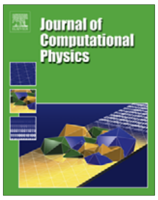
Journal of Computational Physics 228 (2009) 5262–5279

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh

Dhiraj V. Patil, K.N. Lakshmisha *

Department of Aerospace Engineering, Indian Institute of Science, Bangalore 560 012, India

ARTICLE INFO

Article history:
Received 10 July 2008
Received in revised form 24 February 2009
Accepted 9 April 2009
Available online 21 April 2009

PACS:
02.70.Ns
03.40.Gc
05.20.Dd
47.11.+j

Keywords:
Lattice Boltzmann
Unstructured mesh
Finite volume
TVD scheme
Limiters

ABSTRACT

A numerical scheme is presented for accurate simulation of fluid flow using the lattice Boltzmann equation (LBE) on unstructured mesh. A finite volume approach is adopted to discretize the LBE on a cell-centered, arbitrary shaped, triangular tessellation. The formulation includes a formal, second order discretization using a Total Variation Diminishing (TVD) scheme for the terms representing advection of the distribution function in physical space, due to microscopic particle motion. The advantage of the LBE approach is exploited by implementing the scheme in a new computer code to run on a parallel computing system. Performance of the new formulation is systematically investigated by simulating four benchmark flows of increasing complexity, namely (1) flow in a plane channel, (2) unsteady Couette flow, (3) flow caused by a moving lid over a 2D square cavity and (4) flow over a circular cylinder. For each of these flows, the present scheme is validated with the results from Navier–Stokes computations as well as lattice Boltzmann simulations on regular mesh. It is shown that the scheme is robust and accurate for the different test problems studied.

1. Introduction

Lattice Boltzmann Equation (LBE) offers an attractive means to simulate complex fluid flows. Here, the approach is to go to the molecular roots of fluid motion, by computing the dynamic evolution of the particle probability distribution function, as described by the Boltzmann kinetic equation discretized on a space–velocity lattice. Macroscopic flow-variables (velocity, pressure, density) are recovered as moments of the local and instantaneous particle probability distribution function. Over the last ten years, much work has been carried out on the development and application of LBE to simulate a variety of flows. Specifically, owing to its kinetic approach, LBE is expected to perform well for flows at mesoscopic scales where continuum assumption begins to breakdown. Several references [1–4] are available to obtain an entry to the theory and methodology of LBE.

The paradigm shift, from a continuum mechanics based Navier–Stokes (N–S) model to kinetic theory based LBE model, brings about two major changes in the governing equations [5]: (1) non-local effect (advective transport of momentum from neighborhood) becomes linear, and (2) local effect (molecular transport) becomes a non-linear source term. Further, owing to the weak-compressibility assumption, the LBE retains a hyperbolic character even for low Mach number flows in the

* Corresponding author. Tel.: +91 80 22933026; fax: +91 80 23601279.
E-mail address: kn.lakshmisha@yahoo.co.in (K.N. Lakshmisha).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.04.008

incompressible limit. As a result of these changes, numerical solution of LBE offers certain computational advantages. On the other hand, experience has shown that the LBE model as usually implemented, suffers from disadvantages due to which it is still struggling to compete with the N–S model for simulation of common flow problems. Some of these are, the excessive computational times, structured mesh, difficulty of applying boundary conditions etc. The computational resource requirement may be effectively overcome by the relative ease of parallelization of LBE codes, particularly due to availability of large-scale parallel computers.

In LBE, the physical spatial structure of the lattice is intrinsically coupled to the velocity-discretization of particle distribution function. The advantage is the Lagrangian, exact treatment of advective transport (propagation), and hence, a zero numerical diffusion. On the other hand, straightforward integration of the LBE on a non-uniform and/or unstructured mesh is not possible. This leads to difficulty in adapting the mesh to complex flow-structures such as separation, vortices, shear/boundary layers etc., and to satisfy boundary conditions on irregular geometries [6]. Such difficulty may be overcome by decoupling the numerical mesh from the lattice structure, and taking recourse to, one of finite difference (FD) [7–9], finite element (FE) [10–15], or finite volume (FV) [16–19] approaches. In this case, the particle advection step is solved in an Eulerian framework, and the problem of treatment of the advective terms reenters the solution procedure. This problem may be effectively addressed by suitably adapting the various techniques developed to handle the advective terms in the N–S equations solution [20–27]. In fact, in recent times, a few articles dealing with this task, have started to appear in the literature [28,29]. The FV approach and unstructured mesh technology are gaining popularity owing to their robustness and flexibility to handle complex flow-domains. The present work is aimed at making progress in the direction of developing a formulation of accurately solving LBE on an unstructured mesh system using a FV technique.

1.1. Types of unstructured mesh

The FV approach is based on the physical concept of satisfying the balance of different terms in the governing equation over discrete control volumes (CV). The entire flow-domain is discretized into a large number of CV, whose shape may be uniform (structured) or not (unstructured). Barth [30] provides an extensive theory and practical approaches of generating unstructured mesh. In the case of 2D unstructured triangular mesh, there are primarily four different ways of defining the control volume (see Fig. 1). The vertices of the median dual CV (Fig. 1(a)) can potentially lie outside the set of triangles defining the tessellation, thus reducing this to a poor choice. Another option is to form CV using centroid-dual (Fig. 1(b)). In this case a vertex of the control volume is the barycenter (centroid) of a triangle associated with the control volume, and hence always lies inside the triangle. This has been a popular choice with the CFD community, and has been followed previously for the finite-volume LBE (hereafter FV–LBE) simulations as well. However, this poses difficulties in the treatment of boundary conditions [31,32]. The cell-vertex mesh (Fig. 1(c)) is more suitable to the FD approach. In the present work, we develop a formulation on a cell-centered mesh system (Fig. 1(d)). This has not been used previously for LBE, but offers significant

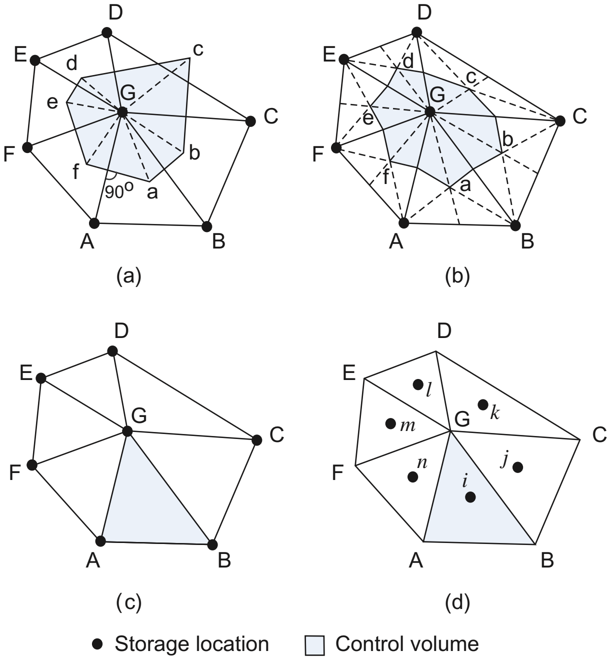


Fig. 1. Different types of unstructured mesh. (a) median dual (b) centroid dual (c) vertex centered and (d) cell centered.

advantages as shown in this paper. Here the CV coincides with a triangle, at the centroid of which the lattice is located. Very recently, a detailed comparison with respect to accuracy and efficiency has been made for centroid dual (vertex/node-centered) and cell-centered unstructured finite-volume discretizations of viscous terms in N–S equations [33]. In contrast to a vertex-centered system, the fixed number of interfaces (namely, three) associated with the cell-center system leads to a simpler implementation with an edge-based data structure, and is well suited for the linear reconstruction procedure proposed in this paper. Furthermore, the cell-center approach for a triangular unstructured mesh can be regarded as a natural extension of its counterpart for a rectangular structured mesh.

1.2. TVD schemes and limiters

Development of higher-order schemes with flux-limiters to solve hyperbolic conservation laws (in the context of Euler- and N–S equations) on unstructured mesh has been one of the most significant achievements of the CFD technology [20,30,34]. This is because, many higher-order schemes (derivatives of Godunov's scheme) which successfully eliminate artificial diffusion on structured mesh cannot be naturally extended to unstructured mesh. Jameson and Mavriplis [24] constructed schemes based on continuous piecewise linear elements, and flux-corrected transport terms. Barth and Jespersen [25] developed a universal flux-limiter for both cell-vertex and cell-centered tessellations. Frink [26] used a second-order scheme without a flux-limiter. Tamamidis [27] proposed a fully upwind and formally second order scheme with the min-mod limiter. Venkatakrishnan [20] observed that "...the unstructured grid technology is almost on par with the structured grid technology although encumbered with additional memory and computational cells; this overhead has to be balanced with the ability to compute flows over complex geometries and the ease of adaptation".

In FV approach, the evaluation of advective transport flux across the cell-surfaces takes a center stage. This calls for a knowledge of solution variables on the surface, which are reconstructed with the help of known solution at the neighboring cells. For solutions having large local gradients, such a reconstruction procedure may lead to spurious oscillations. Therefore, special procedures are devised to produce a high quality reconstruction (higher-order schemes) on the cell faces, leading to accurate evaluation of surface fluxes. These are known as Total Variation Diminishing (TVD) schemes. Such TVD discretization may be achieved by applying suitable 'limiter functions'. These functions are designed to reduce oscillations around steep gradient regions while allowing higher order schemes to operate elsewhere. The procedure is to add an antidiffusion to the low-order approximation in smooth regions. The TVD approach was originally designed for 1D systems [22], but considerable progress has been achieved in extending it to multi-dimensional flow computations. For higher-order upwind schemes to work on unstructured mesh, this true multi-dimensionality should be reflected during the reconstruction stage [20]. Furthermore, the use of limiters allows the control of numerical diffusion by achieving a higher-order solution in regions where the gradients of variables are not high. As shown in this paper, the TVD methodology can be successfully extended to the solution of FV–LBE.

1.3. FD/FV-LBE schemes

He et al. [35] stretched the lattice along the coordinate directions and simply interpolated the solution on locations not coincing with the square lattice. The key problem associated with the FD representation of LBE is that the density function is not well defined due to the lack of a volumetric measure, and Chen [17] offered a FV formulation. Succi and his collaborators [16] solved the integral form of LBE on a finite volume, defining a coarse-grained particle distribution function in each cell. This already incorporated the concepts of cell-centered FV, cell-averages of distribution function, and solution reconstruction. Peng [36] proposed the integration of the differential form of the LBE in finite volumes around grid points. This method works on unstructured grids, thus allowing an increased geometrical flexibility. However, the method suffers from substantial numerical instability compared to the standard LBE models. The computational efficiency of the scheme is not competitive with standard methods. Ubertini and Succi [37] emphasized the need to develop flexible and robust FV–LBE methods. Stiebler et al. [38] introduced a least square, linear reconstruction based upwind discretization scheme for the FV–LBE. The scheme substantially improved the stability properties even on coarser mesh with qualitatively same accuracy. Recently, the FV–LBE has been implemented for 3D flows [39]. The calculations with FV–LBE so far are based on cell-vertex tessellation, wherein large amount of book-keeping is required leading to increased memory-overhead costs. The application of boundary conditions at the boundary cells also becomes critical. The cell-centered approach is an easy way for minimizing the memory storage. Hence, herein we follow the cell-centered approach. Recently, Lee *et al.* [29] adopted the TVD Lax-Wendroff discretization on a FD stencil for the streaming step. They concluded that the development of a monotonicity preserving LBE method on unstructured mesh is urgent for practical applications. Finally, there is a need for better treatment of boundary conditions with unstructured mesh [38]. Considering all these, we have implemented the TVD concept on a cell-centered mesh system for FV–LBE. A new FV–LBE code has been developed to efficiently run on a parallel computing system. Further, we have comprehensively tested the new FV–LBE scheme for accuracy, convergence and robustness. The associated theoretical analysis, including estimation of the apparent viscosity of the FV discretization, will be presented in a companion paper.

The remainder of this article is organized as follows. In Section 2, the detailed mathematical formulation and the algorithm, including application of boundary conditions, of the TVD FV–LBE approach is described. In Section 3, the numerical implementation of the formulation is briefly outlined. In Section 4, the results of simulation of four example flows are presented and discussed. Finally, in Section 5, major conclusions are summarized.

2. Mathematical formulation

In this section, the finite volume TVD formulation of LBE simulation on unstructured mesh system is described. First the standard LBGK equations are presented, followed by the development of FV formulation. Then the procedure for flux calculation using a higher order solution reconstruction with TVD limiters is described. Finally, the treatment of edges on the boundary is explained.

2.1. The LBGK equation

The Boltzmann equation discretized in velocity space, and the collision term modeled with Bhatnagar–Gross–Krook approximation is usually written (see e.g., Ref. [40]) in the following differential form:

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

Here, $f_\alpha(\mathbf{x},t)$ is the particle distribution function and τ the relaxation time. Repeated Greek subscripts do not imply summation. To maintain simplicity of exposition, we consider the physical space to be 2-dimensional only ($\mathbf{x} : \mathbb{R}^2$). In the most commonly used D2Q9 lattice ($N = 8$), discrete particle velocities $\mathbf{e}_\alpha$'s are given as,

$$\mathbf{e}_\alpha = c\begin{cases} 0,0 & \text{for } \alpha = 0, \\ \cos[(\alpha-1)\pi/4], \sin[(\alpha-1)\pi/4] & \text{for } \alpha = 1,3,5,7, \\ \sqrt{2}\cos[(\alpha-1)\pi/4], \sqrt{2}\sin[(\alpha-1)\pi/4] & \text{for } \alpha = 2,4,6,8, \end{cases} \tag{2}$$

where $c = \sqrt{3RT}$ is the constant speed for an isothermal model [40], and the physical directions of α's are shown in Fig. 2. The equilibrium distribution function in its p-form is the low-velocity expansion of the Maxwellian distribution,

$$f_\alpha^{\mathrm{eq}} = w_\alpha\left[p + p_0\left\{3\frac{(\mathbf{e}_\alpha\cdot\mathbf{u})}{c^2} + \frac{9}{2}\frac{(\mathbf{e}_\alpha\cdot\mathbf{u})^2}{c^4} - \frac{3}{2}\frac{(\mathbf{u}\cdot\mathbf{u})}{c^2}\right\}\right]. \tag{3}$$

This form ensures that the incompressible N–S equations are recovered in the limit of small Knudsen and Mach numbers. The weights, w_α, for the D2Q9 model are distinct:

$$w_\alpha = \begin{cases} 4/9 & \text{for } \alpha = 0, \\ 1/9 & \text{for } \alpha = 1,3,5,7, \\ 1/36 & \text{for } \alpha = 2,4,6,8. \end{cases} \tag{4}$$

The local and instantaneous macroscopic flow-variables may be found from the moments as,

$$\begin{bmatrix} p(\mathbf{x},t) \\ \mathbf{u}(\mathbf{x},t) \end{bmatrix} = \sum_{\alpha=0}^{8} f_\alpha(\mathbf{x},t)\begin{bmatrix} 1 \\ \mathbf{e}_\alpha/p_0 \end{bmatrix} \tag{5}$$

The constant p_0 is set equal to 2.7. The local mass density being proportional to pressure, $\rho(\mathbf{x},t) = p(\mathbf{x},t)/c_s^2$.

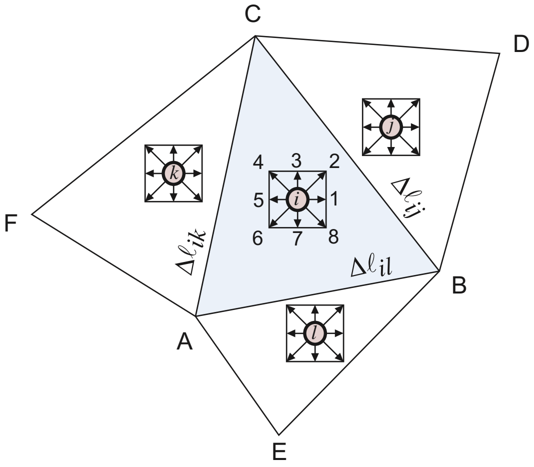


Fig. 2. Schematic of the FV discretization with cell-centered lattice.

2.2. Finite volume formulation

Shown in Fig. 2 is a schematic of the *cell-centered tessellation* [24,26]. The entire flow-domain is discretized into nonoverlapping triangles of arbitrary shape and size but fixed in time, whose vertices are indicated by $A, B, C, \ldots$ (although, in principle the foregoing formulation holds good for any polygonal tessellation). These triangles are the finite areas over which the conservation of f_α (Eq. (1)) is satisfied. The lattice center and the macroscopic flow variables are located at the geometric centroid $i, j, k, \ldots$ of each triangle. The edges of the ith triangle are denoted by $\Delta\ell_{im}(m = j, k, l)$. Important differences between the cell-centered TVD FV–LBE formulation and the LBE on a structured lattice are that, (1) here, a lattice is not a tessellation in the sense that it may, or may not overlap on its neighbors, (2) a lattice defined at a centroid may intrude into a neighboring triangle, and (3) there is no direct propagation from a lattice to its neighbor along the lattice-links in the directions of α's; rather, particle advection occurs across the common edge shared between two neighboring triangles, with appropriate Cartesian projections of $\mathbf{e}_\alpha$'s.

Denoting a generic cell ABC by Ω_i whose boundary is $\partial\Omega_i$ and area $A_i = \int_{\Omega_i} d\Omega$, Eq. (1) is written in an integral form as follows.

$$\frac{\partial}{\partial t}\int_{\Omega_i} f_\alpha d\Omega = -\int_{\Omega_i}\frac{1}{\tau}\left[f_\alpha(\mathbf{x},t) - f_\alpha^{\mathrm{eq}}(\mathbf{x},t)\right]d\Omega - \oint_{\partial\Omega_i}(\mathbf{e}_\alpha\cdot\mathbf{n})f_\alpha\, d\ell \quad \forall i \tag{6}$$

where $\mathbf{n}$ is a unit vector, locally normal to $\partial\Omega_i$ and pointing outward of Ω_i, and $\ell \in \partial\Omega_i$. The set of distribution functions spatially discretized and defined at the centroid $i(\equiv \mathbf{x}_i)$ of each finite volume Ω_i, is denoted by $f_{\alpha,i}(t)$. Further, the distribution function $f_{\alpha,i}$ is assumed to represent a cell-averaged value:

$$f_{\alpha,i}(t) = \frac{1}{A_i}\int_{\Omega_i} f_\alpha(\mathbf{x},t)d\Omega\,. \tag{7}$$

Then, Eq. (6) may be written in a semi-discrete form as

$$\frac{d}{dt}(f_{\alpha,i}) = \mathcal{L}(f_{\alpha,i}) = \mathcal{L}^C_{\alpha,i} + \mathcal{L}^A_{\alpha,i} \quad \forall i, \tag{8}$$

where

$$\mathcal{L}^C_{\alpha,i} = -\frac{1}{A_i\tau}\int_{\Omega_i}[f_\alpha(\mathbf{x},t) - f_\alpha^{\mathrm{eq}}(\mathbf{x},t)]d\Omega\,, \tag{9}$$

$$\mathcal{L}^A_{\alpha,i} = -\frac{1}{A_i}\oint_{\partial\Omega_i} F(f_\alpha)d\ell\,, \tag{10}$$

with the superscripts C and A referring to contributions from intracell particle–collision and intercell particle–advection, respectively. In Eq. (10), $F(f_\alpha)$ is the flux-density of f_α across the boundary,

$$F(f_\alpha) = (\mathbf{e}_\alpha\cdot\mathbf{n})f_\alpha\,. \tag{11}$$

Evaluation of different terms in Eq. (8) is described in the following subsections.

2.3. Time-integration

Let us indicate the discrete $f_{\alpha,i}$'s at time levels t and $t+\Delta t$ by the superscripts n and $n+1$, respectively. Assuming that the solution $f^n_{\alpha,i}$ at time t^n, is known, our goal is to compute the cell averages at the next time step t^{n+1}. Eq. (8) may be numerically integrated using different time-discretizations:

(1) First-order, forward-difference, explicit (forward Euler) scheme:

$$f^{n+1}_{\alpha,i} = f^n_{\alpha,i} + \Delta t\,\mathcal{L}\left(f^n_{\alpha,i}\right) \tag{12}$$

where, $\Delta t = t^{n+1} - t^n$ is the discrete timestep. The CFL (Courant–Friedrichs–Lewy) criterion based on advection is given by [26],

$$\Delta t = \mathrm{CFL}\frac{A_{\min}}{(|e|+c)\left(l^{(x)}_{\min} + l^{(y)}_{\min}\right)}, \tag{13}$$

where, the term CFL, be set less than 1 for stability, $A_{\min}$ is the minimal cell-area in the domain, $l^{(x)}_{\min}$ and $l^{(y)}_{\min}$ are the projected lengths of the minimal area cell on the x and y directions, respectively. This imposes stringent grid-induced stiffness (restriction) on the allowable time-step and local time-stepping [26] may be performed to accelerates the convergence to steady state by advancing the solution at each cell in time at a CFL number near to local stability limit. The collision term also enforce a stability criterion on forward Euler time-marching [37] and is given as,

$$\Delta t \leqslant 2\tau. \tag{14}$$

The smaller of the two time-steps obtained from Eqs. (13) and (14) may be used in the simulations.

(2) Second-order, central-difference, explicit leapfrog scheme:

$$f_{\alpha,i}^{n+1} = f_{\alpha,i}^{n-1} + 2\Delta t\,\mathcal{L}\left(f_{\alpha,i}^{n}\right) \tag{15}$$

(3) Second order, explicit Adams-Bashforth scheme:

$$f_{\alpha,i}^{n+1} = f_{\alpha,i}^{n} + \frac{\Delta t}{2}\left[3\,\mathcal{L}(f_{\alpha,i}^{n}) -\ \mathcal{L}\left(f_{\alpha,i}^{n-1}\right)\right]. \tag{16}$$

This scheme requires saving fields from two previous time-steps and the time-step is limited by a CFL condition.

(4) Second-order TVD Runge–Kutta scheme [41]:

$$f_{\alpha,i}^{(1)} = f_{\alpha,i}^{n} + \Delta t\,\mathcal{L}\left(f_{\alpha,i}^{n}\right) \tag{17}$$

$$f_{\alpha,i}^{n+1} = \frac{1}{2}f_{\alpha,i}^{n} + \frac{1}{2}f_{\alpha,i}^{(1)} + \frac{1}{2}\Delta t\,\mathcal{L}\left(f_{\alpha,i}^{(1)}\right) \tag{18}$$

Gottlieb and Shu [41] discuss the accuracy, stability, and convergence of TVD Runge–Kutta schemes. Strategies like local time-stepping may be employed to accelerate convergence. However, if the flow is unsteady, or is evolving towards a steady-state, then a higher order scheme (e.g., 4th order Runge–Kutta) may be required for integrating Eq. (8). Recently, Ubertini et al. [42] proposed a scheme, which allows for time-step more than an order of magnitude above the standard LBE stability threshold.

For steady state problems, initial conditions are arbitrarily chosen. In the simulations presented in the present paper, all velocities in internal centroids are initialized as zero. Pressure is taken as p_0 (may be set to a representative value of 2.7). This yields the initial values of the functions $f_\alpha, f_\alpha^{\text{eq}}$. Like conventional lattice Boltzmann method, TVD FV–LBE is also time-marched with collision- and streaming-steps carried out in two separate stages. Both of these fluxes are calculated from the solution at t^n and time-marching is performed thereafter.

2.4. Collision terms

Since the distribution functions f_α's (and their corresponding equilibrium counterparts, f_α^{eq}'s) are assumed to be uniform over Ω_i, the integral in Eq. (9) may be evaluated using the trapezoidal rule:

$$\mathcal{L}_{\alpha,i}^{C} = -\frac{1}{A_i\,\tau}\int_{\Omega_i}\left[f_\alpha(\mathbf{x},t) - f_\alpha^{\text{eq}}(\mathbf{x},t)\right]d\Omega = -\frac{1}{\tau}\left(f_{\alpha,i} - f_{\alpha,i}^{\text{eq}}\right). \tag{19}$$

Here nonlinearity enters through the quantities $f_{\alpha,i}^{\text{eq}}$'s which are functions of the local macroscopic flow variables (Eqs. (3) and (5)). The collision term represents the microscopic form of the viscous transport term in macroscopic scale. The relationship between (τ, ν) is $\nu = \tau c_s^2$ in the FV–LBE method, while it is $\nu = \left(\tau - \frac{1}{2}\right)c_s^2$ in the traditional LBE method on a square lattice (see Ref. [43]). Here, the sound-speed of the model is $c_s = \sqrt{RT}$. Further, the relaxation time τ is related to the Reynolds number of the flow as,

$$\tau = \nu/c_s^2 = \frac{u_{\text{ref}}L_{\text{ref}}}{\text{Re}}\,\frac{1}{c_s^2}, \tag{20}$$

where u_{ref} is the characteristic flow-velocity, and L_{ref} is the characteristic flow-dimension. We note in passing that, although the BGK collision model is chosen here, in principle, the formulation can adopt any improved relaxation model, such as given in Ref. [44].

2.5. Advective fluxes

Now we consider a key element of the present formulation: contribution to the balance of particle distribution function $f_{\alpha,i}$ within a cell Ω_i, due to microscopic streaming of particles across the cell-boundary $\partial\Omega_i$. Eq. (10) may be discretized, noting that the integration path $\partial\Omega_i$ is actually a collection of edges of length $\Delta\ell_{im}(m = j,k,l)$ bounding the triangular cell Ω_i,

$$\mathcal{L}_{\alpha,i}^{A} = -\frac{1}{A_i}\sum_m F(f_{\alpha,im})\Delta\ell_{im}, \quad \text{for } m = j,k,l \tag{21}$$

in which $f_{\alpha,im}$ represents the value of function at the midpoint of the edge $\Delta\ell_{im}$. Hence, one needs to consider the evaluation of flux on an edge-by-edge basis for each cell. Two types of edges are possible, namely, those (1) in the interior, and (2) on the boundary of the flow domain, the treatment of which follows subsequently.

Accurate evaluation of RHS of Eq. (21) consists of two steps. In the first step, the midpoint value $f_{\alpha,im}$ of the solution is reconstructed within Ω_i, including $\partial\Omega_i$. The second step involves computing the flux density F along the edge. The solution may then be advanced to the next time level. Particular attention needs to be paid to solution reconstruction and flux-density evaluation, since accuracy of these operations determines the spatial accuracy of steady state problems.

2.5.1. Numerical flux on the edge

Higher-order extension of Godunov's scheme to arbitrary shaped cells assumes piecewise polynomial approximations in each cell (expanded about the centroid). Since these piecewise polynomials are discontinuous from cell to cell, along an edge (see Fig. 3), two distinct values of the solution (f^L_α, f^R_α), and hence, two values of F, may be reconstructed using Eq. (11) from the left- and right-neighboring centroids across the edge. These generate waves traveling at characteristic speeds, one going into the cell and the other going out. Therefore, a sequence of Riemann problems arises, but fortunately F is linear in the present case. This nonuniqueness problem is overcome by supplanting the actual flux $F(f_{\alpha,ij})$ across the edge by a uniquely defined 'numerical flux' $\overline{F}(f^L_\alpha, f^R_\alpha, \mathbf{n}_{ij})$; where, $\mathbf{n}_{ij}$ is a unit outward normal to an edge shared between cell i and j. For this purpose, the Roe's flux-difference splitting scheme [21] for a solution to the approximate Reimann problem is employed:

$$\overline{F}(f_{\alpha,ij}) = \frac{1}{2}\left[F(f^R_\alpha) + F(f^L_\alpha) - |\alpha^{(ij)}|(f^R_\alpha - f^L_\alpha)\right]. \tag{22}$$

The reconstruction procedure for left and the right states of the solution (f^L_α, f^R_α) is described subsequently. The factor $|\alpha^{(ij)}|$ is the scaled characteristic speed, which is taken to be equal to the scaled microscopic velocity normal to the edge, $\mathbf{e}_\alpha \cdot \mathbf{n}_{ij}$. Consequently the corresponding scheme Eq. (21) becomes

$$\mathcal{L}^A_{\alpha,i} = -\frac{1}{A_i}\sum_m \overline{F}(f_{\alpha,im})\Delta\ell_{im}, \quad \text{for } m = j,k,l. \tag{23}$$

Calculation of the left and right states in Eq. (22) is the key to achieve stability and high-order accuracy.

2.5.2. Solution reconstruction; Limiters

A proper reconstruction procedure should satisfy several criteria, as discussed in, for example, Refs. [21,45]. Two most commonly used reconstruction procedures are described below.

(a) *Piecewise constant*. The solution $f_\alpha(\mathbf{x})$ is assumed to be constant everywhere within the cell (including its edges), being equal to its value at the centroid:

$$f_{\alpha,ij} = f_{\alpha,i^*}, \tag{24}$$

which is only first order accurate. Here, i^* indicates the *upwind centroid* for the edge $\ell_{i\sigma}$. Referring to edge BC in Fig. 3,

$$i^* = \begin{cases} i & \text{for } \mathbf{e}_\alpha \cdot \mathbf{n}_{ij} > 0; \\ j & \text{for } \mathbf{e}_\alpha \cdot \mathbf{n}_{ij} \leqslant 0. \end{cases} \tag{25}$$

A space-centered, second-order scheme (used by Peng et al. [36]) has the form

$$f_{\alpha,ij} = \lambda f_{\alpha,i} + (1-\lambda)f_{\alpha,j}, \tag{26}$$

where j is the neighboring centroid across edge ij, and λ is a linear interpolation weight, based on the distances of the centroids i and j from the edge. Although only first order in space, the upwind scheme Eq. (24) is preferred over Eq. (26) owing to better stability when dealing with strong gradients in solution. Central schemes produce wiggles and unphysical oscillations, and possibly negative values of solution f_α. However, the upwind scheme (1) exhibits numerical diffusion and viscosity which may affect the simulation results, and (2) is inconsistent on the unstructured mesh.

(b) *Piecewise linear*. A second order accuracy is achieved by performing 2D, piecewise linear reconstruction of cell-averaged data using a Taylor expansion:

$$f_\alpha(\mathbf{x}) = f_\alpha(\mathbf{x}_i) + \nabla f_{\alpha,i} \cdot (\mathbf{x} - \mathbf{x}_i) \quad \text{for } \mathbf{x} \in \Omega_i, \tag{27}$$

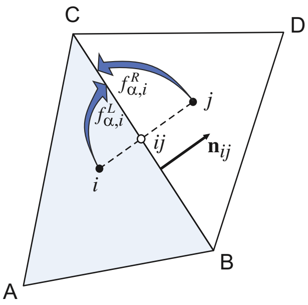


Fig. 3. Advective flux calculations; Reimann problem.

but this requires a knowledge of the gradient of the solution $\nabla f_{\alpha,i} \equiv (\nabla f_\alpha)_{\mathbf{x}_i}$ at the centroids. An upwinding form of Eq. (27) has been used by Stiebler et al. [38] for FV–LBE simulations. The linear reconstruction step is computationally as expensive as the collision and the propagation step, so the total nodal update cost increases by about 50% over that in the original LBE method. In this, the value of reconstructed function varies linearly along an edge. It must be ensured that, such a reconstructed function does not attain a local extremum, but remains limited to within the range of the solution at the neighboring centroids. This is a serious difficulty when dealing with solutions which have large local gradients, and tend to predict under/overshoots. This led to concepts of Total Variation Diminishing (TVD) and limiter functions. Essentially, the second term on R.H.S. of Eq. (27) is replaced by a nonlinear function which adapts to the local solution and its gradient, to yield a high-quality reconstruction on the cell boundary. As discussed in Section 1, this topic is well developed in the CFD of N–S approach, but only currently being implemented in LBE.

In the form introduced by Roe [46] which has become quite popular, the left and right states of the solutions are reconstructed as,

$$\left.\begin{aligned} f_\alpha^L &= f_{\alpha,i} + \tfrac{1}{2}\Phi_{ij}(f_{\alpha,j} - f_{\alpha,i}) \\ f_\alpha^R &= f_{\alpha,j} + \tfrac{1}{2}\Phi_{ij}(f_{\alpha,i} - f_{\alpha,j}) \end{aligned}\right\}. \tag{28}$$

Here, $f_{\alpha,i}$ represents the distribution functions at the centroid i, and $f_{\alpha,j}$ are the distribution functions at its neighboring centroid j. The nonlinear function $\Phi_{ij}(r)$ is known as *flux limiter* and $r > 0$ the *consecutive gradient*, to be discussed below. A wide range of elegantly designed flux limiters have been developed and used for high-order solution reconstruction. Sweby [47] has defined a second-order TVD-region. The upper boundary of this region is equivalent to superbee limiter (most compressive) and that the lower boundary is equivalent to minmod limiter (most diffusive). Herein, following limiter-functions are used for illustrative purpose.

$$(1)\ \text{Piecewise constant}: \Phi_{ij}(r) = 0, \tag{29}$$

$$(2)\ \text{Piecewise linear(without limiter)}: \Phi_{ij}(r) = 1, \tag{30}$$

$$(3)\ \text{Minmod}: \Phi_{ij}(r) = \max[0, \min(r, 1)], \tag{31}$$

$$(4)\ \text{Superbee}: \Phi_{ij}(r) = \max[0, \min(2r, 1), \min(r, 2)]. \tag{32}$$

Although, the use of other limiter-functions such as MUSCL (monotonic upwind-centered scheme for conservation laws) [48] or Venkatakrishnan [49] is similar.

To achieve a high-order reconstruction, the r-factor requires to be evaluated using solution values at three consecutive locations, collinear along the normal $\mathbf{n}_{ij}$; this is easily determined on a structured mesh. Herein lies the main difficulty of implementing TVD schemes on unstructured mesh, and a few suggestions have been offered in the literature [50,51]. We successfully implemented a technique recently suggested [52] to evaluate r-factor. Shown in Fig. 4 is a schematic of the situation in case of a CV whose centroid is i, and solution being reconstructed on the edge BC. Here the task is to identify a 'virtual' upwind node when $\mathbf{e}_\alpha \cdot \mathbf{n}_{ij} \gtrless 0$, but the positive case is considered for illustration. This is achieved by extending the line joining centroids i and j in the upwind direction, by a distance equal to that between i,j, and indicated as s in Fig. 4. This yields the virtual upwind location i^+ lying inside the cell whose centroid is i^*. Then the r-factor used in Eqs. (31) and (32) is evaluated as

$$r = -\frac{\mathbf{d}_{i^* i^+} \cdot \nabla f_{\alpha,i^*}}{f_{\alpha,j} - f_{\alpha,i}} + \frac{f_{\alpha,i} - f_{\alpha,i^*}}{f_{\alpha,j} - f_{\alpha,i^*}}, \tag{33}$$

where, $\mathbf{d}_{i^* i^+}$ is the vectorial distance between i^+ and i^*. To use Eq. (33), the gradient of the solution at the virtual upwind centroid i^* is necessary. This is reconstructed from a least squares approach as described in the following.

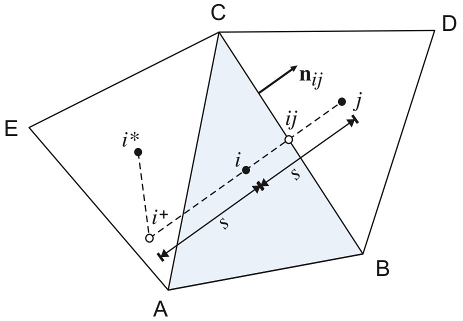


Fig. 4. Finding the 'virtual' upwind centroid i^* to calculate r-factor.

2.5.3. Gradient reconstruction

At each centroid, the gradient of the solution must be determined such that the solutions are optimally extrapolated to the edges. Different procedures have been proposed in the literature for a 2D linear reconstruction of solution gradient at centroid. However, a least-squares minimization method [38] is found suitable and sufficient in the present work. Herein, the gradient in Eq. (33) is chosen such that, the reconstructed value of the distribution function at the neighboring centroids becomes as close as possible to its actual value. Following Stiebler et al. [38], the optimal estimation of $\nabla f_{\alpha,i}$ is the one that minimizes the total squared-error:

$$\min_{\nabla f_{\alpha,i}} \sum_m w_{i,m} \{ f_{\alpha,m} - f_{\alpha,i} - \nabla f_{\alpha,i} \cdot (\mathbf{x}_m - \mathbf{x}_i) \}^2 \tag{34}$$

where, $m = j, k, l$ indicates the three neighboring centroids of the cell i; and $w_{i,m} = 1/(\mathbf{x}_m - \mathbf{x}_i)^2$ is a geometrical weighting factor. These calculated gradients of the distribution functions are used in the Eq. (33) to obtain the consecutive gradient.

2.6. Treatment of boundary conditions

As shown in Fig. 5(a), let ABC be a cell, such that the edge AB forms a part of the boundary of the flow-domain. Here we illustrate the treatment for a no-slip boundary, and other conditions may be similarly treated. A ghost cell ABC' is constructed as the mirror image of the cell ABC, as shown. The centroids of cells ABC and ABC' are denoted by i and i', respectively. The advection flux through the boundary edge needs to be calculated. An accurate estimation of the distribution functions at the mirror-cell centroid (i') should be performed which may mimic the physical boundary condition along AB. For further illustration, refer to Fig. 5(b), which shows the centroids i and i' with the fictitious boundary node, b. We implement the non-equilibrium bounce-back rule [31] at the fictitious boundary node to satisfy the boundary condition for boundaries in the co-ordinate directions. The flow-variables (u, v, p) at the node i and at the boundary are known. These quantities at the mirror-cell centroid (i') may be calculated using extrapolation and corresponding equilibrium distribution functions $\left(f^{eq}_{\alpha,i'}\right)$ may be set at i'. Following set of equations are used to calculated the distribution functions at the fictitious boundary node, b. The unknown distribution functions for the shown geometrical-case in Fig. 5(b) and with D2Q9 model are $f_{2,b}$,$f_{3,b}$ and $f_{4,b}$.

$$\left.\begin{aligned} f_{\alpha,b} &= 0.5\left(f_{\alpha,i} + f^{eq}_{\alpha,i'}\right); \quad \alpha \neq 2,3,4 \\ f_{2,b} &= p_0 u_b/2 + p_0 v_b/6 + f_{6,b} + (f_{5,b} - f_{1,b})/2, \\ f_{3,b} &= 2 p_0 v_b/3 + f_{7,b}, \\ f_{4,b} &= -p_0 u_b/2 + p_0 v_b/6 + f_{8,b} - (f_{5,b} - f_{1,b})/2. \end{aligned}\right\} \tag{35}$$

These unknown distribution functions obtained at the fictitious boundary node, b are extrapolated to the mirror-cell centroid (i'). We use simple bounce-back rule for boundaries not parallel to the co-ordinate directions. For an improved boundary treatment for such an inclined boundaries, authors here suggest, first for rotation of the distribution function space coinciding with an inclined surface and then following the implementation of non-equilibrium bounce-back rule. The buffer layer is used by Ubertini and Succi [37], to exploit the application areas of flow with open boundaries. The use of a single buffer layer to implement constant longitudinal gradient of the flow variables has been made. For periodic boundary condition it has

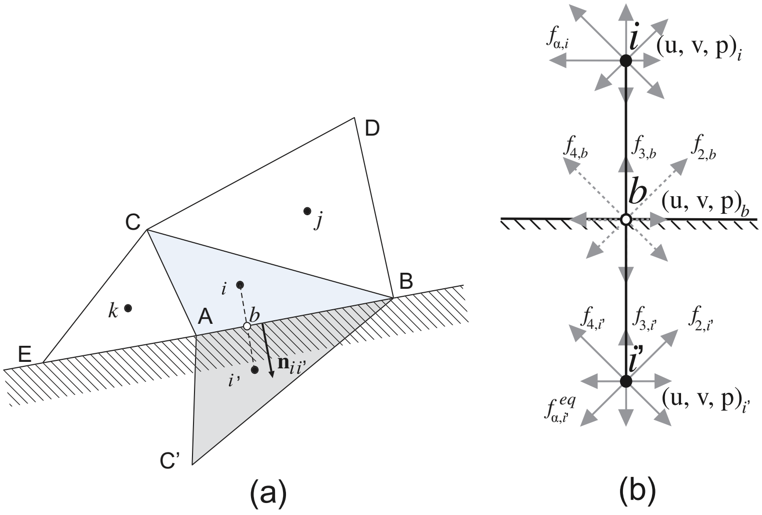


Fig. 5. (a) Boundary cell ABC and ghost cell ABC' and (b) ghost cell distribution functions.

been ensured that the triangular elements at the two boundaries are mirror-cells of each other. Therefore, the flux of distribution functions leaving one boundary cell through its boundary edge, enters through the other cell boundary.

Shown in Fig. 6(a) is a cell *ABC* residing at the boundary. The calculations of the distribution functions at the mirror-cell centroid (i') has been explained earlier in the beginning of this section. The unknown distribution functions should be calculated at the potential 'virtual' upwind nodes (i^+, i^-), which are lying outside the computational domain (see Figs. (4) and (6)). An extrapolation of distribution-functions from interior to the boundary and then from boundary to these 'virtual' upwind nodes may be performed.

3. Numerical implementation

The TVD FV–LBE formulation has been coded in C++ language; the computations were parallelized using 'domain decomposition' technique, with the implementation of Message Passing Instructions (MPI) library subroutines. Our code basically evolved from its predecessor for the structured LBE implementation [53]. The unstructured mesh was generated using the commercial software `FEMLab`. After the mesh is generated, the data (on connectivity, etc) is sorted in a preprocessor to decompose the entire flow domain into a given number of subdomains for parallelization. The individual centroids and the links bounding these are tagged as one of the three types: an internal node, a boundary node, or an inter-processor border node. Our preprocessor consumes just about 2% of total computing time of a steady state run. Each processor performs computations on a certain subdomain and exchanges information with other processors along the borders separating the domains. By using a ghost layer of CV's in the surrounding of the subdomain, the advection step can be isolated from the data exchange step. It was ensured that each processor is assigned an approximately uniform number of mesh points, so that the load among different processors is balanced. After the advection step, the values in the ghost layer are sent to the neighboring processor. Hence, the computation is independently carried out centroid-by-centroid in the TVD FV–LBE method. Due to the local character of the LBE, the parallelization by simple domain decomposition is straightforward and brings good results concerning the parallel speed-up.

Like conventional lattice Boltzmann method, TVD FV–LBE simulation is also time-marched with collision and streaming steps. Both of these fluxes are calculated using the latest solution available, and then time-marching is performed to obtain solution at the next time level. At the end of one time-step, individual processors deliver the solution within their respective subdomains at the new time level. This data is post-processed to prepare a single file for graphical analysis.

4. Results and discussion

The computer code with the new LBE formulation has been used to simulate four benchmark problems. Computations were carried out to investigate convergence and accuracy of solution, and efficiency of parallelization. The results are presented and discussed in this section.

4.1. Laminar flow in a 2D, plane channel

Poiseuelle flow in a 2D channel is a simple benchmark for flows with open boundaries. At the inlet, we imposed a fully developed profile for the *x*-velocity (parabolic in *y*-direction), and the pressure was linearly extrapolated from the interior

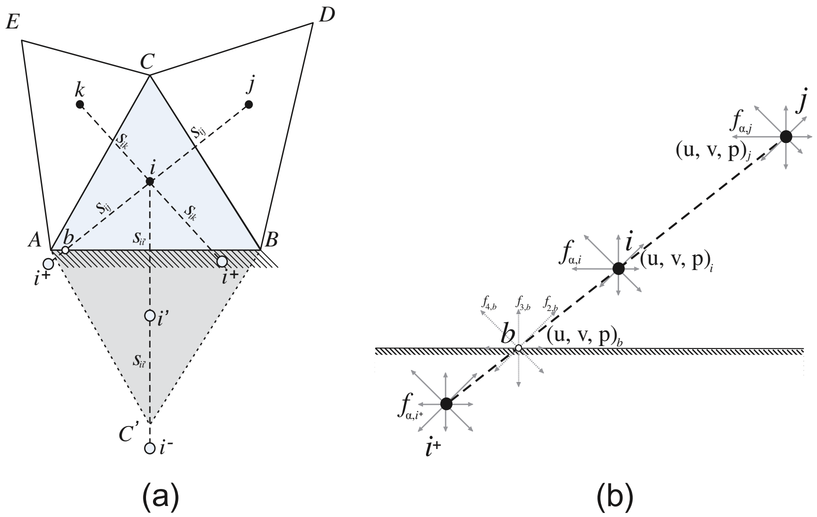


Fig. 6. (a) Virtual upwind nodes to calculate *r*-factor at the boundary and (b) distribution functions at the virtual upwind node.

nodes. At the outlet, the pressure was fixed whereas the velocities were extrapolated from interior nodes. On the two sidewalls, the no-slip boundary condition was imposed for velocities whereas the normal gradient of pressure was taken as zero. Shown in Fig. 7(a), is the computed velocity profile at the outlet section whereas in Fig. 7(b), is the centerline pressure along the length of the channel for Re = 40 (Re is based on channel height and maximum inlet velocity).

The flow was simulated with three mesh systems of progressively higher refinement (7376, 30728 and 53816 control volumes, respectively). The number of elements in the y-direction are progressively refined. The three nonuniform grid-systems used has 20, 40 and 60 elements in the y-direction. Fig. 7 shows a good convergence of the numerical solution towards the exact solution. These simulations were done using the superbee limiter. On the finest mesh, the computed value of the total momentum leaving the exit plane, $\int_{-h/2}^{+h/2} \rho(y)u_x(y)dy$, had an error of about 2.65% of the exact value. Further, the average pressure at the inlet, $\frac{1}{h}\int_{-h/2}^{+h/2} p(y)dy$ was computed within 0.28% of the exact value.

4.1.1. Convergence history

Fig. 8(a) compares the convergence history during the first non-dimensional time of $t = 0.15(t = N^*\Delta t; N^*$ being number of time-steps) of the solution using the limiter functions described earlier in Section 2.5.2. In these simulations, we used a timestep of $\Delta t = \tau/10$ (see Ref. [37]). The r.m.s of the change in pressure field over the entire flow-domain is taken as a criterion to examine the approach of solution to steady-state:

$$\Delta p_{\text{tot}} = \frac{1}{N}\sqrt{\sum_i \left(p_i^n - p_i^{n-1}\right)^2}, \tag{36}$$

where, Δp is the root-mean-square change in pressure, p^n is the pressure at time-level n, and N is the total number of control volumes. In Fig. 8(a), the minmod limiter shows better convergence. This convergence history has a periodic behavior. The r.m.s. change in pressure falls to the value of 10^{-11} and tends to zero for this limiter function. The superbee limiter has an oscillatory behavior in the range of 10^{-8} to 10^{-9}. These simulations were performed on Grid 2. Fig. 8(b) shows the convergence of superbee limiter for different grid-systems studied. The poorer convergence property of superbee limiter may be attributed to its non-differentiable functional form [49].

4.1.2. Parallel efficiency

The efficiency of parallelization was studied with reference to the channel flow problem. The flow domain was partitioned into 2, 4, 8, 16, 20, 28 and 32 number of subdomains, each of which was assigned to individual computer processors. Ghost nodes were defined at the interface of two subdomains, for data-exchange. The parallel computations were carried out on an IBM P720 cluster. The total CPU time for 10,000 timesteps has been compared with reference to the total CPU time required for the same number of timesteps in a 2-processor simulation. Fig. 9 shows the speedup gained by the parallel computations, as well as a linear (ideal) variation. It is observed that the gain in speedup is almost proportional to the number of processors employed. However, when more than 14 processors are employed, there is a dip in the gain, due to the fact that increasingly more CPU is consumed in data-exchange, at the cost of useful CPU spent in actual calculations. However, this effect decreases as the problem size (total number of cells) increases.

4.2. Unsteady Couette flow

The configuration of the unsteady Couette flow is similar to that of the 2D plane channel flow, but now the flow is driven by the top plate moving with a constant velocity U along the x-direction, and the bottom plate is stationary. The non-equi-

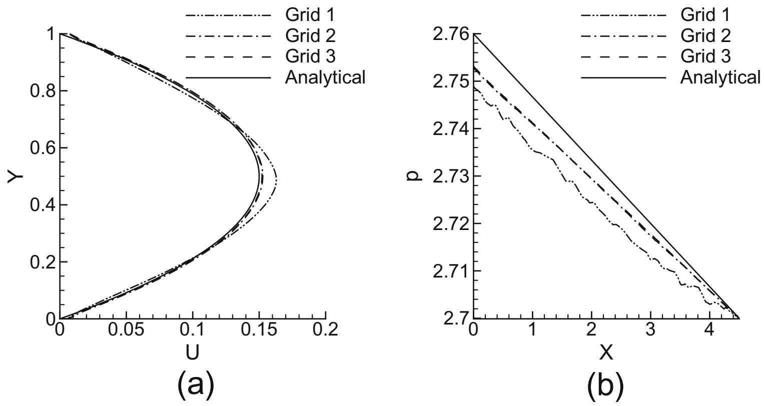


Fig. 7. Simulation of laminar flow in a plane 2D channel. Comparison of present results using different mesh resolution levels (Grid1 = 7376, Grid2 = 30728, Grid3 = 53816 control volumes). (a) Velocity on the exit plane and (b) Pressure along the centerline.

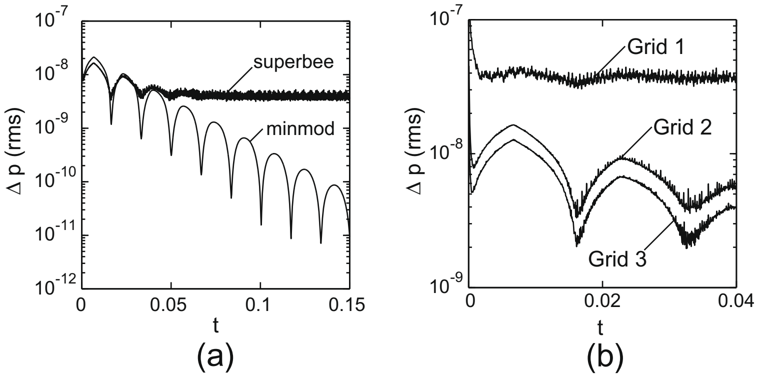


Fig. 8. Convergence history for laminar flow in a plane 2D channel. (a) comparison of limiters on the same mesh (Grid 2) and (b) for superbee limiter, results with different mesh systems.

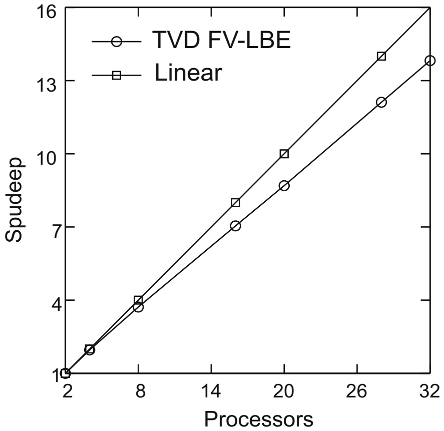


Fig. 9. Parallel speedup relative to 2 processors on IBM P720 cluster (simulation of laminar flow in a plane 2D channel on Grid 2).

librium bounce-back scheme is applied to the top and bottom walls, and the periodic boundary condition is applied in the x-direction. The domain has 4 triangular elements in x-direction, and 256 elements in y-direction. The Reynolds number of this flow is defined as, $\mathrm{Re} = HU/\nu = 10$, where H is channel height in y-direction. We set $H = 1.0, U = 0.1$, and with $\Delta t = 0.0002$. Initially, the velocity is set to zero in the whole computational domain. The expression for analytical solution of unsteady Couette flow may be found in [10]. Fig. 10 shows comparison of velocity profiles at different non-dimensional times using Eq. (12) along with the analytical solution at the corresponding times. The average-error at each of these times between the simulations and the analytical solution is found to be with 0.4–0.7%. We also have estimated numerical diffusion and it is found to be less than 1% for all the simulations performed.

Further, we have simulated the unsteady Couette flow by using TVD–RK2 time-stepping. It was found that the time step for the TVD–RK2 scheme may be taken as 2.1 times greater than that for the Euler explicit scheme. But, with the TVD–RK2 scheme, calculations have to be done two times in each time-step (see Eqs. (17) and (18)). We have also investigated the effect of different relaxation times by changing the Reynolds number of the flow. The relaxation time τ has been set to 0.01, 0.002, 0.001. The numerical results are within 1% of the analytical solutions at these relaxation times. This verifies the conclusion made by Ubertini et al. [43] on the residual viscosity not being contaminating the relation of (τ, ν) even with τ as low as 0.001.

4.3. Lid-driven flow in a 2D square cavity

The third problem we investigated is the lid-driven flow in a 2D square cavity. Previously, LBE simulations have been extensively validated [31,53,54]. The N–S solution results of Ghia et al. [55] are usually taken as a benchmark data. Except for very high Reynolds numbers ($Re \geqslant 5000$), the LBE simulation (on structured mesh) results have compared excellently with the N–S results. Here, Reynolds number is defined as $Re = UL/\nu$, where U is the lid-velocity, L the cavity width, and ν the kinematic viscosity of the enclosed fluid in motion. Results of LBE simulations on unstructured mesh were presented by Ubertini and Succi [37], for a maximum Reynolds number of 1000.

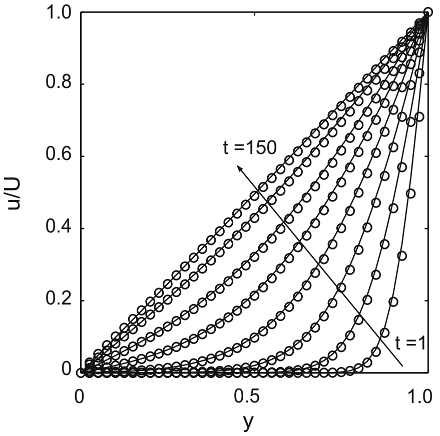


Fig. 10. Unsteady Couette flow; solid line: analytical, circles: simulations.

4.3.1. Grid-convergence studies

To investigate the grid-convergence of the numerical solution, the simulations have been performed on systematically refined grids. We have carried out the grid-refinement starting with a coarse mesh consisting of 32^2 number of uniform triangles. Successively, we divided each triangular element into four triangles (which is equivalent to dividing each edge of the triangle into two parts), as shown in Fig. 11(a). The resultant grid-systems have $64^2, 128^2$ and 256^2 number of uniform triangles. The lid velocities have been set to $u = 0.1, v = 0$ and remaining computational domain has been initialized with $u = 0, v = 0$. The distribution functions at the centroid of each element have been initialized with the equilibrium values corresponding to the initialized flow field. The Reynolds number has been set to 3200. We have used $\Delta t = 3\tau/2$ with a forward Euler time-stepping for the temporal discretization. Fig. 11(c and d) shows normalized u and v velocity profiles along $(0.5, y)$

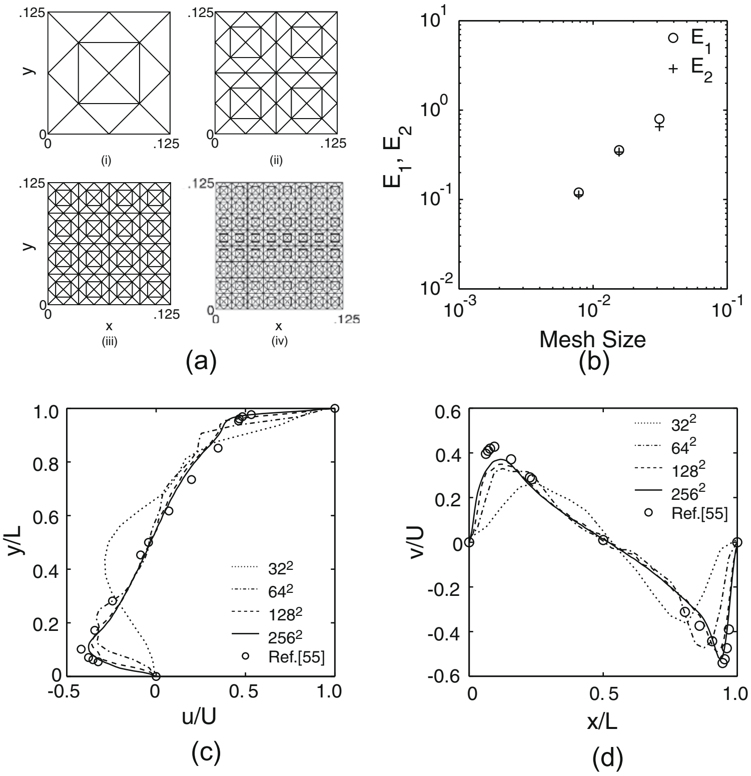


Fig. 11. Lid-driven cavity flow; Re = 3200; (a) Progressive grid-refinement (showing 1/8th of the computational domain); (i) 32^2,(ii) 64^2,(iii) 128^2, (iv) 256^2; (b) L_1 and L_2 norms of error against mesh size, (c) Velocity profiles along $(0.5, y)$ and (d) Velocity profiles along $(x, 0.5)$.

and $(x, 0.5)$ at Re = 3200, for successively refined grids, respectively. Also, the corresponding N–S solutions by Ghia et al. [55] using 128^2 grid points has been shown in Fig. 11(c and d). For the same number of triangles, the present scheme fails to achieve the expected accuracy corresponding to Ghia et al.'s [55] result. This may be because the gradients of the distribution functions were established with first-order procedure in Eq. (34). Also, the first-order extrapolation of the distribution functions at virtual upwind nodes limits the overall accuracy of the scheme. The L_1 and L_2 norms of the error in velocity for $32^2, 64^2$ and 128^2 number of uniform triangles with respect to the solution on 256^2 grid has been shown in Fig. 11(b). The linear-fall of error norms (E_1 and E_2) on log–log scale for cavity flow problem has also been previously obtained by Hou et al. [54]. The velocity profiles shown in Fig. 12(d) are by using 256^2 number of uniform triangles.

4.3.2. Validation for different Reynolds numbers

Shown in Fig. 12(a–c) are the comparison of velocity profiles at Re = 100, 400 and 1000, from the present simulations and those from Refs. [37,55]. The mesh system used consists of 56528 number of triangular elements. The Δt has been set to $\tau/6$. The use of Eq. (12) has been made for time integration with a superbee limiter to obtain the results shown in Fig. 12. As it may be seen, the present results are closer to the N–S results of Ghia et al. [55] than are the results of non-TVD FV–LBE [37], which seems to plagued by nonphysical numerical diffusion.

Table 1 lists the x- and y-coordinates of the centers of the primary eddy, right corner eddy and left corner eddy. It is evident that the present formulation captures the location of the eddies accurately for all the Reynolds numbers studied. Fig. 13 shows the streamline pattern for lid-driven flow at Re = 1000, 3200, from which it is observed that important flow features like the corner eddies have been resolved well. In Table 2, we have compared the horizontal (width) and vertical (height) extents of the corner eddies, as computed presently and from Ref. [55], respectively. At this stage, we have not investigated whether the comparison improves with further mesh refinement. The explicit time integration of advection term is mainly limited by CFL criterion. To improve on the total wall-clock time required to reach steady state, implicit treatment of the advection term is a must. An entry to the methodology of implicit upwind schemes may be obtained from Ref. [56].

4.4. Uniform flow over a 2D circular cylinder

Now, the results of the fourth flow problem we simulated in this work, namely the uniform flow over a 2D circular cylinder, are presented. Viscous flow in the wake of a circular cylinder has been visualized and fully analyzed experimentally

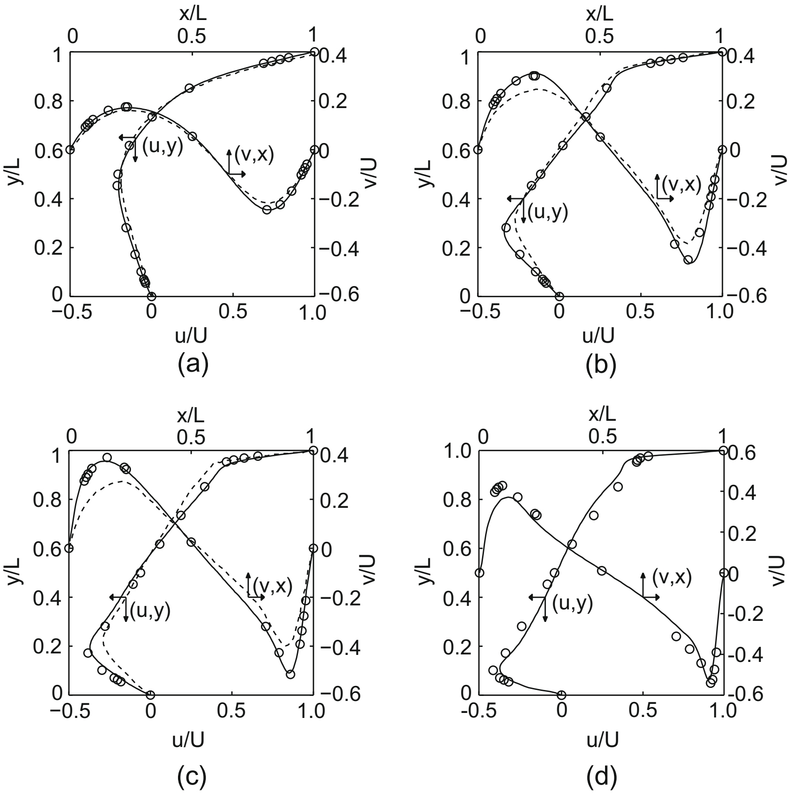


Fig. 12. Velocity profiles along $(0.5, y)$ and $(x, 0.5)$ in a lid-driven flow in a 2D cavity. (a) Re = 100, (b) Re = 400, (c) Re = 1000 and (d) Re = 3200. Comparison of present results (solid line) with results by Ghia et al. [55] (N–S, circles) and Ubertini and Succi [37] (non-TVD FV–LBE, dashes).

Table 1
Lid-driven flow inside a 2D cavity. Location of the centers of primary eddy and corner eddies with reference to the bottom left corner of the cavity. PE: primary eddy; BR: bottom right; BL: bottom left; TL: top left.

Re	Eddy	Ghia et al. [55]	Hou et al. [54]	Patil et al. [53]	Present
	PE	(0.6172, 0.7344)	(0.6196, 0.7373)	–	(0.6161, 0.7296)
100	BR	(0.9453, 0.0625)	(0.9451, 0.0627)	–	(0.9451, 0.0574)
	BL	(0.0313, 0.0391)	(0.0392, 0.0353)	–	(0.0345, 0.0324)
	PE	(0.5547, 0.6055)	(0.5608, 0.6078)	(0.5625, 0.6133)	(0.5506, 0.5972)
400	BR	(0.8906, 0.1250)	(0.8902, 0.1255)	(0.8906, 0.1289)	(0.8862, 0.1258)
	BL	(0.0508, 0.0469)	(0.0549, 0.0510)	(0.0507, 0.0507)	(0.0526, 0.0471)
	PE	(0.5313, 0.5625)	(0.5333, 0.5647)	(0.5391, 0.5703)	(0.5259, 0.5777)
1000	BR	(0.8594, 0.1094)	(0.8667, 0.1137)	(0.8750, 0.1250)	(0.8778, 0.1261)
	BL	(0.0859, 0.0781)	(0.0902, 0.1059)	(0.0937, 0.0859)	(0.0904, 0.0989)
	PE	(0.5165, 0.5469)	–	(0.5195, 0.5469)	(0.5189, 0.5441)
3200	BR	(0.8125, 0.0859)	–	(0.8320, 0.0898)	(0.8619, 0.0971)
	BL	(0.0859, 0.1094)	–	(0.0859, 0.1250)	(0.0993, 0.0963)
	TL	(0.0547, 0.8984)	–	–	(0.0316, 0.8689)

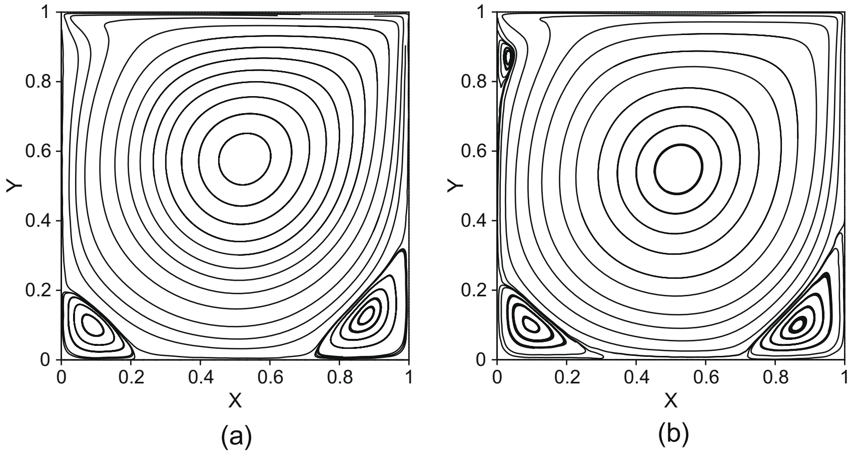
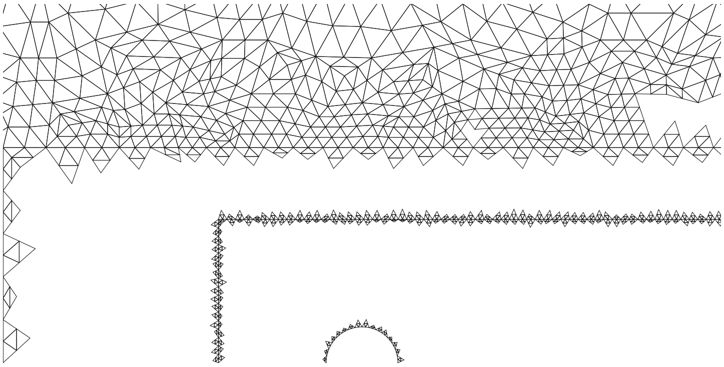


Fig. 13. Streamline pattern of the 2D lid-driven cavity flow; (a) Re = 1000 and (b) Re = 3200.

Table 2
Lid-driven flow inside a 2D cavity (Re = 1000, 3200). Size of the corner eddies.

Re	Reference	Bottom Left		Bottom Right		Top Left	
		Width	Height	Width	Height	Width	Height
1000	Ghia et al. [55]	0.2188	0.1680	0.3034	0.3536	–	–
	Present	0.2167	0.2005	0.2888	0.3431	–	–
3200	Ghia et al. [55]	0.2844	0.2305	0.3406	0.4102	0.0859	0.2057
	Present	0.3146	0.2443	0.3018	0.3925	0.0494	0.1819

[57,58]. In the present work, the steady state solutions are studied at three values of the Reynolds number, Re = 10, 20, 40, respectively. Here Re is defined with reference to the cylinder diameter and incident uniform velocity. Fig. 14 shows part of the mesh system employed for these simulations. The mesh consists of 89608 number of triangular elements. The time-integration has been performed using TVD–RK2 procedure with $\Delta t = \tau/10$. Superbee limiter has been utilized to simulate the flow for the set of Reynolds numbers considered. We used the bounce-back scheme to impose boundary condition on the cylinder surface. Table 3 compares the geometrical characteristics $(L/a, \theta_s)$ and dynamical parameters $(C_D, C_P(\pi)$ and $C_P(0))$ of the flow field at steady state, with previous experimental and computational results. We observe that the present results are within reasonable accuracy in all the cases studied. Fig. 15(a–c) shows the stream function plots at Re = 10, 20 and 40. The coefficient of pressure (C_p) is compared in Fig. 16. This demonstrates that a good accuracy in solution is achieved with the present formulation of the wall boundary conditions on the cell-centered mesh system. Further, we have performed a series of calculations to investigate the effects of limiter-function values $(\Phi(r))$ used in Eq. (28) onto the cylinder wake parameters. We chose to perform this test with the grid-system shown in Fig. 14 at Re = 40. Fig. 15(d) shows the extent of the wake length for four representative limiter-functions, (i) $\Phi(r) = 0$, denoted by L_1; (ii) $\Phi(r) = 1$, denoted by L_2; (iii)



$\Phi(r) =$ minmod, denoted by L_3 and (iv) $\Phi(r) =$ superbee, denoted by L_4. Compared in Table 4 are geometrical parameters $(L/a, \theta_s)$, dynamical parameter (C_D) and % error on L/a predicted with the experimental result of Ref. [58]. Negative value signifies under-prediction over Ref. [58] value. It is evident from this exercise that the results obtained by using superbee limiter are far more accurate than those obtained by using either of $\Phi(r) = 0, 1,$ minmod limiter-functions for the given mesh-system.

5. Conclusions

A finite volume formulation has been presented for simulation of the LBE on unstructured mesh. Specifically, the intercell advection of particle distribution function is calculated with a high-resolution, TVD scheme. The TVD FV–LBE results are well

[7] O. Filippova, D. Hänel, Grid refinement for lattice-BGK models, J. Comput. Phys. 147 (1998) 219–228.

[8] V. Sofonea, R.F. Sekerka, Viscosity of finite difference lattice Boltzmann models, J. Comput. Phys. 184 (2003) 422–434.

[9] N. Cao, S. Chen, S. Jin, D. Martinez, Physical symmetry and lattice symmetry in the lattice Boltzmann method, Phys. Rev. E 55 (1997) R21.

[10] T. Lee, C.-L. Lin, A characteristic Galerkin method for discrete Boltzmann equation, J. Comput. Phys. 171 (2001) 336–356.

[11] T. Lee, C.-L. Lin, An Eulerian description of the streaming process in the lattice Boltzmann equation, J. Comput. Phys. 185 (2003) 445–471.

[12] X. Shi, J. Lin, Z. Yu, Discontinuous Galerkin spectral element lattice Boltzmann method on triangular element, Int. J. Numer. Methods Fluids 42 (2003) 1249–1261.

[13] Y. Li, E.J. LeBoeuf, P.K. Basu, Least-squares finite-element scheme for the lattice Boltzmann method on an unstructured mesh, Phys. Rev. E 72 (2005) 046711.

[14] A. Bardow, I.V. Karlin, A.A. Gusev, General characteristic-based algorithm for off-lattice Boltzmann simulations, Europhys. Lett. 75 (3) (2006) 434–440.

[15] A. Düster, L. Demkowicz, E. Rank, High-order finite elements applied to the discrete Boltzmann equation, Int. J. Numer. Meth. Engng. 67 (2006) 1094–1121.

[16] F. Nannelli, S. Succi, The lattice Boltzmann equation on irregular lattices, J. Stat. Phys. 68 (1992) 401–407.

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

[18] H. Xi, G. Peng, S.-H. Chou, Finite volume lattice Boltzmann method, Phys. Rev. E 59 (5) (1999) 6202.

[19] S. Ubertini, S. Succi, G. Bella, Lattice Boltzmann schemes without coordinates, Phil. Trans. R. Soc. Lond. A 362 (2004) 1763–1771.

[20] V. Venkatakrishnan, Perspective on unstructured grid solvers, AIAA J. 34 (1996) 533–547.

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

[22] A. Harten, On a class of high resolution total-variation-stable finite-difference schemes, SIAM J. Numer. Anal. 21 (1) (1984) 1–23.

[23] P.L. Roe, Characteristic-based schemes for the Euler equations, Ann. Rev. Fluid Mech. 18 (1986) 337–365.

[24] A. Jameson, D. Mavriplis, Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh, AIAA J. 24 (1986) 611–618.

[25] T.J. Barth, D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA Paper No. 89–0366, 1989.

[26] N.T. Frink, Upwind scheme for solving the Euler equations on unstructured tetrahedral meshes, AIAA J. 30 (1992) 70–77.

[27] P. Tamamidis, A new upwind scheme on triangular meshes using the finite volume method, Comput. Methods Appl. Mech. Engrg. 124 (1995) 15–21.

[28] V. Sofonea, R.F. Sekerka, Boundary conditions for the upwind finite difference lattice Boltzmann model: evidence of slip velocity in micro-channel flow, J. Comput. Phys. 207 (2005) 639–659.

[29] T. Lee, C.-L. Lin, L.D. Chen, A lattice Boltzmann algorithm for calculation of the laminar jet diffusion flame, J. Comput. Phys. 215 (2006) 133–152.

[30] T.J. Barth, Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations, von Karman Institute Lecture Series 1994-05, 1994.

[31] Y.T. Chew, C. Shu, Y. Peng, On implementation of boundary conditions in the application of finite volume lattice Boltzmann method, J. Stat. Phys. 107 (2002) 539–556.

[32] F. Dubois, P. Lallemand, On lattice Boltzmann scheme, finite volumes and boundary conditions, Prog. Comput. Fluid Dyn. 8 (1–2) (2008) 11–24.

[33] B. Diskin, J.L. Thomas, E.J. Nielsen, H. Nishikawa, J.A. White, Comparison of node-centered and cell-centered unstructured finite-volume discretizations. Part I: viscous fluxes, AIAA Paper No. AIAA-2009-597, (2009).

[34] D.J. Mavriplis, Unstructured grid techniques, Annu. Rev. Fluid. Mech. 29 (1997) 473–514.

[35] X. He, L.-S. Luo, M. Dembo, Some progress in lattice Boltzmann method. Part. I. Nonuniform mesh grids, J. Comput. Phys. 129 (1996) 357–363.

[36] G. Peng, X. Haowen, C. Duncan, S.-H. Chou, Finite volume scheme for the lattice Boltzmann method on unstructured meshes, Phys. Rev. E 59 (4) (1999) 4675.

[37] S. Ubertini, S. Succi, Recent advances of lattice Boltzmann techniques on unstructured grids, Prog. Comput. Fluid Dyn. 5 (2005) 84–95.

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

[39] N. Rossi, S. Ubertini, G. Bella, S. Succi, Unstructured lattice Boltzmann method in three dimensions, Int. J. Num. Meth. Fluids 91 (2006) 619–633.

[40] X. He, L.-S. Luo, A priori derivation of the lattice Boltzmann equation, Phys. Rev. E 55 (6) (1997) R6333.

[41] S. Gottlieb, C.W. Shu, Total variation diminishing Runge–Kutta schemes, Math. Comput. 67 (1998) 73–85.

[42] S. Ubertini, G. Bella, S. Succi, Unstructured lattice Boltzmann equation with memory, Math. Comput. Simul. 72 (2006) 237–241.

[43] S. Ubertini, G. Bella, S. Succi, Unstructured lattice Boltzmann method: further developments, Phys. Rev. E 68 (2003) 016701.

[44] S.S. Chikatamarla, S. Ansumali, I.V. Karlin, Entropic lattice Boltzmann models for hydrodynamics in three dimensions, Phys. Rev. Lett. 97 (2006) 010201.

[45] T.J. Barth, O.J. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper No. AIAA-90-0013, (1990).

[46] P.L. Roe, Some contributions to the modeling of discontinuous flows, Lect. Appl. Math. 22 (1985) 163–193.

[47] P.K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, SIAM J. Numer. Anal. 21 (1984) 995–1011.

[48] B. Van Leer, Towards the ultimate conservative difference scheme, V.A second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.

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

[50] C.W.S. Bruner, R.W. Walters, Parallelization of the Euler equations on unstructured grids, AIAA Paper No. AIAA-97-1894, (1997).

[51] M.S. Darwish, F. Moukalled, TVD schemes for unstructured grids, Int. J. Heat Mass Trans. 46 (2003) 599–611.

[52] L.-x. Li, H.-s. Liao, L.-j. Qi, An improved *r*-factor algorithm for TVD schemes, Int. J. Heat Mass Trans. 51 (2008) 610–617.

[53] D.V. Patil, K.N. Lakshmisha, B. Rogg, Lattice Boltzmann simulation of lid-driven flow in deep cavities, Comput. Fluids. 35 (2006) 1116–1125.

[54] S. Hou, Q. Zou, S. Chen, G. Doolen, A.C. Cogley, Simulation of cavity flow by the lattice Boltzmann method, J. Comput. Phys. 118 (1995) 329–347.

[55] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.

[56] L. Fezoui, B. Stoufflet, A class of implicit upwind schemes for Euler simulations with unstructured meshes, J. Comput. Phys. 84 (1989) 174–206.

[57] A.S. Grove, F.H. Shair, E.E. Petersen, A. Acrivos, An experimental investigation of the steady separated flow past a circular cylinder, J. Fluid Mech. 19 (1964) 60–80.

[58] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow. Part 2. Unsteady flow, J. Fluid Mech. 79 (1977) 231–256.

[59] S.C.R. Dennis, G.Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, J. Fluid Mech. 42 (1970) 471–489.

[60] F. Nieuwstadt, H.B. Keller, Viscous flow past circular cylinders, Comput. Fluids 1 (1973) 59–71.

[61] X. He, G. Doolen, Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder, J. Comput. Phys. 134 (1997) 306–315.