



Flow force calculation in the lattice Boltzmann method

Shaurya Kaushal ¹, Sauro Succi,² and Santosh Ansumali ¹

¹*Engineering Mechanics Unit, Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore 560064, India*

²*IIT@La Sapienza and Research Affiliate Physics Department Harvard University, Cambridge, Massachusetts 02138, USA*



(Received 20 October 2022; revised 21 February 2023; accepted 17 August 2023; published 6 October 2023)

We revisit force evaluation methodologies on rigid solid particles suspended in a viscous fluid that is simulated via the lattice Boltzmann method (LBM). We point out the noncommutativity of streaming and collision operators in the force evaluation procedure due to the presence of a solid boundary, and provide a theoretical explanation for this observation. Based on this analysis, we propose a discrete force calculation scheme with enhanced accuracy. The proposed scheme is essentially a discrete version of the Reynolds transport theorem (RTT) in the context of a lattice Boltzmann formulation. Besides maintaining satisfactory levels of reliability and accuracy, the method also handles force evaluation on complex geometries in a simple and transparent way. We run benchmark simulations for flow past cylinder and NACA0012 airfoil (for Reynolds numbers ranging from 10^2 to 0.5×10^6) and show that the current approach significantly reduces the grid size requirement for accurate force evaluation.

DOI: [10.1103/PhysRevE.108.045304](https://doi.org/10.1103/PhysRevE.108.045304)

I. INTRODUCTION

The lattice Boltzmann method (LBM) is a discrete space-time kinetic theory that has made major leaps in solving hydrodynamic problems at low Mach numbers [1–8]. The theoretical foundation of the algorithm is well established [9–11], and boundary conditions at the level of discrete populations are well developed with a number of variants ranging from the simple and efficient bounce-back (BB) boundary condition to the microscopic diffusive boundary condition [12]. An important asset of the method is its ability to handle fluid-solid systems, comprising of complex solid shapes and moving boundaries in a simplified but efficient manner. For such systems, along with appropriate treatment of boundary conditions, an accurate calculation of force (lift and drag) or torque on the solid body is often crucial.

The two widely used methodologies for computing these hydrodynamic forces on solid objects are the stress integration (SI) approach [13–15] and the momentum exchange algorithm (MEA) [16]. In the stress integration method (as the name suggests), the force is computed by integrating the stress tensor on the surface of the body. In the momentum exchange algorithm (MEA), the force is computed by accounting for microscopic exchanges of momentum at the wall between the fluid and solid boundary, directly in terms of the discrete probability density function. Even though both methods are well established, the momentum exchange method is considered to be more accurate than the stress integration method at moderate to high Reynolds numbers [17]. In an attempt to accurately capture the fluid-solid interaction, a variety of methods have been successfully developed for smoothing curved solid surfaces. These include, the immersed boundary method [18–20], the partially saturated method (PSM) [21,22], the local second-order boundary approach (LSOB) [23], near boundary interpolation [24], and multiresolution frameworks. There has also been significant work in the

direction of improving the stability range of the bounce-back family of boundary stencil [25–29]. In its most basic version, the solid boundary in LBM is approximated in a discrete sense at the middle of every fluid-solid link (marked by stars in Fig. 1 is the discrete boundary as per half way bounce back), each of which crosses the boundary and connects a fluid node. Being a link based formulation, the bounce-back boundary condition minimizes, but does not avoid errors due to the staircase geometry of the boundary. Even in this case, when the boundary of solid objects is laid down approximately, MEA is shown to be quite effective [16,30].

In this work, without loss of any generality, we have adhered to the simple and widely used half way bounce-back (HWBB) treatment of the complex boundary. We point out that the essence of this work remains intact even if one chooses a more sophisticated boundary treatment. LBM is a repeated sequence of collision and streaming [31,32] (for second order temporal accuracy refer to Dellar *et al.* [31]), and the presence of a solid boundary further breaks the symmetry and adds additional noncommutativity between operators. Streaming is the generator of spatial translations while the boundary operator, by construction, must break translational invariance. Hence, in the presence of solid boundaries, the correct order of operations to calculate the flow forces, is not self evident. We have described two possible sequences of operations indicated by algorithm A and algorithm B in Fig. 1. Even though, the use of algorithm B is noticed in some literature [16,33], a detailed comparison between the two approaches and an analysis as to why one is better than the other is not present in any work. We have simulated three test cases in order to compare the two algorithms, that is, a 2D gravity driven channel (Table I), flow past a 2D circular cylinder (Fig. 2) and flow past a 3D NACA0012 airfoil (Fig. 3), at moderate Reynolds numbers. These are standard nontrivial test cases extensively used for validation of numerical schemes and a good amount of experimental benchmark and computational

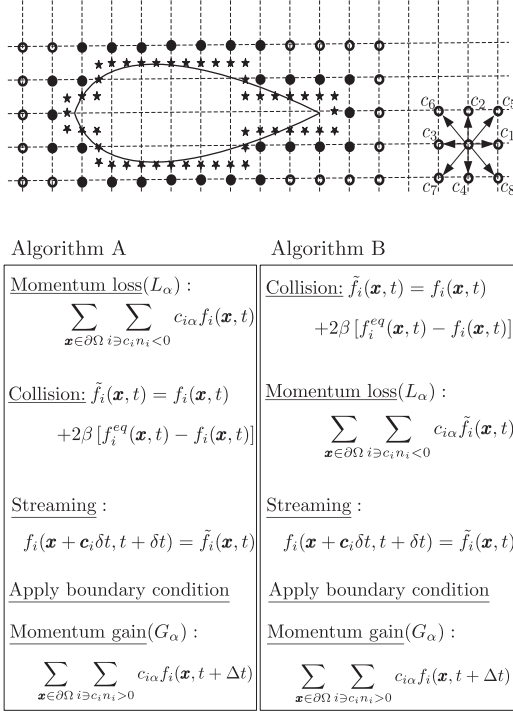


FIG. 1. Top: An airfoil shaped solid object placed in a 2D cartesian grid (fluid nodes marked by hollow circles, boundary fluid nodes marked by solid circles) with a nine velocity lattice boltzmann model. The discrete boundary points (marked by stars) are placed exactly halfway between the fluid-solid links for a simple bounce-back boundary condition. Bottom: The two algorithms are displayed next to each other for comparison. The surface normal pointing into the fluid is denoted by n_i , and $\partial\Omega$ represents the boundary grid points.

data is available for the same [34–37]. To emphasize that the discrepancy in drag values between the two algorithms is generic, the force is computed with two discrete velocity models (D2Q9 and RD3Q41).

A detailed analysis of the force evaluation routine, naturally lead us to the next part of the work, where we formulated a discrete analog of the Reynolds transport theorem (DRTT) in the discrete space-time setting of the lattice boltzmann equation and devised a simple and effective way for flow

TABLE I. Steady state values of coefficient of drag for a 2D gravity driven channel using a D2Q9 model at $Re = 1, 10, 100$ and $N_y = 100$ as the number of grid points in the transverse direction. The drag values are calculated at the bottom boundary point ($N_x/2, 1$) and compared against the analytical solution which is readily available for the velocity profile in such a system (known to be parabolic). We have also presented results using the open source example code provided by OpenLB, the formulation for which is very similar to algorithm B.

Re	C_d	C_d	C_d	C_d
	(Algorithm A)	(Algorithm B)	(Open LB)	(Analytical)
1	1.68×10^{-1}	1.60×10^{-1}	1.59×10^{-1}	1.57×10^{-1}
10	2.79×10^{-2}	1.78×10^{-2}	1.79×10^{-2}	1.76×10^{-2}
100	10.8×10^{-3}	1.62×10^{-3}	1.62×10^{-3}	1.61×10^{-3}

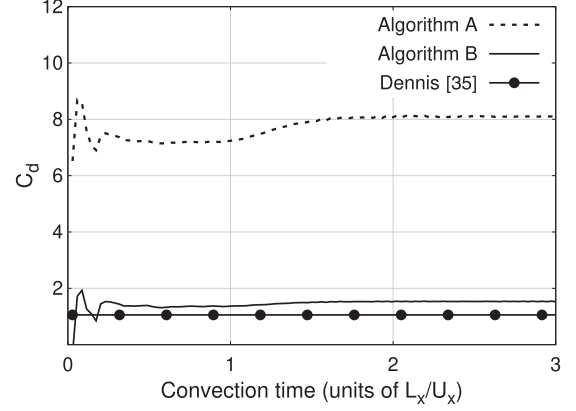


FIG. 2. Coefficient of drag vs convection time for flow past a two dimensional circular cylinder of diameter D solved using a D2Q9 model at $Re = 100$ using 20 grid points per diameter. The aspect ratio of the computational domain is taken to be 8:1 with $120D \times 15D$ points for the channel length and width, respectively. The cylinder center is symmetrically placed at $30D$ from the inlet. L_x represents the length of the computational channel and U_x represents the inlet velocity. The lattice Boltzmann results are compared with established steady state drag values in literature [35].

force calculation. The main idea of this approach is that the complex-surfaced solid can be enclosed inside a simpler cartesian-friendly bounding box, creating a control volume that facilitates balancing of momentum fluxes and accurate calculation of surface stresses on the solid boundary.

To demonstrate this approach, we consider a 1D toy model with X_1 as the boundary fluid grid point of the control volume and X_2, X_3, X_4 , and so on, as the bulk fluid grid points (see Fig. 4). The standard lattice Boltzmann equation (in one dimensional space) with a relaxation term is a two step evolution given by

$$\begin{aligned} \tilde{f}_i(X, t) &= f_i(X, t) + 2\beta [f_i^{eq}(X, t) - f_i(X, t)], \\ f_i(X + c_i \delta t, t + \delta t) &= \tilde{f}_i(X, t). \end{aligned} \quad (1)$$

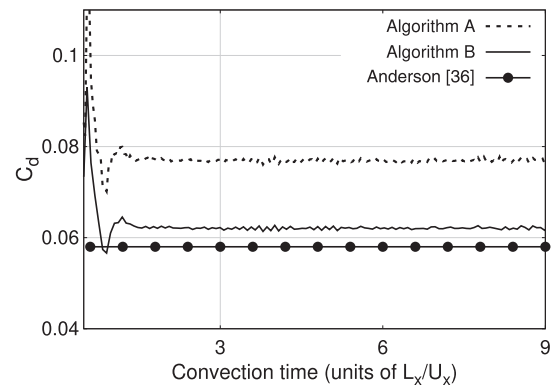


FIG. 3. Coefficient of drag vs convection time for flow past a 3D NACA0012 airfoil (12% thickness to chord length ratio) at $AoA = 0^\circ$, using a D2Q9 model at $Re = 5000$, using 320 grid points per chord (ppc) length. The size of the computational domain is taken to be $(80 \times 8 \times 4)$ ppc with the airfoil symmetrically placed at 24ppc from the inlet. The lattice Boltzmann results are compared with established steady state drag values in literature [36].

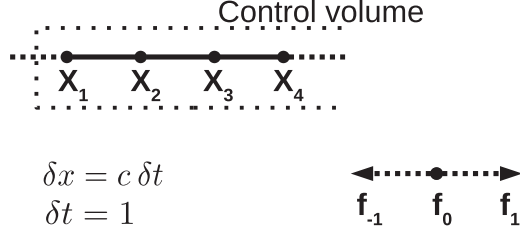


FIG. 4. A D1Q3 lattice with grid points (X_1, X_2, X_3, X_4) , where X_1 is the boundary grid point of the control volume and X_2, X_3, X_4 are the bulk grid points. In the toy example, we assume that the grid is unbounded on the right end and continues to X_5, X_6 , and so on.

A more detailed description of notations used in lattice Boltzmann literature is given in the next section. It would be convenient for further analysis to define the global momentum calculated over the control volume, at time t as

$$J_X(t) = \sum_{q=1,2,3,4,\dots} \sum_i f_i(X_q, t) c_{iX}. \quad (2)$$

Similarly, the global momentum at time $t + 1$ is, term by term, given by

$$\begin{aligned} J_X(t + 1) = & [f_1(X_1, t + 1) - f_{-1}(X_1, t + 1)] \\ & + [f_1(X_2, t + 1) - f_{-1}(X_2, t + 1)] \\ & + [f_1(X_3, t + 1) - f_{-1}(X_3, t + 1)] \\ & + [f_1(X_4, t + 1) - f_{-1}(X_4, t + 1)] + \dots \end{aligned} \quad (3)$$

The streaming step, as per Eq. (1), can be used to simplify every term to

$$\begin{aligned} J_X(t + 1) = & f_1^B(X_1, t + 1) + \tilde{f}_1(X_1, t) \\ & + [\tilde{f}_1(X_2, t) - \tilde{f}_{-1}(X_2, t)] \\ & + [\tilde{f}_1(X_3, t) - \tilde{f}_{-1}(X_3, t)] \\ & + [\tilde{f}_1(X_4, t) - \tilde{f}_{-1}(X_4, t)] + \dots, \end{aligned} \quad (4)$$

where, $f_1^B(X_1, t + 1)$ is the boundary condition for a simple halfway bounce-back condition. Upon subtracting with $J_X(t)$, the evolution equation for global momentum derived from Eq. (1), for a semi-infinite control volume with only X_1 as the boundary point, simplifies to

$$\begin{aligned} J_X(t + 1) - J_X(t) = & f_1^B(X_1, t + 1) + \tilde{f}_1(X_1, t) \\ & - (\tilde{f}_1(X_1, t) - \tilde{f}_{-1}(X_1, t)), \\ = & f_1^B(X_1, t + 1) + \tilde{f}_{-1}(X_1, t). \end{aligned} \quad (5)$$

According to Newton's second law of motion, the RHS of global momentum balance in Eq. (5) is the total force that the boundary applies on the system. It is important to note that for a HWBB boundary condition [in accordance with the steps in Fig. 1, $f_1^B(X_1, t + 1) = \tilde{f}_{-1}(X_1, t)$], this result can be simplified to give

$$J_X(t + 1) - J_X(t) = 2\tilde{f}_{-1}(X_1, t). \quad (6)$$

As per algorithm A (see Fig. 1), for the 1D toy problem, the force is given by

$$\begin{aligned} F_X^A = & \underbrace{\tilde{f}_{-1}(X_1, t)}_{\text{gain}} - \underbrace{(-f_{-1}(X_1, t))}_{\text{loss}}, \\ = & 2\tilde{f}_{-1}(X_1, t) - 2\beta[f_{-1}^{\text{eq}}(X_1, t) - f_{-1}(X_1, t)], \end{aligned} \quad (7)$$

and as per algorithm B, for the 1D toy problem the force is given by

$$\begin{aligned} F_X^B = & \underbrace{\tilde{f}_{-1}(X_1, t)}_{\text{gain}} - \underbrace{(-\tilde{f}_{-1}(X_1, t))}_{\text{loss}}, \\ = & 2\tilde{f}_{-1}(X_1, t). \end{aligned} \quad (8)$$

Thus, one must note that the force calculation routine in algorithm B exactly matches the control volume derivation in Eq. (6). The implications of this difference between the algorithms can be seen in Table I, Figs. 2, and 3, where algorithm A consistently over predicts the drag values.

The outline of this paper is as follows: In Sec. II, we give a brief description of LBM. In Sec. III, we review widely used methods for flow force calculation in LBM literature, namely stress integration and momentum exchange. In Sec. IV, we formulate a discrete analog of the Reynolds transport theorem. In Sec. V, we demonstrate some numerical simulations to validate our findings.

II. LATTICE BOLTZMANN MODEL

Lattice Boltzmann method represents hydrodynamics by a discrete space-time kinetic theory. The starting point for the method, is the construction of a discrete velocity set, \mathcal{C} , consisting of N_d discrete velocities given by \mathbf{c}_i ($i = 1, 2, \dots, N_d$). The set of basic variables, \mathbf{f} , for discrete kinetic theory, are populations $f_i(\mathbf{x}, t)$ (of discrete velocities \mathbf{c}_i) defined at location \mathbf{x} and time t . The hydrodynamic variables are the mass density ρ , the fluid velocity \mathbf{u} , and scaled temperature θ , defined in terms of Boltzmann constant k_B and mass of the particle m , as $\theta = k_B T / m$. These macroscopic variables are related to the populations as

$$\rho = \sum_i^{N_d} f_i, \quad \rho u_\alpha = \sum_i^{N_d} f_i c_{i\alpha}, \quad \rho u^2 + D\rho\theta = \sum_i^{N_d} f_i c_i^2, \quad (9)$$

where D signifies the dimension of the setup. For these discrete velocity models, typically the kinetic equation is written with a single relaxation (BGK) term as

$$\frac{\partial f_i}{\partial t} + \mathbf{c}_i \cdot \nabla f_i = -\frac{1}{\tau} [f_i - f_i^{\text{eq}}(\mathcal{M}^{\text{Slow}})], \quad (10)$$

where τ is related to mean free time and $f_i^{\text{eq}}(\mathcal{M}^{\text{Slow}})$ is the discrete analog of Maxwell-Boltzmann distribution with $\mathcal{M}^{\text{Slow}}$ being the set of slow hydrodynamic moments of populations. In continuous kinetic theory, the slow moments consist of mass, momentum, and energy density ($\mathcal{M}^{\text{Slow}} = \{\rho, \rho\mathbf{u}, \rho u^2 + 3\rho\theta\}$), while in the discrete case one can often ignore energy conservation to focus on isothermal hydrodynamics only ($\mathcal{M}^{\text{Slow}} = \{\rho, \rho\mathbf{u}\}$). The lattice Boltzmann equation is obtained by further discretizing Eq. (10) in space

and time, as a sequence of discrete collisions followed by free flight (streaming) as

$$\begin{aligned} \tilde{f}_i(x, t) &= f_i(x, t) + 2\beta \\ &\quad \times [f_i^{\text{eq}}(\rho(x, t), u(x, t)) - f_i(x, t)], \\ f_i(x + c_i \delta t, t + \delta t) &= \tilde{f}_i(x, t), \end{aligned} \quad (11)$$

where $\beta = \delta t / (2\tau + \delta t)$.

The main theoretical ingredient ensuring accurate hydrodynamics is the construction of the discrete equilibrium. Starting with a zero velocity equilibrium ($w_i > 0$), a second order series approximation to the equilibrium is often written as

$$f_i^{\text{eq}} = w_i \rho \left[1 + \frac{(u \cdot c_i)}{\theta_0} + \frac{1}{2\theta_0^2} ((u \cdot c_i)^2 - \theta_0 u^2) \right]. \quad (12)$$

Here, it is important that the zero velocity equilibrium at a fixed temperature is positive ($w_i > 0$) and satisfies the condition that lower order moments are the same as Maxwell-Boltzmann. In particular,

$$\begin{aligned} \sum_i^{N_d} w_i &= 1, \quad \sum_i^{N_d} w_i c_{i\alpha} c_{i\beta} = \theta_0 \delta_{\alpha\beta}, \\ \sum_i^{N_d} w_i c_{i\alpha} c_{i\beta} c_{i\gamma} &= \theta_0^2 \Delta_{\alpha\beta\gamma\kappa}. \end{aligned} \quad (13)$$

These conditions on zero velocity equilibrium ensures

$$P_{\alpha\beta}^{\text{eq}} \equiv \sum_i f_i^{\text{eq}} c_{i\alpha} c_{i\beta} = \rho u_\alpha u_\beta + \rho \theta_0 \delta_{\alpha\beta}. \quad (14)$$

Later with higher order lattices, the method was extended for finite, but subsonic Mach number case by ensuring that the contracted sixth order moment [38,39] is

$$\sum_i w_i c_i^2 c_{i\alpha} c_{i\beta} c_{i\gamma} = 7\theta_0^2 \Delta_{\alpha\beta\gamma\kappa}. \quad (15)$$

This allows one to write the equilibrium for a small departure from reference temperature, at least at zero velocity (second order in $\Delta\theta$), as

$$f_i^{(0)} = w_i \left[1 + \frac{\Delta\theta}{2} (c_i - 3) + \frac{\Delta\theta^2}{8} (c_i^2 - 10c_i + 15) \right], \quad (16)$$

which gives a second order (in velocity) approximation to the equilibrium

$$\frac{f_i^{\text{Eq}}}{\rho f_i^{(0)}} = 1 + \frac{u \cdot c_i}{\theta} + \frac{(u \cdot c_i)^2 - \theta u^2 - \frac{5\bar{R}u^2}{6+15\bar{R}} (c_i^2 - 3\theta)}{2\theta^2}. \quad (17)$$

The definition of \bar{R} can be found in the cited reference [38]. The weights and discrete velocity set for two used models (D2Q9 and RD3Q41) are provided in Tables II and III. The energy shells for both the models is depicted in Figs. 1 and 5.

Arguably, the most crucial element of the lattice Boltzmann method is the boundary condition at a solid-fluid interface. Even though the macroscopic boundary conditions are quite straight forward, the mesoscopic boundary conditions on the discrete population [$f_i(x, t)$] require some

TABLE II. Energy shells and their corresponding velocities with weights for D2Q9 with $\theta_0 = 1/3$.

Shells	Discrete velocities(c_i)	Weight(w_i)
Zero	(0, 0)	16/36
SC-1	($\pm 1, 0$), (0, ± 1)	4/36
FCC-1	($\pm 1, \pm 1$)	1/36

explanation. Historically, the two commonly used boundary conditions are halfway bounce-back (HWBB) and diffusive boundary conditions [4,16,40–42]. In recent years, hybrid formulations, like the diffused bounce-back boundary condition [43,44], aimed at expanding the scope of application to a variety of problems are also being used.

In order to keep the message of this paper clear, we stick to the universally used HWBB boundary condition proposed by Ladd [16]. The basic implementation of HWBB boundary condition assumes that a molecule hits the wall and reverses its direction, carrying the same magnitude of momentum. If $x \in x_f$ and $x + c_i \delta t \in x_s$ (where x_f and x_s denote the fluid and solid domain respectively), the boundary point, x_b , is assumed to lie at the midpoint of the vector joining x_f and x_s , regardless of the physical position of the boundary (see Fig. 1). The distribution for the fluid points near the boundary is given by the boundary condition [16],

$$\begin{aligned} \text{stationary wall: } f_{\tilde{i}}(x_f, t + \delta t) &= \tilde{f}_i(x_f, t), \\ \text{moving wall: } f_{\tilde{i}}(x_f, t + \delta t) &= \tilde{f}_i(x_f, t) - 2w_i \rho \frac{c_i \cdot u_w}{c_s^2}, \end{aligned} \quad (18)$$

where \tilde{i} is the direction opposite to i , and u_w is the velocity of the solid wall. Here, \tilde{f} represents the post-collision population. There are several higher-order interpolation based curved boundary conditions in literature that address the drawbacks in accuracy of HWBB boundary condition when dealing with curved surfaces [24]. However, the essential ideas of this paper remain unchanged even with higher-order interpolation based schemes and can be transferred to the same.

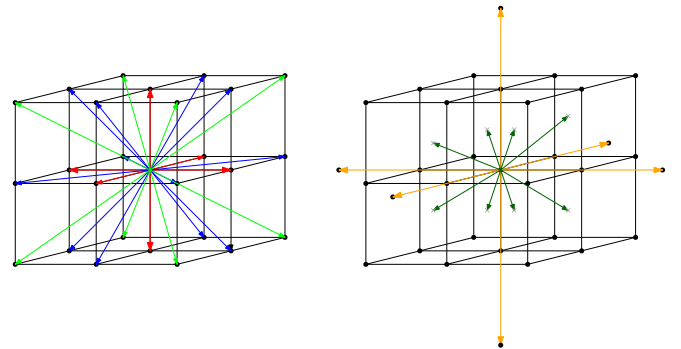


FIG. 5. (a) Energy shells of the D2Q9 model. (b) Energy shells of the RD3Q41 model: fcc-1 (blue), sc-1 (red), bcc-1 (light green) shown on a regular lattice along with sc-2 (orange), bcc-1/2 (dark green).

TABLE III. Velocities and their corresponding weights for the RD3Q41 model with $\theta_0 = 0.2948964908710633$.

Shells	Discrete velocities(c_i)	Weight(w_i)
Zero	(0, 0, 0)	$(52 - 323\theta_0 + 921\theta_0^2 - 1036\theta_0^3)/52$
SC-1	$(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)$	$\theta_0(12 - 38\theta_0 + 63\theta_0^2)/39$
SC-2	$(\pm 2, 0, 0), (0, \pm 2, 0), (0, 0, \pm 2)$	$\theta_0(3 - 29\theta_0 + 84\theta_0^2)/312$
FCC-1	$(\pm 1, \pm 1, 0), (\pm 1, 0, \pm 1), (0, \pm 1, \pm 1)$	$\theta_0(45\theta_0 - 6 - 77\theta_0^2)/26$
BCC-1	$(\pm 1, \pm 1, \pm 1)$	$\theta_0(20 - 163\theta_0 + 378\theta_0^2)/312$
BCC-0.5	$(\pm 0.5, \pm 0.5, \pm 0.5)$	$8\theta_0(4 - 17\theta_0 + 21\theta_0^2)/39$

III. FORCE COMPUTATION IN LBM

Very often, fluid simulations require accurate determination of forces experienced by solid objects immersed in them. In this section, we begin by briefly reviewing the two widely used force evaluation schemes in the context of LBM. For fluid simulations with any method, an intuitive way of calculating forces is to perform integration of the total stress on the contact surface. This method and all its variants are referred to as stress integration (SI) algorithms.

In terms of the unit normal n pointing out of the solid boundary $\partial\Omega$, the stress tensor on the boundary of a solid for an incompressible flow is

$$F = \int_{\partial\Omega} dA n \cdot (-pI + \rho v[(\nabla : u) + (\nabla : u)^T]). \quad (19)$$

In LBM formulation, the pressure term $p = \rho\theta$ ($\theta = \theta_0$ for isothermal models) can be computed easily at every grid point. However, the deviatoric part of the stress tensor involves the velocity gradient tensor $\partial_\alpha u_\beta$ that needs to be approximated via finite differences, which typically introduces additional error. This inaccuracy in calculation of velocity gradient tensor is circumvented in LBM by observation that the stresses are evaluated as the second moment of the nonequilibrium part of the population [14],

$$\sigma_{\alpha\beta} = -p\delta_{\alpha\beta} - \left(1 - \frac{1}{2\tau}\right) \Sigma_i [f_i(x, t) - f_i^{\text{eq}}(x, t)] \left(c_{i\alpha} c_{i\beta} - \frac{\theta_0}{D} \delta_{\alpha\beta} \right). \quad (20)$$

However, LBM being a Cartesian grid based method, another uncertainty in force estimation is added by the calculation of a surface normal (n) for complex geometries. Thus, for a typical flow setup the SI calculation results have lower accuracy in contrast to the alternate, the momentum exchange method [17]. A computationally efficient and simple method to calculate flow forces, is the momentum exchange (ME) algorithm [16] where the total forces are computed as a pairwise sum of momentum difference in discrete directions as populations bounces back from body surface near boundary points. In its basic version (see Fig. 1), boundary is approximated quite effectively in a staircase manner at the middle of every fluid-solid link, each of which crosses the boundary and connects a fluid node. The total force experienced by the solid object is given by

$$F_b = \sum_b -[f_{\tilde{i}}(x_f, t + \delta t)c_{\tilde{i}} - \tilde{f}_{\tilde{i}}(x_f, t)c_i], \quad (21)$$

where, \tilde{i} is the direction opposite to i and the summation runs over all fluid points next to discrete boundary.

Many references on MEA overlook the ambiguity in the choice of algorithm (A or B), pointed out in the introduction of the paper, and intuitively suggest using post collision population for the calculation of forces [33,45,46]. Through the course of a simple 1D example (see Fig. 4) and test simulations (see Table I, Figs. 2, and 3), we have established that algorithm B is the right choice. Another important observation is that MEA is a limit to the more general Reynolds transport theorem [47]. In the next section we derive the discrete version of the Reynolds transport theorem and arrive at a simplistic force evaluation routine for complex shaped objects.

IV. DISCRETE REYNOLDS TRANSPORT THEOREM

Consider a complex shaped solid object with a continuous boundary marked by $\partial\Omega$ as shown in Fig. 6, placed on a Cartesian grid. The solid is surrounded by fluid nodes (Ω) and the boundary fluid nodes are represented by $\partial\Omega^D$ (marked by solid circles).

The total momentum for all the fluid grid nodes in the bulk and boundary is given by

$$J_\alpha^\Omega(t) = \sum_{x \in \Omega} \sum_i c_{i\alpha} f_i(x, t). \quad (22)$$

The momentum balance for the fluid points as the system evolves from t to $t + \Delta t$ implies

$$J_\alpha^\Omega(t + \Delta t) = J_\alpha^\Omega(t) + J_\alpha^G(t + \Delta t) - J_\alpha^L(t), \quad (23)$$

where $J_\alpha^G(t + \Delta t)$ and $J_\alpha^L(t)$ represents the gained and lost momentum at the boundary [47], as seen in Fig. 6. It must be noted that J_α^G is calculated over the boundary grid nodes where population is added to the system and J_α^L is calculated over the boundary grid nodes where population leaves the system when the system displaces from t to $t + \Delta t$. The gained momentum is simply written as

$$\begin{aligned} J_\alpha^G(t + \Delta t) &= \sum_{x \in \partial\Omega^D} \left[\sum_{i \ni c_i n_i > 0} c_{i\alpha} f_i(x, t + \Delta t) \right], \\ &= \sum_{x \in \partial\Omega^D} \left[\sum_{i \ni c_i n_i > 0} c_{i\alpha} \tilde{f}_{\tilde{i}}(x - c\Delta t, t) \right], \end{aligned} \quad (24)$$

where $\tilde{f}_{\tilde{i}}$ denotes a collided population. As per the discussion in the introduction section of this article, there are two ways

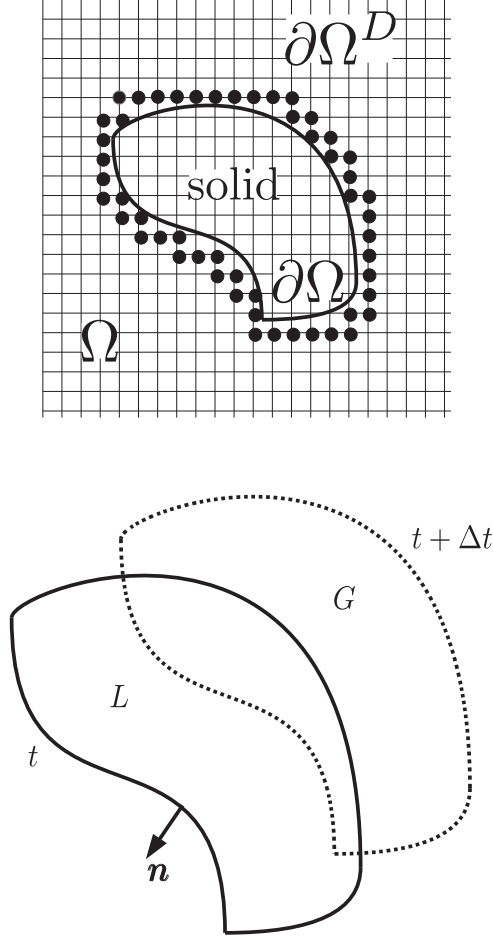


FIG. 6. Top: A solid object submerged in a fluid marked by Ω on a Cartesian grid. The circular dots represent the fluid nodes that make up the discrete boundary ($\partial\Omega^D$) analogous to the continuous solid boundary ($\partial\Omega$). Bottom: system evolving from t to $t + \Delta t$.

of calculating $J_\alpha^L(t)$ based on the choice of algorithm (A or B):

$$\begin{aligned} \text{algorithm A: } J_\alpha^L(t) &= \sum_{x \in \partial\Omega^D} \left[\sum_{i \ni c_i n_i < 0} c_{i\alpha} f_i(x, t) \right], \\ \text{algorithm B: } J_\alpha^L(t) &= \sum_{x \in \partial\Omega^D} \left[\sum_{i \ni c_i n_i < 0} c_{i\alpha} \tilde{f}_i(x, t) \right]. \end{aligned} \quad (25)$$

The choice of algorithm B simplifies Eq. (23) to give us the complete momentum balance for fluid nodes as

$$\begin{aligned} J_\alpha^\Omega(t + \Delta t) - J_\alpha^\Omega(t) &= \sum_{x \in \partial\Omega^D} \left[\sum_{i \ni c_i n_i > 0} c_{i\alpha} \tilde{f}_i(x - c\Delta t, t) \right] \\ &\quad - \sum_{x \in \partial\Omega^D} \left[\sum_{i \ni c_i n_i < 0} c_{i\alpha} \tilde{f}_i(x, t) \right]. \end{aligned} \quad (26)$$

The boundary surface contribution on RHS of Eq. (26) is exactly the momentum exchange algorithm proposed by Ladd [16], whereas the choice of algorithm A would lead to a slightly more complicated expression. Using Newton's

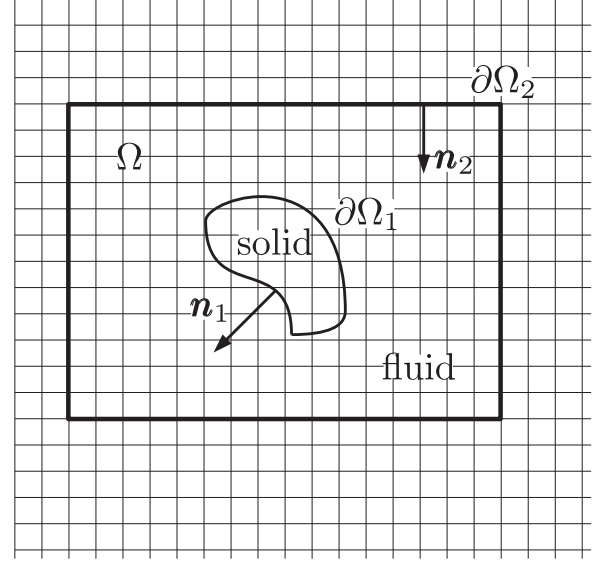


FIG. 7. A complex shaped solid object placed inside a control volume. The solid surface is marked by $\partial\Omega_1$ and the outer rectangular control volume surface is marked by $\partial\Omega_2$.

second law, one can estimate the forces exerted by the fluid on submerged solid objects as

$$\begin{aligned} F_\alpha(t) &= - \sum_{x \in \partial\Omega^D} \left[\sum_{i \ni c_i n_i > 0} c_{i\alpha} \tilde{f}_i(x - c\Delta t, t) \right] \\ &\quad + \sum_{x \in \partial\Omega^D} \left[\sum_{i \ni c_i n_i < 0} c_{i\alpha} \tilde{f}_i(x, t) \right]. \end{aligned} \quad (27)$$

The first term in the force formula is dictated by the boundary condition used in the algorithm. For a simple bounce-back boundary condition, the force formula simplifies to

$$F_\alpha(t) = 2 \sum_{x \in \partial\Omega^D} \left[\sum_{i \ni c_i n_i < 0} c_{i\alpha} \tilde{f}_i(x, t) \right]. \quad (28)$$

It is important to note that the derivation for the existing formulation assumes a semi-infinite control volume with only one control surface which is at the solid boundary. For all resolution bound numerical simulations, it is safe to assume that the control volume is actually bounded and not semi-infinite.

We extend the derivation carried out for a single control surface to a system with two surfaces, as given in Fig. 7, where the fluid nodes are given by Ω , and are sandwiched between the two surfaces $\partial\Omega_1$ and $\partial\Omega_2$, such that the momentum balance gives

$$\begin{aligned} J_\alpha^\Omega(t + \Delta t) - J_\alpha^\Omega(t) &= \sum_{x \in \partial\Omega_2} \left[\sum_{i \ni c_i n_{2i} > 0} c_{i\alpha} \tilde{f}_i(x - c\Delta t, t) \right] \\ &\quad - \sum_{x \in \partial\Omega_2} \left[\sum_{i \ni c_i n_{2i} < 0} c_{i\alpha} \tilde{f}_i(x, t) \right] \end{aligned}$$

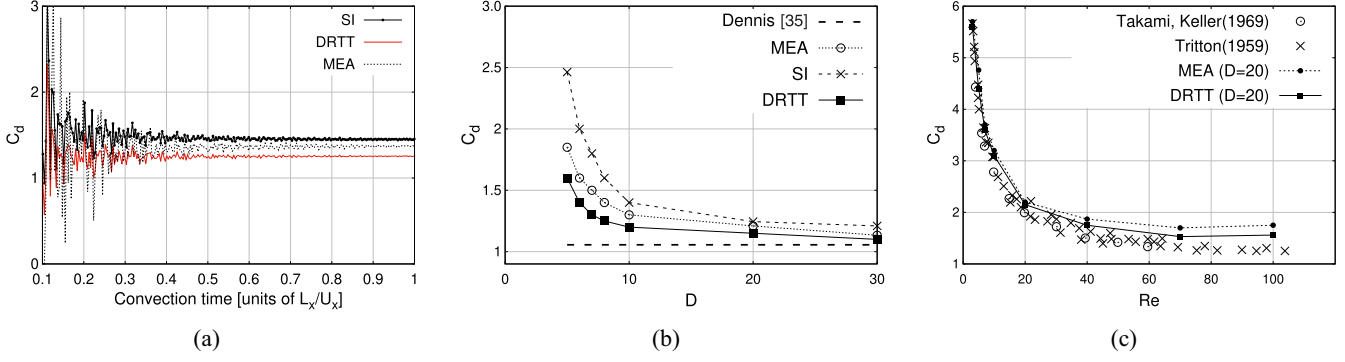


FIG. 8. The flow past a circular cylinder in two dimensions is a good bench-marking test case for lattice Boltzmann methods [34]. (a) depicts the time evolution of coefficient of drag at $Re = 100$ and $D = 20$, for MEA, SI, and DRTT. (b) depicts convergence, comparing the drag coefficient (C_d) as a function of number of grid point per diameter. (c) The computational domain size is kept fixed with $D = 20$, while the drag coefficient (C_d) is measured for several values of Reynolds numbers. The results of MEA and DRTT are then compared with literature.

$$\begin{aligned}
 & + \sum_{x \in \partial\Omega_1} \left[\sum_{i \ni c_i n_{1i} > 0} c_{i\alpha} \tilde{f}_i(x - c\Delta t, t) \right] \\
 & - \sum_{x \in \partial\Omega_1} \left[\sum_{i \ni c_i n_{1i} < 0} c_{i\alpha} \tilde{f}_i(x, t) \right]. \quad (29)
 \end{aligned}$$

The construction of the outer boundary $\partial\Omega_2$ is done in such a way that n_2 and the location of $\partial\Omega_2$ is trivial to resolve on a lattice. A rectangular bounding box aligned with the grid is a good example of the same. This construction helps us bypass calculations at the solid surface ($\partial\Omega_1$) by rearranging Eq. (30) to

$$\begin{aligned}
 F_\alpha(t) &= J_\alpha^\Omega(t) - J_\alpha^\Omega(t + \Delta t) \\
 & + \sum_{x \in \partial\Omega_2} \left[\sum_{i \ni c_i n_{2i} > 0} c_{i\alpha} \tilde{f}_i(x - c\Delta t, t) \right] \\
 & - \sum_{x \in \partial\Omega_2} \left[\sum_{i \ni c_i n_{2i} < 0} c_{i\alpha} \tilde{f}_i(x, t) \right], \quad (30)
 \end{aligned}$$

where $F_\alpha(t)$ is the force exerted by the fluid on the surface $\partial\Omega_1$. We label this force evaluation method as DRTT and conduct simulations in the next section to validate the method.

V. RESULTS

A. Flow past 2D circular cylinder

As a first example, we consider the flow past a circular cylinder in two dimensions, which is a regularly used benchmark case in computational fluid dynamics. For all the simulations in this section, we have used a simple bounce-back routine for the top and bottom boundary and Grad's approximation for the inlet and outlet populations [48]. The most basic calculation of the pressure coefficient on the surface of the cylinder involves calculation over boundary fluid nodes without any kind of extrapolation. The pressure coefficient at one boundary fluid node is given

by

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho U^2}, \quad (31)$$

where p_∞ is the far upstream pressure.

Figure 9 shows the coefficient of pressure vs θ for $Re = 100$ case. We see that, as expected, even at a low resolution of $D = 20$, the C_p curve is a close match with literature. Once the correctness of C_p is established, the flow force is calculated using all three methods discussed in the previous section. The drag coefficient over a circular cylinder of diameter D is given by

$$C_d = \frac{2|F_x|}{\rho U^2 D}, \quad (32)$$

where x is taken to be the direction along the flow. The force term (F_x) is calculated using popular methods (MEA and SI),

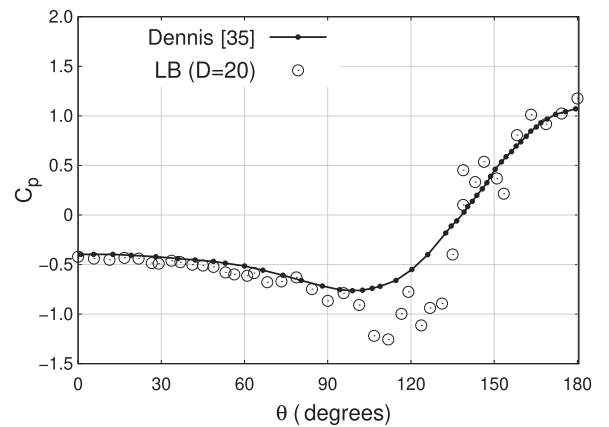


FIG. 9. Coefficient of pressure vs θ for flow past a two dimensional circular cylinder of diameter D solved using a D2Q9 model at $Re = 100$ using 20 grid points per diameter. The aspect ratio of the computational domain is taken to be 8 : 1 with $120D \times 15D$ points for the channel length and width, respectively. The cylinder center is symmetrically placed at $30D$ from the inlet. The lattice Boltzmann results are compared with established steady state drag values in literature [35].

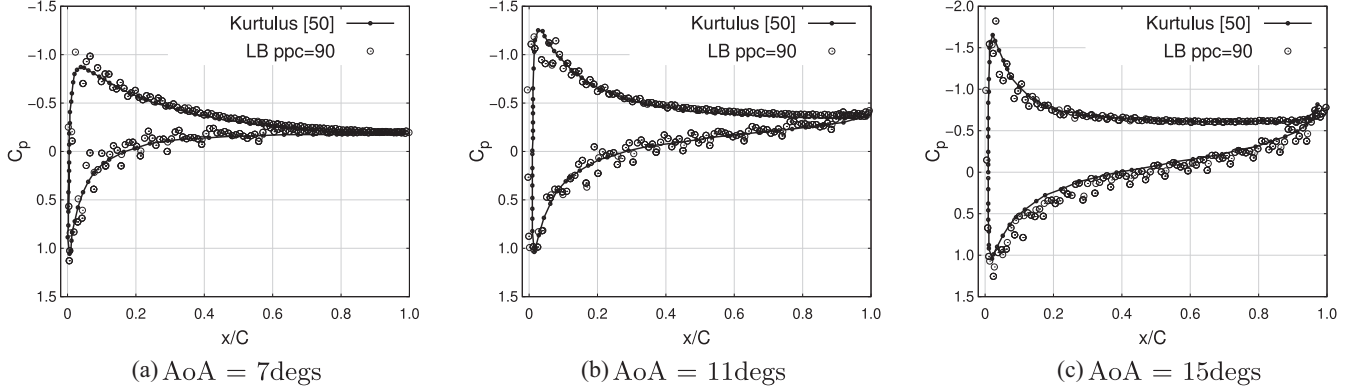


FIG. 10. Flow past a NACA0012 airfoil in two dimensions for Reynolds=1000 [37]. The coefficient of pressure (C_p) is calculated over the boundary fluid nodes for a computational domain size of 90 points per chord length. This is repeated for three distinct angles of attack ($\alpha = 7, 11, 15$)⁰ and the results are matched with literature [49].

along with the discrete Reynolds transport theorem (DRTT) algorithm. The two dimensional circular cylinder of diameter D is solved using a D2Q9 model at $Re = 100$ using 20 grid points per diameter. The aspect ratio of the computational domain is taken to be 8:1 with $120D \times 15D$ points for the channel length and width, respectively. The cylinder center is symmetrically placed at $30D$ from the inlet. We have plotted the C_d as a function of channel convection time (L_x/U_x) for all three methods in Fig. 8(a). The steady state value of C_d is calculated by taking a running average and is plotted in Figs. 8(b) and 8(c). The grid convergence at $Re = 100$ is plotted in Fig. 8(b). DRTT consistently does better than both SI and MEA at low and moderate resolutions. We see that SI does the worst among all the methods at a given resolution and henceforth, we shall not discuss it in the following sections on flow past a NACA airfoil. We ran simulations for a range of Reynolds numbers and plot steady state drag coefficients [see Fig. 8(c)] in order to study the dependence of the drag coefficient on Re . As seen in Fig. 8, DRTT does much better than MEA at a given resolution. One can also notice that the difference between DRTT and MEA becomes more prominent at high Re where the flow starts to become unsteady.

B. Flow past 2D NACA0012 airfoil ($Re = 1000$ benchmark)

The airfoil of interest in our work is the NACA0012 airfoil, which has been extensively used for validation cases for turbulence models. The NACA0012 airfoil is a skewed geometry with different grid requirements in x and y directions. This makes it an important case study for algorithms like lattice Boltzmann that use uniform meshes ($\Delta x = \Delta y$). In order to validate our code, we reviewed the coefficient of pressure (C_p) measured over the airfoil surface at a Reynolds number of 1000 and three angles of attack (α) [37]. Figure 10 shows us the match with literature. We plot for angle of attack = (7, 11, 15)⁰, respectively, and with a resolution of 90 points per chord.

The computational domain size is measured in the chord length (C) of the airfoil, such that the number of grid points per chord length is given by ppc . The aspect ratio of the rectangular box is taken to be 12:1. The length of the domain (in the direction of the flow) is taken to be $60 \times C$ with the

airfoil placed symmetrically at $15 \times C$ from the inlet. The lift coefficient (C_l) is given by

$$C_l = \frac{2|F_y|}{\rho U^2 D}. \quad (33)$$

For the next set of results, simulations are run for a Reynolds of 1000, but the angle of attack is varied starting from 0 and finishing at 30 degrees. The drag and lift coefficients are calculated using both the MEA and DRTT flow force calculation methods. Both these methods are carried out for a resolution of 30 points per chord and the results are given in Fig. 11 and Fig. 12, respectively.

We can see that MEA and DRTT do almost equally well at low angles of attack. But as the angle of attack rises and the flow begins to separate, DRTT is able to more accurately calculate the drag and lift values. We attribute this to the fact that DRTT, being a volumetric method, does not have to resolve the complex boundary of the airfoil at low and moderate grid resolution.

C. Flow past 3D NACA0012 airfoil

The final set of simulations for this study are for a three dimensional case of the NACA0012 airfoil. We conducted

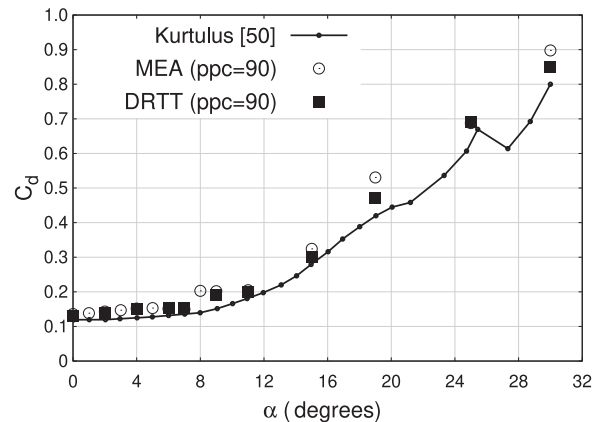


FIG. 11. Flow past a NACA0012 airfoil in two dimensions, using a D2Q9 lattice model and 90 points per chord length. The coefficient of drag (C_d) plotted against angle of attack (α) at $Re=1000$.

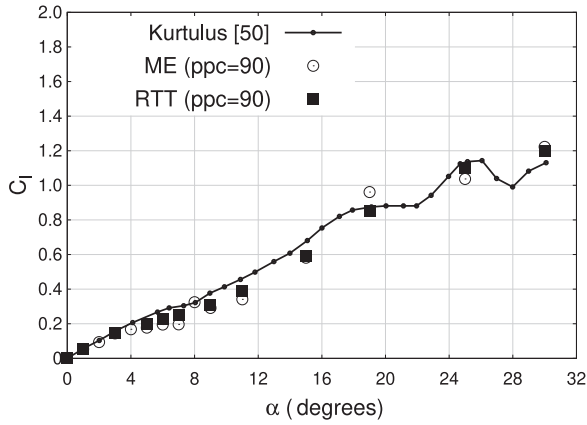


FIG. 12. Flow past a NACA0012 airfoil in two dimensions for Reynolds=1000. The coefficient of lift (C_l) plotted against angle of attack (α).

this at a fixed angle of attack of zero degrees but increased the Reynolds from a small value of 100 to 0.5 million. A 41 velocity lattice boltzmann model [38] was used for the three dimensional simulations. The goal of the study was to validate the DRTT approach for high Re values and also compare its behavior with MEA. The computational domain size is measured in the chord length of the airfoil (C). The number of grid points taken for $1C$ is given by ppc (points per chord length). The aspect ratio is fixed to 10:1 in the x and y directions. The number of points in the z direction was varied with Reynolds number in order to accommodate a developing flow in the z direction at very high Reynolds. The numerically converged steady state drag coefficient using DRTT is measured and compared with literature (see Fig. 13), and Table IV gives details of the computational grid requirements for a converged result. The resolution demands (measured in points per chord) for a converged result vary as a function of Reynolds number. We assign an error percentage of two percent in order to call a result converged and tabulate the results in Table IV. The converged C_d vs Re curve as seen in Fig. 13 shows a good match with literature. Figure 14 shows a close match with a

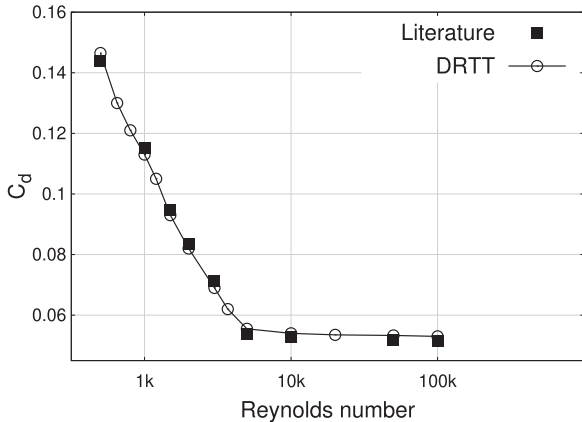


FIG. 13. The coefficient of drag from a three dimensional simulation of NACA0012 using RD3Q41 lattice Boltzmann model. Converged results have been plotted.

TABLE IV. The list of grid size requirements for a three dimensional NACA0012 simulation using a RD3Q41 model. The aspect ratio of the computational domain is taken to be 10:1 with the length of the channel being $10 \times$ points per chord.

Re	Points per chord length	Points in z direction	%error in C_d (MEA)	%error in C_d (DRTT)
1.0×10^3	128	12	1.75	1.25
1.0×10^4	256	12	2.67	1.74
0.1×10^6	960	24	3.1	1.97
0.2×10^6	1536	48	3.22	1.87
0.4×10^6	2816	48	5.16	1.95
0.5×10^6	3072	60	9.21	2.06

$Re^{1/2}$ scaling which can be explained using boundary layer theory.

VI. CONCLUSIONS

In this work we revisited the force evaluation methodologies in the lattice Boltzmann method. The first half of the work emphasized on the noncommutativity of the streaming and collision operations when dealing with flows around solid objects. This breakdown of commutativity opens up two possibilities for force evaluation, as shown in Fig. 1, the details of which have not been fully clarified in literature. Subsequent theoretical and computational analysis showed the superiority of algorithm B over algorithm A. In the process of establishing the aforementioned claim, the authors have suggested a simple and elegant force evaluation routine called DRTT. The elegance of DRTT lies in its compatibility with Cartesian grid based methods, like lattice Boltzmann, where the accurate resolution of complex shaped geometries is a major issue. The DRTT routine is compared with the extensively used momentum-exchange method for a variety of flow problems, including flow past a two-dimensional cylinder and airfoil (NACA0012), and flow past a three-dimensional NACA0012 airfoil.

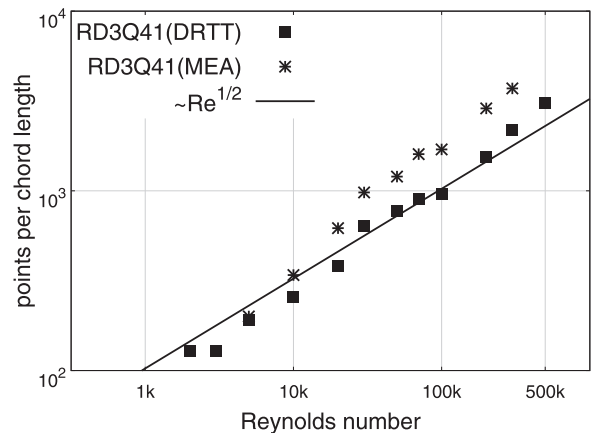


FIG. 14. Total number of computational points required to correctly measure the drag values for NACA0012 using the DRTT approach. The x axis is the Reynolds number and the y axis is the points per chord length of NACA0012.

ACKNOWLEDGMENTS

The authors are thankful to the National Supercomputing Mission (NSM) for providing the computational facility for simulations. S.S. wishes to acknowledge funding from the Eu-

ropean Research Council -PoC Droptack (Fast and automated droplet tracking tool for microfluidics, No. 101081171). The authors also wish to acknowledge the valuable inputs from P.K. Kolluru and C. Thantapanally from SankhyaSutra Labs, India.

-
- [1] S. Chen and G. D. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* **30**, 329 (1998).
- [2] R. Benzi, S. Succi, and M. Vergassola, The lattice Boltzmann equation: Theory and applications, *Phys. Rep.* **222**, 145 (1992).
- [3] S. Succi, *The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond* (Oxford university press, Oxford, 2001).
- [4] S. Ansumali, I. V. Karlin, S. Arcidiacono, A. Abbas, and N. Prasianakis, Hydrodynamics beyond Navier-Stokes: Exact solution to the lattice Boltzmann hierarchy, *Phys. Rev. Lett.* **98**, 124502 (2007).
- [5] C. Thantapanally, D. V. Patil, S. Succi, and S. Ansumali, Universal mechanism for saturation of vorticity growth in fully developed fluid turbulence, *J. Fluid Mech.* **728**, R4 (2013).
- [6] S. Singh, G. Subramanian, and S. Ansumali, A lattice Boltzmann method for dilute polymer solutions, *Philos. Trans. R. Soc. A* **369**, 2301 (2011).
- [7] S. P. Thampi, S. Ansumali, R. Adhikari, and S. Succi, Isotropic discrete Laplacian operators from lattice hydrodynamics, *J. Comput. Phys.* **234**, 1 (2013).
- [8] S. Succi and S. Succi, *The Lattice Boltzmann Equation: For Complex States of Flowing Matter* (Oxford University Press, 2018).
- [9] C. K. Aidun and J. R. Clausen, Lattice-Boltzmann method for complex flows, *Annu. Rev. Fluid Mech.* **42**, 439 (2010).
- [10] F. Higuera, S. Succi, and R. Benzi, Lattice gas dynamics with enhanced collisions, *Europhys. Lett.* **9**, 345 (1989).
- [11] F. Higuera and S. Succi, Simulating the flow around a circular cylinder with a lattice Boltzmann equation, *Europhys. Lett.* **8**, 517 (1989).
- [12] S. Ansumali and I. V. Karlin, Kinetic boundary conditions in the lattice Boltzmann method, *Phys. Rev. E* **66**, 026311 (2002).
- [13] H. Li, X. Lu, H. Fang, and Y. Qian, Force evaluations in lattice Boltzmann simulations with moving boundaries in two dimensions, *Phys. Rev. E* **70**, 026701 (2004).
- [14] O. Filippova and D. Hänel, Grid refinement for lattice-BGK models, *J. Comput. Phys.* **147**, 219 (1998).
- [15] T. Inamuro, K. Maeba, and F. Ogino, Flow between parallel walls containing the lines of neutrally buoyant circular cylinders, *Int. J. Multiphase Flow* **26**, 1981 (2000).
- [16] A. J. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation, *J. Fluid Mech.* **271**, 285 (1994).
- [17] R. Mei, D. Yu, W. Shyy, and L.-S. Luo, Force evaluation in the lattice Boltzmann method involving curved geometry, *Phys. Rev. E* **65**, 041203 (2002).
- [18] C. S. Peskin, The immersed boundary method, *Acta Numerica* **11**, 479 (2002).
- [19] S. Majumder, A. Ghosh, D. N. Basu, and G. Natarajan, Computational assessment of immersed boundary–lattice Boltzmann method for complex moving boundary problems, *Comput. Part. Mech.* **10**, 155 (2022).
- [20] E. Adeb, B. A. Haider, and C. H. Sohn, Flow interference of two side-by-side square cylinders using IB-LBM—effect of corner radius, *Results Phys.* **10**, 256 (2018).
- [21] D. Noble and J. Torczynski, A lattice-Boltzmann method for partially saturated computational cells, *Int. J. Mod. Phys. C* **09**, 1189 (1998).
- [22] O. E. Strack and B. K. Cook, Three-dimensional immersed boundary conditions for moving solids in the lattice-Boltzmann method, *Int. J. Numer. Methods Fluids* **55**, 103 (2007).
- [23] G. Silva and I. Ginzburg, Reviving the local second-order boundary approach within the two-relaxation-time lattice Boltzmann modelling, *Phil. Trans. R. Soc. A* **378**, 20190404 (2020).
- [24] R. Mei, W. Shyy, D. Yu, and L.-S. Luo, Lattice Boltzmann method for 3-d flows with curved boundary, *J. Comput. Phys.* **161**, 680 (2000).
- [25] J. Latt, B. Chopard, O. Malaspinas, M. Deville, and A. Michler, Straight velocity boundaries in the lattice Boltzmann method, *Phys. Rev. E* **77**, 056703 (2008).
- [26] O. Malaspinas, B. Chopard, and J. Latt, General regularized boundary condition for multi-speed lattice Boltzmann models, *Comput. Fluids* **49**, 29 (2011).
- [27] S. Mohammed and T. Reis, Using the lid-driven cavity flow to validate moment-based boundary conditions for the lattice Boltzmann equation, *Arch. Mech. Eng.* **64**, 57 (2017).
- [28] L. A. Hegele, A. Scagliarini, M. Sbragaglia, K. K. Mattila, P. C. Philippi, D. F. Puleri, J. Gounley, and A. Randles, High-Reynolds-number turbulent cavity flow using the lattice Boltzmann method, *Phys. Rev. E* **98**, 043302 (2018).
- [29] R. L. M. Bazarin Jr, P. Philippi, A. Randles, and L. Hegele Jr, Moments-based method for boundary conditions in the lattice Boltzmann framework: A comparative analysis for the lid driven cavity flow, *Comput. Fluids* **230**, 105142 (2021).
- [30] A. J. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results, *J. Fluid Mech.* **271**, 311 (1994).
- [31] P. J. Dellar, An interpretation and derivation of the lattice Boltzmann method using strang splitting, *Comput. Math. Appl.* **65**, 129 (2013).
- [32] S. Ansumali, S. Arcidiacono, S. Chikatamarla, N. Prasianakis, A. Gorban, and I. Karlin, Quasi-equilibrium lattice Boltzmann method, *Eur. Phys. J. B* **56**, 135 (2007).
- [33] A. Caiazzo and M. Junk, Boundary forces in lattice Boltzmann: Analysis of momentum exchange algorithm, *Comput. Math. Appl.* **55**, 1415 (2008).
- [34] S. Succi, E. Foti, and F. Higuera, Three-dimensional flows in complex geometries with the lattice Boltzmann method, *Europhys. Lett.* **10**, 433 (1989).
- [35] S. Dennis and G.-Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, *J. Fluid Mech.* **42**, 471 (1970).

- [36] G. Warren, W. Anderson, J. Thomas, and S. Krist, Grid convergence for adaptive methods, in *10th Computational Fluid Dynamics Conference* (1991), p. 1592.
- [37] G. Di Ilio, D. Chiappini, S. Ubertini, G. Bella, and S. Succi, Fluid flow around NACA 0012 airfoil at low-Reynolds numbers with hybrid lattice Boltzmann method, *Comput. Fluids* **166**, 200 (2018).
- [38] P. K. Kolluru, M. Atif, M. Namburi, and S. Ansumali, Lattice Boltzmann model for weakly compressible flows, *Phys. Rev. E* **101**, 013309 (2020).
- [39] M. Atif, M. Namburi, and S. Ansumali, Higher-order lattice Boltzmann model for thermohydrodynamics, *Phys. Rev. E* **98**, 053311 (2018).
- [40] C. K. Aidun, Y. Lu, and E.-J. Ding, Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation, *J. Fluid Mech.* **373**, 287 (1998).
- [41] D. R. Noble, S. Chen, J. G. Georgiadis, and R. O. Buckius, A consistent hydrodynamic boundary condition for the lattice Boltzmann method, *Phys. Fluids* **7**, 203 (1995).
- [42] Y. Shi, P. L. Brookes, Y. W. Yap, and J. E. Sader, Accuracy of the lattice Boltzmann method for low-speed noncontinuum flows, *Phys. Rev. E* **83**, 045701(R) (2011).
- [43] S. Krithivasan, S. Wahal, and S. Ansumali, Diffused bounce-back condition and refill algorithm for the lattice Boltzmann method, *Phys. Rev. E* **89**, 033313 (2014).
- [44] G. Liu and T. Lee, Diffuse bounce back condition for lattice Boltzmann method, *Comput. Fluids* **220**, 104884 (2021).
- [45] Y. Chen, Q. Cai, Z. Xia, M. Wang, and S. Chen, Momentum-exchange method in lattice Boltzmann simulations of particle-fluid interactions, *Phys. Rev. E* **88**, 013303 (2013).
- [46] B. Wen, C. Zhang, and H. Fang, Hydrodynamic force evaluation by momentum exchange method in lattice Boltzmann simulations, *Entropy* **17**, 8240 (2015).
- [47] J. P. Giovacchini and O. E. Ortiz, Flow force and torque on submerged bodies in lattice-Boltzmann methods via momentum exchange, *Phys. Rev. E* **92**, 063302 (2015).
- [48] S. Chikatamarla, S. Ansumali, and I. Karlin, Grad's approximation for missing data in lattice Boltzmann simulations, *Europhys. Lett.* **74**, 215 (2006).
- [49] D. F. Kurtulus, On the unsteady behavior of the flow around NACA 0012 airfoil with steady external conditions at $Re = 1000$, *Int. J. Micro Air Veh.* **7**, 301 (2015).