

Article

From Lattice Boltzmann Method to Lattice Boltzmann Flux Solver

Yan Wang ¹, Liming Yang ² and Chang Shu ^{1,*}

¹ Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore; E-Mail: mpewngy@nus.edu.sg

² Department of Aerodynamics, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Yudao Street, Nanjing 210016, China; E-Mail: yangliming_2011@126.com

* Author to whom correspondence should be addressed; E-Mail: mpeshuc@nus.edu.sg; Tel.: +65-6516-6476.

Academic Editors: Sauro Succi and Ignazio Licata

Received: 10 September 2015 / Accepted: 2 November 2015 / Published: 13 November 2015

Abstract: Based on the lattice Boltzmann method (LBM), the lattice Boltzmann flux solver (LBFS), which combines the advantages of conventional Navier–Stokes solvers and lattice Boltzmann solvers, was proposed recently. Specifically, LBFS applies the finite volume method to solve the macroscopic governing equations which provide solutions for macroscopic flow variables at cell centers. In the meantime, numerical fluxes at each cell interface are evaluated by local reconstruction of LBM solution. In other words, in LBFS, LBM is only locally applied at the cell interface for one streaming step. This is quite different from the conventional LBM, which is globally applied in the whole flow domain. This paper shows three different versions of LBFS respectively for isothermal, thermal and compressible flows and their relationships with the standard LBM. In particular, the performance of isothermal LBFS in terms of accuracy, efficiency and stability is investigated by comparing it with the standard LBM. The thermal LBFS is simplified by using the D2Q4 lattice velocity model and its performance is examined by its application to simulate natural convection with high Rayleigh numbers. It is demonstrated that the compressible LBFS can be effectively used to simulate both inviscid and viscous flows by incorporating non-equilibrium effects into the process for inviscid flux reconstruction. Several numerical examples, including lid-driven cavity flow, natural convection in a square cavity at Rayleigh numbers of 10^7 and 10^8 and transonic flow around a staggered-biplane configuration, are tested on structured or unstructured grids to examine the performance of three LBFS versions. Good agreements

have been achieved with the published data, which validates the capability of LBFS in simulating a variety of flow problems.

Keywords: lattice Boltzmann flux solver; Navier–Stokes equation; lattice Boltzmann equation; incompressible flow; compressible flow

1. Introduction

Since its earliest appearance in 1988 [1], the lattice Boltzmann equation (LBE)-based method [2–11] has been developed into an effective and efficient solver for simulating a variety of complex fluid flow problems, such as isothermal and thermal flows [9,10,12], multi-phase and multi-component flows [8,13–15], compressible flows [16–18] and micro non-equilibrium flows [19–21]. To effectively study these problems, a large number of LBE-based methods have been continuously proposed and refined [9–23].

Historically, the LBM was developed from the lattice gas cellular automata (LGCA) method, aiming to remove its statistical noise and limitation to use Boolean numbers. It was later proven that the solid foundation of LBM roots in gas kinetic theory and the Chapman–Enskog expansion analysis, through which both continuous and lattice Boltzmann equation can recover the Navier–Stokes equations. This confers the LBM with an appealing kinetic nature. Besides, LBM has very simple numerical algorithms with algebraic manipulations. In particular, two simple steps of streaming and collision are involved in its solution process. The streaming process is linear, which moves particles with different distribution functions to neighboring points, but effectively considers non-linear physics in fluid dynamics. The collision process takes place locally in either lattice velocity space or macroscopic moment space according to different collision models [2,4,22]. Both single relaxation time (SRT) model (lattice Bhatnagar–Gross–Krook (LBGK) model) [4] and multiple-relaxation-time (MRT) model [22] can be applied. It seems that the MRT model eliminates some defects of the LBGK model and enriches the connotation of LBM, although the computational effort is also increased. With these distinguishing features, the LBM has emerged as an alternative powerful tool in many complex fluid flow problems [13,20,23–25]. However, the LBM also suffers from some drawbacks. Due to lattice regularity, the standard LBM usually restricts its applications to uniform grids, which hinders its direct application in certain problems with curved boundaries. In addition, as compared with conventional Navier–Stokes solvers, LBM usually needs more virtual storage to store both the distribution functions and flow variables. Furthermore, the time step in LBM is coupled with mesh spacing, which presents a great challenge for applications on multi-block and adaptive grids. Moreover, LBM is mostly applied to simulate incompressible flows since the equilibrium distribution function is obtained from low Mach number approximation.

To remove the drawbacks of standard LBM, some efforts [5–7,21] have been made to directly solve the discrete velocity Boltzmann equation (DVBE) with well-established numerical approaches. This way eliminates the coupling issue between the mesh spacing and time step in standard LBM and can be effectively applied on non-uniform grids. However, it loses the primary advantages of LBM such

as simple implementation and algebraic operation. It may also involve a large numerical dissipation and encounter numerical instability [26,27].

Recently, from Chapman–Enskog (C-E) analysis, several versions of lattice Boltzmann flux solver (LBFS) [8–11] have been developed based on the standard LBM for simulating isothermal, thermal, multiphase and compressible flows. LBFS is a finite volume solver for direct update of the macroscopic flow variables at cell centers. The key idea of the LBFS is to evaluate numerical fluxes at each cell interface by local application of LBM solution for one streaming step. In other words, in LBFS, LBM is applied locally and independently at each interface, and the streaming time step is nothing to do with the real time step. Instead, it is only applied to calculate the relaxation parameter used in the process of solution reconstruction by LBM. For a control cell with four cell interfaces of a two-dimensional case, the local LBM solution at different interface is reconstructed independently and the streaming time step could be different. This gives us a great flexibility for application of the solver to non-uniform meshes and complex geometry. This feature effectively removes the drawback of coupling issue between the mesh spacing and time step in the conventional LBM. In addition, the LBFS does not track the evolution of distribution function and the dependent variables are the macroscopic flow variable. As a consequence, the required virtual memory is reduced substantially, and the physical boundary conditions can be implemented directly without converting to those for distribution functions. Moreover, as compared with conventional incompressible Navier–Stokes solvers, such as the well-known semi-implicit method for pressure linked equations (SIMPLE) method and its variants [28], the LBFS overcomes their drawbacks of tedious discretization of the second order derivatives, the requirement of staggered grids for preventing pressure oscillations and slow convergence on fine grids [29]. Indeed, the LBFS combines the advantages of lattice Boltzmann solver (simplicity and kinetic nature) with those of Navier–Stokes solver by the finite volume method (FVM) (geometric flexibility and easy implementation of boundary conditions). It shows the progressive development from LBM to LBFS.

Since LBFS is a new solver, its performance has not been fully investigated and further improvements and simplifications for different flow problems may be required. Particularly, there is a lack of a systematic investigation on its accuracy, efficiency, stability and capability in simulating isothermal flows, thermal flows at high Reynolds/Rayleigh numbers, and compressible flows. Motivated by this, the LBFS will be refined and examined comprehensively in this paper. Firstly, numerical simulations of two-dimensional lid-driven cavity flows [30–34] are considered. A detailed comparison of its performances with the standard LBM will be carried out. Secondly, the thermal LBFS will be simplified in this work by using the D2Q4 model and then applied to simulate natural convection at two high Rayleigh numbers of $Ra = 10^7$ and 10^8 . The obtained results on different grids will be compared with those by MRT-based LBM [35]. Thirdly, the compressible LBFS will be improved by incorporating non-equilibrium effects into the process for inviscid flux reconstruction and introducing a switch function to control numerical diffusion. Its performance will be examined by simulating transonic inviscid and viscous flows around a staggered-biplane configuration on unstructured grids. The obtained results will be compared with the published data.

2. Lattice Boltzmann Method

For a general case, we consider incompressible thermal fluid flows and the LBE with BGK approximation can be written as [34,36]:

$$f_\alpha(\mathbf{r} + \mathbf{e}_\alpha \delta_t, t + \delta_t) = f_\alpha(\mathbf{r}, t) + \frac{f_\alpha^{eq}(\mathbf{r}, t) - f_\alpha(\mathbf{r}, t)}{\tau_v}, \alpha = 0, 1, \dots, N, \tag{1}$$

$$g_\alpha(\mathbf{r} + \mathbf{e}_\alpha \delta_t, t + \delta_t) = g_\alpha(\mathbf{r}, t) + \frac{g_\alpha^{eq}(\mathbf{r}, t) - g_\alpha(\mathbf{r}, t)}{\tau_c}, \alpha = 0, 1, \dots, M, \tag{2}$$

where f_α and g_α are density distribution functions (DDF) and temperature distribution functions (TDF) respectively; f_α^{eq} and g_α^{eq} are equilibrium states of f_α and g_α ; τ_v and τ_c are relaxation parameters; δ_t is the streaming time; M and N are the total number of discrete particles for f_α and g_α . The macroscopic fluid properties of density ρ , velocity \mathbf{u} and temperature T are evaluated from lattice moments of f_α and g_α :

$$\rho = \sum_{\alpha=0}^N f_\alpha, \quad \rho \mathbf{u} = \sum_{\alpha=0}^N f_\alpha \mathbf{e}_\alpha, \quad T = \sum_{\alpha=0}^M g_\alpha \tag{3}$$

The equilibrium DDF f_α^{eq} is given by [4]:

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

The D2Q9 model, whose lattice components are written as $\mathbf{e}_\alpha = (0, 0), (\pm 1, 0), (0, \pm 1)$ and $(\pm 1, \pm 1)$, is commonly applied for f_α^{eq} . The weights w_α and the sound speed c_s in Equation (4) are $w_0 = 4/9, w_{1-4} = 1/9, w_{5-8} = 1/36$ and $c_s = 1/\sqrt{3}$. The equilibrium TDF g_α^{eq} is given by [37]:

$$g_\alpha^{eq}(\mathbf{r}, t) = \frac{T}{4} [1 + 2\mathbf{e}_\alpha \cdot \mathbf{u}] \tag{5}$$

The D2Q4 lattice velocity model, whose lattice components are written as $\mathbf{e}_\alpha = (\pm 1, \pm 1)$, is usually used for g_α^{eq} . τ_v and τ_c are respectively calculated by the dynamic viscosity μ and thermal diffusivity κ :

$$\mu = \left(\tau_v - \frac{1}{2} \right) \rho c_s^2 \delta_t \tag{6a}$$

$$\kappa = \frac{1}{2} \left(\tau_c - \frac{1}{2} \right) \delta_t \tag{6b}$$

Note that Equations (1) and (2) are applied to simulate thermal flows. If isothermal flows are considered, we can just apply Equation (1). For the application of LBM, we have to apply Equations (1) and (2) in the whole computational domain and track the evolution of distribution functions f_α and g_α . In addition, physical boundary conditions for macroscopic variables are converted to those for f_α and g_α . For no-slip condition, bounce-back scheme can be easily applied. In some cases, accurate

implementation of other boundary conditions in the LBM may not be as straightforward as in Navier–Stokes solvers.

3. Chapman–Enskog Analysis

As a mesoscopic method with microscopic particle distribution functions, the LBM described in Section 2 has been well applied to study weakly compressible fluid flows in incompressible limit. Conventionally, such flows are governed and solved by the macroscopic conservation laws of mass, momentum and energy. This indicates that the solution of a physical flow problem can be either obtained by applying the mesoscopic LBM or the macroscopic Navier–Stokes solver. It is interesting to note that these two intrinsically different numerical methods, one from microscopic statistical physics and the other from macroscopic conservation laws, are both able to study the same fluid flow problem. From this point of view, it can be inferred that these two methods should have some connections. Indeed, it has been proven that the lattice Boltzmann equation is able to recover the Navier–Stokes equation through the multi-scale Chapman–Enskog expansion analysis, which is briefly introduced below.

A multi-scale expansion of the DDF, TDF, the temporal derivative and the spatial derivative can be respectively given by:

$$f_\alpha = f_\alpha^{(0)} + \varepsilon f_\alpha^{(1)} + \varepsilon^2 f_\alpha^{(2)} \tag{7a}$$

$$g_\alpha = g_\alpha^{(0)} + \varepsilon g_\alpha^{(1)} + \varepsilon^2 g_\alpha^{(2)} \tag{7b}$$

$$\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_0} + \varepsilon^2 \frac{\partial}{\partial t_1} \tag{7c}$$

$$\nabla_r = \varepsilon \nabla_{r1} \tag{7d}$$

where ε is a small parameter proportional to the Knudsen number. Applying the second order Taylor-series expansion to Equations (1) and (2) gives:

$$\left(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\right) f_\alpha + \frac{\delta_t}{2} \left(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\right)^2 f_\alpha + \frac{1}{\tau \delta_t} (f_\alpha - f_\alpha^{eq}) + O(\delta_t^2) = 0, \tag{8}$$

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

The macroscopic Navier–Stokes equations for conservation laws can be recovered by substituting Equation (7) into Equations (8) and (9) [9,10]:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F} = 0 \tag{10}$$

where:

$$\mathbf{W} = \begin{Bmatrix} \rho \\ \rho \mathbf{u} \\ T \end{Bmatrix} \text{ and } \mathbf{F} = \begin{Bmatrix} \mathbf{P} \\ \mathbf{\Pi} \\ \mathbf{Q} \end{Bmatrix} \tag{11}$$

In Equation (11), \mathbf{P} , $\mathbf{\Pi}$ and \mathbf{Q} are respectively mass, momentum and energy fluxes and can be given by lattice summations of DDFs and TDFs [9,10]:

$$\mathbf{P} = \rho \mathbf{u} = \sum_{\alpha=0}^N f_{\alpha}^{eq} \mathbf{e}_{\alpha} \tag{12a}$$

$$\mathbf{\Pi} = \rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I} - \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) = \sum_{\alpha=0}^N \mathbf{e}_{\alpha} \mathbf{e}_{\alpha} f_{\alpha}^{\wedge} \tag{12b}$$

$$\mathbf{Q} = \mathbf{u} T - \kappa \nabla T = \sum_{\alpha=0}^N \mathbf{e}_{\alpha} g_{\alpha}^{\wedge} \tag{12c}$$

where:

$$f_{\alpha}^{\wedge} = f_{\alpha}^{eq} + \left(1 - \frac{1}{2\tau_v}\right) f_{\alpha}^{neq} \quad \text{and} \quad g_{\alpha}^{\wedge} = g_{\alpha}^{eq} + \left(1 - \frac{1}{2\tau_c}\right) g_{\alpha}^{neq} \tag{13}$$

$$f_{\alpha}^{neq} = -\tau_v \delta_t \left(\frac{\partial}{\partial t} + \mathbf{e}_{\alpha} \cdot \nabla \right) f_{\alpha}^{eq} \tag{14a}$$

$$g_{\alpha}^{neq} = -\tau_c \delta_t \left(\frac{\partial}{\partial t} + \mathbf{e}_{\alpha} \cdot \nabla \right) g_{\alpha}^{eq} \tag{14b}$$

In the conventional LBM, the Chapman–Enskog (C-E) analysis is usually used to verify that the flow variables obtained by LBM can satisfy the macroscopic Navier–Stokes equations. In some interesting works, this C-E analysis is also applied to build a hybrid solver by combing the LBM with the FVM scheme for Navier–Stokes equations [37–39]. In our recent work [9,10], it was found that the relationship between flow variables/fluxes and particle distribution functions given in Equation (12) can be applied to build a new solver named LBFS with a better performance, which effectively combines the advantages of Navier–Stokes solver and lattice Boltzmann solver and at the same time removes some of their drawbacks. The reliability and flexibility of the LBFS have been demonstrated in [8–10].

4. LBFS for Isothermal and Thermal Incompressible Flows

Based on the LBM given by Equations (1) and (2), the LBFS [9] has been proposed by solving Equation (10) directly. Unlike conventional Navier–Stokes solver, which applies well-established schemes to discretize Equation (10), the LBFS is a finite volume solver and reconstructs its fluxes locally with Equation (12) derived from Chapman–Enskog analysis. It may be pointed out that Equation (12) for flux reconstruction can be applied with different lattice velocity models. In our previous work [9], the D2Q9 thermal model for temperature field is applied, which is obviously complex and inefficient as compared with the D2Q4 model [34]. In this work, this drawback will be removed by applying the D2Q4 model. The details are given below.

4.1. Finite Volume Discretization

A finite volume discretization of Equation (10) over a control volume Ω_i gives the following formulation:

$$\frac{dW_i}{dt} = -\frac{1}{dV_i} \sum_k \mathbf{R}_k dl_k, \quad \mathbf{R}_k = (\mathbf{n} \cdot \mathbf{F})_k \tag{15}$$

where dV_i is the area of the control cell Ω_i , dl_k is the length of the k -th control surface enclosed Ω_i and $\mathbf{n} = (n_x, n_y)$ is the unit normal vector on the k -th control surface. With the D2Q9 and D2Q4 models, the flux \mathbf{R}_k at each cell interface can be written as follows:

$$\mathbf{R}_k = \begin{pmatrix} n_x (f_1^{eq} - f_3^{eq} + f_5^{eq} - f_6^{eq} - f_7^{eq} + f_8^{eq}) + n_y (f_2^{eq} - f_4^{eq} + f_5^{eq} + f_6^{eq} - f_7^{eq} - f_8^{eq}) \\ n_x (\hat{f}_1 + \hat{f}_3 + \hat{f}_5 + \hat{f}_6 + \hat{f}_7 + \hat{f}_8) + n_y (\hat{f}_5 - \hat{f}_6 + \hat{f}_7 - \hat{f}_8) \\ n_x (\hat{f}_5 - \hat{f}_6 + \hat{f}_7 - \hat{f}_8) + n_y (\hat{f}_2 + \hat{f}_4 + \hat{f}_5 + \hat{f}_6 + \hat{f}_7 + \hat{f}_8) \\ n_x (\hat{g}_1 - \hat{g}_3) + n_y (\hat{g}_2 - \hat{g}_4) \end{pmatrix}_k \quad (16)$$

As can be seen, the formulation of the energy flux reconstructed by the simplified D2Q4 lattice velocity model is much simpler than that given by the original D2Q9 model [9]. In Equation (16), the unknowns are \hat{f}_α and \hat{g}_α defined in Equation (13), which include both equilibrium and non-equilibrium density and temperature distribution functions f_α^{eq} , f_α^{neq} , g_α^{eq} and g_α^{neq} . All these unknowns can be obtained through local reconstruction of the LBE solution without tracking the evolution of the DDF and TDF.

4.2. Local Reconstruction of Fluxes at Each Interface

Figure 1 shows an interface between two adjacent control cells for local reconstruction of fluxes [10], in which the D2Q9 and D2Q4 lattice velocity models respectively for DDF and TDF are embedded. The non-equilibrium parts for DDF and TDF are given by Equation (14). At each cell interface, discretization of Equation (14) with the second-order Taylor-series expansion gives the following equations [9]:

$$f_\alpha^{neq}(\mathbf{r}, t) = -\tau_v [f_\alpha^{eq}(\mathbf{r}, t) - f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t)] \quad (17a)$$

$$g_\alpha^{neq}(\mathbf{r}, t) = -\tau_c [g_\alpha^{eq}(\mathbf{r}, t) - g_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t)] \quad (17b)$$

where \mathbf{r} represents the location of the cell interface. Equation (17) shows that both f_α^{neq} and g_α^{neq} can be approximated from f_α^{eq} and g_α^{eq} at the interface and its surrounding points $\mathbf{r} - \mathbf{e}_\alpha \delta_t$. Following the convention in LBM, f_α^{eq} and g_α^{eq} at the position and time $(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t)$ are computed by using Equations (4) and (5). The involved flow quantities of density ρ , velocity \mathbf{u} and temperature T at $(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t)$ can be obtained through interpolations:

$$\psi(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t) = \begin{cases} \psi(\mathbf{r}_i) + (\mathbf{r} - \mathbf{e}_\alpha \delta_t - \mathbf{r}_i) \cdot \nabla \psi(\mathbf{r}_i), & \text{when } \mathbf{r} - \mathbf{e}_\alpha \delta_t \text{ is in } \Omega_i \\ \psi(\mathbf{r}_{i+1}) + (\mathbf{r} - \mathbf{e}_\alpha \delta_t - \mathbf{r}_{i+1}) \cdot \nabla \psi(\mathbf{r}_{i+1}), & \text{when } \mathbf{r} - \mathbf{e}_\alpha \delta_t \text{ is in } \Omega_{i+1} \end{cases} \quad (18)$$

where \mathbf{r}_i and \mathbf{r}_{i+1} are the locations of the two cell centers and ψ represents any of the flow variables. After interpolation, $f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t)$ and $g_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t)$ can be obtained.

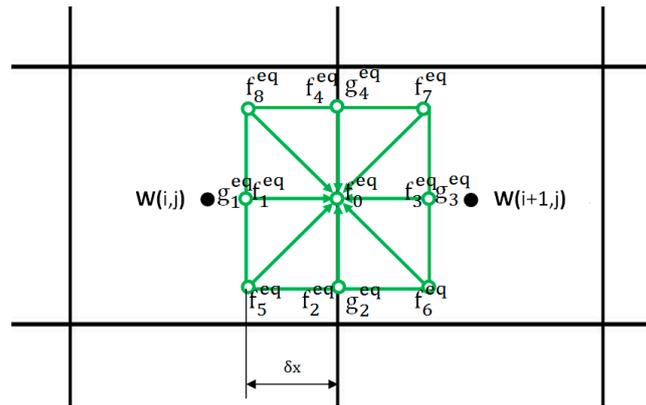


Figure 1. Evaluation of flux at an interface between two control cells.

Similarly, $f_\alpha^{eq}(\mathbf{r}, t)$ and $g_\alpha^{eq}(\mathbf{r}, t)$ can be calculated by Equations (4) and (5) if macroscopic flow variables are known. So, the challenging issue is to evaluate ρ , \mathbf{u} and T at the cell interface (\mathbf{r}, t) . Previous studies [9,10] have shown that they can be reconstructed locally by streaming the particle from the neighboring point $(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t)$ to the cell interface. As a result, we have:

$$\rho(\mathbf{r}, t) = \sum_{\alpha=0}^N f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t), \tag{19a}$$

$$\rho(\mathbf{r}, t)\mathbf{u}(\mathbf{r}, t) = \sum_{\alpha=0}^N f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t)\mathbf{e}_\alpha \tag{19b}$$

$$T(\mathbf{r}, t) = \sum_{\alpha=0}^N g_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta_t, t - \delta_t) \tag{19c}$$

With the flow quantities obtained from Equation (19), $f_\alpha^{eq}(\mathbf{r}, t)$ and $g_\alpha^{eq}(\mathbf{r}, t)$ can be computed by using Equations (4) and (5). Once f_α^{eq} , f_α^{neq} , g_α^{eq} and g_α^{neq} are obtained, the fluxes at each cell interface can be evaluated numerically. Then Equation (15) can be solved by conventional Runge–Kutta scheme.

5. LBFS for Compressible Flows

5.1. Navier–Stokes Equations Discretized by FVM and Compressible Lattice Boltzmann Model

In Section 4, the local LBM solution is reconstructed by using the conventional lattice Boltzmann model, which is limited to incompressible flows. Thus, LBFS in Section 4 can only be applied to simulate incompressible flows. To simulate compressible flows by LBFS, we need to use the compressible lattice Boltzmann model to reconstruct the solution at the cell interface. However, the existing compressible lattice Boltzmann models are very complicated and inefficient for 2D and 3D cases. To simplify the solution process and make LBFS be applicable for simulation of compressible flows, we apply the 1D compressible lattice Boltzmann model along the normal direction of cell interface to evaluate the inviscid flux, and the viscous flux is still approximated by conventional finite difference schemes. This process is equivalent to developing a Riemann solver by 1D compressible lattice Boltzmann model.

In LBFS, the Navier–Stokes equations are discretized by FVM and the fluxes at the cell interface are evaluated by local reconstruction of LBM solutions [11]. Thanks to the application of FVM, it is convenient for LBFS to apply on arbitrary meshes. The compressible Navier–Stokes equations discretized by FVM can be written as:

$$\frac{d\mathbf{W}_i}{dt} = -\frac{1}{dV_i} \sum_{k=1}^{N_f} (\mathbf{F}_{ck} - \mathbf{F}_{vk}) dl_k \tag{20}$$

where i is the index of a control volume, dV_i and N_f represent the volume and the number of interfaces of the control volume i , respectively. \mathbf{W} , \mathbf{F}_c and \mathbf{F}_v are the conservative variables at the cell center and the inviscid and viscous fluxes at the cell interface given by:

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \mathbf{F}_c = \begin{bmatrix} \rho U_n \\ \rho u U_n + n_x p \\ \rho v U_n + n_y p \\ (\rho E + p) U_n \end{bmatrix}, \quad \mathbf{F}_v = \begin{bmatrix} 0 \\ n_x \tau_{xx} + n_y \tau_{xy} \\ n_x \tau_{yx} + n_y \tau_{yy} \\ n_x \Theta_x + n_y \Theta_y \end{bmatrix} \tag{21}$$

Here, ρ and p are the density and pressure of mean flow, respectively. (u, v) and (n_x, n_y) denote the velocity vector and unit normal vector on the control surface in the Cartesian coordinate system, respectively. E is the total energy of mean flow. U_n represents the normal velocity. Furthermore, τ_{ij} denotes the components of viscous stress tensor and Θ_i represents the term describing the work of viscous stress and the heat conduction in the fluid.

To compute \mathbf{F}_c by LBFS, the compressible lattice Boltzmann model is required. In this work, the non-free parameter D1Q4 model [11] is utilized. The configuration of this model is shown in Figure 2, and it is given by:

$$\begin{aligned} g_1 &= \frac{\rho(-d_1 d_2^2 - d_2^2 u + d_1 u^2 + d_1 c^2 + u^3 + 3uc^2)}{2d_1(d_1^2 - d_2^2)} \\ g_2 &= \frac{\rho(-d_1 d_2^2 + d_2^2 u + d_1 u^2 + d_1 c^2 - u^3 - 3uc^2)}{2d_1(d_1^2 - d_2^2)} \\ g_3 &= \frac{\rho(d_1^2 d_2 + d_1^2 u - d_2 u^2 - d_2 c^2 - u^3 - 3uc^2)}{2d_2(d_1^2 - d_2^2)} \\ g_4 &= \frac{\rho(d_1^2 d_2 - d_1^2 u - d_2 u^2 - d_2 c^2 + u^3 + 3uc^2)}{2d_2(d_1^2 - d_2^2)} \\ d_1 &= \sqrt{u^2 + 3c^2 - \sqrt{4u^2 c^2 + 6c^4}} \\ d_2 &= \sqrt{u^2 + 3c^2 + \sqrt{4u^2 c^2 + 6c^4}} \end{aligned} \tag{22}$$

where c represents the peculiar velocity of particles defined as $c = \sqrt{p/\rho}$. Note that in Equation (22), g is the equilibrium distribution function f^{eq} . As D1Q4 model is a one-dimensional model, it needs to be applied along the normal direction of cell interface for multi-dimensional problems [11]. That is,

the velocity u in Equations (22) and (23) should be replaced by normal velocity U_n when multi-dimensional problems are considered.

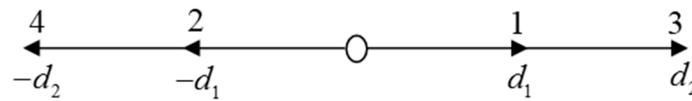


Figure 2. Configuration of non-free parameter D1Q4 model.

Suppose that the cell interface is located at $\mathbf{r} = 0$. For simplicity, a local-coordinate system with x -axis pointing to the normal direction and y -axis pointing to the tangential direction of the cell interface is used. As a result, the inviscid flux at the cell interface in the local-coordinate system can be written as:

$$\mathbf{F}_c^* = [\rho U_n \quad \rho U_n U_n + p \quad \rho U_n U_\tau \quad (\rho E + p) U_n]^T = \mathbf{F}_c^n + \mathbf{F}_c^t \tag{24a}$$

$$\mathbf{F}_c^n = \left[\rho U_n \quad \rho U_n U_n + p \quad 0 \quad \left(\rho \left(\frac{1}{2} U_n^2 + e \right) + p \right) U_n \right]^T \tag{24b}$$

$$\mathbf{F}_c^t = \left[0 \quad 0 \quad \rho U_n U_\tau \quad \frac{1}{2} \rho U_\tau^2 U_n \right]^T \tag{24c}$$

Here, \mathbf{F}_c^n and \mathbf{F}_c^t are respectively the flux attributed to the normal velocity and tangential velocity, e is the potential energy of mean flow. From Chapman–Enskog analysis, and with the use of non-free parameter D1Q4 model, \mathbf{F}_c^n can be computed by:

$$\mathbf{F}_c^n = \sum_{i=1}^4 \xi_i \boldsymbol{\phi}_i f_i(0, t) \tag{25}$$

where:

$$f_i(0, t) = f^{eq}(0, t) + f^{neq}(0, t) = g_i(0, t) - \tau_0 [g_i(0, t) - g_i(-\xi_i \delta t, t - \delta t)], \tag{26}$$

ξ_i is the particle velocity in the i -direction, *i.e.*, $\xi_1 = d_1$, $\xi_2 = -d_1$, $\xi_3 = d_2$ and $\xi_4 = -d_2$. $\boldsymbol{\phi}_i$ stands for the moments:

$$\boldsymbol{\phi}_i = \left(1, \xi_i, 0, \frac{1}{2} \xi_i^2 + e_p \right)^T \tag{27}$$

Here, e_p is the potential energy of particles; $f_i(0, t)$ is the distribution function at the cell interface. $g_i(0, t)$ and $g_i(-\xi_i \delta t, t - \delta t)$ are respectively the equilibrium distribution function at the cell interface and the surrounding point of the cell interface. The non-equilibrium part in Equation (26) is viewed as the numerical dissipation since only the inviscid flux is computed by LBFS. τ_0 is the dimensionless collision time, which can be regarded as the weight of the numerical dissipation [40,41]. Substituting Equation (26) into Equation (25), we have the final expression of \mathbf{F}_c^n as follows:

$$\mathbf{F}_c^n = (1 - \tau_0) \mathbf{F}_c^{n,I} + \tau_0 \mathbf{F}_c^{n,II} \tag{28}$$

where:

$$\mathbf{F}_c^{n,l} = \sum_{i=1}^4 \xi_i \boldsymbol{\varphi}_a g_i(0, t) \quad \text{and} \quad \mathbf{F}_c^{n,r} = \sum_{i=1}^4 \xi_i \boldsymbol{\varphi}_a g_i(-\xi_i \delta t, t - \delta t) \tag{29}$$

To evaluate the flux attributed to the tangential velocity, one of the feasible ways can be expressed as:

$$\mathbf{F}_c^t(3) = \sum_{i=1,3} \xi_i f_i(0, t) U_\tau^L + \sum_{i=2,4} \xi_i f_i(0, t) U_\tau^R \tag{30a}$$

$$\mathbf{F}_c^t(4) = \frac{1}{2} \left[\sum_{i=1,3} \xi_i f_i(0, t) (U_\tau^L)^2 + \sum_{i=2,4} \xi_i f_i(0, t) (U_\tau^R)^2 \right] \tag{30b}$$

where, U_τ^L and U_τ^R are the tangential velocity at the left and right side of cell interface, respectively. From \mathbf{F}_c^* , the actual inviscid flux \mathbf{F}_c in Equation (20) in the Cartesian coordinate system can be obtained by a transformation [41]:

$$\mathbf{F}_c = [F_c^*(1) \quad F_c^*(2)n_x - F_c^*(3)n_y \quad F_c^*(2)n_y + F_c^*(3)n_x \quad F_c^*(4)] \tag{31}$$

Equation (31) forms the LBFS for inviscid compressible flows. It can be treated as a Riemann solver developed by 1D compressible lattice Boltzmann model. As for the viscous flux \mathbf{F}_v , it is still approximated by conventional finite difference schemes in this work.

5.2. Evaluation of Inviscid Flux by LBFS

To compute $\mathbf{F}_c^{n,l}$, the equilibrium distribution function at the cell interface should be computed in advance. As the equilibrium distribution function is the function of conservative variables, we just need to calculate the conservative variables at the cell interface first. The conservative variables at the cell interface in the local-coordinate system can be expressed as:

$$\mathbf{W}^* = [\rho \quad \rho U_n \quad \rho U_\tau \quad \rho E]^T = \mathbf{W}^n + \mathbf{W}^t \tag{32a}$$

$$\mathbf{W}^n = \left[\rho \quad \rho U_n \quad 0 \quad \rho \left(\frac{1}{2} U_n^2 + e \right) \right]^T \tag{32b}$$

$$\mathbf{W}^t = \left[0 \quad 0 \quad \rho U_\tau \quad \frac{1}{2} \rho U_\tau^2 \right]^T \tag{32c}$$

Like Equation (19), the conservative variables at the cell interface attributed to the normal velocity can be evaluated by:

$$\mathbf{W}^n = \sum_{i=1}^4 \boldsymbol{\varphi}_i g_i(-\xi_i \delta t, t - \delta t) \tag{33}$$

It is assumed that a local Riemann problem is formed at the cell interface. Then, the equilibrium distribution function $g_i(-\xi_i \delta t, t - \delta t)$ can be given by:

$$g_i(-\xi_i \delta t, t - \delta t) = \begin{cases} g_i^L & \text{if } i = 1, 3 \\ g_i^R & \text{if } i = 2, 4 \end{cases} \tag{34}$$

where g_i^L and g_i^R are the equilibrium distribution functions at the left and right sides of cell interface as shown in Figure 3. In addition, the conservative variables attributed to the tangential velocity can be approximated by:

$$W^t(3) = \sum_{i=1,3} g_i(-\xi_i \delta t, t - \delta t) U_\tau^L + \sum_{i=2,4} g_i(-\xi_i \delta t, t - \delta t) U_\tau^R \tag{35a}$$

$$W^t(4) = \frac{1}{2} \left[\sum_{i=1,3} g_i(-\xi_i \delta t, t - \delta t) (U_\tau^L)^2 + \sum_{i=2,4} g_i(-\xi_i \delta t, t - \delta t) (U_\tau^R)^2 \right] \tag{35b}$$

Once the conservative variables at the cell interface W^* are obtained, the flux attributed to the equilibrium distribution function at the cell interface $F_c^{n,I}$ can be calculated by substituting the above conservative variables directly into the expression of inviscid flux. As for the flux $F_c^{n,II}$, it can be obtained by substituting Equation (34) into Equation (29) directly.

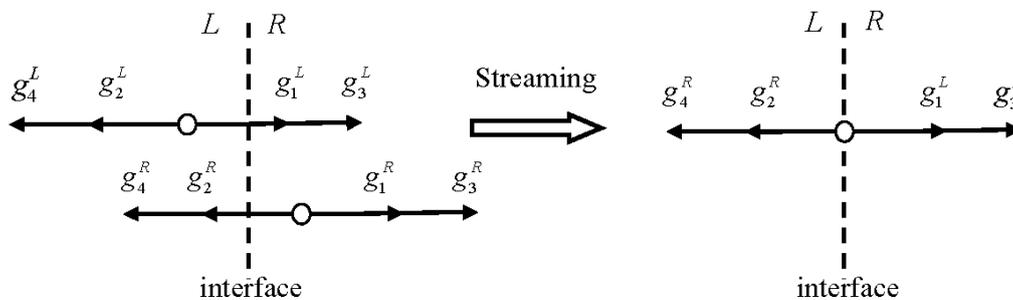


Figure 3. Streaming process of D1Q4 model at the cell interface.

6. Numerical Examples and Discussion

In this section, several benchmark cases will be studied to examine the performance of newly-developed LBFS. In particular, lid-driven cavity flows and steady natural convection in a square cavity at high Rayleigh numbers of 10^7 and 10^8 will be simulated on non-uniform grids by the incompressible LBFS. Transonic flows around a staggered-biplane configuration will be studied on unstructured grids by using the compressible LBFS.

6.1. Lid-Driven Cavity Flows

As a benchmark problem, the classical lid-driven cavity flow has been studied extensively by many researchers [30–34]. This problem involves several geometrical and flow parameters: the length of the cavity L , the velocity of the lid U and the density and dynamic viscosity of the fluid ρ and μ . With these parameters, the Reynolds number is defined as $Re = \rho UL / \mu$. In present study, this problem will be solved by using the LBFS and the standard LBM to compare their accuracy, stability and efficiency in detail.

At first, the accuracy of the LBFS and LBM are examined by comparing the pressure and velocity profiles for lid-driven cavity flows at $Re = 5000$. In the computation, a non-uniform grid of 101×101 for the LBFS and uniform grid of 301×301 for LBM are applied. Note that much finer grid is required

for the LBM to maintain numerical stability at this Reynolds number. Figure 4 compares the pressure and velocity profiles obtained by these two methods. It can be seen that excellent agreements have been achieved. Note that the LBFS uses about one-ninth of the total grid points adopted by the LBM, which shows its advantage and capability in applying non-uniform grids. It may also be noted that both LBFS and LBM have the second order of accuracy and quantitatively similar solutions will be obtained when the same grids are applied [8–10].

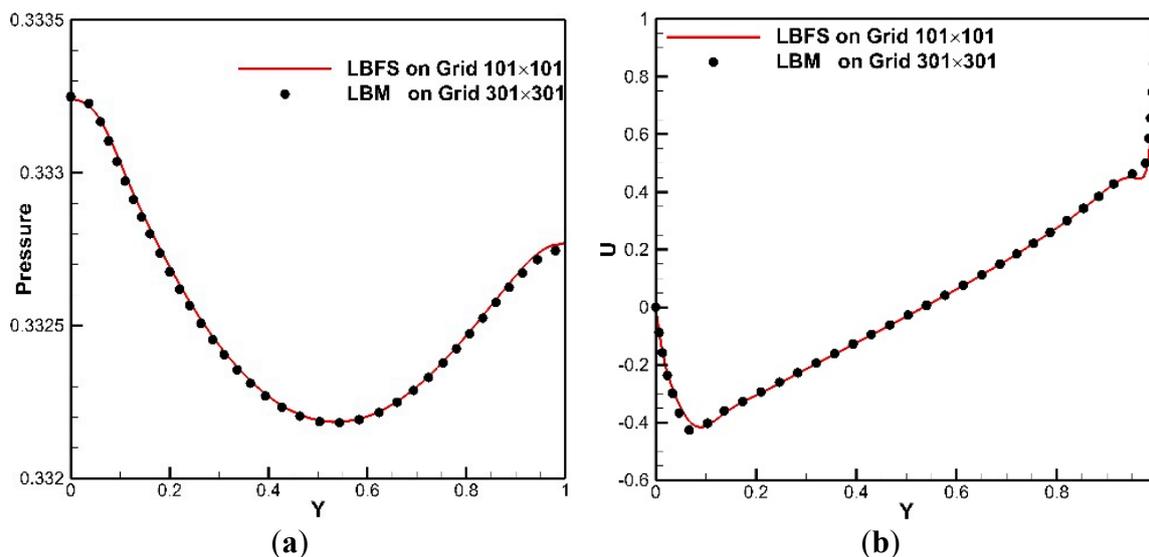


Figure 4. Comparison of pressure and velocity profiles on the vertical centerline for lid-driven cavity flows at $Re = 5000$. (a) Pressure; (b) Velocity.

After that, the pure stability of the LBFS and LBM is investigated without considering numerical accuracy. This test is conducted by simulating the lid-driven cavity flows at Reynolds numbers from 100 to 5000. The minimum grids to get stable solution required respectively by the LBFS and LBM are recorded and shown in Table 1. It can be seen that, with the increase of Reynolds number, the LBM requires more grid points to maintain stability while the LBFS only needs a grid size of 4×4 for all cases considered. This feature indicates that the LBFS is more stable than LBM. In addition, the stability of these two methods can be further examined by comparing numerical solutions for pressure. Figure 5 compares the pressure contours of lid-driven cavity flows at Reynolds number of 5000. A grid size of 101×101 is applied by the LBFS and 301×301 is applied by the LBM. It is obvious that the results of LBM for pressure field have substantial unphysical oscillations at the top left, top right and bottom right corners while those of the LBFS are smooth all over the flow domain. This phenomenon indicates that the stability of LBFS is superior to that of the LBM.

Table 1. Minimum required mesh resolution of the lattice Boltzmann flux solver (LBFS) and lattice Boltzmann method (LBM) for lid-driven cavity flows at different Reynolds numbers.

Re	100	1000	5000	7500
LBGK	5×5	39×39	219×219	341×341
LBFS	4×4	4×4	4×4	4×4

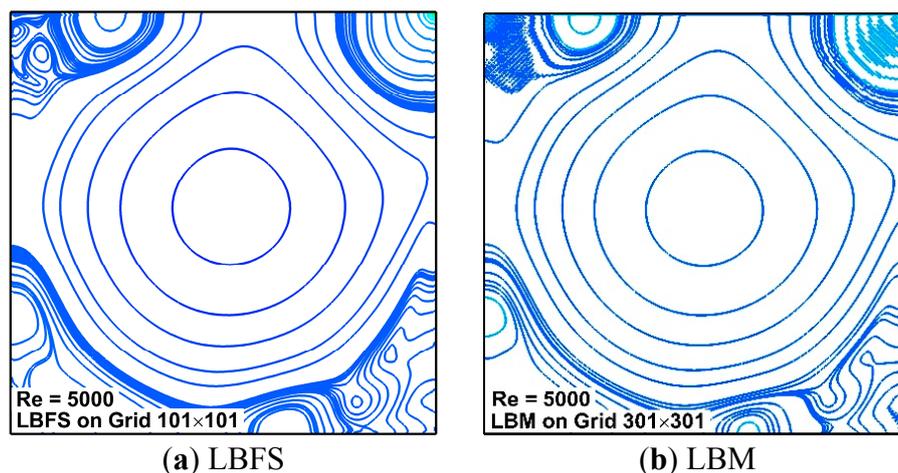


Figure 5. Comparison of pressure contours for lid-driven cavity flows at $Re = 5000$. (a) lattice Boltzmann flux solver (LBFS); (b) lattice Boltzmann method (LBM).

Moreover, the efficiency of the LBFS and LBM are investigated by considering two cases. The first one is their efficiency on the same uniform grid of 301×301 by simulating the lid-driven cavity flow at $Re = 5000$. The other case is to examine the computational effort of the LBFS on non-uniform grids when its results agree well with those of the LBM. Table 2 compares the CPU time consumed by each solver in these two cases. As can be seen, when applied on the same grid, the LBFS takes about 3.8 times the computational time and 47% of the virtual memory that are required by the LBM. This is mainly attributed to the fact that, in the LBFS, interpolations of physical quantities are performed at each cell for flux reconstruction, which degrades its efficiency. On the other hand, when the solution of the LBFS compares well with that of LBM as shown in Figure 4, the LBFS only needs a non-uniform grid of 101×101 . As compared with those of LBM, only about 16.6% of computational time and 5.3% of virtual memory are consumed by the LBFS. This indicates that the LBFS needs less computational resources and can be more efficient for applications on non-uniform grids.

Table 2. Comparison of performance of the lattice Boltzmann flux solver (LBFS) and lattice Boltzmann method (LBM) on different grids for 2D lid-driven cavity flows at $Re = 5000$.

Method	Grid Size	Time (s)	Memory (Mbytes)
LBFS	301×301 (Uniform)	20,838	11.05
	101×101 (Non-uniform)	909	1.23
LBM	301×301 (Uniform)	5458	23.5

6.2. Natural Convection in a Square Cavity at High Rayleigh Number

Natural convection in a square cavity is a benchmark case for validating various numerical methods [35,37,42]. For instance, Peng *et al.* [42] simulated this problem at Rayleigh numbers from $Ra = 10^3$ to 10^6 to validate their simplified SRT-based thermal LB model. Guo *et al.* [37] studied this problem to examine their D2Q4 thermal LB model. Contrino *et al.* [35] validated their MRT-based thermal LB model by simulating this flow at high Rayleigh numbers of 10^7 and 10^8 . Recently, with SRT-based D2Q9 thermal LB model, the LBFS [9] has also been validated through its application to simulate this problem from $Ra = 10^3$ to 10^6 . It is noticed that, to study this problem, all versions of LBM

restrict their computations on uniform grids. However, non-uniform grids may be more accurate and efficient to capture thin boundary layers, especially for flows at high Rayleigh numbers. This provides a good chance to examine the performance of the LBFS with D2Q4 thermal LB model.

The flow pattern of this problem is characterized by two normalized parameters: the Prandtl number Pr and the Rayleigh number Ra , which can be respectively defined as:

$$Pr = \frac{\nu}{\chi}, \quad Ra = \frac{g\beta \cdot \Delta T \cdot L^3}{\nu \cdot \chi} = \frac{V_c^2 \cdot L^2}{\nu \cdot \chi} \quad (36)$$

where L is the length of the square cavity, ΔT is the temperature difference between the hot and cold walls and $V_c = \sqrt{g\beta L \cdot \Delta T}$ is the characteristic thermal velocity. The flow parameters are set as follows: $L = 1$, $V_c = 0.1$, $Pr = 0.71$. Two high Rayleigh numbers of $Ra = 10^7$ and 10^8 are considered. Non-uniform grids with different grid sizes (201×201 and 301×301 for $Ra = 10^7$; 301×301 and 401×401 for $Ra = 10^8$) are applied. To quantify the result, the mean Nusselt number $Nu_{1/2}$ along the line of $x = L/2$ is computed and compared:

$$Nu_{1/2} = \frac{L}{\chi \cdot \Delta T} \cdot \frac{1}{L} \oint_{x=L/2} \left(uT - \chi \frac{\partial T}{\partial x} \right) dl \quad (37)$$

Tables 3 and 4 show the maximum absolute value of the stream-function $|\varphi|_{\max}$ and its position, the mean Nusselt number $Nu_{1/2}$ along the vertical centerline, the maximum u -velocity U_{\max} along $x = L/2$ and its vertical position, the maximum v -velocity V_{\max} along $y = L/2$ and its horizontal position. The numerical results of Contrino *et al.* [35] obtained by the MRT-LBM and those of Quere [43] obtained by a high order pseudo-spectral method are also included for comparison. With the increase of the grid size, the present solution is closer to the benchmark solution of Quere [43] for all cases considered. In particular, the relative error of $|\varphi|_{\max}$ between the present solution on the finest grid and those of Quere [43] is within 0.26% and that of $Nu_{1/2}$ is within 0.08%. This indicates that both the strength of the flow field represented by the stream-function and the heat transfer rate represented by the mean Nusselt number are well predicted by the LBFS. Figures 6 and 7 show the streamlines and isotherms at Rayleigh numbers of 10^7 and 10^8 respectively. At $Ra = 10^7$, a large clock-wise recirculation is formed and attached to all walls. Both the flow and temperature boundary layers close to the hot and cold walls are very thin. The temperature at the same height of the cavity is almost a constant near central area. As Ra is increased to 10^8 , the boundary layer separates near the bottom left and top right region. Figure 8 shows the u -velocity along the vertical centerline and v -velocity along the horizontal centerline. As can be seen, the velocity component v is almost zero in a large zone except for those near the wall. This indicates that vertical convection in the central area can be very weak and heat conduction dominates this region. This is consistent with the observation that the temperature on the same latitude is almost a constant in the central area of the cavity.

Table 3. Comparison of representative quantities for natural convection in a square cavity at $Ra = 10^7$.

Grid Size	Present		Contrino <i>et al.</i> [35]			Quere [43]
	201 ²	301 ²	379 ²	1019 ²	1531 ²	
$ \varphi _{\max}$	30.165	30.164	30.349	30.310	30.185	30.165
x	0.0868	0.0857	0.0848	0.0856	0.0857	0.86
y	0.5545	0.5559	0.5578	0.5562	0.5559	0.556
$Nu_{1/2}$	16.550	16.543	16.526	16.523	16.523	16.52
U_{\max}	148.17	148.84	148.48	148.57	148.58	148.59
y	0.8788	0.8789	0.8794	0.8793	0.8793	0.879
V_{\max}	699.19	699.91	699.11	699.27	699.31	699.18
x	0.0204	0.0216	0.0214	0.0213	0.0213	0.021

Table 4. Comparison of representative quantities for natural convection in a square cavity at $Ra = 10^8$.

Grid Size	Present		Contrino <i>et al.</i> [35]			Quere [43]
	301	401 ²	379 ²	1019 ²	1531 ²	
$ \varphi _{\max}$	53.955	53.893	54.870	54.106	53.953	53.85
x	0.0482	0.4760	0.0469	0.0478	0.0480	0.048
y	0.5536	0.5528	0.5594	0.5545	0.5533	0.553
$Nu_{1/2}$	30.353	30.301	30.257	30.229	30.227	30.225
U_{\max}	316.07	323.65	315.08	320.74	321.37	321.9
x	0.9267	0.9288	0.9239	0.9273	0.9276	0.928
V_{\max}	2221.1	2222.9	2221.4	2222.1	2222.3	2222
y	0.0118	0.1192	0.0121	0.0120	0.0120	0.012

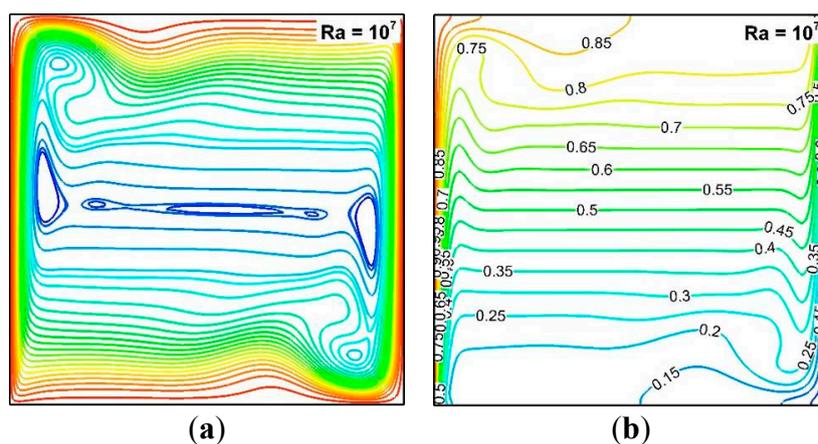


Figure 6. Streamlines and isotherms for natural convection at $Ra = 10^7$. (a) Streamlines; (b) Isotherms.

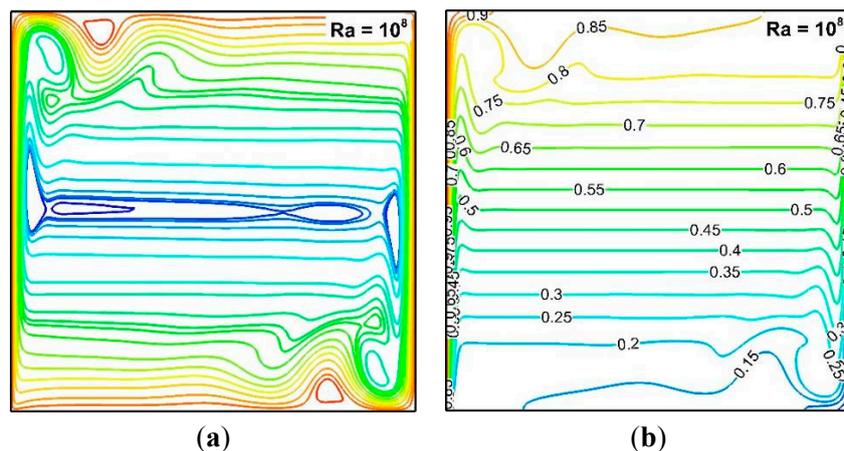


Figure 7. Streamlines and isotherms for natural convection at $Ra = 10^8$. (a) Streamlines; (b) Isotherms.

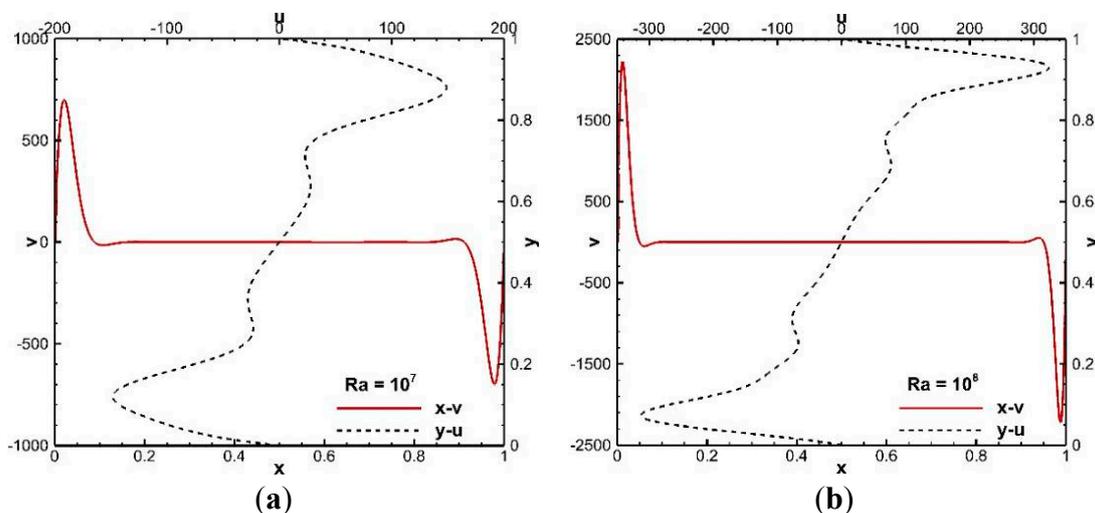


Figure 8. Illustration of the velocity profiles at the centerlines for natural convection at $Ra = 10^7$ and 10^8 . (a) $Ra = 10^7$; (b) $Ra = 10^8$.

6.3. Inviscid and Viscous Transonic Flows Around a Staggered-Biplane Configuration

To validate the compressible LBFS for simulation of flows with complex geometry, the inviscid and viscous transonic flows around a staggered-biplane configuration are simulated. This test example is taken from the work of Jawahar and Kamath [44]. It comprises two NACA0012 airfoils, staggered by half a chord length in the pitchwise as well as chordwise directions. At first, the inviscid flow with the free-stream Mach number of 0.7 and the angle of attack of 0 degree is simulated. In the test, the unstructured grid containing 256 points on each airfoil and 36,727 triangular cells in the computational domain is utilized, and its partial view is shown in Figure 9a. The pressure contours obtained from present scheme are shown in Figure 9b. It can be seen from the figure that, a strong normal shock is formed between two airfoils and near the trailing edge of bottom airfoil, which is in line with those observed by Jawahar and Kamath [44] and Lerat and Wu [45]. Figure 10 shows the pressure coefficient distribution on the airfoil surface computed by present scheme. Also displayed in this figure are the

results of Jawahar and Kamath [44] and Lerat and Wu [45]. Clearly, the results of current scheme agree well with the published data.

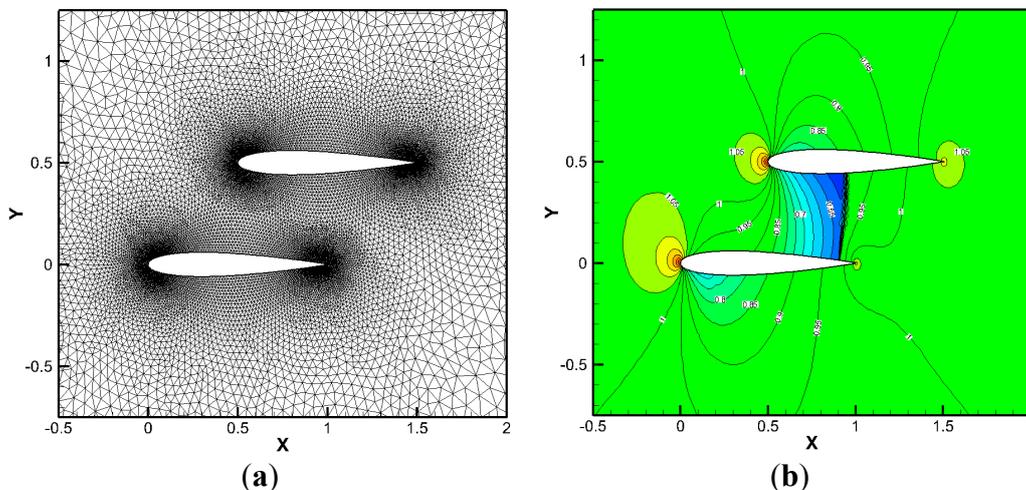


Figure 9. Partial view of computational mesh and pressure contours for inviscid biplane configuration. (a) Computational mesh; (b) Pressure contours.

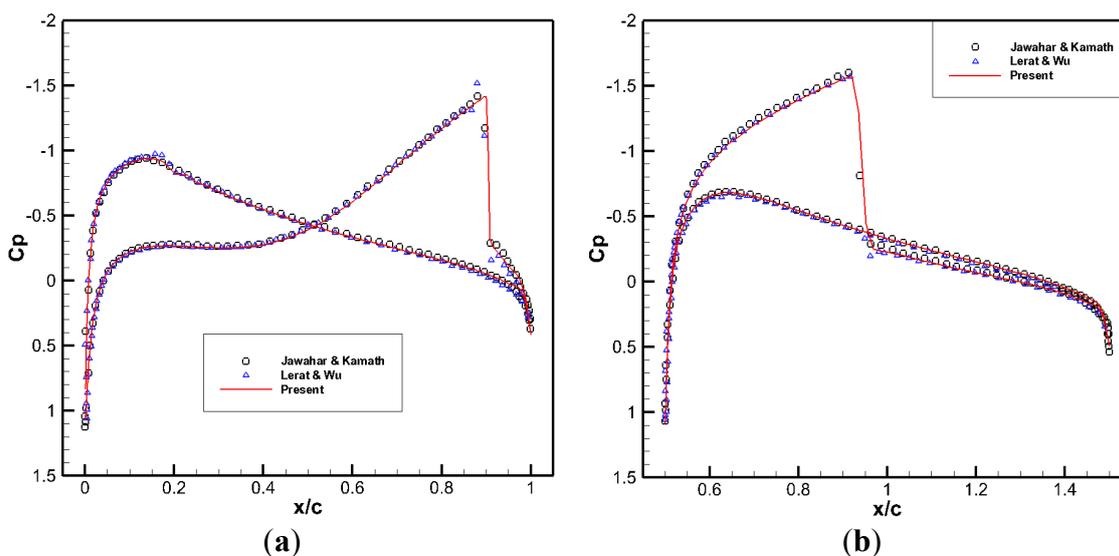


Figure 10. Comparison of pressure coefficient distribution on the surface of bottom and top airfoils for inviscid biplane configuration. (a) Bottom airfoil; (b) Top airfoil.

In addition, the viscous flow with the free-stream Mach number of 0.8, the Reynolds number of 500 and the angle of attack of 10 degree is simulated. In the simulation, the unstructured grid containing 512 points on each airfoil and 65,861 cells in the computational domain is used, and its partial view is shown in Figure 11a. Figure 11b shows the streamline pattern obtained from present scheme. From this figure, it can be observed that the separation region on the upper surface of top airfoil reveals two vortices. This observation is consistent with the results reported in [44]. As pointed out by Jawahar and Kamath [44], the secondary vortex is introduced by the bottom airfoil. The comparison of pressure coefficient and skin friction coefficient distributions on the airfoil surface obtained by present scheme

with those reported in [44] is shown in Figure 12. Once again, the results obtained by present scheme compare well with the reference data.

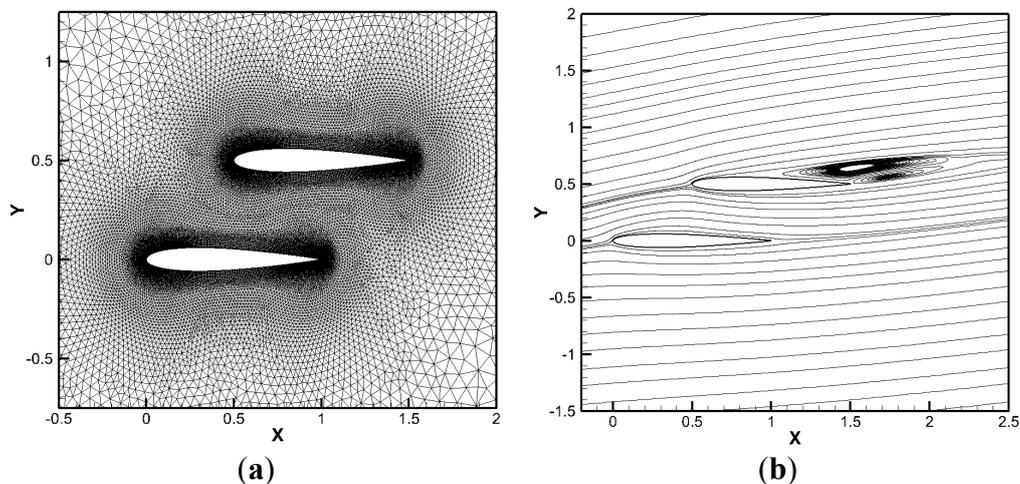


Figure 11. Partial view of computational mesh and streamline pattern for viscous biplane configuration. (a) Computational mesh; (b) Streamline pattern.

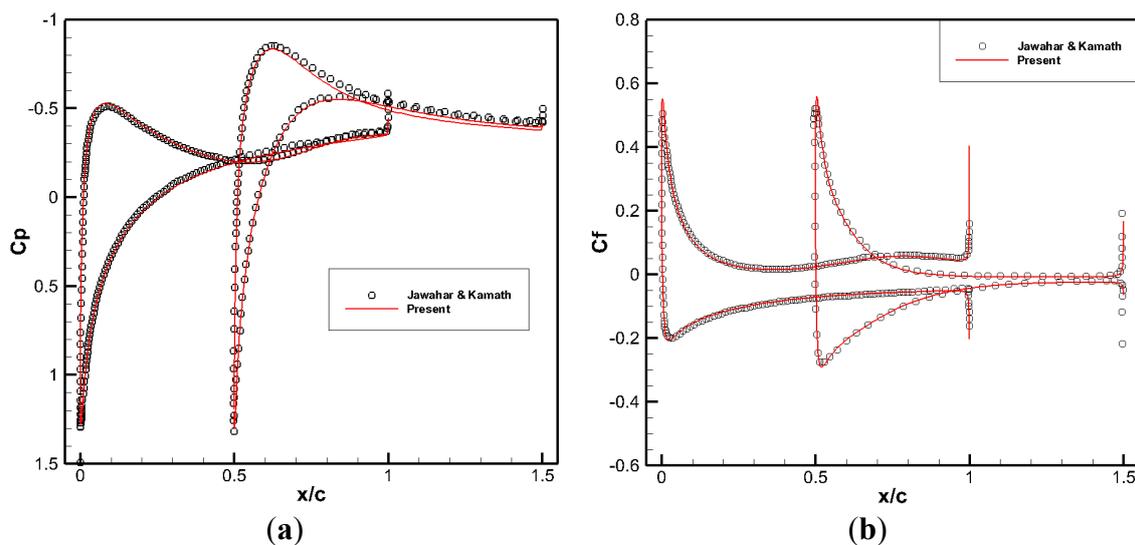


Figure 12. Comparison of pressure coefficient and skin friction coefficient distributions for viscous biplane configuration. (a) Pressure coefficient distribution; (b) Skin friction distribution.

7. Conclusions

As a finite-volume solver, the LBFS directly updates macroscopic flow variables at cell centers by solving macroscopic governing equations. Its fluxes are reconstructed locally at each interface through lattice moments of particle distribution functions, in which the relationships obtained from the Chapman–Enskog theory are applied. During local reconstruction, the LBM is applied locally in one streaming time step, which is different from the global application of the conventional LBM. As a consequence, the LBFS is able to combine the advantages of the finite volume method and the LBM.

In this work, the historic development from the LBM to the LBFS is briefly introduced and their relationships with the macroscopic conservation laws are also described through the multi-scale Chapman–Enskog analysis. The major contribution of this work is to refine and examine three different versions of the LBFS, proposed respectively for isothermal, thermal and compressible flows. In particular, the accuracy, stability and efficiency of the isothermal LBFS are compared with the LBM in detail. The LBFS for temperature field is simplified by the D2Q4 model, which reduces computational effort as compared with that by D2Q9 model. The LBFS for compressible flows is improved by incorporating non-equilibrium effects into the process for inviscid flux reconstruction, in which numerical dissipation can be controlled through a switch function.

Several benchmark problems, including lid-driven cavity flows, natural convection in a square cavity at high Rayleigh numbers of 10^7 and 10^8 and transonic flows around a staggered-biplane configuration, have been carried out to examine the solvers. Numerical results show that the LBFS is able to obtain comparable solutions with much less non-uniform grid points and its efficiency can be greatly improved. It is also shown that LBFS is much more stable than LBM and does not generate unphysical pressure oscillations for lid-driven cavity flows. With the application of one-dimensional compressible lattice Boltzmann model, the LBFS can be effectively applied for simulation of compressible flows on unstructured grids.

Author Contributions

The authors contributed equally to the research and writing of this article. All authors have read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. McNamara, G.R.; Zanetti, G. Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. *Phys. Rev. Lett.* **1988**, *61*, 2332–2335.
2. Higuera, F.J.; Succi, S.; Benzi, R. Lattice gas-dynamics with enhanced collisions. *Europhys. Lett.* **1989**, *9*, 345–349.
3. Chen, S.Y.; Chen, H.D.; Martinez, D.; Matthaeus, W. Lattice Boltzmann model for simulation of magnetohydrodynamics. *Phys. Rev. Lett.* **1991**, *67*, 3776–3779.
4. Qian, Y.H.; D’Humières, D.; Lallemand, P. Lattice BGK models for Navier–Stokes equation. *Europhys. Lett.* **1992**, *17*, 479–484.
5. Mei, R.; Shyy, W. On the Finite Difference-Based Lattice Boltzmann Method in Curvilinear Coordinates. *J. Comput. Phys.* **1998**, *143*, 426–448.
6. Hejranfar, K.; Ezzatneshan, E. Implementation of a high-order compact finite-difference lattice Boltzmann method in generalized curvilinear coordinates. *J. Comput. Phys.* **2014**, *267*, 28–49.
7. Zarghami, A.; Ubertini, S.; Succi, S. Finite-volume lattice Boltzmann modeling of thermal transport in nanofluids. *Comput. Fluids* **2013**, *77*, 56–65.

8. Wang, Y.; Shu, C.; Huang, H.B.; Teo, C.J. Multiphase lattice Boltzmann flux solver for incompressible multiphase flows with large density ratio. *J. Comput. Phys.* **2015**, *280*, 404–423.
9. Wang, Y.; Shu, C.; Teo, C.J. Thermal lattice Boltzmann flux solver and its application for simulation of incompressible thermal flows. *Comput. Fluids* **2014**, *94*, 98–111.
10. Shu, C.; Wang, Y.; Teo, C.J.; Wu, J. Development of Lattice Boltzmann Flux Solver for Simulation of Incompressible Flows. *Adv. Appl. Math. Mech.* **2014**, *6*, 436–460.
11. Yang, L.M.; Shu, C.; Wu, J. A moment conservation-based non-free parameter compressible lattice Boltzmann model and its application for flux evaluation at cell interface. *Comput. Fluids* **2013**, *79*, 190–199.
12. Guo, Z.; Shi, B.; Wang, N. Lattice BGK Model for Incompressible Navier–Stokes Equation. *J. Comput. Phys.* **2000**, *165*, 288–306.
13. Shan, X.; Chen, H. Lattice Boltzmann model for simulating flows with multiple phases and components. *Phys. Rev. E* **1993**, *47*, 1815–1819.
14. Lee, T.; Lin, C.-L. A stable discretization of the lattice Boltzmann equation for simulation of incompressible two-phase flows at high density ratio. *J. Comput. Phys.* **2005**, *206*, 16–47.
15. He, X.; Chen, S.; Zhang, R. A Lattice Boltzmann Scheme for Incompressible Multiphase Flow and Its Application in Simulation of Rayleigh–Taylor Instability. *J. Comput. Phys.* **1999**, *152*, 642–663.
16. Li, Q.; He, Y.L.; Wang, Y.; Tao, W.Q. Coupled double-distribution-function lattice Boltzmann method for the compressible Navier–Stokes equations. *Phys. Rev. E* **2007**, *76*, 056705.
17. Chen, F.; Xu, A.; Zhang, G.; Li, Y.; Succi, S. Multiple-relaxation-time lattice Boltzmann approach to compressible flows with flexible specific-heat ratio and Prandtl number. *Europhys. Lett.* **2010**, *90*, doi:10.1209/0295-5075/90/54003.
18. He, Y.-L.; Liu, Q.; Li, Q. Three-dimensional finite-difference lattice Boltzmann model and its application to inviscid compressible flows with shock waves. *Physica A* **2013**, *392*, 4884–4896.
19. Nie, X.; Doolen, G.D.; Chen, S. Lattice-Boltzmann Simulations of Fluid Flows in MEMS. *J. Stat. Phys.* **2002**, *107*, 279–289.
20. Lim, C.Y.; Shu, C.; Niu, X.D.; Chew, Y.T. Application of lattice Boltzmann method to simulate microchannel flows. *Phys. Fluids* **2002**, *14*, 2299–2308.
21. Meng, J.; Zhang, Y. Accuracy analysis of high-order lattice Boltzmann models for rarefied gas flows. *J. Comput. Phys.* **2011**, *230*, 835–849.
22. Lallemand, P.; Luo, L.-S. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E* **2000**, *61*, 6546–6562.
23. Niu, X.D.; Shu, C.; Chew, Y.T. A lattice Boltzmann BGK model for simulation of micro flows. *Europhys. Lett.* **2004**, *67*, 600–606.
24. Yuan, H.-Z.; Niu, X.-D.; Shu, S.; Li, M.; Yamaguchi, H. A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating a flexible filament in an incompressible flow. *Comput. Math. Appl.* **2014**, *67*, 1039–1056.
25. Yuan, H.-Z.; Shu, S.; Niu, X.-D.; Li, M.; Hu, Y. A Numerical Study of Jet Propulsion of an Oblate Jelly fish Using a Momentum Exchange-Based Immersed Boundary-Lattice Boltzmann Method. *Adv. Appl. Math. Mech.* **2014**, *6*, 307–326.

26. Ubertini, S.; Succi, S. Recent advances of Lattice Boltzmann techniques on unstructured grids. *Prog. Comput. Fluid Dyn. Int. J.* **2005**, *5*, 85–96.
27. Rossi, N.; Ubertini, S.; Bella, G.; Succi, S. Unstructured lattice Boltzmann method in three dimensions. *Int. J. Numer. Methods Fluids* **2005**, *49*, 619–633.
28. Patankar, S.V.; Spalding, D.B. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transf.* **1972**, *15*, 1787–1806.
29. Sun, D.L.; Qu, Z.G.; He, Y.L.; Tao, W.Q. An efficient segregated algorithm for incompressible fluid flow and heat transfer problems-IDEAL (Inner Doubly Iterative Efficient Algorithm for Linked Equations) Part I: Mathematical formulation and solution procedure. *Numer. Heat Transf. B* **2008**, *53*, doi:10.1080/10407790701632543.
30. Bruneau, C.-H.; Saad, M. The 2D lid-driven cavity problem revisited. *Comput. Fluids* **2006**, *35*, 326–348.
31. Kalita, J.C.; Gupta, M.M. A streamfunction–velocity approach for 2D transient incompressible viscous flows. *Int. J. Numer. Methods Fluids* **2010**, *62*, 237–266.
32. Lin, L.-S.; Chang, H.-W.; Lin, C.-A. Multi relaxation time lattice Boltzmann simulations of transition in deep 2D lid driven cavity using GPU. *Comput. Fluids* **2013**, *80*, 381–387.
33. Zhuo, C.; Zhong, C.; Cao, J. Filter-matrix lattice Boltzmann simulation of lid-driven deep-cavity flows, Part I—Steady flows. *Comput. Math. Appl.* **2013**, *65*, 1863–1882.
34. Zhuo, C.; Zhong, C.; Cao, J. Filter-matrix lattice Boltzmann simulation of lid-driven deep-cavity flows, Part II—Flow bifurcation. *Comput. Math. Appl.* **2013**, *65*, 1883–1893.
35. Contrino, D.; Lallemand, P.; Asinari, P.; Luo, L.-S. Lattice-Boltzmann simulations of the thermally driven 2D square cavity at high Rayleigh numbers. *J. Comput. Phys.* **2014**, *275*, 257–272.
36. Luan, H.-B.; Xu, H.; Chen, L.; Sun, D.-L.; He, Y.-L.; Tao, W.-Q. Evaluation of the coupling scheme of FVM and LBM for fluid flows around complex geometries. *Int. J. Heat Mass Transf.* **2011**, *54*, 1975–1985.
37. Guo, Z.; Shi, B.; Zheng, C. A coupled lattice BGK model for the Boussinesq equations. *Int. J. Numer. Methods Fluids* **2002**, *39*, 325–342.
38. Chen, L.; He, Y.-L.; Kang, Q.; Tao, W.-Q. Coupled numerical approach combining finite volume and lattice Boltzmann methods for multi-scale multi-physicochemical processes. *J. Comput. Phys.* **2013**, *255*, 83–105.
39. Chen, L.; Luan, H.; Feng, Y.; Song, C.; He, Y.-L.; Tao, W.-Q. Coupling between finite volume method and lattice Boltzmann method and its application to fluid flow and mass transport in proton exchange membrane fuel cell. *Int. J. Heat Mass Transf.* **2012**, *55*, 3834–3848.
40. Yang, L.M.; Shu, C.; Wu, J.; Zhao, N.; Lu, Z.L. Circular function-based gas-kinetic scheme for simulation of inviscid compressible flows. *J. Comput. Phys.* **2013**, *255*, 540–557.
41. Yang, L.M.; Shu, C.; Wu, J. A three-dimensional explicit sphere function-based gas-kinetic flux solver for simulation of inviscid compressible flows. *J. Comput. Phys.* **2015**, *295*, 322–339.
42. Peng, Y.; Shu, C.; Chew, Y.T. Simplified thermal lattice Boltzmann model for incompressible thermal flows. *Phys. Rev. E* **2003**, *68*, 026701.
43. Le Quéré, P. Accurate solutions to the square thermally driven cavity at high Rayleigh number. *Comput. Fluids* **1991**, *20*, 29–41.

44. Jawahar, P.; Kamath, H. A High-Resolution Procedure for Euler and Navier–Stokes Computations on Unstructured Grids. *J. Comput. Phys.* **2000**, *164*, 165–203.
45. Lerat, A.; Wu, Z.N. Stable Conservative Multidomain Treatments for Implicit Euler Solvers. *J. Comput. Phys.* **1996**, *123*, 45–64.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).