A semi-Lagrangian implicit BGK collision model for the finite volume discrete Boltzmann method

Leitao Chen, Laura Schaefer

Department of Mechanical Engineering, Rice University, Houston, Texas 77005, USA

Abstract: In order to increase the temporal accuracy, reduce the computational cost and improve the stability due to collisions for the finite volume discrete Boltzmann method (FVDBM), a new implicit BGK collision model using a semi-Lagrangian approach is proposed in this paper. Unlike existing models, in which the implicit BGK collision is resolved either by a linear extrapolation in time or by a variable transformation, the new model removes the implicitness by tracing the particle distribution functions (PDFs) back in time along their characteristic paths during the collision process. An interpolation scheme is needed to evaluate the PDFs at the traced-back locations. In this paper, the first-order interpolation is used, and the resulting model allows for the straightforward replacement of \( f_{\alpha}^{eq,n+1} \) by \( f_{\alpha}^{eq,n} \) no matter where it appears. After comparing the new model with the existing models under different numerical conditions (e.g. flux scheme and time marching scheme) and using the new model to successfully modify the variable transformation technique, three conclusions can be made. First, the new model can dramatically improve the accuracy; second, it can reduce the computational cost; and third, the new model can significantly improve or preserve the \( \Delta t/\tau \) limits of the existing models.

Keywords: lattice Boltzmann method; discrete Boltzmann method; finite volume method; unstructured mesh; BGK collision; semi-Lagrangian method; accuracy; computational cost; stability

1. Background

After over three decades of development, the lattice Boltzmann method (LBM) has become a successful alternative to the conventional computational fluid dynamics (CFD) models that are built on the Navier-Stokes equation (NSE). Its success is supported by two important features that are missing from conventional CFD models. First, physically, the LBM can inherently solve problems over a wider range of length scales than the NSE [1]. This is why the LBM is considered to be a multiscale method. Second, numerically, the LBM very efficiently achieves second-order accuracy both in time and space with only a first-order numerical scheme [2]. The reason for this is that the advection term \( e_\alpha \cdot \nabla f_{\alpha} \) in the LBM is linear (\( e_\alpha \) before the gradient is constant); in the NSE, the advection term \( u \cdot \nabla u \) is nonlinear (\( u \) before the gradient is not constant). Due to the linear advection, the LBM can couple the discretizations of all three dimensions: the microscopic velocity \( e \), space \( x \), and time \( t \). By doing this, the variables that are being advected (in this case, the particle distribution functions) will stop exactly at a grid point after each advection step. According to the definition of the Courant-Friedrichs-Lewy (CFL) number, the CFL of the microscopic velocities in the LBM becomes one globally, regardless of location and time, which creates a universal second-order accuracy.

Although this unique multi-dimensional coupling mechanism gives the LBM an unparalleled edge, it also brings with it a substantial challenge. Since the LBM couples the discretizations of all three dimensions,
this limits the freedom to choose a different way of individually discretizing any of the three dimensions, which is especially restrictive for the spatial dimension (x). Therefore, the mesh, which is the result of discretizing the space, has to copy the lattice structure (a lattice tells how the velocity is discretized), and has to be uniform (in order to achieve \(CFL=1\) location-wise) and rigid (in order to achieve \(CFL=1\) time-wise). Consequently, such a uniform and rigid mesh structure makes it difficult for the LBM to accurately accommodate problems with curved or complicated boundaries [3], which are, however, ubiquitous in fluid flow problems.

Numerically speaking, the LBM (with its coupling feature mentioned above) is derived from the discrete Boltzmann equation (DBE) whose space and time are still continuous. Therefore, by solving the DBE, one can select an arbitrary discretization for the space. As a result, complex boundaries can be easily captured with a body-fitting mesh, just like in the conventional CFD models. The original work following this idea was a series of paper by Peng et al. [4-6], in which the DBE is solved with the finite volume method (FVM) on an unstructured triangular mesh. Their work belongs to a category called the finite volume discrete Boltzmann method (FVDBM), which has witnessed a rapid progression in the following years [7-14]. Another major method that could also use this approach is the finite element discrete Boltzmann method (FEDBM) [15-17], because the finite element method (FEM) can easily integrate unstructured meshes as well. However, this has not gained the same popularity as the FVDBM due to mathematical simplicity and the built-in conservation of the FVM.

Unfortunately, as a result of the mesh flexibility, the FVDBM (as well as other methods based on solving the DBE) currently exhibits a lower accuracy and higher computational cost than the LBM. Since the space and time in the FVDBM is decoupled, the accuracy and computational cost in space as well as in time must be handled separately.

In the spatial dimension, the accuracy of the FVDBM is mostly limited by the diffusion error. Per a Fourier stability analysis, this diffusion error automatically appears when \(CFL<1\), which is required to maintain proper stability when solving the DBE on irregular meshes [18]. Such a diffusion error in the FVDBM has been well acknowledged even in the early stages of its development [19, 20]. However, very few publications have provided solutions on how to reduce this diffusion error with a reasonable cost. As a result, it was conveniently concluded that the FVDBM was not a competitive alternative to the LBM [10]. As an effort to address this issue, one of our previous papers provided a systematic approach that could produce Godunov-type flux schemes with different orders of accuracy for the advection in the FVDBM, which could significantly reduce the diffusion error beyond that of the conventional upwinding schemes [21]. We also developed a new second-order interpolation scheme designated the plane-fitting least-square (PFLS) approach to reduce the diffusion error during the interpolation step of the FVDBM [22], which displayed a faster speed as well as a slightly better accuracy than the conventional least-square interpolation scheme.

In the time dimension, the time marching scheme should be carefully chosen because, when solving the DBE, the maximum \(\Delta t\) is not only limited by the CFL, which is controlled by the advection, but also limited by the relaxation time, which is affected by the collision. The explanation is that \(\Delta t\), which is the numerical time interval for updating the solution, cannot be too large compared to the physical time the
system takes during each time step to relax towards the equilibrium state. As a result, solving the DBE requires a very small $\Delta t$ when modeling steady-state high-Re flows in which the relaxation time is very small. Therefore, the selected time marching scheme should allow the use of a $\Delta t$ that is as large as possible, as long it is within the physical limit. The standard approach to achieve this is to make the collision implicit for the time marching [15, 16, 24], which, however, creates an implicit (nonlinear) equilibrium term that requires additional treatment. Currently, there are basically two approaches to resolve this implicitness: the temporal extrapolation (TE) scheme that calculates the implicit value based on two previous time steps [24], and the variable transformation (VT) technique that can wrap the implicit term into a new variable [25-33]. It was reported that the VT can dramatically improve the stability beyond the TE scheme [29, 33]. However, the comparison in accuracy and cost between these two is unknown.

In this paper, we develop a new scheme to resolve the implicit collision during time marching. The new scheme is based on applying the semi-Lagrangian (SL) treatment to the implicit collision term. After a holistic quantitative comparison in accuracy, cost, as well as the stability, between the new scheme and the two mentioned above on the Bhatnagar-Gross-Krook (BGK) collision model [23], it is found that the new scheme is not only more accurate (in time) and faster than the existing schemes, but also more stable than the TE scheme while having the same stability as the VT scheme.

2. The FVDBM with an implicit BGK collision

The DBE with the BGK collision model (on the right-hand side) is defined as:

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

where $f_{\alpha}$ and $f_{\alpha}^{eq}$ are the particle distribution function (PDF) and equilibrium PDF, respectively, in the $\alpha$th direction of a total of $N$ components, $\mathbf{e}_{\alpha}$ is the $\alpha$th of $N$ total lattice velocities, and $\tau$ is the relaxation time. With the help the FVM, Eq. (1) can be integrated over a control volume (CV). Then, after a rearrangement, the FVDBM in its general form is shown as:

$$T_{\alpha} = C_{\alpha} - F_{\alpha} \tag{2}$$

where $T_{\alpha}$, $C_{\alpha}$ and $F_{\alpha}$ are the temporal term, collision term and flux term, respectively. The temporal and collision terms are:

$$T_{\alpha} = \frac{\partial f_{\alpha}}{\partial t} \tag{3}$$

$$C_{\alpha} = \frac{1}{\tau} \left( f_{\alpha}^{eq} - f_{\alpha} \right) \tag{4}$$

It is worth noting that, so far, Eq. (2) is still continuous both in space and time. When discretizing the space with a mesh such as an unstructured one, the total flux through the surface closure of each CV becomes the summation of the flux through each of $K$ total surface segments of the CV. Then, the flux term in Eq. (2) becomes:
\[ F_{\alpha} = \frac{1}{V_{CV}} \sum_{i=1}^{K} F_{\alpha,i} \]  

(5)

where \( V_{CV} \) is the volume of the CV. In the current study, cell-centered triangular meshes are used, which makes \( K = 3 \). After discretizing the time with a proper time marching scheme, Eq. (2) can be solved numerically. With the standard forward Euler method, Eq. (2) becomes:

\[ T_{\alpha}^n = C_{\alpha}^n - F_{\alpha}^n \]  

(6)

where

\[ T_{\alpha}^n = \frac{f_{\alpha}^{n+1} - f_{\alpha}^n}{\Delta t} \]  

(7)

and \( C_{\alpha}^n \) and \( F_{\alpha}^n \) are the collision and flux terms that are evaluated at time step \( t_n \). By plugging Eq. (7) into Eq. (6), replacing \( C_{\alpha}^n \) with its definition (Eq. (4)), and combining the terms that contain \( f_{\alpha}^n \), the simplest form for the FVDBM is:

\[ f_{\alpha}^{n+1} = \left( 1 - \frac{\Delta t}{\tau} \right) f_{\alpha}^n + \frac{\Delta t}{\tau} f_{\alpha}^{eq,n} - \Delta t F_{\alpha}^n \]  

(8)

However, it is well known the forward Euler method is explicit and is not as stable as implicit methods. Lee et al. [15, 16] and Bardow et al. [25] separately tried to introduce implicitity into the system. They developed a general formula that keeps the collision implicit and the advection explicit. After applying this method to the FVDBM, it becomes:

\[ T_{\alpha}^n = [(1 - \theta)C_{\alpha}^n + \theta C_{\alpha}^{n+1}] - F_{\alpha}^n \]  

(9)

where \( \theta \) is a tuning parameter that varies between 0 and 1. The collision term in Eq. (9) becomes fully explicit if \( \theta = 0 \), and fully implicit once \( \theta = 1 \). Here we start from a simple case in which \( \theta = 1 \). Then Eq. (9) becomes:

\[ T_{\alpha}^n = C_{\alpha}^{n+1} - F_{\alpha}^n \]  

(10)

By plugging Eqs. (4) and (7) into Eq. (10) to replace the collision and temporal terms and rearranging the equation, Eq. (10) becomes:

\[ f_{\alpha}^{n+1} = \frac{\tau}{\tau + \Delta t} f_{\alpha}^n + \frac{\Delta t}{\tau + \Delta t} f_{\alpha}^{eq,n+1} - \frac{\tau \Delta t}{\tau + \Delta t} F_{\alpha}^n \]  

(11)

Eq. (11) is more stable than Eq. (8). However, there is still implicitness left untouched in \( f_{\alpha}^{eq,n+1} \) that needs to be resolved, which will be discussed in the next section. It is important to note that a proper flux scheme is required to calculate the flux term \( F_{\alpha}^n \) in order to close the system, which will be discussed later.

3. The state of art of resolving the implicit BGK collision
The most simple and straightforward approach to resolve implicitness for any problem is to solve the implicit variables with an iterative process. For the current application, the procedure should be performed with the following steps:

Step 1: Guess an initial value for $f_{\alpha}^{eq,n+1}$;
Step 2: Calculate $f_{\alpha}^{n+1}$ with Eq. (11);
Step 3: Calculate the moments with $f_{\alpha}^{n+1}$ from step 2;
Step 4: Calculate the new $f_{\alpha}^{eq,n+1}$ with the moments from step 3;
Step 5: Check the difference between the new $f_{\alpha}^{eq,n+1}$ and its value in the last iteration. If it is converged, finish; otherwise, repeat steps 2 to 5.

This iterative process is very costly since the calculation of moments (step 3) and the calculation of the equilibrium PDF (step 4) are computationally intense and the convergence criteria must be met at all grid locations. Therefore, this method is not studied in this paper. Instead, all methods discussed in this paper are non-iterative.

3.1 The temporal extrapolation (TE) scheme

The first approach is to directly solve $f_{\alpha}^{eq,n+1}$ and put it back in Eq. (11) to close the system. According to Mei and Shyy [24], $f_{\alpha}^{eq,n+1}$ can be linearly extrapolated, as a whole, by using its own values in the two previous time steps, namely:

$$f_{\alpha}^{eq,n+1} = 2f_{\alpha}^{eq,n} - f_{\alpha}^{eq,n-1}$$ (12)

which is termed the temporal extrapolation (TE) scheme in this paper. For the standard two-dimensional nine-velocity model (D2Q9), the equilibrium PDF at any time step is computed as:

$$f_{\alpha}^{eq} = \omega_{\alpha} \rho \left[ 1 + \frac{e_{\alpha}u}{c_s^2} + \frac{(e_{\alpha}u)^2}{2c_s^4} - \frac{u \cdot u}{2c_s^2} \right]$$ (13)

where $\omega_{\alpha}$ is the weight in each corresponding direction, $c_s$ is the speed of sound, and $u$ and $\rho$ are the macroscopic velocity and density, or moments, which can be calculated as:

$$\begin{bmatrix} \rho \\ \mathbf{u} \end{bmatrix} = \sum_{\alpha=0}^{N} \begin{bmatrix} f_{\alpha} \\ e_{\alpha}f_{\alpha} \end{bmatrix}$$ (14)

As the result, the computation procedure of the FVDBM using the TE scheme during each time step is as follows:

Step 1: Calculate the moments with Eq. (14) with newest $f_{\alpha}$;
Step 2: Calculate $f_{\alpha}^{eq}$ with Eq. (13);
Step 3: Calculate $f_{\alpha}^{eq,n+1}$ with Eq. (12);
Step 4: Update $f_{\alpha}$ with Eq. (11).
It should be mentioned that the step 2 requires only one computation of \( f_{\alpha}^{eq} \) but additional memory allocation to store its value at \( t_{n-1} \).

### 3.2 The variable transformation (VT) scheme

The TE scheme is very easy to implement. However, it was noted by Mei and Shyy [24] that the TE scheme is prone to instability due to the extrapolation. In order to address this, Bardow et al. [25] introduced a technique called variable transformation (VT) to avoid the need for a temporal extrapolation. It has been shown that the VT scheme is much more stable than the TE [29, 33], and therefore it has become a widely accepted approach [26–33]. The VT scheme does not focus on solving \( f_{\alpha}^{eq,n+1} \) by itself. Instead, it treats the entire collision term as a whole. In the context of the FVDBM, the VT scheme starts from the following governing equation, which is the result of replacing \( T_{\alpha}^n \) in Eq. (10) with Eq. (7):

\[
f_{\alpha}^{n+1} = f_{\alpha}^n + \Delta t C_{\alpha}^{n+1} - \Delta t F_{\alpha}^n
\]  

(15)

By defining a new variable \( g_{\alpha} \) as:

\[
g_{\alpha} = f_{\alpha} - \Delta t C_{\alpha}
\]  

(16)

therefore, at the time step \( t_{n+1} \), it holds that:

\[
g_{\alpha}^{n+1} = f_{\alpha}^{n+1} - \Delta t C_{\alpha}^{n+1}
\]  

(17)

With the help of Eq. (17), Eq. (15) becomes:

\[
g_{\alpha}^{n+1} = f_{\alpha}^n - \Delta t F_{\alpha}^n
\]  

(18)

It can be seen that there is no implicitness on the right-hand-side of Eq. (18), so \( g_{\alpha}^{n+1} \) can be computed after the flux calculation is finished. The next task is to recover \( f_{\alpha}^{n+1} \) from \( g_{\alpha}^{n+1} \). By rearranging Eq. (17) after expanding the BGK collision term and combining the terms that contain \( f_{\alpha}^{n+1} \), it can be obtained that:

\[
f_{\alpha}^{n+1} = \frac{\tau}{\tau + \Delta t} \left( g_{\alpha}^{n+1} + \frac{\Delta t}{\tau} f_{\alpha}^{eq,n+1} \right)
\]  

(19)

The new variable \( g_{\alpha} \) satisfies the condition that it preserve the moments of \( f_{\alpha} \), therefore:

\[
\begin{bmatrix} \rho \\ \mathbf{u} \end{bmatrix} = \sum_{\alpha=0}^{N} \begin{bmatrix} g_{\alpha} \\ e_{\alpha} g_{\alpha} \end{bmatrix} = \sum_{\alpha=0}^{N} \begin{bmatrix} f_{\alpha} \\ e_{\alpha} f_{\alpha} \end{bmatrix}
\]  

(20)

Since the equilibrium PDF can be exclusively determined by its moments, as shown in Eq. (13), it also holds that:

\[
f_{\alpha}^{eq,n+1} = g_{\alpha}^{eq,n+1}
\]  

(21)

As a result, the procedure of using the VT scheme for the FVDBM during each time is:
Step 1: Calculate $g_{\alpha}^{n+1}$ with Eq. (18);
Step 2: Calculate the moments based on $g_{\alpha}^{n+1}$ with Eq. (20);
Step 3: Calculate $g_{\alpha}^{eq,n+1}$ with Eq. (13) based on the moments from the step 2;
Step 4: Update $f_{\alpha}$ with Eq. (19) by applying Eq. (21).

4. The semi-Lagrangian (SL) BGK collision scheme

The semi-Lagrangian (SL) method originated in the applied math community for solving the general transport equation [34-37]. It preserves the mesh flexibility of the Eulerian method while maintaining a good level of accuracy and large CFL numbers of the Lagrangian method. The SL method has not been introduced to the LBM community until recently when Krämer et al. [38] applied the SL method to their work on the off-lattice Boltzmann method (OLBM). They used the SL method to solve the streaming step and found that the SL method can increase the computational efficiency by allowing a larger time step size. However, the SL method has never been applied to the collision of any LBM or DBM work. In this section, a new approach that resolves the implicitness by applying the SL method to the implicit BGK collision term is developed. This approach is completely different from the TE and VT methods discussed in the previous section. The development of the SL scheme will be explained in detail in the rest of this section by starting from the examination of $f^{eq}$, which is defined as the Maxwellian distribution that is a function of moments (density $\rho$, macroscopic velocity $u$, etc.) such that:

$$f^{eq} = \frac{\rho}{(2\pi RT)^{\frac{D}{2}}} e^{\frac{-\left(e-u\right)^2}{2RT}}$$ (22)

where $T$ is the temperature, $D$ is the number of dimensions, and $R$ is the ideal gas constant. The microscopic velocity $e$ in Eq. (22) is still continuous, which needs to be discretized in order to be solved computationally. Once discretized, $f^{eq}$ can be computed by performing a Taylor expansion on the Maxwellian. Eq. (13) is the discrete form of $f^{eq}$ for the D2Q9 lattice with a second-order truncation. After the discretization, the moments can be recovered by taking an ensemble of the PDFs, as shown in Eq. (14).

From Eqs. (13) and (14) it can be seen that $f_{\alpha}^{eq}$ is a function of moments that is further a function of $f_{\alpha}$, which can be depicted by the following notation:

$$f_{\alpha}^{eq} = M\{m[f_{\alpha}]\}$$ (23)

where $m[ ]$ is the operator that calculates the moments from the PDFs, and $M\{\}$ is the operator that calculates the equilibrium PDFs from the moments. (Technically speaking, $M\{\}$ is the Maxwellian operator that calculates the equilibrium PDFs by Eq. (22). However, its notation is borrowed here to specifically represent the calculation of equilibrium PDFs with discrete velocities.) As shown in Eq. (23), $f_{\alpha}^{eq}$ is an indirect function of $f_{\alpha}$, which is at the same time the solution of the DBE (Eq. (1)). This is the reason why the implicitness in $f_{\alpha}^{eq,n+1}$ is difficult to treat.
Unlike TE and VT schemes, the SL method tracks the PDFs along their characteristics back in time in a Lagrangian way and was initially proposed by Groppi et al. [39]. The SL method consists of two steps, which will be explained as follows. First, according to Eq. (23), at the barycenter $P$ of any CV, as shown in Fig. 1, and at $t_{n+1}$, it holds that:

$$f_{\alpha}^{eq,n+1}(P) = M\{m[f_{\alpha}^{n+1}(P)]\}$$  (24)

Second, as pointed out by Groppi et al. [39], the PDFs preserve their values along their characteristic lines when being advected, followed by their moments. Therefore, the PDFs at $t_{n+1}$ are the same as the PDFs at $t_n$ at their previous locations rendered by being tracked back in time along the characteristic path. Taking the D2Q9 lattice model as example, whose structure and numbering of different directions are shown in the inset of Fig. 1, the nine PDFs that rendezvous at the location $P$ at $t_{n+1}$ were advected from different locations ($P_0$ to $P_8$) at the previous time step $t_n$. Therefore, it holds that:

$$f_{\alpha}^{n+1}(P) = f_{\alpha}^{n}(P_{\alpha})$$  (25)

Since the PDFs keep their values along their characteristic paths, the moments that are calculated based on these PDFs also stay the same. Therefore, there exists:

$$m[f_{\alpha}^{n+1}(P)] = m[f_{\alpha}^{n}(P_{\alpha})]$$  (26)

For the D2Q9 lattice, Eq. (26) means that:

$$\begin{bmatrix} \rho^{n+1}(P) \\ u^{n+1}(P) \end{bmatrix} = \sum_{\alpha=0}^{8} \begin{bmatrix} f_{\alpha}^{n}(P_{\alpha}) \\ e_{\alpha} f_{\alpha}^{n}(P_{\alpha}) \end{bmatrix}$$  (27)
The tracked-back locations, \( P_0 \) to \( P_8 \) (\( P_0 \) is the same location as \( P \) since \( e_0 \) is 0), are generally not located at grid points (in the cell-centered meshes, they are not located at barycenters). Therefore, an interpolation scheme is needed to evaluate \( f^n_\alpha(P_\alpha) \). If using \( X \) to denote the coordinate of a point, then the coordinates of \( P_\alpha \) are known as:

\[
X(P_\alpha) = X(P) - e_\alpha \Delta t
\] (28)

This location information is deterministic and can be used to calculate \( f^n_\alpha(P_\alpha) \) in a chosen interpolation scheme with a desired order of accuracy. Once \( f^n_\alpha(P_\alpha) \) becomes known, the implicitness can be closed as:

\[
f^{eq,n+1}_\alpha(P) = M[m[f^n_\alpha(P_\alpha)]]
\] (29)

If a first-order interpolation scheme is selected, which means the PDF distributions are constant in each CV, it can be assumed that:

\[
f^n_\alpha(P_\alpha) = f^n_\alpha(P)
\] (30)

which says the PDF at any location within the same CV is equal to its value at the barycenter of the CV. Finally, Eq. (29) can be reduced to:

\[
f^{eq,n+1}_\alpha(P) = M[m[f^n_\alpha(P)]]
\] (31)

By revisiting Eq. (23) for the definition of \( f^{eq}_\alpha \), it can be seen that the right hand side of Eq. (31) is actually \( f^{eq,n}_\alpha(P) \). Therefore, Eq. (31) can be further reduced to:

\[
f^{eq,n+1}_\alpha(P) = f^{eq,n}_\alpha(P) \quad \text{or} \quad f^{eq,n+1}_\alpha = f^{eq,n}_\alpha
\] (32)

Eq. (32) is the final form of the SL scheme for the implicit BGK collision and will be the one investigated in the current paper. However, it is important to note that Eq. (32) is only a special case when applying the first-order interpolation to the more general form Eq. (29). A higher-order interpolation scheme will yield a completely different formula and will be saved for future work. Finally, the numerical sequence of updating the FVDBM solution with the SL scheme during each time is:

Step 1: Calculate the moments with Eq. (14) with the newest \( f_\alpha \);  
Step 2: Calculate \( f^{eq,n}_\alpha \) with Eq. (13) based on the moments from step 1;  
Step 3: Update \( f_\alpha \) with Eq. (11) by using Eq. (32) for \( f^{eq,n+1}_\alpha \).

5. Simulation results and discussions: a preliminary study

Taylor-Green vortex (TGV) flow is chosen as the example case for this study. The analytical velocities \((u_x, u_y)\) at any location \((x, y)\) and any time \( t \) are defined as:

\[
u_x = -u_0 \cos(k_1 x) \sin(k_2 y) e^{[-\nu(k_1^2+k_2^2)t]}\] (33)
\[ u_y = -u_0 \frac{k_1}{k_2} \sin(k_1 x) \cos(k_2 y) e^{[-\nu(k_1^2+k_2^2)t]} \]  

(34)

where \( u_0 \) is a reference velocity, \( \nu \) is the kinematic viscosity, and \( k_1 \) and \( k_2 \) are defined as:

\[ k_1 = \frac{2\pi}{D_x}, \quad k_2 = \frac{2\pi}{D_y} \]  

(35)

where \( D_x \) and \( D_y \) are the length and the height of the flow domain. The SL scheme is compared with the TE and VT schemes by solving the FVDBM with \( \theta = 1 \) for the TGV flow. The same second-order upwind (SOU) is used to calculate the flux term \( F_\alpha^N \) in Eq. (11). The comparisons in terms of accuracy, computational cost, and stability are made for all three implicit schemes, which will be discussed in detail in the following subsections.

5.1 Accuracy

TGV flow is characterized by a group of vortices that decay with time. The critical time \( t_c \) is the instant when the flow has decayed to exactly 50% of its initial strength. The \( L_2 \) errors (with respect to the analytical solution in Eqs. (33) and (34)) of the FVDBM solutions with three implicit collision schemes are calculated. In order to see the effect of \( \Delta t \) on the transient solutions, the transient errors in the window from 0 to \( t_c \) with four different sizes of \( \Delta t \) are measured and compared in Fig. 2. It can be seen that at all sizes of \( \Delta t \) the errors for all schemes grow with time because the errors will accumulate during each time step. However, the error from the SL scheme grows at a much slower pace than the TE and VT schemes as time progresses, which clearly indicates that the SL scheme can generate much less error than the TE and VT schemes. In addition, it can be seen that the TE and VT schemes generate very similar results and their difference is too small to be seen in Fig. 2. As a result, the error difference between the TE and VT schemes (TE minus VT) of the transient solutions is plotted against different sizes of \( \Delta t \) in Fig. 3, from which it can be seen that the VT scheme always generates slightly less error than the TE scheme, and that such a difference also grow with time and becomes larger with the increase of \( \Delta t \).
Figure 2. The $L_2$ errors of transient FVDBM solutions with different implicit collision schemes during the time span from 0 to $t_c$: (a) $\Delta t = 0.2\tau$; (b) $\Delta t = 0.4\tau$; (c) $\Delta t = 0.6\tau$; (d) $\Delta t = 0.8\tau$

Figure 3. The $L_2$ error difference between the TE and VT schemes (TE-VT) of transient FVDBM solutions during the time span from 0 to $t_c$

By examining each plot in Fig. 2 again, it is not difficult to conclude that the change of $\Delta t$ has different effects on the FVDBM solvers with different implicit collision schemes. For the TE and VT schemes, the change of $\Delta t$ does not strongly affect the error. On the other hand, an increasing $\Delta t$ will decrease the error of the solver with the SL scheme. These can be seen more clearly in Fig. 4, in which the errors with different schemes at the time instance $t = t_c$ are plotted against $\Delta t$. The error of a transient solution at any instance $t$ is the error accumulation from the initial time step to the current time step. In other words:

$$E_t = \epsilon \cdot S$$  \hspace{1cm} (36)

where $E_t$ is the error of the transient solution at the time $t$, $\epsilon$ is the error generated during one time step, and $S$ is the total number of time steps. For the TE and VT schemes, although $\epsilon$ will increase when increasing $\Delta t$, it takes fewer time steps to reach the same time $t$. As a result, the total accumulated error stays the same. Therefore, the FVDBM with the TE and VT schemes belongs to the family that satisfies the condition that:

$$\frac{\epsilon_1}{\epsilon_2} = \frac{\Delta t_1}{\Delta t_2}$$  \hspace{1cm} (37)
where $\Delta t_1$ and $\Delta t_2$ are two different time step sizes where $\Delta t_2 > \Delta t_1$, and $\varepsilon_1$ and $\varepsilon_2$ are the corresponding errors generated during one time step.

On the contrary, the error of the solution with the SL scheme at $t = t_c$ dramatically decreases with an increase in $\Delta t$. At $\Delta t = 0.2\tau$, the SL generates a 40% error compared to the analytical solution, which is a 25% improvement compared to the TE or VT scheme. When increasing the time step size to $\Delta t = 0.8\tau$, the SL scheme produces only a 10% error, which is an 80% improvement over the TE and VT schemes. The SL scheme behaves like this because it belongs to another family of schemes that satisfies:

$$\frac{\varepsilon_1}{\varepsilon_2} > \frac{\Delta t_1}{\Delta t_2}$$

(38)

This is an important feature since a larger $\Delta t$ will not only improve the accuracy of the solution but also bring down the computational cost by taking fewer time steps if a steady-state solution is sought. This feature of the SL scheme was also observed in the publication of Qiu & Shu [36] although the SL was applied to the advection term, not the collision.

![Figure 4](image)

*Figure 4. The effects of $\Delta t$ on the errors of the FVDBM solutions with different implicit collision schemes*

### 5.2 Computational cost

To be clear, the computational cost in this paper indicates the CPU cost only, which is the cost exclusively expended on “computing,” and the memory cost is not considered. The measured computational cost is the update time, $t_U$, in this paper, which is defined as:

$$t_U = \frac{t_T}{MS}$$

(39)

where $t_T$ is the total runtime for the simulation, $M$ is the total number of control volumes and $S$ is the total number of time steps or iterations. Therefore, $t_U$ is the overhead on all computational tasks in the FVDBM solver, not just the time spent on the implicit collision scheme. However, $t_U$ is able to reflect the difference in computational costs among different implicit collision schemes because all other numerical ingredients in the solver are the same. Table 1 lists the $t_U$ for the entire solver with different implicit collision schemes. All simulations are performed on an Intel i7-7700 3.6GHz CPU. Multiple measurements are taken and then averaged.
From the measurements it can be seen that all three schemes have almost the same $t_U$. However, the SL scheme is the fastest technique while the VT is the slowest. The SL is slightly faster than the TE scheme because the SL scheme avoids the computational time for $f_{e,q,n+1}$ by Eq. (12) that is required in the TE scheme; the VT is slower than the TE scheme because the VT scheme requires two times for variable transformations, one of which is from $f$ to $g$ (Eq. (17)) and the other is from $g$ to $f$ (Eq. (19)).

Table 1. Update time for the FVDBM solver with different implicit collision schemes

| Implicit collision scheme | $t_U$ (Unit: second) |
|---------------------------|-----------------------|
| TE                        | $4.159 \times 10^{-5}$ |
| VT                        | $4.211 \times 10^{-5}$ |
| SL                        | $4.145 \times 10^{-5}$ |

5.3 Stability

As discussed in the Background section, the stability of solving the DBE is determined both by the advection and collision. For the advection, the maximum $\Delta t$ is limited by $\Delta x$, which is the characteristic grid size and defined in following equation for the triangular mesh used in this paper:

$$\Delta x = \sqrt{2V_{CV}}$$

For the collision, the maximum $\Delta t$ is capped by the relaxation time $\tau$. The stability region of each implicit collision scheme is shown in Fig. 5, in which ● and × represent stable and unstable points, respectively. First, by comparing Fig. 5(a) and 5(b), one can see that the VT scheme significantly improves the stability in the $\Delta t/\tau$ limit, which is changed from 2.6 to 100. A similar improvement was also reported in other work [29, 33]. Second, Fig. 5(a) and 5(c) show that the SL scheme can also improve the $\Delta t/\tau$ limit by the same amount as the VT scheme. However, it is worth noting that the stability tests for the VT and SL schemes are capped at $\Delta t/\tau = 100$ because that ratio is considered to be quite good in practice. Therefore, the VT and SL schemes may display different $\Delta t/\tau$ behavior beyond 100, but that is not studied in the current paper. Finally, last but not least, by comparing Fig. 5(a), 5(b) and 5(c), it is clear that all of the schemes share the same $\Delta t/\Delta x$ limit, which is 0.15. This is expected because changing of the implicit collision scheme should not numerically affect the advection process or alter the $\Delta t/\Delta x$ limit.
5.4 Preliminary Conclusions and Further Discussions

Some preliminary conclusions can be made in this section. The comparison in accuracy reveals that, first, the VT scheme is slightly more accurate than the TE scheme, while both satisfy Eq. (37); second, the SL scheme is much more accurate than the TE and VT schemes, and satisfies Eq. (38). In the tested range of $\Delta t$, the SL scheme can improve the accuracy by more than 50% on average. The tests on computational costs shows that the SL is the fastest scheme because it is mathematically simpler than the other two and therefore requires fewer numerical operations. The stability test shows that the SL scheme has the same $\Delta t/\tau$ limit as the VT scheme in the tested range of $\Delta t/\tau$, which is an order of magnitude higher than the TE scheme.

By comparing the mathematical forms of the TE scheme (Eq. (12)) and the SL scheme (Eq. (32)) side by side, one could draw the erroneous conclusion that these two schemes belong to the same mathematical family that is based on the temporal extrapolation of equilibrium PDFs, because Eq. (12) is a second-order extrapolation that utilizes the equilibrium PDFs at the previous two time steps $t_n$ and $t_{n-1}$, and Eq. (32) appears to be a first order extrapolation that uses the equilibrium PDFs at only one time step $t_n$. However, Eq. (32) actually is not connected to a temporal extrapolation at all. There are two supporting pieces of evidence for this. First, if the SL and TE schemes belong to the same family but have different orders of accuracy, one would have been able to see that the SL generates a higher, not smaller, error than the TE scheme in Fig. 2. Second, the vigorous derivation in Section 4 reveals the true origin of Eq. (32), which is completely unrelated to temporal extrapolation. The reason why it appears to be a first-order temporal extrapolation scheme is only because Eq. (32) is a special case that applies the first-order interpolation to the general form of the semi-Lagrangian BGK model (Eq. (29)).

6. The application of the SL implicit BGK model: a secondary study

From the previous discussion, we can conclude that the rule of thumb for using the proposed SL implicit BGK model is very simple, which is:
It is important to recall that the numerical results and conclusions on the implicit collision schemes in the previous section are made in the specific numerical context that the time marching of the FVDBM is chosen to have $\theta = 1$ in Eq. (9) and the flux term $F_{n}^{\alpha}$ is completed with the second-order upwind scheme (SOU). In order to demonstrate that the numerical advantages of the SL scheme in terms of accuracy, computational cost and stability will generally hold, the same comparisons from the previous section will be performed again, but this time on the FVDBM with a different time marching scheme and a different flux calculation. Starting from Eq. (9) again for the general FVDBM, and choosing $\theta = \frac{1}{2}$, which is a popular choice for a second-order accuracy [15, 16, 33], Eq. (9) then becomes:

$$T_{\alpha}^{n} = \frac{1}{2}(C_{\alpha}^{n} + C_{\alpha}^{n+1}) - F_{\alpha}^{n}$$  \hspace{1cm} (41)

After using Eq. (4) for the BGK collision terms $C_{\alpha}^{n}$ and $C_{\alpha}^{n+1}$ and using Eq. (7) to replace $T_{\alpha}^{n}$, Eq. (41) is further transformed to:

$$f_{\alpha}^{n+1} = f_{\alpha}^{n} - \Delta t \left( \frac{f_{\alpha}^{n} - f_{\alpha}^{eq,n}}{2\tau} + \frac{(f_{\alpha}^{n+1} - f_{\alpha}^{eq,n+1})}{2\tau} \right) - \Delta t F_{\alpha}^{n}$$  \hspace{1cm} (42)

After combining the terms that contain $f_{\alpha}^{n}$ and $f_{\alpha}^{n+1}$, Eq. (42) becomes:

$$f_{\alpha}^{n+1} = \frac{2\tau - \Delta t}{2\tau + \Delta t} f_{\alpha}^{n} + \frac{\Delta t}{2\tau + \Delta t} (f_{\alpha}^{eq,n} + f_{\alpha}^{eq,n+1}) - \frac{2\Delta t}{2\tau + \Delta t} F_{\alpha}^{n}$$  \hspace{1cm} (43)

Now, instead of using the SOU for $F_{\alpha}^{n}$, it is calculated with the piece-wise linear (PL) Godunov-type flux scheme (which is also second-order) developed in [21]. The PL flux scheme is more accurate than the SOU approach because it calculates the averaged flux from $t_{n}$ to $t_{n+1}$. As a result, the PL flux scheme is also a function of $\Delta t$, so that the flux calculation will become more accurate as $\Delta t$ becomes smaller (please refer to [21] for more details). It can be seen that there is still unresolved implicitness embedded in $f_{\alpha}^{eq,n+1}$ in Eq. (43) (and Eq. (42)), which will be treated with different schemes in the following subsections.

### 6.1 The temporal extrapolation (TE) scheme

The same numerical procedure in Section 3.1 can be repeated here:

- **Step 1:** Calculate the moments with Eq. (14) with the newest $f_{\alpha}$;
- **Step 2:** Calculate $f_{\alpha}^{eq}$ with Eq. (13) based on the moments from the step 1;
- **Step 3:** Calculate $f_{\alpha}^{eq,n+1}$ with Eq. (12);
- **Step 4:** Update $f_{\alpha}$ with Eq. (43).

### 6.2 The variable transformation (VT) scheme

Since a different $\theta$ is used, the variable transformation process should be redone accordingly from the beginning. By defining another new variable $h_{\alpha}$ as:
\[ h_\alpha = f_\alpha - \frac{\Delta t}{2} C_\alpha \tag{44} \]

we have:

\[ h_\alpha^{n+1} = f_\alpha^{n+1} - \frac{\Delta t}{2} C_\alpha^{n+1} = f_\alpha^{n+1} + \frac{\Delta t}{2\tau} (f_\alpha^{n+1} - f_\alpha^{eq,n+1}) \tag{45} \]

Eq. (42) can then be rewritten as:

\[ h_\alpha^{n+1} = \frac{2\tau - \Delta t}{2\tau} f_\alpha^{n} + \frac{\Delta t}{2\tau} f_\alpha^{eq,n} - \Delta t F_\alpha^{n} \tag{46} \]

After \( h_\alpha^{n+1} \) is computed, \( f_\alpha^{n+1} \) can be recovered by using Eq. (45) such that:

\[ f_\alpha^{n+1} = \frac{2\tau}{2\tau + \Delta t} \left( h_\alpha^{n+1} + \frac{\Delta t}{2\tau} f_\alpha^{eq,n+1} \right) \tag{47} \]

Again, \( f_\alpha^{eq,n+1} \) can be resolved by the following relation because the new variable \( h \) has the same moments as the \( f \) variable:

\[ f_\alpha^{eq,n+1} = h_\alpha^{eq,n+1} \tag{48} \]

Then \( h_\alpha^{eq,n+1} \) is computed by Eq. (13) with the moments that are calculated by:

\[ \left[ \frac{\rho}{u} \right] = \Sigma_{\alpha=0}^{N} \left[ e_\alpha h_\alpha \right] \tag{49} \]

As a result, the new procedure of using the VT scheme on the new FVDBM is:

Step 1: Calculate the moments with Eq. (14) with the newest \( f_\alpha \);
Step 2: Calculate \( f_\alpha^{eq,n} \) with Eq. (13) based on the moments from the step 1;
Step 3: Calculate \( h_\alpha^{n+1} \) with Eq. (46);
Step 4: Calculate the moments based on \( h_\alpha^{n+1} \) with Eq. (49);
Step 5: Calculate \( h_\alpha^{eq,n+1} \) with Eq. (13) based on the moments from the step 4;
Step 6: Update \( f_\alpha \) with Eq. (47) by applying Eq. (48).

By comparing the numerical procedures of the new and the old FVDBM in Section 3.2, it can be seen that the new FVDBM requires the calculation of \( f_\alpha^{eq,n} \), which does not appear in the older approach. This is solely because both \( f_\alpha^{eq,n} \) and \( f_\alpha^{eq,n+1} \) are needed to update \( f_\alpha \) in the new FVDBM in which \( \theta = \frac{1}{2} \).

### 6.3. The semi-Lagrangian (SL) scheme

The calculation of the new FVDBM with the SL scheme becomes very simple:

Step 1: Calculate the moments with Eq. (14) with the latest \( f_\alpha \);
Step 2: Calculate \( f_\alpha^{eq,n} \) with Eq. (13);
Step 3: Update $f_\alpha$ with Eq. (43) by replacing $f_\alpha^{eq,n+1}$ with $f_\alpha^{eq,n}$ from step 2, which can further reduce Eq. (43) to:

$$
f_\alpha^{n+1} = \frac{2\tau\Delta t}{2\tau+\Delta t} f_\alpha^n + \frac{2\Delta t}{2\tau+\Delta t} f_\alpha^{eq,n} - \frac{2\tau\Delta t}{2\tau+\Delta t} f_\alpha^n
$$

(50)

6.4 The VT scheme modified by SL

As stated in the rule of thumb, $f_\alpha^{eq,n+1}$ can be replaced by $f_\alpha^{eq,n}$ anywhere it appears, including in Eq. (47) for the VT scheme. As a result, the calculations of the moments based on $h$ (Eq. (49)), the subsequent calculation of $h_\alpha^{eq,n+1}$ (Eq. (13)), and the final update of $f_\alpha^{eq,n+1}$ (Eq. (48)) can be completely avoided. Therefore, the VT scheme modified by the SL scheme can be realized in the following sequence:

Step 1: Calculate the moments with Eq. (14) with newest $f_\alpha$;
Step 2: Calculate $f_\alpha^{eq,n}$ with Eq. (13) based on the moments from step 1;
Step 3: Calculate $g_\alpha^{n+1}$ with Eq. (46);
Step 4: Update $f_\alpha$ with Eq. (47) by replacing $f_\alpha^{eq,n+1}$ with $f_\alpha^{eq,n}$ from step 2.

However, it is worth noting that the modified VT scheme will generate exactly the same solution as the SL scheme, because after replacing $h_\alpha^{n+1}$ with Eq. (46), Eq. (47) becomes Eq. (43) automatically. After that, substituting $f_\alpha^{eq,n+1}$ with $f_\alpha^{eq,n}$ will have the same effect as for the SL scheme.

6.5 Numerical comparisons and discussions

The four schemes, TE, VT, VT modified by SL, and SL, will be compared in this section in terms of accuracy, computational cost and stability. The numerical settings for the new comparison will be the same as the ones in Section 5, unless otherwise stated.

6.5.1 Accuracy

The transient errors from 0 to $t_c$ for the four schemes on the new FVDBM (new time marching and new flux scheme) with four sizes of $\Delta t$ are compared in Fig. 6. The first observation is that, as for the previous FVDBM, the SL is still able to generate much less error than the TE and VT scheme. In addition, after being modified by the SL approach, VT can also improve its accuracy. From the results, it can be seen that the SL and the VT modified by SL produce the same results, which echoes the analysis at the end of Section 6.4. However, all schemes in Fig. 6 increase the error with an increase in $\Delta t$ which differs from the behavior observed in Fig. 2, which can be more easily seen in Fig. 7 in which the percentage errors of different schemes are plotted against different sizes of $\Delta t$. This is because the error of the PL Godunov flux scheme increases with an increase in $\Delta t$, which changes the nature of the entire FVDBM solver to the one that satisfies the following condition, no matter which implicit collision scheme is used:

$$
\frac{\epsilon_1}{\epsilon_2} < \frac{\Delta t_1}{\Delta t_2}
$$

(51)

However, it is interesting to note in Fig. 7 that the errors of the modified VT and SL schemes grow at a lower rate than the other schemes when $\Delta t$ is increasing. This is because although the PL Godunov flux
scheme generates a higher error with a larger $\Delta t$, the SL and the modified VT implicit collision schemes themselves still decrease the error with a growing $\Delta t$ (Eq. (38)) while the other two are not affected by $\Delta t$ (Eq. (37)). As a result, the combined effects of the flux scheme and the implicit collision scheme will behave as shown in Fig. 7.

**Figure 6.** The $L_2$ errors of transient solutions of the new FVDBM with different implicit collision schemes during the time span from $0$ to $t_c$. (a) $\Delta t = 0.2\tau$; (b) $\Delta t = 0.4\tau$; (c) $\Delta t = 0.6\tau$; (d) $\Delta t = 0.8\tau$

**Figure 7.** The effects of $\Delta t$ on the errors of the new FVDBM solutions with different implicit collision schemes.
6.5.2 Computational cost

The update time $t_U$ (Eq. (39)) of new FVDBM solver with each of the four implicit schemes is measured and listed in Table 2. Again, it can be seen that the SL scheme is the fastest. In addition, the modified VT scheme is faster than the original VT scheme due to the saved computations for Eqs. (13) and (49), but still slower than the SL scheme. However, there are two major differences compared to the observations in Table 1 for the old FVDBM solutions. First, the update times for the new FVDBM solver with all implicit collision models are universally shorter than their counterparts in Table 1. This is because the PL Godunov flux scheme is faster than the SOU flux scheme. Second, in Table 1, the SL is roughly 2% faster than the VT scheme, but for the new FVDBM, the SL presents a 13% improvement in speed. The reason for this is that in the old FVDM, in which $\theta = 1$, both the VT and SL schemes require just one calculation of the equilibrium PDF. However, the VT scheme in the new FVDBM with $\theta = \frac{1}{2}$ requires the calculation of the equilibrium PDF two times, which was also stated at the end of Section 6.2, while the SL scheme still just needs one. Since the equilibrium PDF calculation is very costly, the difference in speed becomes larger.

Table 2. Update time for the new FVDBM solver with different implicit collision schemes

| Implicit collision scheme | $t_U$ (Unit: second) |
|---------------------------|----------------------|
| TE                        | $3.589 \times 10^{-5}$ |
| VT                        | $4.064 \times 10^{-5}$ |
| VT modified by SL         | $3.762 \times 10^{-5}$ |
| SL                        | $3.521 \times 10^{-5}$ |

6.5.3 Stability

The last test is the stability test on the new FVDBM with the four implicit collision schemes. The stability region for each scheme is shown in Fig. 8. The first difference from the results shown in Fig. 5 is that the $\Delta t/\Delta x$ limit becomes larger. This is the advantage of PL Godunov flux scheme over the SOU flux scheme. Consistent with Fig. 5, it can be seen as well that the SL scheme has a much higher $\Delta t/\tau$ limit than the TE scheme, and the same limit as the VT scheme. Additionally, the stability of the modified VT does not change compared to its original version, and also stays the same as the SL scheme since the SL and modified VT scheme are mathematically the same, as explained as the end of the Section 6.4.
7. Conclusions

In this paper, a new method to resolve the implicit BGK collision is developed for the finite volume discrete Boltzmann method (FVDBM). This new method stems from applying the semi-Lagrangian (SL) approach to the implicit equilibrium PDFs in the BGK collision. With the help of the first-order interpolation, the SL scheme becomes as simple as enforcing $f_{\alpha,eq}^{n+1} = f_{\alpha,eq}^n$. By comparing this new scheme with two existing ones, the temporal extrapolation (TE) and the variable transformation (VT) approaches, on the FVDBM with different time marching and flux calculations schemes, three advantages consistently can be found:

1) The SL scheme can dramatically improve the accuracy;

2) The SL scheme can lower the computational cost;
3) The SL scheme can significantly improve or preserve the stability in the $\Delta t/\tau$ limit compared to existing methods, depending on the comparative scheme;

In addition, this paper also demonstrates that the SL approach can be easily applied in any place where $f_{eq,n+1}$ is present by modifying the VT scheme with the SL scheme. The test results show that the modified VT scheme still presents the three advantages listed above.

8. Acknowledgements

Special thanks to Dr. Jing-Mei Qiu and Dr. Xiaofeng Cai from the University of Delaware for their constructive discussions and advice. The authors also thank the National Science Foundation for supporting this work under grant No. CBET-1233106.

References

[1] G. A. Bird, Molecular gas dynamics and the direct simulation of gas flows: Clarendon Press, 1994.
[2] S. Chen, G. D. Doolen, Lattice Boltzmann method for fluid flows, Annual Review of Fluid Mechanics 30 (1998) 329–364.
[3] S. Succi, G. Amati, R. Benzi. Challenges in lattice Boltzmann computing. Journal of statistical physics. 81, 5(1995).
[4] Gongwen Peng and Haowen Xi. On boundary conditions in the finite volume lattice Boltzmann method on unstructured meshes. International Journal of Modern Physics C, 10(6):1003–1016, 1999.
[5] Gongwen Peng, Haowen Xi, Comer Duncan, and So-Hsiang Chou. Lattice Boltzmann method on irregular meshes. Physical Review E, 58(4):R4124–R4127, oct 1998.
[6] Gongwen Peng, Haowen Xi, Comer Duncan, and So-Hsiang Chou. Finite volume scheme for the lattice Boltzmann method on unstructured meshes. Physical Review E, 59(4):4675–4682, apr 1999.
[7] S. Ubertini, G. Bella, S. Succi, Lattice Boltzmann method on unstructured grids: Further developments, Phys. Rev. E 68 (2003) 016701.
[8] A. Zarghami, M. Maghrebi, J. Ghasemi, S. Ubertini, Lattice Boltzmann finite volume formulation with improved stability, Commun. Comput. Phys. 12 (2012) 42.
[9] H. Xi, G. Peng, S. Chou, Finite-volume lattice Boltzmann method, Phys. Rev. E 59 (1999) 6202.
[10] M. Stiebler, J. Tölke, M. Krafczyk, An upwind discretization scheme for the finite volume lattice Boltzmann method, Comput. & Fluids 35 (2006) 814.
[11] D. V. Patil, K. N. Lakshmisha, Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh, J. Comput. Phys. 228 (2009) 5262.
[12] D. V. Patil, K. N. Lakshmisha, Two-dimensional flow past circular cylinders using finite volume lattice Boltzmann formulations, Internat. J. Numer. Methods Fluids 69 (2012) 1149.
[13] S. K. Choi, C. L. Lin, A simple finite-volume formulation of the lattice Boltzmann method for laminar and turbulent flows, Numer. Heat Transfer Part B. 58 (2010) 242.
[14] L. Chen, L. Schaefer. A unified and preserved Dirichlet boundary treatment for the cell-centered finite volume discrete Boltzmann method. Physics of Fluids. 27, 027104(2015).
[15] T. Lee and C. Lin, A characteristic Galerkin method for discrete Boltzmann equation, J. Comput. Phys. 171 (2001) 336.
[16] T. Lee and C. Lin, An Eulerian description of the streaming process in the lattice Boltzmann equation, J. Comput. Phys. 185 (2003) 445.
[17] K. E. Wardle and T. Lee, Finite element lattice Boltzmann simulations of free surface flow in a concentric cylinder, Comput. Math. Appl. 65 (2013) 230.

[18] R. Zhang, H. Fan and H. Chen. A Lattice Boltzmann approach for solving scalar transport equations. Philosophical Transactions of The Royal Society A. 369, 2264(2011).

[19] S. Ubertini, G. Bella, S. Succi. Lattice Boltzmann method on unstructured grids: Further developments. Physical review E. 68, 016701(2003).

[20] D Yu, R. Mei, L. Luo and W. Shyy. Viscous flow computations with the method of lattice Boltzmann equation. Progress in Aerospace Sciences. 39, 329(2003).

[21] L. Chen, L. Schaefer, Godunov-type upwind flux schemes of the two-dimensional finite volume discrete Boltzmann method. Computers & Mathematics with Applications, 75, 3105(2018).

[22] L. Chen, L. Schaefer and X. Cai. A fast and accurate second-order interpolation scheme for reducing the diffusion error of the finite volume discrete Boltzmann method. Journal of Scientific Computing, under review.

[23] 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 System. Physical review, 94, 511(1954). [24] A. Bardow, I. V. Karlin, and A. A. Gusev. General characteristic-based algorithm for off-lattice Boltzmann simulations. Europhysics Letters, 75(3): 434–440, 2006.

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

[25] A. Bardow, I. V. Karlin, and A. A. Gusev. General characteristic-based algorithm for off-lattice Boltzmann simulations. Europhysics Letters, 75(3): 434–440, 2006.

[26] X. He, S. Chen, G.D. Doolen. A novel thermal model for the lattice Boltzmann method in incompressible limit. J. Comput. Phys., 146 (1) (1998) 282.

[27] F. Bösch, I.V. Karlin. Exact lattice Boltzmann equation. Phys. Rev. Lett., 111 (9) (2013), p. 090601.

[28] S. Ubertini, P. Asinari, S. Succi. Three ways to lattice Boltzmann: a unified time-marching picture. Phys. Rev. E, 81 (2010), p. 016311.

[29] Z. Guo, T.S. Zhao. Explicit finite-difference lattice Boltzmann method for curvilinear coordinates. Phys. Rev. E, 67 (2003), p. 066709.

[30] Z. Guo, R. Wang, and K. Xu, Discrete unified gas kinetic scheme for all Knudsen number flows: Low speed isothermal case, Phys. Rev. E, vol. 88, p. 033305, 2013.

[31] Z. Guo, R. Wang, and K. Xu, Discrete unified gas kinetic scheme for all Knudsen number flows. II. Thermal compressible case, Phys. Rev. E, vol. 91, no. 3, pp. 1–11, 2015.

[32] Y. Zhang, L. Zhu, R. Wang, and Z. Guo. Discrete unified gas kinetic scheme for all Knudsen number flows. III. Binary gas mixtures of Maxwell molecules, Phys. Rev. E, vol. 97, no. 5, pp. 1–15, 2018.

[33] P. R. Rao and L. A. Schaefer, Numerical stability of explicit off-lattice Boltzmann schemes: A comparative study, J. Comput. Phys., vol. 285, pp. 251–264, 2015.

[34] M. Begue, A. Ghizzo, P. Bertrand, E. Sonnendrucker, O. Coulaud, Two-dimensional semi-Lagrangian Vlasov simulations of laser–plasma interaction in the relativistic regime, Journal of Plasma Physics 62 (1999) 367–388.

[35] N. Besse, E. Sonnendrucker, Semi-Lagrangian schemes for the Vlasov equation on an unstructured mesh of phase space, Journal of Computational Physics 191 (2003) 341–376.

[36] J. M. Qiu and C. W. Shu, Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow, J. Comput. Phys., vol. 230, no. 4, pp. 863–889, 2011.

[37] X. Cai, W. Guo, and J. M. Qiu, A High Order Conservative Semi-Lagrangian Discontinuous Galerkin Method for Two-Dimensional Transport Simulations, J. Sci. Comput., vol. 73, no. 2–3, pp. 514–542, 2017.

[38] A. Krämer, K. Källner, D. Reith, W. Joppich, and H. Foysi, Semi-Lagrangian off-lattice Boltzmann method for weakly compressible flows, Phys. Rev. E, vol. 95, no. 2, pp. 1–12, 2017.
[39] M. Groppi, G. Russo, and G. Stracquadanio. High order semilagrangian methods for the BGK equation. Communications in Mathematical Sciences, 14 (2): 389-414, 2016.