

$$\begin{aligned}\mathbf{D}_x &= \mathbf{I} \otimes \mathbf{I} \otimes \hat{\mathbf{D}}_x, \\ \mathbf{D}_y &= \mathbf{I} \otimes \hat{\mathbf{D}}_y \otimes \mathbf{I}, \\ \mathbf{D}_z &= \hat{\mathbf{D}}_z \otimes \mathbf{I} \otimes \mathbf{I}.\end{aligned}\tag{23}$$

After we transfer the domain to the reference domain, from  $[x, y, z]$  to  $[r, s, t]$ , the chain rule needs to be considered and the change of variables should be done from  $\Omega$  to  $\hat{\Omega}$ :

$$\mathbf{C}_\alpha = \underline{\phi}^T \mathbf{e}_\alpha \mathbf{G} \mathbf{D}^* \underline{\phi},\tag{24}$$

where now

$$\mathbf{D}^* = \begin{cases} (\mathbf{D}_r, \mathbf{D}_s)^T, & d = 2; \\ (\mathbf{D}_r, \mathbf{D}_s, \mathbf{D}_t)^T, & d = 3, \end{cases}\tag{25}$$

The geometry information is then stored in the matrix  $\mathbf{G}$ , which is composed of  $d^2$  diagonal matrices:

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{pmatrix}, d = 2,\tag{26}$$

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & \mathbf{G}_{13} \\ \mathbf{G}_{21} & \mathbf{G}_{22} & \mathbf{G}_{23} \\ \mathbf{G}_{31} & \mathbf{G}_{32} & \mathbf{G}_{33} \end{pmatrix}, d = 3.\tag{27}$$

The diagonal matrix can then be expressed as

$$\mathbf{G}_{ij}[\hat{l}, \hat{l}] = \rho_l \rho_m \rho_n \sum_{k=1}^d \frac{\partial r_i}{\partial x_k} J(\mathbf{r}),\tag{28}$$

where  $\hat{l} = 1 + l + (N + 1)m + (N + 1)^2n$ .

### 2.2.1. Flux Bounce-Back Boundary Condition

From the scenario of the discontinuous Galerkin method, it is intuitive to numerically compute the flux at the interface as well as the boundary condition. Here, in our continuous Galerkin LBM, we weakly impose the boundary condition only at the boundary nodes, for which the bounce back scheme is implicitly applied [17]. As shown in previous work [17], we further introduce a surface integration of the flux term from the outer boundary rather than each element interface, and the entire equation can be expressed as:

$$\mathbf{M} \frac{\partial \bar{\mathbf{f}}_\alpha}{\partial t} + \mathbf{C}_\alpha \bar{\mathbf{f}}_\alpha = \mathbf{M} \mathbf{F}_\alpha^* + \mathbf{R}(\mathbf{n} \cdot \mathbf{j}_\alpha), \quad (29)$$

where  $\mathbf{R} = \int_{\partial\Omega_e} \phi_i \phi_j d\bar{\Omega}$  is the surface integration,  $\mathbf{n}$  is the normal vector, and  $\mathbf{j}_\alpha$  is the flux at the boundary. In this work, we apply the Lax-Friedrichs flux:

$$\mathbf{j}_\alpha = \begin{cases} \mathbf{e}_\alpha [\bar{\mathbf{f}}_\alpha]_{bc}, & \mathbf{n} \cdot \mathbf{e}_\alpha < 0, \\ \mathbf{0} & \mathbf{n} \cdot \mathbf{e}_\alpha \geq 0. \end{cases}$$

When we apply the bounce-back at the boundary, the flux term is expressed as:

$$[\bar{\mathbf{f}}_\alpha]_{bc} = \bar{\mathbf{f}}_\alpha - \bar{\mathbf{f}}_\beta - 2t_\alpha \rho_0 (\mathbf{e}_\alpha \cdot \mathbf{u}_b) / c_s^2, \quad (30)$$

for which  $\beta$  is the opposite direction of  $\alpha$ . When no slip is applied to the simulation, we simply apply flux bounce-back at the boundary. When we have a moving boundary with velocity  $\mathbf{u}_b$ , we further add the last term to maintain a constant velocity.

It is noted that the integration of the flux term is applied only to the distribution functions for incoming directions, while the other distribution functions will not be affected by this flux term. Essentially, when we have a flat surface boundary, the flux bounce-back will yield the same results as the normal bounce-back approach.

### 2.2.2. Time marching method

We consider the Runge-Kutta to explicitly solve the distribution function. To solve the distribution function  $\bar{\mathbf{f}}_\alpha$  in Eq. (29), we propose the 3rd order strong stability-preserving Runge-Kutta method for:

$$\frac{d\bar{\mathbf{f}}_\alpha}{dt} = \mathbf{L} \bar{\mathbf{f}}_\alpha, \quad (31)$$

where  $\mathbf{L} = \mathbf{I} + \mathbf{M}^{-1}(\mathbf{R} - \mathbf{C}_\alpha)$  is the spatial operator. The each turn of the SSP RK3 is then:

$$\bar{\mathbf{f}}_\alpha^1 = \bar{\mathbf{f}}_\alpha(t - \Delta t) + \frac{\Delta t}{3} \mathbf{L} \bar{\mathbf{f}}_\alpha(t - \Delta t), \quad (32)$$

$$\bar{\mathbf{f}}_\alpha^2 = \bar{\mathbf{f}}_\alpha^1 - \frac{3\Delta t}{4} \mathbf{L} (\bar{\mathbf{f}}_\alpha(t - \Delta t) - \bar{\mathbf{f}}_\alpha^1), \quad (33)$$

$$\bar{\mathbf{f}}_\alpha^3 = \bar{\mathbf{f}}_\alpha^2 + \frac{2\Delta t}{3} \mathbf{L} (\bar{\mathbf{f}}_\alpha(t - \Delta t) - \bar{\mathbf{f}}_\alpha^1 + \bar{\mathbf{f}}_\alpha^2), \quad (34)$$

$$\bar{\mathbf{f}}_\alpha(t) = \bar{\mathbf{f}}_\alpha^3 - \frac{\Delta t}{4} \mathbf{L} (\bar{\mathbf{f}}_\alpha(t - \Delta t) - \bar{\mathbf{f}}_\alpha^1 + \bar{\mathbf{f}}_\alpha^2 - \bar{\mathbf{f}}_\alpha^3). \quad (35)$$

As we get the the current time distribution function  $\bar{\mathbf{f}}_\alpha(t)$ , we can further evaluate the macroscopic density and momentum from Eq. (11) and (12).

### 2.2.3. Filter & Stabilization

One key difference between the current scheme and the conventional LBM lies in the streaming process. The conventional LBM employs perfect shift streaming without any numerical error. As a result, the stability of the conventional LBM is typically enhanced through the collision process [30]. Methods such as multi-relaxation time (MRT) collision [31], two-relaxation time (TRT) collision [32], increasing bulk viscosity [33], and regularizing the collision [34] operator are commonly used to stabilize the LBM solver for high Reynolds number simulations. Filtering macroscopic values, such as density or velocity, can introduce conservation issues and is therefore generally avoided [30]. In the current scheme, the perfect shift streaming is changed to the spectral element and RK scheme. Although the continuous Galerkin method ensures a continuous distribution function across the element interface, the gradient of the distribution function term, which appears in the convective term  $\mathbf{e}_\alpha \cdot \nabla f_\alpha$ , is not exactly continuous, which may induce stability issues.

In Nek5000 [35], when simulating a fluid system with a high Reynolds number, an explicit filter is used to suppress the spurious modes of pressure and momentum at the end of each time step. This filter interpolates the nodal solution using a pseudo-projection [24, 36]. After each solving process, the velocity and pressure terms in space are replaced with filtered solutions by applying a filter transfer function  $\sigma$ . For example, in an  $N^{\text{th}}$ -order spectral

element system, the filtered pressure term can be expressed as:

$$p = \sum_{k=0}^N \sigma_k p_k \phi_k(\xi), \quad (36)$$

where  $p_k$  is the physical space solution, and  $\phi_k(\xi)$  is the test function. The filter transfer function can be expressed as:

$$\sigma_k = \begin{cases} 1 - \gamma \left( \frac{k - k_c}{N - k_c} \right)^2, & k > k_c, \\ 1, & k \leq k_c. \end{cases} \quad (37)$$

In the above equation,  $k_c$  is defined as the cutoff mode, and  $\gamma$  denotes the amplitude. For example, when we choose  $k_c = 2$  and  $\gamma = 0.1$ , the filter is applied starting from the 2<sup>nd</sup> order solution, and the interpolation magnitude for higher-order solutions is 0.1. Further details on the construction of the filter and interpolation can be found in [24, 36]. However, it is important to note that, although this is a non-dissipative method, it violates the divergence-free condition for incompressible flow when we use the explicit filter on the velocity and pressure in Nek5000.

For our approach, we do not apply the explicit filter to the density and momentum, as this would violate the conservations  $\sum f_\alpha^{eq} = \sum f_\alpha$  and  $\sum f_\alpha^{eq} e_\alpha = \sum f_\alpha e_\alpha$ . Instead, the explicit filter is directly applied to the distribution function  $f_\alpha$  before the update step. In this case, conservation relations are preserved.

We summarize the full process of the scheme as:

- 1.** Initialize of the distribution on each node and element;
- 2.** Compute the loop from the collision by BGK, with higher order forcing term;
- 3.** Solve streaming as a hyperbolic partial differential equation with external lower order forcing term and boundary flux by SSP RK3;
- 4.** Apply the filter for each direction distribution function;
- 5.** Update the macroscopic parameters which are essential for the next collision step.

### 3. Simulation Results

The validations are performed for the investigation of the stability and boundary conditions. Our results are generated from a modified version

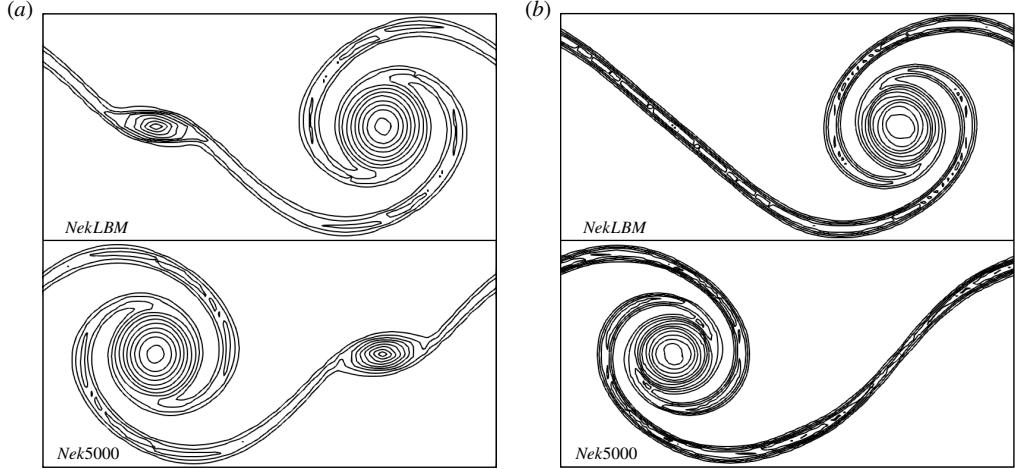


Figure 1: Comparison of the double shear layer simulation conducted by *NekLBM* and Nek5000 for  $128^2$  number of nodes. (a) Vortex contours  $\Delta\omega = 5$  for  $N_e = 32$ ,  $N = 4$  by *NekLBM*(top), Nek5000 (bottom). (b) Vortex contours  $\Delta\omega = 5$  for  $N_e = 16$ ,  $N = 8$  by *NekLBM*(top), Nek5000 (bottom).

of the Nek5000 code, which we henceforth refer to as NekLBM. NekLBM leverages many of the numerical operators and parallelization inherent in Nek5000 with the distinguishing feature of solving the discrete Boltzmann equations rather than the Navier-Stokes equations.

### 3.1. Double shear layer

The first validation case considered is the periodic double shear layer simulation, which is studied to evaluate the solver's stability. In previous LBM studies, Dellar increased the bulk viscosity to effectively resolve small vortices induced by instability [33], while Ricot et al. investigated various types of filters applied to the distribution function to address the problem of small vortices [30]. In this work, it is necessary to revisit this problem using the current model.

We initialize a velocity profile:

$$u_x = \begin{cases} u_0 \tanh[\kappa(y - 0.25)], & y \leq 0.5, \\ u_0 \tanh[\kappa(0.75 - y)], & y > 0.5, \end{cases} \quad (38)$$

$$u_y = \delta \sin(2\pi(x + 0.25))$$

in a square domain. The initial velocity magnitude  $u_0$  is set to satisfy  $Re = \rho u_0 L / \mu = 10,000$ , and the parameters related to the velocity profile are given as  $\delta = 0.05$ ,  $\kappa = 80$ . The simulation is conducted over the time interval  $T/t_0 = [0, 10]$ , where the time scale is  $t_0 = L/u_0$  and we compare the results from our scheme with those from Nek5000 using the same setup at  $T/t_0 = 10$ , as shown in Figure 1. In Figure 1(a), we fix the number of elements  $N_e = 32$  on each direction, and set the polynomial order  $N = 4$  for both methods. The small vortices appear in both simulation results because of the inaccurate evaluation of the convective term. When we increase the polynomial order to  $N = 8$  while keeping the total number of nodes in each direction  $N_{tot} = 128$ , our scheme produces better results, which are free from the small vortices, and consistent with Nek5000. As the polynomial order increases, the evaluation of the convective term  $\mathbf{e}_\alpha \cdot \nabla f_\alpha$  is more accurate in space. Although the total number of nodes is the same, a higher order simulation will generally provide more accurate results. It is worth noting that for both schemes, an explicit filter with amplitude  $\gamma = 0.1$ , and cutoff mode  $k_c = 2$  is applied. Without the explicit filter, both schemes fail to produce meaningful results.

### 3.2. Unsteady Couette flow

A convergence test for 2D Couette flow simulation is investigated to validate the boundary condition for the current method. We initialize a moving wall at the top of the square domain, and keep a stationary no-slip boundary on the bottom. The left and right boundary conditions are set to periodic. We first test the case with  $Re = u_{top}L/\nu = 10$  and  $Ma = u_{top}/c_s = 0.1$ . The viscous stress will induce the flow over the domain. For the parameters, the number of elements in each direction is set to  $N_e = 6$  and the polynomial order is set to  $N = 4$ . The simulation results are compared with the analytic solutions shown in Figure 2(a) and consistent results are obtained.

We further increase the Reynolds number to  $Re = 2000$ , and set  $Ma = 0.05$ . We evaluate the  $L_\infty$  errors at  $T/t_0 = 40$ , where  $t_0 = L/u_{top}$ , and conduct the convergence test as we increase  $N$  from  $N = 4$  to  $N = 14$ . As shown in Figure 2(b), the errors show exponential convergence as  $N$  increases until it is saturated by temporal error.

### 3.3. Planar Poiseuille flow

We further test the current model by simulating a 2D planar Poiseuille flow to study the "numerical slip" and the effect of implementing the flux

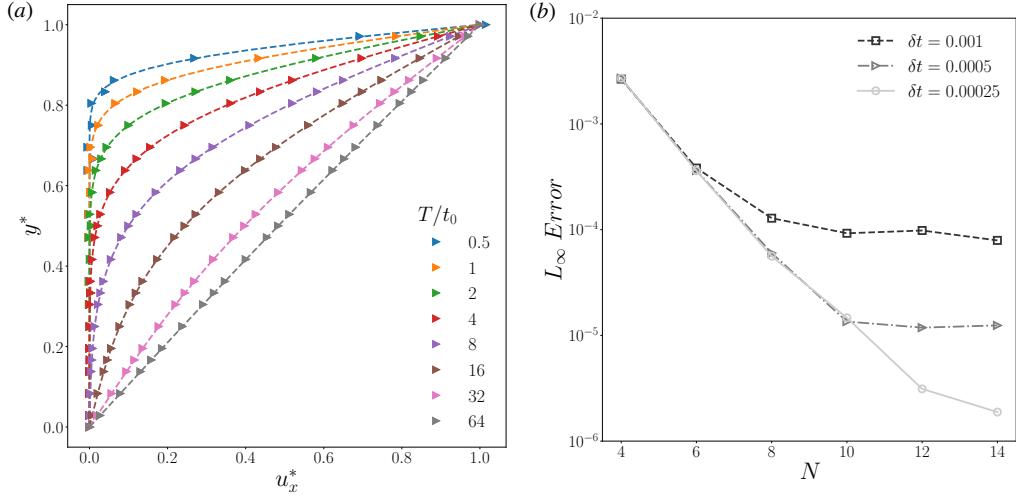


Figure 2: (a) Comparison between simulation results (triangular markers) and analytical solutions (dashed line) of the unsteady Couette flow during  $T/t_0 = [0.5 - 64]$  for  $Re = 10$ ,  $Ma = 0.1$ . (b) Spatial convergence for the unsteady Couette flow with different  $\delta t$  for  $Re = 2000$ ,  $Ma = 0.05$ .

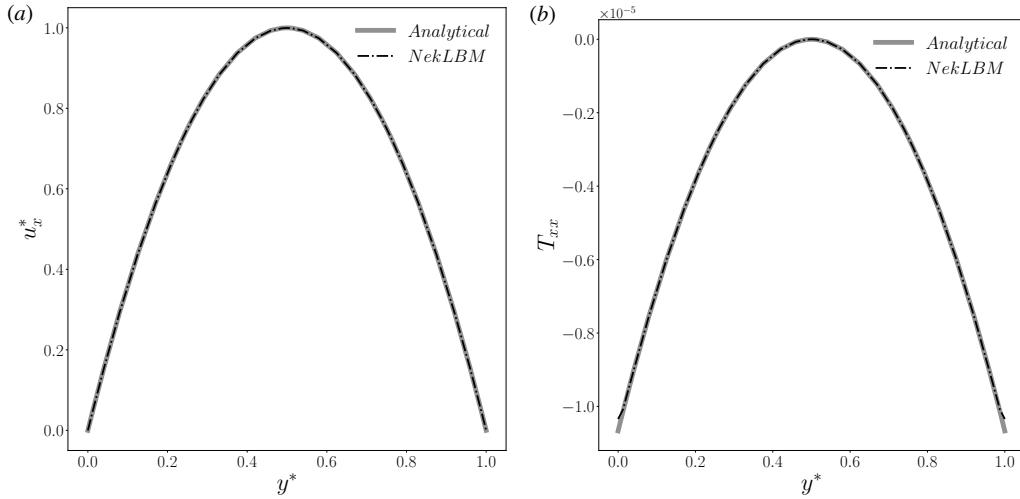


Figure 3: Comparison between simulation results (black dashed line) and analytical solutions (gray) of (a) velocity and (b) stress for the 2D planar Poiseuille flow at  $T/t_0 = 64$  with  $Ma = 0.1/\sqrt{3}$ ,  $Re = 100$ .

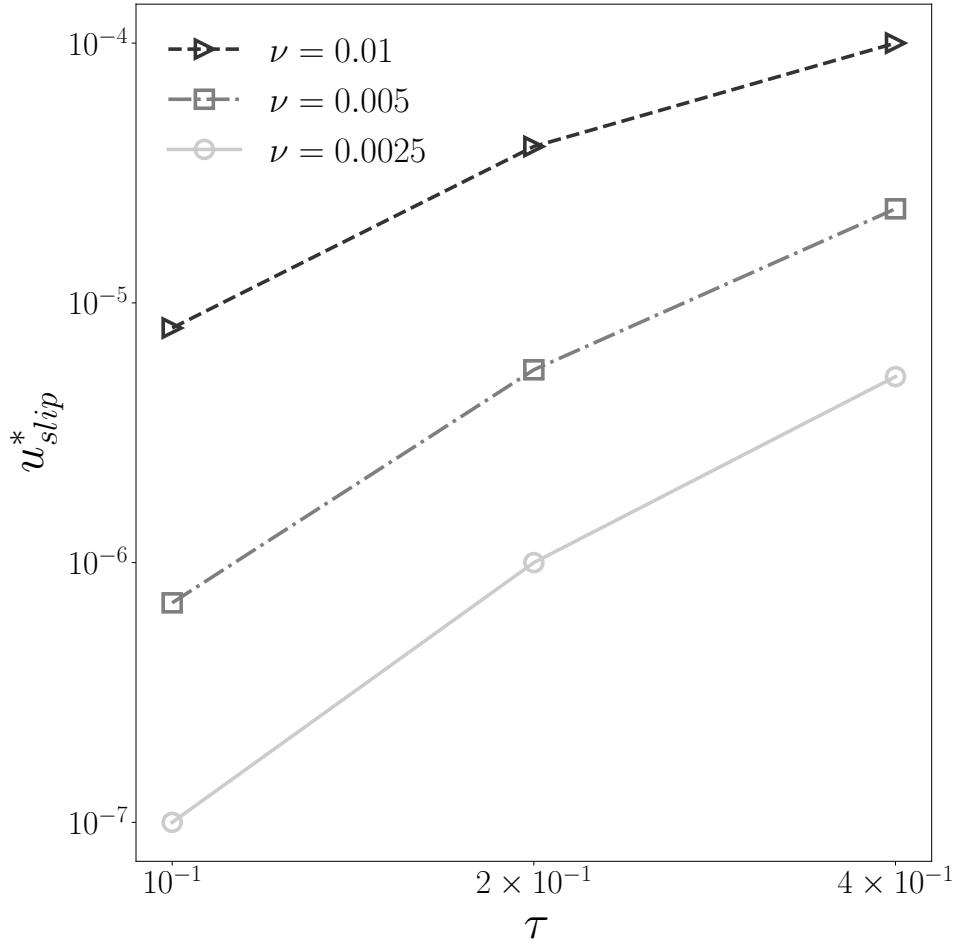


Figure 4: Convergence test of the dimensionless slip velocity  $u_{slip}^* = u_x(y=0)/u_{max}$  with relaxation time  $\tau = [0.1, 0.4]$ , and kinematic viscosity  $\nu = [0.0025, 0.01]$ .

boundary condition. The nonphysical "numerical slip" at the boundary of solids cannot be fully removed by node-based bounce-back method [37, 38]. We can improve the method by an improved collision operator i.e. MRT/TRT or coupling linked-based bounce-back with momentum exchange boundary to recover exact solutions [39, 40]. The current boundary condition considers the momentum exchange through adding the flux term. Moreover, this method can highly improve the stability for unstructured mesh boundaries.

The simulation is conducted in a square domain with  $N_e = 4$  and  $N = 8$ . The initial velocity is set to  $\mathbf{u} = \mathbf{0}$ . To drive the flow in the confined planar, we initialize a pressure gradient,  $G$ , between the left inlet boundary and the right outlet boundary. The steady state velocity profile can be analytically computed as:

$$u_x = \frac{G}{2\mu}y(y - L), \quad (39)$$

where  $y$  is the vertical coordinate,  $L$  is the side length, and  $\mu$  represents the viscosity. The top and bottom boundaries are set to no slip boundary condition. It is noted that besides the steady velocity, the deviatoric stress  $\mathbf{T}$  has to be concerned, since it is naturally recovered by the high order moments of the distribution function. The simulation with large relaxation time will affect the accuracy of the recovered Navier-Stokes equation [41]. In this example, the normal stress,  $T_{xx}$ , which can be expressed as

$$T_{xx} = -2\mu\tau(u'_x)^2 \quad (40)$$

decades to zero on the boundary [21].

We first look at the consistency between the numerical results and the analytical solutions shown in Figure 3. In this example, we fixed the Mach number  $Ma = 0.1/\sqrt{3}$ , and the Reynolds number  $Re = 100$ . We evaluate the dimensionless horizontal velocity  $u_x^* = u_x/u_{max}$  and normal stress along the  $y$  axis when the system achieves the steady state. The velocity profile is highly consistent with the analytical solution. In addition, we do not see obvious spurious oscillations near the boundary region. However, we notice a small difference for stress at the boundary.

We further test the effect of the relaxation time and investigate the convergence of the current boundary as shown in Figure 4. In this test, we fix the maximum velocity by fixing the pressure gradient  $G$  but change the relaxation parameter,  $\tau$ , and the time interval,  $\Delta t$ . We first notice that the current model converges to the small time interval, that is, as we decrease

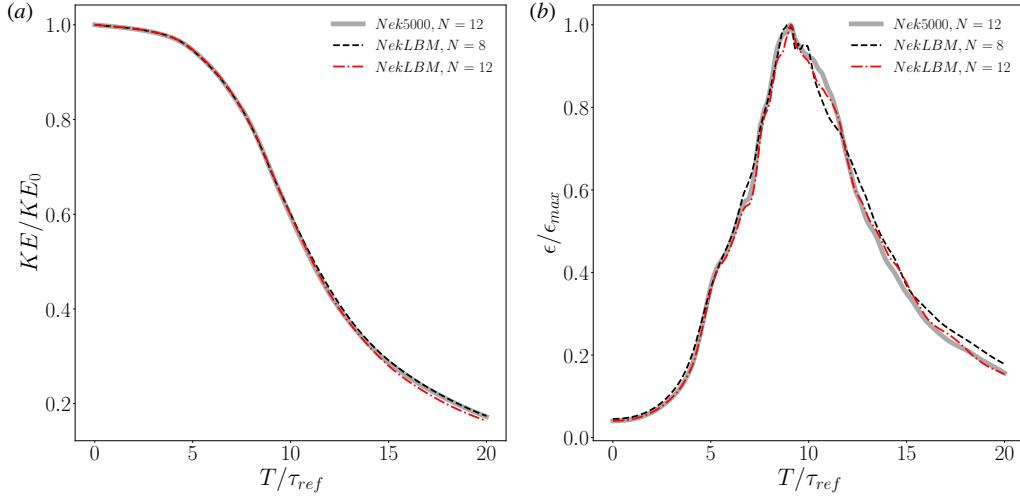


Figure 5: Evolution of (a) Kinetic energy and (b) energy dissipation during  $T/t_0 = [0, 20]$  of the Taylor Green vortex simulation for the same element number  $N_e = 16$  on each direction, and different polynomial order  $N = 8, N = 12$ .

the time interval, we will get a smaller "numerical error". However, for the same time interval, as we increase the dimensionless relaxation parameter, a large slip velocity occurs. We learn that since the time marching is solved by SSPRK3 rather than a perfect shift, we can expect a time error at the boundaries. Secondly, the pressure gradient is determined from the viscosity and the relaxation parameter, which may induce a large slip error at the boundary. Besides that, we still notice a more than linear increase in slip velocity which is induced by collision. A further improvement of the boundary condition can be considered by implementing different collision model to obtain a smaller slip velocity.

### 3.4. Taylor Green vortex

We then verify the current approach using the 3D Taylor Green vortex case [42] to evaluate the energy dissipation in isotropic turbulence. For this simulation, we use a D3Q19 LBM model [6, 1]. The simulation is initialized with an isotropic velocity profile within a 3D cube with a side length of

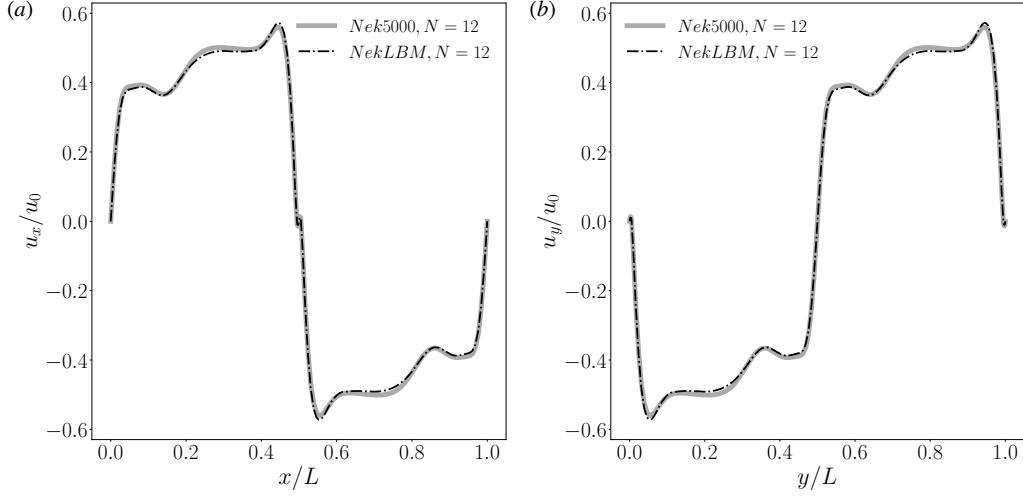


Figure 6: Velocity in the (a)  $x$  ( $u_x$ ) and (b)  $y$  ( $u_y$ ) directions along the  $x$  and  $y$  axial centers at  $T/t_0 = 12.11$ .

$L/L_0 = 2\pi$ :

$$\begin{aligned} u_x(x, y, z, 0) &= u_0 \sin\left(\frac{x}{L_0}\right) \cos\left(\frac{y}{L_0}\right) \cos\left(\frac{z}{L_0}\right), \\ u_y(x, y, z, 0) &= -u_0 \cos\left(\frac{x}{L_0}\right) \sin\left(\frac{y}{L_0}\right) \cos\left(\frac{z}{L_0}\right), \\ u_z(x, y, z, 0) &= 0, \end{aligned}$$

For this simulation, we set  $Ma = u_0/c_s = 0.1/\sqrt{3}$ ,  $Re = \rho_0 L_0 / \nu_0 = 1600$ . The time scale is defined as  $t_0 = L_0/u_0$ . Based on previous research, we evaluate the energy evolution and dissipation rate over the time interval  $T/t_0 = [0, 20]$ , comparing the simulation results with Nek5000. Two simulations are performed using the current method, with same element numbers on each direction  $N_e = 16$ , employing the polynomial order  $N = 8$  and  $N = 12$ . The first comparison focuses on the kinetic energy  $KE = 0.5\rho_0 \mathbf{u}^2$ , scaled by the initial kinetic energy  $KE_0$ . As shown in Figure 5(a), both simulations exhibit a consistent trend with Nek5000 for  $N = 12$ .

Another comparison, shown in Figure 5(b), evaluates the dissipation rate as:

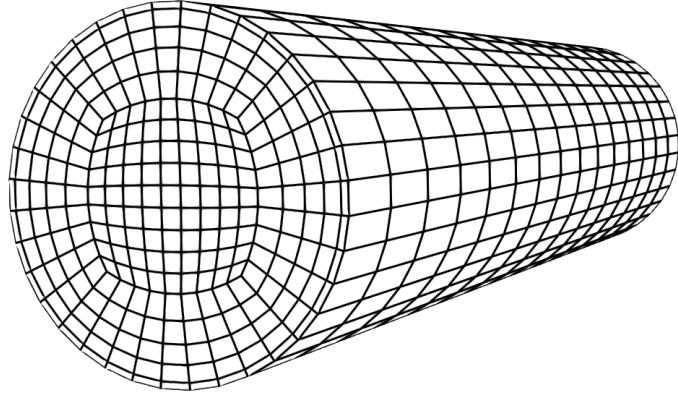


Figure 7: Elements over the pipe.

$$\begin{aligned} \epsilon = & 2\mu \left[ \left( \frac{\partial u_x}{\partial x} \right)^2 + \left( \frac{\partial u_y}{\partial y} \right)^2 + \left( \frac{\partial u_z}{\partial z} \right)^2 - \frac{1}{3} \nabla \cdot \mathbf{u} \right] + \\ & \mu \left[ \left( \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y} \right)^2 + \left( \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \right)^2 + \left( \frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x} \right)^2 \right]. \end{aligned}$$

The dissipation rate is scaled by the maximum dissipation,  $\epsilon_{max}$ . Our approach captures the consistent increase and decrease in the dissipation rate for both cases.

To perform a more sensitive comparison, we analyze the velocity profile at  $T/t_0 = 12.11$  along the centerlines in the  $x$  and  $y$  directions. This comparison, conducted between our approach and Nek5000, is shown in Figure 6. Our approach with  $N = 12$  produces highly consistent results compared to Nek5000 with same set up.

### 3.5. Pipe Poiseuille flow

Finally, we perform a 3D pipe Poiseuille flow simulation using the current scheme. The geometry of the pipe is constructed using GMSH, with the ratio between the diameter and the length  $L/D = 4$ . An example of the mesh, shown in Figure 7, consists of  $N_e = 3915$  elements. To capture detailed

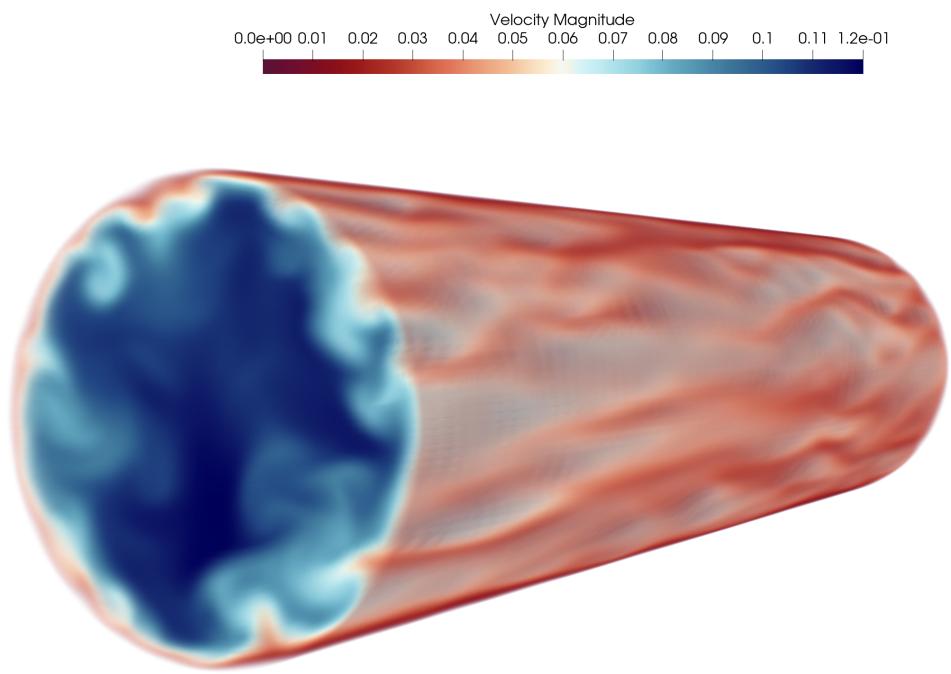


Figure 8: Volume contours of instantaneous streamwise velocity of a fully-developed turbulent pipe flow at  $T/t_0 = 7.2$  where  $Re \approx 5300$ . The simulation is conducted with  $N_{tot} = 3915$ , and  $N = 6$ .

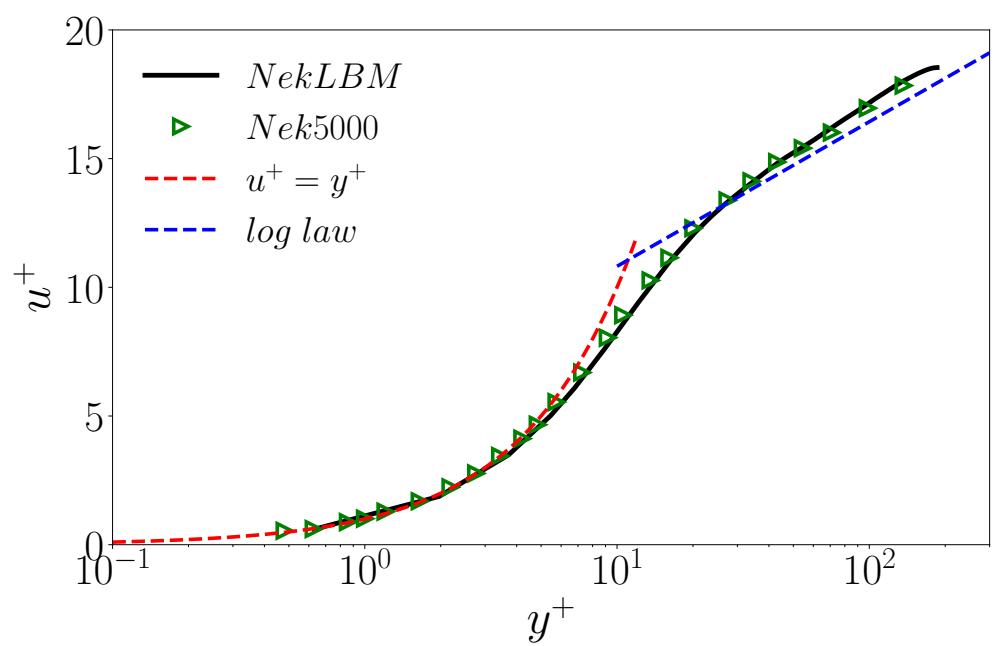


Figure 9: Comparison of the simulation results of turbulent flow in a cylindrical pipe with the law of the wall and Nek5000.

boundary layer information, relatively refined elements are assigned near the edge regions.

To obtain fully developed turbulence, we initialize the velocity profile which contains the viscous sublayer and inertial sublayer:

$$u^+ = \begin{cases} y^+, & \text{viscous region;} \\ \frac{1}{\kappa} \ln y^+ + C^+, & \text{inertial region.} \end{cases} \quad (41)$$

Here, the parameter  $\kappa = 0.41$  represents the Von Kármán constant, and  $C^+ \approx 5$ . In the equations above,  $u^+ = u_z/u_\tau$  and  $u_\tau = \sqrt{\tau_w/\rho}$  are the dimensionless velocity and friction velocity, respectively. The scaled wall coordinate is  $y^+ = yu_\tau/\nu$ , where  $\nu$  is the kinematic viscosity.

We apply the forcing approach suggested in [43] to drive the turbulence. To maintain turbulence, the flow is forced by the body force  $(F_x, F_y, F_z) = (0, 0, \rho g)$ , where  $g$  is related to the friction velocity  $u_\tau$  as  $g = 2u_\tau^2/R$ . A further non-uniform, divergence-free force field to the flow during the first three large-eddy turnover times of the simulation,  $T/t_0 < 3$  where  $t_0 = R/u_\tau$ , is given as:

$$F'_r = -g\kappa B_0 \frac{R k_z l}{r L} \sin\left(\frac{2\pi t}{P}\right) \left\{ 1 - \cos\left[\frac{2\pi(R - r - l_0)}{l}\right] \right\} \cos\left(k_z \frac{2\pi z}{L}\right) \cos(k_\theta \theta), \quad (42)$$

$$F'_\theta = g(1-\kappa) B_0 \frac{k_z}{k_\theta} \frac{2\pi R}{L} \sin\left(\frac{2\pi t}{P}\right) \sin\left[\frac{2\pi(R - r - l_0)}{l}\right] \cos\left(k_z \frac{2\pi z}{L}\right) \sin(k_\theta \theta), \quad (43)$$

$$F'_z = -g B_0 \frac{R}{r} \sin\left(\frac{2\pi t}{P}\right) \sin\left[\frac{2\pi(R - r - l_0)}{l}\right] \sin\left(k_z \frac{2\pi z}{L}\right) \cos(k_\theta \theta). \quad (44)$$

In the above forcing equations,  $k_r$  and  $k_\theta$  are wavenumbers of the perturbation in streamwise and azimuthal directions.  $P$  is the forcing period. In this case, it is set as  $P/t_0 = 8$ . The forcing magnitude is expressed as  $B_0 = 50.0$ , and the weighting parameter is denoted as  $\kappa = 0.5$ . The above forcing is added only to the region  $l_0 \leq R - r \leq l_0 + l$  in radial direction, where  $l_0 = 0.2R$ ,  $l = 0.4R$ . In current scheme, the forcing is then transferred to the cartesian coordinate as,  $(F'_x, F'_y, F'_z) = (F'_r \cos \theta - F'_\theta \sin \theta, F'_r \sin \theta + F'_\theta \cos \theta, F'_z)$ . Finally, the above forcing field is applied to collision and streaming by means

of the force split method shown in Eqs. (4) and (5). The further validations and details about this forcing approach can be found in [43].

The simulation is carried out based on the friction Reynolds number  $Re_\tau = u_\tau R/\nu = 180$ , and the fully developed turbulence is obtained based on the Reynolds number  $Re = u_0 D/\nu \approx 5300$ , where  $u_0$  is the mean velocity. We present the volume contours of the fully developed turbulence at  $T/t_0 = 7.2$  in Figure 8. Our goal is to recover the law of the wall, which provides a reliable approximation of the velocity profile in natural streams. We evaluate the wall shear stress,  $\tau_w = \mu(\sin\theta\partial u_z/\partial y + \cos\theta\partial u_z/\partial x)$ , using the global average value located at the boundary nodes. The average velocity profile in the axial direction,  $u_z$ , is approximated by both the time and spatial averages. For a high Reynolds number, in the inner region of the pipe, the mean velocity parallel to the wall follows a self-similar logarithmic law, and the viscous effects can be neglected. However, near the wall, viscosity significantly affects the flow field.

As shown in Figure 9, the simulation results are consistent with the law of the wall. Closest to the wall, the velocity profile follows the behavior of the near-wall laminar sublayer. This is followed by a transitional buffer region. In the inner region of the pipe far from the wall, we observe the log-law behavior.

#### 4. Conclusion Remarks

We present a spectral element continuous Galerkin Lattice Boltzmann method (LBM) to evaluate its capability in solving fluid flow problems in an unstructured mesh. The boundary condition is solved by the flux boundary method which is first applied to discontinuous Galerkin LBM in [17]. Instead of using the perfect shift streaming operator in regular LBM, we solve the streaming step using the SSPRK3 scheme. Unlike previous studies employing the discontinuous Galerkin method, we explore the continuous Galerkin method to further reduce the dissipation introduced by the flux term at element interfaces. To enhance stability, we validate an explicit filter through sensitivity for the double shear layer simulation.

The flux boundary condition is rigorously validated using unsteady Couette flow and planar Poiseuille flow. Both the velocity and stress profiles exhibit excellent agreement with analytical solutions. However, as the relaxation time increases, a noticeable increase in slip velocity is observed.

Conversely, reducing the time step results in a relatively smaller slip velocity. Further investigation is required to understand the relationship between slip velocity and relaxation time.

We also validate the dissipation characteristics of the current scheme using the Taylor Green vortex test case. As the number of elements increases, we observe a convergence trend and consistent energy dissipation, which aligns with results from Nek5000.

Finally, we apply the proposed scheme to simulate fully developed turbulent flow in a cylindrical pipe. The simulation is initialized with a constant body forcing between the inlet and outlet. In addition, we apply a non-uniform, divergence-free forcing field to the flow. As the Reynolds number and shear stress reach equilibrium, we compute the average velocity profile and boundary stress. The results recover the turbulent boundary layer profile, accurately capturing the viscous region, buffer region, and log-law region.

## 5. Acknowledgment

This material is based upon work supported by the U.S. Department of Energy (DOE), Office of Nuclear Energy, under Award No. DE-NE0009420, the DOE Office of Science-Advanced Scientific Computing Research Program, under Contract No. DE-AC02-06CH11357, and the National Science Foundation under Grant No. 2344147. This research used resources of the Argonne Leadership Computing Facility, a DOE Office of Science user facility at Argonne National Laboratory.

## References

- [1] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E. M. Viggen, *The lattice boltzmann method*, Springer International Publishing 10 (978-3) (2017) 4–15.
- [2] A. Mohamad, *Lattice boltzmann method*, Vol. 70, Springer, 2011.
- [3] D. R. Noble, J. G. Georgiadis, R. O. Buckius, Comparison of accuracy and performance for lattice boltzmann and finite difference simulations of steady viscous flow, *International Journal for Numerical Methods in Fluids* 23 (1) (1996) 1–18.

- [4] C. Zhao, S. Patel, T. Lee, A difference-free conservative phase-field lattice boltzmann method, *Computers & Fluids* (2025) 106674.
- [5] C. Zhao, Ternary flow simulation based on the conservative phase field lattice boltzmann method, Ph.D. thesis, The City College of New York (2022).
- [6] X. He, L.-S. Luo, Lattice boltzmann model for the incompressible navier–stokes equation, *Journal of statistical Physics* 88 (1997) 927–944.
- [7] M. Junk, A. Klar, L.-S. Luo, Asymptotic analysis of the lattice boltzmann equation, *Journal of Computational Physics* 210 (2) (2005) 676–704.
- [8] F. Dubois, Equivalent partial differential equations of a lattice boltzmann scheme, *Computers & Mathematics with Applications* 55 (7) (2008) 1441–1449.
- [9] T. Lee, C.-L. Lin, An eulerian description of the streaming process in the lattice boltzmann equation, *Journal of Computational Physics* 185 (2) (2003) 445–471.
- [10] R. V. Garimella, M. S. Shephard, Boundary layer mesh generation for viscous flow simulations, *International Journal for Numerical Methods in Engineering* 49 (1-2) (2000) 193–218.
- [11] T. K. Perkins, O. Johnston, A review of diffusion and dispersion in porous media, *Society of Petroleum Engineers Journal* 3 (01) (1963) 70–84.
- [12] S. Jiang, J. Tu, X. Yang, N. Gui, A review of pebble flow study for pebble bed high temperature gas-cooled reactor, *Experimental and Computational Multiphase Flow* 1 (2019) 159–176.
- [13] H. Xi, G. Peng, S.-H. Chou, Finite-volume lattice boltzmann method, *Physical Review E* 59 (5) (1999) 6202.
- [14] W. Li, L.-S. Luo, Finite volume lattice boltzmann method for nearly incompressible flows on arbitrary unstructured meshes, *Communications in Computational Physics* 20 (2) (2016) 301–324.

- [15] S. C. Mishra, H. K. Roy, Solving transient conduction and radiation heat transfer problems using the lattice boltzmann method and the finite volume method, *Journal of Computational Physics* 223 (1) (2007) 89–107.
- [16] D. V. Patil, K. Lakshmisha, Finite volume tvd formulation of lattice boltzmann simulation on unstructured mesh, *Journal of Computational Physics* 228 (14) (2009) 5262–5279.
- [17] M. Min, T. Lee, A spectral-element discontinuous galerkin lattice boltzmann method for nearly incompressible flows, *Journal of Computational Physics* 230 (1) (2011) 245–259.
- [18] T. Lee, C.-L. Lin, A characteristic galerkin method for discrete boltzmann equation, *Journal of Computational Physics* 171 (1) (2001) 336–356.
- [19] M. H. Saadat, I. V. Karlin, Arbitrary lagrangian–eulerian formulation of lattice boltzmann model for compressible flows on unstructured moving meshes, *Physics of Fluids* 32 (4) (2020).
- [20] T. Reis, P. J. Dellar, Lattice boltzmann simulations of pressure-driven flows in microchannels using navier–maxwell slip boundary conditions, *Physics of Fluids* 24 (11) (2012).
- [21] T. Reis, On the lattice boltzmann deviatoric stress: analysis, boundary conditions, and optimal relaxation times, *SIAM Journal on Scientific Computing* 42 (2) (2020) B397–B424.
- [22] T. Zhang, B. Shi, Z. Guo, Z. Chai, J. Lu, General bounce-back scheme for concentration boundary condition in the lattice-boltzmann method, *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 85 (1) (2012) 016701.
- [23] W. Li, W. Li, A gas-kinetic bgk scheme for the finite volume lattice boltzmann method for nearly incompressible flows, *Computers & Fluids* 162 (2018) 126–138.
- [24] M. O. Deville, P. F. Fischer, E. H. Mund, High-order methods for incompressible fluid flow, Vol. 9, Cambridge university press, 2002.

- [25] C. Zhao, T. Lee, Interaction between a rising bubble and a stationary droplet immersed in a liquid pool using a ternary conservative phase-field lattice boltzmann method, *Physical Review E* 107 (2) (2023) 025308.
- [26] T. Lee, C.-L. Lin, A stable discretization of the lattice boltzmann equation for simulation of incompressible two-phase flows at high density ratio, *Journal of Computational Physics* 206 (1) (2005) 16–47.
- [27] S. Patel, T. Lee, A new splitting scheme to the discrete boltzmann equation for non-ideal gases on non-uniform meshes, *Journal of Computational Physics* 327 (2016) 799–809.
- [28] T. Lee, P. Fischer, Eliminating parasitic currents in the lattice boltzmann equation method for nonideal gases, *Physical Review E* 74 (2006) 046709.
- [29] P. Fischer, S. Kerkemeier, M. Min, Y.-H. Lan, M. Phillips, T. Rathnayake, E. Merzari, A. Tomboulides, A. Karakus, N. Chalmers, et al., Nekrs, a gpu-accelerated spectral element navier–stokes solver, *Parallel Computing* 114 (2022) 102982.
- [30] D. Ricot, S. Marié, P. Sagaut, C. Bailly, Lattice boltzmann method with selective viscosity filter, *Journal of Computational Physics* 228 (12) (2009) 4478–4490.
- [31] L.-S. Luo, W. Liao, X. Chen, Y. Peng, W. Zhang, Numerics of the lattice boltzmann method: Effects of collision models on the lattice boltzmann simulations, *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 83 (5) (2011) 056710.
- [32] I. Ginzburg, F. Verhaeghe, D. d’Humieres, et al., Two-relaxation-time lattice boltzmann scheme: About parametrization, velocity, pressure and mixed boundary conditions, *Communications in computational physics* 3 (2) (2008) 427–478.
- [33] P. J. Dellar, Bulk and shear viscosities in lattice boltzmann equations, *Physical Review E* 64 (3) (2001) 031203.
- [34] J. Latt, B. Chopard, Lattice boltzmann method with regularized pre-collision distribution functions, *Mathematics and Computers in Simulation* 72 (2-6) (2006) 165–168.

- [35] Nek5000: <https://github.com/Nek5000/Nek5000>, GitHub repository.
- [36] P. Fischer, J. Mullen, Filter-based stabilization of spectral element methods, Comptes Rendus de l'Académie des Sciences-Series I-Mathematics 332 (3) (2001) 265–270.
- [37] X. He, Q. Zou, L.-S. Luo, M. Dembo, Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice boltzmann bgk model, Journal of Statistical Physics 87 (1997) 115–136.
- [38] I. Ginzbourg, P. Adler, Boundary flow condition analysis for the three-dimensional lattice boltzmann model, Journal de Physique II 4 (2) (1994) 191–214.
- [39] C. Cercignani, Kinetic theory with “bounce-back” boundary conditions, Transport Theory and Statistical Physics 18 (1) (1989) 125–131.
- [40] A. J. Ladd, Numerical simulations of particulate suspensions via a discretized boltzmann equation. part 1. theoretical foundation, Journal of fluid mechanics 271 (1994) 285–309.
- [41] T. Reis, Burnett order stress and spatially-dependent boundary conditions for the lattice boltzmann method, Commun. Comput. Phys. 27 (2020) 167–197.
- [42] M. E. Brachet, D. I. Meiron, S. A. Orszag, B. Nickel, R. H. Morf, U. Frisch, Small-scale structure of the taylor–green vortex, Journal of Fluid Mechanics 130 (1983) 411–452.
- [43] C. Peng, N. Geneva, Z. Guo, L.-P. Wang, Direct numerical simulation of turbulent pipe flow using the lattice boltzmann method, Journal of Computational Physics 357 (2018) 16–42.