Journal of Computational Physics 301 (2015) 102–110

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# A near-boundary modification for the link bounce-back boundary condition in the lattice Boltzmann method

Nathan M. Olson

*Indian Institute of Technology Gandhinagar, VGEC Complex, Chandkheda, Ahmedabad 382424, India*

ARTICLE INFO

*Article history:*
Received 25 December 2014
Received in revised form 22 June 2015
Accepted 15 August 2015
Available online 21 August 2015

*Keywords:*
Lattice Boltzmann method
Boundary conditions
Bounce-back

ABSTRACT

The bounce-back boundary condition in the lattice Boltzmann method distorts curved or inclined boundaries by forcing them to conform to a rectangular grid. This paper proposes a modification that reduces the effect of this discretization on the fluid flow. The modification takes the form of the addition of a type of node that is neither solid nor fluid, called the "sticky node". Sticky nodes are used in all cells that contain both fluid and solid. They are treated like fluid nodes with modified viscosity, body force, and velocity calculation. The method is applied to the LBGK model on a D2Q9 grid, and the accuracy of the method is evaluated using several test cases. Decreased discretization artifacts and decreased sensitivity to grid alignment are demonstrated, compared to the standard link bounce-back boundary condition. The method is computationally efficient, local, and demonstrates good numerical stability.



## 1. Introduction

The lattice Boltzmann method (LBM) has found many applications in modeling complex viscous flows such as multiphase flows [1,2], chemically reacting flows [3], and flows through porous media [4,5]. The LBM was originally derived from lattice gas cellular automata [6,7], but can also be considered as a discretization of the Boltzmann equation of statistical mechanics [8,9]. The LBM can also be shown to be equivalent to the incompressible Navier–Stokes equation in its incompressible limit [10,11]. Instead of working directly with continuum variables such as pressure and velocity, the lattice Boltzmann equation governs the behavior of distribution functions of particles moving in discrete steps along a grid. The LBM has several advantages over conventional fluid dynamics solvers, including simplicity of programming, accessibility of microscale dynamics, and ease of parallelism. Although many LBM models have been used, the lattice BGK (LBGK) model [10] is still widely used because of its simplicity and robustness [12–15].

The use of the LBM to solve fluid dynamics problems with large solid-fluid interfacial areas makes the treatment of boundaries especially important in this method. Retained from early work in lattice gas simulations, the classical bounce-back boundary condition [16] has often been used to model the presence of a solid wall. However, it was shown to satisfy the no-slip condition at the wall with only first-order accuracy in space discretization [17,18]. Second order accuracy using a link-based bounce-back scheme was obtained by placing the boundary halfway between the fluid and bounce-back nodes [19,20]. Alternate boundary conditions were proposed that performed collision at the boundary nodes [21], added a row of boundary nodes between the fluid and wall nodes [22], or used the assumption that the unknown distribution function is at equilibrium [23]. Skordos presented a method that uses the gradient of the fluid velocity to calculate the unknown

*E-mail address:* nathanmolson@gmail.com.

http://dx.doi.org/10.1016/j.jcp.2015.08.021
0021-9991/

distributions at boundary nodes [24]. In the regularized boundary condition, all particle distributions at the boundary are replaced in a way that is consistent with the hydrodynamic limit of the LBM [25]. These methods satisfy the no-slip condition with second-order accuracy at solid walls, but are all applicable only to boundaries that are aligned with the computational grid [26,27]. The use of these boundary conditions with non-aligned boundaries leads to nonzero slip velocities and discretization effects on the fluid flow.

More recently, boundary conditions have been developed that provide an accurate treatment of curved and non-aligned boundaries. Noble and Torczynski allowed for non-aligned boundaries by applying a modified collision operator in partially saturated cells [28]. Verberg and Ladd presented a sub-grid boundary condition that applies the bounce-back condition in a continuous sense based on the volume fraction of fluid within a cell [29,30]. Their method has the advantage of requiring only the fluid volume fraction and not the surface normal or position. Filippova and Hänel developed a boundary condition for curved surfaces based on inter-extrapolation of the distribution functions [31] that was later improved by other researchers [32,33]. All of these methods were subject to mass leakage at the boundaries, which was reported and addressed by Bao et al. [34]. Bouzidi et al. proposed a method that combines the bounce-back concept with quadratic interpolation of the distribution functions [35]. [33] Yin and Zhang modified the inter-extrapolation idea by applying it to the prescribed velocity at the midpoint between a boundary node and a fluid node, instead of to the boundary distribution functions [36]. Although these methods have offered improved accuracy over bounce-back, bounce-back is still widely used due to its simplicity and computational efficiency.

In this paper, we propose a new boundary treatment called the sticky node (SN) method to deal with solid boundaries that do not conform to the lattice grid. In this method, so-called “sticky nodes” are placed in grid cells that contain both solid and fluid. These nodes obey the same update equations as the fluid nodes but with a modified viscosity and body force that account for the presence of the solid in the cell. The fluid velocity at the center of the cell can be calculated from the distribution function for majority-fluid sticky nodes. The method was applied to the BGK lattice Boltzmann model on a D2Q9 grid. Simulations were performed for gravity-driven Poiseuille flow in aligned and inclined channels and for pressure-driven flow over an array of circular cylinders. The results compare favorably with the link bounce-back (LBB) method from which the SN method is derived, and in some cases with more complex methods. In all cases the SN method greatly reduces the grid artifacts that LBB causes by forcing the boundary to align to the grid. The SN method is easy to program and computationally inexpensive. This makes it an attractive scheme for lattice Boltzmann simulations, especially those that already use the LBB method.

## 2. Lattice Boltzmann method

The Boltzmann equation of statistical mechanics can be discretized to give the lattice Boltzmann equation [7]:

$$f_i(\vec{x}+\vec{c}_i t, t+\Delta t) - f_i(\vec{x}, t) = \Omega_i(f) \tag{1}$$

where $f_i(\vec{x}, t)$ is the particle distribution moving at the $i$th lattice velocity $\vec{c}_i$ at position $\vec{x}$ and time $t$, $\Delta t$ is the time step, and $\Omega_i$ is the collision operator. If the popular LBGK collision operator is used equation (1) becomes [37]

$$f_i(\vec{x}+\vec{c}_i t, t+\Delta t) - f_i(\vec{x}, t) = \omega[f_i^{eq}(\vec{x}, t) - f_i(\vec{x}, t)] \tag{2}$$

Here $\omega$ is a relaxation parameter, the inverse of the relaxation time $\tau$ commonly seen in the literature. The equilibrium distribution function is given by

$$f_i^{eq} = \frac{\rho w_i}{c_s^2}[c_s^2 + \vec{u}\cdot\vec{c}_i + \frac{1}{2}(\vec{u}\cdot\vec{c}_i)^2 - \frac{1}{2}u^2] \tag{3}$$

where $\rho$ is the fluid density and $\vec{u}$ is the fluid velocity. The fluid density and velocity are related to the particle distribution functions by $\rho = \sum_i f_i$ and

$$\rho\vec{u} = \sum_i f_i \vec{c}_i \tag{4}$$

The Navier–Stokes equations can be recovered from equation (2) using a Chapman–Enskog expansion [7]. In this article, the LBGK equation is applied to the two dimensional D2Q9 grid with lattice velocities

$$\vec{c}_i = \begin{cases} [0,0] & : \quad i=0 \\ [\cos(i-1)\pi/2, \sin(i-1)\pi/2]\frac{\Delta x}{\Delta t} & : \quad i=1,2,3,4 \\ [\cos(2i-9)\pi/4, \sin(2i-9)\pi/4]\frac{\Delta x}{\Delta t} & : \quad i=5,6,7,8 \end{cases} \tag{5}$$

and lattice weights

$$w_i = \begin{cases} 4/9 & : \quad i=0 \\ 1/9 & : \quad i=1,2,3,4 \\ 1/36 & : \quad i=5,6,7,8 \end{cases} \tag{6}$$