Research Article

A Coupled Lattice Boltzmann-Volume Penalization for Flows Past Fixed Solid Obstacles with Local Mesh Refinement

Kai Guo,[1] Xiongwei Cui [ID],[1] and Minghao Liu[2]

[1]*College of Shipbuilding and Engineering, Harbin Engineering University, Harbin 150001, China*
[2]*School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85281, USA*

Correspondence should be addressed to Xiongwei Cui; cuixiongwei@yahoo.com

A coupled Lattice Boltzmann-Volume Penalization (LBM-VP) with local mesh refinement is presented to simulate flows past obstacles in this article. Based on the finite-difference LBM, the local mesh refinement is incorporated into the LBM to improve computing efficiency. The volume penalization method is introduced into the LBM by an external forcing term. In the LBM-VP method, the processes of interpolating velocities on the boundaries points and distributing the force density to the Eulerian points near the boundaries are unnecessary. Performing the LBM-VP on a certain point, only the variables of this point are needed, which means the whole procedure can be conducted parallelly. As a consequence, the whole computing efficiency can be improved. To verify the presented method, flows past a single circular cylinder, a pair of cylinders in tandem arrangement, and a NACA-0012 are investigated. A good agreement between the present results and the data in the previous literatures is achieved, which demonstrates the accuracy and effectiveness of the present method to solve the flows past obstacle problems.

1. Introduction

Flows past obstacles are related to many applications, especially in marine and offshore engineering, aerospace engineering, current or wind turbines, and so on. Numerical studies and simulations of flows past obstacles are of great use in these areas. As an alternative to the traditional Navier-Stokes (N-S) equation solver, the Lattice Boltzmann method (LBM) has been used widely in the simulations of flows past obstacles [1]. The reason why LBM is used so popularly is that it has many noticeable advantages like simplicity in coding, parallel computation, and explicit calculation. Just like in the traditional N-S equation solvers, two main methods are used in LBM when dealing with complex boundary: body-fitted grid methods and immersed boundary methods (IBM).

For the body-fitted grid methods, generating a body-fitted grid is an extremely expensive process when complex boundaries are involved. Still, it is also difficult to create a high quality body-fitted grid with simple boundaries. Besides, in the process of creating body-fitted grids, the structured and unstructured grids are frequently used. However on the

structured and unstructured grids, the order of accuracy is lower than that on the Cartesian grids [2]. Compared with the body-fitted grid methods, the immersed boundary methods, which are proposed by Peskin [3], can be implemented easily. In IBM, there is no need to create body-fitted grids. The flow field is represented on a fixed Cartesian mesh on which the N-S equations are solved. And the boundary is represented by a Lagrangian grid. The variables on these two grids are related by a discrete delta function interpolation. The brightest spot of IBM is that it uses a restoring force to reflect the boundary effect on the flow. So the N-S equations can be solved without considering the boundaries, which means that the N-S equations can be solved on the whole Cartesian mesh. Feng and Michaelides [4] incorporated the IBM into the LBM firstly to solve fluid particles interaction problems. As mentioned above, the boundary effect on the flows is reflected by restoring force. However, the nonslip boundary conditions are not always guaranteed. As a result, in the simulations of flows past boundaries some streamlines may penetrate the solid body boundary. Wu and Shu [5] proposed a new version of IBM, in which the velocity correction near

the boundary is considered as unknown and is computed implicitly to guarantee the nonslip boundary conditions. In the IBM, a Vandermonde matrix should be recomputed at every time step, which means more computing time is needed [2].

Recently, Benamour et al. [6] incorporated another type of IBM, Volume Penalization (VP), which is firstly proposed by Arquis and Caltagirone [7], into LBM. In the VP, the solid boundary is considered as a porous medium with very small permeability. By using a mask function, the solid boundaries are modeled on the Euler grids. So the Lagrangian grids of the VP are just part of the Euler grids which are marked by a mask function. This method has been successfully used by several researchers to simulate and study flows past obstacles [8–10]. By the forcing term proposed by Guo et al. [11], the force induced by boundaries in the VP method is incorporated into LBM. Compared with the LBM-IBM, there is no need to use a delta function to interpolate the velocity at boundaries in the LBM-VP. Also, the process of distributing the force density into the Eulerian points near the boundaries is unnecessary. Performing the VP procedure on a certain Lagrangian point just needs the variables of this certain Euler point, which means that the VP can be performed parallelly. Considering the parallelizability of the LBM, the whole LBM-VP calculating procedure can be conducted parallelly.

For the standard LBM, the whole calculation process consists of two subprocesses: the collision and the streaming. In the streaming subprocess, particles move from one mesh point to its neighbor points within a time step. So the calculation mesh is limited to uniform mesh, which limits the application of the LBM greatly. To eliminate this disadvantage, many researchers have proposed several improvements to implement the LBM on the nonuniform mesh [12–16] and adaptive mesh [17–19]. In this article, the finite-difference LBM, proposed by Lee and Lin [20], is adopted to perform the LBM-VP on the nonuniform grids aiming to improve the computing efficiency. One of the highlights of this finite-difference LBM is that, with the help of using a special finite-difference Lattice Boltzmann scheme, all of the blocks at different refinement levels can be advanced in time with the same time step. So it is very easy to conduct the LBM on the nonuniform mesh, especially when the VP or other IBM are incorporated into LBM.

In this article, to validate the proposed LBM-VP with local mesh refinement and its computing efficiency, flows past a circular cylinder, flows past a pair of circular cylinders in tandem arrangement, and flows past the NACA-0012 airfoil with 10° angle of attack (AOA) and 15° AOA are studied. And the results are compared with available data in previous literatures. The computing time of the LBM-VP performed on nonuniform mesh is also compared with that of the LBM-VP performed on uniform mesh to show the computing efficiency. The rest of this paper is arranged as follows. In Section 2, the volume penalization method, the Lattice Boltzmann method, and the local mesh refinement are described. Also the whole computing procedure is given in this section. The numerical experiments and the comparison of results are given in Section 3. In Section 4, some concluding remarks and recommendations for the future work are presented.
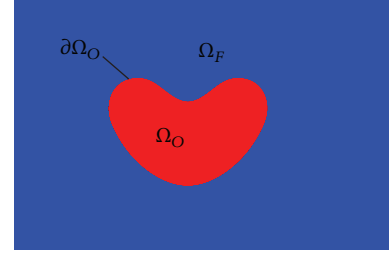

FIGURE 1: The computational domain of obstacle and fluid.

2. Mathematical and Numerical Formulation

In this section, the fluid-solid interaction (FSI) technique based on the volume penalization method is briefly introduced firstly. Then the incorporation of volume penalization method into the Lattice Boltzmann method and the LMR technique used in the Lattice Boltzmann are introduced in detail. Finally, a whole computing procedure is given.

2.1. Brief Review of the Volume Penalization Method and LBM. Let us consider the fluid-structure interaction (FSI) between incompressible viscous fluid and rigid obstacles in the fluid. The motion of the fluid is governed by the incompressible Navier-Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{F}_b, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

where $\mathbf{u}$ is the velocity of the fluid, μ is the dynamic viscosity, ρ is the density, p is the pressure, and $\mathbf{F}_b$ is the body force. The no-slip boundary conditions on the boundary of the obstacle domain Ω_O in the fluids can be described as

$$\mathbf{u}|_{\partial\Omega_O} = \mathbf{U}_O, \tag{3}$$

where $\partial\Omega_O$ is the boundary of the obstacles and $\mathbf{U}_O$ is the velocity of the obstacles. For problems involving only fixed obstacles are considered in the present method, the velocity of the obstacles $\mathbf{U}_O$ is equal to zero. The computational domain is shown in Figure 1. Ω_F is the fluid domain. The union of these two domains $\Omega = \Omega_F \cup \Omega_O$ is the entire domain.

The Dirichlet problem (1)–(3) can be solved by the volume penalization method [8, 21]. In the volume penalization method, the solid obstacles are modeled as porous media. By adding a penalization term on the velocity, the momentum equation (1) is modified as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{F}_b$$
$$- \frac{\chi(\mathbf{x},t)\rho}{\eta}(\mathbf{u} - \mathbf{U}_O), \tag{4}$$

where

$$\chi(\mathbf{x},t) = \begin{cases} 1, & x \in \Omega_O, \\ 0, & \text{other,} \end{cases} \tag{5}$$

is the mask function used to describe the obstacles' geometry and $\eta \ll 1$ is the penalization parameter. It can be seen that there is no Dirichlet boundary conditions in (4). The solution of the penalized N-S equation (4) tends towards the exact solution of N-S equation imposing no-slip boundary conditions with $\eta \rightarrow 0$ [8, 21, 22]. The hydrodynamic forces acting on the fixed obstacle can be obtained through

$$
\begin{aligned}
\mathbf{F}_O &= \lim_{\eta \rightarrow 0} \int_{\Omega} \frac{\chi}{\eta} \left(\mathbf{u} - \mathbf{U}_O \right) d\Omega \\
&= \lim_{\eta \rightarrow 0} \int_{\Omega_O} \frac{\chi \rho}{\eta} \left(\mathbf{u} - \mathbf{U}_O \right) d\Omega.
\end{aligned}
\tag{6}
$$

The N-S equation can be recovered by the Lattice Boltzmann equation through the multiscale Chapman-Enskog analysis. The discrete Boltzmann equation (DBE) with single-relaxation time (SRT) without a forcing term [23] can be written as

$$
\frac{\partial f_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla f_\alpha = -\frac{f_a - f_\alpha^{\text{eq}}}{\lambda},
\tag{7}
$$

in which f_α is the particle distribution function; t is the time; $\mathbf{e}_\alpha$ is the particle velocity in the αth direction; f_α^{eq} is the equilibrium distribution function, and λ is the relaxation parameter. For the nine-velocity lattice in two dimensions (D2Q9), the discrete velocity vectors can be defined as

$$
\begin{aligned}
&\mathbf{e}_\alpha \\
&= \begin{cases}
(0,0), & \alpha = 0, \\
(\cos \theta_\alpha, \sin \theta_a), & \theta_a = \frac{(\alpha - 1)\pi}{2}, \ \alpha = 1\text{--}4, \\
\sqrt{2} (\cos \theta_\alpha, \sin \theta_a), & \theta_a = \frac{(2\alpha - 9)\pi}{4}, \ \alpha = 5\text{--}8,
\end{cases}
\end{aligned}
\tag{8}
$$

and the equilibrium distribution function can be expressed as

$$
f_\alpha^{\text{eq}} = \rho w_\alpha \left[1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_a \cdot \mathbf{u})^2 - c_s^2 |\mathbf{u}|^2}{2c_s^4} \right],
\tag{9}
$$

in which $c_s = 1/\sqrt{3}$ is the sonic speed and the weight factors are $w_0 = 4/9$, $w_{1-4} = 1/9$, and $w_{5-8} = 1/36$. The macroscopic density, ρ, and velocity, $\mathbf{u}$, are defined as follows:

$$
\rho = \sum_\alpha f_\alpha,
\tag{10}
$$

$$
\rho \mathbf{u} = \sum_\alpha \mathbf{e}_\alpha f_a.
\tag{11}
$$

Equation (7) can be split into two substeps [20]: collision:

$$
g_\alpha = f_\alpha - \frac{f_\alpha - f_\alpha^{\text{eq}}}{\tau + 0.5},
\tag{12}
$$

and streaming:

$$
g_\alpha (\mathbf{x}, t + \Delta t) = g_\alpha (\mathbf{x} - \Delta \mathbf{x}_\alpha, t),
\tag{13}
$$

where τ is the single-relaxation parameter and τ is related to the kinematic viscosity υ through the equation $\tau = \upsilon/(c_s^2 \Delta t)$.

2.2. The Incorporation of VP into LBM.

The added penalization term on the velocity in the modified N-S equation can been regarded as an external force density,

$$
\mathbf{F}_{\text{VP}} = -\frac{\chi \rho}{\eta} \left(\mathbf{u} - \mathbf{U}_O \right),
\tag{14}
$$

acting on the fluid phase. Similarly, the DBE should be also modified by adding an external forcing term. In our article, the form of DBE with an external force term proposed by Guo et al. [11] is adopted. And the external force term is introduced into the collision substep. The collision substep is modified as

$$
g_\alpha = f_\alpha - \frac{f_\alpha - f_\alpha^{\text{eq}}}{\tau + 0.5} + F_\alpha \cdot \Delta t,
\tag{15}
$$

$$
F_\alpha = \left(1 - \frac{1}{2\lambda} \right) w_\alpha \left(\frac{\mathbf{e}_\alpha - \mathbf{u}}{c_s^2} + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^4} \cdot \mathbf{e}_\alpha \right) \cdot \mathbf{F}_{\text{VP}},
\tag{16}
$$

$$
\rho \mathbf{u} = \sum_\alpha \mathbf{e}_\alpha f_\alpha + \frac{1}{2} \mathbf{F}_{\text{VP}} \cdot \Delta t.
\tag{17}
$$

From (17), the fluid velocity is made up by two parts [5]. The first part is contributed by the density distribution function represented by the intermediate velocity $\mathbf{u}^*$. And the second part is contributed by the force density $\mathbf{F}_{\text{VP}}$ represented by $\delta \mathbf{u}$. The $\mathbf{u}^*$ can be written as

$$
\rho \mathbf{u}^* = \sum_\alpha \mathbf{e}_\alpha f_\alpha,
\tag{18}
$$

and the $\delta \mathbf{u}$ as

$$
\rho \delta \mathbf{u} = \frac{1}{2} \mathbf{F}_{\text{VP}} \Delta \mathbf{t}.
\tag{19}
$$

So (17) can be written as

$$
\mathbf{u} = \mathbf{u}^* + \delta \mathbf{u}.
\tag{20}
$$

Substituting (20) into (14), the force density can be expressed as

$$
\mathbf{F}_{\text{VP}} = -\frac{\chi \rho}{\eta} \left(\mathbf{u}^* + \delta \mathbf{u} - \mathbf{U}_O \right).
\tag{21}
$$

Then substituting (19) into (21), the second part of fluid velocity can be obtained as

$$
\delta \mathbf{u} = \frac{\chi (\mathbf{U}_O - \mathbf{u}^*)}{2\eta/\Delta t + \chi},
\tag{22}
$$

and the force density can be expressed as

$$
\mathbf{F}_{\text{VP}} = \frac{2\rho \chi (\mathbf{U}_O - \mathbf{u}^*)}{2\eta + \chi \Delta t}.
\tag{23}
$$

2.3. The Local Mesh Refinement Technique in LBM.

As proposed in [20], the streaming substep (13) is solved by using

○ Fine domain node
◯ Coarse domain node
⊠ In-line hanging node
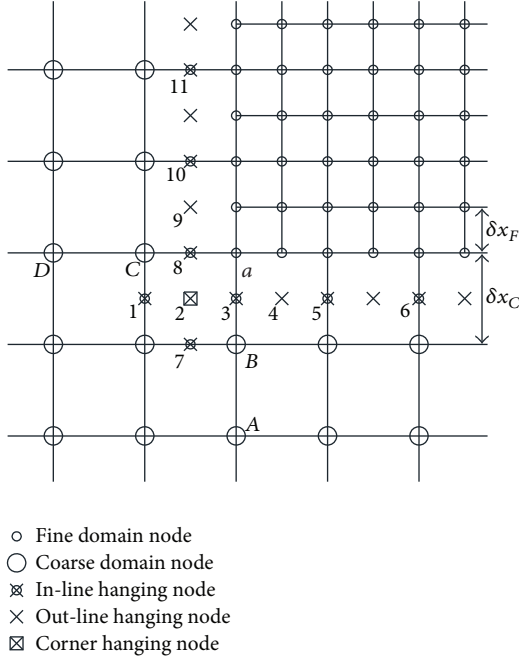× Out-line hanging node
⊠ Corner hanging node

FIGURE 2: Grid structure with difference refinement levels.

a Lax-Wendroff scheme with second-order accuracy in both time and space. The streaming substep can be rewritten as

$$g_\alpha(\mathbf{x}, t + \Delta t) = \frac{\sigma(1+\sigma)}{2} g_\alpha(\mathbf{x} - \Delta\mathbf{x}_\alpha, t)$$

$$+ \left(1 - \sigma^2\right) g_\alpha(\mathbf{x}, t) \qquad (24)$$

$$- \frac{\sigma(1-\sigma)}{2} g_\alpha(\mathbf{x} + \Delta\mathbf{x}_\alpha, t),$$

where $\sigma = |\mathbf{e}_\alpha|\Delta t/|\Delta\mathbf{x}_\alpha|$ is the Courant-Friedrichs-Lewy (CFL) number; Δt is the time step and $\Delta\mathbf{x}_\alpha$ is the grid spacing in the direction of $\mathbf{e}_\alpha$. It is can be seen in (24) that when the CFL number is equal to 1 ($\sigma = 1$) the streaming process becomes equivalent to the perfect shift $g_\alpha(\mathbf{x}, t + \Delta t) = g_\alpha(\mathbf{x} - \Delta\mathbf{x}_\alpha, t)$, that is, (13). When the CFL number is less than 1 ($\sigma < 1$), the streaming substep is solved with second-order accuracy in both time and space. It should also be pointed out that when the time step Δt is defined, the CFL number changes only due to the grid spacing.

Without loss of generality, a local fine-grid domain which is surrounded by a coarse-grid is considered, as shown in Figure 2. For the collision substep's localization feature, there is no need to transfer information between two different refinement levels grids. However, in the streaming substep, the transfer of information is required. To conduct this information transfer, some hanging nodes are needed. These hanging nodes can be classified into three types: in-line hanging node (ILHN), out-line hanging node (OLHN), and corner hanging node (CHN). The in-line hanging nodes will be calculated firstly. By using an interpolation scheme, the

density distribution function values of in-line hanging node 3 can be calculated as

$$f_\alpha(\mathbf{x} = \mathbf{x}_3, t) = \frac{3}{8} f_\alpha(\mathbf{x} = \mathbf{x}_a, t) + \frac{6}{8} f_\alpha(\mathbf{x} = \mathbf{x}_B, t)$$

$$- \frac{1}{8} f_\alpha(\mathbf{x} = \mathbf{x}_A, t). \qquad (25)$$

Similarly, all other in-line hanging nodes can be obtained. Then the out-line hanging node 4 can be calculated as

$$f_\alpha(\mathbf{x} = \mathbf{x}_4, t) = \frac{3}{8} f_\alpha(\mathbf{x} = \mathbf{x}_5, t) + \frac{6}{8} f_\alpha(\mathbf{x} = \mathbf{x}_3, t)$$

$$- \frac{1}{8} f_\alpha(\mathbf{x} = \mathbf{x}_1, t). \qquad (26)$$

After all other out-line hanging nodes are calculated, similarly, the out-line hanging node 2 is gotten. The corner hanging node 2 can be calculated:

$$f_\alpha(\mathbf{x} = \mathbf{x}_2, t) = \frac{1}{2} \left\{ \frac{3}{8} f_\alpha(\mathbf{x} = \mathbf{x}_1, t) + \frac{6}{8} f_\alpha(\mathbf{x} = \mathbf{x}_3, t) \right.$$

$$- \frac{1}{8} f_\alpha(\mathbf{x} = \mathbf{x}_4, t) \right\} + \frac{1}{2} \left\{ \frac{3}{8} f_\alpha(\mathbf{x} = \mathbf{x}_7, t) \right. \qquad (27)$$

$$\left. + \frac{6}{8} f_\alpha(\mathbf{x} = \mathbf{x}_8, t) - \frac{1}{8} f_\alpha(\mathbf{x} = \mathbf{x}_{10}, t) \right\}.$$

When density distribution function values at all hanging nodes are obtained, the streaming substep can be conducted.

In summary, the general steps of the algorithm are as follows.

(1) Design the computational grid, and arrange initial values on the computational grid.

(2) Use (15) to obtain the density distribution function after the collision substep of the $t = t_n$ step on the points of different refinement levels (initially setting $F_\alpha = 0$).

(3) Use (25), (26), and (27) to obtain the density distribution function on the hanging nodes.

(4) Use (24) to obtain the density distribution function on the nodes of computational grid.

(5) Use (10) and (18) to obtain the macroscopic density and intermediate velocity.

(6) Use (22) and (23) to obtain the velocity corrections and the force density.

(7) Correct the fluid velocity using (20) and compute the equilibrium distribution function using (9).

(8) Repeat step (2) to step (7) until the convergence is reached.

3. Numerical Results and Discussions

To verify the present Lattice Boltzmann-Volume Penalization with local mesh refinement, test cases involving incompressible viscous flows past single fixed circular cylinder are

chosen as numerical experiments. Then flows past a pair of circular cylinders in tandem arrangement and flows past NACA-0012 airfoil with 10° AOA and 15° AOA are conducted to validate the present method for complex boundaries. Some existing results are chosen as references for comparison. The computational efficiency is given as well.

In the following numerical experiments, the Reynolds number (Re) is defined as

$$Re = \frac{U_\infty D}{v}, \tag{28}$$

where U_∞ is the free stream velocity and D is the diameter of the cylinder and v is the kinematic viscosity of the fluid. The drag coefficient of cylinder is defined as

$$C_d = \frac{2F_D}{\rho U_\infty^2 D}, \tag{29}$$

where F_D is the drag force. The lift coefficient is defined as

$$C_l = \frac{2F_L}{\rho U_\infty^2 D}, \tag{30}$$

where F_L is the lift force. For the unsteady flow, the Strouhal number is defined as

$$S_t = \frac{f_q D}{U_\infty}, \tag{31}$$

where f_q is the vortex shedding frequency.

3.1. Flow Past a Fixed Cylinder.

In the experiments of this article, the density ρ of the fluid is set as 1.0. The free stream velocity U_∞ is 0.1. The diameter D of the cylinder is 40. The numerical simulations occupy a rectangle domain with the height H of 1024 and the length L of 2048. The center of the fixed cylinder is located at $(H/2, H/2)$, as shown in Figure 3. The computational meshes for these experiments are shown in Figure 4. For the cases of Re = 20 and 40, the flows will reach a steady state. Behind the fixed cylinder, a pair of stationary recirculating eddies will develop. With the Reynolds number increasing, the end of the wake will also move farther away from the rearmost point of the cylinder. The comparison of drag coefficient C_d, length of the recirculation region L (scaled by $D/2$), and separation angle θ_s with the results of previous literatures is shown in Table 1. The contour plots of velocities and streamlines are shown in Figure 5.

From Table 1, the present numerical results agree well with those in the previous studies. It can be seen in Figure 5 that the region of the recirculation becomes bigger when the Reynolds number grows from 20 to 40. These mean that the computational meshes set up for the cases of Re = 20, 40 are good enough to get correct results.

For the cases of Re = 100 and 200, the flows will reach a unsteady state. A Kármán vortex street will develop behind the cylinder. Also, a lift force will act on the cylinder. With the augmentation of Reynolds number, the vortices value and the area of vortices behind the cylinder will increase. The
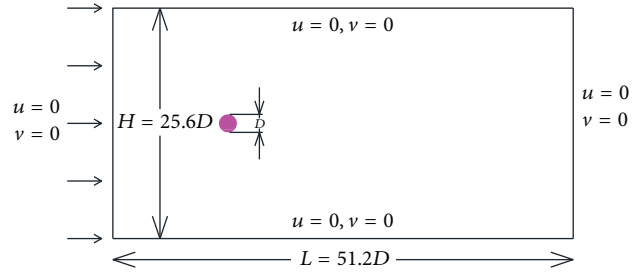

FIGURE 3: Computational domain and boundary conditions for flows past a fixed cylinder.

TABLE 1: Drag coefficient, length of bubbles, and separation angle for flows past a cylinder at Re = 20 and Re = 40.

Case	References	C_d	L_s	θ_s
	Niu et al. [24]	2.144	0.945	42.9°
	Shukla et al. [25]	2.07	0.92	43.2°
Re = 20	Wu and Shu [5]	2.091	0.93	-
	Present uniform-mesh	2.11	0.912	42.9°
	Present local-refined-mesh	2.11	0.913	42.9°
	Niu et al. [24]	1.589	2.26	53.86°
	Shukla et al. [25]	1.55	2.34	52.7°
Re = 40	Wu and Shu [5]	1.539	2.23	-
	Present uniform-mesh	1.57	2.23	52.8°
	Present local-refined-mesh	1.58	2.24	52.8°

drag coefficient, the lift coefficient, and Strouhal number are compared with those of previous studies in Table 2. From Table 2, it can be found that a good agreement is gotten between the presents results and those in previous studies. The evolution of drag and lift coefficients for the cases of Re = 100 and 200 are given in Figure 6, and the streamlines and velocity contours are shown in Figure 7.

3.2. Flow Past Two Tandem Arranged Cylinders.

To verify the capability of the present method to simulate complex flows, flows past a pair of circular cylinders tandem arranged, which have been studied by many researchers [28–31], are chosen as experiments. Compared with the flows past a single circular cylinder, the flows past a pair of cylinders tandem arranged are more complicated. For this problem, the distance between these two cylinders represented by Lg/D, as shown in Figure 8, plays an extremely important role in the development of vertex structures of the flows, as well as the evolution of drag and lift coefficients. In Figure 8, the computational domain and boundary conditions are presented, and the computational meshes are shown in Figure 9. For the presented experiments, the Reynolds number Re is set as 200. For different Lg/D experiments conducted, $Lg/D = 1.5, 2.0, 3.0, 4.0$.

From Figure 10, it can be seen that when Lg/D is set as $1.5, 2.0, 3.0$, vortex shedding develops behind the downstream cylinder. But between the upstream cylinder
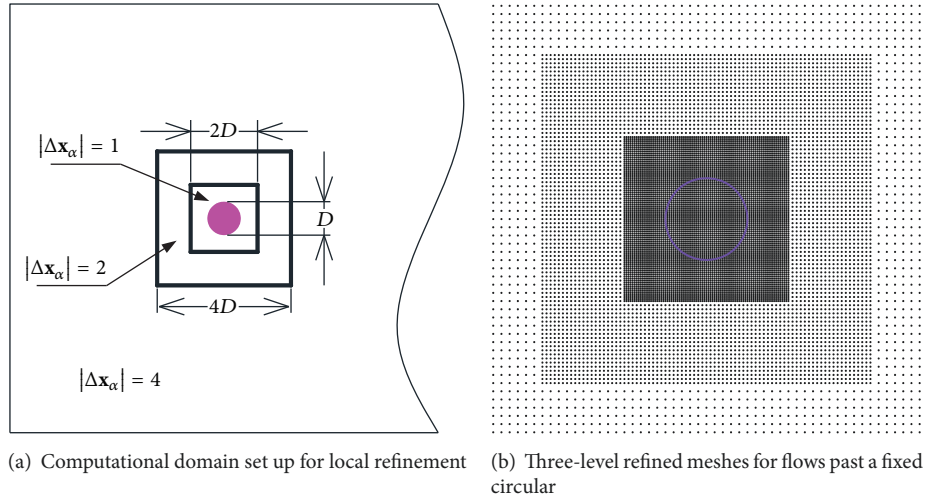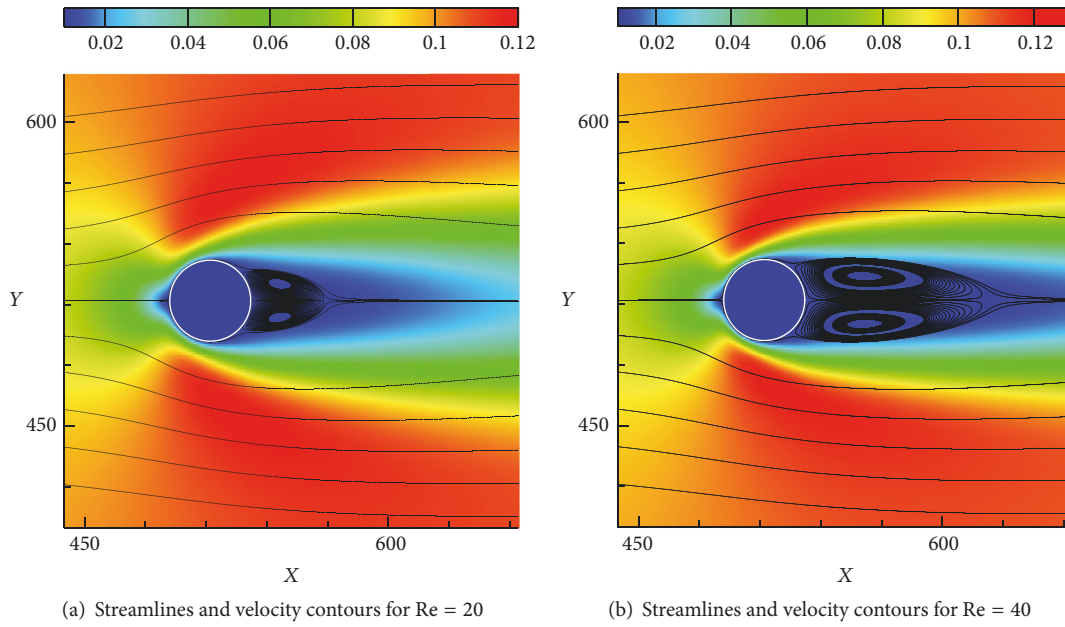
(a) Computational domain set up for local refinement

(b) Three-level refined meshes for flows past a fixed circular

FIGURE 4: Computational meshes for flows past a cylinder.

(a) Streamlines and velocity contours for Re = 20

(b) Streamlines and velocity contours for Re = 40

FIGURE 5: Steady flow past a cylinder at Re = 20 and Re = 40.

TABLE 2: Drag coefficient, length of bubbles, and separation angle for flows past a cylinder at Re = 100 and Re = 200.

Case	References	C_d	C_l	S_t
Re = 100	Benson et al. [26]	1.46 ± 0.01	±0.38	0.17
	Ding et al. [27]	1.325 ± 0.008	±0.28	0.164
	Wu and Shu [5]	1.334 ± 0.012	±0.37	0.163
	Present uniform-mesh	1.361 ± 0.099	±0.345	0.166
	Present local-refined-mesh	1.361 ± 0.099	±0.346	0.166
Re = 200	Benson et al. [26]	1.45 ± 0.04	±0.65	0.193
	Ding et al. [27]	1.327 ± 0.045	±0.60	0.196
	Wu and Shu [5]	1.43 ± 0.051	±0.75	0.195
	Present uniform-mesh	1.337 ± 0.046	±0.688	0.196
	Present local-refined-mesh	1.338 ± 0.045	±0.688	0.196
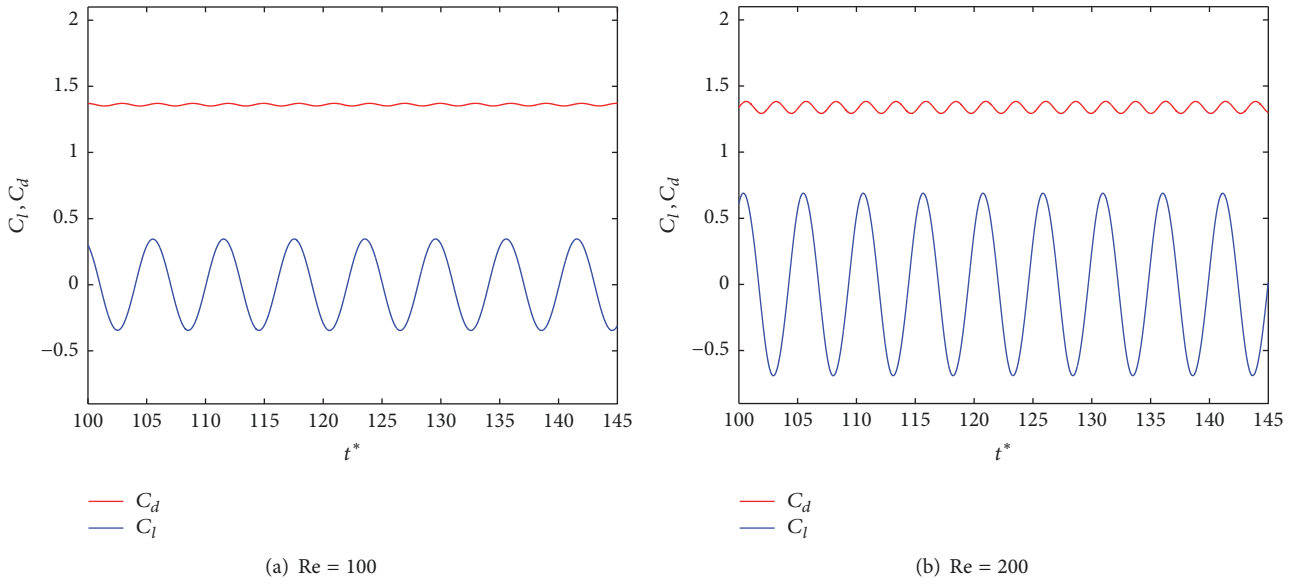
(a) Re = 100

(b) Re = 200

FIGURE 6: Evolution of drag and lift coefficients at Re = 100 and Re = 200.

(a) Streamlines and velocity contours for Re = 100

(b) Streamlines and velocity contours for Re = 200
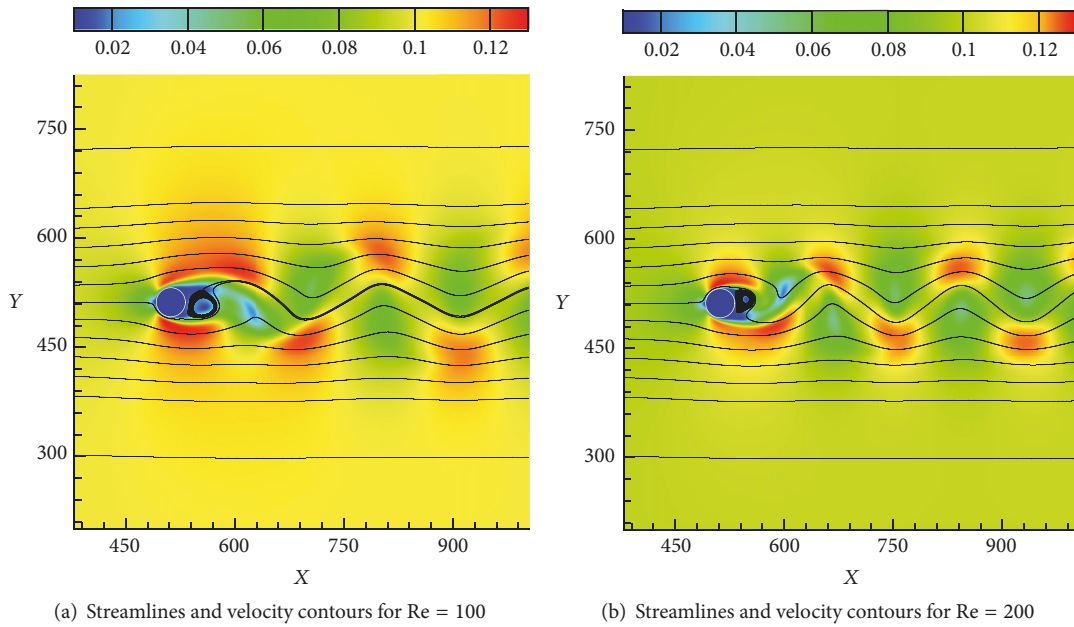
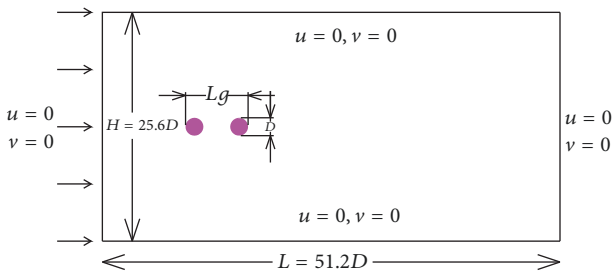FIGURE 7: Unsteady flow past a cylinder at Re = 100 and Re = 200.

FIGURE 8: Computational domain and boundary conditions for flows past two tandem arranged cylinders.

and the downstream cylinder, there is no vortex shedding. For these three experiments, the drag coefficients of the upstream cylinders are positive, as shown in Figure 11, but the downstream cylinders' are negative. The drag coefficient amplitudes of both upstream and downstream cylinders reduced further with the augment of the distance between two cylinders. When Lg/D turns to 4.0, great changes happen. The vortex shedding develops not only behind the downstream cylinder but also between the upstream cylinder and the downstream cylinder. The drag coefficient becomes positive. For all the four experiments, the drag coefficients
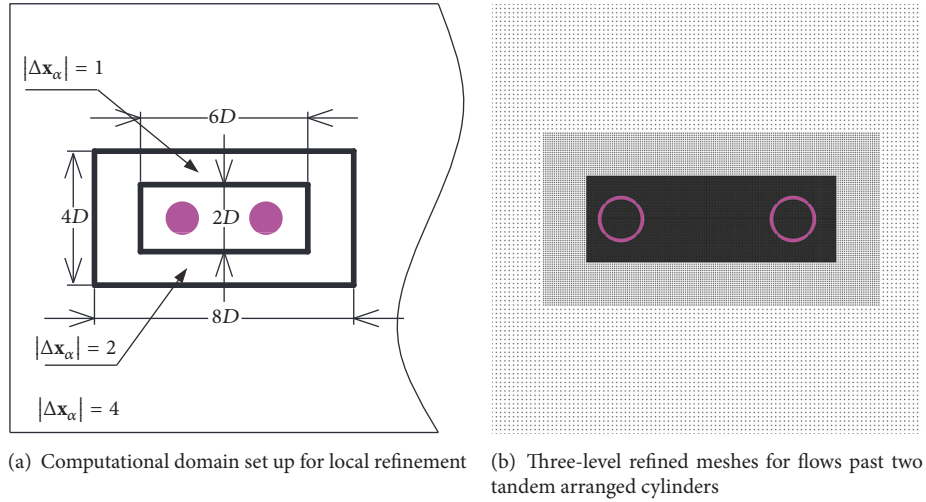
(a) Computational domain set up for local refinement

(b) Three-level refined meshes for flows past two tandem arranged cylinders

FIGURE 9: Computational meshes for flows past a pair of circular cylinders in tandem arrangement.

(a) $Lg = 1.5$

(b) $Lg = 2.0D$
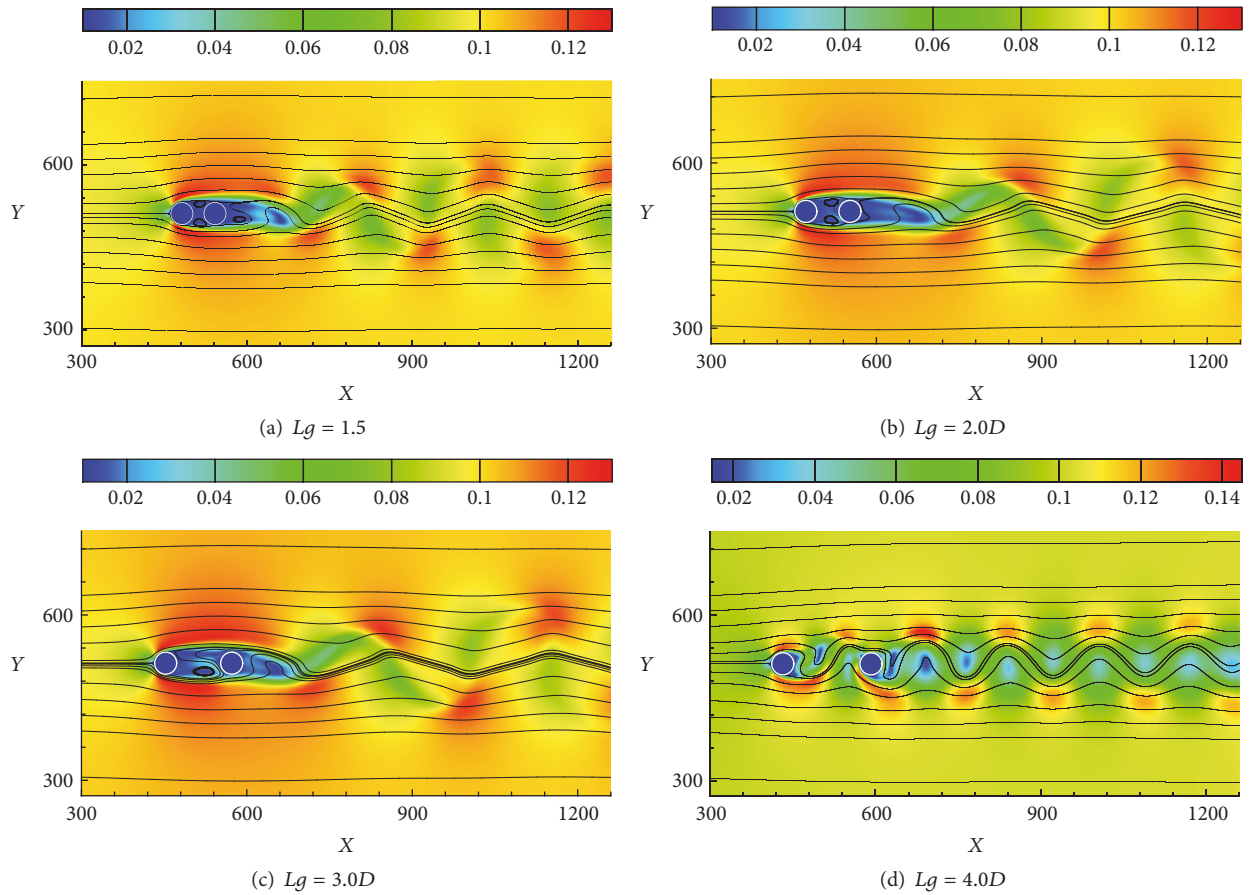
(c) $Lg = 3.0D$

(d) $Lg = 4.0D$

FIGURE 10: Streamlines and velocity contours for flows past two cylinders in tandem arrangement.

of upstream and downstream cylinders oscillate at the same frequency, which means the Strouhal numbers are equal. The comparison of the present results with the data in previous literatures is represented in Table 3, from which a good agreement can be seen.

3.3. Flow Past a NACA-0012 Airfoil. For the practical application of the LBM-VP with local mesh refinement, the flows past a NACA-0012 airfoil at Re = 500 with AOA = 0° and at Re = 1000 with AOA = 15° are selected for numerical experiments. Just like in the previous numerical experiments,

(a) $Lg = 1.5$

(b) $Lg = 2.0$

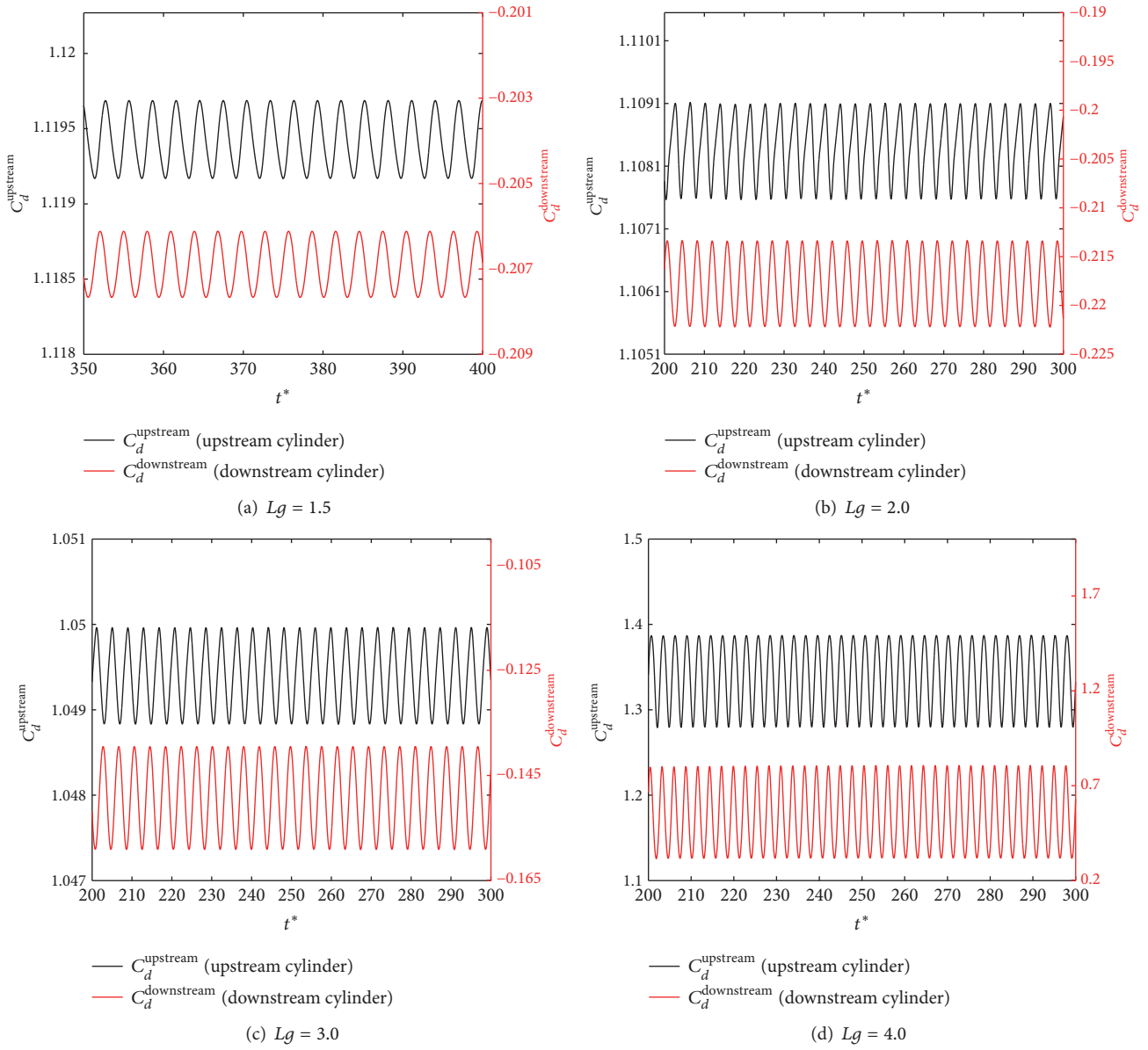(c) $Lg = 3.0$

(d) $Lg = 4.0$

FIGURE 11: Evolution of drag coefficients of two cylinders in tandem arrangement.

TABLE 3: Drag coefficient and Strouhal number for flows past a pair of circular cylinders in tandem arrangement at Re = 200.

	Lg/D	1.5	2.0	3.0	4.0
	Meneghini et al. [28]	1.06	1.03	1.00	1.18
Upstream cylinder	Hu et al. [29]	1.158	1.126	1.080	1.355
	Present uniform-mesh	1.120	1.107	1.050	1.338
	Present local-refined-mesh	1.120	1.108	1.049	1.339
	Meneghini et al. [28]	−0.18	−0.17	−0.08	0.38
Downstream cylinder	Hu et al. [29]	−0.197	−0.209	−0.140	0.582
	Present uniform-mesh	−0.206	−0.216	−0.150	0.545
	Present local-refined-mesh	−0.206	−0.217	−0.149	0.545

TABLE 4: Comparisons of the computing time per step and number of the used points.

	Case study	Uniform-mesh	Local-refined-mesh	Speedup
Time	Flows past a single circular cylinder	6.50	0.624	10.41
	Flows past a pair of circular cylinders	6.50	0.636	10.22
	Flows past a NACA-0012 airfoil	6.50	0.640	10.16
Number of the used points	Flows past a single circular cylinder	1024 × 2048	141601	14.81
	Flows past a pair of circular cylinders	1024 × 2048	156121	13.43
	Flows past a NACA-0012 airfoil	1024 × 2048	162791	12.88
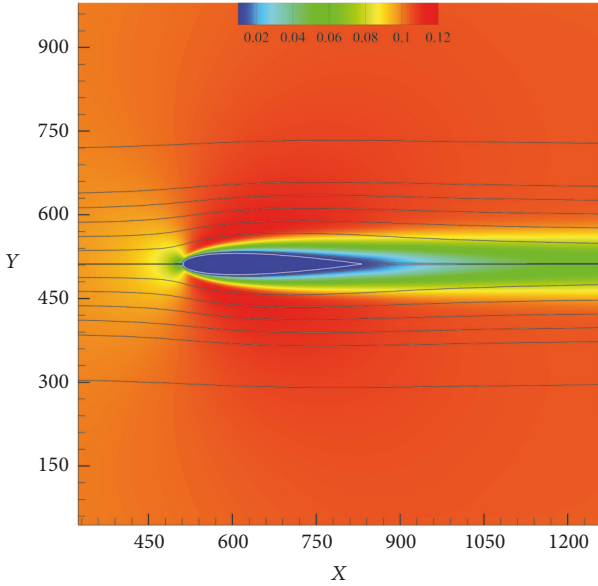
FIGURE 12: Streamline and velocity contours for flows past a NACA-0012 airfoil at Re = 500 with AOA = 0°.
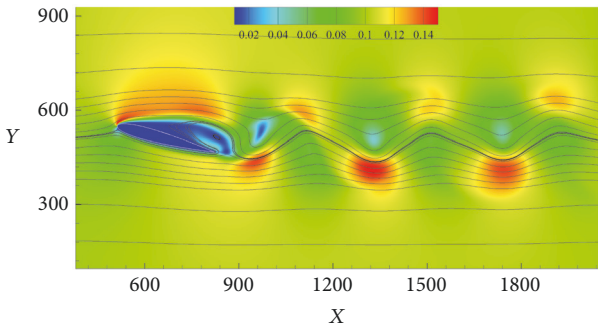
FIGURE 13: Streamline and velocity contours for flows past a NACA-0012 airfoil at Re = 1000 with AOA = 15°.

the fluid density is set as $\rho = 1.0$ and the free stream velocity is 0.1. The chord of the airfoil is 320.

The streamlines and the velocity contours for flows past the NACA-0012 airfoil are shown in Figures 12 and 13. The flow behind the NACA-0012 airfoil at Re = 500 with zero angle of attack will reach a steady state. And the drag coefficient of the present experiment is 0.179, which agrees well with the results given by Lockard et al. [32] with 0.1762 and Wu and Shu [5] with 0.1759. For the case of Re = 1000

with AOA = 15°, the flow becomes unsteady, and a lift force will act on the airfoil. In the present experiment, the Strouhal number for the drag coefficient is 0.72, while the value given by Suzuki et al. [33] is 0.73 and that given by Kurtulus [34] is 0.712. Obviously, a good agreement is gotten between our result and those in the previous literatures. From the two experiments, it can be concluded that the present LBM-VP with local mesh refinement can solve the practical problems with complex boundaries.

To demonstrate the advantage of the LBM-VP with local mesh refinement, we compare the computing time and the number of the used mesh points of the LBM-VP on the local refined mesh with those on the uniform mesh code. All the experiments are conducted by eight threads on a PC with Intel(R) Core(TM) i7-4790K 4.00 GHz CPU and 16.0 GB RAM. In Table 4, the computing time per step and the number of the used points of the LBM-VP on local refined mesh are compared with the uniform mesh code for all the experiments in this article. For all the experiments, it can be seen that the local mesh refinement technique used in the LBM-VP can reduce the computing time per step.

4. Conclusions

In this article, the coupled Lattice Boltzmann-Volume penalization method (LBM-VP) with local mesh refinement is presented to simulate flows past obstacles. Compared to the direct-forcing IBM and the velocity IBM, there is no need to use a delta function to interpolate the velocity at boundaries and to distribute the force density into the Eulerian points near the boundaries. On a certain Lagrangian point, only the variables of this certain Euler point are needed to perform the VP procedure, which means that the LBM-VP can be conducted parallelly. By incorporating the local mesh refinement technique based on the finite-difference LBM, the LBM-VP are performed on the local refined mesh. As a result, the number of used mesh points is reduced, thus improving the computing efficiency. To justify the present LBM-VP with local mesh refinement, flows past a single circular cylinder with Re = 20, 40, 100, 200 are carried out firstly. Then slightly more complex experiments, flows past a pair of circular cylinders in tandem arrangement, are studied. For the practical application of the present method, the flows past a NACA-0012 airfoil at Re = 500 with AOA = 0° and at Re = 1000 with AOA = 15° are chosen as experiments.

By comparing the numerical experiment results with the data in previous literatures, our simulations show good

agreement with available data. And the local mesh refinement technique reduces the number of used points of the LBM-VP. As a result, the computing time of per step is reduced, meaning that the computing efficiency improved. Although the current method is developed in 2D, it is easy to extend it to 3D by replacing D2Q9 lattice with D3Q15 or D3Q19.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] S. Chen and G. D. Doolen, "Lattice boltzmann method for fluid flows," *Annual Review of Fluid Mechanics*, vol. 30, no. 1, pp. 329–364, 1998.

[2] A. Piquet, O. Roussel, and A. Hadjadj, "A comparative study of brinkman penalization and direct-forcing immersed boundary methods for compressible viscous flows," in *Computers and Fluids*, vol. 136, C, pp. 272–284, 2016.

[3] C. S. Peskin, "The immersed boundary method," *Acta Numerica*, vol. 11, pp. 479–517, 2002.

[4] Z.-G. Feng and E. E. Michaelides, "The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems," *Journal of Computational Physics*, vol. 195, no. 2, pp. 602–628, 2004.

[5] J. Wu and C. Shu, "Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications," *Journal of Computational Physics*, vol. 228, no. 6, pp. 1963–1979, 2009.

[6] M. Benamour, E. Liberge, and C. Béghein, "Lattice Boltzmann method for fluid flow around bodies using volume penalization," *The International Journal of Multiphysics*, vol. 9, no. 3, pp. 299–316, 2015.

[7] E. Arquis and J. Caltagirone, "Sur les conditions hydrodynamiques au voisinage dune interface milieu fluide-milieu poreux: Application la convection naturelle," *C. R. Acad. Sci. Paris*, vol. 299, no. 1, 1984.

[8] P. Angot, C.-H. Bruneau, and P. Fabrie, "A penalization method to take into account obstacles in incompressible viscous flows," *Numerische Mathematik*, vol. 81, no. 4, pp. 497–520, 1999.

[9] N. K. Kevlahan and J. Ghidaglia, "Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization," *European Journal of Mechanics - B/Fluids*, vol. 20, no. 3, pp. 333–350, 2001.

[10] K. Schneider, "Numerical simulation of the transient flow behaviour in chemical reactors using a penalisation method," *Computers and Fluids*, vol. 34, no. 10, pp. 1223–1238, 2005.

[11] Z. Guo, C. Zheng, and B. Shi, "Discrete lattice effects on the forcing term in the lattice Boltzmann method," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 65, no. 4, Article ID 046308, 2002.

[12] X. He, L. Luo, and M. Dembo, "Some progress in lattice boltzmann method. Part I. nonuniform mesh grids," *Journal of Computational Physics*, vol. 129, no. 2, pp. 357–363, 1996.

[13] T. Imamura, K. Suzuki, T. Nakamura, and M. Yoshida, "Acceleration of steady-state lattice Boltzmann simulations on non-uniform mesh using local time step method," *Journal of Computational Physics*, vol. 202, no. 2, pp. 645–663, 2005.

[14] C. Shu, X. D. Niu, and Y. T. Chew, "Taylor series expansion and least squares-based lattice boltzmann method: Three-dimensional formulation and its applications," *International Journal of Modern Physics C*, vol. 14, no. 07, pp. 925–944, 2003.

[15] M. B. Reider and J. D. Sterling, "Accuracy of discrete-velocity BGK models for the simulation of the incompressible Navier-Stokes equations," *Computers and Fluids*, vol. 24, no. 4, pp. 459–467, 1995.

[16] S. Succi, G. Amati, and R. Benzi, "Challenges in lattice Boltzmann computing," *Journal of Statistical Physics*, vol. 81, no. 1-2, pp. 5–16, 1995.

[17] J. Wu and C. Shu, "A solution-adaptive lattice Boltzmann method for two-dimensional incompressible viscous flows," *Journal of Computational Physics*, vol. 230, no. 6, pp. 2246–2269, 2011.

[18] A. Fakhari and T. Lee, "Finite-difference lattice Boltzmann method with a block-structured adaptive-mesh-refinement technique," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 89, no. 3, 2014.

[19] X. Cui, X. Yao, Z. Wang, and M. Liu, "A hybrid wavelet-based adaptive immersed boundary finite-difference lattice Boltzmann method for two-dimensional fluid-structure interaction," *Journal of Computational Physics*, vol. 333, pp. 24–48, 2017.

[20] T. Lee and C. Lin, "An Eulerian description of the streaming process in the lattice Boltzmann equation," *Journal of Computational Physics*, vol. 185, no. 2, pp. 445–471, 2003.

[21] G. Carbou and P. Fabrie, "Boundary layer for a penalization method for viscous incompressible flow," *Advances in Differential Equations*, vol. 8, no. 12, pp. 1453–1480, 2003.

[22] T. Engels, D. Kolomenskiy, K. Schneider, and J. Sesterhenn, "Numerical simulation of fluid-structure interaction with the volume penalization method," *Journal of Computational Physics*, vol. 281, pp. 96–115, 2015.

[23] X. He and L. Luo, "Lattice boltzmann model for the incompressible navier–stokes equation," *Journal of Statistical Physics*, vol. 88, no. 3, pp. 927–944, 1997.

[24] X. Niu, C. Shu, Y. Chew, and Y. Peng, "A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows," *Physics Letters A*, vol. 354, no. 3, pp. 173–182, 2006.

[25] R. K. Shukla, M. Tatineni, and X. Zhong, "Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 224, no. 2, pp. 1064–1094, 2007.

[26] M. G. Benson, P. G. Bellamy-Knights, J. H. Gerrard, and I. Gladwell, "A viscous splitting algorithm applied to low Reynolds number flows round a circular cylinder," *Journal of Fluids and Structures*, vol. 3, no. 5, pp. 439–479, 1989.

[27] H. Ding, C. Shu, K. S. Yeo, and D. Xu, "Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method," *Computer Methods Applied Mechanics and Engineering*, vol. 193, no. 9-11, pp. 727–744, 2004.

[28] J. R. Meneghini, F. Saltara, C. L. R. Siqueira, and J. A. Ferrari Jr., "Numerical simulation of flow interference between two circular cylinders in tandem and side-by-side arrangements," *Journal of Fluids and Structures*, vol. 15, no. 2, pp. 327–350, 2001.

[29] Y. Hu, H. Yuan, S. Shu, X. Niu, and M. Li, "An improved momentum exchanged-based immersed boundary attice boltzmann method by using an iterative technique," *Computers and Mathematics with Applications*, vol. 68, no. 3, pp. 140–155, 2014.

[30] M. M. Zdravkovich, "REVIEW—review of flow interference between two circular cylinders in various arrangements," *Journal of Fluids Engineering*, vol. 99, no. 4, pp. 618–633, 1977.

[31] T. Igarashi, "Characteristics of the flow around two circular cylinders arranged in tandem: 1st report," *Journal of the Japan Society of Mechanical Engineers*, vol. 24, no. 188, pp. 323–331, 1981.

[32] D. P. Lockard, L.-S. Luo, S. D. Milder, and B. A. Singer, "Evaluation of powerflow for aerodynamic applications," *Journal of Statistical Physics*, vol. 107, no. 1, Article ID 1014539411062, pp. 423–478, 2002.

[33] T. Suzuki, H. Ji, and F. Yamamoto, "Unsteady PTV velocity field past an airfoil solved with DNS: Part 1. Algorithm of hybrid simulation and hybrid velocity field at Re $\approx$ 103," *Experiments in Fluids*, vol. 47, no. 6, pp. 957–976, 2009.

[34] D. F. Kurtulus, "On the unsteady behavior of the flow around NACA 0012 airfoil with steady external conditions at Re=1000," *International Journal of Micro Air Vehicles*, vol. 7, no. 3, pp. 301–326, 2015.