Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **8**, No. 6, pp. 887-910

DOI: 10.4208/aamm.2015.m1172
December 2016

A Hybrid Lattice Boltzmann Flux Solver for Simulation of Viscous Compressible Flows

L. M. Yang[1,2], C. Shu[3,*] and J. Wu[2]

[1] *State Key Laboratory of Aerodynamics, China Aerodynamics Research and Development Center, Mianyang 621000, China*
[2] *Department of Aerodynamics, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Yudao Street, Nanjing 210016, China*
[3] *Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore*

Received 26 June 2015; Accepted (in revised version) 8 September 2015

Abstract. In this paper, a hybrid lattice Boltzmann flux solver (LBFS) is proposed for simulation of viscous compressible flows. In the solver, the finite volume method is applied to solve the Navier-Stokes equations. Different from conventional Navier-Stokes solvers, in this work, the inviscid flux across the cell interface is evaluated by local reconstruction of solution using one-dimensional lattice Boltzmann model, while the viscous flux is still approximated by conventional smooth function approximation. The present work overcomes the two major drawbacks of existing LBFS [28–31], which is used for simulation of inviscid flows. The first one is its ability to simulate viscous flows by including evaluation of viscous flux. The second one is its ability to effectively capture both strong shock waves and thin boundary layers through introduction of a switch function for evaluation of inviscid flux, which takes a value close to zero in the boundary layer and one around the strong shock wave. Numerical experiments demonstrate that the present solver can accurately and effectively simulate hypersonic viscous flows.

AMS subject classifications: 76M12

Key words: Lattice Boltzmann flux solver, hybrid, 1D lattice Boltzmann model, switch function.

1 Introduction

As fast development of computer hardware and numerical approaches, computational fluid dynamics (CFD) plays a more and more important role in industrial applications. The CFD is to apply a numerical method to solve governing equations (Navier-Stokes

[*] Corresponding author.
Email: mpeshuc@nus.edu.sg (C. Shu), yangliming_2011@126.com (L. M. Yang)

equations) on the computer. Currently, there are a number of numerical methods available [1–17]. Among them, the finite volume method (FVM) [7] is widely used. The application of FVM is in line with application of physical conservation laws to a control volume. Thus, its major advantage is to keep numerical conservation of physical quantities. It is also flexible and suitable for solving problems with complex geometry. The key issue in applying FVM is to develop an appropriate numerical scheme, which is also known as flux solver, for evaluation of inviscid and viscous fluxes at cell interface.

In the conventional application of FVM to simulate viscous compressible flows, evaluation of inviscid flux and viscous flux is made by different ways. The viscous flux is evaluated by using a smooth function approximation such as polynomial approximation, while inviscid flux is computed by using various upwind schemes such as Roe scheme [10], van Leer scheme [11], and AUSM (Advection Upstream Splitting Method) scheme [12]. Roe scheme is widely used in simulation of compressible viscous flows due to its high accuracy for boundary layers and good resolution for shock waves. However, for the simulation of hypersonic flows, the Roe scheme often exhibits carbuncle phenomenon and induces numerical instability [13]. Van Leer scheme has a good performance in solving Euler equations, but may smear out the solution of boundary layers and also lead to inaccurate stagnation and wall temperature for simulation of viscous compressible flows [14].

An alternative approach for evaluation of fluxes is the Boltzmann equation-based flux solver, which is also known as gas kinetic scheme. Different from the conventional CFD approaches, the Boltzmann equation-based schemes evaluate fluxes by local reconstruction of solution for the Boltzmann equation at a cell interface. This kind of scheme can be used to simulate both incompressible and compressible flows. Kinetic Flux Vector Splitting (KFVS) scheme [15] and gas-kinetic Bhatnagar-Gross-Krook (BGK) scheme [16,17] are the commonly-used Boltzmann equation-based flux solvers. Basically, for the KFVS scheme, the collisionless Boltzmann equation is solved in the gas evolution stage, and the collision process is controlled by a numerical time step. As a result, the KFVS scheme usually gives poorer results than those obtained from Roe scheme [10] and AUSM scheme [12]. Unlike KFVS scheme, the gas-kinetic BGK scheme considers the particle collisions during the gas evolution stage by the BGK model. As a consequence, the dissipation in the streaming process is controlled by the collision time rather than by the numerical time step. Numerical results showed that the gas-kinetic BGK scheme can accurately simulate both inviscid and viscous flows [17, 18]. On the other hand, as far as we know, most of existing gas kinetic schemes are based on the Maxwellian distribution function [15–18]. Due to complexity of the Maxwellian function, these schemes are usually more complicated and less efficient than the traditional numerical schemes [10–12].

Recently, lattice Boltzmann method (LBM) [19, 20] receives more and more attention due to its kinetic nature, simplicity, and easy implementation. However, the conventional LBM is only limited to the simulation of incompressible flows since its equilibrium distribution function is approximated from truncated Taylor series expansion of Maxwellian function in terms of Mach number. To simulate compressible flows by LBM, one has to use new equilibrium distribution functions. Some efforts have been made in the literature [21, 24]. It is noted that the new equilibrium functions are usually very complicated, especially for multi-dimensional problems. In addition, to get the stable solution, one often needs to apply an upwind scheme to solve the discrete velocity Boltzmann equation (DVBE) [21–27]. The solution process of DVBE is quite tedious and computational efficiency is low. To develop a more efficient solver for compressible flows, a lattice Boltzmann flux solver (LBFS for short) for simulation of inviscid compressible flows is proposed by Ji et al. [28], and further improved by Yang et al. [29, 30] and Shu et al. [31]. In LBFS, FVM is applied to discretize Euler equations, and the inviscid flux at the cell interface is reconstructed from local solution of one-dimensional (1D) compressible lattice Boltzmann model. It has been demonstrated that the LBFS provides good positivity property for simulation of flows with strong shock wave and expansion wave [30]. On the other hand, it should be indicated that the existing LBFS has two major shortcomings. One is due to the use of 1D model for reconstruction of local solution. The 1D model can make the scheme very simple, but it can only be applied along the normal direction to the cell interface. Since the tangential effect is not properly considered, the existing LBFS can only be applied to simulate inviscid flows. In other words, it only evaluates inviscid flux at the cell interface. To solve viscous flows, evaluation of viscous flux should also be taken into account. The second drawback is from the use of distribution function to compute the inviscid flux. In the existing LBFS [28–31], the distribution functions at cell interface streamed from neighboring points are directly used to compute the inviscid flux. On the other hand, from Chapman-Enskog expansion analysis [19, 20], it is known that the inviscid flux can be fully determined by equilibrium distribution function. From this process, it is clear that the non-equilibrium part of the distribution function at the cell interface introduces numerical dissipation for calculation of inviscid flux. The introduced numerical dissipation is very useful to capture the strong shock wave, but it will affect the true solution in smooth regions such as in boundary layers. To capture both strong shock waves and boundary layers, the numerical dissipation has to be controlled. That is, numerical dissipation should only be introduced in the region around strong shock waves. How to remove the above two drawbacks of existing LBFS motivates the present work.

In this work, following Chapman-Enskog analysis, a hybrid LBFS is proposed for simulation of viscous flows. In the hybrid solver, the inviscid flux is evaluated by LBFS (local reconstruction of solution using 1D lattice Boltzmann model), and viscous flux is computed by the smooth function approximation. The numerical dissipation is controlled by introducing a switch function which ranges from 0 to 1. In the smooth region such as in boundary layer, the switch function takes a value close to zero, while around the shock wave, it tends to one. The present hybrid solver is validated by its application to solve some test examples including the Couette flow, Blasius boundary layer, laminar flows over NACA0012 airfoil and hypersonic flow around a half of cylinder. Numerical results demonstrate that the hybrid LBFS can effectively simulate compressible viscous flows with strong shock wave and boundary layer.

2 Methodology

2.1 Navier-Stokes equations discretized by finite volume method

In general, the integral form of Navier-Stokes equations without source term can be written as

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{W}d\Omega+\oint_{\Gamma}(\mathbf{F}_c-\mathbf{F}_v)dS=0, \tag{2.1}$$

where, the conservative variables $\mathbf{W}$, inviscid flux $\mathbf{F}_c$ and viscous flux $\mathbf{F}_v$ are given by

$$\mathbf{W}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix},\qquad \mathbf{F}_c=\begin{bmatrix}\rho U_n\\ \rho u U_n+n_x p\\ \rho v U_n+n_y p\\ (\rho E+p)U_n\end{bmatrix},\qquad \mathbf{F}_v=\begin{bmatrix}0\\ n_x\tau_{xx}+n_y\tau_{xy}\\ n_x\tau_{yx}+n_y\tau_{yy}\\ n_x\Theta_x+n_y\Theta_y\end{bmatrix}. \tag{2.2}$$

Here, ρ and p are the density and pressure of the 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. U_n represents the normal velocity, which is defined as the scalar product of the velocity vector and the unit normal vector, i.e.,

$$U_n=n_x u+n_y v. \tag{2.3}$$

E is the total energy of the mean flow, which is defined as

$$E=e+\frac{1}{2}(u^2+v^2). \tag{2.4}$$

Here, $e=p/[(\gamma-1)\rho]$ is the potential energy of the mean flow, where γ is the specific heat ratio. 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.

In the FVM, the integral of the fluxes in Eq. (2.1) is approximated by a sum of the fluxes crossing the cell interface, i.e.,

$$\frac{d\mathbf{W}_I}{dt}=-\frac{1}{\Omega_I}\sum_{i=1}^{N_f}(\mathbf{F}_{ci}-\mathbf{F}_{vi})S_i, \tag{2.5}$$

where I is the index of a control volume, Ω_I and N_f represent the volume and the number of the faces of the control volume I, respectively. S_i denotes the area of the ith face of the control volume. The key for solving Eq. (2.5) is to evaluate the inviscid flux $\mathbf{F}_c$ and viscous flux $\mathbf{F}_v$. In line with the elliptic nature of the viscous flux, $\mathbf{F}_v$ is usually approximated by smooth functions such as polynomial. Therefore, the only remaining problem is to calculate the inviscid flux $\mathbf{F}_c$.

2.2 Evaluation of inviscid flux by local solution of 1D lattice Boltzmann model

To effectively simulate inviscid compressible flows, Shu and his coworkers [28–31] proposed a lattice Boltzmann flux solver (LBFS). In the LBFS, the local solution of one-dimensional (1D) lattice Boltzmann (LB) model is used to construct inviscid flux solver at the cell interface, while the FVM is used to discretize the macroscopic governing equations. In the existing LBFS [28–31], the numerical dissipation takes effect uniformly in the whole computational domain, and is too large in smooth regions to affect the true solution for simulation of viscous flows. In the present work, we start from the Chapman-Enskog analysis [19,20,32,33] and consider both the equilibrium part and non-equilibrium part of the distribution function at the cell interface. The non-equilibrium part is viewed as numerical dissipation since LBFS is only used to evaluate inviscid flux in this work. For evaluation of the non-equilibrium part, a simple difference scheme is applied, i.e., the non-equilibrium part is computed by the difference of equilibrium distribution functions at the cell interface and its surrounding point. At the same time, the weight of the non-equilibrium part is controlled by introducing a switch function. Before we move on, the 1D LB model used in this work will be introduced first.

Most of existing 1D LB models [21–23] involve a number of user-specified parameters, which could significantly affect the performance of LBFS. To remove this drawback, Yang et al. [29,30] proposed a platform to develop non-free parameter LB models from conservation forms of moments. A typical such model is the non-free parameter D1Q4 model as shown in Fig. 1, which contains six unknowns (4 equilibrium distribution functions g_1, g_2, g_3, g_4 and 2 lattice velocities d_1, d_2). Using conservation forms of moments, g_1, g_2, g_3, g_4 and d_1, d_2 can be given as [29,30],

$$g_1=\frac{\rho(-d_1d_2^2-d_2^2u+d_1u^2+d_1c^2+u^3+3uc^2)}{2d_1(d_1^2-d_2^2)}, \tag{2.6a}$$

$$g_2=\frac{\rho(-d_1d_2^2+d_2^2u+d_1u^2+d_1c^2-u^3-3uc^2)}{2d_1(d_1^2-d_2^2)}, \tag{2.6b}$$

$$g_3=\frac{\rho(d_1^2d_2+d_1^2u-d_2u^2-d_2c^2-u^3-3uc^2)}{2d_2(d_1^2-d_2^2)}, \tag{2.6c}$$

$$g_4=\frac{\rho(d_1^2d_2-d_1^2u-d_2u^2-d_2c^2+u^3+3uc^2)}{2d_2(d_1^2-d_2^2)}, \tag{2.6d}$$

$$d_1=\sqrt{u^2+3c^2-\sqrt{4u^2c^2+6c^4}}, \tag{2.6e}$$

$$d_2=\sqrt{u^2+3c^2+\sqrt{4u^2c^2+6c^4}}, \tag{2.6f}$$

where, c represents the peculiar velocity of particles defined as $c=\sqrt{Dp/\rho}$, D denotes the space dimension, $D=1$ means one-dimension.

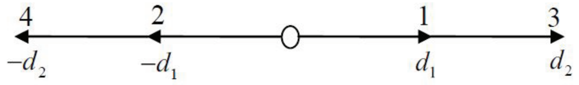

Figure 1: Distribution of discrete lattice velocities for D1Q4 model.

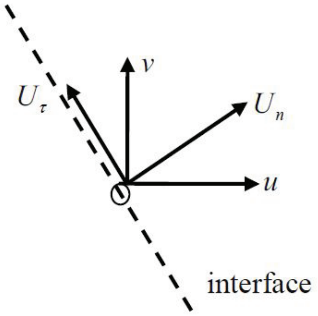

Figure 2: Application of 1D model to 2D case.

After obtaining the equilibrium distribution functions and lattice velocities, the physical conservation laws give

$$\rho=\sum_{i=1}^{4}g_i, \tag{2.7a}$$

$$\rho u=\sum_{i=1}^{4}g_i\xi_i, \tag{2.7b}$$

$$\rho uu+p=\sum_{i=1}^{4}g_i\xi_i\xi_i, \tag{2.7c}$$

$$\rho E=\sum_{i=1}^{4}g_i\Big(\frac{1}{2}\xi_i\xi_i+e_p\Big), \tag{2.7d}$$

$$(\rho E+p)u=\sum_{i=1}^{4}g_i\Big(\frac{1}{2}\xi_i\xi_i+e_p\Big)\xi_i, \tag{2.7e}$$

where, 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 $\xi_4=-d_2$. e_p is the potential energy of particles, $e_p=[1-\frac{D}{2}(\gamma-1)]e$. It is necessary to point out that when multi-dimensional problems are considered, the above 1D model needs to be applied along the normal direction of cell interface [29,30]. For example, for the 2D case, as shown in Fig. 2, we can use the normal velocity U_n to replace u and set n_x as 1 in the 1D model (2.7).

After introducing the non-free parameter D1Q4 model, we now can construct a corresponding LBFS by using this model. For the general case, D1Q4 model is applied along the normal direction of a cell interface. Suppose that this cell interface is located at $x=0$.

Then the inviscid flux at the cell interface can be computed by

$$\mathbf{F}^*_{c,i+1/2}=\Big[\rho U_n \quad \rho U_n U_n+p \quad \Big(\rho\Big(\frac{1}{2}U_nU_n+e\Big)+p\Big)U_n\Big]^T=\sum_{i=1}^{4}\xi_i\boldsymbol{\varphi}_\alpha f_i(0,t), \tag{2.8}$$

where ξ_i has been defined previously, $\boldsymbol{\varphi}_\alpha$ stands for the moments

$$\boldsymbol{\varphi}_\alpha=\Big(1,\xi_i,\frac{1}{2}\xi_i^2+e_p\Big)^T, \tag{2.9}$$

and $f_i(0,t)$ is the distribution function at the cell interface. The notation $*$ means numerical value. Usually, $f_i(0,t)$ consists of equilibrium part $f_i^{eq}(0,t)$ and non-equilibrium part $f_i^{neq}(0,t)$, which can be written as

$$f_i(0,t)=f_i^{eq}(0,t)+f_i^{neq}(0,t). \tag{2.10}$$

From Chapman-Enskog analysis [19, 20, 32, 33], to recover Navier-Stokes equations by Boltzmann equation, the non-equilibrium part f_i^{neq} can be written as

$$f_i^{neq}(0,t)=-\tau\Big(\frac{\partial g_i}{\partial t}+\xi_i\frac{\partial g_i}{\partial x}\Big)\Big|_{(0,t)}. \tag{2.11}$$

By applying Taylor series expansion in time and physical space, the above equation can be further simplified to

$$f_i^{neq}(0,t)=-\frac{\tau}{\delta t}[g_i(0,t)-g_i(-\xi_i\delta t,t-\delta t)]+\mathcal{O}(\delta t), \tag{2.12}$$

where $g_i(0,t)=f_i^{eq}(0,t)$ is the equilibrium distribution function at the cell interface, and $g_i(-\xi_i\delta t,t-\delta t)$ is the equilibrium distribution function at the surrounding point of the cell interface. Substituting Eq. (2.12) into Eq. (2.10) gives

$$f_i(0,t)=g_i(0,t)-\tau_0[g_i(0,t)-g_i(-\xi_i\delta t,t-\delta t)]+\mathcal{O}(\delta t), \tag{2.13}$$

where $\tau_0=\tau/\delta t$ is the dimensionless collision time and δt is the streaming time step. It is known from the Chapman-Enskog analysis that the equilibrium part of the distribution function contributes to the inviscid flux while its non-equilibrium part contributes to the viscous flux. As LBFS is only used to evaluate inviscid flux in this work, $f_i^{neq}(0,t)$ can be viewed as numerical dissipation, and τ_0 can be regarded as the weight of the numerical dissipation. In other words, to apply LBFS for evaluation of inviscid flux with minimum numerical dissipation, τ_0 should be taken as 0. This kind of scheme is noted as Scheme I in this work. As will be shown in this paper, Scheme I provides very accurate results for boundary layer flows as very little numerical dissipation is introduced. However, this scheme often exhibits oscillation or even diverges for simulation of hypersonic flows with strong shock waves. To capture strong shock waves with stable solution, we need to introduce numerical dissipation and set a non-zero value to τ_0. In the existing LBFS [28–31], τ_0 is actually taken as 1. In fact, the existing LBFS is based on collisionless Boltzmann equation. For this case, the distribution function $f_i(0,t)$ at the cell interface is streamed from the surrounding points initialized by the equilibrium state, which can be given as [29–31]

$$f_i(0,t)=g_i(-\xi_i\delta t,t-\delta t), \tag{2.14}$$

Eq. (2.14) is exactly the same as Eq. (2.13) when we set τ_0 as 1. In this work, we note this scheme as Scheme II. Obviously, Scheme II introduces a large numerical dissipation. That is why the existing LBFS can well simulate hypersonic inviscid flows with very high Mach number [29–31]. On the other hand, as shown in this paper, the large numerical dissipation pollutes the solution in smooth regions, and the scheme cannot provide accurate solution in the boundary layer. To capture both the strong shock waves and thin boundary layers accurately, we need to carefully control the numerical dissipation. This can be done by introducing a switch function to control τ_0. In other words, τ_0 should be taken as a value close to zero in the boundary layer and one around the strong shock wave. In this work, the scheme involving the switch function is termed Scheme III. The details of the three schemes will be shown below.

Scheme I: Evaluation of inviscid flux by setting $\tau_0=0$

By taking $\tau_0=0$, Eq. (2.13) can be reduced to

$$f_i(0,t)=g_i(0,t). \tag{2.15}$$

In Eq. (2.15), in order to obtain the equilibrium distribution function $g_i(0,t)$, the conservative variables at the cell interface should be computed in advance. According to relationships (2.7), at the cell interface, the density, momentum in the normal direction and energy attributed to normal velocity can be computed by

$$\mathbf{W}^*_{i+1/2}=\begin{bmatrix}\rho & \rho U_n & \frac{1}{2}\rho U_nU_n+\rho e\end{bmatrix}^T=\sum_{i=1}^{4}\boldsymbol{\varphi}_\alpha f_i(0,t)=\sum_{i=1}^{4}\boldsymbol{\varphi}_\alpha g_i(0,t). \tag{2.16}$$

According to the compatibility condition [33], the non-equilibrium part of distribution function has no contribution in calculation of conservative variables. That is,

$$\sum_{i=1}^{4}\boldsymbol{\varphi}_\alpha f_i^{neq}(0,t)=-\sum_{i=1}^{4}\boldsymbol{\varphi}_\alpha\frac{\tau}{\delta t}[g_i(0,t)-g_i(-\xi_i\delta t,t-\delta t)]=0. \tag{2.17}$$

Substituting Eq. (2.17) into Eq. (2.16) gives

$$\mathbf{W}^*_{i+1/2}=\sum_{i=1}^{4}\boldsymbol{\varphi}_\alpha g_i(0,t)=\sum_{i=1}^{4}\boldsymbol{\varphi}_\alpha g_i(-\xi_i\delta t,t-\delta t). \tag{2.18}$$

From Eq. (2.18), it can be seen that the conservative variables at the cell interface can be computed from $g_i(-\xi_i\delta t, t-\delta t)$, which is the function of the conservative variables at the surrounding point of the cell interface. Like the conventional upwind schemes, it is assumed that a local Riemann problem is formed at the cell interface. Thus, the equilibrium distribution function $g_i(-\xi_i\delta t, t-\delta t)$ can be given according to the location of $-\xi_i\delta t$. Specifically, $g_i(-\xi_i\delta t, t-\delta t)$ can be written as

$$g_i(-\xi_i\delta t, t-\delta t) = \begin{cases} g_i^L, & \text{if} \quad -\xi_i\delta t \le 0, \\ g_i^R, & \text{if} \quad -\xi_i\delta t > 0. \end{cases} \tag{2.19}$$

Since the streaming time step δt is always positive, Eq. (2.19) can be rewritten as

$$g_i(-\xi_i\delta t, t-\delta t) = \begin{cases} g_i^L, & \text{if} \quad \xi_i \ge 0, \\ g_i^R, & \text{if} \quad \xi_i < 0, \end{cases} \tag{2.20}$$

where, g_i^L and g_i^R are the equilibrium distribution functions at the left and right side of cell interface. For the non-free parameter D1Q4 model, Eq. (2.20) can be further reduced to

$$g_i(-\xi_i\delta t, t-\delta t) = \begin{cases} g_i^L, & \text{if} \quad i = 1,3, \\ g_i^R, & \text{if} \quad i = 2,4. \end{cases} \tag{2.21}$$

The above process is illustrated in Fig. 3 and the details can be found in [29]. Substituting Eq. (2.21) into Eq. (2.18), we can get

$$\mathbf{W}^*_{i+1/2} = \sum_{i=1,3} \boldsymbol{\varphi}_\alpha g_i^L + \sum_{i=2,4} \boldsymbol{\varphi}_\alpha g_i^R. \tag{2.22}$$

Since the 1D LB model is used, in Eq. (2.22), only the conservative variables attributed to the normal velocity are obtained. To evaluate the tangential velocity at the cell interface, one of the feasible ways can be expressed as

$$(\rho U_\tau)^* = \sum_{i=1}^{4} g_i \cdot U_\tau^* = \sum_{i=1,3} g_i^L \cdot U_\tau^L + \sum_{i=2,4} g_i^R \cdot U_\tau^R, \tag{2.23}$$

where, U_τ^*, U_τ^L and U_τ^R are the tangential velocity at the cell interface, and the left and right side of cell interface, respectively. With Eqs. (2.22) and (2.23), we can obtain the primitive variables ρ^*, U_n^*, U_τ^* and p^* in a straightforward way. Furthermore, substituting these variables into Eqs. (2.6a)-(2.6f), we can get the equilibrium distribution function at the cell interface $g_i(0,t)$. Finally, the inviscid flux across the cell interface can be computed by substituting Eq. (2.15) into Eq. (2.8). An alternative but more efficient way is to

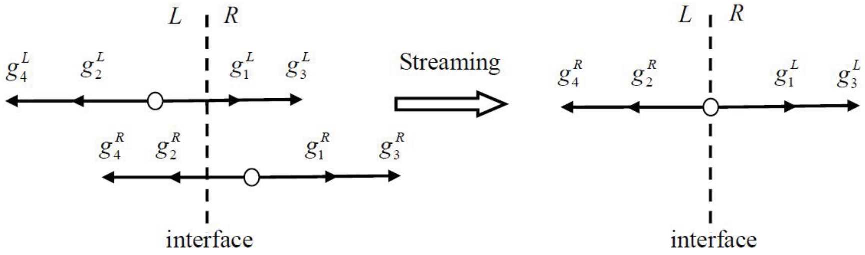

Figure 3: Streaming process of D1Q4 model at the cell interface.

substitute the above primitive variables directly into Eq. (2.2). As such, the inviscid flux at the cell interface $\mathbf{F}_{c,i+1/2}^{(I)}$ can be expressed as

$$\mathbf{F}_{c,i+1/2}^{(I)}=\begin{bmatrix}\rho U_n\\(\rho U_nU_n+p)n_x-\rho U_nU_\tau n_y\\(\rho U_nU_n+p)n_y+\rho U_nU_\tau n_x\\(\rho E+p)U_n\end{bmatrix}_{i+1/2}^{*}. \tag{2.24}$$

Note that in Eq. (2.24), the relationships of $u=n_xU_n-n_yU_\tau$ and $v=n_xU_\tau+n_yU_n$ have been used. For the sake of discussion, we name this approach as Scheme I.

Since the distribution function at the cell interface is completely computed from the equilibrium part, the numerical dissipation in Scheme I is very little. As a consequence, it can provide accurate results for compressible viscous flows with weak shock waves. But it will encounter numerical instability for simulation of hypersonic flows due to the lack of numerical dissipation.

Scheme II: Evaluation of inviscid flux by setting $\tau_0=1$

When $\tau_0=1$ is considered, Eq. (2.13) is reduced to Eq. (2.14). Different from Scheme I, in Eq. (2.14), the distribution function at cell interface is determined by the equilibrium distribution function at the surrounding point of the cell interface $g_i(-\xi_i\delta t,t-\delta t)$. Since $g_i(-\xi_i\delta t,t-\delta t)$ has been determined by Eq. (2.21), the inviscid flux can be computed directly for this case. Substituting Eq. (2.14) into Eq. (2.8), we can obtain the mass flux, momentum flux in the normal direction and energy flux attributed to the normal velocity across the interface as

$$\mathbf{F}_{c,i+1/2}^{*}=\Big[\rho U_n \quad \rho U_nU_n+p \quad \Big(\rho\Big(\frac{1}{2}U_nU_n+e\Big)+p\Big)U_n\Big]^T=\sum_{i=1}^{4}\xi_i\boldsymbol{\varphi}_\alpha g_i(-\xi_i\delta t,t-\delta t). \tag{2.25}$$

And further substituting Eq. (2.21) into Eq. (2.25) gives

$$\mathbf{F}_{c,i+1/2}^{*}=\sum_{i=1}^{4}\xi_i\boldsymbol{\varphi}_\alpha g_i(-\xi_i\delta t,t-\delta t)=\sum_{i=1,3}\xi_i\boldsymbol{\varphi}_\alpha g_i^L+\sum_{i=2,4}\xi_i\boldsymbol{\varphi}_\alpha g_i^R. \tag{2.26}$$

In addition, the contribution of the tangential velocity to the momentum flux in the tangential direction and energy flux can be approximated by

$$(\rho U_n U_\tau)^* = \sum_{i=1}^{4} \xi_i g_i(-\xi_i \delta t, t-\delta t)\cdot U_\tau^* = \sum_{i=1,3} \xi_i g_i^L \cdot U_\tau^L + \sum_{i=2,4} \xi_i g_i^R \cdot U_\tau^R, \tag{2.27a}$$

$$(\rho U_n U_\tau^2)^* = \sum_{i=1}^{4} \xi_i g_i(-\xi_i \delta t, t-\delta t)\cdot (U_\tau^*)^2 = \sum_{i=1,3} \xi_i g_i^L \cdot (U_\tau^L)^2 + \sum_{i=2,4} \xi_i g_i^R \cdot (U_\tau^R)^2. \tag{2.27b}$$

With Eqs. (2.26)-(2.27b), we can obtain the total convective flux vectors across the cell interface as follows

$$\mathbf{F}_{c,i+1/2}^{(II)} = \begin{bmatrix} \sum\limits_{i=1}^{4} \xi_i g_i \\ \sum\limits_{i=1}^{4} \xi_i \xi_i g_i n_x - \sum\limits_{i=1}^{4} \xi_i g_i \cdot U_\tau^* n_y \\ \sum\limits_{i=1}^{4} \xi_i \xi_i g_i n_y + \sum\limits_{i=1}^{4} \xi_i g_i \cdot U_\tau^* n_x \\ \sum\limits_{i=1}^{4} \xi_i \left(\dfrac{1}{2}\xi_i \xi_i + e_p\right) g_i + \dfrac{1}{2}\sum\limits_{i=1}^{4} \xi_i g_i \cdot (U_\tau^*)^2 \end{bmatrix}, \tag{2.28}$$

where, $g_i = g_i(-\xi_i \delta t, t-\delta t)$. It can be seen that Eq. (2.28) is the same as the inviscid flux solver in [28–31]. In this work, this scheme is noted as Scheme II.

Since a large numerical dissipation is introduced by setting $\tau_0 = 1$, Scheme II can capture strong shock waves stably and it is more robust than Scheme I. However, since the weight of non-equilibrium part of distribution function is fixed with value 1, the numerical dissipation of Scheme II takes effect uniformly in the whole computational domain. As a result, the numerical dissipation of the scheme is too large in smooth regions to affect the true solution. This is undesirable in smooth regions.

Scheme III: Evaluation of inviscid flux by a variable switch function

As mentioned above, Scheme I has a good performance in capturing the thin boundary layer but it is usually incompetent for simulation of hypersonic flows, while Scheme II can capture strong shock waves stably but gives poor result for capturing the thin boundary layer. Naturally, one may expect to develop a new scheme which can combine the advantages of Scheme I and Scheme II. This motivates the development of Scheme III by introducing a switch function α defined as

$$\alpha = \tanh\left(C\frac{|p_L - p_R|}{p_L + p_R}\right), \tag{2.29}$$

where, $\tanh(x)$ is the hyperbolic tangent function, p_L and p_R are the pressure at the left and right side of cell interface. C is the amplification factor, and $C = 10$ is used in this work. From Eq. (2.29), it can be seen that α ranges from 0 to 1. As a result, the total inviscid flux across the cell interface can be written as

$$\mathbf{F}^{(III)}_{c,i+1/2}=(1-\alpha)\mathbf{F}^{(I)}_{c,i+1/2}+\alpha\mathbf{F}^{(II)}_{c,i+1/2}. \tag{2.30}$$

By comparing Eq. (2.30) with Eq. (2.13), it is found that τ_0 in Eq. (2.13) actually corresponds to α in Eq. (2.30). In other words, the value of τ_0 is no long treated as a constant as used in Scheme I and Scheme II. It is a variable in Scheme III. From Eq. (2.30), it can be observed that the inviscid flux $\mathbf{F}^{(III)}_{c,i+1/2}$ is close to that in Scheme I in smooth regions due to the small pressure difference at two sides of the cell interface and approaches that in Scheme II in the vicinity of strong shock wave.

On the other hand, it should be noted that the heat flux in the vicinity of stagnation point exhibits oscillations when Eq. (2.30) is applied to simulate hypersonic flows. This phenomenon can be observed in Case 4 of this work. The main reason may be that the switch function defined in Eq. (2.29) cannot well switch Scheme III into Scheme II around strong shock waves. To overcome this deficiency, a slight modification is applied to Eq. (2.29). This remedy is inspired by the construction of limiter of Barth and Jespersen [34]. The new form of switch function is defined as

$$\alpha^*=\max\{\alpha_L,\alpha_R\}, \tag{2.31}$$

where, α_L and α_R are the maximum values of the switch function of the left and right control volumes, respectively, which are defined as

$$\alpha_L=\max_{i=1,N_{fL}}\{\alpha_i\},\quad \alpha_R=\max_{i=1,N_{fR}}\{\alpha_i\}, \tag{2.32}$$

where, N_{fL} and N_{fR} are the number of the faces of the left and right control volumes, respectively. Accordingly, Eq. (2.30) is changed as

$$\mathbf{F}^{(III)}_{c,i+1/2}=(1-\alpha^*)\mathbf{F}^{(I)}_{c,i+1/2}+\alpha^*\mathbf{F}^{(II)}_{c,i+1/2}. \tag{2.33}$$

We name the above scheme as Scheme III* to distinguish it from Scheme III.

2.3 Evaluation of viscous flux by employing central difference scheme

In this section, we give a brief description on calculation of viscous flux. In Eq. (2.2), the viscous flux can be rewritten as

$$\mathbf{F_v}=\begin{bmatrix} 0 \\ n_x\tau_{xx}+n_y\tau_{xy} \\ n_x\tau_{yx}+n_y\tau_{yy} \\ n_x\left(u\tau_{xx}+v\tau_{xy}+k\dfrac{\partial T}{\partial x}\right)+n_y\left(u\tau_{yx}+v\tau_{yy}+k\dfrac{\partial T}{\partial y}\right)\end{bmatrix}, \tag{2.34}$$

where, k is the thermal conductivity. The stress tensor $\overline{\overline{\tau}}$ in 2D case is given by

$$\overline{\overline{\tau}}=\mu\begin{bmatrix}\frac{4}{3}\frac{\partial u}{\partial x}-\frac{2}{3}\frac{\partial v}{\partial y} & \frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\\ \frac{\partial u}{\partial y}+\frac{\partial v}{\partial x} & \frac{4}{3}\frac{\partial v}{\partial y}-\frac{2}{3}\frac{\partial u}{\partial x}\end{bmatrix}, \tag{2.35}$$

where, μ is the dynamic viscosity, which is determined from the Sutherland's law or the relationship of Reynolds number.

Due to elliptic nature of the viscous flux, we can evaluate $\mathbf{F_v}$ from variables averaged at the cell interface. At the cell interface, the values of the velocity components (u,v), the dynamic viscosity μ and the thermal conductivity k can be calculated by simple averaging technique, i.e.,

$$\phi_{i+1/2}=\frac{1}{2}(\phi_i+\phi_{i+1}), \tag{2.36}$$

where, ϕ stands for any of the above flow variables. Next, we need to compute the derivatives of the velocity components in Eq. (2.35) and of temperature in Eq. (2.34). Since the gradients of flow variable inside each control volume have been computed in the stage of reconstruction, unless otherwise stated, these gradients will be used to get the mean value at the cell interface, i.e.,

$$\nabla\phi_{i+1/2}=\frac{1}{2}(\nabla\phi_i+\nabla\phi_{i+1}). \tag{2.37}$$

It should be noted that the above approach may lead to strong odd-even decoupling in the simulation of hypersonic flows [35]. In Case 4 of this work, to well capture the strong shock waves, the finite difference scheme is applied to calculate the derivatives in Eqs. (2.34) and (2.35). The details can be found in [36].

3 Numerical examples

To validate the present solver, the Couette flow, laminar boundary layer, laminar flow over NACA0012 airfoil and hypersonic flow around a half of circular cylinder are simulated. Except for the case of hypersonic flow around a half of cylinder, the conservative variables at two sides of cell interface are interpolated from those at cell centers and the Venkatakrishnan's limiter [37] is used. For temporal discretisation of Eq. (2.5), three-stages Runge-Kutta method is applied. Unless otherwise stated, in all numerical examples reported in this work, the CFL number is set as 2 and the Prandtl number is taken as 0.72.

Case 1: Couette flow

Couette flow is a classical heat-transfer problem which can be used to examine the capability of a flow solver to describe the viscous heat dissipation. This problem can be considered as a viscous fluid flow between two infinite parallel plates separated by a distance of H. The bottom plate is stationary and the top one is moving at a speed u in the horizontal direction. The temperature at the bottom and top walls is fixed as T_0 and T_1 respectively. In a steady state, under the assumption of constant viscosity and heat conduction coefficient, the temperature profiles can be obtained as

$$T-T_0=\mathrm{Pr}\frac{u^2}{2c_p}\frac{y}{H}\Big(1-\frac{y}{H}\Big), \qquad \text{when} \quad T_0=T_1, \tag{3.1a}$$

$$\frac{T-T_0}{T_1-T_0}=\frac{y}{H}+\mathrm{Pr}\frac{Ec}{2}\frac{y}{H}\Big(1-\frac{y}{H}\Big), \qquad \text{when} \quad T_0\neq T_1, \tag{3.1b}$$

where, y is the distance from the bottom boundary, Pr is the Prandtl number, c_p is the specific heat ratio at constant pressure, and $Ec=u^2/c_p(T_1-T_0)$ is the Eckert number. In this case, we set H as $H=1$. At the inlet and outlet, a periodic boundary condition is implemented. In addition, the specific heat ratio γ is chosen as 1.4 for all simulations.

At first, the test cases with different Prandtl numbers of $\mathrm{Pr}=1$ and $\mathrm{Pr}=2$ and $u=u_0$, $T_1=T_0$ are simulated. A uniform mesh with 10×40 cells is used. The temperature profiles along the vertical central line obtained from Scheme I, Scheme II and Scheme III are shown in Fig. 4. It is observed that all the results of three schemes match well with the analytic data. In addition, we also simulated the cases with different top plate velocities of $u=u_0$ and $u=2u_0$, and $\mathrm{Pr}=2$, $T_1=(1+0.5)T_0$. In these tests, two kinds of uniform meshes with 10×40 cells and 10×80 cells are used. Fig. 5 shows the comparison of temperature profiles along the vertical central line obtained by three schemes and analytic solution. As shown in this figure, the results of three schemes agree fairly well with analytical data except that the result of Scheme II with mesh size of 10×40 and top plate velocity of $u=2u_0$ shows some deviation from analytical solution. However, when the

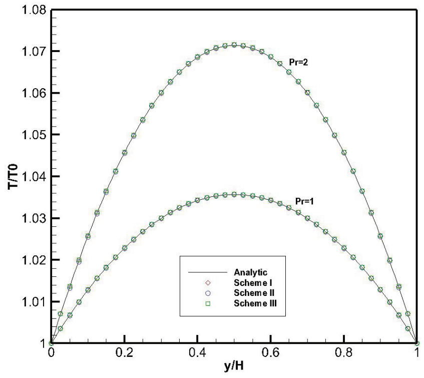

Figure 4: Comparison of temperature profiles along vertical central line for different Prandtl numbers.

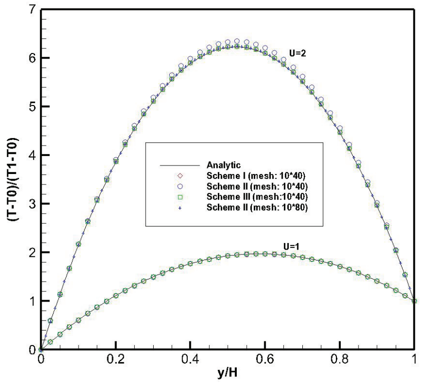

Figure 5: Comparison of temperature profiles along vertical central line for different moving velocities at top plate.

mesh size is refined to 10 × 80, the result of Scheme II can match excellently well with the analytical data. This finding clearly reveals that the numerical viscosity of Scheme II is larger than Scheme I and Scheme III, and it can be reduced by refining the mesh.

Case 2: Laminar boundary layer

The second case is the laminar boundary layer. For this case, the Reynolds number is taken as 10^5 and the free stream Mach number is set as 0.15. At first, a non-uniform mesh with 160 × 60 cells as shown in Fig. 6 is used. The u-velocity contours obtained from Scheme I at the steady state are shown in Fig. 7. Fig. 8 and Fig. 9 show the u-velocity (upper) and v-velocity (lower) profiles along the vertical lines of $x = 100,200$ and 250

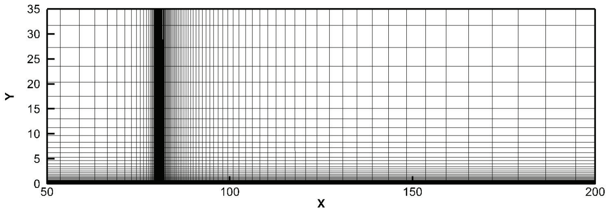

Figure 6: Computational mesh for laminar boundary layer flow (mesh: 160 × 60).

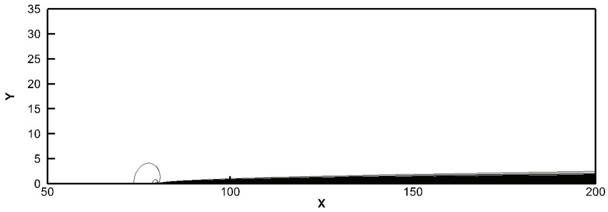

Figure 7: u-velocity contours of laminar boundary layer flow obtained by Scheme I (mesh: 160 × 60).

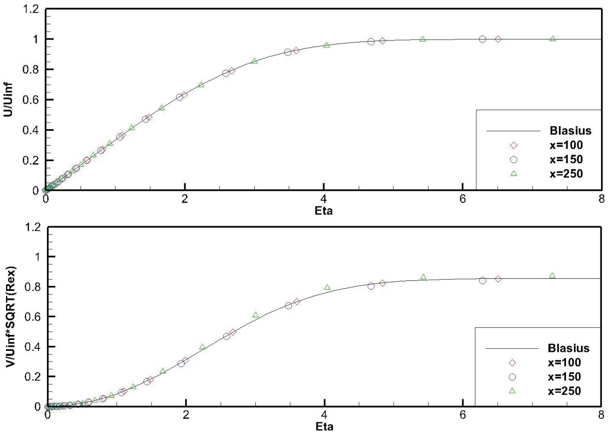

Figure 8: Comparison of u-velocity (upper) and v-velocity (lower) profiles along the vertical lines of $x=100,200$ and 250 for laminar boundary layer flow obtained by Scheme I (mesh: 160×60).

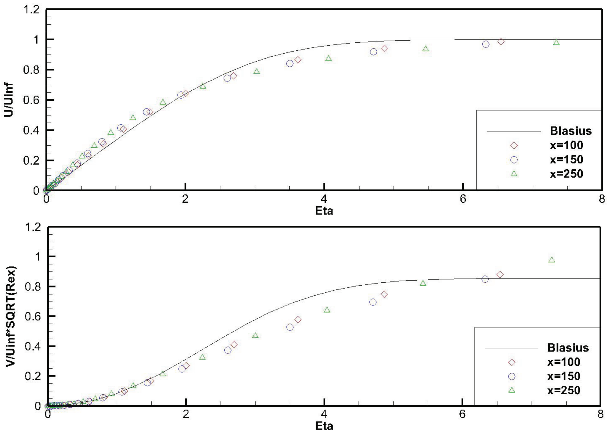

Figure 9: Comparison of u-velocity (upper) and v-velocity (lower) profiles along the vertical lines of $x=100,200$ and 250 for laminar boundary layer flow obtained by Scheme II (mesh: 160×60).

computed by Scheme I and Scheme II, respectively. Also displayed in these figures is the Blasius solution. As shown in these figures, Scheme I provides accurate solutions for laminar boundary layers, while Scheme II can hardly capture the correct shear layers. Note that, the results of Scheme III, which have not been shown in the figure to save the space, are essentially the same as those of Scheme I. The above results show again that the numerical dissipation of Scheme II is larger than Scheme I and Scheme III. Like the previous example, the numerical dissipation is usually related to the mesh spacing. To investigate this phenomenon, a refined mesh with 480×180 cells is used to re-solve the problem by Scheme II, and the resultant solution is shown in Fig. 10. It can be seen clearly that as mesh size is refined, Scheme II can also provide accurate solution for viscous flows. This is because the numerical dissipation is significantly reduced when the mesh size is refined.

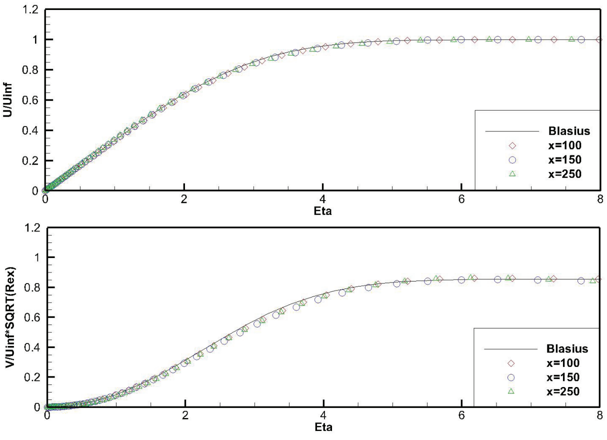

Figure 10: Comparison of u-velocity (upper) and v-velocity (lower) profiles along the vertical lines of $x=100,200$ and 250 for laminar boundary layer flow obtained by Scheme II (mesh: 480×180).

Case 3: Laminar flow over NACA0012 airfoil

The third test case is a laminar flow over NACA0012 airfoil. For this case, the free-stream Mach number is 0.8, the Reynolds number is 500 and the angle of attack is 10 degree. Unstructured grid with 10382 cells is used, and its partial view is shown in Fig. 11. Along the surface of airfoil, there are 150 grid points. The computational domain of $-10\leq x\leq 10$, $-10\leq y\leq 10$ is used. This is the test case A3 from GAMM workshop [38]. Fig. 12 shows the streamline patterns obtained from Scheme III (left) and given by Jawahar et al. [39]. It can be observed that both patterns are almost the same. The comparison of pressure coefficient (left) and skin friction coefficient (right) distributions on the airfoil surface obtained from three schemes is shown in Fig. 13. Also presented in the figure are

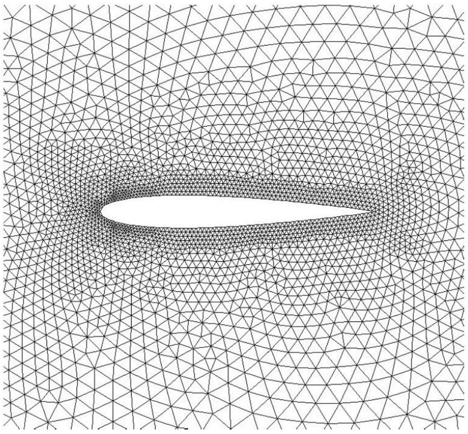

Figure 11: Partial view of the grid around NACA0012 airfoil.

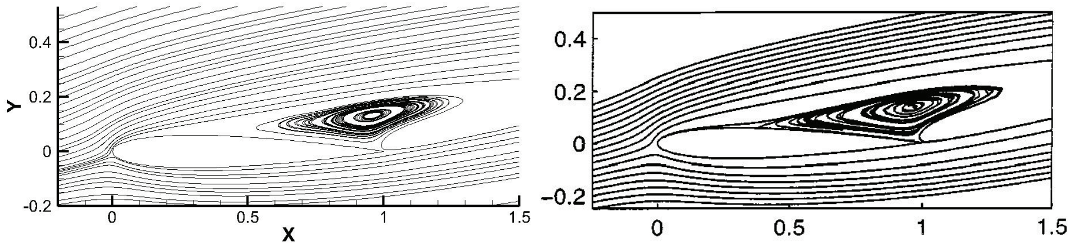

Figure 12: Streamline pattern around NACA0012 airfoil obtained by Scheme III (left) and given by Jawahar et al. [39] (right).

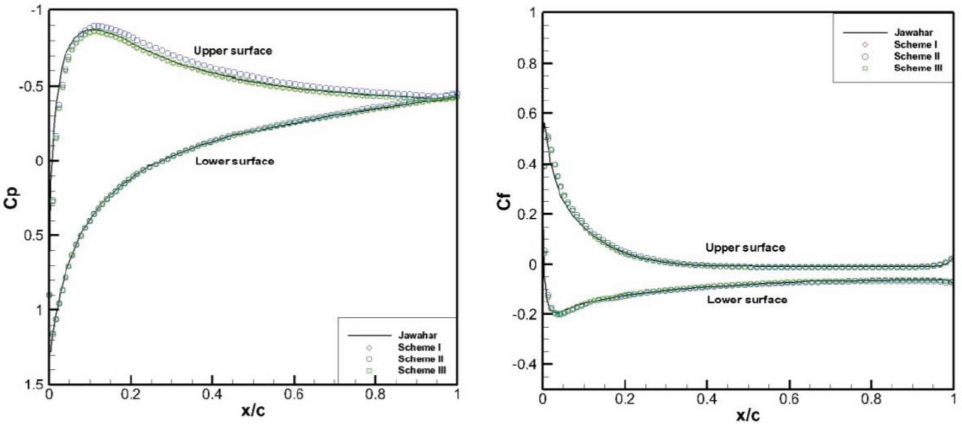

Figure 13: Comparison of pressure coefficient (left) and skin friction coefficient (right) distributions on the NACA0012 airfoil surface.

the results of [39]. As can be seen from the pressure coefficient distribution, the results of Scheme I and Scheme III are closer to the result of [39] as compared with result of Scheme II. In the meantime, the lift and drag coefficients obtained by three schemes are compared in Table 1 with reference data of GAMM [38] and Jawahar et al. [39]. It can be observed that the results of Scheme I and Scheme III are close each other, and they basically agree well with the reference data. Once again, this test example shows that the numerical dissipation in Scheme I and Scheme III has little effect in simulation of viscous compressible flows.

Table 1: Comparison of lift coefficient and drag coefficient for NACA0012 case.

References	C_{d_p}	C_{d_f}	$C_{d_{total}}$	$C_{l_{total}}$
Jawahar et al. [39]	0.15287	0.12439	0.27726	0.50231
GAMM [38]	—	—	0.243-0.2868	0.4145-0.517
Scheme I	0.15353	0.13012	0.28365	0.43593
Scheme II	0.16031	0.13439	0.29470	0.46350
Scheme III	0.15387	0.13027	0.28414	0.43751

Case 4: Hypersonic flow around a half of circular cylinder

The above three test cases demonstrate that Scheme I and Scheme III have a good property for capturing viscous heat dissipation and boundary layer. To investigate the capability of three schemes for simulation of hypersonic flows with strong shock waves and boundary layer, the hypersonic flow around a half of circular cylinder is simulated. For this case, the free-stream Mach number is 8.03, the Reynolds number is 1.835×10^5, the free-stream temperature is 124.94K and the wall temperature is fixed as 299.44K. In this test problem, structured grid with 160×160 cells is used, and the cell Reynolds number is taken as $Re_{cell}=\rho_\infty u_\infty \Delta r/\mu_\infty=1.835$, where Δr is the mesh spacing of the first cell in the normal direction adjacent to the cylinder surface, ρ_∞, u_∞ and μ_∞ are the density, velocity and dynamic viscosity of the free stream respectively. In addition, the van Leer limiter [40] and LU-SGS scheme [41] are applied. The convergence criterion is set by the condition that the maximum residual of governing equation is decreased to less than 10^{-7}.

Numerical experiments revealed that the computation will diverge when Scheme I is applied for this test case. The main reason may be that Scheme I lacks essential numerical dissipation to suppress the instability of strong shock waves. Fig. 14 shows the temperature contours obtained from Scheme II (left), Scheme III (middle) and Scheme III* (right). It can be seen that the result of Scheme III exhibits oscillation in the post-shock region. The reason may be attributed to the switch function defined in Eq. (2.29), which cannot well switch Scheme III into Scheme II around the strong shock wave. This can be well demonstrated by Fig. 15, which displays the switch function contours used in Scheme III (left) and Scheme III* (right). The values of the switch function of Scheme III are obviously less than those of Scheme III* in the vicinity of shock wave. Fig. 16 presents the pressure (upper) and heat flux (bottom) distributions along the cylindrical surface com-

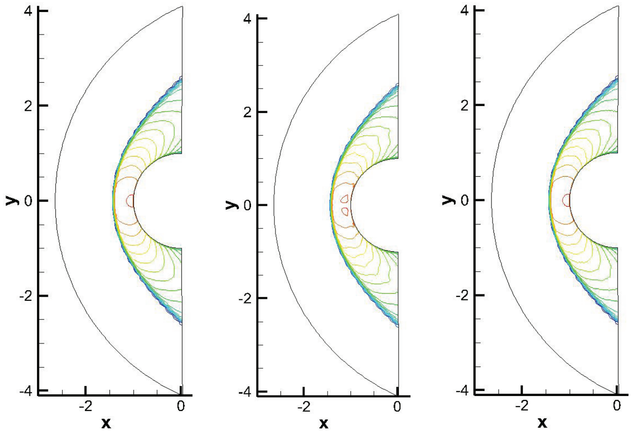

Figure 14: Temperature contours of hypersonic flow around a half of circular cylinder obtained by Scheme II (left), Scheme III (middle) and Scheme III* (right).

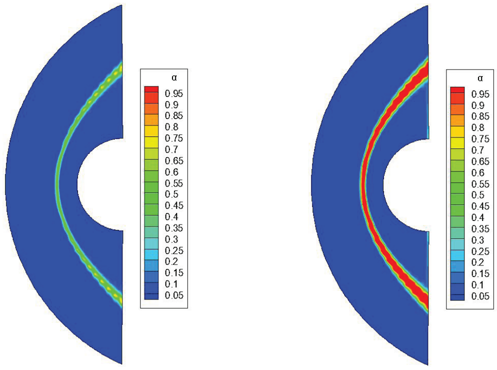

Figure 15: Switch function contours for hypersonic flow around a half of circular cylinder used in Scheme III (left) and Scheme III* (right).

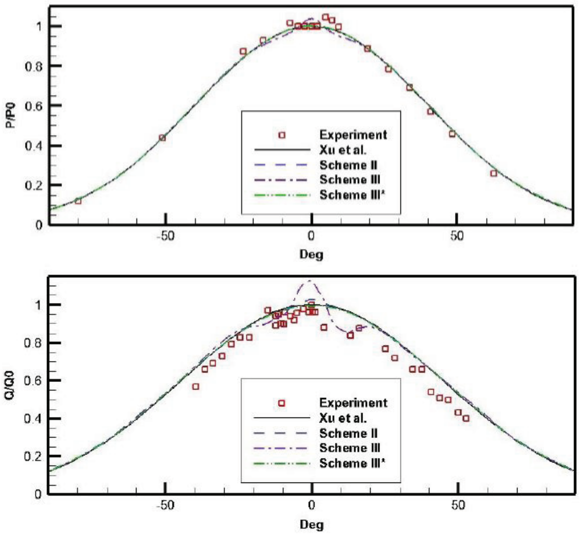

Figure 16: Comparison of pressure (upper) and heat flux (bottom) distributions along the cylindrical surface for hypersonic flow around a half of circular cylinder.

puted by Scheme II, Scheme III and Scheme III*, where p_0 and q_0 are the exact solutions, $p_0 = 0.9209$ and $q_0 = 0.003655$. Also displayed in this figure are the experimental data [42] and results calculated by gas-kinetic BGK scheme [43]. It can be observed that the pressure and heat flux distributions of Scheme III jitter apparently in the vicinity of stagnation point, and the heat flux distribution of Scheme II deviates significantly from the reference data [43] near the stagnation point. The comparison of computed pressure and heat flux at stagnation point by three schemes with exact solution is made in Table 2. As shown in this table, the results of Scheme III* agree very well with exact solution. This example shows that Scheme III* can well capture strong shock waves and thin boundary layer in numerical simulation of hypersonic viscous flows.

Table 2: Comparison of stagnation pressure and heat flux for hypersonic flow around a half of cylinder.

References	$p/\rho_\infty u_\infty^2$	$q/\rho_\infty u_\infty^3$
Exact solutions	0.9209	0.003655
Scheme I	–	–
Scheme II	0.9223	0.003717
Scheme III	0.9581	0.004108
Scheme III*	0.9214	0.003650

4 Conclusions

In this paper, the lattice Boltzmann flux solver (LBFS), which is developed in our previous work for simulation of inviscid compressible flows [28–31], is extended to simulate viscous compressible flows. In the solver, the inviscid flux across the cell interface is evaluated by LBFS, while the viscous flux is still approximated by conventional smooth function approximation. From Chapman-Enskog expansion analysis [19, 20, 32, 33], it is known that the equilibrium part of the distribution function contributes to the inviscid flux while its non-equilibrium part contributes to the viscous flux. As LBFS is only used to evaluate inviscid flux in this work, the non-equilibrium part in LBFS can be viewed as numerical dissipation, and the dimensionless collision time τ_0 can be regarded as the weight of the numerical dissipation. By taking τ_0 as 0, 1 and a variable switch function, three schemes of LBFS are presented. For $\tau_0 = 0$ (Scheme I), the distribution function at the cell interface is completely computed from the equilibrium part, and the numerical dissipation in the scheme is very little. As a consequence, it can provide accurate results for boundary layer flows. But it often exhibits oscillation or even diverges for simulation of hypersonic flows due to the lack of numerical dissipation. For $\tau_0 = 1$ (Scheme II), the distribution functions at cell interface streamed from neighboring points are directly used to compute the inviscid flux. This scheme has relatively large numerical dissipation to get stable solution for strong shock waves. However, the large numerical dissipation pollutes the solution in smooth regions. To capture both the strong shock waves and thin boundary layers accurately, a switch function to control τ_0 is introduced and Scheme III and Scheme III* are proposed. In Scheme III and Scheme III*, τ_0 is taken as a value close to zero in the boundary layer and one around the strong shock wave.

To compare the performances of the above schemes, several numerical examples are tested. Numerical results showed that (1) Scheme I can well simulate viscous boundary layers with little numerical dissipation but it is usually incompetent for simulation of hypersonic flows; (2) Scheme II has relatively large numerical dissipation to get stable solution for strong shock waves, and the numerical dissipation can be significantly reduced through refinement of mesh size; (3) Scheme III* can capture well both strong shock waves and thin boundary layer in numerical simulation of hypersonic viscous flows. From the present study, we may conclude that for simulation of compressible flows with relatively low Mach number, Scheme I and Scheme III are better than Scheme

III* due to lower cost of their computational effort. For simulation of supersonic and hypersonic flows, Scheme III* could be the best choice.

Acknowledgments

This work has been supported by the State Key Laboratory of Aerodynamics of China (No. SKLA201401).

References

[1] X. L. Li, D. X. Fu and Y. W. Ma, *Optimized group velocity control scheme and DNS of decaying compressible turbulence of relative high turbulent Mach number,* Int. J. Numer. Meth. Fluids, 48 (2005), pp. 835–852.

[2] J. X. Qiu, B. C. Khoo and C. W. Shu, *A numerical study for the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes,* J. Comput. Phys., 212 (2006), pp. 540–565.

[3] H. W. Zheng, C. Shu and Y. T. Chew, *An object-oriented and quadrilateral-mesh based solution adaptive algorithm for compressible multi-fluid flows,* J. Comput. Phys., 227 (2008), pp. 6895–6921.

[4] L. M. Yang, C. Shu, J. Wu, N. Zhao and Z. L. Lu, *Circular function-based gas-kinetic scheme for simulation of inviscid compressible flows,* J. Comput. Phys., 255 (2013), pp. 540–557.

[5] S. Z. Chen, K. Xu, C. B. Lee and Q. D. Cai, *A unified gas kinetic scheme with moving mesh and velocity space adaptation,* J. Comput. Phys., 231 (2012), pp. 6643–6664.

[6] A. Main and C. Farhat, *A second-order time-accurate implicit finite volume method with exact two-phase Riemann problems for compressible multi-phase fluid and fluid-structure problems,* J. Comput. Phys., 258 (2014), pp. 613–633.

[7] P. W. McDonald, *The computation of transonic flow through tow-dimensional gas turbine cascades,* ASME Paper 71-GT-89, 1971.

[8] S. V. Patankar and D. B. Spalding, *A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows,* Int. J. Heat Mass Transfer, 15 (1972), pp. 1787–1806.

[9] A. J. Chorin, *A numerical method for solving incompressible viscous flow problems*, J. Comput. Phys., 2 (1967), pp. 12–26.

[10] P. L. Roe, *Approximate Riemann solvers, parameter vectors, and difference schemes,* J. Comput. Phys., 43 (1981), pp. 357–372.

[11] B. van Leer, *Flux vector splitting for the Euler equations,* Lecture Notes in Physics, 170 (1982), pp. 507–512.

[12] M. S. Liou and C. J. Steffen, *A new flux splitting scheme,* J. Comput. Phys., 107 (1993), pp. 23–39.

[13] K. Kitamura, E. Shima and P. L. Roe, *Evaluation of Euler fluxes for hypersonic heating computations,* AIAA J., 48 (2010), pp. 763–776.

[14] B. van Leer, J. L. Thomas, P. L. Roe and R. W. Newsome, *A comparison of numerical flux formulas for the Euler and Navier-Stokes equations*, AIAA Paper, 87-1104, 1987.

[15] S. Y. Chou and D. Baganoff, *Kinetic flux-vector splitting for the Navier-Stokes equations,* J. Comput. Phys., 130 (1997), pp. 217–230.

[16] D. Chae, C. Kim and O. H. Rho, *Development of an improved gas-kinetic BGK scheme for inviscid and viscous flows,* J. Comput. Phys., 158 (2000), pp. 1–27.

[17] K. Xu, *A gas-kinetic BGK scheme for the Navier-Stokes equations and its connection with artificial dissipation and Godunov method,* J. Comput. Phys., 171 (2001), pp. 289–335.

[18] K. Xu, *Gas-kinetic schemes for unsteady compressible flow simulations,* VKI for Fluid Dynamics Lecture Series, 1998-03 (1998).

[19] R. Benzi, S. Succi and M. Vergassola, *The lattice Boltzmann equation: theory and application,* Physics Report, 1992.

[20] Z. L. Guo and C. Shu, Lattice Boltzmann Method and Its Applications in Engineering, World Scientific Publishing, 2013.

[21] T. Kataoka and M. Tsutahara, *Lattice Boltzmann method for the compressible Euler equations,* Phys. Rev. E., 69 (2004), 056702.

[22] K. Qu, C. Shu and Y. T. Chew, *Alternative method to construct equilibrium distribution functions in lattice-Boltzmann method simulation of inviscid compressible flows at high Mach number,* Phys. Rev. E., 75 (2007), 036706.

[23] Q. Li, Y. L. He, Y. Wang and G. H. Tang, *Three-dimensional non-free-parameter lattice-Boltzmann model and its application to inviscid compressible flows,* Phys. Lett. A, 373 (2009), pp. 2101–2108.

[24] C. W. Zhong, K. Li, J. H. Sun, C. S. Zhou and J. F. Xie, *Compressible flow simulation around airfoil based on lattice Boltzmann method,* Transactions of Nanjing University of Aeronautics and Astronautics, 26 (2009), pp. 206–211.

[25] H. W. Xi, G. W. Peng and S. H. Chou, *Finite-volume lattice Boltzmann method,* Phys. Rev. E., 59 (1999), pp. 6202–6205.

[26] S. Ubertini, G. Bella and S. Succi, *Lattice Boltzmann method on unstructured grids: further developments,* Phys. Rev. E., 68 (2003), 016701.

[27] M. Stiebler, J. Tölke and M. Krafczyk, *An upwind discretization scheme for the finite volume lattice Boltzmann method,* Comput. Fluids, 35 (2006), pp. 814–819.

[28] C. Z. Ji, C. Shu and N. Zhao, *A lattice Boltzmann method-based flux solver and its application to solve shock tube problem,* Mod. Phys. Lett. B., 23 (2009), pp. 313–316.

[29] L. M. Yang, C. Shu and J. Wu, *Development and comparative studies of three non-free parameter lattice Boltzmann models for simulation of compressible flows,* Adv. Appl. Math. Mech., 4 (2012), pp. 454–472.

[30] L. M. Yang, C. Shu and J. Wu, *A moment conservation-based non-free parameter compressible lattice Boltzmann model and its application for flux evaluation at cell interface,* Comput. Fluids, 79 (2013), pp. 190–199.

[31] C. Shu, Y. Wang, L. M. Yang and J. Wu, *Lattice Boltzmann flux solver: an efficient approach for numerical simulation of fluid flows,* Transactions of Nanjing University of Aeronautics and Astronautics, 31 (2014), pp. 1–15.

[32] P. L. Bhatnagar, E. P. Gross and M. Krook, *A model for collision processes in gases. I: small amplitude processes in charged and neutral one-component systems,* Phys. Rev., 94 (1954), pp. 511–525.

[33] K. Xu and X. Y. He, *Lattice Boltzmann method and gas-kinetic BGK scheme in the low-Mach number viscous flow simulations,* J. Comput. Phys., 190 (2003), pp. 100–117.

[34] T. J. Barth and D. C. Jespersen, *The design and application of upwind schemes on unstructured meshes,* AIAA Paper, 89-0366, 1989.

[35] J. Blazek, Computation Fluid Dynamics: Principle and Application, Elsevier, 2001.

[36] R. C. Swanson and R. Radespiel, *Cell centered and cell vertex multigrid schemes for the*

Navier-Stokes equations, AIAA J., 29 (1991), pp. 697–703.

[37] V. Venkatakrishnan, *Convergence to steady-state solutions of the Euler equations on unstructured grids with limiters,* J. Comput. Phys., 118 (1995), pp. 120–130.

[38] M. O. Bristeau, R. Glowinski, J. Periaux and H. Viviand, *Numerical simulation of compressible Navier-Stokes flows*, Vieweg and Sonh Braunschweig, Wiesbaden, (1987).

[39] P. Jawahar and H. Kamath, *A high-resolution procedure for Euler and Navier-Stokes computations on unstructured grids,* J. Comput. Phys., 164 (2000), pp. 165–203.

[40] B. van Leer, *Toward the ultimate conservative difference scheme iv, a new approach to numerical convection,* J. Comput. Phys., 23 (1977), pp. 276–299.

[41] S. Yoon and A. Jameson, *Lower-upper Symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations,* AIAA J., 26 (1988), pp. 1025–1026.

[42] A. R. Wieting, *Experimental study of shock wave interface heating on a cylindrical leading edge,* NASA TM-100484, 1987.

[43] K. Xu, M. L. Mao and L. Tang, *A multidimensional gas-kinetic BGK scheme for hypersonic viscous flow,* J. Comput. Phys., 203 (2005), pp. 405–421.