Chapter 6
Ghost Fluid Lattice Boltzmann Methods for Complex Geometries

Arpit Tiwari, Daniel D. Marsh, and Surya P. Vanka

6.1 Lattice Boltzmann Method

Lattice Boltzmann method (LBM) (Chen and Doolen 1998; Luo 2000) has emerged as a powerful alternate computational tool for simulating microscopic and macroscopic flows in complex configurations. In conventional computational fluid dynamics (CFD) methods, the Navier–Stokes equations describing the continuum behavior of fluid flows are solved numerically. The equations describing the conservation of mass, momentum and energy are solved to determine the macroscopic variables (velocity, pressure and temperature). On the other hand, LBM is a meso-scale method which solves reduced versions of the microscopic Boltzmann kinetic equations for particle distribution functions. Simplified kinetic models are developed that retain only specific details of the molecular motion sufficient to recover macroscopic hydrodynamic behavior. LBM is, therefore, an intermediate approach between the continuum and the more fundamental approach of molecular dynamics (MD) simulations.

LBM evolved from lattice gas automata (LGA), in which a simplified kinetic model is constructed for simulating fictitious particles in discrete lattice space and time. The LGA model proposed by Frisch et al. (1986) consists of a two-dimensional equilateral triangular lattice space with hexagonal symmetry. Particles point toward the nearest lattice site; the kinetic model consists of collision and streaming based on certain rules. LGA is based on Boolean operation, thus suffers from statistical noise. This problem was cured by replacing Boolean particle distribution variables with ensemble-averaged particle distribution functions (McNamara and Zanetti 1988), which formed the basis of LBM. However, the primitive formulations of LBM were computationally inefficient because of the complexity of the collision operator in

A. Tiwari (✉) · D. D. Marsh
Gamma Technologies LLC, Westmont, IL 60559, USA
e-mail: arpit01@gmail.com

S. P. Vanka
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

© Springer Nature Singapore Pte Ltd. 2020

S. Roy et al. (eds.), *Immersed Boundary Method*, Computational Methods in Engineering & the Sciences, https://doi.org/10.1007/978-981-15-3940-4_6

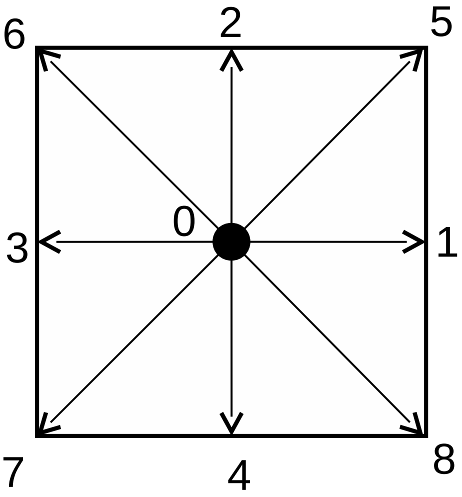
the ensemble form. The major breakthrough in efficiency was achieved through the linearization of the collision operator (Higuera and Jiménez 1989) assuming local equilibrium. This was further simplified using the Bhatnagar–Gross–Krook (BGK) approximation (Bhatnagar et al. 1954) of single relaxation time toward equilibrium leading to the lattice Bhatnagar–Gross–Krook (LBGK) model; the local equilibrium functions are chosen such that macroscopic equations are recovered (Qian et al. 1992; Chen et al. 1992).

Since LBM is an intermediate approach between the macroscopic and microscopic methods, the Navier–Stokes equations can be obtained by carrying out multi-scale expansion of the LB equations (Chen and Doolen 1998; Luo 2000). Similarly, LB equations can be derived from the continuum Boltzmann BGK equations, in which the equilibrium distribution is described by the Boltzmann–Maxwellian function. Low Mach number reduction of Boltzmann BGK equations leads to LB equations (Chen and Doolen 1998; Luo 2000). There is an extensive literature on analysis and advancement of LBM for various applications.

6.1.1 Basic Formulation of LBM

In LBM, particle distribution functions are advanced in time via two processes: collision and streaming. Various advanced formulations have been developed over the years for these processes (e.g., multi-relaxation-time LBM and entropic LBM). However, since the focus of this book in on boundary conditions, for conciseness, we present a widely used basic formulation of LBM here—the single relaxation-time D2Q9 model. As the name suggests, it is a two-dimensional model, in which the collision process employs a single relaxation-time parameter, and particles are restricted to move along nine velocity vectors during streaming, as shown in Fig. 6.1

Fig. 6.1 D2Q9 model of LBM



(see Sect. 6.4.3 for a three-dimensional LBM implementation). The discrete equation governing these processes is

$$f_i(\mathbf{x} + c\,\mathbf{v}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = -\Omega_i, \quad i = 0, 1, \ldots, 8, \tag{6.1}$$

in which, the left- and right-hand sides represent streaming and collision processes, respectively. f_i is the discrete particle distribution function, $\mathbf{x}$ is the spatial location vector, $\mathbf{v}_i$ is the particle velocity, t is time and Δt is the time step. $c = \Delta x / \Delta t$ is the lattice speed, where Δx is the lattice spacing. Ω_i denotes discrete collision operation. The nine velocities are given by

$$\mathbf{v}_i = \begin{cases} (0, 0) & \text{for } i = 0, \\ (\cos((i-1)\pi/2),\ \sin((i-1)\pi/2)) & \text{for } i = 1 \text{ to } 4, \\ \sqrt{2}(\cos((i-5)\pi/2 + \pi/4),\ \sin((i-5)\pi/2 + \pi/4)) & \text{for } i = 5 \text{ to } 8. \end{cases} \tag{6.2}$$

The collision term Ω_i can take various forms provided the conservation laws are obeyed; the linearized collision function based on BGK approximation takes the form

$$\Omega_i = \frac{f_i - f_i^{\text{eq}}}{\tau}, \tag{6.3}$$

where τ is a relaxation-time parameter, and f_i^{eq} is the equilibrium particle distribution function:

$$f_i^{\text{eq}} = w_i \rho \left(1 + \frac{c\,\mathbf{v}_i \cdot \mathbf{u}}{c_s^2} + \frac{(c\,\mathbf{v}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right), \tag{6.4}$$

where $c_s = c/\sqrt{3}$ is the lattice speed of sound and w_i is the weighing function:

$$w_i = \begin{cases} 4/9 & \text{for } i = 0, \\ 1/9 & \text{for } i = 1 \text{ to } 4, \\ 1/36 & \text{for } i = 5 \text{ to } 8. \end{cases} \tag{6.5}$$

ρ and $\mathbf{u}$ are the density and flow velocity, respectively, obtained from the particle distribution functions and velocities using

$$\rho = \sum f_i \text{ and } \rho\,\mathbf{u} = \sum c\, f_i\, \mathbf{v}_i. \tag{6.6}$$

Applying a Chapman–Enskog procedure on the LB equations, macroscopic continuity and momentum equations can be derived in the low Mach number limit, with pressure (p) and kinematic viscosity (ν) given by $p = \rho\, c_s^2$ and $\nu = (\tau - 1/2)\, c_s^2\, \Delta t$.

6.1.2 Advantages of LBM

LBM offers a number of advantages compared to the conventional CFD methods. The simplified kinetic model utilized in LBM has a linear streaming operator, while Navier–Stokes equations have a nonlinear convection term. The nonlinearity associated with the collision operator in LBM is localized, which makes it highly suited for parallelization (see Sect. 6.4.4 for a discussion on parallelization using GPUs). In LBM, pressure variations are implicitly expressed as a function of density variations, thus eliminating the well-known pressure–velocity coupling issue that needs special treatment in conventional incompressible CFD solvers. Furthermore, since LB equations are obtained by reducing Boltzmann equations, (1) micro-scale multiphase physics are easier to incorporate in LBM and (2) coupling with molecular dynamics (MD) simulations is straightforward (discussed in Sect. 6.4.5). It is thus widely used in the analysis of complex fluids and multi-phase flows. Conventional solvers need special care in dealing with multi-phase flows (Shukla et al. 2010; Tiwari et al. 2013). LBM also offers advantages compared to molecular simulations. In LBM, the simplified kinetic model utilizes a small set of velocities in the phase space. This makes computations significantly faster compared to solving the Boltzmann equations of molecular motion derived from the kinetic theory utilizing the Boltzmann–Maxwellian equilibrium distribution function, where the phase space is continuous and infinite.

6.2 Boundary Conditions in LBM

Accurate implementation of boundary conditions is challenging in LBM due to the difficulty in obtaining particle distribution functions from the prescribed hydrodynamic conditions at the boundaries. This is because the number of unknown particle distribution functions is typically more than the number of hydrodynamic boundary conditions. For stationary walls, the basic implementation of the well-known bounce-back condition simply inverts the particle velocities at the wall. However, it is only first-order accurate, and not applicable to moving walls. Several advancements have been proposed to improve its applicability and accuracy (Ziegler 1993; Ladd 1994; Filippova and Hänel 1998; Mei et al. 1999; Ginzburg and d'Humieres 2003; Lallemand and Luo 2003; Yu et al. 2003).

In the so-called hydrodynamic boundary implementation (Noble et al. 1995), the unknown (incoming) particle distribution functions are obtained from the prescribed hydrodynamic conditions at the boundaries using Eq. (6.6). The original implementation was only limited to those LBM models in which the number of unknown distribution functions is equal to the number of hydrodynamic boundary conditions. This limitation was addressed later by proposing additional rules to handle LBM models in which the number of missing functions is more than the prescribed boundary conditions (Maier et al. 1996; Zou and He 1997). Chen et al. (1996) proposed an extrapolation approach, in which the incoming distribution functions are extrapolated from the interior distribution functions, and the equilibrium distribution functions at the boundaries are computed from the prescribed hydrodynamic boundary conditions using Eq. (6.4). Guo et al. (2002) developed an extension of this method by splitting the incoming distribution functions into equilibrium and non-equilibrium parts. Equilibrium parts are computed using Eq. (6.4), while the non-equilibrium parts are extrapolated from the interior functions.

All the boundary approaches mentioned above need special care when dealing with complex geometries. For conventional CFD methods, immersed boundary methods (IBM) are now widely used to deal with curved boundaries. Since its first introduction by Peskin (1972), there has been an extensive amount of research on IBM. Among the approaches developed are forcing via deformation of elastic region tracked by Lagrangian points (Peskin 1972; Goldstein et al. 1993; Lai and Peskin 2000; Lee and LeVeque 2003), forcing via Lagrange multipliers (Glowinski et al. 1999; Taira and Colonius 2007), direct forcing by modifying discrete momentum equations (Mohd-Yusof 1997; Fadlun et al. 2000; Balaras 2004; Gilmanov and Sotiropoulos 2005; Uhlmann 2005), direct boundary implementation using Cartesian grid method (Ye et al. 1999) and direct boundary implementation using ghost cells (Majumdar et al. 2001; Tseng and Ferziger 2003). In the ghost-cell technique, hypothetical (ghost) cells are placed outside the fluid domain such that each cell has at least one neighbor inside the domain. Various options have been developed to extrapolate values to these cells to enforce boundary conditions. One widely used implementation obtains values via locating image points inside the fluid domain along the boundary normal (Majumdar et al. 2001).

IBM has been implemented into LBM as well. The original IB formulation of forcing via deformation of elastic region was coupled with LBM by Feng and Michaelides (2004). The same authors later employed direct-forcing IB formulation in LBM (Feng and Michaelides 2005). An alternative way of calculating the forcing term was developed by Niu et al. (2006) via the momentum exchange method of Ladd (1994). Several researchers have extended/improved forcing function-based IBM for various applications (Peng et al. 2006; Zhang et al. 2007; Dupuis et al. 2008; Tian et al. 2011; Kang and Hassan 2011). A drawback of this approach is that the no-slip condition is not strictly enforced at the walls. Wu and Shu (2009) proposed a velocity correction method to solve this issue. Another way of implementing strict boundary conditions is using the ghost cells-based IB method. Tiwari and Vanka (2012) first coupled this method with LBM and demonstrated its efficiency, accuracy, generality and ease of implementation. This approach and the subsequent research works toward its applications and improvements are discussed in the following sections.

6.3 Ghost Fluid LBM

Here, we describe a ghost fluid immersed boundary lattice Boltzmann method (GF-IB-LBM) developed by Tiwari and Vanka (2012), which imposes hydrodynamic boundary conditions via ghost nodes, rather than using the forcing concept. A general approach using extrapolation along boundary normal is employed to obtain hydrodynamic values at the ghost nodes, which are then used to obtain equilibrium particle distribution functions. The non-equilibrium particle distribution functions are simply extrapolated from the fluid domain. The two contributions are then added to obtain particle distribution functions at the ghost nodes.

6.3.1 Algorithm and Implementation

For conciseness, we restrict our focus to wall (stationary as well as moving) boundary conditions in curved geometries. (The method detailed below can be extended to other types of boundary conditions in a straightforward fashion). The implementation in two dimensions (see Sect. 6.4.3 for three-dimensional extension) briefly involves the following steps.

1. Ghost node and corresponding image point identification: Before streaming operation, ghost nodes adjacent to a boundary are identified such that each ghost node has at least one neighboring node in the fluid domain. For each ghost node, an image point is located inside the fluid domain along the boundary normal. This is shown in Fig. 6.2.
2. Density and velocity determination at image points: A special bilinear interpolation procedure is developed by Tiwari and Vanka (2012) to obtain hydrodynamic values at the image points. A four-point interpolation is used when all the surrounding nodes are interior (Fig. 6.2a). If a surrounding node is not interior (Fig. 6.2b), then it is replaced by the point of intersection of the normal from that node with the boundary curve. For velocity, the values at the wall intersection points are simply the prescribed boundary values. These points are used along with the interior nodes to obtain velocities at image points. This approach to obtain bilinear coefficients can be expressed using this general formula:

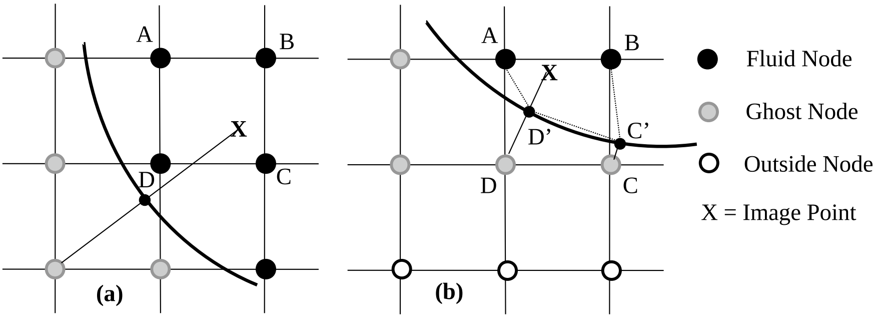


Fig. 6.2 Interpolation at image points: **a** all neighboring nodes inside, and **b** two inside and two outside nodes (Tiwari and Vanka 2012)

$$\begin{array}{ll} a\,x_j + b\,y_j + c\,x_j\,y_j + d = \mathbf{u}_j & \text{if node } j \text{ is inside,} \\ a\,x'_j + b\,y'_j + c\,x'_j\,y'_j + d = \mathbf{u}'_j & \text{otherwise,} \end{array}$$

where $j = 1{:}4$ denote the four surrounding nodes, a to d are the bilinear coefficients, $(x_j,\ y_j)$ and $\mathbf{u}_j$ are the spatial Cartesian coordinates and velocity, respectively, at node j and $(x'_j,\ y'_j)$ and $\mathbf{u'}_j$ are the spatial Cartesian coordinates and velocity, respectively, at the wall intersection point of node j. For density, the interpolation formula is modified such that it utilizes the zero normal gradient condition at the wall intersection points. This can also be expressed using this general formula:

$$\begin{array}{ll} a\,x_j + b\,y_j + c\,x_j\,y_j + d = \rho_j & \text{if node } j \text{ is inside,} \\ a\,n_{xj} + b\,n_{yj} + c\,(x_j\,n_{yj} + y_j\,n_{xj}) = 0 & \text{otherwise,} \end{array}$$

where $(n_{xj},\ n_{yj})$ denotes the boundary normal from node j, and ρ_j is the density at node j.

3. Density and velocity determination at ghost nodes: Velocity and density values at the image points are extrapolated to the ghost points along the normal direction such that the prescribed velocity and zero density gradient conditions are satisfied at the wall.
4. Particle distribution function determination at ghost nodes: The equilibrium part (f_i^{eq}) is obtained from density and velocity values at the ghost nodes using Eq. (6.4). The non-equilibrium part ($f_i^{\mathrm{neq}} = f_i - f_i^{\mathrm{eq}}$) is obtained analogous to density computation using the aforementioned special interpolation procedure. The two contributions are then summed to obtain particle distribution functions at the ghost nodes, which are then streamed inside the fluid domain during streaming operation. Note that second-order accuracy of the equilibrium part is ensured by the second-order accurate bilinear interpolation of density and velocity. However, the simple extrapolation of non-equilibrium part is only first-order accurate. Overall, second-order accuracy is attained because the non-equilibrium part corresponds to the first-order term in the asymptotic expansion of the particle distribution functions (Tiwari and Vanka 2012; demonstrated in Sects. 6.4.1 and 6.4.2).

6.3.2 Advantages

The overall approach is simple and efficient and preserves second-order accuracy for curved boundaries. A major advantage is its generality—applicable to inflow/outflow, moving wall, symmetric and periodic boundary conditions (including Dirichlet as well as Neumann conditions; Tiwari and Vanka 2012). An illustration is presented in Sect. 6.4.2. Furthermore, the method by design imposes hydrodynamics conditions strictly at the boundaries. Boundary enforcement is local, hence preserves high parallelism of LBM (discussed in Sect. 6.4.4). It also enables straightforward coupling with molecular dynamics simulations via ghost nodes, as demonstrated in Sect. 6.4.5.

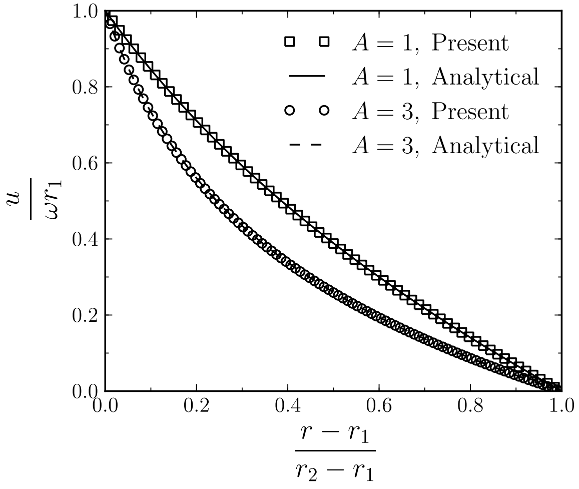
6.4 Application of GF-IB-LBM

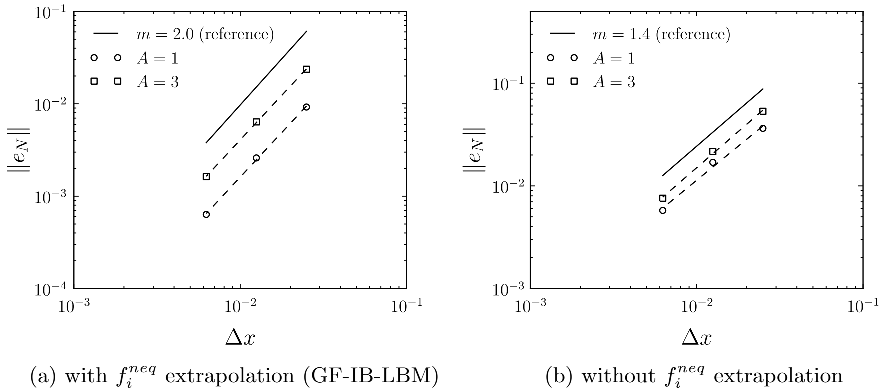
We first demonstrate the accuracy of GF-IB-LBM on four test problems involving curved and moving boundaries. Then, we discuss its coupling with molecular dynamics in a GPU-parallelized framework. For conciseness, only key results are presented here; we refer to Tiwari and Vanka (2012), Tiwari et al. (2009) and Marsh (2010) for more details.

6.4.1 One-Dimensional Problem

We first consider cylindrical Couette flow problem to (1) compare results with analytical solution and (2) demonstrate the importance of extrapolation of non-equilibrium distribution function in achieving second-order accuracy. The flow is assumed laminar, which makes this problem inherently one dimensional, which we solve using a two-dimensional lattice for demonstration (Tiwari and Vanka 2012). We consider an inner cylinder of radius r_1 rotating with an angular velocity ω and a stationary outer cylinder of radius r_2. Angular velocity is chosen such that Reynolds number based on the inner cylinder's diameter and tangential speed is 50. Figure 6.3 shows good agreement of velocity (u) variation along the radial (r) direction with the analytical solution. In Fig. 6.4a, the L_2 error norm ($\|e\|_N$) of velocity along the radial direction is plotted against grid spacing to demonstrate second-order accuracy of the method. We also plot in Fig. 6.4b, the results obtained without extrapolating the non-equilibrium part (f_i^{neq}), which shows a slope $\approx$ 1.4. This demonstrates the importance of extrapolation of the non-equilibrium part.

Fig. 6.3 Comparison of radial velocity profile for cylindrical Couette flow using a 321 × 321 grid in a $2.5r_2 \times 2.5r_2$ square domain with analytical solution for two aspect ratios ($A = (r_2 - r_1)/r_1$) (Tiwari and Vanka 2012)





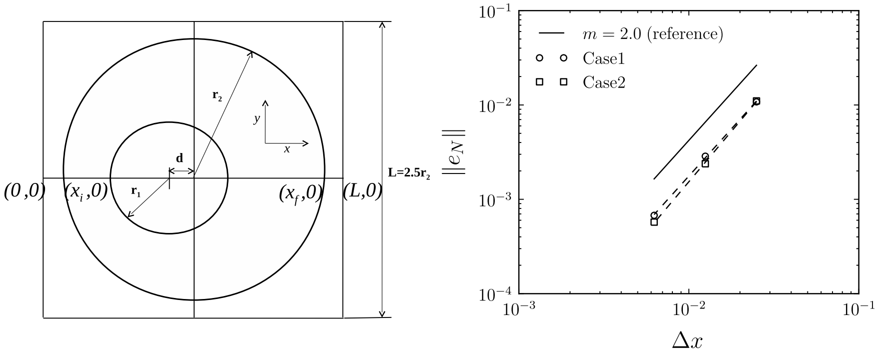
(a) with f_i^{neq} extrapolation (GF-IB-LBM)

(b) without f_i^{neq} extrapolation

Fig. 6.4 Variation of L_2 error norm ($\|e\|_N$) of radial velocity with grid spacing (Δx) for cylindrical Couette flow for two aspect ratios ($A = (r_2 - r_1)/r_1$); m denotes slope (Tiwari and Vanka 2012)

6.4.2 Two-Dimensional Problems

We next consider flow between two rotating eccentric cylinders (Fig. 6.5a). Due to a misalignment in their rotation axes, the flow between them is two dimensional. This configuration is included here to show second-order accuracy of GF-IB-LBM for a two-dimensional problem. Tiwari and Vanka (2012) considered four different combinations of eccentricity, radius ratio and rotational speeds. For conciseness, we present here grid convergence of two cases: (1) inner cylinder rotating and (2) outer cylinder rotating. Figure 6.5b demonstrates second-order accuracy of the method.

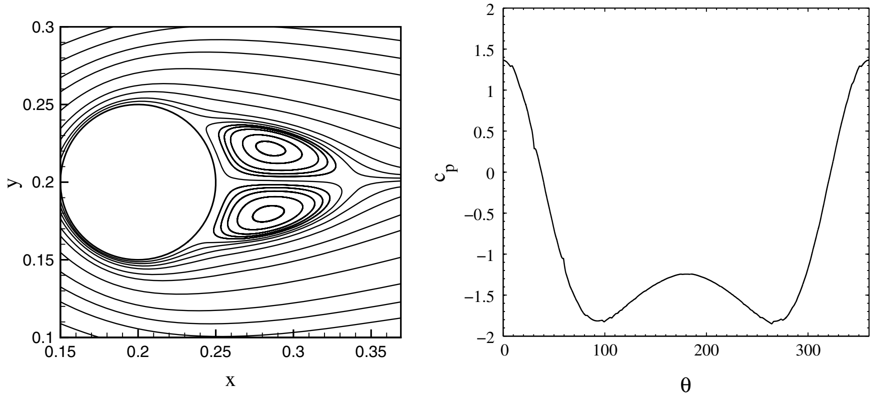


(a) schematic

(b) error norm

Fig. 6.5 Schematic of the domain considered and variation of L_2 error norm ($\|e\|_N$) of radial velocity with grid spacing (Δx) for two cases of flow between rotating eccentric cylinders: case 1 has inner cylinder rotating and case 2 has outer cylinder rotating; m denotes slope (Tiwari and Vanka 2012)

We next consider flow over a cylinder in a channel. The previous two problems had only wall boundaries, therefore, this problem is chosen to demonstrate generality of the method for other types of boundary conditions. In this problem, parabolic velocity boundary condition is applied at the inlet, and constant pressure as well as fully developed conditions are separately considered at the outlet (Tiwari and Vanka 2012). This is a widely used verification problem; extensive benchmarking data exists (Schäfer et al. 1996). Reynolds number based on average inlet velocity and cylinder diameter is 20 for the simulated configuration. We consider three uniformly spaced lattices with $n = 16$, 32 and 64, where n denotes the number of nodes across the diameter. Figure 6.6a shows streamlines in the recirculation zone, and Fig. 6.6b shows variation of pressure coefficient c_p with cylinder angle θ using GF-IB-LBM ($n = 64$) with constant pressure boundary condition at the outlet. The drag and lift coefficients compare well with those reported by the high-resolution study of Schäfer et al. (1996) in Table 6.1.



(a) streamlines (b) pressure coefficient

Fig. 6.6 Streamlines in the recirculation zone and variation of pressure coefficient with cylinder angle for flow over a cylinder using GF-IB-LBM with $n = 64$ (Tiwari and Vanka 2012)

Table 6.1 Comparison of drag and lift coefficients (Tiwari and Vanka 2012) with Schäfer et al. (1996)

Coefficient	$n = 16$	$n = 32$	$n = 64$	Schäfer et al. (1996)
Drag	5.3203	5.4772	5.5799	5.5700–5.5900
Lift	0.0488	0.0141	0.0101	0.0104–0.0110

6.4.3 Three-Dimensional Problem

We demonstrate the extension of the approach to three dimensions in this section. Single relaxation-time D3Q27 LBM model is used here. As the name suggests, it consists of a three-dimensional lattice with particles restricted to move along 27 directions. The ghost fluid technique described in Sect. 6.3 is extended to three dimensions via tri-linear interpolation. In this case, an image point is surrounded by eight neighboring nodes; during interpolation, the outside nodes are replaced with the intersection of normal from those nodes with the boundary, analogous to the two-dimensional implementation. For demonstration, we simulate Taylor–Couette flow between cylinders (Fig. 6.7) using this approach (Tiwari et al. 2009). We consider the following configuration: radius ratio (inner radius/outer radius) is 0.5 and aspect ratio (height/inner radius) is 3.8. Rotational speed is chosen such that the Reynolds number based on the gap between the annulus and the tangential speed is 100. The generation of toroidal vortices due to flow instability at high Reynolds number is well known and widely studied (Wereley and Lueptow 1998) for the Taylor–Couette problem. Tiwari et al. (2009) simulated this configuration using GF-IB-LBM; Fig. 6.7 shows generation of vortices with a 125 × 125 × 95 lattice.

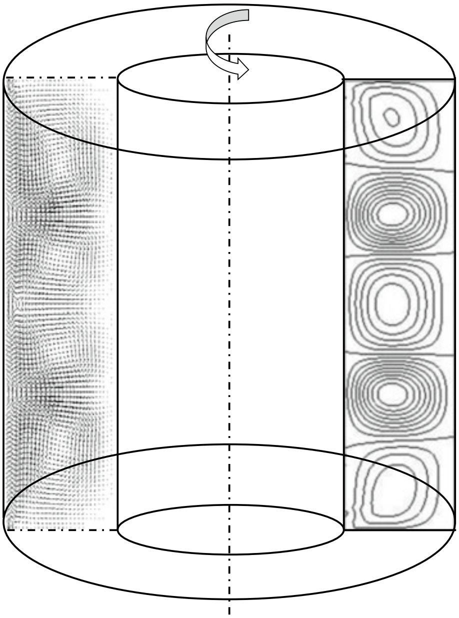


Fig. 6.7 Schematic, velocity vectors and contours of horizontal velocity component obtained by simulating Taylor–Couette problem using three-dimensional GF-IB-LBM (Tiwari et al. 2009)

6.4.4 Parallelization Using Graphical Processors

Lattice Boltzmann method is inherently highly parallelizable, hence significant speed-up can be obtained by utilizing graphical processing units (GPUs). One of the advantages of GF-IB-LBM is that it preserves the locality of the underlying LBM model—only needs information of the nodes surrounding the image points to enforce boundary conditions. Marsh (2010) developed a parallel implementation of LBM on GPUs and highlighted that parallelization of the ghost fluid technique was straightforward. They used Compute Unified Device Architecture (CUDA) (NVIDIA: https://developer.nvidia.com/cuda-zone), which is the melding of hardware and software that NVIDIA has provided to allow scientific applications to be more easily written and executed on an NVIDIA GPU. CUDA operates by executing threads on multiprocessors contained within the GPU. They reported 50–75 times speed-up on a modern GPU compared to a modern central processing unit (CPU) for their test problems using GF-IB-LBM.

6.4.5 Coupling with Molecular Dynamics

We next discuss the applicability of ghost fluid method in coupling LBM with molecular dynamics (MD) simulations. MD is an atomistic method, which has been used in computational studies for a long time, but has only recently become feasible for many applications due to its high computational cost. One of the target areas of MD simulations is nano- and micro-scale flows, where the flow behavior is not well described by the Navier–Stokes equations. MD examines physical phenomena on the atomistic scale by considering individual molecules and their interactions with each other. Extensive literature exists on interaction potentials (Allen et al. 2004); the Lennard-Jones potential is often used:

$$V_{ij}(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{6.7}$$

where V_{ij} and r_{ij} are the potential and distance, respectively, between molecules i and j. The two parameters ϵ and σ are used to characterize the interaction strength and length scale, respectively.

Coupling molecular dynamics with a non-atomistic CFD code such as lattice Boltzmann is advantageous when boundary wall effects are important to capture at the atomistic resolution, but the computational penalty of such high fidelity in the bulk flow is not desired. For example, consider a simple two-dimensional, planar channel flow as shown in Fig. 6.8. Marsh (2010) decomposed it into three domains for coupling: an MD domain near walls, an LBM domain for bulk flow and an overlap domain. They used Schwarz alternating method (SAM) (Dolean et al. 2015) to couple the two methods. This approach decouples both time and length scales

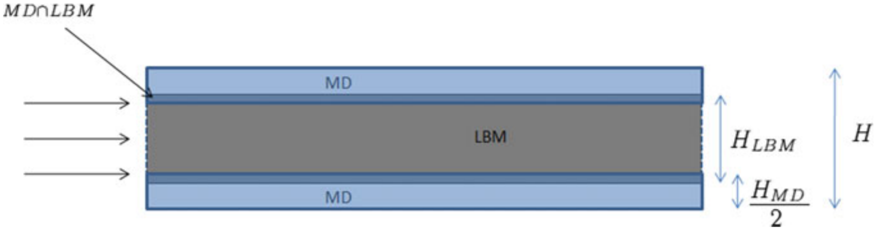
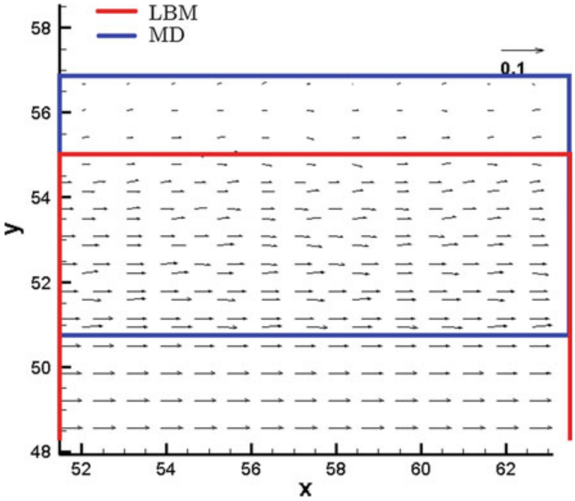


Fig. 6.8 Schematic of LBM-MD coupling in a planar channel (Marsh 2010)

Fig. 6.9 Velocity halfway through channel flow in the overlap region (Marsh 2010)



allowing for a fully hybrid scheme. LBM and MD individually solve their respective domains including the overlap region. SAM operates by first advancing LBM by one time step (Δt). Boundary conditions are then applied to the atomistic region via the overlap region. In this step, the velocity of a molecule in the overlap region is set to the fluid velocity at the nearest lattice node. MD is then advanced via multiple (p) smaller time steps (δt) such that $\Delta t = p\,\delta t$. Boundary conditions are then applied to the bulk region. GF-IB-LBM offers straightforward enforcement of boundary conditions from MD to LBM in this step. It just requires using the averaged values from MD and imposing them on the appropriate ghost nodes. Marsh (2010) developed a parallel implementation of the coupling approach on GPUs (as discussed in Sect. 6.4.4), and simulated the aforementioned channel flow configuration. For demonstration, we show in Fig. 6.9 the velocity vectors near wall obtained from the coupled simulation; we refer to Marsh (2010) for more details.

6.5 Recent Advances

Khazaeli et al. (2013) implemented ghost fluid technique on thermal LBM. Their approach uses interpolation–extrapolation methodology based on image points similar to Tiwari and Vanka's (2012) GFM. Thermal LBM contains an additional distribution function for internal energy, whose values they obtain at ghost nodes analogous to the particle distribution function calculation via extrapolation from image points. They used an inverse distance-weighing approach for interpolation and demonstrated second-order accuracy for several problems with curvilinear boundaries.

Chen et al. (2013) investigated pressure oscillations that appear due to boundary implementation in LBM. Their investigation focused on the ghost fluid IB method (Tiwari and Vanka 2012), for which, they implemented a cut-cell-based weighting strategy to enforce geometric conservation to suppress these oscillations. They tested this method on four problems and demonstrated that the modified GFM reduces pressure oscillations while preserving its accuracy. Chen et al. (2014) compared different bounce-back and IB schemes and concluded that the unified-interpolation bounce-back of Yu et al. (2003), the direct-forcing approach of Kang and Hassan (2011) and the ghost fluid approach of Tiwari and Vanka (2012) are best suited for the acoustic problems they considered.

Kaneda et al. (2014) developed a multi-relaxation-time extension of the single relaxation-time GFM (Tiwari and Vanka 2012). They compared their results with the standard bounce-back scheme and confirmed that GFM has better accuracy, but found a defect in density calculation at image points. They attributed this to a larger predicted pressure and proposed an improvement by implementing a normal moment relation (the balance of centrifugal force and pressure) for the estimation of density distribution functions at the boundaries. Jahanshaloo et al.'s (2016) review provides an overview of several approaches developed to impose boundary conditions in LBM (including thermal LBM).

Mozafari-Shamsi et al. (2016a) developed an extension of GFM (Tiwari and Vanka 2012) for thermal LBM and implemented it for Dirichlet as well as Neumann thermal boundary conditions. As mentioned earlier, thermal LBM contains an additional equation to evolve internal energy distribution functions. Internal energy distribution functions at the ghost nodes are obtained analogous to the calculation of particle distribution functions as described in Sect. 6.3; they employed GFM's inherent feature of computing gradient of the macroscopic variables normal to the curved boundaries to formulate heat flux (Neumann) boundary conditions. In a later study, they (Mozafari-Shamsi et al. 2016b) used this GFM approach to formulate conjugate heat transfer boundary conditions at curved interfaces of two materials having different thermal properties. Boundary conditions for conjugate heat transfer are difficult to impose because heat fluxes must match in addition to imposing a common temperature value at the boundary points. Here, again, GFM's (Tiwari and Vanka 2012) normal gradient calculation makes it suitable to enforce such interface conditions. They verified the accuracy and stability of GFM computations on three test problems and confirmed its second-order accuracy.

Li et al. (2016) developed a quadratic interpolation (QGFM) variant of bilinear interpolation GFM (BGFM) (Tiwari and Vanka 2012). They compared them with Guo et al.'s (2002) extrapolation method, linear interpolation bounce-back (LIBB) method and quadratic interpolation bounce-back (QIBB) method. They found that LIBB, QIBB, Guo et al.'s scheme and BGFM are comparable in efficiency for the problems simulated, but QGFM takes about 10% more computation time due to a larger stencil construction. As expected, they observed that quadratic interpolation schemes (QGFM and QIBB) are more accurate compared to their linear counterparts (BGFM and LIBB). However, they found that the conventional bounce-back schemes are more accurate than ghost fluid interpolation schemes for the problems studied. They also compared these techniques for boundary pressure oscillations in LBM and found that oscillations are best suppressed by Guo et al.'s scheme. The authors also studied the influence of different collision models, refilling techniques and force evaluation methods in suppressing pressure oscillations.

Xu et al. (2018) recently proposed a forcing-based IB-LBM scheme for fluid–structure interaction problems. To improve numerical stability, their scheme approximates the feedback coefficient explicitly and splits Lagrangian force into traction from surrounding flow and inertial force from boundary acceleration. They also developed a dynamic geometry-adaptive grid refinement strategy, which improves the efficiency of the coupled solution by having fine resolution only near the fluid–structure interfaces.

References

Allen MP et al (2004) Introduction to molecular dynamics simulation. Computational soft matter: from synthetic polymers to proteins, vol 23, pp 1–28

Balaras E (2004) Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. Comput Fluids 33(3):375–404

Bhatnagar PL, Gross EP, Krook M (1954) A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys Rev 94 (3):511

Chen S, Doolen GD (1998) Lattice Boltzmann method for fluid flows. Annu Rev Fluid Mech 30(1):329–364

Chen H, Chen S, Matthaeus WH (1992) Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. Phys Rev A 45(8):R5339

Chen S, Martinez D, Mei R (1996) On boundary conditions in lattice Boltzmann methods. Phys Fluids 8(9):2527–2536

Chen L, Yu Y, Hou G (2013) Sharp-interface immersed boundary lattice Boltzmann method with reduced spurious-pressure oscillations for moving boundaries. Phys Rev E 87(5):053306

Chen L, Yu Y, Lu J, Hou G (2014) A comparative study of lattice Boltzmann methods using bounce-back schemes and immersed boundary ones for flow acoustic problems. Int J Numer Methods Fluids 74(6):439–467

Dolean V, Jolivet P, Nataf F (2015) An introduction to domain decomposition methods: algorithms, theory, and parallel implementation, vol 144. SIAM

Dupuis A, Chatelain P, Koumoutsakos P (2008) An immersed boundary-lattice-Boltzmann method for the simulation of the flow past an impulsively started cylinder. J Comput Phys 227(9):4486–4498

Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J (2000) Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. J Comput Phys 161(1):35–60

Feng Z-G, Michaelides EE (2004) The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. J Comput Phys 195(2):602–628

Feng Z-G, Michaelides EE (2005) Proteus: a direct forcing method in the simulations of particulate flows. J Comput Phys 202(1):20–51

Filippova O, Hänel D (1998) Grid refinement for lattice-BGK models. J Comput Phys 147(1):219–228

Frisch U, Hasslacher B, Pomeau Y (1986) Lattice-gas automata for the Navier-Stokes equation. Phys Rev Lett 56(14):1505

Gilmanov A, Sotiropoulos F (2005) A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. J Comput Phys 207(2):457–492

Ginzburg I, d'Humieres D (2003) Multireflection boundary conditions for lattice Boltzmann models. Phys Rev E 68(6):066614

Glowinski R, Pan T-W, Hesla TI, Joseph DD (1999) A distributed Lagrange multiplier/fictitious domain method for particulate flows. Int J Multiph Flow 25(5):755–794

Goldstein D, Handler R, Sirovich L (1993) Modeling a no-slip flow boundary with an external force field. J Comput Phys 105(2):354–366

Guo Z, Zheng C, Shi B (2002) An extrapolation method for boundary conditions in lattice Boltzmann method. Phys Fluids 14(6):2007–2010

Higuera FJ, Jiménez J (1989) Boltzmann approach to lattice gas simulations. EPL (Europhys Lett) 9(7):663

Jahanshaloo L, Sidik NAC, Fazeli A, Mahmoud Pesaran HA (2016) An overview of boundary implementation in lattice Boltzmann method for computational heat and mass transfer. Int Commun Heat Mass Transfer 78:1–12

Kaneda M, Haruna T, Suga K (2014) Ghost-fluid-based boundary treatment in lattice Boltzmann method and its extension to advancing boundary. Appl Therm Eng 72(1):126–134

Kang SK, Hassan YA (2011) A comparative study of direct-forcing immersed boundary-lattice Boltzmann methods for stationary complex boundaries. Int J Numer Methods Fluids 66(9):1132–1158

Khazaeli R, Mortazavi S, Ashrafizaadeh M (2013) Application of a ghost fluid approach for a thermal lattice Boltzmann method. J Comput Phys 250:126–140

Ladd AJC (1994) Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. J Fluid Mech 271:285–309

Lai M-C, Peskin CS (2000) An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. J Comput Phys 160(2):705–719

Lallemand P, Luo L-S (2003) Lattice Boltzmann method for moving boundaries. J Comput Phys 184(2):406–421

Lee L, LeVeque RJ (2003) An immersed interface method for incompressible Navier-Stokes equations. SIAM J Sci Comput 25(3):832–856

Li X, Jiang F, Hu C (2016) Analysis of the accuracy and pressure oscillation of the lattice Boltzmann method for fluid-solid interactions. Comput Fluids 129:33–52

Luo L-S (2000) Theory of the lattice Boltzmann method: lattice Boltzmann models for nonideal gases. Phys Rev E 62(4):4982

Maier RS, Bernard RS, Grunau DW (1996) Boundary conditions for the lattice Boltzmann method. Phys Fluids 8(7):1788–1801

Majumdar S, Iaccarino G, Durbin P (2001) RANS solvers with adaptive structured boundary non-conforming grids. Annual research briefs, Center for Turbulence Research, pp 353–366

Marsh DD (2010) Molecular dynamics-lattice Boltzmann hybrid method on graphics processors. University of Illinois at Urbana-Champaign

McNamara GR, Zanetti G (1988) Use of the Boltzmann equation to simulate lattice-gas automata. Phys Rev Lett 61(20):2332

Mei R, Luo L-S, Shyy W (1999) An accurate curved boundary treatment in the lattice Boltzmann method. J Comput Phys 155(2):307–330

Mohd-Yusof J (1997) Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries. Annual research briefs, Center for Turbulence Research, pp 317–327

Mozafari-Shamsi M, Sefid M, Imani G (2016a) Developing a ghost fluid lattice Boltzmann method for simulation of thermal Dirichlet and Neumann conditions at curved boundaries. Numer Heat Transfer Part B: Fundam 70(3):251–266

Mozafari-Shamsi M, Sefid M, Imani G (2016b) New formulation for the simulation of the conjugate heat transfer at the curved interfaces based on the ghost fluid lattice Boltzmann method. Numer Heat Transfer Part B: Fundam 70(6):559–576

Niu XD, Shu C, Chew YT, Peng Y (2006) A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows. Phys Lett A 354(3):173–182

Noble DR, Chen S, Georgiadis JG, Buckius RO (1995) A consistent hydrodynamic boundary condition for the lattice Boltzmann method. Phys Fluids 7(1):203–209

NVIDIA. CUDA. https://developer.nvidia.com/cuda-zone

Peng Y, Shu C, Chew Y-T, Niu XD, Lu X-Y (2006) Application of multi-block approach in the immersed boundary-lattice Boltzmann method for viscous fluid flows. J Comput Phys 218(2):460–478

Peskin CS (1972) Flow patterns around heart valves: a numerical method. J Comput Phys 10(2):252–271

Qian Y-H, d'Humières D, Lallemand P (1992) Lattice BGK models for Navier-Stokes equation. EPL (Europhys Lett) 17(6):479

Schäfer M, Turek S, Durst F, Krause E, Rannacher R (1996) Benchmark computations of laminar flow around a cylinder. In: Flow simulation with high-performance computers II. Springer, German, pp 547–566

Shukla RK, Pantano C, Freund JB (2010) An interface capturing method for the simulation of multi-phase compressible flows. J Comput Phys 229(19):7411–7439

Taira K, Colonius T (2007) The immersed boundary method: a projection approach. J Comput Phys 225(2):2118–2137

Tian F-B, Luo H, Zhu L, Liao JC, Lu X-Y (2011) An efficient immersed boundary-lattice Boltzmann method for the hydrodynamic interaction of elastic filaments. J Comput Phys 230(19):7266–7283

Tiwari A, Vanka SP (2012) A ghost fluid Lattice Boltzmann method for complex geometries. Int J Numer Methods Fluids 69(2):481–498

Tiwari A, Freund JB, Pantano C (2013) A diffuse interface model with immiscibility preservation. J Comput Phys 252:290–309

Tiwari A, Samala R, Vanka SP (2009) Ghost fluid based immersed boundary treatment for lattice Boltzmann method. In: ASME 2009 international mechanical engineering congress and exposition. American Society of Mechanical Engineers, pp 309–313

Tseng Y-H, Ferziger JH (2003) A ghost-cell immersed boundary method for flow in complex geometry. J Comput Phys 192(2):593–623

Uhlmann M (2005) An immersed boundary method with direct forcing for the simulation of particulate flows. J Comput Phys 209(2):448–476

Wereley ST, Lueptow RM (1998) Spatio-temporal character of non-wavy and wavy Taylor-Couette flow. J Fluid Mech 364:59–80

Wu J, Shu C (2009) Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. J Comput Phys 228(6):1963–1979

Xu L, Tian F-B, Young J, Lai JCS (2018) A novel geometry-adaptive Cartesian grid based immersed boundary-lattice Boltzmann method for fluid-structure interactions at moderate and high Reynolds numbers. J Comput Phys 375:22–56

Ye T, Mittal R, Udaykumar HS, Shyy W (1999) An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. J Comput Phys 156(2):209–240

Yu D, Mei R, Shyy W (2003) A unified boundary treatment in lattice Boltzmann method. In: 41st aerospace sciences meeting and exhibit, p 953

Zhang J, Johnson PC, Popel AS (2007) An immersed boundary lattice Boltzmann approach to simulate deformable liquid capsules and its application to microscopic blood flows. Phys Biol 4(4):285

Ziegler DP (1993) Boundary conditions for lattice Boltzmann simulations. J Stat Phys 71(5–6):1171–1177

Zou Q, He X (1997) On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. Phys Fluids 9(6):1591–1598