Meshless lattice Boltzmann method for the simulation of fluid flows

S. Hossein Musavi[*] and Mahmud Ashrafizaadeh

Department of Mechanical Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

(Received 27 April 2014; revised manuscript received 7 January 2015; published 17 February 2015)

A meshless lattice Boltzmann numerical method is proposed. The collision and streaming operators of the lattice Boltzmann equation are separated, as in the usual lattice Boltzmann models. While the purely local collision equation remains the same, we rewrite the streaming equation as a pure advection equation and discretize the resulting partial differential equation using the Lax-Wendroff scheme in time and the meshless local Petrov-Galerkin scheme based on augmented radial basis functions in space. The meshless feature of the proposed method makes it a more powerful lattice Boltzmann solver, especially for cases in which using meshes introduces significant numerical errors into the solution, or when improving the mesh quality is a complex and time-consuming process. Three well-known benchmark fluid flow problems, namely the plane Couette flow, the circular Couette flow, and the impulsively started cylinder flow, are simulated for the validation of the proposed method. Excellent agreement with analytical solutions or with previous experimental and numerical results in the literature is observed in all the simulations. Although the computational resources required for the meshless method per node are higher compared to that of the standard lattice Boltzmann method, it is shown that for cases in which the total number of nodes is significantly reduced, the present method actually outperforms the standard lattice Boltzmann method.

DOI: 10.1103/PhysRevE.91.023310 PACS number(s): 47.11.−j

I. INTRODUCTION

The lattice Boltzmann method (LBM) was developed in the late 1980s [1–4] and the early 1990s [5–7]. Since then, it has been applied in simulations of different fluid flow problems [8,9]. The lattice Boltzmann method is a significant alternative for computational fluid dynamics applications due to the following reasons: (i) The method is simple, especially in implementing on a computer code; (ii) it has a local nature, which results in a high capability for parallel processing; (iii) it can better represent microscopic interactions, which makes the method successful in interfacial dynamics applications; and (iv) it is not limited to the continuum assumption, which enables the lattice Boltzmann method to simulate higher Knudsen number flow regimes compared to the Navier-Stokes equations solvers.

Although at first the lattice Boltzmann method was considered to be an extension of the lattice gas automata [1], it was shown later by McNamara, Garcia, and Alder [10], and then by Sterling and Chen [11] and He and Luo [12], that the lattice Boltzmann equation is a special discretization of the discrete Boltzmann equation along characteristics, so that the streaming equation becomes the exact solution of the linear advection equation, the so-called perfect shift. As a result, the velocity and the physical space discretizations are coupled, which leads to the restriction of using uniform structured grids in the standard lattice Boltzmann method. This aspect of the lattice Boltzmann method, however, creates some difficulties with regard to accurate treatment of curved or irregular boundaries, especially when trying to satisfy mass and momentum conservation on the boundary.

Several studies in the literature have been dedicated to the treatment of this drawback of the lattice Boltzmann method, and they have led to some extensions of the method to cover nonuniform or unstructured grids. Historically, these efforts can be traced back to the early stages of developing the lattice Boltzmann method [7]. Succi and co-workers [7,13] were apparently the first to extend the lattice Boltzmann method to irregular lattice geometries by suggesting a finite volume formulation for the lattice Boltzmann equation. He *et al.* [14,15] proposed an interpolation-supplemented lattice Boltzmann method by adding an interpolation step to the two collision and streaming steps of the standard lattice Boltzmann method to compute the distribution function values on the nonuniform mesh grids from the lattice sites. Interpolating in every time step makes this method computationally inefficient, especially for three-dimensional (3D) problems, where 27 points are required for the efficient interpolation of each point. Furthermore, negative particle distributions may occur under certain conditions [16]. Shu, Niu, and Chew [17] used a Taylor-series expansion and least-squares-based lattice Boltzmann formulation instead of direct interpolation. Although their method eliminates the need for interpolating in every time step, and it may be considered to have a meshless feature, it still requires geometrical relations between the grid points and the lattice sites, therefore they recommended using structured grids [17]. Filippova and Hänel [18] introduced the grid refinement technique to the lattice Boltzmann simulations. By using a hierarchical second-order grid refinement scheme, in addition to attaining higher computational efficiency, they could treat curved boundaries more accurately. Dupuis and Chopard [19] proposed a simpler grid refinement algorithm that could significantly speed up the flow settlement process. Although the grid refinement algorithms increase the efficiency of the lattice Boltzmann method, particularly in simulating strongly anisotropic flows, e.g., high Reynolds number boundary layer flows, they still have the basic problem of the standard lattice Boltzmann method, namely restriction to the uniform Cartesian grid.

A natural conclusion of the second interpretation of the lattice Boltzmann equation is that for the Boltzmann equation, the

[*] hmusavi@me.iut.ac.ir

velocity space and the physical space can also be discretized independently. As a result, different numerical schemes, such as the finite difference (FD), finite volume (FV), finite element (FE), and discontinuous Galerkin (DG) methods, have been used for the discretization of the physical space in the discrete Boltzmann equation, which leads to introducing FD-LBM [20–23], FV-LBM [24–27], FE-LBM [28–30], and DG-LBM [31–33] methods, respectively. The primary advantage of these methods is to make the lattice Boltzmann method capable of treating curved or irregular boundaries more accurately by using nonuniform or unstructured body-fitted meshes.

Using the aforementioned numerical schemes with the lattice Boltzmann method supplies the method with all the flexibilities of these schemes at the cost of some extra computational effort. However, the accuracy of all these methods depends on the mesh quality. Creating good-quality meshes for domains with complex geometries can be a time-consuming challenge, especially in three-dimensional cases. Furthermore, for problems in which remeshing during simulation is necessary, the process may become prohibitively expensive and can introduce some error in the results.

Meshless methods are a wide variety of numerical schemes aimed at eliminating the need for mesh in representing computational domains. The main advantage of these methods is that they eliminate the time and the cost of creating good-quality meshes, especially in the aforementioned problems. However, this comes at the cost of consuming more memory, as well as additional computational time. In addition, meshless methods are typically more sensitive to point distribution, a feature that can be efficiently treated by applying sensible point distribution and choosing appropriate sizes for integration domains and for local support domains, based on local average point spacing. There have been several studies in the past decades developing new meshless formulations for applications in different branches of physics, such as astrophysics, fluid and solid mechanics, heat transfer, detonation, etc. [34–36].

In this study, we present a meshless lattice Boltzmann method by making use of the meshless local Petrov-Galerkin (MLPG) formulation based on augmented radial basis functions (RBFs). As in the standard lattice Boltzmann method, we split the lattice Boltzmann equation into the two steps of collision and streaming. The collision equation is a strictly local equation to be solved like the standard LBM. The streaming equation, which in the standard LBM is the exact solution of the pure advection equation, is rewritten as the pure advection equation. The resulting partial differential equation is discretized in time using the Lax-Wendroff scheme. The consequent semidiscrete equation is then recast into its local weak form and is discretized by implementing the meshless local Petrov-Galerkin method based on augmented radial basis functions. The boundary conditions are imposed using the bounce-back scheme of the nonequilibrium parts of distribution functions. The validity of our proposed method is investigated through solving a number of benchmark problems, namely the plane Couette flow, the circular Couette flow, and the impulsively started cylinder flow. One should note that these benchmarks are performed only to present the validity and correctness of the proposed method. However, the computational advantages of this method becomes apparent for cases with complex geometries, which is the next step in our ongoing research.

II. FORMULATION AND NUMERICAL PROCEDURE

In this section, first the formulation of the lattice Boltzmann method is given, and then the process of discretizing the streaming equation is described.

A. Lattice Boltzmann equation

The discrete Boltzmann equation with the BGK collision approximation [37] is

$$\frac{\partial f_i}{\partial t} + c_{i,\alpha}\frac{\partial f_i}{\partial x_\alpha} = -\frac{1}{\lambda}\left(f_i - f_i^{\mathrm{eq}}\right), \quad i = 1,\dots,nQ, \tag{1}$$

where f_i is the particle distribution function, $\mathbf{c}_i$ is the discrete microscopic velocity, and f_i^{eq} is the equilibrium distribution, all along direction i, λ is the relaxation time toward equilibrium, and nQ is the number of discrete microscopic velocities. It should be noted that in our notation, index i is a free index showing the specific lattice velocity direction, and all Greek indices, such as α in the above equation, are dummy indices representing space coordinate directions, for which Einstein's rule of summation is applied.

In this study, we only consider two-dimensional cases, and therefore we use the more common D2Q9 lattice given as

$$\begin{aligned}
&\mathbf{c}_0 = \mathbf{0},\\
&\mathbf{c}_i = \cos(i-1)\frac{\pi}{4}\mathbf{e}_x + \sin(i-1)\frac{\pi}{4}\mathbf{e}_y, \quad i = 1,3,5,7,\\
&\mathbf{c}_i = \sqrt{2}\left[\cos(i-1)\frac{\pi}{4}\mathbf{e}_x + \sin(i-1)\frac{\pi}{4}\mathbf{e}_y\right], \quad i = 2,4,6,8,
\end{aligned} \tag{2}$$

where $\mathbf{e}_x$ and $\mathbf{e}_y$ are the unit vectors along the x and y directions, respectively. The equilibrium distribution function is

$$f_i^{\mathrm{eq}} = \rho t_i\left(1 + \frac{c_{i,\alpha}u_\alpha}{c_s^2} + \frac{(c_{i,\alpha}u_\alpha)^2}{2c_s^4} - \frac{u_\alpha^2}{2c_s^2}\right), \tag{3}$$

where ρ and u_α are the macroscopic density and velocity, respectively, $c_s = 1/\sqrt{3}$ is the lattice speed of sound, and t_i is the lattice weighting factor for direction i, where $t_0 = 4/9$, $t_{1,3,5,7} = 1/9$, and $t_{2,4,6,8} = 1/36$.

By integrating Eq. (1) along the characteristics over the time step δt, we have

$$f_i(\mathbf{x}+\mathbf{c}_i\delta t, t+\delta t) - f_i(\mathbf{x},t) = -\frac{1}{\lambda}\int_t^{t+\delta t}\left(f_i - f_i^{\mathrm{eq}}\right)dt'. \tag{4}$$

For the collision operator to be of second order, we use the trapezoidal rule for the integration of the right-hand side of Eq. (4) to obtain

$$\begin{aligned}
&f_i(\mathbf{x}+\mathbf{c}_i\delta t, t+\delta t) - f_i(\mathbf{x},t)\\
&\quad = -\frac{1}{2\tau}\left[\left(f_i - f_i^{\mathrm{eq}}\right)\big|_{(\mathbf{x}+\mathbf{c}_i\delta t, t+\delta t)} + \left(f_i - f_i^{\mathrm{eq}}\right)\big|_{(\mathbf{x},t)}\right],
\end{aligned} \tag{5}$$

where $\tau = \lambda/\delta t$ is the dimensionless relaxation time. Equation (5) is an implicitly coupled equation, which could be transformed into an explicit lattice Boltzmann equation using a modified distribution function $\bar{f}_i$ and its corresponding equilibrium distribution $\bar{f}_i^{\mathrm{eq}}$ introduced as [38]

$$\bar{f}_i = f_i + \frac{1}{2\tau}\left(f_i - f_i^{\mathrm{eq}}\right) \quad \text{and} \quad \bar{f}_i^{\mathrm{eq}} = f_i^{\mathrm{eq}}. \tag{6}$$

By implementing this transformation, we arrive at the explicit lattice Boltzmann equation as

$$\bar{f}_i(\mathbf{x} + \mathbf{c}_i\delta t, t + \delta t) - \bar{f}_i(\mathbf{x},t) = -\frac{1}{\tau + 1/2}\left(\bar{f}_i - \bar{f}_i^{\mathrm{eq}}\right)\big|_{(\mathbf{x},t)}. \tag{7}$$

It is common to solve the lattice Boltzmann equation (7) in two steps, namely the collision,

$$\tilde{f}_i(\mathbf{x},t) = \bar{f}_i(\mathbf{x},t) - \frac{1}{\tau + 1/2}\left(\bar{f}_i - \bar{f}_i^{\mathrm{eq}}\right)\big|_{(\mathbf{x},t)}, \tag{8}$$

and the streaming,

$$\bar{f}_i(\mathbf{x} + \mathbf{c}_i\delta t, t + \delta t) = \tilde{f}_i(\mathbf{x},t). \tag{9}$$

Equation (9) is the exact solution of a pure advection equation and has a Lagrangian framework. However, it restricts the standard LBM to uniform structured meshes with unit CFL number. One way to overcome these deficiencies of the standard LBM is to express the streaming equation in terms of a pure advection equation to be considered in an Eulerian framework, that is,

$$\frac{\partial \bar{f}_i}{\partial t} + c_{i,\alpha}\frac{\partial \bar{f}_i}{\partial x_\alpha} = 0. \tag{10}$$

We discretize Eq. (10) in the next section.

The macroscopic quantities—density and velocity—are computed from the moments of the distribution function as

$$\rho = \sum_{i=0}^{b} \bar{f}_i, \tag{11}$$

$$\rho u_\alpha = \sum_{i=0}^{b} c_{i,\alpha}\bar{f}_i. \tag{12}$$

The equation of state for an ideal gas is applied to relate the pressure to the density as follows:

$$p = \rho c_s^2. \tag{13}$$

It can be demonstrated by the Chapman-Enskog asymptotic analysis that the kinematic viscosity is related to the dimensionless relaxation time as [39]

$$\nu = \tau c_s^2 \delta t. \tag{14}$$

B. Discretization of the advection equation

There are several numerical techniques to treat Eq. (10). In this study, we use the Lax-Wendroff scheme to discretize Eq. (10) in time, and the meshless local Petrov-Galerkin scheme to discretize the resulting semidiscrete equation in space. For brevity, we omit the bar symbol of the modified distribution hereafter.

1. Time discretization: Lax-Wendroff scheme

To discretize Eq. (10) in time using the Lax-Wendroff scheme, we write the Taylor series expansion of the particle distributions with respect to time:

$$f_i^{n+1} = f_i^n + \delta t \left.\frac{\partial f_i}{\partial t}\right|^n + \frac{\delta t^2}{2}\left.\frac{\partial^2 f_i}{\partial t^2}\right|^n + O(\delta t^3), \tag{15}$$

where the superscript n represents the time step number. The time derivatives in the above equation can be replaced by the spacial derivatives using Eq. (10) as

$$\begin{aligned} \left.\frac{\partial f_i}{\partial t}\right|^n &= -c_{i,\alpha}\frac{\partial f_i^n}{\partial x_\alpha}, \\ \left.\frac{\partial^2 f_i}{\partial t^2}\right|^n &= c_{i,\alpha}c_{i,\beta}\frac{\partial^2 f_i^n}{\partial x_\alpha \partial x_\beta}. \end{aligned} \tag{16}$$

Substituting Eqs. (16) into Eq. (15) and eliminating the third-order terms, we have

$$f_i^{n+1} = f_i^n - \delta t c_{i,\alpha}\frac{\partial f_i^n}{\partial x_\alpha} + \frac{\delta t^2}{2}c_{i,\alpha}c_{i,\beta}\frac{\partial^2 f_i^n}{\partial x_\alpha \partial x_\beta}. \tag{17}$$

2. Space discretization: Meshless local Petrov-Galerkin scheme

In the meshless local Petrov-Galerkin method, an arbitrary collection of nodal points is distributed in the computational domain. Subsequently, a local domain, called control volume, is considered around each of the points. These control volumes could have different sizes. There could also be intersections, overlaps, or even gaps between these control volumes. For simplicity of integrations, simple shapes, such as the circle or the rectangle in two-dimensional domains, and the sphere or the cube in three-dimensional domains, are selected for the control volumes.

To apply the meshless local Petrov-Galerkin scheme to discretize Eq. (17) in space, first the local weak form of Eq. (17) on the control volume Ω_I of point I is derived by taking its inner product with a local test function W_I over Ω_I, that is,

$$\begin{aligned} \int_{\Omega_I} W_I f_i^{n+1} d\Omega = \int_{\Omega_I} W_I f_i^n d\Omega - \delta t \int_{\Omega_I} W_I c_{i,\alpha}\frac{\partial f_i^n}{\partial x_\alpha} d\Omega \\ + \frac{\delta t^2}{2}\int_{\Omega_I} W_I c_{i,\alpha}c_{i,\beta}\frac{\partial^2 f_i^n}{\partial x_\alpha \partial x_\beta} d\Omega. \end{aligned} \tag{18}$$

The last integral of Eq. (18) contains second-order derivatives, which are reduced to first-order ones using integration by parts, so that we obtain

$$\int_{\Omega_I} W_I f_i^{n+1} d\Omega = \int_{\Omega_I} W_I f_i^n d\Omega - \int_{\Omega_I}\left(\delta t W_I c_{i,\alpha}\frac{\partial f_i^n}{\partial x_\alpha} + \frac{\delta t^2}{2}c_{i,\alpha}c_{i,\beta}\frac{\partial W_I}{\partial x_\beta}\frac{\partial f_i^n}{\partial x_\alpha}\right)d\Omega + \frac{\delta t^2}{2}\int_{\Gamma_I} W_I c_{i,\alpha}c_{i,\beta}\frac{\partial f_i^n}{\partial x_\alpha}n_\beta d\Gamma, \tag{19}$$

where Γ_I is the boundary of the control volume Ω_I and n_β is the unit outward normal vector of Γ_I. Equation (19) is the local weak form of Eq. (17).

In the next step, the field variable f_i is to be expressed in terms of nodal values $f_{i,J}$ by a local interpolation scheme, that is,

$$f_i(\mathbf{x},t) = \sum_{J=1}^{N_s} \phi_J(\mathbf{x}) f_{i,J}(t) = \Phi^{\mathrm{T}}(\mathbf{x})\mathbf{f}_s(t), \tag{20}$$

where N_s is the number of nodal points in a local interpolation domain of point $\mathbf{x}$ called a support domain, $\Phi^{\mathrm{T}}(\mathbf{x}) = \{\phi_1(\mathbf{x}),\phi_2(\mathbf{x}),\ldots,\phi_{N_s}(\mathbf{x})\}$ is the transpose of the vector of shape functions, and $\mathbf{f}_s(t) = \{f_{i,1}(t), f_{i,2}(t),\ldots, f_{i,N_s}(t)\}^{\mathrm{T}}$ is the vector of the nodal values of f_i in the support domain. In this study, we make use of the local radial point interpolation method (LRPIM) [35], which uses the local radial functions augmented with polynomials as the basis function; thus $\Phi^{\mathrm{T}}(\mathbf{x})$ in the interpolation equation (20) is the transpose of the vector of LRPIM shape functions given as [35]

$$\tilde{\Phi}^{\mathrm{T}}(\mathbf{x}) = \{\mathbf{R}^{\mathrm{T}}(\mathbf{x})\ \mathbf{p}^{\mathrm{T}}(\mathbf{x})\}\mathbf{G}^{-1}, \tag{21}$$

where $\mathbf{R}^{\mathrm{T}}(\mathbf{x}) = \{R_1(\mathbf{x}), R_2(\mathbf{x}),\ldots,R_{N_s}(\mathbf{x})\}$ is the transpose of the vector of radial basis functions (RBF), $\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \{1,x,y,\ldots,p_m(\mathbf{x})\}$ is the transpose of the vector of monomial basis functions, m is the number of monomial basis functions, $\tilde{\Phi}^{\mathrm{T}}(\mathbf{x}) = \{\Phi^{\mathrm{T}}(\mathbf{x}),\phi_{N_s+1}(\mathbf{x}),\ldots,\phi_{N_s+m}(\mathbf{x})\}$ is the transpose of the extended vector of the shape functions, and $\mathbf{G}$ is a symmetric matrix defined as

$$\mathbf{G} = \begin{pmatrix} R_1(\mathbf{x}_1) & R_2(\mathbf{x}_1) & \cdots & R_{N_s}(\mathbf{x}_1) & 1 & x_1 & y_1 & \cdots & p_m(\mathbf{x}_1) \\ R_1(\mathbf{x}_2) & R_2(\mathbf{x}_2) & \cdots & R_{N_s}(\mathbf{x}_2) & 1 & x_2 & y_2 & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ R_1(\mathbf{x}_{N_s}) & R_2(\mathbf{x}_{N_s}) & \cdots & R_{N_s}(\mathbf{x}_{N_s}) & 1 & x_{N_s} & y_{N_s} & \cdots & p_m(\mathbf{x}_{N_s}) \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 \\ x_1 & x_2 & \cdots & x_{N_s} & 0 & 0 & 0 & \cdots & 0 \\ y_1 & y_2 & \cdots & y_{N_s} & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_m(\mathbf{x}_1) & p_m(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_{N_s}) & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{22}$$

Substituting Eq. (20) in Eq. (19) and doing some algebra, we can write formally

$$\begin{aligned} \sum_{J=1}^{N_I}\left[\int_{\Omega_I} W_I\phi_J d\Omega\right] f_{i,J}^{n+1} = \sum_{J=1}^{N_I}\Bigg[\int_{\Omega_I} W_I\phi_J d\Omega - \int_{\Omega_I}\left(\delta t W_I + \frac{\delta t^2}{2}c_{i,\beta}\frac{\partial W_I}{\partial x_\beta}\right)c_{i,\alpha}\frac{\partial \phi_J}{\partial x_\alpha}d\Omega \\ + \frac{\delta t^2}{2}\int_{\Gamma_I} W_I c_{i,\alpha}c_{i,\beta}\frac{\partial\phi_J}{\partial x_\alpha}n_\beta d\Gamma\Bigg] f_{i,J}^{n}, \end{aligned} \tag{23}$$

where N_I is the number of nodal points involved in the interpolation of the field variable on the inner and the boundary points of the control volume Ω_I. By introducing the mass matrix as

$$M_{IJ} = \int_{\Omega_I} W_I\phi_J d\Omega, \tag{24}$$

and the stiffness matrix as

$$\begin{aligned} K_{i,IJ} = & -\int_{\Omega_I}\left(\delta t W_I + \frac{\delta t^2}{2}c_{i,\beta}\frac{\partial W_I}{\partial x_\beta}\right)c_{i,\alpha}\frac{\partial\phi_J}{\partial x_\alpha}d\Omega \\ & + \frac{\delta t^2}{2}\int_{\Gamma_I} W_I c_{i,\alpha}\frac{\partial\phi_J}{\partial x_\alpha}c_{i,\beta}n_\beta d\Gamma, \end{aligned} \tag{25}$$

we rewrite Eq. (23) as follows:

$$\sum_{J=1}^{N_I} M_{IJ} f_{i,J}^{n+1} = \sum_{J=1}^{N_I}[M_{IJ} + K_{i,IJ}] f_{i,J}^{n}. \tag{26}$$

To complete the discretization process, the integrals of Eqs. (24) and (25) are to be evaluated numerically. The Gauss quadrature scheme is employed for this purpose. We have

$$M_{IJ} = \sum_{k=1}^{N_G} \xi_k W_I(\mathbf{x}_k)\phi_J(\mathbf{x}_k)|\mathbf{J}^{\Omega_I}| \tag{27}$$

and

$$\begin{aligned} & K_{i,IJ} \\ & = -\sum_{k=1}^{N_G}\xi_k\left(\delta t W_I(\mathbf{x}_k) + \frac{\delta t^2}{2}c_{i,\beta}\left.\frac{\partial W_I}{\partial x_\beta}\right|_{\mathbf{x}_k}\right)\left(c_{i,\alpha}\left.\frac{\partial\phi_J}{\partial x_\alpha}\right|_{\mathbf{x}_k}\right)|\mathbf{J}^{\Omega_I}| \\ & \quad + \frac{\delta t^2}{2}\sum_{k=1}^{N_G^b}\xi_k W_I(\mathbf{x}_k)\left(c_{i,\alpha}\left.\frac{\partial\phi_J}{\partial x_\alpha}\right|_{\mathbf{x}_k}\right)(c_{i,\beta}n_\beta)|\mathbf{J}^{\Gamma_I}|, \end{aligned} \tag{28}$$

where ξ_k is the Gauss weighting factor for the Gauss quadrature point $\mathbf{x}_k$, $\mathbf{J}^{\Omega_I}$ and $\mathbf{J}^{\Gamma_I}$ are the mapping Jacobian matrices for the domain and the boundary integrations, respectively, and N_G and N_G^b are the number of Gauss points used for the domain and the boundary integrations, respectively.

Now, Eq. (26) becomes the fully discretized equation for the nodal point I. Writing this equation for all of the nodal points in the computational domain ($I = 1,\ldots,N$), and assembling the resulting equations in a global system of equations, we can write

$$\mathbf{M}\mathbf{f}_i^{n+1} = [\mathbf{M} + \mathbf{K}_i]\mathbf{f}_i^n, \quad i = 1,\ldots,nQ, \tag{29}$$

where **M**, **K**, and $\mathbf{f}_i$ are the global mass matrix, stiffness matrix, and particle distribution vector, respectively. Equation (29) is a system of N equations with N unknowns, to be solved separately for each direction i, after imposing the boundary conditions.

C. Boundary conditions

The advection equation of the particle distributions is a hyperbolic equation, which requires boundary conditions for the incoming particles at the boundary ($c_{i,\beta}n_\beta < 0$). In this study, we impose boundary conditions using the bounce-back scheme of nonequilibrium distributions [38], i.e.,

$$f_i - f_i^{\mathrm{eq}} = f_{i^*} - f_{i^*}^{\mathrm{eq}}, \tag{30}$$

where f_{i^*} is the outgoing particle distribution along the opposite direction of the incoming distribution f_i. Substituting the equilibrium distribution of Eq. (3) in the above equation, we obtain

$$f_i = f_{i^*} + 2\rho_b t_i (c_{i,\alpha} u_{b,\alpha})/c_s^2, \tag{31}$$

where ρ_b and $u_{b,\alpha}$ are the macroscopic density and velocity at the boundary. If f_{i^*} in Eq. (31) is considered to be the postcollision (prestreaming) distribution, then Eq. (31) becomes an explicit essential boundary condition for the discretized system of Eq. (29).

It should be noted that in Eq. (31), the values of the macroscopic density and velocity at the boundary are imposed when they are predetermined. Otherwise, the computed values of each time step are used. For example, for the solid wall boundary, the so-called no-slip and impermeability conditions are imposed through equating the velocity components in Eq. (31) to those of the wall, while the computed values of the density at the wall in each time step are used in Eq. (31).

D. Numerical considerations

(i) Several choices are applicable for the test function W_I, which lead to several MLPG schemes [36]. For instance, if we use unity as the test function, the term containing derivatives of W_I in Eq. (23) will vanish; or if we make use of a Gaussian distribution test function such as, for example, the local cubic spline, the boundary integrations of Eq. (23) will be zero, provided that the domain of the test function coincides with the control volume. For the reason illustrated below, we use a cubic spline test function in this study.

(ii) There are a number of radial basis functions (RBFs) in the literature. Each of them have some shape parameters that should be tuned in each problem for both accuracy and good performance [35]. Three classical RBFs are the multiquadric function, the Gaussian function, and the thin plate spline function. In our investigations, the multiquadric function, defined as

$$R_i(\mathbf{x}) = [|\mathbf{x} - \mathbf{x}_i|^2 + (\alpha_c d_c)^2]^q, \tag{32}$$

showed more flexibility in the range of applied shape parameters, and therefore we chose this function for all the simulations. In Eq. (32), d_c is usually taken as the average point spacing in the local support domain, and α_c and q are the shape parameters to be determined for each problem.

(iii) The maximum value for the time step leading to a stable solution is determined using the Courant number, $\mathrm{CFL} = \max\{|\mathbf{c}_i|\}\delta t/\delta x_{\min}$, where $\delta x_{\min}$ is the minimum point spacing in the domain. However, in order to obtain expected convergence rates in the point resolution study of the method, it is necessary that $\delta t \sim \delta x^2$, as will be illustrated in this study.

(iv) The coefficient matrix in Eq. (29) is the mass matrix **M**, which is, because of the local nature of the MLPG method, a sparse matrix. Therefore, the system of equations can be efficiently solved using sparse iterative solvers. However, the explicit nature of the standard lattice Boltzmann method, and the diagonally dominant character of the mass matrix, which is especially the case when a Gaussian distribution test function is used, motivated us to find rational ways of diagonalizing the mass matrix, and thus saving much of the computational time. In the finite-element analysis, the procedure of diagonalizing the mass matrix is referred to as *mass lumping* [40]. In this study, we make use of row-sum lumping, in which the sum of the elements of each row of the mass matrix is used as the diagonal element [40]. As a result, our meshless lattice Boltzmann method becomes an explicit solver for the fluid flow problems.

E. Performance analysis and memory requirement

In this section, we present a comparison of performance and memory usage between our method and the standard lattice Boltzmann method. First, it should be noted that in the present meshless lattice Boltzmann method, all of the works regarding the computation of the mass and the stiffness matrices, including the consideration of a control volume for each nodal point, the creation of shape functions based on local supports, and all the required integrations, are to be performed only once as a preprocessing step and before the beginning of the time-step computations. Therefore, these one time computations should not be included in the performance analysis. To compare the repetitive computations during the time marching part of the two methods, we note that the collision step in our method is the same as that of the standard LBM, and the only difference is in the streaming step. In the standard LBM, the streaming section of the code has the following loop per time step per node:

```
do $i = 1,nQ$
    $f(nb,i) = \tilde{f}(\mathrm{node},i)$
end,
```

where nQ is the number of velocity directions and nb is a neighbor node of the considered node along direction i. Therefore, in the streaming step of the standard LBM, we have nQ operations per time step per node. If the total number of nodes in the domain used in the standard LBM is N_{SLB}, then the total number of operations will be $N_{\mathrm{SLB}} \times nQ$ per time step. In our method, the streaming section of the code contains the following loop per time step per node:

```
do $i = 1,nQ$
  do $j = 1,nnz$
    $\text{rhs(node)} = \text{rhs(node)} + K(j,i) * \tilde{f}(JK(j),i)$
  end
  $f(\text{node},i) = \text{rhs(node)}/M(\text{node},i)$
end,
```

where nnz is the number of nonzero elements in the considered row of the stiffness matrix K, which corresponds to the number of nodes involved in the local interpolation of the field variable in the Gauss points of the control volume of the considered node. It can be seen that the number of operations in the streaming step of our method is $nQ \times (3 \times nnz + 2)$ per time step per node. If the total number of nodes in the domain used in the present meshless LBM is N_{MLB}, then the total number of operations will be $N_{\text{MLB}} \times nQ \times (3 \times nnz + 2)$ per time step. Therefore, the ratio of the total number of operations per time step of the two methods will be

$$\frac{N_{\text{MLB}} \times nQ \times (3 \times nnz + 2)}{N_{\text{SLB}} \times nQ} = \frac{N_{\text{MLB}}}{N_{\text{SLB}}} \times (3 \times nnz + 2). \tag{33}$$

It can be concluded from Eq. (33) that if we have $N_{\text{MLB}} < N_{\text{SLB}}/(3 \times nnz + 2)$, then the total number of operations per time step in the present method becomes less than that of the standard LBM. It is evident that for flows in complex geometries, the required number of grid points in a uniform Cartesian grid used in the standard LBM is considerably larger than that in a controlled body-fitted point distribution that could be used in the meshless LBM. In such circumstances, the meshless method requires less computational time to complete a time step and actually outperforms the standard LBM.

Considering the memory usage, the major arrays used in a standard LB code are $f(N_{\text{SLB}},nQ)$, $\tilde{f}(N_{\text{SLB}},nQ)$, $f^{\text{eq}}(N_{\text{SLB}},nQ)$, $\rho(N_{\text{SLB}})$, and $u(N_{\text{SLB}},nD)$, where nD is the number of dimensions. In our method, we have these arrays (but with sizes of N_{MLB}) plus some additional arrays, the major ones being the lumped mass matrix $M(N_{\text{MLB}},nQ)$, the stiffness matrix in the compact sparse form $K(N_{\text{MLB}} \times nnz,nQ)$, and the pointer matrices of the row and column of nonzero elements of the stiffness matrix, $IK(N_{\text{MLB}} \times nnz,nQ)$ and $JK(N_{\text{MLB}} \times nnz,nQ)$. It can be seen that the ratio of the total number of memory words of our method to that of the standard LB is

$$\frac{N_{\text{MLB}}}{N_{\text{SLB}}} \times \left(1 + \frac{nQ \times (3 \times nnz + 1)}{nQ \times 3 + nD + 1}\right). \tag{34}$$

It is observed from this equation that if the number of nodal points used in the meshless LBM is considerably less than that of the standard LBM, the memory requirement of our method becomes comparable with (or even lower than) that of the standard LBM for the target cases.

III. NUMERICAL SIMULATIONS

To check the validity and correctness of our proposed method, three benchmark fluid flow problems, namely the shear flow between two infinite parallel plates (Couette flow), the flow inside a suddenly rotated circular cylinder (circular Couette flow), and the impulsively started flow around a circular cylinder, are simulated.

A. Shear flow between two parallel plates

The first test case considered in this study is the shear flow between two parallel plates. The plate at $y = 0$ is kept stationary, while the other plate at $y = L$ begins to move with constant velocity $u_0\mathbf{e}_x$. Solving the governing flow equations by imposing the no-slip boundary conditions on the plates yields

$$U(Y,T) = Y + 2\sum_{k=1}^{\infty} \frac{(-1)^k}{k\pi} e^{-(k\pi)^2 T} \sin(k\pi Y), \tag{35}$$

where $Y = y/L$, $T = \nu t/L^2$, and $U = u/u_0$ are the dimensionless y coordinate, time, and x component of the velocity, respectively.

In solving this test case using the proposed method, we consider a square computational domain in the xy plane and discretize it using 5×5, 9×9, 17×17, 33×33, and 65×65 uniform point distributions. The periodic boundary condition is imposed in the x direction. The no-slip and the impermeability boundary conditions at the walls are imposed by equating the velocity components in Eq. (31) to those of the walls. For each point distribution, time iterations continue until a steady state is reached. The Mach number defined as $\text{Ma} = u_0/c_s$ is 0.05 for all the simulations.

To determine the order of accuracy of the numerical scheme, we employ the following relative L^2 error norm:

$$E = \left(\frac{\sum_{I=1}^{N_e}(U_{aI} - U_{nI})^2}{\sum_{I=1}^{N_e} U_{aI}^2}\right)^{1/2}, \tag{36}$$

where U_{aI} and U_{nI} are the analytical and numerical solutions of the velocity at point I, respectively, and N_e is the fixed number of points used for the error analysis. For this test case, a 10×10 uniform point distribution is used for the error analysis. The variation of the above error norm with respect to the point spacing h is sketched in the logarithmic diagram of Fig. 1. It is observed from this figure that the error shows an exponential decrease with reducing h until it is saturated by the temporal errors. The rates of the convergence of our numerical method, computed by the linear regression of the data in Fig. 1, are $R = 2.04$, 2.33, and 2.53 for $\delta t = 0.000\,25$, 0.0001, and 0.000 01, respectively. Moreover, it is demonstrated from this figure that in order to obtain expected convergence rates in the point resolution study, it is necessary that $\delta t \sim \delta x^2$.

To investigate the effects of the values of the shape parameters of the multiquadric radial basis function (MQ-RBF) used in our interpolations, namely α_c and q, on the accuracy of the results, we solve the case with 9×9 nodal points for different values of these shape parameters and compute the aforementioned error norm in each case. The results are sketched in Fig. 2. As this figure shows, by increasing the value

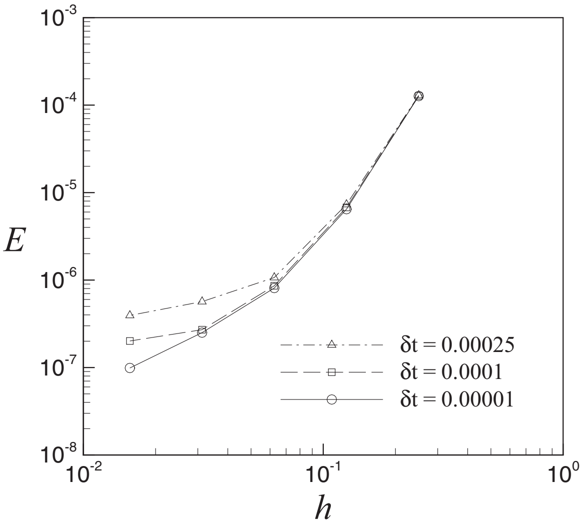


FIG. 1. Numerical convergence of the meshless lattice Boltzmann method in L^2 error norm with respect to the point spacing for the Couette flow.

of q from 0.5 to 1.5–2.03, the error norm reduces by one order of magnitude. Then, by increasing q to 2.5, the error increases by one order of magnitude. Therefore, the values of q in the range 1.5–2.03 seem to give the best results. However, the best values for α_c depend on the selected value for q. For the cases in which q is in the range 1.5–2.03, any value for α_c less than 8 results in the error of the lowest order of magnitude in our simulations. Another important result observed in this figure is that by increasing the value of q, the applicable range of α_c decreases.

B. Flow inside a suddenly rotated circular cylinder

Another benchmark problem in which an exact solution to the Navier-Stokes equations exists is the two-dimensional flow inside a suddenly rotated long circular cylinder. We consider a cylinder with radius r_0 that impulsively starts to rotate at constant angular velocity ω. For small values of ω, where the two-dimensional assumption remains true, solving the Navier-Stokes equations determines the velocity and the pressure distributions in dimensionless form as

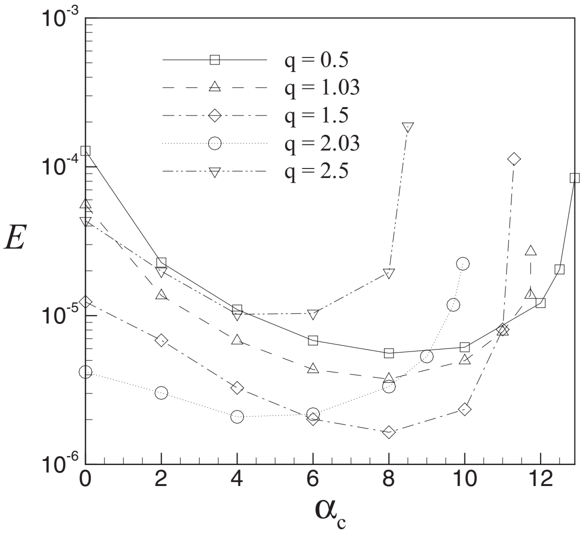


FIG. 2. Effect of the MQ-RBF shape parameters on the error norm for the Couette flow.

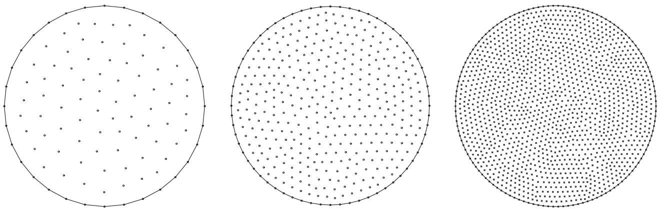


FIG. 3. Point distributions consist of 108, 409, and 1499 irregularly distributed points, left to right, respectively, for the simulation of the flow inside a suddenly rotated circular cylinder.

$$U_\theta(R,T) = R + 2\sum_{k=1}^{\infty} \frac{J_1(\alpha_k R)}{\alpha_k J_0(\alpha_k)} e^{-\alpha_k^2 T}, \tag{37}$$

$$P(R,T) = -2\int_R^1 \frac{U_\theta^2}{R} dR, \tag{38}$$

where J_0 and J_1 are the zeroth- and the first-order Bessel functions of the first kind, respectively, α_k is the kth root of J_1, and the dimensionless parameters are defined as $R = r/r_0$, $T = \nu t/r_0^2$, $U_\theta = u_\theta/r_0\omega$, and $P = (p - p_0)/(1/2\rho_0 r_0^2\omega^2)$, where p_0 (or ρ_0) is the prescribed pressure (or density) at the cylinder wall.

The computational domains with a number of point distributions used in our meshless lattice Boltzmann simulations are sketched in Fig. 3. We discretize the domain using 108, 409, 1499, and 5886 arbitrary distributed points. The no-slip and the impermeability boundary conditions at the cylinder wall are imposed through the bounce-back scheme of Eq. (31). The Mach number defined as $\mathrm{Ma} = r_0\omega/c_s$ is 0.05 in all the simulations. To determine the order of accuracy of our numerical scheme for this test case, we employ the error norm defined in Eq. (36) and investigate the variation of this error with respect to the average point spacing h defined as

$$h = \frac{\sqrt{S}}{\sqrt{N} - 1}, \tag{39}$$

where S is the surface area and N is the number of the nodal points of the entire computational domain. The result is illustrated in Fig. 4 for $\delta t = 0.0001$. The rate of the convergence of the method for this test case is obtained as $R = 1.73$.

The velocity and the pressure distributions along a radius of the cylinder at different dimensionless times are depicted and compared with the analytical solutions of Eqs. (37) and (38) in Figs. 5 and 6. The results of the meshless lattice Boltzmann method presented in these figures are for the finest point resolution ($N = 5886$) and are in excellent agreement with the analytical solution.

C. Flow past an impulsively started cylinder

The third test case considered here is the impulsively started flow around a circular cylinder. The computational domain and

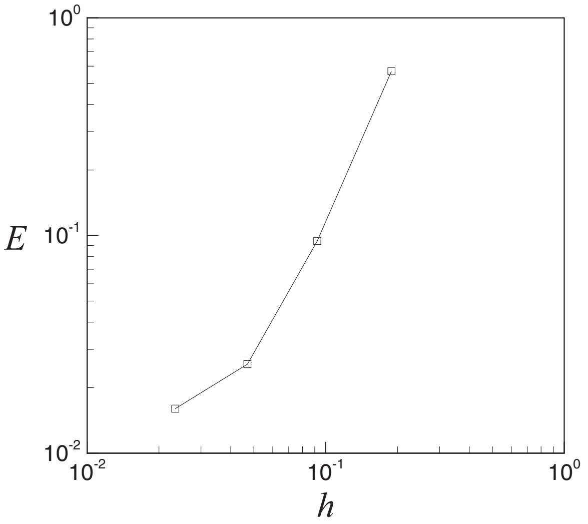


FIG. 4. Numerical convergence of the meshless lattice Boltzmann method in L^2 error norm with respect to the average point spacing for the circular Couette flow.

the point distribution for this test case are sketched in Fig. 7. To capture the boundary layer velocity gradients so that the characteristic quantities of the flow, such as the separation points and the drag coefficients, are computed accurately, the point distribution is made considerably denser next to the cylinder wall. It is common in the literature that a potential flow is assumed to exist at $t = 0^+$ in order to reduce the impulsive start effects. At $t > 0$, the zero velocity condition at the cylinder wall, and the constant density and velocity condition at the external borders of the computational domain, are imposed using the bounce-back scheme of Eq. (31).

The dimensionless parameters for this flow are $T = 2u_\infty t/D$, $X = x/D$, $Y = y/D$, $U = u/u_\infty$, $V = v/u_\infty$, and $C_p = (p - p_\infty)/(1/2\rho_\infty u_\infty^2)$, where u_∞, ρ_∞, and p_∞ are the uniform velocity, density, and pressure of the flow far from the cylinder (the external borders of the computational domain in our simulation), respectively, and D is the diameter of the cylinder. The Reynolds number for this flow is defined as

$$\mathrm{Re} = \frac{u_\infty D}{\nu}. \tag{40}$$

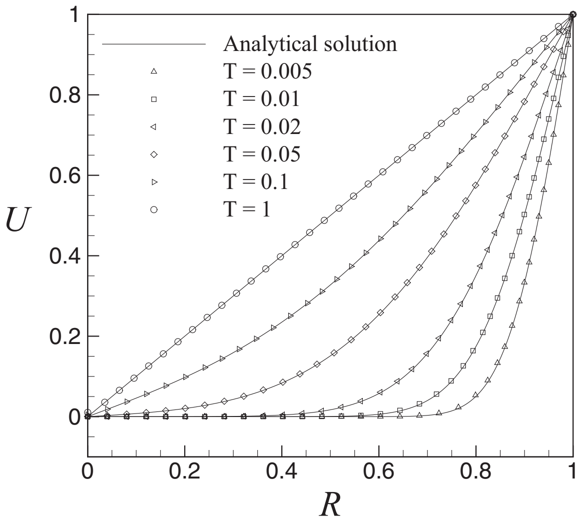


FIG. 5. The velocity distributions along a radius of a suddenly rotated circular cylinder at different dimensionless times. Lines, analytical solution; symbols, present study.

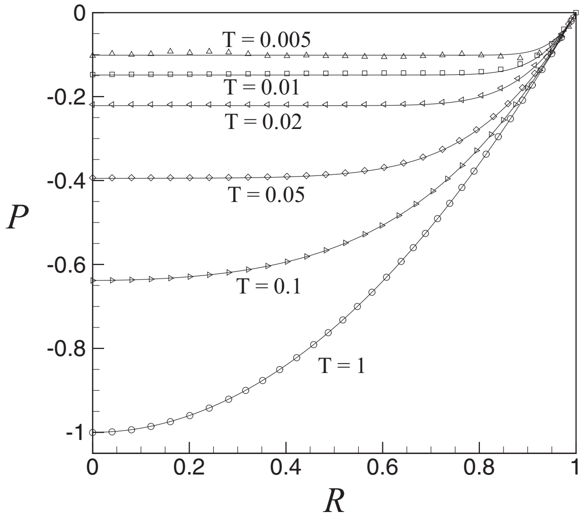
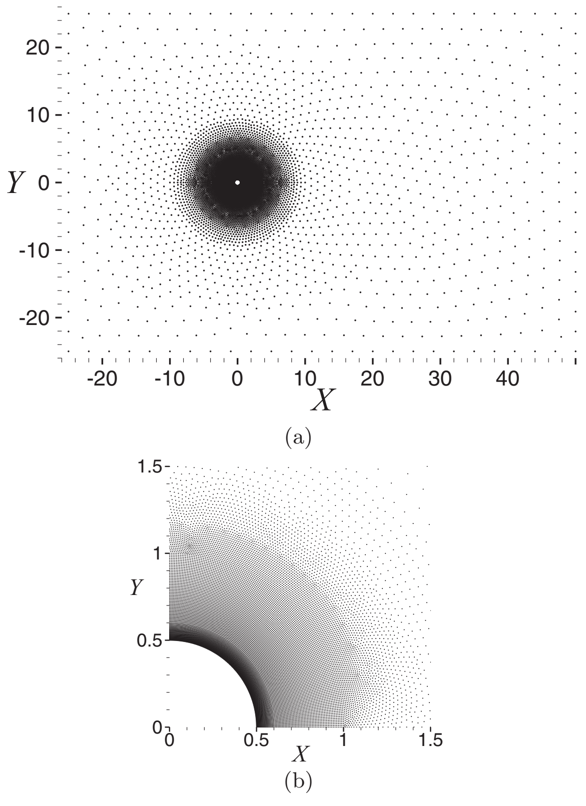


FIG. 6. The pressure distributions along a radius of a suddenly rotated circular cylinder at different dimensionless times. Lines, analytical solution; symbols, present study.



(a)

(b)

FIG. 7. Computational domain specified with 68 112 distributed points for the impulsively started cylinder flow. (a) The global domain, (b) a closeup view.

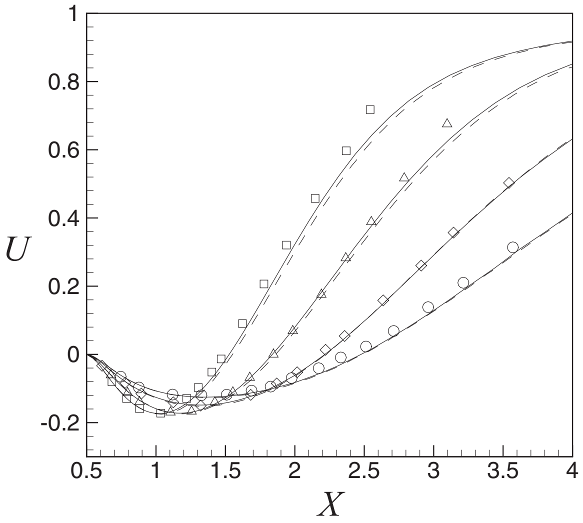


FIG. 8. The x component velocity distributions along the horizontal symmetry axis behind an impulsively started cylinder at different times for Re = 40. Solid lines, present study: $T = 5.4$, 7.4, 10.6, and 14; dashed lines, standard LB, Li *et al.* [44]: $T = 5.5$, 7.5, 10.5, and 14; symbols, experimental results, Coutanceau and Bouard [41]: □, $T = 5.4$; △, $T = 7.4$; ◇, $T = 10.6$; ○, $T = 14$.

Simulations are performed for Re = 40, 550, and 3000, for which a lot of experimental and numerical data exist in the literature [41–44]. The Mach number defined as $\mathrm{Ma} = u_\infty/c_s$ is 0.1 in all of the simulations of this test case.

Figures 8 and 9 illustrate the x component velocity distributions along the horizontal symmetry axis behind the cylinder at different dimensionless times, for Re = 40 and 550, respectively. Our results are compared with the experimental results of Coutanceau and Bouard [41,42], and the results of the standard LBM of Li *et al.* [44], in these figures. Excellent agreement is observed between our method and the standard LBM (note that the times of the standard LBM results are a little different for Re = 40). When comparing the results of the two numerical methods with the experimental results, very good agreement is observed in the trends of the curves and in the estimated reattachment point of the flow behind the cylinder. However, some minor discrepancies can be seen between the numerical and the experimental results, which may be attributed to the difference between the cylinder-domain size ratio of the computational domains and the experimental setup, and the three-dimensional effects in the experiment.

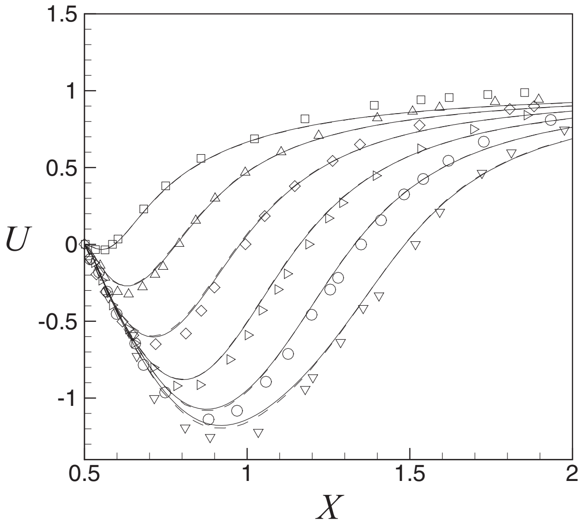


FIG. 9. The x component velocity distributions along the horizontal symmetry axis behind an impulsively started cylinder at different times for Re = 550. Solid lines, present study: $T = 1$, 2, 3, 4, 5, and $T = 6$; dashed lines, standard LB, Li *et al.* [44]: $T = 1$, 2, 3, 4, 5, and $T = 6$; symbols, experimental results, Bouard and Coutanceau [42]: □, $T = 1$; △, $T = 2$; ◇, $T = 3$; ▷, $T = 4$; ○, $T = 5$; ▽, $T = 6$.

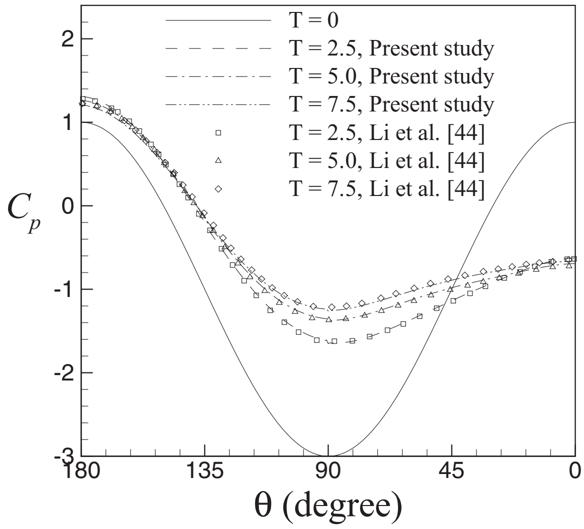


FIG. 10. The dimensionless pressure distribution on the cylinder surface for Re = 40 at different dimensionless times.

To investigate the accuracy of the computed pressure field, we compare the computed pressure coefficient, C_p, on the cylinder surface with the values reported by Li *et al.* [44] using the standard LBM. Figures 10 and 11 present these comparisons. It is observed that the results of our method for the pressure coefficient completely match those of the standard LBM at different dimensionless times.

One of the most important characteristic parameters of the flow around an object is the drag coefficient, which is the dimensionless form of the horizontal force exerted by the flow on the object or vice versa. The total drag coefficient on the cylinder surface is calculated from

$$C_D = \frac{\mathbf{F}\cdot\mathbf{e}_x}{1/2\rho_\infty u_\infty^2 D}, \tag{41}$$

where $\mathbf{F}$ is the total force per unit depth on the lateral surface of the cylinder, evaluated by

$$\mathbf{F} = \mathbf{F}_p + \mathbf{F}_f = \int_C p\mathbf{n}\,dA + \int_C \boldsymbol{\tau}\cdot\mathbf{n}\,dA, \tag{42}$$

where $\mathbf{F}_p$ and $\mathbf{F}_f$ are the pressure and the friction forces on the cylinder, respectively, $\mathbf{n}$ is the local unit normal vector of the lateral surface of the cylinder C, and $\boldsymbol{\tau}$ is the deviatoric

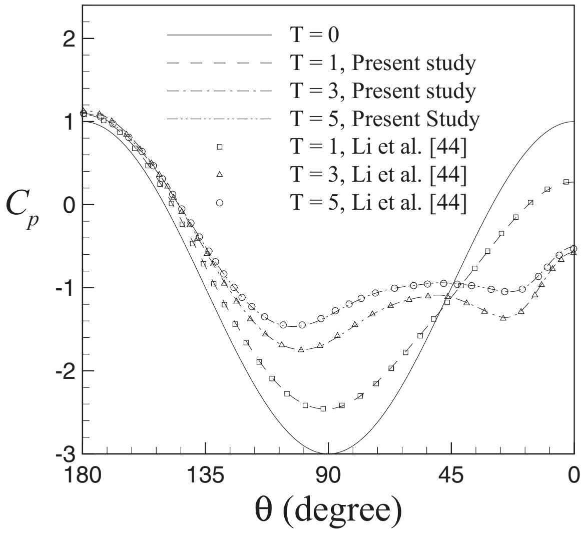


FIG. 11. The dimensionless pressure distribution on the cylinder surface for Re = 550 at different dimensionless times.

stress tensor for the incompressible flow, $\boldsymbol{\tau} = \rho\nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$, which can be computed using the nonequilibrium part of the distribution function as [45]

$$\tau_{\alpha\beta} = \left(1 - \frac{1}{2\tau + 1}\right)\sum_{i=1}^{b}\left(f_i - f_i^{\mathrm{eq}}\right)\left(e_{i\alpha}e_{i\beta} - \frac{1}{2}e_{i\gamma}e_{i\gamma}\delta_{\alpha\beta}\right). \tag{43}$$

The time evolution of the pressure drag, the friction drag, and the total drag coefficients for the impulsively started cylinder flow is illustrated in Figs. 12, 13, and 14 for Re = 40, 550, and 3000, respectively. In these figures, we compare the results of our method with those of the standard LBM of Li *et al.* [44] and the vortex method of Koumoutsakos and Leonard [43]. The agreement between the results of the three methods is very good for the friction drag. For the pressure drag, excellent agreement is observed between our method and the standard LBM. However, at the initial times ($T < 0.6$), a major difference can be seen between the pressure drag coefficients obtained by the lattice Boltzmann methods (including our method and the standard LBM) and those of the vortex method. This difference is expected as a consequence of the compressible nature of all lattice Boltzmann methods, which results in the development of a pressure wave, due to the impulsive start, propagating from the cylinder at a finite speed of sound. As a result, a relative maximum point is

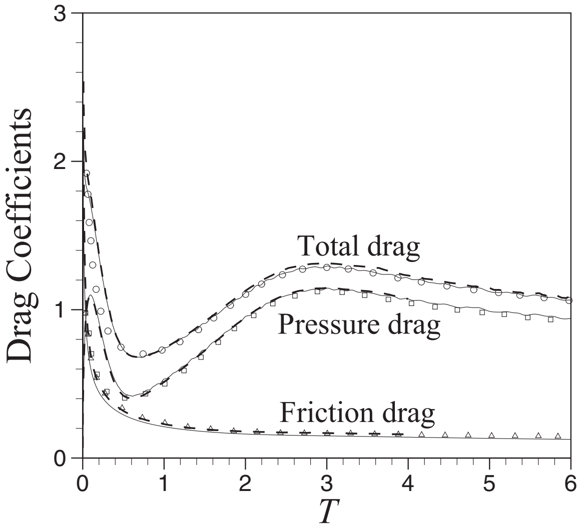


FIG. 13. Time evolution of the drag coefficients of the impulsively started cylinder flow at Re = 550 and Ma = 0.1. Solid lines, present study; dashed lines, standard LB, Li *et al.* [44]; symbols, vortex method, Koumoutsakos and Leonard [43].

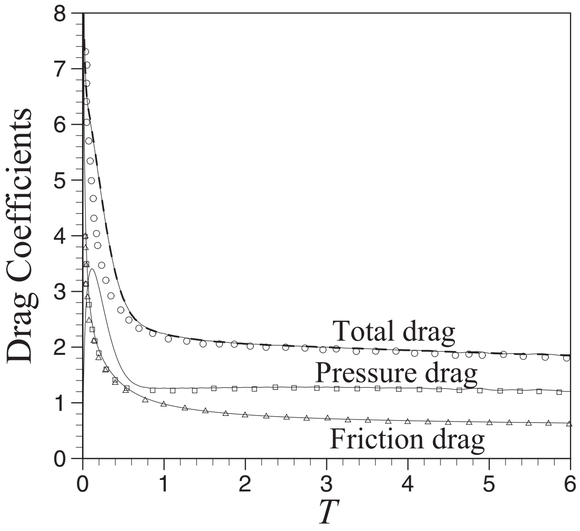


FIG. 12. Time evolution of the drag coefficients of the impulsively started cylinder flow at Re = 40 and Ma = 0.1. Solid lines, present study; dashed lines, standard LB, Li *et al.* [44]; symbols, vortex method, Koumoutsakos and Leonard [43].

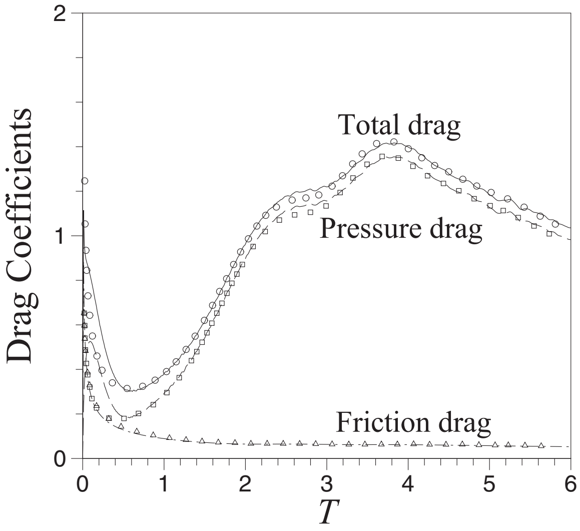


FIG. 14. Time evolution of the drag coefficients of the impulsively started cylinder flow at Re = 3000 and Ma = 0.1. Solid line, present study; symbols, vortex method, Koumoutsakos and Leonard [43].

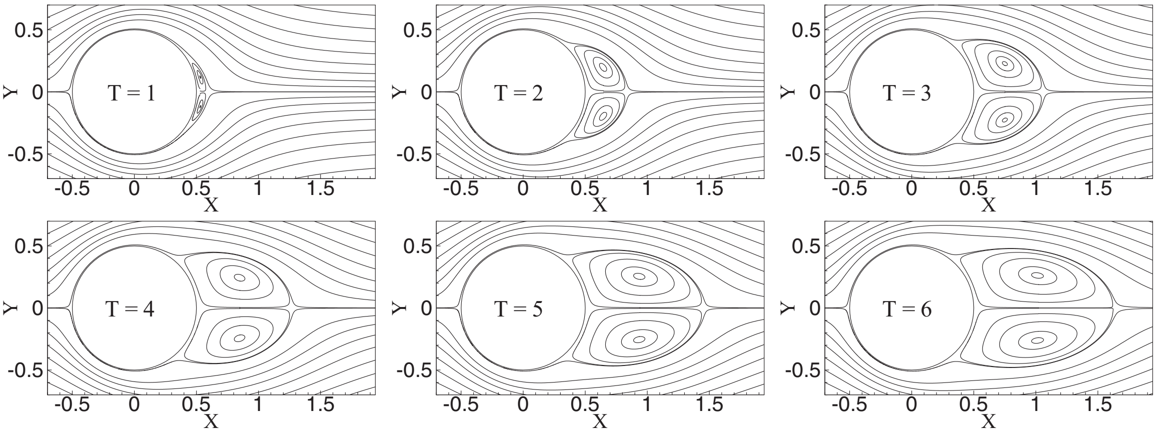


FIG. 15. Time evolution of the streamlines of the impulsively started cylinder flow at Re = 40 and Ma = 0.1.

observed in the initial times of the pressure drag diagrams of the lattice Boltzmann methods, while for the incompressible Navier-Stokes solvers, the pressure field's response to the impulsive start is instantaneous, and therefore the pressure drag is a strictly decreasing function of time with the inverse square root trend as expected from the theory. Further discussions of this phenomenon can be found in Ref. [44].

Figures 15, 16, and 17 show the streamlines of the flow past an impulsively started cylinder at the initial stages of the flow, from $T = 1$ to 6, for Re = 40, 550, and 3000, respectively. The formation and the development of the vortices at different Reynolds numbers are in qualitative agreement with the previous experimental and numerical studies [41–44].

IV. CONCLUSIONS

In this study, we have presented a meshless lattice Boltzmann method for the simulation of fluid flows at the nearly incompressible limit. In our method, the collision and the streaming steps are split, as in the standard lattice Boltzmann method. We rewrite the streaming equation, which is the exact solution of the pure advection equation, as the pure advection equation and recast it into its local weak form. The Lax-Wendroff scheme is used for the time discretization, and the meshless local Petrov-Galerkin scheme with augmented radial basis functions is applied for the space discretization. The boundary conditions are imposed through the bounce-back scheme for the nonequilibrium parts of the distribution functions.

To validate the proposed numerical technique, three benchmark problems have been solved, namely the plane Couette flow, the circular Couette flow, and the impulsively started cylinder flow at the Reynolds numbers of 40, 550, and 3000. Based on the analytical solutions of the first two test cases, our method shows approximately second-order accuracy in space. Furthermore, our method illustrates very good agreement with some previous experimental and numerical studies in the literature for the third test case.

Based on the results presented in this study, the proposed meshless lattice Boltzmann method is a promising extension of the lattice Boltzmann family. The main advantage of the meshless lattice Boltzmann method with respect to the previous extensions of the lattice Boltzmann method is the elimination of the need for meshing the computational

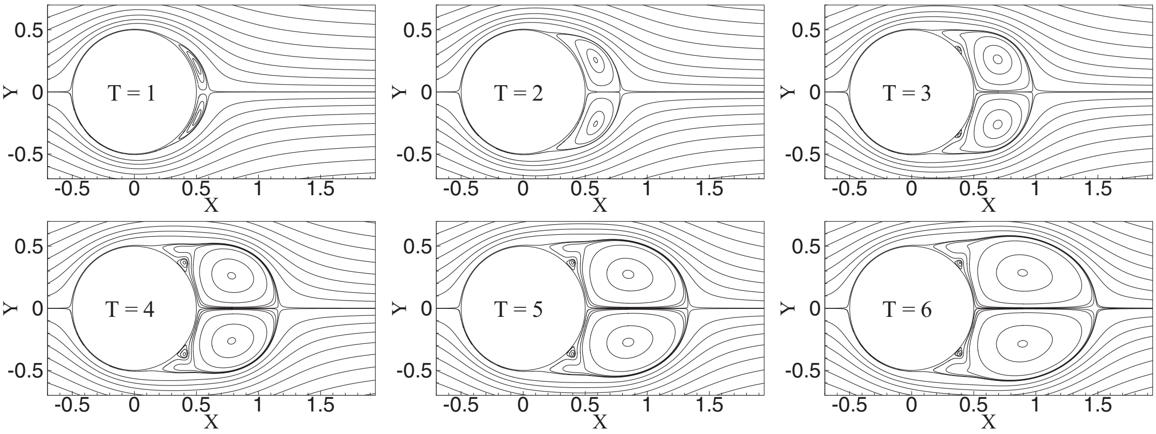


FIG. 16. Time evolution of the streamlines of the impulsively started cylinder flow at Re = 550 and Ma = 0.1.

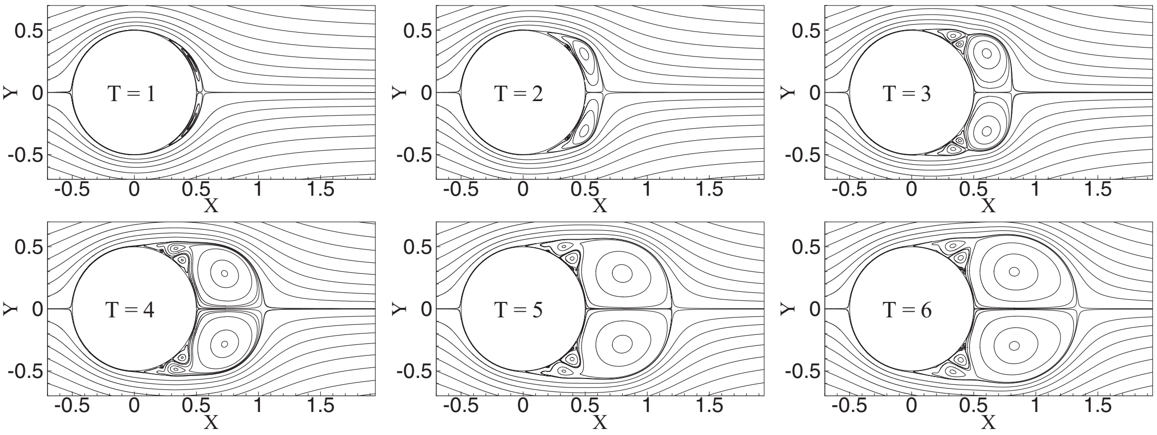


FIG. 17. Time evolution of the streamlines of the impulsively started cylinder flow at Re = 3000 and Ma = 0.1.

domain. This is of great importance because using any meshes introduce significant numerical errors in the solution, and meshing a domain with sufficiently good-quality elements is a cumbersome or even impossible process. These situations are more likely to arise in geometrically complex domains, especially in three-dimensional problems.

[1] G. R. McNamara and G. Zanetti, Phys. Rev. Lett. **61**, 2332 (1988).

[2] F. Higuera, S. Succi, and R. Benzi, Europhys. Lett. **9**, 345 (1989).

[3] F. Higuera and S. Succi, Europhys. Lett. **8**, 517 (1989).

[4] F. Higuera and J. Jimenez, Europhys. Lett. **9**, 663 (1989).

[5] S. Chen, H. Chen, D. Martinez, and W. Matthaeus, Phys. Rev. Lett. **67**, 3776 (1991).

[6] Y. Qian, D. d'Humières, and P. Lallemand, Europhys. Lett. **17**, 479 (1992).

[7] R. Benzi, S. Succi, and M. Vergassola, Phys. Rep. **222**, 145 (1992).

[8] S. Chen and G. D. Doolen, Annu. Rev. Fluid Mech. **30**, 329 (1998).

[9] C. K. Aidun and J. R. Clausen, Annu. Rev. Fluid Mech. **42**, 439 (2010).

[10] G. R. McNamara, A. L. Garcia, and B. J. Alder, J. Stat. Phys. **81**, 395 (1995).

[11] J. D. Sterling and S. Chen, J. Comput. Phys. **123**, 196 (1996).

[12] X. He and L.-S. Luo, Phys. Rev. E **56**, 6811 (1997).

[13] F. Nannelli and S. Succi, J. Stat. Phys. **68**, 401 (1992).

[14] X. He, L.-S. Luo, and M. Dembo, J. Comput. Phys. **129**, 357 (1996).

[15] X. He and G. Doolen, J. Comput. Phys. **134**, 306 (1997).

[16] H. Chen, Phys. Rev. E **58**, 3955 (1998).

[17] C. Shu, X. D. Niu, and Y. T. Chew, Phys. Rev. E **65**, 036708 (2002).

[18] O. Filippova and D. Hänel, J. Comput. Phys. **147**, 219 (1998).

[19] A. Dupuis and B. Chopard, Phys. Rev. E **67**, 066707 (2003).

[20] N. Cao, S. Chen, S. Jin, and D. Martinez, Phys. Rev. E **55**, R21 (1997).

[21] R. Mei and W. Shyy, J. Comput. Phys. **143**, 426 (1998).

[22] V. Sofonea and R. F. Sekerka, J. Comput. Phys. **184**, 422 (2003).

[23] A. Bardow, I. Karlin, and A. Gusev, Europhys. Lett. **75**, 434 (2006).

[24] H. Xi, G. Peng, and S.-H. Chou, Phys. Rev. E **59**, 6202 (1999).

[25] S. Ubertini, G. Bella, and S. Succi, Phys. Rev. E **68**, 016701 (2003).

[26] M. Stiebler, J. Tölke, and M. Krafczyk, Comput. Fluids **35**, 814 (2006).

[27] D. V. Patil and K. Lakshmisha, J. Comput. Phys. **228**, 5262 (2009).

[28] T. Lee and C.-L. Lin, J. Comput. Phys. **171**, 336 (2001).

[29] T. Lee and C.-L. Lin, J. Comput. Phys. **185**, 445 (2003).

[30] Y. Li, E. J. LeBoeuf, and P. K. Basu, Phys. Rev. E **72**, 046711 (2005).

[31] X. Shi, J. Lin, and Z. Yu, Int. J. Numer. Methods Fluids **42**, 1249 (2003).

[32] A. Düster, L. Demkowicz, and E. Rank, Int. J. Numer. Methods Eng. **67**, 1094 (2006).

[33] M. Min and T. Lee, J. Comput. Phys. **230**, 245 (2011).

[34] G.-R. Liu, *Mesh Free Methods: Moving Beyond the Finite Element Method* (CRC Press, Boca Raton, FL, 2009), Vol. 712.

[35] G.-R. Liu and Y.-T. Gu, *An Introduction to Meshfree Methods and their Programming* (Springer Netherlands, Dordrecht, 2005).

[36] S. N. Atluri, *The Meshless Method (MLPG) for Domain & BIE Discretizations* (Tech Science Press, Forsyth, GA, 2004), Vol. 677.

[37] P. L. Bhatnagar, E. P. Gross, and M. Krook, Phys. Rev. **94**, 511 (1954).

[38] X. He, S. Chen, and G. D. Doolen, J. Comput. Phys. **146**, 282 (1998).

[39] D. A. Wolf-Gladrow, *Lattice-gas Cellular Automata and Lattice Boltzmann Models: An Introduction* (Springer-Verlag, Berlin, 2000), Vol. 1725.

[40] J. N. Reddy, *An Introduction to the Finite Element Method* (McGraw-Hill, New York, 2006), Vol. 2.

[41] M. Coutanceau and R. Bouard, J. Fluid Mech. **79**, 257 (2006).

[42] R. Bouard and M. Coutanceau, J. Fluid Mech. **101**, 583 (2006).

[43] P. Koumoutsakos and A. Leonard, J. Fluid Mech. **296**, 1 (2006).

[44] Y. Li, R. Shock, R. Zhang, and H. Chen, J. Fluid Mech. **519**, 273 (2004).

[45] D. Yu, R. Mei, L.-S. Luo, and W. Shyy, Progr. Aerospace Sci. **39**, 329 (2003).