Computers and Fluids 148 (2017) 157–165

Contents lists available at ScienceDirect

Computers and Fluids

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

A matrix-free, implicit finite volume lattice Boltzmann method for steady flows

Weidong Li [a,b,*]

[a] Department of Mathematics & Statistics, Old Dominion University, Norfolk, Virginia 23529, USA
[b] Beijing Computational Science Research Center, Hai-Dian District, Beijing 100193, China

ARTICLE INFO

Article history:
Received 23 August 2016
Revised 27 January 2017
Accepted 27 February 2017
Available online 28 February 2017

Keywords:
Matrix-free
Implicit scheme
Finite volume lattice Boltzmann method

ABSTRACT

In the present paper, a matrix-free, implicit finite volume lattice Boltzmann method for steady flow on unstructured mesh is proposed. The approximate linear system arising from the implicit finite volume discretization of lattice Boltzmann equation (LBE) is solved by a novel algorithm, which combines the lower-upper symmetric Gauss–Seidel (LU-SGS) and the Jacobi iteration schemes. A remarkable feature of the present implicit method is that the storage of the Jacobian matrixes of the convection and collision terms can be completely eliminated by approximating the Jacobian matrix-solution incremental vector product with appropriate numerical flux incremental vector and the numerical increment of collision term, resulting a matrix-free implicit scheme. The present method is validated by several two-dimensional testing cases.

1. Introduction

The mesoscopic kinetic based lattice Boltzmann method (LBM), as a new CFD way to solve NS equations, has received great attention from numerical simulation community [1,2]. Since it originates and inherits the essential character from the LGA, the standard LBM should satisfy the lattice-uniformity and work on uniform grids, which gives a discretization along characteristics of the LBE and offers exact results perfectly without any phase and amplitude error for the convection term of the LBE [3]. However, time and space are coupled in this method, which greatly hampers its practical engineering applications involving complex geometries. To overcome such drawback, the strong couple between time and space in the standard LBM should be given up [4]. Therefore, the finite-difference LBM (FD-LBM) [5], the structured grid based finite-volume LBM (FD-LBM) [6] and unstructured mesh based FV-LBM [7–13] were developing during the past two decades. However, these LBE schemes are either explicit or semi-implicit (only keeping the collision term implicit), in which the time step is constrained by the Courant-Friedrichs-Lewy (CFL) condition and/or the collision relaxation time, which will limits its applications involving high-Reynolds-number flows [14]. To release restrictions on the time step, several fully implicit LBE schemes exist, such as the implicit finite-difference LBM developed by Tölk et al. [14], Huang, Yang and Cai [15], the implicit Taylor-Galerkin finite-element LBM presented by Lee and Lin [3]. However, in these methods, the implicit Jacobian matrix has to be computed and stored, which may be a major impediment for three dimensional lattice Boltzmann models. Recently, Li and Luo [16] developed an implicit block LU-SGS FV-LBM, in which, not the whole, but the block-diagonal part of the implicit Jacobian matrix needs to be stored. Therefore, block LU-SGS FV-LBM reduces the storage requirement of the implicit schemes a lot. The block LU-SGS FV-LBM, however, is still not matrix-free. For some lattice Boltzmann models such as some high order lattice Boltzmann models with huge number of lattices velocities, the storage requirement of the block LU-SGS FV-LBM may still be an important issue. For such lattice Boltzmann models, even if the implicit block LU-SGS FV-LBM is used, the block-diagonal part of the implicit Jacobian matrix becomes so big that storing and manipulating it would cost a lot of memory and computer time. Moreover, compared with the Euler/Navier-Stokes equations, the LBE has more equations to be solve and more variables to be stored. Designing highly efficient and low-memory requirement implicit schemes is still one of the most significant topics in Euler/Navier–Stokes equation based CFD [17,18]. However, in the LBE based CFD, there are still lack of many very successful implicit schemes, and thus, there is still a long way to go on developing high efficient, low-memory requirement implicit schemes for LBE/DBE. In this work, we focus on the development of a matrix-free, fully implicit scheme based on the FV-LBM previously developed by present author for steady flows [13]. In the present

* Department of Mathematics & Statistics, Old Dominion University, Norfolk, Virginia 23529, USA
E-mail address: lwd_1982.4.8@163.com

http://dx.doi.org/10.1016/j.compfluid.2017.02.019
0045-7930/

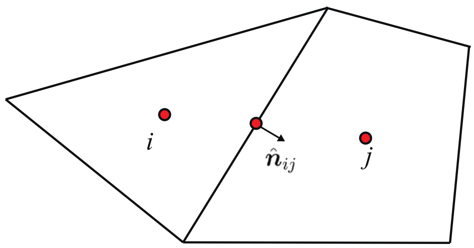

Fig. 1. Face *ij* and its left hand and right hand cells.

method, both of the convection and collision terms are implicit and linearized and the yielded implicit system is solved by an efficient hybrid of the LU-SGS [19] and the Jacobi iteration schemes without the storage of the Jacobian matrix.

The remainder of this paper is organized as follows. In Section 2, to be self-contained, the unstructured cell-centered FV-LBM formulation developed in [13] is introduced. In Section 3, the formulation of the present matrix-free implicit scheme is discussed in detail, and in Section 4, numerical examples are provided and discussed to validate the present implicit finite-volume lattice Boltzmann (FV-LB) scheme. Finally, in Section 5, conclusions of the present paper are given.

2. Numerical formulation of the FVLBM

2.1. Space discretization of LBE on unstructured meshes

The LBE with the collision term can be given as

$$\frac{\partial \mathbf{f}}{\partial t} + \nabla \cdot \boldsymbol{J} = \boldsymbol{\Omega}, \tag{1}$$

with

$$\mathbf{f} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{(N-1)} \end{pmatrix}, \boldsymbol{J} = \begin{pmatrix} \boldsymbol{\xi}_0 f_0 \\ \boldsymbol{\xi}_1 f_1 \\ \vdots \\ \boldsymbol{\xi}_{(N-1)} f_{(N-1)} \end{pmatrix}, \boldsymbol{\Omega} = \begin{pmatrix} \Omega_0 \\ \Omega_1 \\ \vdots \\ \Omega_{(N-1)} \end{pmatrix}. \tag{2}$$

where f_α, $\alpha = 0, 1, \ldots, N-1$ is the α-th distribution function corresponding to the microscopic velocity $\boldsymbol{\xi}_\alpha$. $\boldsymbol{\Omega}$ is the collision matrix and N is the total number of microscopic velocities, which depends on the specific lattice velocity model.

The macro density ρ and the momentum vector $\rho\mathbf{U}$ can be obtained by the following moments:

$$\begin{bmatrix} \rho \\ \rho\mathbf{U} \end{bmatrix} = \sum_{i=0}^{N-1} \begin{bmatrix} f_i \\ \boldsymbol{\xi}_i f_i \end{bmatrix}. \tag{3}$$

Integrating Eq. (1) on cell i, we have the integral form as following:

$$\frac{\partial}{\partial t} \int_{V_i} \mathbf{f}\, dV + \oint_{\partial V_i} \mathbf{H}\, dS = \int_{V_i} \boldsymbol{\Omega}\, dV, \tag{4}$$

where V_i and ∂V_i denote the cell i and its boundaries, respectively, and $\mathbf{H} := \boldsymbol{J} \cdot \hat{\boldsymbol{n}}$ is the convection flux and $\hat{\boldsymbol{n}}$ is the unit vector out normal to the surface element dS. Approximating the volume integration by a simple quadrature and the divergence by Guass theorem in Eq. (4), we have

$$\frac{\partial \mathbf{f}_i}{\partial t} V_i + \sum_{j \in N(i)} \mathbf{H}_{ij} S_{ij} = \boldsymbol{\Omega}_i V_i, \tag{5}$$

where the set $N(i) = \{j |$ cell j is the nearest neighbour of cell $i\}$ and $\hat{\boldsymbol{n}}_{ij}$ is the unit normal vector of the face ij shared by the left hand cell i and the right hand cell j shown in Fig. 1. Moreover, S_{ij} is the area of the face ij.

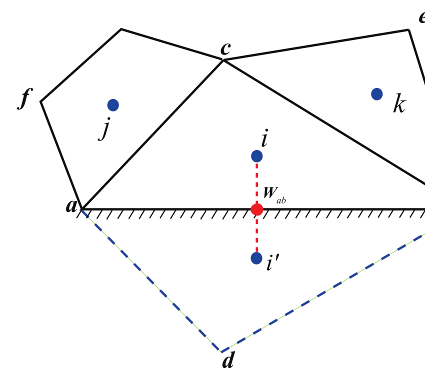

Fig. 2. Boundary cell *abc* and its ghost cell *abd*.

In Eq. (5), the fluxes $\mathbf{H}_{ij}$ are calculated by a low-diffusion Roe scheme [12], *i.e.*,

$$\mathbf{H}_{ij} = \frac{1}{2}[\mathbf{H}(\mathbf{f}_L) + \mathbf{H}(\mathbf{f}_R) - U_{ref} \max_{0 \le l \le (N-1)} (|\boldsymbol{\xi}_l \cdot \hat{\boldsymbol{n}}_{ij}|)(\mathbf{f}_R - \mathbf{f}_L)], \tag{6}$$

with a local characteristic velocity U_{ref} defined as

$$U_{ref} = \max(\min(k|\mathbf{u}_{ij} \cdot \hat{\boldsymbol{n}}_{ij}|, 1.0), \nu/\triangle x, 1.0e-05), \tag{7}$$

and $\mathbf{u}_{ij} = (\mathbf{u}_i + \mathbf{u}_j)/2$, where variables (i.e., $\mathbf{f}_L$, $\mathbf{f}_R$) on both sides of S_{ij} are reconstructed from cell i and cell j by a linear least-square method, which is second order accuracy. The details of reconstruction can be found in [20]. It should be pointed out that, to be more accurate, a quadratic least-square reconstruction method can also be used [21]. In Eq. (7), $k (k \ge 1)$ is an adjusting parameter. In this work, $k = 1$ is adopted. In addition, $\triangle x = |\boldsymbol{x}_{ci} - \boldsymbol{x}_{cj}|/2$, where $\boldsymbol{x}_{ci}$ and $\boldsymbol{x}_{cj}$ are the center coordinate vectors of cell i and cell j, respectively.

2.2. Boundary conditions of the FVLBM

In the present work, the ghost cell method [20] is used for dealing with boundary conditions. In this method, the information of macro-variables in a ghost cell should be reconstructed by the corresponding boundary conditions.

In Fig. 2, there are cell *abc* and its ghost cell *abd* on the boundary *ab*. The centroid of cell *abc* and cell *abd* are denoted by i and i', respectively, and the face center of the boundary *ab* is labeled by W_{ab}.

At t_n time, from macro boundary conditions, the macro flow variables on the boundary *ab* can be obtained, but distribution functions at W_{ab} are still unknown. To obtain them, the non-equilibrium exploration scheme [22] is extended to the present unstructured mesh. From this scheme, the non-equilibrium part of distribution function at i is taken as a good approximation of the counterpart at W_{ab}. Therefore, the distribution functions at W_{ab} can be reconstructed as:

$$\mathbf{f}_{W_{ab}} \approx (\mathbf{g}_{W_{ab}}) + (\mathbf{f}_i)^{neq}, \tag{8}$$

where $\mathbf{g}_{W_{ab}}$ and $(\mathbf{f}_i)^{neq}$ denote the equilibrium distribution function vector on the boundary *ab* and non-equilibrium distribution function vector at the cell center of cell i, respectively. Further, the distribution functions at i' can be determined by the central difference scheme:

$$\mathbf{f}_{i'} = 2.0\mathbf{f}_{W_{ab}} - \mathbf{f}_i. \tag{9}$$

3. Matrix-free implicit time-marching scheme of the FVLBM

One of the difficulties on developing an implicit scheme for the LBE is the nonlinear feature of the collision term $\boldsymbol{\Omega}$ in the LBE. The