High-order flux reconstruction thermal lattice Boltzmann flux solver for simulation of incompressible thermal flows

Chao Ma,[1,2] Jie Wu,[1,2,*] Xiangyu Gu,[3,†] and Liming Yang[2]

[1]*State Key Laboratory of Mechanics and Control of Mechanical Structures, Nanjing University of Aeronautics and Astronautics, Yudao Street 29, Nanjing, Jiangsu 210016, China*

[2]*Key Laboratory of Unsteady Aerodynamics and Flow Control, Ministry of Industry and Information Technology, and Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics, Yudao Street 29, Nanjing, Jiangsu 210016, China*

[3]*China Academy of Launch Vehicle Technology, Nandahongmen Street 1, Beijing 100076, China*

(Received 20 April 2022; revised 28 June 2022; accepted 17 August 2022; published 6 September 2022)

In this paper, a high-order solver combining the flux reconstruction (FR) method and the thermal lattice Boltzmann flux solver (FRTLBFS) is developed for accurately and efficiently simulating incompressible thermal flows. The conservative differential equations recovered from Chapman-Enskog analysis of the thermal lattice Boltzmann equation are solved by the high-order FR method. The thermal lattice Boltzmann method is only applied to reconstruct the local solution used for evaluating fluxes at the solution and flux points. Unlike the traditional Navier-Stokes–Boussinesq (NSB) solvers where the inviscid and viscous terms are treated separately, the inviscid and viscous fluxes in the current FRTLBFS are coupled and computed uniformly. In comparison with the recently developed high-order flux reconstruction thermal lattice Boltzmann method, the FRTLBFS holds advantages such as high-order accuracy, good stability, and compactness but is more efficient and low storage, since only macroscopic flow variables including density, velocity, and temperature are stored and evolved. In addition, the physical boundary conditions in FRTLBFS can be directly implemented by using the same method as in conventional NSB solvers. Numerical validations of the proposed method are implemented by simulating (a) the porous plate problem, (b) natural convection in a square cavity, (c) unsteady natural convection in a tall cavity, and (d) thermal lid-driven cavity flow. Numerical results demonstrate that the present solver is an attractive tool to simulate incompressible thermal flows due to its high-order accuracy, stability, and low memory cost.

DOI: 10.1103/PhysRevE.106.035301

I. INTRODUCTION

Accurate and efficient simulation of thermal flows has always been an important topic in the numerical heat transfer (NHT) community. The present popular high-order method is able to reach the same accuracy with fewer degrees of freedom (DOF) compared with a low-order method, or, in other words, to obtain a result of higher accuracy with the same number of DOF [1]. Several investigations have been reported to adopt high-order schemes to solve the Navier-Stokes–Boussinesq (NSB) system or compressible NS equations at low Mach (Ma) numbers. Bassi *et al.* [2] applied the discontinuous Galerkin (DG) method together with the artificial compressibility (AC) method to solve natural convection problems. Wakashima and Saitoh [3] used the high-order finite difference (FD) scheme to get the benchmark numerical solutions for a three-dimensional natural convection heat transfer problem in a cubical cavity. Busto *et al.* [4] proposed a new family of high-order staggered semi-implicit DG methods for the simulations of natural convection problems. Zhao and Tian [5] developed a high-order upwind compact scheme for the solutions of time-dependent nature convection problems based on the stream function–vorticity form of NS equations. Klein *et al.* [6] adopted the DG and semi-implicit method for pressure-linked equation (SIMPLE) to simulate variable density flows at low Ma numbers. Recently, the high-order flux reconstruction (FR) or correction procedure via reconstruction (CPR) methods have become appealing since they unify several existing high-order schemes, including the nodal DG and spectral difference (SD) methods, within a single framework [7,8]. The FR scheme solves the governing equations in differential form, and no surface or volume integrals are required. It is simpler and more efficient than DG, yet it retains the advantages of DG, such as compactness and good adaptability of complicated geometries [9]. At present, a number of compressible and incompressible flow solvers based on the FR scheme have been proposed [10–13]. Recently, Yu *et al.* [14] combined FR with the preconditioning method to solve two-dimensional (2D) steady NS equations at low Ma numbers, and vertical heated natural convection cases with both small and large temperature differences are studied.

Apart from the macroscopic scale, other kinds of methods are constructed on the mesoscopic scale; a typical representative is the lattice Boltzmann method (LBM), which has developed quickly over the past two decades due to its simplicity, kinetic nature, low numerical dissipation, and easy parallel computing. Several thermal models presented in the framework of LBM include the multispeed model [15,16], the

[*] wuj@nuaa.edu.cn

[†] Xiangyu_gu@outlook.com

double-distribution function (DDF) model [17–19], and the hybrid model [20,21]. However, the standard LBM has some restrictions, such as the use of uniform Cartesian grids, the tie-up between time step and mesh spacing, and the poor stability at high Reynolds number (Re) or Rayleigh number (Ra). In addition, the standard LBM is a second-order scheme, and it is quite tedious to construct a high-order scheme of LBM due to its unique discretization form. These shortcomings greatly limit the applications of this method to complex geometries and flows. In order to be free of these constraints of standard LBM, two types of off-lattice approaches were proposed in the early years. One is the interpolation method [22,23] and the other is directly solving the discrete velocity Boltzmann equation (DVBE) by some conventional discretization methods [24–26]. Moreover, the corresponding high-order schemes of the two off-lattice methods have also been developed [27,28]. However, off-lattice methods still suffer from high cost in virtual memory and inconvenient implementation of physical boundary condition. To address the problems, without the need to solve the discrete distribution functions separately, two sorts of methods, which are called simplified LBM (SLBM) [29] and lattice Boltzmann flux solver (LBFS) [30], were proposed. In recent years, both of these methods have been developed to simulate various types of flows and the corresponding high-order schemes are constructed [31–33]. In this paper, the thermal LBFS (TLBFS) proposed by Wang *et al.* [34] is adopted. The original TLBFS is based on a second-order FV scheme and the fluxes at the cell interface are evaluated by local reconstruction of the thermal LBM (TLBM) solution. TLBFS overcomes the previously mentioned drawbacks of LBM, and it can be flexibly applied on nonuniform grids with curved boundaries. Recently, Liu *et al.* [35] have combined the least-square-based finite difference (LSFD) scheme with the TLBFS to construct a high-order FV thermal flow solver on unstructured grids. Their method is accurate and efficient. Moreover, their method inherits the advantages of the FV scheme, such as good adaptability of complicated geometries and conservation. However, as a high-order FV scheme, wide stencils are required to construct a high-order polynomial of the solution variable. As a result, the algorithm is relatively complex and not compact.

Since the TLBFS is actually a unique way to reconstruct the local flux, besides the FV scheme, other schemes also could be flexibly adopted in TLBFS to solve governing equations. From this motivation, the popular high-order FR scheme is combined with the TLBFS in this work. With the help of Chapman-Enskog (CE) analysis, an equivalent form of NSB equations, containing both macroscopic flow variables and microscopic particle distribution functions, has been derived from the lattice Boltzmann equations (LBE). The present governing equations have no second-order partial derivative term, so no extra techniques, such as the second procedure of Bassi and Rebay (BR2) [36], local DG (LDG) [37], hybridizable DG (HDG) [38], and so on, for discretization of the viscous terms are needed. In addition, no special techniques, such as projection or the AC method, are conducted to solve incompressible NS equations, since the present method is actually a weakly compressible (WC) model. Compared with the high-order off-lattice thermal flow solvers, the number of governing equations in the present solver is much less than that of off-lattice methods. Thus, the corresponding memory and computational time cost are reduced. Furthermore, the macroscopic physical boundaries rather than the kinematic boundaries are imposed by directly calculating the fluxes at the boundaries. In general, the present solver inherits the advantages of FR and TLBFS, such as high-order accuracy, low dissipation, well compactness, and low storage. For the convenience of presentation, hereafter the present solver is called FRTLBFS.

The remainder of this work is organized as follows. The macroscopic governing equations evolved from the thermal lattice Boltzmann model are given in Sec. II. A detailed description of the algorithm and remarks about FRTLBFS are provided in Sec. III. Then Sec. IV shows the numerical results and discussion of the present method. Conclusions can be found in Sec. V.

II. GOVERNING EQUATIONS

A. Thermal lattice Boltzmann model

The double-distribution function based thermal lattice Boltzmann model with the single relaxation time and the collision term in BGK approximation can be written as

$$f_\alpha(\mathbf{r}+\mathbf{e}_\alpha\delta_t, t+\delta_t) = f_\alpha(\mathbf{r}, t) + \frac{f_\alpha^{eq}(\mathbf{r}, t) - f_\alpha(\mathbf{r}, t)}{\tau_f}, \quad 0 \leqslant \alpha < q, \tag{1}$$

$$g_\alpha(\mathbf{r}+\mathbf{e}_\alpha\delta_t, t+\delta_t) = g_\alpha(\mathbf{r}, t) + \frac{g_\alpha^{eq}(\mathbf{r}, t) - g_\alpha(\mathbf{r}, t)}{\tau_g}, \quad 0 \leqslant \alpha < q, \tag{2}$$

where $\mathbf{r}$ represents a physical location; f_α and g_α are the particle distribution functions for the velocity and temperature fields, respectively; f_α^{eq} and g_α^{eq} are their corresponding equilibrium states; $\mathbf{e}_\alpha$ is the discrete velocity; τ_f and τ_g are the relaxation times for the velocity and temperature fields, respectively; δ_t is the streaming time step; and q is the number of discrete velocities. In this work, two-dimensional (2D) problems are considered, and the D2Q9 lattice velocity model is adopted:

$$\mathbf{e}_\alpha = \begin{cases} (0,0), & \alpha = 0 \\ (\pm 1, 0), (0, \pm 1), & \alpha = 1-4. \\ (\pm 1, \pm 1), & \alpha = 5-8 \end{cases} \tag{3}$$

The equilibrium distribution function is defined as

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

$$g_\alpha^{eq} = T\omega_\alpha\left[1 + \frac{\mathbf{e}_\alpha\cdot\mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha\cdot\mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}\cdot\mathbf{u}}{2c_s^2}\right], \tag{5}$$

where $c_s = 1/\sqrt{3}$ is the lattice sound speed in the LBM. The weight coefficient ω_α for the D2Q9 model is

$$\omega_\alpha = \begin{cases} 1/9, & \alpha = 0 \\ 4/9, & \alpha = 1-4. \\ 1/36, & \alpha = 5-8 \end{cases} \tag{6}$$

The macroscopic fluid density ρ, velocity $\mathbf{u}$, and temperature T are obtained by

$$\rho = \sum_{\alpha} f_\alpha, \quad \rho\mathbf{u} = \sum_{\alpha} \mathbf{e}_\alpha f_\alpha, \quad T = \sum_{\alpha} g_\alpha. \tag{7}$$

The pressure can be calculated from the equation of state:

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

B. Chapman-Enskog expansion analysis

By using the Chapman-Enskog expansion analysis [39], the mass, momentum, and energy equations for incompressible flows based on LBM evolution equations can be recovered. First, by performing the Taylor-series expansion to Eqs. (1) and (2), the corresponding equations can be written as

$$\left(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\right) f_\alpha + \frac{\delta_t}{2}\left(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\right)^2 f_\alpha + \frac{1}{\tau_f \delta_t}\left(f_\alpha - f_\alpha^{eq}\right) + O\left(\delta_t^2\right) = 0, \tag{9}$$

$$\left(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\right) g_\alpha + \frac{\delta_t}{2}\left(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\right)^2 g_\alpha + \frac{1}{\tau_g \delta_t}\left(g_\alpha - g_\alpha^{eq}\right) + O\left(\delta_t^2\right) = 0. \tag{10}$$

Then applying the multiscale expansion, the density and temperature distribution functions of f_α and g_α, the temporal derivative, and the spatial derivative can be expanded, respectively, as

$$f_\alpha = f_\alpha^{(0)} + \varepsilon f_\alpha^{(1)} + \varepsilon^2 f_\alpha^{(2)}, \tag{11a}$$

$$g_\alpha = g_\alpha^{(0)} + \varepsilon g_\alpha^{(1)} + \varepsilon^2 g_\alpha^{(2)}, \tag{11b}$$

$$\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_0} + \varepsilon^2 \frac{\partial}{\partial t_1}, \tag{11c}$$

$$\nabla_r = \varepsilon \nabla_{r1}, \tag{11d}$$

where ε is a small parameter proportional to the Knudsen number. Substituting Eq. (11) into Eqs. (9) and (10) and introducing conservation laws for mass, momentum, and energy, the following equations can be obtained:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{P} = 0, \tag{12}$$

$$\frac{\partial \rho\mathbf{u}}{\partial t} + \nabla \cdot \Pi = 0, \tag{13}$$

$$\frac{\partial T}{\partial t} + \nabla \cdot \mathbf{Q} = 0, \tag{14}$$

where $\mathbf{P}$ is the mass flux vector, Π is the momentum flux tensor, and $\mathbf{Q}$ is the thermal flux vector. They are defined as

$$\mathbf{P} = \sum_{\alpha=0}^{q} \mathbf{e}_\alpha f_\alpha^{eq}, \tag{15}$$

$$\Pi_{\beta\gamma} = \sum_{\alpha=0}^{q} (\mathbf{e}_\alpha)_\beta (\mathbf{e}_\alpha)_\gamma \left[f_\alpha^{eq} + \left(1 - \frac{1}{2\tau_f}\right) f_\alpha^{\text{neq}}\right], \tag{16}$$

$$\mathbf{Q} = \sum_{\alpha=0}^{q} \left(\mathbf{e}_\alpha \left[g_\alpha^{eq} + \left(1 - \frac{1}{2\tau_g}\right) g_\alpha^{\text{neq}}\right]\right). \tag{17}$$

In the above equations, β and γ represent the x and y directions, respectively. f_α^{neq} and g_α^{neq} represent the nonequilibrium distribution function, and they can be computed by the following equations:

$$f_\alpha^{\text{neq}} = f_\alpha - f_\alpha^{eq} \approx \varepsilon f_\alpha^{(1)} = -\tau_f \delta_t \left(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\right) f_\alpha^{eq}, \tag{18}$$

$$g_\alpha^{\text{neq}} = g_\alpha - g_\alpha^{eq} \approx \varepsilon g_\alpha^{(1)} = -\tau_g \delta_t \left(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\right) g_\alpha^{eq}. \tag{19}$$

Using relationships (3)–(7) and (15)–(19), Eqs. (12)–(14) can recover the following differential equations, which are identical to the macroscopic governing equations of incompressible thermal flows in a weakly compressible form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = 0, \tag{20}$$

$$\frac{\partial \rho\mathbf{u}}{\partial t} + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) = -\nabla p + \nu \nabla \cdot [\nabla \rho\mathbf{u} + (\nabla \rho\mathbf{u})^T], \tag{21}$$

$$\frac{\partial T}{\partial t} + \nabla \cdot (T\mathbf{u}) = \nabla \cdot (\kappa \nabla T), \tag{22}$$

where the kinematic viscosity coefficient ν and the thermal diffusion coefficient κ are in connection with the relaxation times correspondingly:

$$\nu = \left(\tau_f - \tfrac{1}{2}\right) c_s^2 \delta_t, \tag{23}$$

$$\kappa = \left(\tau_g - \tfrac{1}{2}\right) c_s^2 \delta_t. \tag{24}$$

One can refer to Ref. [30] for more detailed derivations. Equations (12)–(14) can be regarded as other expressions of Eqs. (20)–(22), respectively, and they can be written in the following vector form:

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = 0, \tag{25}$$

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ T \end{pmatrix}, \quad \mathbf{F}_x = \begin{pmatrix} P_x \\ \Pi_{xx} \\ \Pi_{yx} \\ Q_x \end{pmatrix}, \quad \mathbf{F}_y = \begin{pmatrix} P_y \\ \Pi_{xy} \\ \Pi_{yy} \\ Q_y \end{pmatrix}, \tag{26}$$

where u and v are the components of $\mathbf{u}$ in the x and y directions, respectively. Unlike the conventional NSB equations, which only contain macroscopic variables, the convection terms in Eqs. (12)–(14) consist of equilibrium and nonequilibrium distribution functions. In addition, the second-order viscous term in Eq. (21) disappears in Eq. (13), and it is contained in the convection term. The buoyancy force $\mathbf{F}_E$ can be directly added to the right-hand side of Eq. (25). Using the Boussinesq approximation, $\mathbf{F}_E$ can be defined as

$$\mathbf{F}_E = \begin{pmatrix} 0 \\ 0 \\ -\rho g \beta (T - T_m) \\ 0 \end{pmatrix}, \tag{27}$$

where g represents the acceleration due to gravity, β is the thermal expansion coefficient, and T_m is the average temperature.

III. FLUX RECONSTRUCTION THERMAL LATTICE BOLTZMANN FLUX SOLVER

A. Flux reconstruction scheme

Since the multidimensional FR scheme can be extended from the one-dimensional (1D) version by tensor production, let us consider the one-dimensional conservation law,

$$\frac{\partial u}{\partial t}+\frac{\partial f(u)}{\partial x}=0, \tag{28}$$

in domain Ω. Firstly, partition the domain into m nonoverlapping elements,

$$\Omega=\bigcup_{n=1}^{m}\Omega_n, \tag{29}$$

and $\Omega_n=[x_n,x_{n+1}]$ for $n=1,\dots,m$. In order to facilitate the calculation, each element is mapped to a standard element, $I=[-1,1]$. According to a mapping function $M_n(\xi)$, the coordinate transformation should be carried out between the global coordinate system x in the physical domain and the local coordinate system ξ of the standard element. This computational space is discretized with $(p+1)^d$ solution points, and $2d(p+1)^{d-1}$ flux points, placed at the edges of the subdomain, where d represents the dimension. The solution and flux point locations for multidimensional elements are typically determined using a tensor grid of a 1D quadrature. Thereafter, the evolution of u within each Ω_n can be determined with the following transformed conservation equation,

$$\frac{\partial \hat{u}}{\partial t}+\frac{\partial \hat{f}}{\partial \xi}=0, \tag{30}$$

where $\hat{u}=J_n u_n(M_n(\xi),t)$ and $\hat{f}=J_n f_n(M_n(\xi),t)$; J_n denotes the Jacobian of the mapping function $M_n(\xi)$. With this domain set up, our task is to construct a continuous flux polynomial from the discontinuous segments. At the first stage, the flux interpolation polynomials can be constructed by the approximation of the exact values,

$$\hat{f}^{\delta}=\sum_{i=1}^{N}\hat{f}_i^{\delta}l_i(\xi), \tag{31}$$

where N is the number of the solution points, and $l_i(X)$ is the Lagrange interpolation basis, which is defined as

$$l_i(X)=\prod_{s=1,s\neq i}^{N}\left(\frac{X-X_s}{X_i-X_s}\right),\quad i=1,2,\dots,N. \tag{32}$$

The leap at each interface in the aforementioned flux construction for each element is unreasonable. Revision of the discontinuous flux polynomials and consideration of common fluxes at element borders are necessary in order to reconstruct continuous flux polynomials. The common fluxes at the left and right boundaries of the element n in ξ direction are $(\hat{f}^{\delta})^{\mathrm{com}}_{i-\frac{1}{2}}$ and $(\hat{f}^{\delta})^{\mathrm{com}}_{i+\frac{1}{2}}$, respectively. Defining the polynomial $(\hat{f}^{\delta})^{\mathrm{c}}_i(\xi)$ as the continuous flux function of the element n in the ξ direction, the correction to the discontinuous flux function is made by

$$\begin{aligned}(\hat{f}^{\delta})^{\mathrm{c}}_i(\xi)=(\hat{f}^{\delta})_i(\xi)&+\big[(\hat{f}^{\delta})^{\mathrm{com}}_{i-\frac{1}{2}}-(\hat{f}^{\delta})_i(-1)\big]g_L(\xi)\\&+\big[(\hat{f}^{\delta})^{\mathrm{com}}_{i+\frac{1}{2}}-(\hat{f}^{\delta})_i(1)\big]g_R(\xi),\end{aligned} \tag{33}$$

where $(\hat{f}^{\delta})_i(-1)$ and $(\hat{f}^{\delta})_i(1)$, are respectively, the values of the discontinuous flux polynomial at the left and right element boundaries, and they can be determined by Eq. (31). g_L and g_R are, respectively, the correction functions related to the left and the right end points of the element, and the following conditions should be satisfied:

$$g_L(-1)=1,\quad g_L(1)=0,\quad g_R(-1)=0,\quad g_R(1)=1. \tag{34}$$

The choice of correction function makes the FR scheme have different properties on the accuracy and stability. In the present study, we use the following expression for the right and left Radau polynomials, respectively. Thus, following the analysis of Huynh [7], a particular nodal DG scheme is recovered.

$$g_L=\frac{(-1)^k}{2}(L_k-L_{k+1}), \tag{35}$$

$$g_R=\frac{1}{2}(L_k+L_{k+1}), \tag{36}$$

where L_k is the k degree Legendre polynomial and k is the order of the solution polynomial within each element. The present solver with the solution polynomial of degree k ($k=N-1$) is called p^k FRTLBFS.

Finally, the flux derivative can be discretized as

$$\begin{aligned}\frac{d(\hat{f}^{\delta})^{\mathrm{c}}_i}{d\xi}=\frac{d(\hat{f}^{\delta})_i}{d\xi}&+\big[(\hat{f}^{\delta})^{\mathrm{com}}_{i-\frac{1}{2}}-(\hat{f}^{\delta})_i(-1)\big]\frac{dg_{\mathrm{LB}}}{d\xi}\\&+\big[(\hat{f}^{\delta})^{\mathrm{com}}_{i+\frac{1}{2}}-(\hat{f}^{\delta})_i(1)\big]\frac{dg_{\mathrm{RB}}}{d\xi}.\end{aligned} \tag{37}$$

More details of the FR scheme can be found in Ref. [7].

B. Evaluation of fluxes at solution and flux points by TLBFS

According to Eqs. (12)–(17), the fluxes of the present scheme should be calculated by f_α^{eq}, g_α^{eq}, f_α^{neq}, and g_α^{neq}. It is known that two types of flux are needed in the FR scheme. One is the original fluxes stored at the solution points to construct the discontinuous flux polynomials, and the other is the common fluxes stored at the flux points to correct the discontinuous flux polynomials. In the most previous FR solvers based on conventional NS equations, the common inviscid fluxes at an interface $(\hat{\mathbf{F}}_x)^{\mathrm{com}}_{i\pm\frac{1}{2}}$ are calculated by utilizing the Riemann solver [40], which is constructed from the mathematical approach. In the present work, both fluxes are simultaneously calculated by LBFS, which is physical evaluation of numerical fluxes. Following the work of Wang *et al.* [34], the nonequilibrium distribution function f_α^{neq} and g_α^{neq} in Eqs. (18) and (19), respectively, can be approximated by

$$f_\alpha^{\mathrm{neq}}(\mathbf{r},t) = -\tau_f\left[f_\alpha^{eq}(\mathbf{r},t) - f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)\right] + O\left(\delta_t^2\right), \tag{38}$$

$$g_\alpha^{\mathrm{neq}}(\mathbf{r},t) = -\tau_g\left[g_\alpha^{eq}(\mathbf{r},t) - g_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)\right] + O\left(\delta_t^2\right), \tag{39}$$

where $\mathbf{r}$ is the physical location and t is the physical time after streaming. $f_\alpha^{eq}(\mathbf{r},t)$ and $g_\alpha^{eq}(\mathbf{r},t)$ are the equilibrium distribution functions at the solution or flux points after streaming, while $f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$ and $g_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$ are the equilibrium distribution functions at its surrounding nodes $\mathbf{r}-\mathbf{e}_\alpha\delta_t$ before streaming, respectively. The lattice spacing δ_x is equal to the streaming time step δ_t.

Since the macroscopic variables at the solution points are known at the time level $t-\delta_t$, it is convenient to construct a high-order macroscopic variable interpolation polynomial similar to Eq. (31) directly in the FR framework:

$$\varphi(\xi,\eta) = \sum_{j=1}^{N}\sum_{i=1}^{N} \boldsymbol{\Psi}_{i,j}[l_i(\xi)l_j(\eta)], \tag{40}$$

where $\boldsymbol{\Psi}$ can be fluid density ρ, velocity $\mathbf{u}$, or temperature T. Then the macroscopic variables at the surrounding nodes of a solution or flux point are achieved. The process is briefly depicted in Fig. 3. Note when the surrounding nodes of a flux point are located at the cell interface, their macroscopic variables ρ, $\mathbf{u}$, or T can be obtained by averaging the values on the left and right sides. Thereafter, $f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$ and $g_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$ can be calculated by Eqs. (4) and (5), respectively. Furthermore, the macroscopic variables after streaming are given by

$$\rho(\mathbf{r},t) = \sum_{\alpha=0}^{q} f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t), \tag{41}$$

$$\rho(\mathbf{r},t)\mathbf{u}(\mathbf{r},t) = \sum_{\alpha=0}^{q} f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)\mathbf{e}_\alpha, \tag{42}$$

$$T(\mathbf{r},t) = \sum_{\alpha=0}^{q} g_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t). \tag{43}$$

The derivation details of Eqs. (41)–(43) can be found in Ref. [34]. Thereafter, $f_\alpha^{eq}(\mathbf{r},t)$ and $g_\alpha^{eq}(\mathbf{r},t)$ can be determined via Eqs. (4) and (5), respectively. Once $f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$, $g_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$, $f_\alpha^{eq}(\mathbf{r},t)$, and $g_\alpha^{eq}(\mathbf{r},t)$ are available, $f_\alpha^{\mathrm{neq}}(\mathbf{r},t)$ and $g_\alpha^{\mathrm{neq}}(\mathbf{r},t)$ can be calculated via Eqs. (38) and (39), respectively. Ultimately, the fluxes at the solution and flux points are evaluated using Eqs. (15)–(17).

C. Implementation of boundary conditions

Since Eqs. (12)–(14) are a particular form of Eqs. (20)–(22), the boundary fluxes can be obtained from the conventional NSB equations. The following equivalence can be established:

$$\begin{pmatrix} P_x \\ \Pi_{xx} \\ \Pi_{yx} \\ Q_x \end{pmatrix} = \begin{pmatrix} \rho u \\ \rho u^2 + p - \tau_{11} \\ \rho uv - \tau_{12} \\ Tu - \kappa\frac{\partial T}{\partial x} \end{pmatrix},$$

$$\begin{pmatrix} P_y \\ \Pi_{xy} \\ \Pi_{yy} \\ Q_y \end{pmatrix} = \begin{pmatrix} \rho v \\ \rho uv - \tau_{21} \\ \rho v^2 + p - \tau_{22} \\ Tv - \kappa\frac{\partial T}{\partial y} \end{pmatrix}, \tag{44}$$

where the pressure p is computed by Eq. (9). τ_{ij} is the viscous stress defined as

$$\tau_{ij} = \begin{cases} \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), & i \neq j \\ \mu\left(2\frac{\partial u_i}{\partial x_i} - \frac{2}{3}\nabla\cdot\mathbf{u}\right), & i = j \end{cases}. \tag{45}$$

For a no-slip isothermal wall, the velocity and temperature are known. The density or pressure can be determined by interpolation while the derivatives in Eq. (44) can be calculated by the FR method. For a no-slip adiabatic wall, the only difference is $\partial T/\partial n = 0$, and the temperature of the boundary can be determined by interpolation. It is noted that the transformation from computational space to physical space is required when computing the derivatives. In this way, the fluxes of the flux points located on the physical boundary face are obtained from the macroscopic physical quantities.

D. Computational sequence

The computational sequence for the present high-order FRTLBFS is summarized as follows:

(1) Specify a streaming distance δ_x ($\delta_t = \delta_x$), make sure the virtual lattice velocity points of a solution point in the D2Q9 model are located in their own cell, while those of a flux point are within either the left cell or the right cell of the interface. The single relaxation parameters τ_f and τ_g are then calculated using Eqs. (5) and (6), respectively. Transform the physical location of the virtual lattice velocity point to the computational location in a standard element. Calculate the values of the Lagrangian interpolation basis at the virtual lattice velocity points and flux points with Eq. (32).

(2) Reconstruct the macroscopic flow variables at the surrounding positions $\mathbf{r}-\mathbf{e}_\alpha\delta_t$ of solution and flux points with Eq. (40). Calculate $f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$ and $g_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t, t-\delta_t)$ using Eqs. (4) and (5), respectively; then calculate the macroscopic variables at the solution and flux points after streaming via Eqs. (41)–(43). Calculate $f_\alpha^{eq}(\mathbf{r},t)$ and $f_\alpha^{\mathrm{neq}}(\mathbf{r},t)$ using Eqs. (4) and (38), respectively. Calculate $g_\alpha^{eq}(\mathbf{r},t)$ and $g_\alpha^{\mathrm{neq}}(\mathbf{r},t)$ using Eqs. (5) and (39), respectively. The fluxes at the solution points and the common fluxes at the flux points are obtained via Eqs. (15)–(17).

(3) Construct the discontinuous flux polynomials, i.e., Eq. (31), using the fluxes at the solution points. Reconstruct discontinuous fluxes at the interfaces of the cells.

(4) Construct the continuous flux polynomial, i.e., Eq. (33), using the common fluxes at the flux points and the correction functions g_L and g_R, i.e., Eqs. (35) and (36). Then, the derivatives of the fluxes at the solution points can be discretized by Eq. (37).

(5) Calculate the external force $\mathbf{F}_E$, i.e., Eq. (27).

(6) Choose appropriate time integration shemes for time marching. In the present work, the third-order explicit strong stability preserving Runge-Kutta (SSPRK) scheme is adopted for unsteady cases, while the first-order explicit Euler scheme is adopted for steady cases. The time step of cell i is determined by the Courant-Friedrichs-Lewy (CFL) condition:

$$\Delta t_i = \sigma \frac{\sqrt{V_i}}{\Lambda_i}/(2N+1), \tag{46}$$

where σ, V_i, and Λ_i are the CFL number, the volume of cell i, and the convective spectral radius of cell i, respectively.

(7) Repeat steps (2)–(6) until the converged solution or the specified time is reached.

E. Remarks

Recently, we reported the work using the FR scheme to solve the double-distribution discrete velocity Boltzmann equation (DVBE) for simulation of incompressible thermal flow, named FRTLBM [41]. Except for the same spatial discretization scheme, i.e., FR, the present work is quite different from our previous work in the following aspects:

(1) The governing equations in FRTLBM are DVBE while those in FRTLBFS are the particular form of Navier-Stokes–Boussinesq (NSB) equations. The convection term of DVBE is linear while that of NSB equations is nonlinear. For the 2D problem and the D2Q9 model, the number of double-distribution DVBE is 18 while that of NSB equations is four. The DVBE will produce stiffness due to the right-side collision term when Re or Ra is very large while the NSB equations will not.

(2) The common flux in FRTLBM is the Roe scheme while that in FRTLBFS is the lattice Boltzmann scheme. The Roe scheme is a pure numerical flux while LBFS is more physical since it involves the evolution of space and time and the solution at the cell interface also satisfies the governing equations.

(3) A particular time advance scheme should be adapted in FRTLBFS in order to overcome the stiffness of DVBE, e.g., the second-order implicit-explicit Runge-Kutta scheme (IMEXRK). For FRTLBFS, any time integration scheme can be adapted and the explicit Euler scheme is adequate which is efficient for the steady problem.

(4) In the present method, only the macroscopic boundary is considered while in FRTLBFS, the kinematic boundary should be imposed. However, it is difficult to construct a high-order and stable kinematic boundary implementation method.

IV. NUMERICAL RESULTS AND DISCUSSION

In this section, four typical incompressible thermal flow problems, including the porous plate problem, natural convection in a square cavity, unsteady natural convection in a tall cavity, and thermal lid-driven cavity flow, are used to test the comprehensive performance of the present FRTLBFS. In these cases, two dimensionless characteristic parameters for thermal flows, i.e., the Prandtl number Pr and the Rayleigh number Ra, are defined as

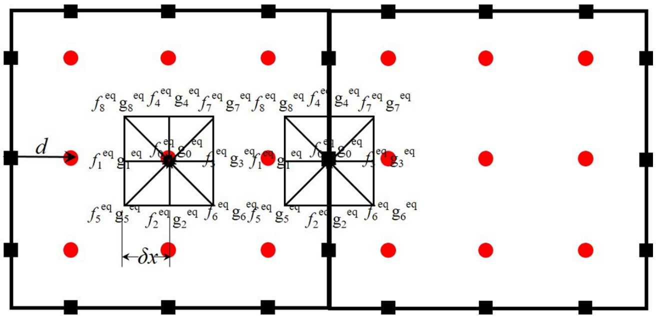


FIG. 1. Local reconstruction of thermal lattice Boltzmann solution at solution and flux points.

$$\text{Pr} = \frac{\nu}{\kappa}, \tag{47}$$

$$\text{Ra} = \frac{g\beta \Delta T L^3}{\kappa \nu}, \tag{48}$$

where L and ΔT are the characteristic length and the characteristic temperature difference of the system, respectively. The lattice spacing δ_x can be selected as $\gamma d_{\min}$, where γ is a control parameter with a value from 0 to 1, and $d_{\min}$ is the smallest physical distance between the first solution point and the corresponding left-side flux point among all the cells, as presented in Fig. 1. Results obtained for these cases are compared with those of the existing high-order thermal flow solvers and also the available numerical and experimental results. All the present computations are performed on a computer with an Intel Xeon Platinum 8168 CPU, 2.70 GHz, with a 64-bit operating system. In all the steady simulations, the convergence criterion is set to be

$$\text{Error} = \max\,(V_{\text{error}}, T_{\text{error}}) < 1\times 10^{-8}, \tag{49}$$

where V_{error} and T_{error} are, respectively, the relative errors of the velocity and temperature fields, which are defined as

$$V_{\text{error}} = \frac{\sqrt{\sum_{N_{\text{total}}} \left[\left(u_x^{t+\Delta t} - u_x^t\right)^2 + \left(u_y^{t+\Delta t} - u_y^t\right)^2\right]}}{\sqrt{\sum_{N_{\text{total}}} \left[\left(u_x^{t+\Delta t}\right)^2 + \left(u_y^{t+\Delta t}\right)^2\right]}}, \tag{50}$$

$$T_{\text{error}} = \frac{\sqrt{\sum_{N_{\text{total}}} (T^{t+\Delta t} - T^t)^2}}{\sqrt{\sum_{N_{\text{total}}} (T^{t+\Delta t})^2}}, \tag{51}$$

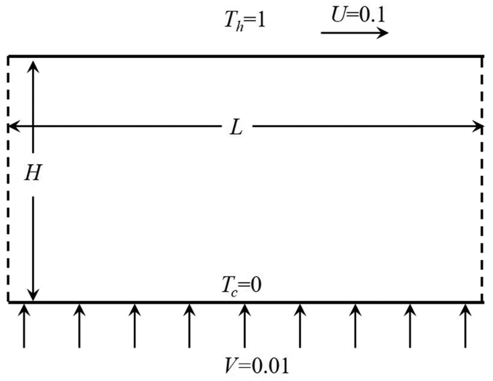


FIG. 2. Schematic of the porous plate problem.

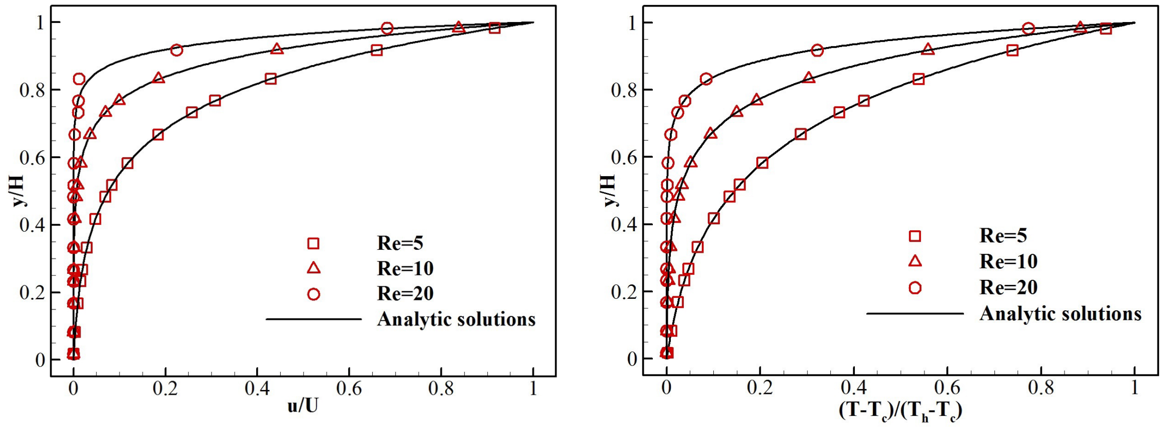


FIG. 3. The comparison of the x component of velocity vector (left) and temperature (right) profiles of the porous plate problem for Re = 5, 10, and 20.

where N_{total} is the total number of solution points. The integration quantities in all the test cases are approximated using Gaussian numerical quadrature, and the solution points coincide with the quadrature points.

A. Porous plate problem

First of all, the porous plate problem, which has analytical solutions for both velocity and temperature, is solved here to test the accuracy of the present high-order method [41,35]. As depicted in Fig. 2, the flow is sheared between two porous plates, while a similar fluid is injected normally from the bottom plate to the upper plate and withdrawn at the same rate. The steady-state governing equations can be given as

$$V\frac{\partial u}{\partial y} = \nu\frac{\partial^2 u}{\partial y^2},$$
$$\frac{\partial p}{\partial y} = g\beta(T - T_m), \tag{52}$$
$$v_0\frac{\partial T}{\partial y} = \kappa\frac{\partial^2 T}{\partial y^2},$$

where V is the injection velocity, T_h is the temperature for the upper plate, T_c is the temperature for the bottom plate, and $T_m = (T_h + T_c)/2$ is the average temperature. The exact solutions for horizontal velocity and temperature field in a steady state are obtained as

$$u^* = U\left(\frac{e^{\text{Re}(y/H)} - 1}{e^{\text{Re}} - 1}\right),$$
$$T^* = T_c + \left(\frac{e^{\text{Pr Re}(y/H)} - 1}{e^{\text{Pr Re}} - 1}\right)\Delta T, \tag{53}$$

where $\Delta T{=}T_h - T_c$ is the temperature difference between the upper and bottom plates, and $\text{Re} = VH/\nu$ is the Reynolds number associated with the injected velocity V and the distance between the plates H.

The computation is on a rectangle domain, whose size is $L{\times}H$ with $L = 2$ and $H = 1$ at $\text{Pr} = 0.71$ and $\text{Ra} = 100$. Periodic boundary conditions are applied to the left and right sides of the domain, while two constant isothermal temperature boundaries $T_h = 1.0$ and $T_c = 0.0$ are used at the top and bottom plates, respectively. The horizontal velocity is set to be 0.1 to satisfy the incompressible limit. Since the TLBFS has a second-order local accuracy, to avoid polluting the overall accuracy, the lattice spacing δ_x is set small enough

TABLE I. Orders of accuracy and L_2 errors of u and T on different meshes for the porous plate problem at Re = 10 by the p^2 and p^3 FRTLBFS.

		u		T	
Scheme	Mesh size	L_2	Order	L_2	Order
p^2	1/4	1.47×10^{-3}	–	8.07×10^{-3}	–
	1/8	2.82×10^{-4}	2.385	1.42×10^{-3}	2.506
	1/16	3.90×10^{-5}	2.853	2.14×10^{-4}	2.734
	1/20	1.93×10^{-5}	3.150	1.11×10^{-4}	2.949
p^3	1/2	1.68×10^{-3}	–	7.74×10^{-3}	–
	1/4	2.08×10^{-4}	3.014	7.53×10^{-4}	3.361
	1/6	4.78×10^{-5}	3.625	1.67×10^{-4}	3.721
	1/8	1.64×10^{-5}	3.714	5.07×10^{-5}	4.137

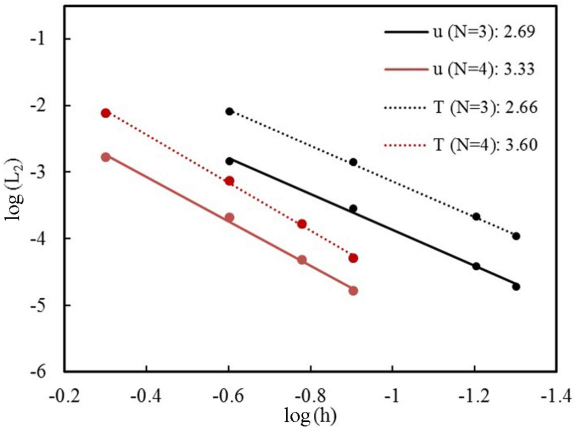


FIG. 4. Linearly fitted lines of L_2 errors of u and T on different meshes for the porous plate problem at Re = 10 by the p^2 and p^3 FRTLBFS (the base of log is 10).

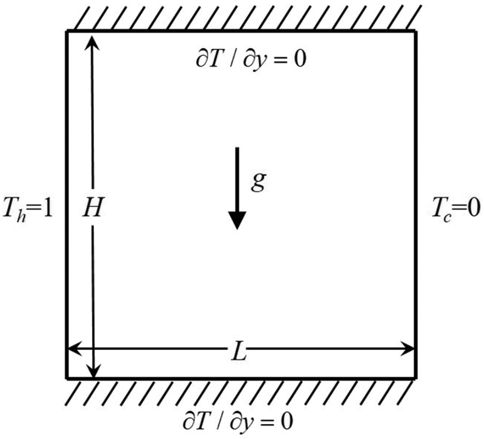


FIG. 5. Schematic of the natural convection in a square cavity.

and the control parameter γ is 0.1. All the meshes in this case are uniform. The cases with Re = 5, 10, and 20 are simulated first by the present p^3 scheme ($N = 4$) with fixed grids $N_x \times N_y = 2 \times 4$. The comparisons between the simulated and the analytic results of the x component of velocity and temperature profiles are shown in Fig. 3, respectively. From the figure, it can be seen that good agreements are achieved between the simulated and the analytic results.

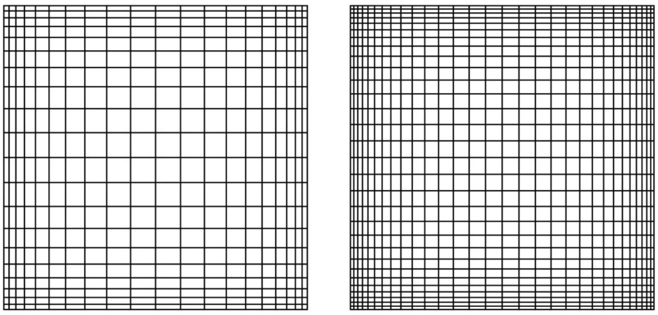


FIG. 6. Nonuniform mesh distribution with 20 × 20 (left) and 30 × 30 (right) for natural convection in a square cavity.

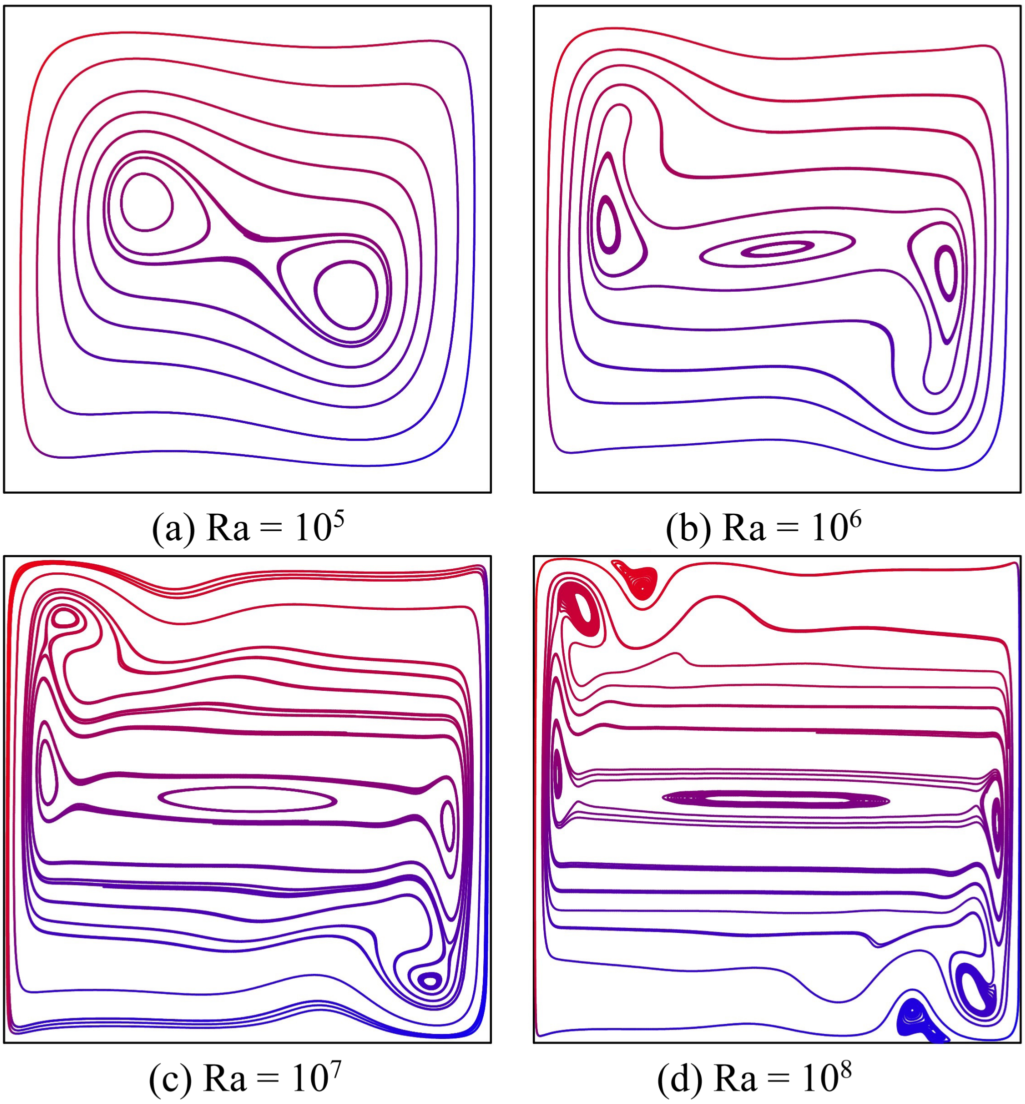
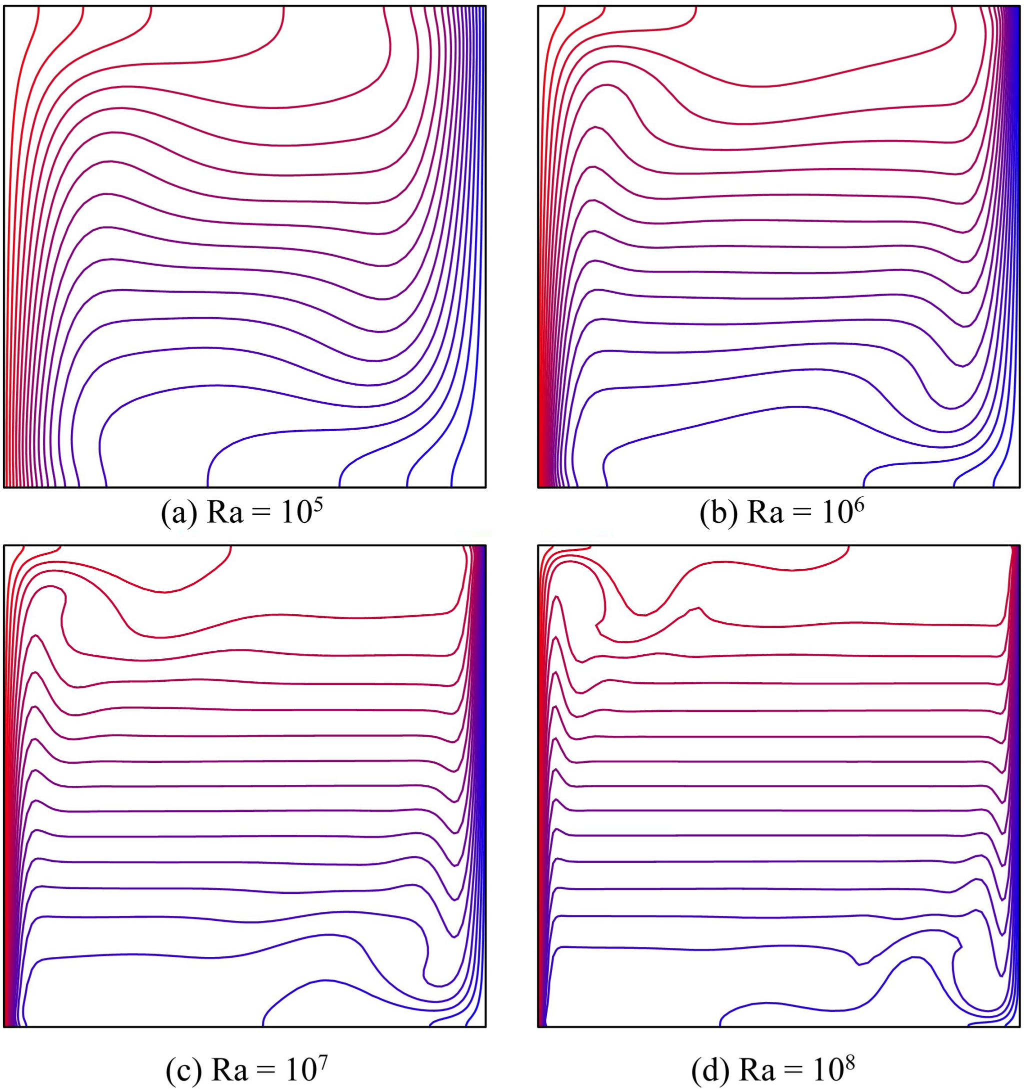


FIG. 7. Streamlines for natural convection in a square cavity by the p^3 FRTLBFS.


(a) Ra = 10^5

(b) Ra = 10^6

(c) Ra = 10^7

(d) Ra = 10^8

FIG. 8. Isotherms for natural convection in a square cavity by the p^3 FRTLBFS.

Then a h convergence study is performed to measure the convergence order of the present high-order method at Re = 10. The numerical error is quantified by using the L_2 norm, which is defined as

$$L_2\text{error} = \|\phi - \phi^*\|_2 = \sqrt{\frac{1}{N_x N_y N^2}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{m=1}^{N}\sum_{n=1}^{N}[(\phi)_{ijmn} - (\phi^*)_{ijmn}]^2}, \tag{54}$$

where N_x and N_y are the mesh numbers in the x and y directions, respectively; $\phi_{ijmn}(u \text{ or } T) = \phi(x_{ijmn}, y_{ijmn})$ and $\phi^*_{ijmn} = \phi^*(x_{ijmn}, y_{ijmn})$ are the numerical solution and the analytical solution given by Eq. (53), respectively. x_{ijmn} and y_{ijmn} are, respectively, the x and y coordinates of the solution point (m, n) in cell (i, j) corresponding to the solution point (ξ_m^S, η_n^S) in the $[-1, 1]\times[-1, 1]$ standard cell. In the test, four regular quadrilateral grids with the mesh sizes of $h = 1/4,\ 1/8,\ 1/16,$ and $1/20$ for the p^2 scheme $(N = 3)$ and $h = 1/2,\ 1/4,\ 1/6,$ and $1/8$ for the p^3 scheme $(N = 4)$ are used. The computed L_2 errors of velocity component u and temperature T and convergence orders are shown in Table I. From the table, it can be found that the L_2 errors of the p^3 scheme with less mesh are much smaller than that of the p^2 scheme. The slopes of the linearly fitted lines are presented in Fig. 4. It can be seen that the p^2 scheme can reach more than second-order accuracy, while the p^3 scheme can achieve more than third-order accuracy for both u and T. The results prove that the current method can expectedly achieve the designed high-order accuracy for both flow field and temperature field. The accuracy order can be easily adjusted by changing the order of cell local polynomials, i.e., the number of solution points.

B. Natural convection in a square cavity

Natural convection in a square cavity is frequently simulated to demonstrate the efficacy of numerical methods for thermal flows at a wide range of Rayleigh numbers [34,35].

TABLE II. Comparisons of indicated quantities obtained from the present simulations with the data from the literature of natural convection in a square cavity at Ra = 10^5, 10^6, 10^7, and 10^8.

Ra		DOF	$u_{\max}$	y	$v_{\max}$	x	$\overline{\mathrm{Nu}}$	$\mathrm{Nu}_{1/2}$	Nu_0
10^5	Wang *et al.* [42]	321^2	34.743	0.855	68.632	0.0659	4.521	4.521	4.522
	Dixit and Babu [44]	256^2	35.521	0.854	68.655	0.066	4.546	4.519	4.523
	p^2 FRTLBFS	90^2	34.666	0.853	68.529	0.0686	4.518	4.515	4.523
	p^3 FRTLBFS	80^2	34.650	0.854	68.696	0.0661	4.517	4.517	4.520
	p^2 error (%)		0.221	0.164	0.149	0.789	0.075	0.133	0.027
	p^3 error (%)		0.267	0.070	0.093	0.395	0.088	0.093	0.046
10^6	Wang *et al.* [42]	321^2	64.828	0.850	220.551	0.0378	8.825	8.826	8.827
	Dixit & Babu [44]	512^2	64.186	0.850	219.866	0.0371	8.653	8.507	8.805
	p^2 FRTLBFS	90^2	65.197	0.850	221.205	0.0378	8.841	8.852	8.846
	p^3 FRTLBFS	80^2	64.877	0.854	220.705	0.0379	8.826	8.824	8.827
	p^2 error (%)		0.569	0.059	0.297	0.106	0.181	0.303	0.211
	p^3 error (%)		0.076	0.494	0.070	0.397	0.018	0.017	0.006
10^7	Xu *et al.* [43]	2048^2	148.572	0.879	699.339	0.0213	16.522	16.524	16.524
	Ma *et al.* [41]	120^2	148.457	0.884	673.499	0.0198	16.426	16.501	16.504
	p^2 FRTLBFS	90^2	149.485	0.881	701.641	0.0225	16.588	16.613	16.754
	p^3 FRTLBFS	80^2	148.681	0.866	698.866	0.0204	16.597	16.652	16.458
	p^2 error (%)		0.615	0.227	0.329	5.634	0.395	0.537	1.392
	p^3 error (%)		0.073	1.547	0.068	4.225	0.455	0.773	0.401
10^8	Xu *et al.* [43]	2048^2	320.988	0.928	2223.466	0.0119	30.227	30.230	30.231
	Ma *et al.* [41]	120^2	329.841	0.928	2223.010	0.0118	30.298	30.253	30.194
	p^2 FRTLBFS	90^2	333.143	0.940	2227.920	0.0124	30.542	30.911	30.789
	p^3 FRTLBFS	80^2	326.910	0.929	2221.651	0.0115	30.311	30.350	30.587
	p^2 error (%)		3.787	1.326	0.200	4.202	1.045	2.252	1.849
	p^3 error (%)		1.845	0.161	0.082	3.361	0.278	0.395	1.180

The configuration of the problem is shown in Fig. 5. To be specific, the flow domain is a two-dimensional square cavity with $(x, y) \in [0, 1]\times[0, 1]$, i.e., $L = H = 1.0$. The adiabatic condition, i.e., $\partial T/\partial y = 0$, is set on the top and bottom walls, while isothermal conditions with fixed temperatures of $T_h = 1$ and $T_c = 0$ are, respectively, applied to the left and right walls. The no-slip boundary condition is applied on all the walls. In addition, a gravitational body force exists in the y direction. Initially, there is no velocity and the density is set as 1. The initial temperature is set as the reference temperature, i.e., $T_0 = (T_h + T_c)/2$. The Prandtl number is set as $\mathrm{Pr} = 0.71$, and four typical Rayleigh numbers, i.e., 10^5, 10^6, 10^7, and 10^8, are employed to investigate the proposed scheme. $\mathrm{Ma} = V_0/c_s$ is taken as 0.1 for all the cases to satisfy the incompressible limit, where $V_0 = \sqrt{g\beta\Delta TH}$ is the characteristic velocity of the flow. In order to capture the thin thermal boundary layer, the nonuniform mesh is used, and it is generated by the following transformation:

$$x = \frac{1}{2a}[a + \tanh(\theta\xi)], \quad y = \frac{1}{2a}[a + \tanh(\theta\eta)], \tag{55}$$

where $a = \tanh(\theta)$ and $\theta = 1.5$. The p^2 and p^3 FRTLBFS are tested in this subsection. The sizes of the mesh are 30×30 for the p^2 scheme and 20×20 for the p^3 scheme, and the corresponding DOF that refers to the number of solution points are 90×90 and 80×80, respectively. The nonuniform mesh distributions for the sizes of 20×20 and 30×30 are displayed in Fig. 6. The velocity field $\mathbf{u}$ is normalized by a reference velocity $U_0 = \kappa/H$. In the simulations, the Nusselt number Nu is used to evaluate the heat transfer rate. The

TABLE III. Grid independence study for natural convection in a square cavity at Ra = 10^6.

	Grids	$u_{\max}$	y	$v_{\max}$	x	$\overline{\mathrm{Nu}}$	$\mathrm{Nu}_{1/2}$	Nu_0
Wang *et al.* [42]	321^2	64.828	0.8498	220.551	0.03779	8.8247	8.8256	8.8272
p^2 FRTLBFS	25^2	64.897	0.8559	220.369	0.04000	8.8391	8.8484	8.8639
	30^2	64.877	0.8540	220.705	0.03794	8.8407	8.8523	8.8458
	35^2	64.750	0.8468	220.620	0.03788	8.8264	8.8148	8.8357
p^3 FRTLBFS	15^2	65.607	0.8543	220.729	0.03515	8.8349	8.8426	8.7990
	20^2	64.877	0.8540	220.705	0.03794	8.8263	8.8241	8.8267
	25^2	64.903	0.8470	220.565	0.03745	8.8270	8.8266	8.8255

TABLE IV. Comparisons of memory cost and computational time between FRTLBFS and FRTLBM in natural convection in a square cavity at Ra = 10^5.

Schemes	Grids	DOF	$\overline{\text{Nu}}$	Iteration steps	Virtual memory (megabytes)	Memory ratio	CPU time (s)	Time ratio
p^2 FRTLBFS	30^2	8100	4.5177	885000	4.3	21.18%	4906.24	43.38%
p^2 FRTLBM [41]	30^2	8100	4.5168	885000	20.3	–	11311	–
p^3 FRTLBFS	20^2	6400	4.5173	725000	2.8	18.06%	3568.01	62.95%
p^3 FRTLBM [41]	20^2	6400	4.5170	725000	15.5	–	5667.74	–

averaged Nusselt numbers over the whole computational domain and that along the line of $x = 0$ and $x = L/2$ are, respectively, defined as

$$\overline{\text{Nu}} = \frac{L}{\kappa \Delta T}\frac{1}{L^2}\oiint_{\Omega}\left(uT - \kappa\frac{\partial T}{\partial x}\right)d\Omega, \tag{56}$$

$$\text{Nu}_0 = \frac{L}{\kappa \Delta T}\frac{1}{L}\int_{x=0}\left(uT - \kappa\frac{\partial T}{\partial x}\right)dl, \tag{57}$$

$$\text{Nu}_{1/2} = \frac{L}{\kappa \Delta T}\frac{1}{L}\int_{x=L/2}\left(uT - \kappa\frac{\partial T}{\partial x}\right)dl. \tag{58}$$

Figures 7 and 8, respectively, show the streamlines and isotherms computed by the p^3 scheme. It can be concluded that as Ra increases, the isotherms in the cavity's center change from nearly vertical to horizontal, and the thermal boundary layers around the hot and cold walls' corners become thinner. The temperature boundary layers get smaller as Ra increases. All of these phenomena are consistent with those reported in the literature [41,32,34] whereas the present mesh size is smaller.

Table II shows the comparisons of the quantities $U_{\max}$, $V_{\max}$ together with their corresponding locations y and x and the averaged Nusselt numbers $\overline{\text{Nu}}$, $\text{Nu}_{1/2}$, Nu_0 for the four Ra. The quantity $U_{\max}$ indicates the maximum value of the x component of velocity on the vertical central line, while $V_{\max}$ represents the maximum value of the y component of velocity on the horizontal central line. The results of Wang *et al.* [42] using multiple-relaxation-time LBM (MRTLBM) with fine grids of 321×321 are chosen as the benchmark solution for moderate Ra (10^5 and 10^6), while the MRTLBM results of Xu *et al.* [43] obtained with extremely fine grids of 2048×2048 are chosen as the benchmark solution for high Ra (10^7 and 10^8). Additionally, the LBM results of Dixit and Babu [44] at moderate Ra and the results of high Ra obtained by the fourth-order flux reconstruction thermal lattice Boltzmann method (FRTLBM) [41] are also listed. All the quantities computed by the p^2 and p^3 FRTLBFS agree well with the reference data reported by Wang *et al.* [42], Dixit and Babu [44], Xu *et al.* [43], and our recently proposed FRTLBM [41], although the present DOF are much less than those using second-order methods in the references. Generally speaking, the p^3 scheme achieves less error than p^2, despite the fact that the p^2 scheme uses more DOF. The superiority of the high-order method is clearly presented from the comparisons. Then, a grid-dependent study of the natural convection problem at Ra = 10^6 is conducted on three different grids for the present p^2 and p^3 scheme, as listed in Table III. It can be proven that the grid resolution used in the present simulations is sufficient to obtain grid independence results.

Next, the computational efficiency, virtual memory cost, and stability at high Ra of the present method are compared with those of recently developed FRTLBM [41] with the same hardware and system. The same grids and the same CFL number 0.5 are fixed for the two methods. The comparisons of memory cost and computational time at Ra = 10^5 are given in Table IV. It is shown that the present third-order method takes 21.18% of memory cost and 43.38% of computational time needed by the third-order FRTLBM, and

TABLE V. Comparison of numerical stability between FRTLBFS and FRTLBM in natural convection in a square cavity at high Ra.

Schemes	Ra	Minimum grids	DOF
p^2 FRTLBFS	10^7	12^2	1296
	10^8	25^2	5625
p^2 FRTLBM [41]	10^7	14^2	1764
	10^8	27^2	6561
p^3 FRTLBFS	10^7	8^2	1024
	10^8	16^2	4096
p^3 FRTLBM [41]	10^7	10^2	1600
	10^8	20^2	6400

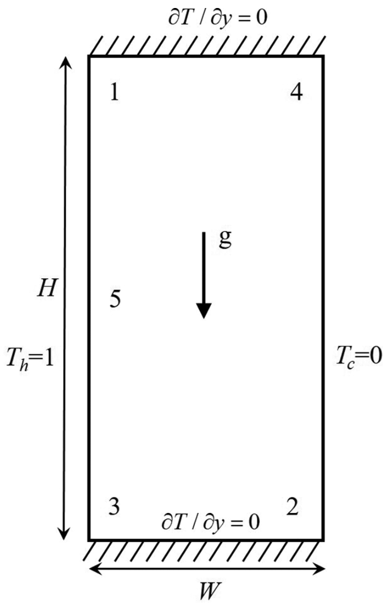


FIG. 9. Schematic of the unsteady natural convection in a tall cavity.

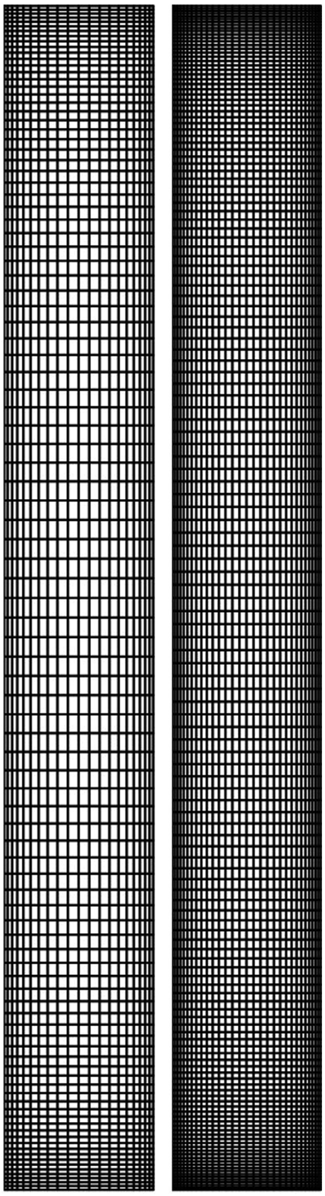


FIG. 10. Nonuniform mesh distribution with 20×100 (left) and 30×150 (right) for unsteady natural convection in a tall cavity.

the present fourth-order method takes 18.06% of memory cost and 62.95% of computational time needed by the fourth-order FRTLBM. For the present method, the most time-consuming procedure is the interpolation to obtain the macroscopic flow variables at the surrounding positions $\mathbf{r} - \mathbf{e}_\alpha \delta_t$ of the solution and flux points with Eq. (45). However, in FRTLBM, the number of governing equations, i.e., DVBE and the stored variables, i.e., macroscopic variables and particle distribution functions, is a lot more. Thus, the high efficiency and low memory storage of the present method is clearly demonstrated. Furthermore, what can be expected is that the present method extended to three dimensions is much more efficient and low storage than methods based on DVBE, since the number of three-dimensional (3D) DVBE and particle distribution functions significantly increases, especially for the DDF model. In order to compare the stability of the present method and FRTLBM, the minimum uniform mesh numbers required for each method to obtain a convergence solution at high Ra are compared in Table V. It can be seen that the two methods both can get a stable solution with relatively coarse mesh, and the present method needs fewer grids, which indicates that the present method is more stable.

TABLE VI. Coordinates for the time-history points shown in Fig. 9.

Point	x coordinate	y coordinate
1	0.1810	7.3700
2	0.8190	0.6300
3	0.1810	0.6300
4	0.8190	7.3700
5	0.1810	4.0000

C. Unsteady natural convection in a tall cavity

Herein, unsteady natural convection in a tall cavity is chosen to verify the accuracy of the present scheme for simulating transient flow and the heat transfer problem. The problem is based upon the geometry shown in Fig. 9, where W is the width and H the height of the enclosure. The enclosure aspect ratio is $A = H/W$ and takes on the value $A = 8$. Boundary and initial conditions are the same as those in the second example. This case was presented at the MIT conference on the computational fluid and solid dynamics [45] in 2001. It is an important problem in the field of heat transfer and convection flow phenomena. This challenging problem, as pointed out by Christon *et al.* [45], serves as a mechanism to test the performance of numerical methods to simulate complex physical mechanisms, such as vertical and horizontal boundary layers, several instability mechanisms, traveling waves in vertical boundary layers, and thermal instabilities along horizontal walls in particular, and all of them can interact strongly with internal wave dynamics. Above a critical Rayleigh number $\mathrm{Ra}_c \approx 3.1\times10^5$, this model problem demonstrates oscillatory transient flow behavior. In the following simulations, $\mathrm{Pr} = 0.71$ and $\mathrm{Ra} = 3.4\times10^5$ are used, which are consistent with the values in the literature. It is worth mentioning that a low-order scheme may be not adequate to capture flow unsteadiness, even on a very fine grid, as indicated by Bassi *et al.* [2]. The present third-order scheme is adopted. As suggested by Christon *et al.* [45], graded nonuniform meshes with approximately a 1:5 x-to-y ratio of elements starting with a grid of 20×100 and increasing the grid resolution to a size of 30×150 are adopted, which are depicted in Fig. 10. A third-order SSPRK time integration scheme is employed for such unsteady flow problem, and the CFL number is set as 0.8. To the best of our knowledge, this challenging test case is rarely simulated in the LBM community.

The data used in the transient and steady-state computations can be divided into three categories: point, wall, and global data. For all the time-dependent computations, the average value, peak-to-valley oscillation amplitude, and the period of oscillation for all the mandatory data are tabulated. The calculation of an average value is predicated on attaining a statistically stationary state with virtually constant period

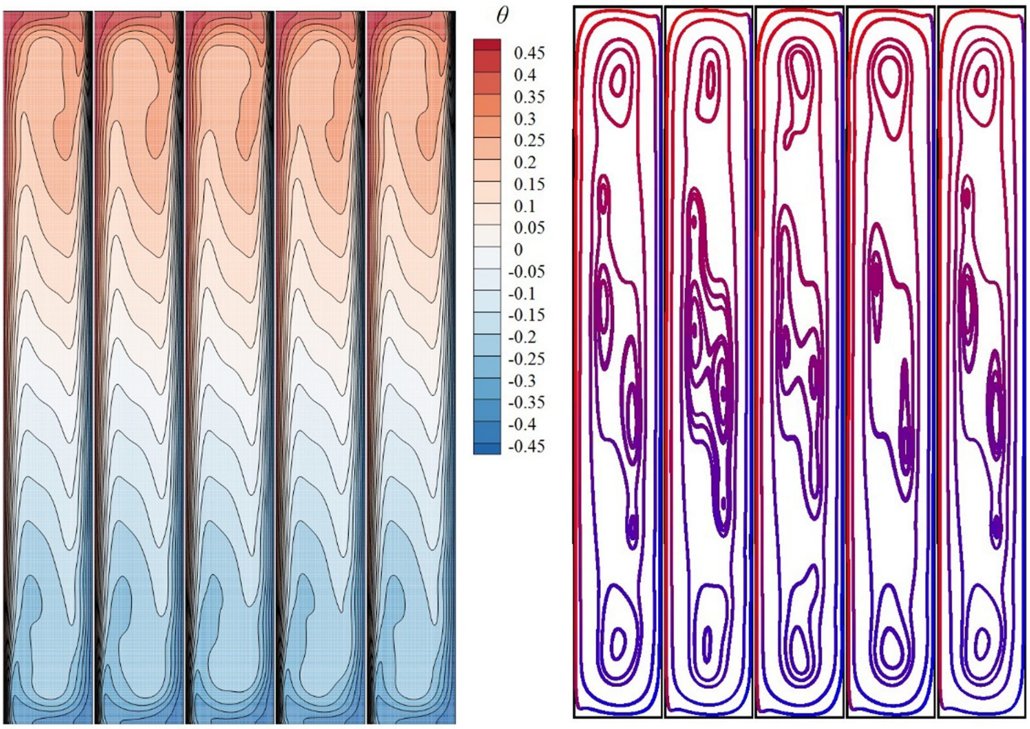


FIG. 11. Variation of temperature contours (left) and streamlines (right) at time instants 980, 980.86, 981.61, 982.47, and 983.43 ND time units during one total time period of 3.427 ND time units.

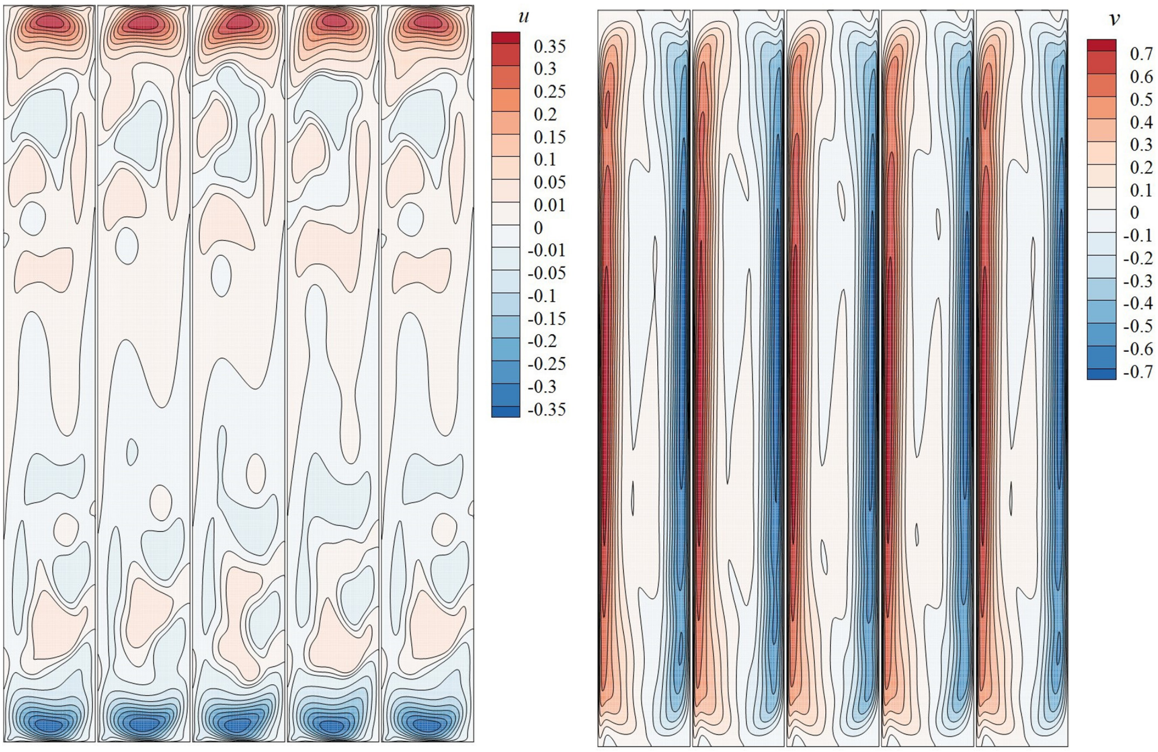


FIG. 12. Variation of u-velocity contours (left) and v-velocity contours (right) during one period at time instants 980, 980.86, 981.61, 982.47, and 983.43 ND time units.

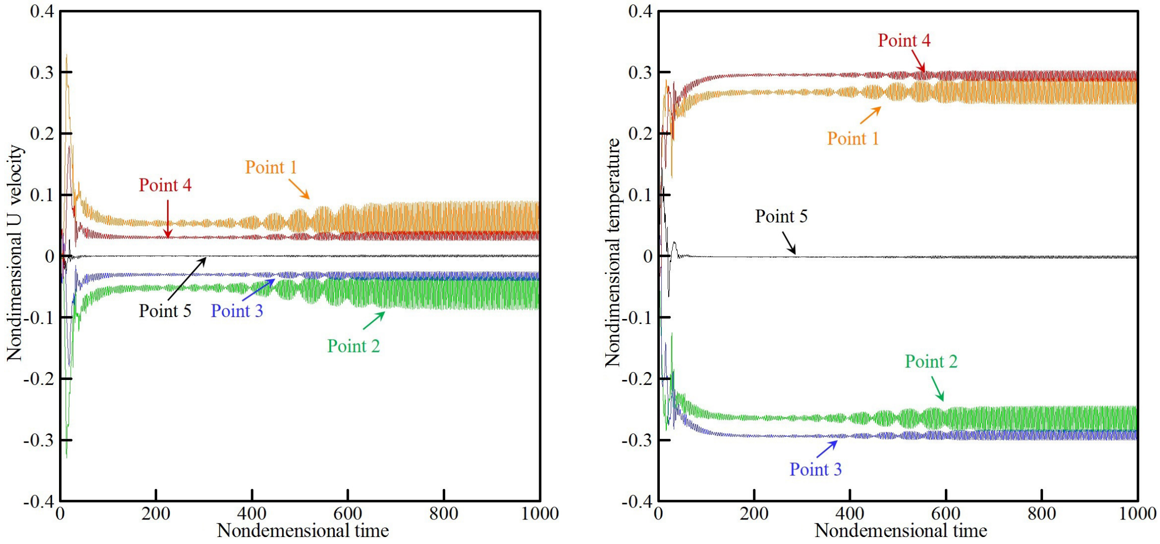


FIG. 13. Oscillation of ND x-component velocity values (left) and ND temperature values (right) at designated points 1–5 with reference to the ND time.

and amplitude. Time-history data are requested at the time-history points shown in Fig. 9, and they are identified in Table VI. The wall Nusselt numbers are also tabulated, which are defined as

$$\mathrm{Nu}(t)|_{x=0,W} = \frac{1}{H}\int_0^H \left|\frac{\partial\theta}{\partial x}\right|_{x=0,W} dy, \tag{59}$$

where θ is the nondimensional (ND) temperature, which is defined as

$$\theta = \frac{T - T_0}{T_h - T_c}. \tag{60}$$

The measure of the global average velocity is defined as

$$\hat{u}(t) = \sqrt{\frac{1}{2\Omega}\int_\Omega \mathbf{u}\cdot\mathbf{u}\,d\Omega}, \tag{61}$$

where Ω is the area of the enclosure. The velocity is normalized by the reference velocity $V_c = \sqrt{g\beta W\Delta T}$.

Figure 11 presents the variation of temperature contours and streamlines at time instants 980, 980.86, 981.61, 982.47, and 983.43 ND time units during one period of 3.427 ND time units. Variation of u-velocity contours and v-velocity contours during one period is shown in Fig. 12. The transient behavior of flow and temperature fields is clearly demonstrated, which is compared with Arpino *et al.* [46] and Balam and

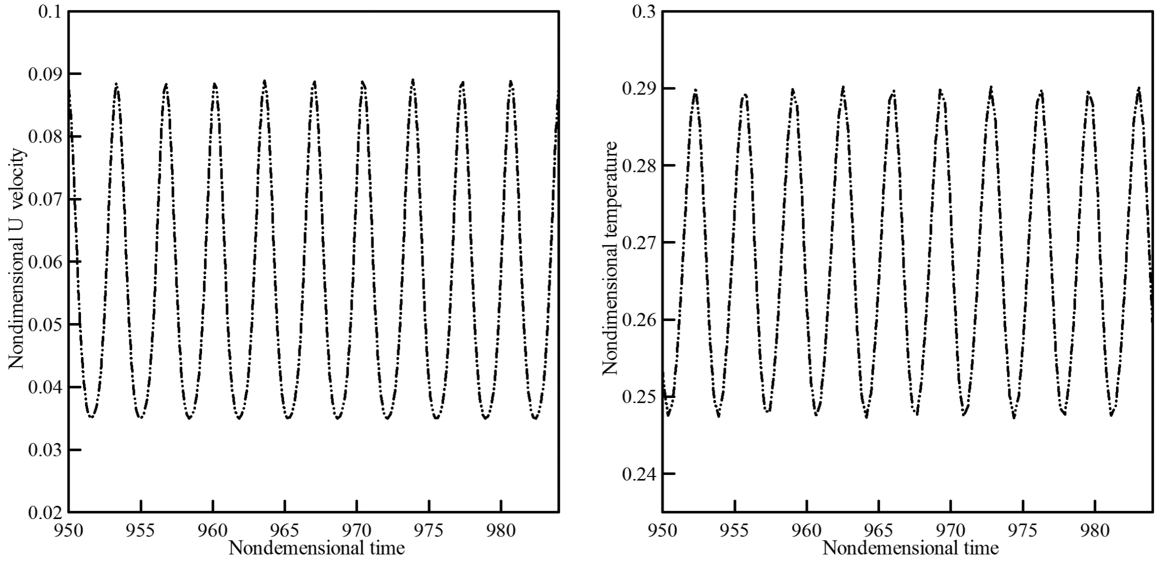


FIG. 14. Details of oscillation of ND x-component velocity values (left) and ND temperature values (right) at designated points 1 in 10 periods.

TABLE VII. Comparisons of point data at reference point 1 with literature results.

Scheme	$\bar{u}$	u'	$\bar{\theta}$	θ'
Present study with 20×100 mesh	0.056594	0.055131	0.267511	0.042829
Present study with 30×150 mesh	0.056237	0.054128	0.267425	0.042169
Paolucci from Christon *et al.* [45]	0.056453	0.051948	0.267803	0.040280
Xin and Le Quéré [48]	0.056345	0.054767	0.26548	0.042689
Bassi *et al.* [2]	0.056177	0.054505	0.26546	0.042443
Gjesdal *et al.* [49]	0.056356	0.054880	0.26561	0.042774
Arpino *et al.* [46]	0.058803	0.068647	0.266672	0.053328
Balam and Gupta [47]	0.057013	0.056342	0.265781	0.043944

Gupta [47] and shows an excellent match with their published contour diagrams. The symmetric character of the periodic flow behavior along the height of the cavity is another key aspect. Figure 13 shows the ND x-velocity component and ND temperature variation over time in correspondence to the reference points shown in Fig. 9, which indicates the oscillatory behavior of the solution and the occurrence of convergence. From these figures, it is observed that the stable oscillation behaviour is obtained after 700 ND time units approximately. In addition, the reference points 1 and 2, as well as points 3 and 4, behave symmetrically with respect to reference point 5. Correspondingly, the oscillation amplitude is much larger in correspondence of the four corners of the cavity (reference points from 1 to 4 in Fig. 9), but it is hardly appreciable at reference point 5. These observations are in good agreement with previous studies [45–47]. However, the time to obtain a stable oscillation behavior by the present scheme is longer than that achieved by the conventional macroscopic incompressible methods. The reason may be that the present method is a weakly compressible method and the divergence-free condition cannot be strictly satisfied. The details of oscillation of ND x-component velocity values and ND temperature values at designated points 1 in 10 periods are given in Fig. 14. Tables VII and VIII, respectively, summarize the point data and the wall and global data at reference point 1, and they are compared with the reference results obtained by high-order methods [2,45–49]. The parameters compared in these tables are the time averaged ND quantities including x-component velocity $\bar{u}$; temperature $\bar{\theta}$; hot wall Nusselt number $\overline{\mathrm{Nu}}$; velocity $\bar{\hat{u}}$; and the corresponding fluctuating quantities u', θ', Nu′, and $\hat{u}'$. It can be observed that the present results have a quite small deviation from those of other high-order methods, which infers that the coarse grid density is sufficient for the required accuracy. In addition, the p^1 scheme, whose theoretical accuracy should be second order, is also tested. Unfortunately, the transient phenomenon cannot be achieved even using a fine mesh with 40×200 elements. The superiority of the present high-order method is clearly demonstrated.

D. Thermal lid-driven cavity flow

The test cases above are all natural convection. In the last example, the present solver is validated by simulating mixed convection in a lid-driven cavity. The flow behavior in this problem is driven by both natural and forced convections. As presented in Fig. 15, the problem consists of a 2D square cavity, in which the top wall moves at a constant velocity U_0, while the other three walls remain stationary. The top and bottom walls are kept at constant cold ($T_c = 0$) and hot ($T_h = 1$) temperatures, respectively. On the right and left walls, a Neumann boundary condition for temperature with a normal gradient of zero is implemented. Apart from the Prandtl number, three other typical nondimensional parameters, i.e., Reynolds number, Grashof number, and Richardson number, are defined for this problem as follows:

$$\mathrm{Re} = \frac{U_0 L}{\nu}, \tag{62}$$

$$\mathrm{Gr} = \frac{g\beta(T_h - T_c)L^3}{\nu^2}, \tag{63}$$

$$\mathrm{Ri} = \frac{\mathrm{Gr}}{\mathrm{Re}^2}. \tag{64}$$

To keep in accordance with the setup in Refs. [32,50], the Grashof number and Prandtl number are fixed at $\mathrm{Gr} = 10^6$ and $\mathrm{Pr} = 0.71$, respectively. Cases with three groups of typical

TABLE VIII. Comparisons of wall and global data with literature results.

Scheme	$\overline{\mathrm{Nu}}$	Nu′	$\bar{\hat{u}}$	$\hat{u}'$
Present study with 20×100 mesh	4.58182	0.007710	0.23999	3.900×10^{-5}
Present study with 30×150 mesh	4.57843	0.007520	0.23982	4.182×10^{-5}
Paolucci [45]	4.57942	0.007050	0.2395	3.460×10^{-5}
Xin and Le Quéré [48]	4.57946	0.0070918	–	–
Bassi *et al.* [2]	4.57939	0.007075	0.23949	3.4021×10^{-5}
Gjesdal *et al.* [49]	4.57933	0.0071026	0.2395	3.354×10^{-5}
Arpino *et al.* [46]	4.52497	0.00861	–	–
Balam and Gupta [47]	4.565	0.00713	0.2397	4.1250×10^{-5}

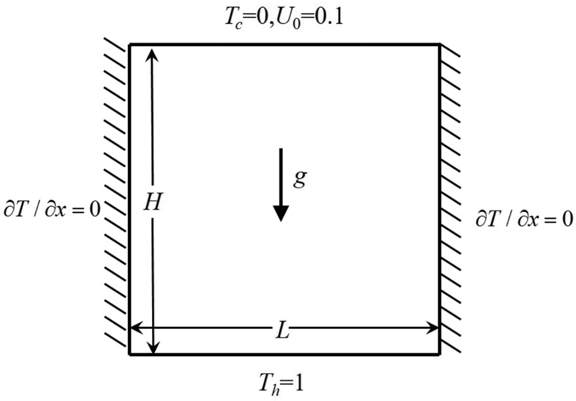


FIG. 15. Schematic of the thermal lid-driven cavity flow.

Richardson numbers (Ri = 10, 1.0, and 0.1) and the corresponding Reynolds numbers (Re = 316, 1000, and 3162) are simulated. It turns out that the problem of the combined shear and buoyancy-driven convection can be classified into three flow regimes: a pure natural convection for Ri $\gg$ 1, a mixed convection for Ri = 1, and a pure forced convection for Ri $\ll$ 1. A uniform mesh size of 20×20 for p^3 scheme is adopted, which is much smaller than that in a previous study using a second-order scheme [51].

Isotherms and streamlines obtained by the p^3 scheme are shown in Fig. 16. It can be seen that for the natural convection dominant case (Ri = 10, Re = 316), two major counterclockwise vortices—one is driven by the moving top lid and the other is driven by buoyancy—appear. This is caused by the interaction of buoyancy and shear force. Owing to the fact that the buoyancy effect outweighs the shear force effect, the vortex is larger in the lower half of the cavity than in the upper region. Furthermore, the hot and cold fluid mixing in these locations leads to a higher temperature gradient between the counterclockwise vortex and the top and bottom walls. For the mixed convection case (Ri = 1, Re = 1000), the counterclockwise vortices become almost equal, and the appearance of the vortex in the lower part of the cavity is related to the opposite action of the moving lid. For the forced convection dominant case (Ri = 0.1, Re = 3162), the flow pattern is extremely similar to the isothermal lid-driven cavity flow at the same Reynolds number. All of these observations agree well with the previous studies [32,50], which validates the physical reliability of the method.

Table IX compares the average Nusselt number on the top and bottom walls for the three values of Richardson number with the reference data of Bettaibi *et al.* [50] using the second-order hybrid MRTLBM and that of Chen *et al.* [32] using the third-order simplified LBM. From the table, it is found that the heat transmission at the hot wall increases as the Richardson number (Ri) decreases due to the increasing buoyancy effect in the lower half of the cavity. Despite a coarse mesh being used, the present results are in good agreement with the reference data, which quantitatively validates the accuracy of the present scheme for simulating mixed convection.

TABLE IX. Comparisons of average Nusselt number for 2D thermal lid-driven cavity.

Ri	Bettaibi *et al.* [50]	Chen *et al.* [32]	FRTLBFS
10	4.848	4.811	4.835
1	5.739	5.966	5.767
0.1	12.138	12.979	12.703

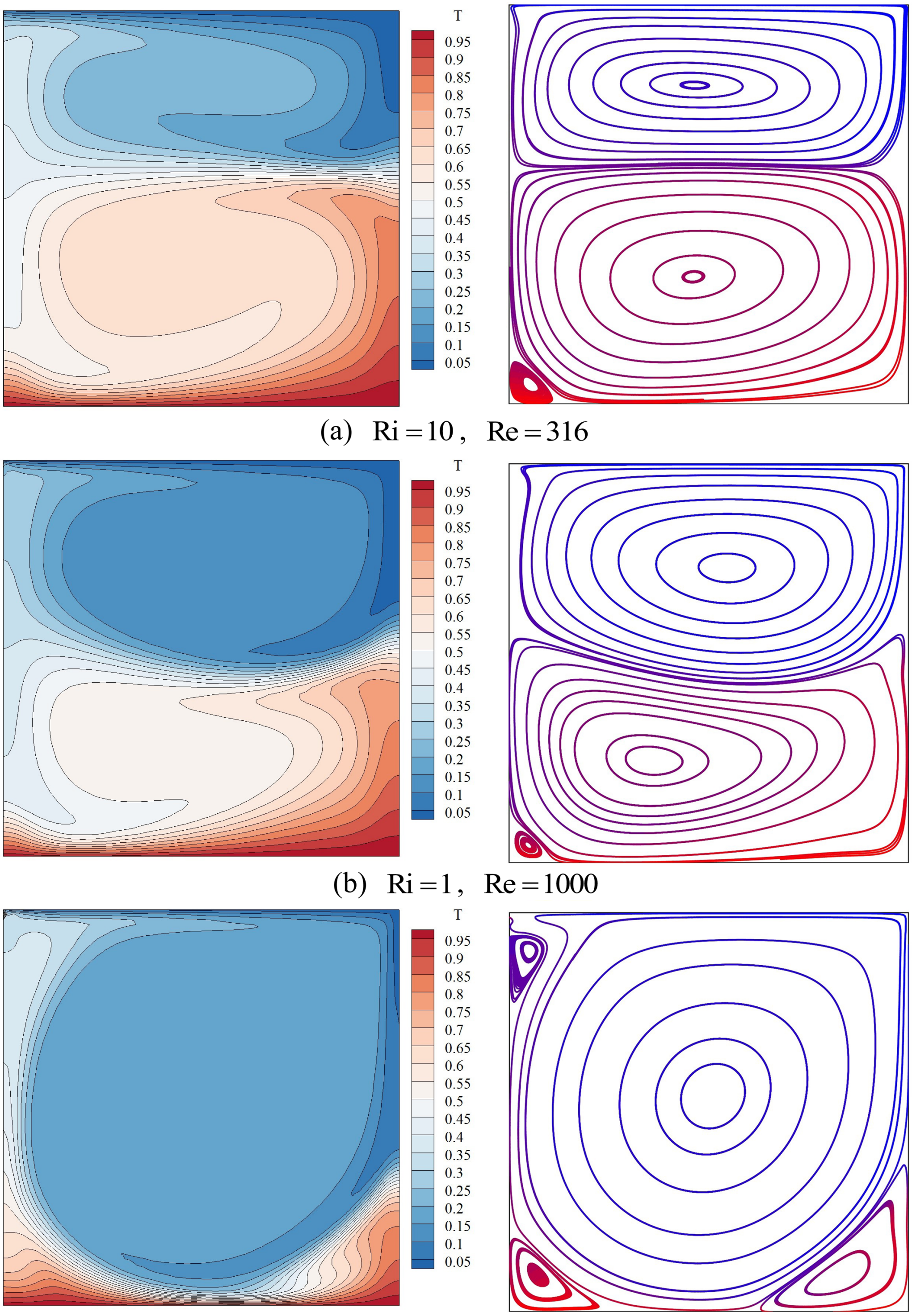
V. CONCLUSIONS

Based on the flux reconstruction method together with the thermal lattice Boltzmann flux solver, this paper presents a high-order solver (FRTLBFS) for effective and accurate simulations of incompressible thermal flows. In this method, a particular form of the Navier-Stokes–Boussinesq equation, which is derived from the thermal lattice Boltzmann equation and contains both macroscopic flow variables and microscopic particle distribution function, is solved by a high-order energy stable flux reconstruction scheme. The inviscid and viscous fluxes of the solution and flux points are coupled and computed uniformly and simultaneously by the thermal lattice Boltzmann flux solver. As a result, there is no need to discretize the second-order partial derivative term and to deal with pressure-velocity coupling, which brings the present scheme conciseness and compactness.

The proposed method is validated by simulating several two-dimensional cases: (a) the porous plate problem, (b) natural convection in a square cavity, (c) unsteady natural convection in a tall cavity, and (d) thermal lid-driven cavity flow. For both steady and unsteady flows and heat transfer, the results obtained with the current scheme are in good agreement with analytical or benchmark data. The high-order accuracy for both flow and temperature fields is verified. Compared with the recently developed flux reconstruction thermal lattice Boltzmann method (FRTLBM), the present scheme is more efficient and low storage since only the macroscopic variables are stored and evolved. The present method also inherits the good stability of the lattice Boltzmann flux solver. Benefiting from high-order accuracy, the present solver can accurately predict the transient natural convection when the Rayleigh number is slightly larger than the critical value. Apart from natural convection, the present solver can also give an accurate result of mixed convection. Although not displayed in this paper, the present method can be directly extended to unstructured meshes. The obtained numerical results indicate that the current FRTLBFS is a promising tool to simulate incompressible thermal flow problems due to its accuracy, efficiency, and low memory cost.

ACKNOWLEDGMENTS

J.W. acknowledges the support of the National Natural Science Foundation of China (Grant No. 12072158), the Natural Science Foundation of Jiangsu Province (Grant No. BK20191271), and the Research Fund of Key Laboratory of Computational Aerodynamics, AVIC Aerodynamics Research Institute (Grant No. YL2022XFX0402). This work is also supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).



(a) Ri = 10, Re = 316

(b) Ri = 1, Re = 1000

(c) Ri = 0.1, Re = 3162

FIG. 16. Isotherms (left) and streamlines (right) of 2D thermal lid-driven cavity flow.

[1] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal, High-order CFD methods: Current status and perspective, Int. J. Numer. Methods Fluids **72**, 811 (2013).

[2] F. Bassi, A. Crivellini, D. A. Di Pietro, and S. Rebay, An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows, Comput. Fluids **36**, 1529 (2007).

[3] S. Wakashima and T. S. Saitoh, Benchmark solutions for natural convection in a cubic cavity using the high-order time-space method, Int. J. Heat Mass Transfer **47**, 853 (2004).

[4] S. Busto, M. Tavelli, W. Boscheri, and M. Dumbser, Efficient high order accurate staggered semi-implicit discontinuous Galerkin methods for natural convection problems, Comput. Fluids **198**, 104399 (2020).

[5] B. X. Zhao and Z. F. Tian, High-resolution high-order upwind compact scheme-based numerical computation of natural convection flows in a square cavity, Int. J. Heat Mass Transfer **98**, 313 (2016).

[6] B. Klein, B. Müller, F. Kummer, and M. Oberlack, A high-order discontinuous Galerkin solver for low Mach number flows, Int. J. Numer. Methods Fluids **81**, 489 (2016).

[7] H. T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, in *Proceedings of 18th AIAA Computational Fluid Dynamics Conference* (AIAA Press, Reston, VA, 2007), p. 4079.

[8] Z. J. Wang and H. Gao, A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids, J. Comput. Phys. **228**, 8161 (2009).

[9] M. Yu, Z. J. Wang, and Y. Liu, On the accuracy and efficiency of discontinuous Galerkin, spectral difference and correction procedure via reconstruction methods, J. Comput. Phys. **259**, 70 (2014).

[10] C. Cox, C. Liang, and M. W. Plesniak, A high-order solver for unsteady incompressible Navier-Stokes equations using the flux reconstruction method on unstructured grids with implicit dual time stepping, J. Comput. Phys. **314**, 414 (2016).

[11] Z. J. Wang, Y. Li, F. Jia, G. M. Laskowski, J. Kopriva, U. Paliath, and R. Bhaskaran, Towards industrial large eddy simulation using the FR/CPR method, Comput. Fluids **156**, 579 (2017).

[12] N. A. Loppi, F. D. Witherden, A. Jameson, and P. E. Vincent, A high-order cross-platform incompressible Navier-Stokes solver via artificial compressibility with application to a turbulent jet, Comput. Phys. Commun. **233**, 193 (2018).

[13] F. D. Witherden, A. M. Farrington, and P. E. Vincent, PYFR: An open source framework for solving advection-diffusion type problems on streaming architectures using the flux reconstruction approach, Comput. Phys. Commun. **185**, 3028 (2014).

[14] Y. Yu, F. Liu, T. Zhou, C. Gao, and Y. Liu, Numerical solutions of 2-D steady compressible natural convection using high-order flux reconstruction, Acta Mech. Sin. **35**, 401 (2019).

[15] G. McNamara and B. Alder, Analysis of the lattice Boltzmann treatment of hydrodynamics, Phys. A (Amsterdam) **194**, 218 (1993).

[16] M. Atif, M. Namburi, and S. Ansumali, Higher-order lattice Boltzmann model for thermohydrodynamics, Phys. Rev. E **98**, 053311 (2018).

[17] X. He, S. Chen, and G. D. Doolen, A novel thermal model for the lattice Boltzmann method in incompressible limit, J. Comput. Phys. **146**, 282 (1998).

[18] Z. Guo, B. Shi, and C. Zheng, A coupled lattice BGK model for the Boussinesq equations, Int. J. Numer. Methods Fluids **39**, 325 (2002).

[19] Y. Peng, C. Shu, and Y. T. Chew, Simplified thermal lattice Boltzmann model for incompressible thermal flows, Phys. Rev. E **68**, 026701 (2003).

[20] A. Mezrhab, M. Bouzidi, and P. Lallemand, Hybrid lattice-Boltzmann finite-difference simulation of convective flows, Comput. Fluids **33**, 623 (2004).

[21] Z. Li, M. Yang, and Y. Zhang, A coupled lattice Boltzmann and finite volume method for natural convection simulation, Int. J. Heat Mass Transfer **70**, 864 (2014).

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

[23] C. Shu, X. D. Niu, and Y. T. Chew, Taylor-series expansion and least-squares-based lattice Boltzmann method: Two-dimensional formulation and its applications, Phys. Rev. E **65**, 036708 (2002).

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

[25] H. Chen, Volumetric formulation of the lattice Boltzmann method for fluid dynamics: Basic concept, Phys. Rev. E **58**, 3955 (1998).

[26] T. Lee and C. L. Lin, A characteristic Galerkin method for discrete Boltzmann equation, J. Comput. Phys. **171**, 336 (2001).

[27] A. Krämer, K. Küllmer, D. Reith, W. Joppich, and H. Foysi, Semi-Lagrangian off-lattice Boltzmann method for weakly compressible flows, Phys. Rev. E **95**, 023305 (2017).

[28] W. Li, High order spectral difference lattice Boltzmann method for incompressible hydrodynamics, J. Comput. Phys. **345**, 618 (2017).

[29] Z. Chen, C. Shu, Y. Wang, L. M. Yang, and D. Tan, A simplified lattice Boltzmann method without evolution of distribution function, Adv. Appl. Math. Mech. **9**, 1 (2017).

[30] C. Shu, Y. Wang, C. J. Teo, and J. Wu, Development of lattice Boltzmann flux solver for simulation of incompressible flows, Adv. Appl. Math. Mech. **6**, 436 (2014).

[31] Z. Chen, C. Shu, and D. Tan, Highly accurate simplified lattice Boltzmann method, Phys. Fluids **30**, 103605 (2018).

[32] Z. Chen, C. Shu, and D. Tan, High-order simplified thermal lattice Boltzmann method for incompressible thermal flows, Int. J. Heat Mass Transfer **127**, 1 (2018).

[33] Y. Y. Liu, C. Shu, H. W. Zhang, and L. M. Yang, A high order least square-based finite difference-finite volume method with lattice Boltzmann flux solver for simulation of incompressible flows on unstructured grids, J. Comput. Phys. **401**, 109019 (2020).

[34] Y. Wang, C. Shu, and C. J. Teo, Thermal lattice Boltzmann flux solver and its application for simulation of incompressible thermal flows, Comput. Fluids **94**, 98 (2014).

[35] Y. Y. Liu, H. W. Zhang, L. M. Yang, and C. Shu, High-order least-square-based finite-difference–finite-volume method for simulation of incompressible thermal flows on arbitrary grids, Phys. Rev. E **100**, 063308 (2019).

[36] F. Bassi and S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the

compressible Navier-Stokes equations, J. Comput. Phys. **131**, 267 (1997).

[37] Y. Xu and C. W. Shu, Local discontinuous Galerkin methods for high-order time-dependent partial differential equations, Commun. Comput. Phys. **7**, 1 (2010).

[38] N. C. Nguyen, J. Peraire, and B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations, J. Comput. Phys. **230**, 1147 (2011).

[39] S. Chen and G. D. Doolen, Lattice Boltzmann method for fluid flows, Annu. Rev. Fluid Mech. **30**, 329 (1998).

[40] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction* (Springer Science & Business Media, Berlin, 2013).

[41] C. Ma, J. Wu, L. Yang, and H. Dong, A coupled high-order implicit-explicit flux reconstruction lattice Boltzmann method for nearly incompressible thermal flows, Int. J. Heat Mass Transfer **187**, 122575 (2022).

[42] J. Wang, D. Wang, P. Lallemand, and L. S. Luo, Lattice Boltzmann simulations of thermal convective flows in two dimensions, Comput. Math. Appl. **65**, 262 (2013).

[43] A. Xu, L. Shi, and T. S. Zhao, Accelerated lattice Boltzmann simulation using GPU and OpenACC with data management, Int. J. Heat Mass Transfer **109**, 577 (2017).

[44] H. N. Dixit and V. Babu, Simulation of high Rayleigh number natural convection in a square cavity using the lattice Boltzmann method, Int. J. Heat Mass Transfer **49**, 727 (2006).

[45] M. A. Christon, P. M. Gresho, and S. B. Sutton, Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution), Int. J. Numer. Methods Fluids **40**, 953 (2002).

[46] F. Arpino, G. Cortellessa, M. Dell'Isola, N. Massarotti, and A. Mauro, High order explicit solutions for the transient natural convection of incompressible fluids in tall cavities, Numer. Heat Transfer, Part A **66**, 839 (2014).

[47] N. B. Balam and A. Gupta, A fourth-order accurate finite difference method to evaluate the true transient behaviour of natural convection flow in enclosures, Int. J. Heat Fluid Flow **30**, 1233 (2020).

[48] S. Xin and P. Le Quéré, An extended Chebyshev pseudo-spectral benchmark for the 8:1 differentially heated cavity, Int. J. Numer. Methods Fluids **40**, 981 (2002).

[49] T. Gjesdal, C. E. Wasberg, and B. A. P. Reif, Spectral element benchmark simulations of natural convection in two-dimensional cavities, Int. J. Numer. Methods Fluids **50**, 1297 (2006).

[50] S. Bettaibi, F. Kuznik, and E. Sediki, Hybrid lattice Boltzmann finite difference simulation of mixed convection flows in a lid-driven square cavity, Phys. Lett. A **378**, 2429 (2014).

[51] Z. Chen, C. Shu, and D. Tan, A truly second-order and unconditionally stable thermal lattice Boltzmann method, Appl. Sci. **7**, 277 (2017).