

Improving lattice Boltzmann simulation of moving particles in a viscous flow using local grid refinement



Songying Chen^{a,*}, Cheng Peng^b, Yihua Teng^c, Lian-Ping Wang^{b,d}, Kun Zhang^a

^a School of Mechanical Engineering, Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Shandong University, Jinan 250062, P.R. China

^b Department of Mechanical Engineering, 126 Spencer Laboratory, University of Delaware, Newark, Delaware 19716-3140, USA

^c Department of Energy and Resource Engineering, Peking University, Beijing, P.R. China

^d National Laboratory of Coal Combustion, Huazhong University of Science and Technology, Wuhan, P.R. China

ARTICLE INFO

Article history:

Received 4 December 2015

Revised 1 June 2016

Accepted 7 June 2016

Available online 9 June 2016

Keywords:

Lattice Boltzmann method

Local grid refinement

Moving particles

Force fluctuation

Viscous flow

ABSTRACT

Accurate simulations of moving particles in a viscous flow require an adequate grid resolution near the surface of a moving particle. Within the framework of the lattice Boltzmann approach, inadequate grid resolution could also lead to numerical instability and large fluctuations of the computed hydrodynamic force and torque. Here we explore the use of local grid refinement around a moving particle to improve the simulation results using the multiple-relaxation-time (MRT) lattice Boltzmann method (LBM). We first re-examine the necessary relationships, within MRT LBM, between the relaxation parameters and the distribution functions on the coarse and fine grids, in order to meet the physical requirements of the fluid hydrodynamics. We also propose additional relationships based on the Chapman–Enskog multi-scaling analysis. Several aspects of the implementation details are discussed, including the treatment of interface buffer nodes, the method to transfer information between the coarse domain and fine domain, and the computation of macroscopic variables including stress components. Our approach is then applied in two numerical tests to demonstrate that the local grid refinement can significantly improve the physical results with a high computational efficiency. We compare simulation results from three grid configurations: a uniformly coarse grid, a uniformly coarse grid with local refinement, and a uniformly fine grid. For the lid-driven cavity flow, the local refinement essentially yields a local flow field that is comparable to the use of uniformly fine grid, but with much less computational cost. In the Couette flow with a moving cylinder, the local refinement suppresses the level of force fluctuations. In these tests, the stress profiles are carefully examined to help illustrate the benefits of local grid refinement. We also confirm that the coarse-fine grid relationships between the non-equilibrium moments of energy square and energy fluxes do not affect the simulation results.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

As a highly efficient and capable mesoscopic computational method, the lattice Boltzmann method (LBM) [1–4] has been widely employed to solve a variety of fluid dynamics problems. The standard LBM describes the fluid as made up by imaginative elements which can stream along a uniform lattice grid and collide with one another only at lattice nodes. The method solves a quasi-linear collision-streaming equation for a set of distribution functions associated with discrete microscopic velocities. The macro-

scopic hydrodynamic variables such as pressure and velocity are obtained by taking the moments of the distribution functions.

One of the popular LBM schemes is based on the single relaxation time approach (*i.e.*, Bhatnagar–Gross–Krook collision operator [5]), which is known as the LBGK model. Another popular scheme is based on the multiple-relaxation-time (MRT) collision model [6,7]. The MRT collision is performed in the moment space with different moments relaxing at different rates. By decomposing the particle relaxation process into several independent relaxation processes, the MRT model has been shown to not only improve the computational stability but also the accuracy [6–8].

In order to obtain accurate macroscopic quantities, such as force and torque acting on a solid particle or the boundary, small grid spacing is needed near the solid particle or a boundary. Away from the boundary or fluid–solid interfaces, the flow may be more smoother so a coarser grid is adequate to resolve the flow. The

* Corresponding author. Fax: +86053188988791.

E-mail addresses: chensy66@sdu.edu.cn (S. Chen), cpengxpp@udel.edu (C. Peng), tengyihua99@gmail.com (Y. Teng), lwang@udel.edu (L.-P. Wang), 254509511@qq.com (K. Zhang).

most efficient approach in terms of both memory and overall accuracy is thus to use a coarse grid for most of the bulk flow region, combined with a local grid refinement near a fluid-solid interface or wall boundary.

Within the LBGK model, local grid refinement has been considered for some time to simulate incompressible viscous flows with complex geometries. Filippova and Hanel [9] was among the first to consider patching certain regions with a fine grid in a domain mostly covered by a coarse grid, values of the distribution functions on the coarse grid which are coming from the fine patches are calculated on the nodes common to both grids. Filippova and Hanel [10] presented an accelerated implementation of grid refinement by using different particle speeds on the coarse and fine grids. A smaller time step size was used on the fine grid while the spatial or temporal accuracy was kept. Steady-state and time-dependent problems were studied and the CPU time per time step was reduced by about 50% relative to the results computed by the standard LBM. Yu et al. [11] proposed a multi-block technique with the LBGK model. Different mesh sizes are used for different blocks that do not overlap. Macro-variables such as mass, momentum, and stress components are assumed to be continuous across the block-block interface, and this condition determines the relaxation parameters in the fine domain. The cubic spline scheme was used for spatial interpolation and the three-point Lagrangian formula was used for temporal interpolation on all nodes at the fine block boundary after the distribution functions are transferred from the coarse domain to the fine domain. Yu and Girimaji [12] extended their approach to 3D using the LBGK model. Two 3D test cases, an isotropic decaying turbulence and a lid-driven cavity flow, were presented to show the improved computational efficiency. Farhat and Lee [13] was the first to suggest a migrating multi-block scheme to combine with the Gunstensen model for immiscible mixtures in 2D geometries. A fine grid block covered the entire fluid interface and was allowed to migrate by tracking the center of mass of the body. The upstream and downstream coarse blocks are separated by the fine block. After the fine domain shifted, extrapolation was used to compute the variables at all the newly created fine and coarse nodes as well as the old diminishing coarse nodes. Two benchmark simulations, namely, single phase flow around an asymmetrically placed cylinder in a channel and the motion of a neutrally buoyant drop in a parabolic flow, were simulated to validate their model.

A more general and efficient approach is adaptive grid refinement, where the computational grid is modified dynamically, with as many refinement levels as needed and each level can be of arbitrary shape. Typically, a physical criterion is applied to determine whether a local grid cell should be refined further or can be coarsened. An early example of adaptive grid refinement applied within LBGK is the study of Crouse et al. [14] in which they used a local divergence sensor to dynamically adjust the local grid spacing. Eitel-Amor et al. [15] introduced a cell-centered lattice structure to reconstruct the pre-collision distribution functions via spatial interpolation in LBGK model. They showed that, with hierarchically refined meshes, each cell can be refined or coarsened regardless of the refinement level of neighbor cells. Lagrava [16] introduced a decimation technique to guarantee the stability of the numerical scheme especially at high flow Reynolds number when the information is transferred from the coarse nodes to the fine nodes. Dietzel and Sommerfeld [17] calculated flow resistance over agglomerates with different morphology through LBGK local grid refinement. They slightly overlapped the coarse and fine regions and designed a method to communicate the distribution functions between the two grids at the interface. Premnath et al. [18] presented a staggered mesh arrangement in large-eddy simulation of a complex turbulent separated flow, using the MRT D3Q19 model. Sub-grid scale model was employed in conjunction with the MRT to

augment the relaxation time scales of hydrodynamic modes, in order to represent the effect of subgrid scale fluid motion.

One of the first attempts to implement local grid refinement within MRT was performed by Peng et al. [19] using the D2Q9 model, where they related the distribution functions and relaxation parameters in the two domains based on the continuity of macro-variables at the coarse-fine interface. The method to communicate distributions functions between the two domains was derived. They used lid-driven cavity flow, steady and unsteady flows past a circular cylinder, and flow over an airfoil to validate their approach. As will be shown later in this paper, their implementation was not fully consistent since they did not explicitly consider the relationship of energy relaxation parameters between the fine and coarse domains. Geller et al. [20,21] considered local grid refinement within MRT and they noted that only the two stress moments in the D2Q9 model between the two grid levels need to be properly rescaled, while other relaxation times can be made flexible. They considered pre-collision rescaling in 2D. The same group later extended local grid refinement to 3D LBM-MRT models under the same strategy in order to simulate multi-phase flow with deformable interfaces [22], or to perform large-eddy simulations of a turbulent flow around a sphere in a channel [23] or a turbulent jet flow [24]. The rescaling in these papers was also performed before collision. Grid refinements using MRT have also been considered in another recent study [25], without a careful consideration of the full flexibility in MRT to relate the distribution functions on fine and coarse grids. A shifting discontinuous-grid-block lattice Boltzmann method for moving boundary simulations using MRT model was recently reported by Arora et al. [26], where the fine domain moves with the moving body. Three simulation cases, namely, a cylinder in a shear flow, a single wing employing 'clap and fling' motion, and rigid plunging flat plate, are presented to show the accuracy of their model.

The above literature survey indicates that there are two general ways of communicating the distribution functions at the fine-coarse domain interface, namely, post-collision rescaling and pre-collision rescaling [9,19]. The connection of the rescaling process to the Chapman-Enskog expansion for the MRT model has not been fully explored. Our first objective is to re-examine the details of coupling the distribution functions and model parameters between the fine and coarse grids within the MRT framework. Our second objective is to test local grid refinement in moving particle simulation. In Section 2, a brief background description of the D2Q9 MRT model is provided. Details of local grid refinement implementation are discussed in Section 3, together with the development of relationships between the coarse and fine grid. Our implementation follows in spirit the previous studies of Tölke et al. [20,22]. Necessary interpolation details at the coarse-fine interface are presented in Section 4. In section 5, we then validate our methodology using 2D lid-driven cavity flow, and a Couette flow with a fixed or a moving cylinder. In the case of the moving cylinder, the refined region also moves with the cylinder. Test results demonstrate that local grid refinement indeed improves the accuracy in the moving particle simulation. We also compare CPU time and memory consumption when different grid arrangements (e.g., hybrid coarse / fine grid versus uniformly fine) are used. Key conclusions are summarized in Section 6.

2. The multiple-relaxation-time lattice Boltzmann method

In this section, we briefly introduce the multiple-relaxation-time (MRT) lattice Boltzmann method (LBM) in order to prepare for the discussions on the local grid refinement. The detailed description of MRT LBM can be found in [6,7].

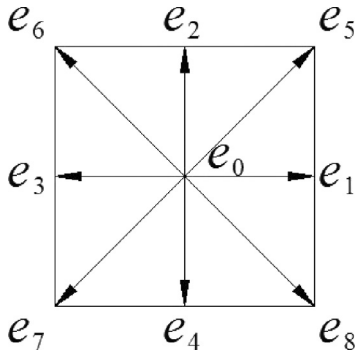


Fig. 1. The D2Q9 model with nine discrete velocities on two dimensional square lattice.

Specifically the D2Q9 model [27,28] is considered (Fig. 1), with discrete velocities given by:

$$e_i = \left\{ \begin{matrix} (0, 0) & (1, 0) & (0, 1) & (-1, 0) & (0, -1) \\ (1, 1) & (-1, 1) & (-1, -1) & (1, -1) & \end{matrix} \right\} c. \quad (1)$$

where $i = 0, 1, 2, \dots, 8$, $c = \delta x / \delta t$. All variables are given in lattice units such that $c = 1$. The MRT LBM evolution equation [6,7] can be written as

$$\mathbf{f}(\mathbf{x} + \mathbf{e}_i \delta t, t + \delta t) = \mathbf{f}(\mathbf{x}, t) - M^{-1} S [\mathbf{m} - \mathbf{m}^{eq}]. \quad (2)$$

where \mathbf{f} is a vector representing a set of distribution functions defined at a lattice node, \mathbf{m} represents a set of independent moments, \mathbf{m}^{eq} is the equilibrium of \mathbf{m} , M is an orthogonal transformation matrix that transforms \mathbf{f} into \mathbf{m}

$$\mathbf{m} = M \mathbf{f}, \quad \mathbf{f} = M^{-1} \mathbf{m}. \quad (3)$$

The macroscopic hydrodynamic variables, including density ρ , velocity $\mathbf{u} = (u_x, u_y)$, and pressure p are obtained from the moments of the mesoscopic distribution function \mathbf{f} . We use the nearly incompressible formulation [6], namely, the density is partitioned as $\rho = \rho_0 + \delta\rho$ with $\rho_0 = 1$, and $\delta\rho = \sum_i f_i$, $\rho_0 \mathbf{u} = \sum_i f_i \mathbf{e}_i$, and $p = \delta\rho c_s^2$, where $c_s = 1/\sqrt{3}$ is the model speed of sound. For the D2Q9 MRT model, other moments and the transformation matrix are designed as

$$\mathbf{m} = \left\{ \begin{matrix} \rho \\ e \\ \varepsilon \\ j_x \\ q_x \\ j_y \\ q_y \\ p_{xx} \\ p_{xy} \end{matrix} \right\} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix} \left\{ \begin{matrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \end{matrix} \right\}. \quad (4)$$

The three hydrodynamic moments, ρ , $j_x = \rho_0 u_x$, $j_y = \rho_0 u_y$, are locally conserved. The other six moments are not conserved, they are

energy e , energy square ε , energy flux in x and y directions q_x and q_y , normal stress p_{xx} , and shear stress p_{xy} . These non-conserved moments are relaxed as follows

$$\begin{aligned} \tilde{e} &= e - S_e (e - e^{eq}) \\ \tilde{\varepsilon} &= \varepsilon - S_\varepsilon (\varepsilon - \varepsilon^{eq}) \\ \tilde{q}_x &= q_x - S_q (q_x - q_x^{eq}) \\ \tilde{q}_y &= q_y - S_q (q_y - q_y^{eq}) \\ \tilde{p}_{xx} &= p_{xx} - S_v (p_{xx} - p_{xx}^{eq}) \\ \tilde{p}_{xy} &= p_{xy} - S_v (p_{xy} - p_{xy}^{eq}), \end{aligned} \quad (5)$$

where the symbol \sim denotes the post-collision value. The equilibrium values of the non-conserved moments are designed to match the Euler and Navier-Stokes equation through the Chapman-Enskog analysis, and the results are

$$\begin{aligned} e^{eq} &= -2\delta\rho + 3\rho_0 u^2 \\ \varepsilon^{eq} &= \delta\rho - 3\rho_0 u^2 \\ q_x^{eq} &= -\rho_0 u_x \\ q_y^{eq} &= -\rho_0 u_y \\ p_{xx}^{eq} &= \rho_0 (u_x^2 - u_y^2) \\ p_{xy}^{eq} &= \rho_0 u_x u_y, \end{aligned} \quad (6)$$

where $u^2 = u_x^2 + u_y^2$. While the equilibrium ε^{eq} plays no role in the Navier-Stokes equation and thus its form can be flexible, the specific form stated above leads to the standard $\mathbf{f}^{eq} = M^{-1} \mathbf{m}^{eq}$ as

$$f_i^{eq} = w_i \left\{ \delta\rho + \rho_0 \left[\frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right] \right\}. \quad (7)$$

where the weighting coefficient w_i is given as

$$w_0 = \frac{4}{9}, \quad w_i = \frac{1}{9} (i = 1, 2, 3, 4), \quad w_i = \frac{1}{36} (i = 5, 6, 7, 8). \quad (8)$$

It follows that the diagonal relaxation matrix S is

$$S = \text{Diag} \left[0 \quad S_e \quad S_\varepsilon \quad 0 \quad S_q \quad 0 \quad S_q \quad S_v \quad S_v \right]. \quad (9)$$

The shear viscosity and bulk viscosity in the MRT model can be derived from the Chapman-Enskog analysis as

$$\begin{aligned} \nu &= c_s^2 \left(\frac{1}{S_v} - 0.5 \right) \delta t \\ \xi &= c_s^2 \left(\frac{1}{S_e} - 0.5 \right) \delta t. \end{aligned} \quad (10)$$

The evolution equation for \mathbf{f} can be divided into two sub-steps: collision and streaming, as

$$\tilde{\mathbf{f}}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) - M^{-1} S [\mathbf{m} - \mathbf{m}^{eq}], \quad (11)$$

$$\mathbf{f}(\mathbf{x} + \mathbf{e}_i \delta t, t + \delta t) = \tilde{\mathbf{f}}(\mathbf{x}, t). \quad (12)$$

In summary, the key features of the MRT model is that collision is performed in moment space, and streaming occurs in discrete velocity space. The two relaxation times S_v and S_e determine the shear and bulk viscosity, the other two relaxation parameters S_q and S_ε may be viewed as free parameters that can be used to improve the accuracy of boundary conditions or to enhance numerical stability [6].

3. Implementation of local grid refinement in MRT LBM

Without loss of generality, we consider a local fine-grid domain surrounded by a coarse grid, as shown in Fig. 2. In the sketch, the boundary of the coarse domain is defined by $ABCD$ and that of the fine grid domain by $KLMNK$, without any overlap. However, to facilitate implementation and information transfer between the two

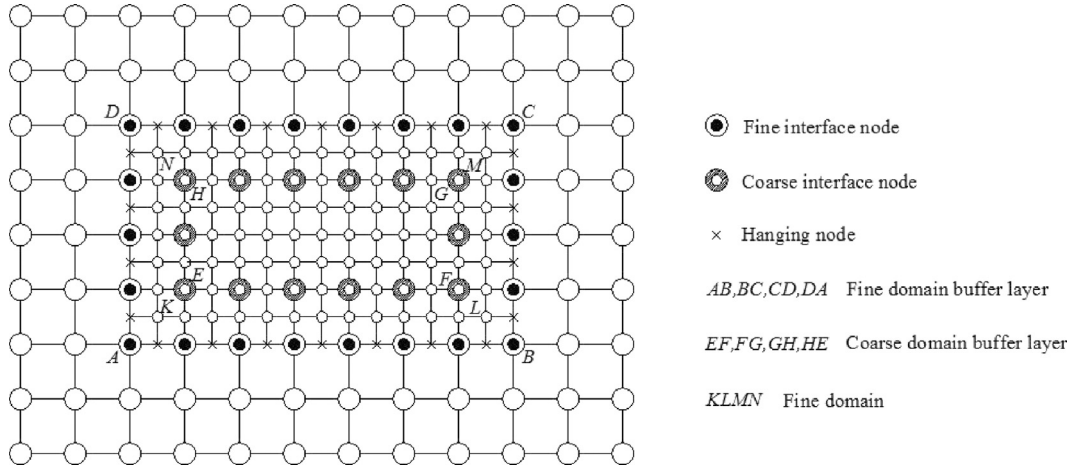


Fig. 2. Grids structure with different space size in coarse and fine block.

grids, we add a buffer layer $EFGHE$ to the coarse domain, which is located inside the fine domain, the corresponding coarse grid nodes are referred to as the coarse interface nodes. Likewise, a buffer layer for the fine domain, which coincides with the boundary of the coarse domain are used to define additional populations for the fine domain, which consists of fine interface nodes (those coinciding with the coarse boundary nodes) and hanging nodes (those placed in-between the coarse boundary nodes). The buffer layers are used to supply necessary information so that, after streaming, all populations on the fine-grid nodes and coarse-grid nodes are available. This arrangement followed the work of [17].

As a multi-block scheme [9], the fine domain has a grid spacing of δx^f and the coarse domain has a grid spacing of δx^c . The size ratio

$$n = \frac{\delta x^c}{\delta x^f} = \frac{\delta t^c}{\delta t^f} \quad (13)$$

is assumed to be an integer, where the superscript c and f denote the related variables located in coarse domain and fine domain. The sketch shown in Fig. 2 has $n = 2$. The time steps for the coarse and fine grids, δt^c and δt^f , are defined such that the streaming in each domain shifts a particle from a lattice grid node to a neighboring node. By this setting, the lattice velocity unit in the two domains are the same, namely, $\delta x^c / \delta t^c = \delta x^f / \delta t^f$. This also implies that the speed of sound and the transformation matrix are all identical for the two domains.

The first physical requirement is that the hydrodynamic variables $\delta \rho$, u_x , u_y must be continuous across the domain interface, we must have $\delta \rho^f = \delta \rho^c$, $j_x^f = j_x^c$, and $j_y^f = j_y^c$. All equilibrium moments defined on the two grids must be the same, namely, $\mathbf{m}^{eq,f} = \mathbf{m}^{eq,c}$. We only need to determine the relationships between the non-equilibrium parts of the non-conservative moments.

The second physical requirement is that the physical shear viscosity and bulk viscosity should be the same in the two domains, leading to

$$c_s^2 \left(\frac{1}{S_v^c} - 0.5 \right) \delta t^c = c_s^2 \left(\frac{1}{S_v^f} - 0.5 \right) \delta t^f, \quad (14)$$

$$c_s^2 \left(\frac{1}{S_e^c} - 0.5 \right) \delta t^c = c_s^2 \left(\frac{1}{S_e^f} - 0.5 \right) \delta t^f. \quad (15)$$

Namely, the relaxation parameters for the fine domain should be related to those in the coarse domain as

$$\frac{1}{S_v^f} = 0.5 + n \left(\frac{1}{S_v^c} - 0.5 \right), \quad (16)$$

$$\frac{1}{S_e^f} = 0.5 + n \left(\frac{1}{S_e^c} - 0.5 \right). \quad (17)$$

The third physical requirement is that the normal and shear stress components should be the same at the domain interface. The Chapman–Enskog analysis states that

$$\begin{aligned} \tau_{xx} &= -\frac{1}{6} (1 - 0.5S_e) e^{neq} - \frac{1}{2} (1 - 0.5S_v) p_{xx}^{neq} \\ \tau_{yy} &= -\frac{1}{6} (1 - 0.5S_e) e^{neq} + \frac{1}{2} (1 - 0.5S_v) p_{xx}^{neq} \\ \tau_{xy} &= -(1 - 0.5S_v) p_{xy}^{neq}. \end{aligned} \quad (18)$$

Here we demand the total normal stress components be continuous, to be more consistent with the hydrodynamics. Therefore,

$$\begin{aligned} -\frac{1}{6} (1 - 0.5S_e^c) e^{neq,c} - \frac{1}{2} (1 - 0.5S_v^c) p_{xx}^{neq,c} \\ = -\frac{1}{6} (1 - 0.5S_e^f) e^{neq,f} - \frac{1}{2} (1 - 0.5S_v^f) p_{xx}^{neq,f}, \end{aligned} \quad (19)$$

$$\begin{aligned} -\frac{1}{6} (1 - 0.5S_e^c) e^{neq,c} + \frac{1}{2} (1 - 0.5S_v^c) p_{xx}^{neq,c} \\ = -\frac{1}{6} (1 - 0.5S_e^f) e^{neq,f} + \frac{1}{2} (1 - 0.5S_v^f) p_{xx}^{neq,f}, \end{aligned} \quad (20)$$

$$-(1 - 0.5S_v^c) p_{xy}^{neq,c} = -(1 - 0.5S_v^f) p_{xy}^{neq,f}. \quad (21)$$

Solving these three equations and in view of the conditions given by Eqs. (16) and (17), we obtain the following relationships between three non-equilibrium moments in the two domains, as

$$e^{neq,f} = \frac{S_e^c}{n S_e^f} e^{neq,c}, \quad p_{xx}^{neq,f} = \frac{S_v^c}{n S_v^f} p_{xx}^{neq,c}, \quad p_{xy}^{neq,f} = \frac{S_v^c}{n S_v^f} p_{xy}^{neq,c}. \quad (22)$$

For the three remaining non-equilibrium moments: energy square and two energy flux components, the Chapman–Enskog analysis shows that

$$\partial_{t1} (\delta \rho - 3\rho_0 u^2) + \partial_{1x} (-\rho_0 u_x) + \partial_{1y} (-\rho_0 u_y) = -\frac{S_e \varepsilon^{neq}}{dt}, \quad (23)$$

$$\partial_{t1} (-\rho_0 u_x) + \partial_{1x} (-1 + 6u_y^2 - 3u^2) + \partial_{1y} (\rho_0 u_x u_y) = -\frac{S_q q_x^{neq}}{dt}, \quad (24)$$

$$\partial_{t1} (-\rho_0 u_y) + \partial_{1x} (\rho_0 u_x u_y) + \partial_{1y} (-1 + 6u_x^2 - 3u^2) = -\frac{S_q q_y^{neq}}{dt}. \quad (25)$$

fine interface nodes, using either Eq. (30) or Eq. (38), depending on whether the conversion was done before or after the collision sub-step. Second, there are no coarse nodes defined at the locations of the hanging nodes, so the fine-grid distribution functions at the hanging nodes are obtained from the fine-grid distribution functions at the fine interface nodes. We employ the cubic spline interpolation at each edge of the coarse-domain boundary, namely,

$$f_i(s_i) = a_i(s - s_i)^3 + b_i(s - s_i)^2 + c_i(s - s_i) + d_i, \quad (41)$$

where f_i is the distribution function being interpolated, s is the local coordinate at the edge, s_i is the location of the hanging nodes (the interpolation points), as shown in Fig. 2, $s - s_i = 1/2$, a_i , b_i , c_i , d_i are cubic spline coefficients which are determined by fitting the known values at finite interface nodes, following Eq. (30) in [13].

Furthermore, for each time step corresponding to the coarse domain, there are n time steps for the fine domain. The distribution functions at these sub-timesteps are interpolated in time between t and $t + \delta t$. The converted fine-grid distribution functions at the fine interface nodes and hanging nodes are then streamed onto the fine boundary nodes. If the conversion between the two grids at the buffer layers are performed before the collision sub-step, the collision operation should be executed on the buffer layers before the streaming substep.

The arrangement of the buffer layers for the fine and coarse regions ensures each domain is fully extended, such that the distribution functions at all nodes in the fine or coarse domain are complete after the streaming sub-step. The two domains do not overlap, and the buffer layers provide the bridges for information transfer.

Fig. 3 provides a flow chart for the code when the local refinement is applied to a moving particle, namely, the grid refinement is done around a moving solid particle, with the fine domain moving with the solid particle. For this chart, it is assumed that the distribution functions between the coarse and fine grids are communicated *after* the collision sub-step (i.e., post-collision rescaling) using Eqs. (38) and (40), as in the work of Peng et al. [19]. Hereafter we shall refer to this treatment as the post-collision rescaling implementation.

The better approach, free of the rescaling singularity, is to perform the rescaling before the collision sub-step (i.e., the pre-collision rescaling implementation). The flow diagram of this pre-collision rescaling implementation with local grid refinement around a moving particle is provided in Fig. 4. In this case, data transfer as implied by Eqs. (30) and (31), becomes

$$\mathbf{f}^f = M^{-1}[\mathbf{m}^{eq,c} + T^c(M\mathbf{f}^c - \mathbf{m}^{eq,c})], \quad (42)$$

$$\mathbf{f}^c = M^{-1}[\mathbf{m}^{eq,f} + T^f(M\mathbf{f}^f - \mathbf{m}^{eq,f})]. \quad (43)$$

In summary, we have developed two versions of local grid-refinement implementation around a moving particle: the post-collision rescaling implementation (Fig. 3) and the pre-collision rescaling implementation (Fig. 4). In the first case, Eqs. (38) and (40) should be used for coarse-fine data conversions; while in the second case, Eqs. (42) and (43) are used instead for coarse-fine data conversions. It is important to point out that the hydrodynamic variables and the normal and shear stress components must always be computed after the streaming sub-step. Another important difference between the two implementations is that in the post-collision rescaling implementation, no collision operation is needed for the interface nodes; while in the pre-collision rescaling implementation, the distribution functions on the interface grid nodes must participate in the collision sub-step. When the above implementation details and the conversion relationships are rigorously followed, we found that the results from the two implementations are identical.

In the moving-solid-particle case, the fine domain surrounds the solid particle. Initially, the fine domain is chosen with its center coinciding with the center of the solid particle. When the solid particle moves one coarse grid length in any direction, the particle center moves from O to O' , as shown in Fig. 5, the fine domain simultaneously shifts its center by the same distance following the center of the solid particle, namely, the initial fine domain ABCD shifts to a newly defined fine domain $A'B'C'D'$. The fine domain shifting philosophy is similar to that reported in Arora et al. [26]. The distribution functions of the nodes sitting on the left buffer layer $A'D'$ of the new fine domain and the right buffer layer BC of the old fine domain have already been computed. However, the unknown distribution functions for the fine interface nodes EF and the new hanging nodes on the right buffer layer $B'C'$ of the new fine domain must be constructed. For the new fine interface nodes on $B'C'$, the conversion relation, Eqs. (38), can be used for the post-collision rescaling implementation, or Eq. (42) is used for the pre-collision rescaling implementation. The hanging nodes on $B'C'$ can then be constructed by interpolation, and the fine-domain distribution functions on EF are interpolated from the distribution functions of the two layers BC and $B'C'$. At the end of the above process, the array indices for the fine domain are shifted accordingly to keep the data array size of the fine-domain distributions unchanged during domain shift.

At the same time, the old coarse-domain buffer layers KL , LM , MN , NK must be shifted accordingly to $K'L'$, $L'M'$, $M'N'$, $N'K'$. Along the new layer $N'K'$ which lies within the old fine domain, the coarse-domain distributions can be computed from the fine-domain distributions, using Eq. (40) and Eq. (43) for the post-collision rescaling case and the pre-collision rescaling case, respectively. The coarse-domain distributions on the remaining three sides, $K'L'$, $L'M'$, and $M'N'$, of the new coarse domain are already available. At this point, the distributions on all buffer layers of new fine domain and new coarse domain are available, and the treatment for the fine-domain shift associated with the moving solid particle is complete.

Here, we should point out that the differences between our model and the model of Farhat and Lee [13] include:

- (1) The FL's model was based on the BGK collision and ours assumed the use of MRT collision model.
- (2) We explicitly demonstrate that there are two ways of transferring data between the fine and coarse grids, one is doing this before the collision sub-step, and the second is after the collision sub-step. To our knowledge, the comparison of two implementations in one study has not been performed previously.
- (3) In FL's paper, the flow field was divided into three separated blocks, named as the upstream coarse block, the fine block and the downstream block. The fine domain size is too large and obviously requires more storage space than our model.
- (4) In FL's paper, after the fine domain was migrated, extrapolation was used to compute the variables at all the newly created fine and coarse nodes. As we know, the extrapolation may cause some numerical diffusion which affects the accuracy of LBM. In our paper, no extrapolation is used.
- (5) In our paper, we also investigate the minimum or optimal fine domain size in Section 5.2.2, which is relevant to the overall numerical efficiency.

The differences between our model and the model of Arora et al. [26] include points (2) and (5) above, as well as the following: (1) We have developed a method for relating all the non-equilibrium moments across the subdomain boundaries, based on the Chapman–Enskog analysis. We obtain the relationship between non-equilibrium energy, energy square and energy flux moments in two different domains. Our derivations are more systematic and

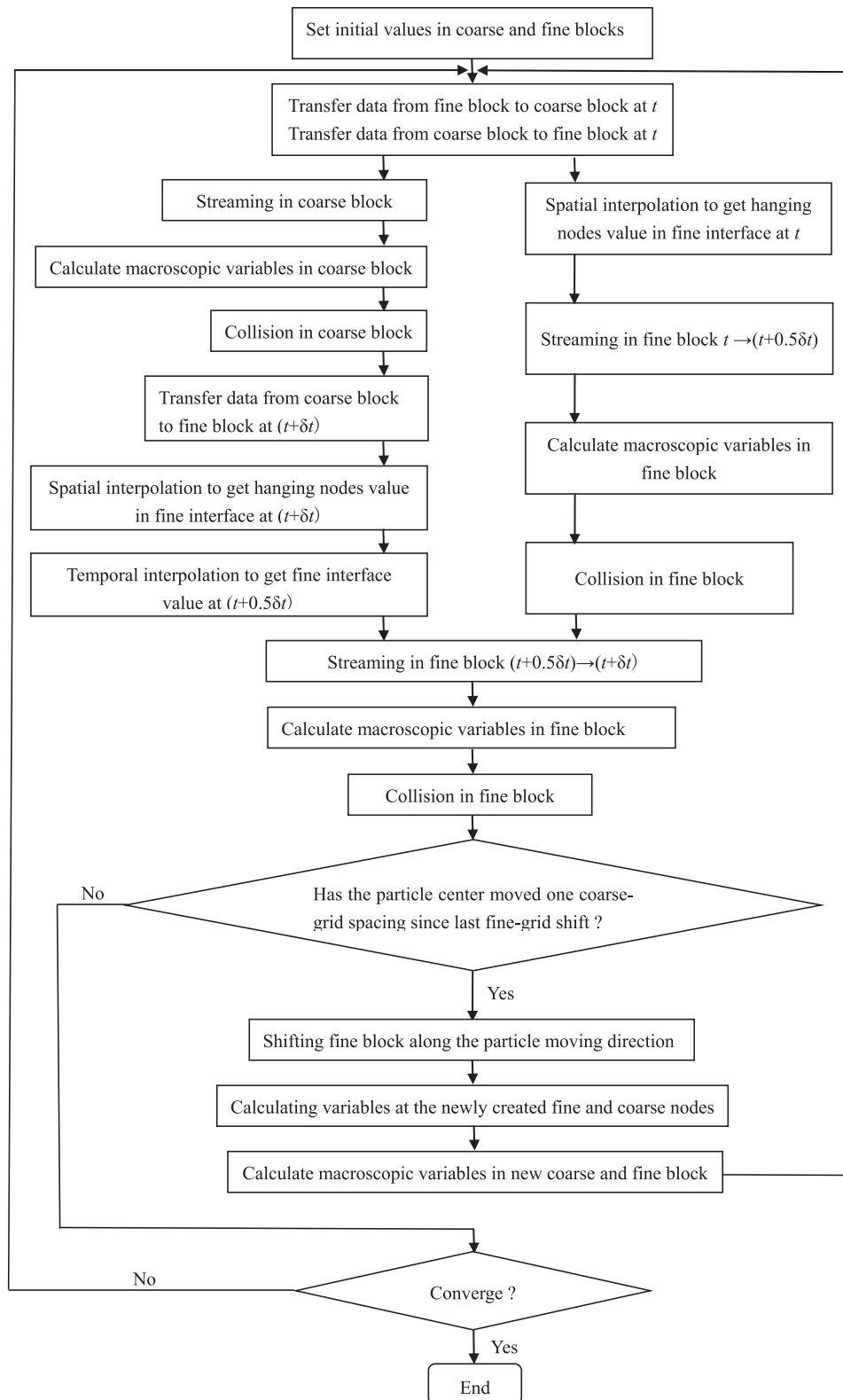


Fig. 3. The flow chart for the post-collision rescaling implementation with fine block shifting, assuming $n = 2$.

complete, and LBM users may find these details beneficial. On the other hand, only two non-equilibrium stress moments and three conserved moments were explicitly related according to the transfer matrix of T , stated in Eq. (23) in Arora et al.'s model. (2) Also

related to the present model, we will demonstrate that the ghost moments could not be completely ignored during the data transferring between the coarse and fine subdomains in Section 5.2.4. In Arora et al.'s model, the ghost moments were not considered.

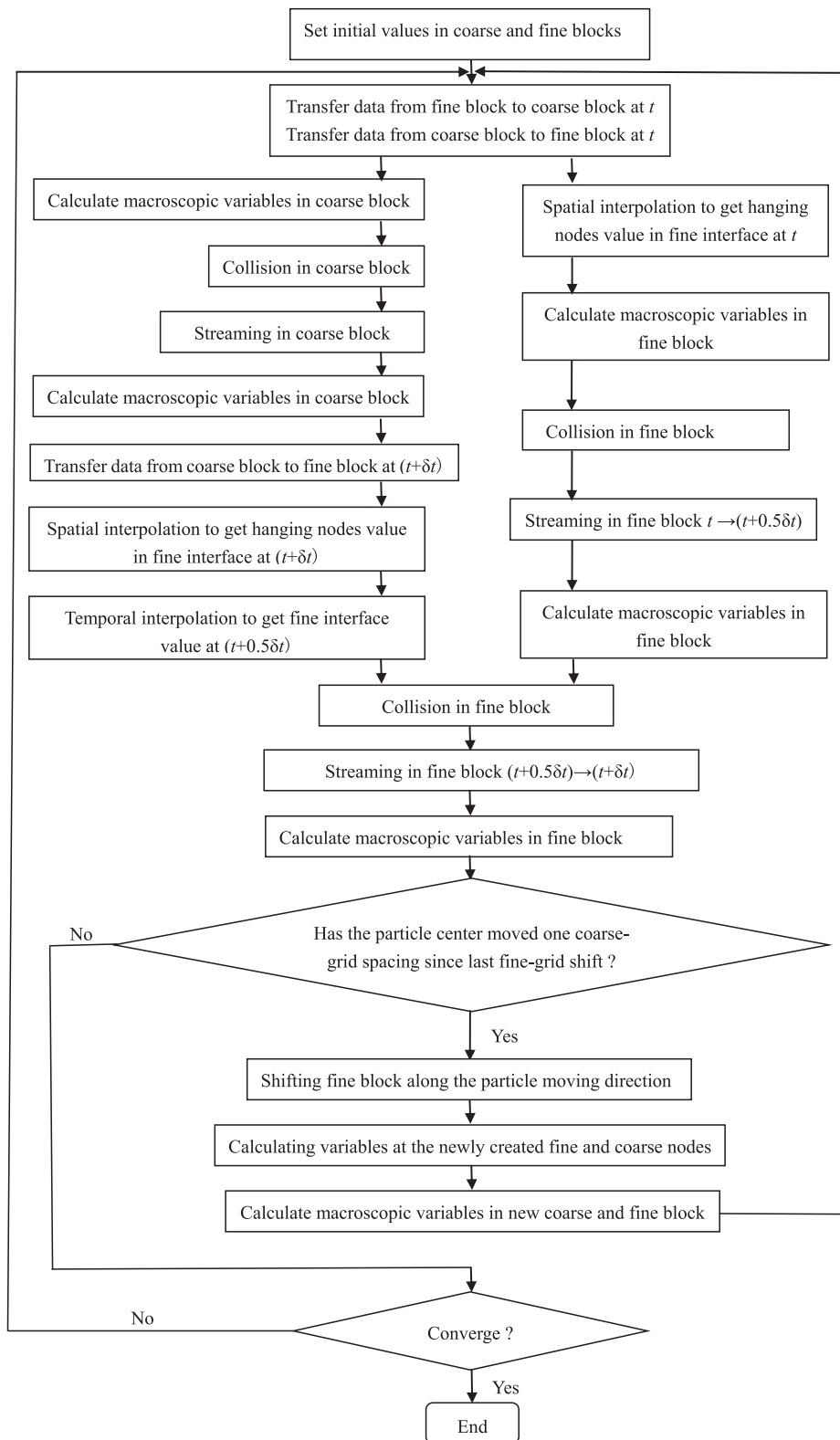


Fig. 4. The flow chart for the pre-collision rescaling implementation with fine block shifting, assuming $n = 2$.

5. Results from numerical simulations and discussions

In order to validate the approach and to highlight the benefits of local grid refinement in improving computational accuracy using the MRT LBM model, we apply the approach to solve two flow

problems. The first is the lid-driven cavity flow and we implement local grid refinement on the upper-left corner, to show that the local grid refinement can provide more accurate results for this region where large velocity gradients exist. The second is a 2D Couette flow over a fixed or moving cylinder. Here local grid refine-

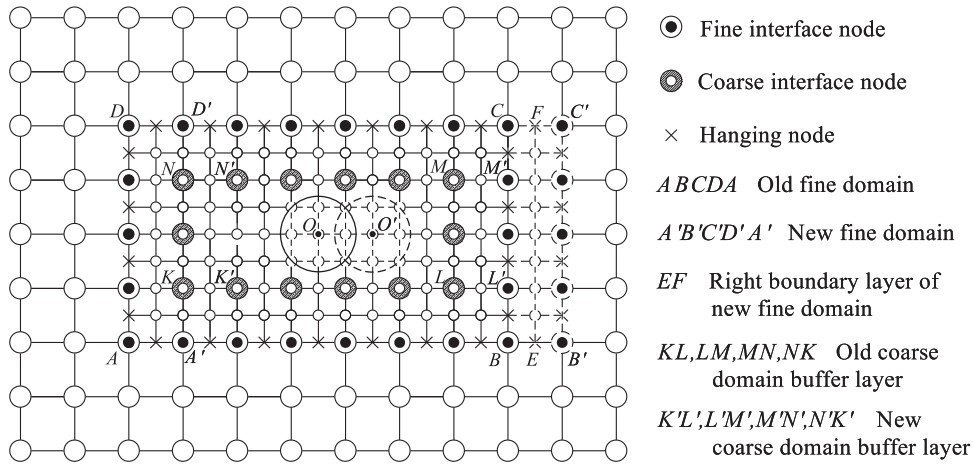


Fig. 5. A sketch to illustrate the necessary rearrangements of the buffer layer nodes for the new coarse and fine domains after the fine domain shifts its location relative to the coarse domain.

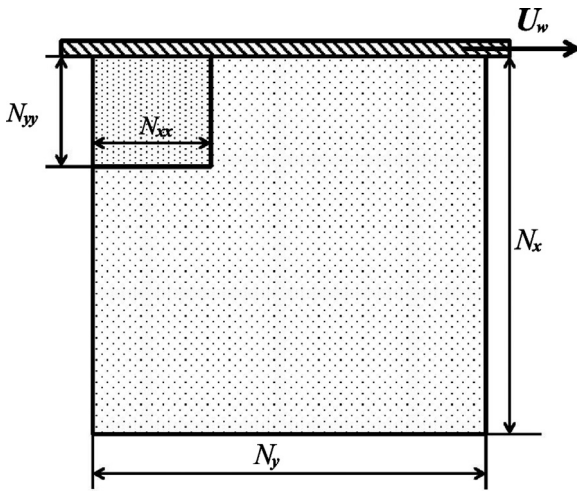


Fig. 6. Local grid refinement block layout for a 2D cavity flow.

ment is applied to a region near the cylinder to demonstrate that grid refinement can suppresses fluctuations in the hydrodynamic force acting on the moving cylinder.

5.1. Lid driven cavity flow

The lid-driven cavity flow has been extensively used as a benchmark case to test a numerical method [29]. In this flow, the two corners under the moving lid are singular points, and higher grid resolution is desired in order to obtain more accurate stress distribution near the corner points. We apply local grid refinement to the top-left corner (Fig. 6). The cavity flow Reynolds number $Re = LU_w/\nu$ is fixed to 1000, where L is the width of the square cavity, U_w is the lid velocity, $\nu = 0.01$ is the kinematic viscosity.

The bulk viscosity ξ was set to be equal to ν , and these lead to $S_c^c = 1.8868$, $S_v^c = 1.8868$ in the coarse region where $\delta t^c = 1$ and $\delta x^c = 1$. The two remaining relaxation parameters in the coarse region are $S_c^e = 1.54$ and $S_q^c = 1.9$.

For the local grid refinement case, a refined grid with $dx^f = 0.5$ is applied to a region near the top-left corner of the size $N_{xx} \times N_{yy}$. These lead to the setting that $S_c^f = 0.641$, $S_v^f = 0.641$, $S_c^e = 1.54$, and $S_q^f = 1.9$.

In order to demonstrate quantitatively the convergence of our uniform coarse grid with local grid refinement method (UCG-L) with the grid resolution, we set the fine-domain grid-length to

half of the coarse-domain grid-length. The flow is initially at rest. After a sufficiently long time (typically over 200,000 coarse-grid time steps), the flow reaches to a steady state. All velocity profiles shown below are results at steady state. We computed the absolute error relative to the benchmark data of Ghia et al. [29]. As shown in Fig. 7, the normalized velocity profiles along the vertical mid-plane and horizontal mid-plane of the cavity agree well with the benchmark data, for all the three resolutions (129×129 , 149×149 and 199×199) considered. Using the same 17 discrete locations of the benchmark data, the normalized velocities u_x/U_w are interpolated to determine the average normalized velocity, $\frac{1}{17} \sum_1^{17} u_x/U_w$, and the average absolute error, $\frac{1}{17} \sum_1^{17} \left| \frac{u_x/U_w}{u_{Ghia}} - 1 \right|$. Both are shown in Table 1. The resolutions in the parentheses on the first row indicate the fine domain size in the coarse-domain unit. Table 1 shows that the computed absolute error decreases from 1.27% – 0.86% as the grid resolution is increased from 129×129 to 199×199 . Fig. 7 shows how the velocity profiles converge to the benchmark as the grid resolution is increased, consistent with the error data in Table 1.

The results in Fig. 7 show that a coarse mesh resolution of 129×129 is adequate in resolving the flow in the central region of the cavity at $Re = 1000$. In Fig. 8, we demonstrate that the profiles in the center region are independent of the grid refinement applied in the corner region and the size of the refined domain. Here three fine-domain sizes 26×26 , 39×39 , 80×80 for the upper left corner all used a grid length of $0.5L/129$, so they covered a spatial region of the size $0.1L \times 0.1L$, $0.15L \times 0.15L$, and $0.31L \times 0.31L$, respectively.

An important question is whether the local grid refinement affects the numerical stability. We examined this aspect by finding the minimum shear viscosity that one can use without encountering numerical instability for $Re = 1000$. Table 2 provides this minimum shear viscosity value. The uniform coarse grid (UCG) simulations are numerically stable for a shear viscosity above 0.00829 while the UCG-L simulations are stable with a shear viscosity at and above 0.00835. The difference in the minimum viscosity is therefore negligibly small. Therefore, we conclude that the local grid refinement has a very little impact on the numerical stability of the LBM scheme.

Next, we demonstrate the effects of local grid refinement on the quality of the simulation results, by comparing to the results obtained with the uniform coarse grid. First, we compare the pressure contours in Fig. 9 for the case of $Re = 400$, where Fig. 9(a) shows the contours with the uniform coarse grid, Fig. 9(b) shows the contours using the same coarse grid but with local refine-

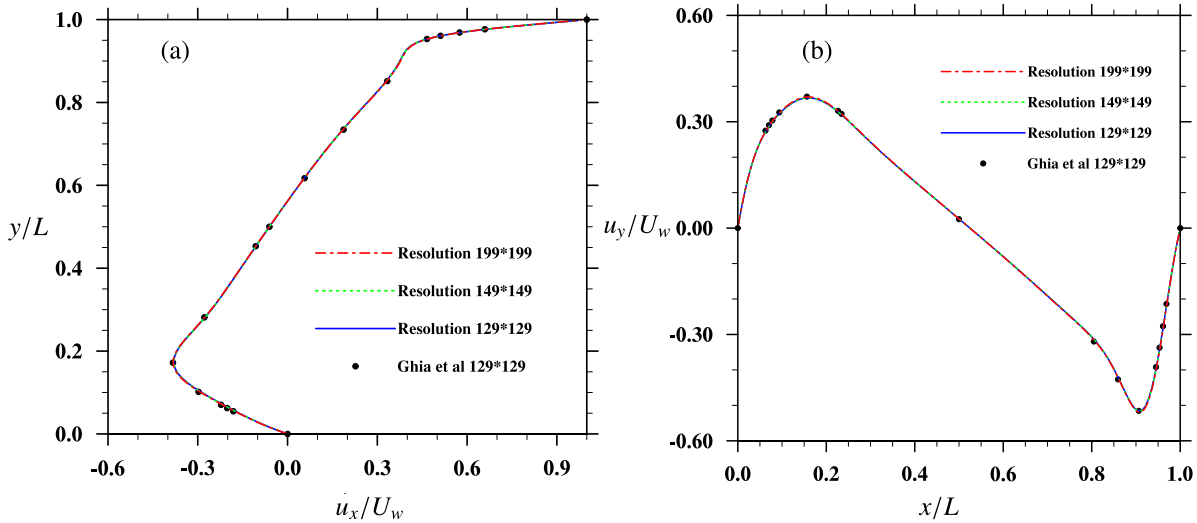


Fig. 7. Comparison of normalized velocity profile computed with UCG-L and Reference 29 (Ghia et al), (a): u_x/U_w along the vertical line through geometric center; (b): u_y/U_w along the horizontal line through geometric center. $Re = 1000$.

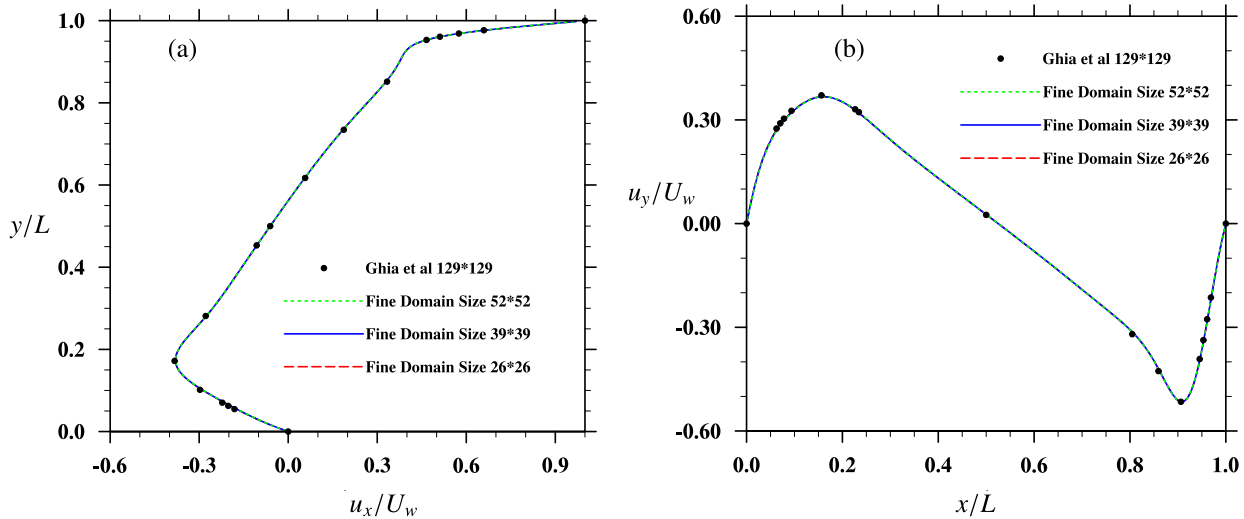


Fig. 8. Comparison of normalized velocity profile computed with same coarse mesh 129×129 , different fine resolution using UCG-L and Reference 29 (Ghia et al), (a): u_x/U_w along the vertical line through geometric center; (b): u_y/U_w along the horizontal line through geometric center. $Re = 1000$.

Table 1
Simulation error of u_x/U_w along vertical mid-plane with different resolution in UCG-L via Reference 29 (Ghia et al).

	Ghia 129*129	129*129 (26*26)	149*149 (39*39)	199*199 (80*80)
Average u_x/U_w	0.12105	0.12318	0.12290	0.12249
Absolute Error		1.27%	1.19%	0.86%

Table 2
Parameters for numerical stability test of cavity flow ($Re=1000$).

	Shear viscosity ν	Grids number	Driven velocity U_w	Fine domain size
UCG	0.00829	16641	0.064264	
UCG-L	0.00835	18721	0.064981	26×26

ment at the left top corner, and Fig. 9(c) shows the contours with the uniform fine grid. The refinement region is indicated by the black square in Fig. 9(b). Here the shear viscosity is set to 0.01. The coarse resolution is 129×129 , and the refinement domain

uses a grid of 39×39 . Several observations can be made here. First, the noise in the pressure field within the refined region is somewhat reduced by the local refinement, as seen by the size of the noisy region there. The local refinement yields essentially the same results as the uniform fine grid case in Fig. 9(c). The origin of the pressure noise is associated with the corner singularity occurring at both top left and top right corners, where the local velocity is discontinuous. Second, the contour lines are smooth across the boundaries of the refinement region in Fig. 9(b), validating the interface treatment. Third, the local refinement also seems to improve the quality of the contours outside the refinement region. Finally, the contour lines are very similar in regions away from the corner singularity.

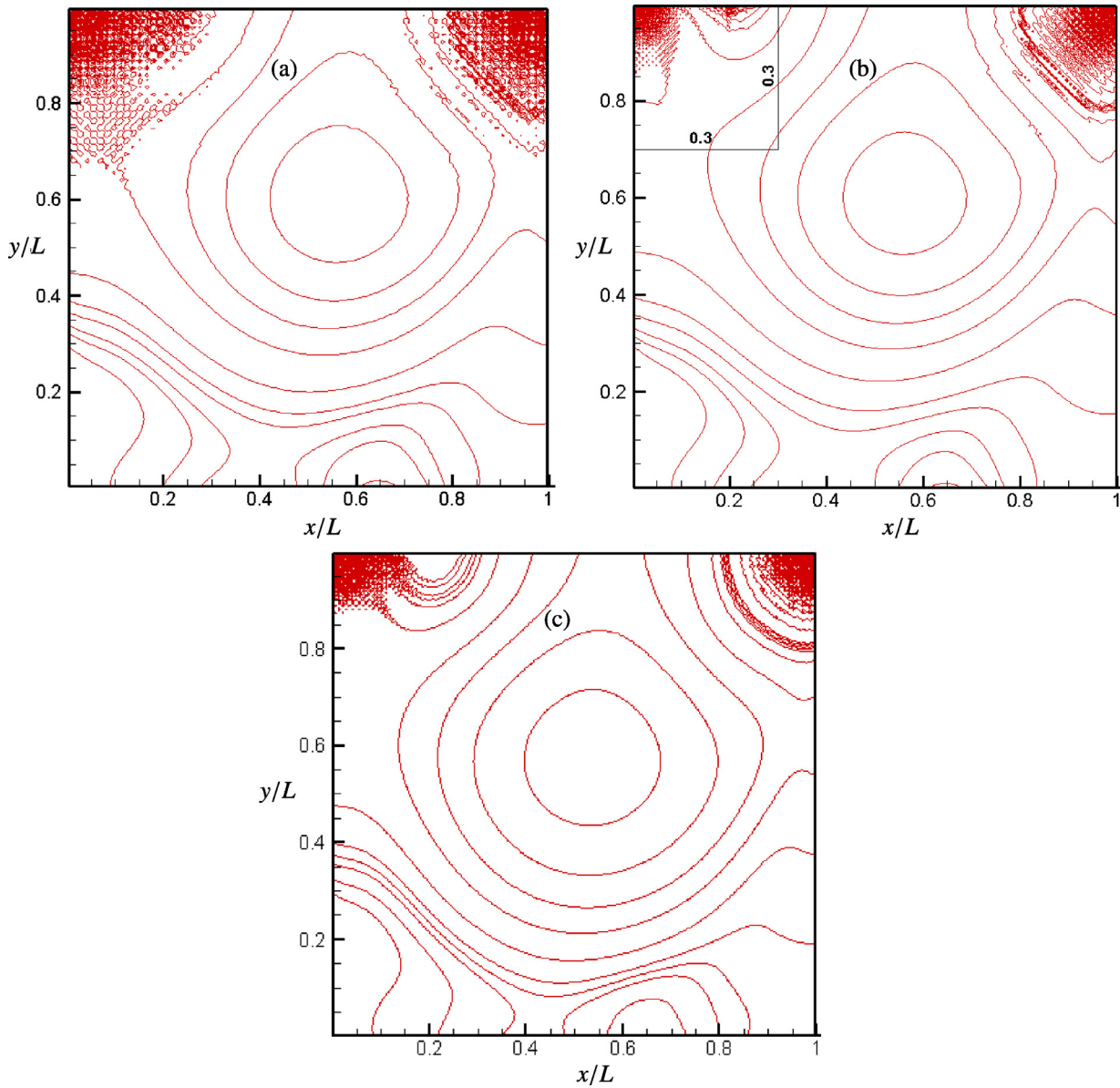


Fig. 9. Comparison of pressure contours of the simulated cavity flow with $Re = 400$. (a): computed by uniform coarse grid; (b): computed by local grid refinement; (c): computed by uniform fine grid. The two thin lines in (b) denote the interface of refinement region.

A more quantitative comparison is now discussed, with simulations performed using three different grid configurations: a uniform coarse grid (UCG), a uniform coarse grid with local grid refinement (UCG-L) at the top-left corner, and a uniform fine grid (UFG). The UCG run has a grid resolution of 129×129 . In the UCG-L run, the refined region has a grid resolution of 26×26 . The physical and simulation parameters are listed in Table 3.

The two velocity components along a vertical line at $x = 13$ and a horizontal line at $y = 116$ are shown in Fig. 10 and 11, respectively. Both lines cut through the fine domain. Clearly, the profiles are continuous at the fine-coarse boundary (marked by the vertical line). Second, the results from the three grid configurations essentially overlap, but the UCG-L profiles match better the UFG results, when compared to the UCG results. This shows that the local grid refinement improves the physical results.

Near the upper lid and left wall, there exists an area of large velocity gradients. The shear stress (τ_{xy}) and normal stress components (τ_{xx} and τ_{yy}) profiles obtained from the three grid configurations are shown in Fig. 12 and 13, respectively. The viscous

Table 3

Parameters used in the simulations of two dimensional square cavity flow under three different grid configurations.

	UCG	UCG-L	UFG
U_w	0.07752	0.07782	0.03876
L	129	128.5	258
ν	0.01	0.01	0.01
Re	1000	1000	1000
$N_x \times N_y$	129×129	129×129	258×258
$N_{xx} \times N_{yy}$		26×26	

stress components are computed using Eq. (18). The continuity of the stress profiles at the fine-coarse boundary (marked by the black vertical line) validates our implementation. Such stress profiles are rarely shown in published literature. Interestingly, there are some oscillations in the profiles of normal viscous stresses that could be caused by undamped acoustic waves or simply due to in-

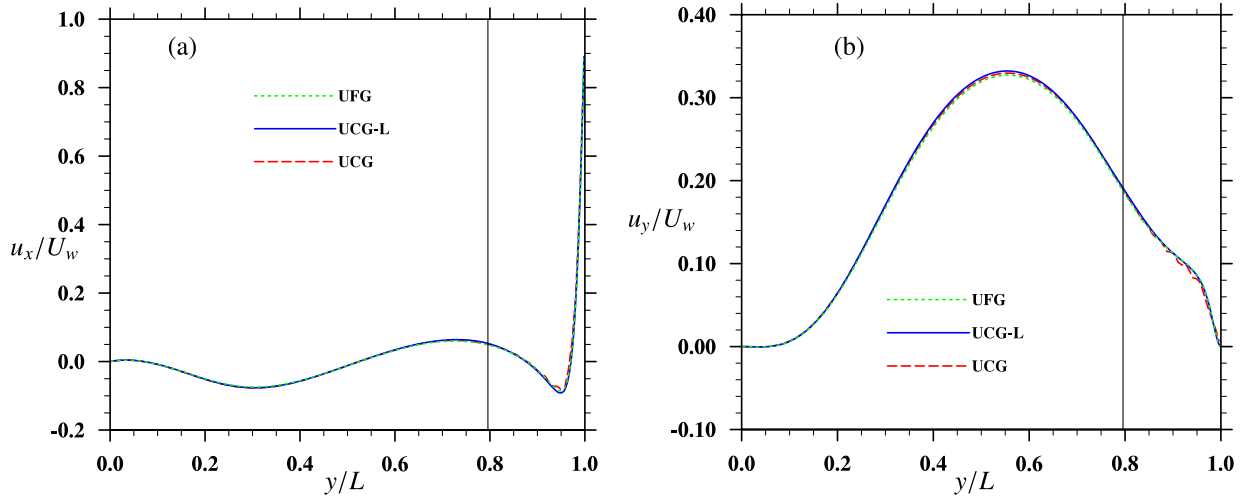


Fig. 10. Comparison of normalized velocity profiles along vertical line $x = 13$ computed with three grid configurations, $Re = 1000$. (a): u_x/U_w ; (b): u_y/U_w . The vertical line marks the boundary between the coarse domain and the fine domain.

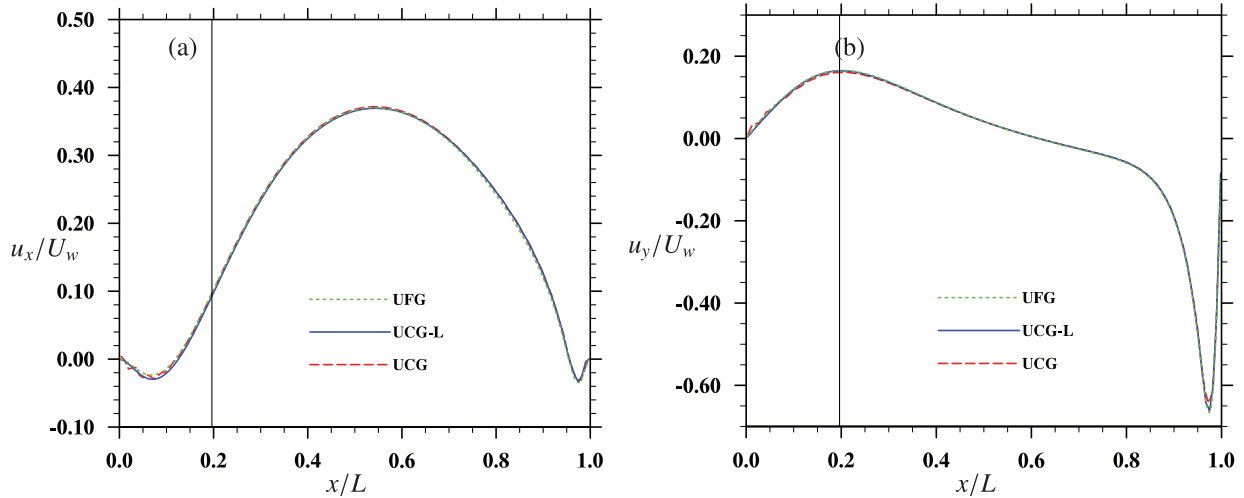


Fig. 11. Comparison of normalized velocity profiles along horizontal line $y = 116$ computed via three grid configurations, $Re = 1000$. (a): u_x/U_w ; (b): u_y/U_w . The vertical line marks the boundary between the coarse domain and the fine domain.

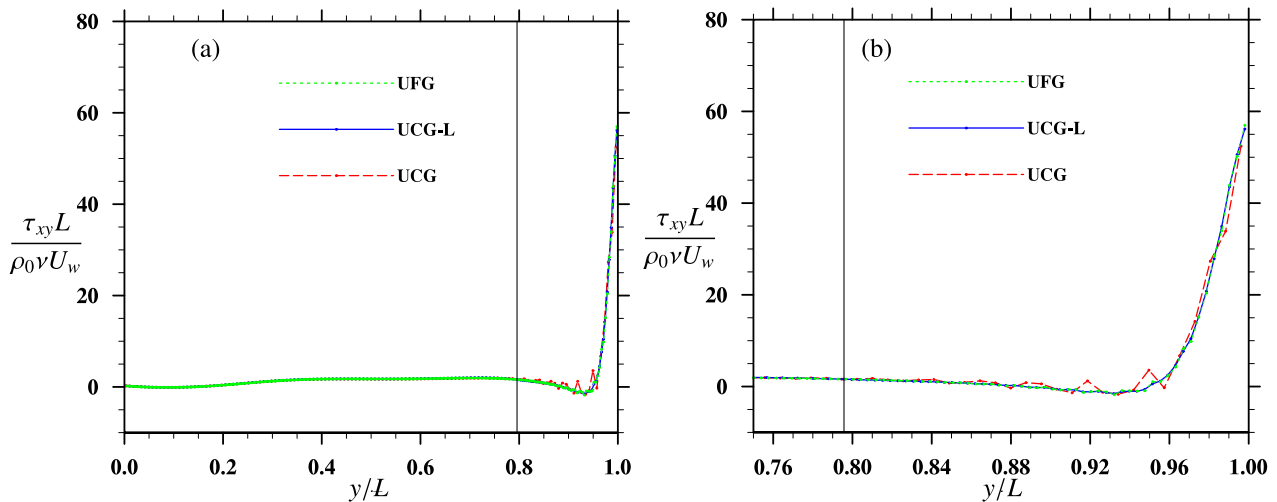


Fig. 12. Comparison of normalized shear stress τ_{xy} along vertical line $x = 13$ computed via three grid configurations, $Re = 1000$. (a): whole domain view; (b): the zoom-in plot. The vertical line marks the boundary between the coarse domain and the fine domain.

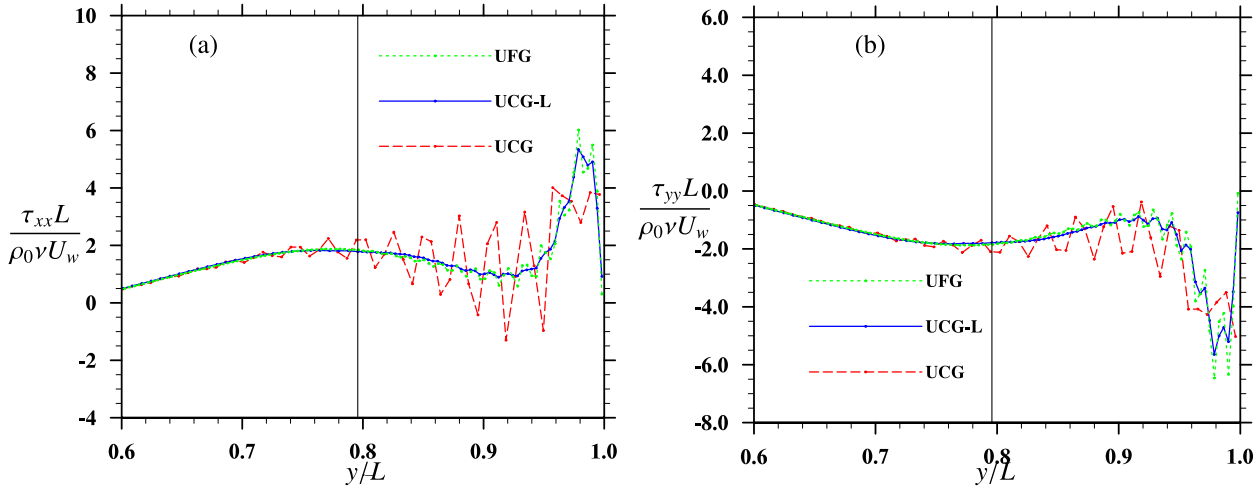


Fig. 13. Comparison of normalized normal stress along vertical line $x = 13$ computed via three grid configurations, $Re = 1000$. (a): τ_{xx} ; (b): τ_{yy} . The vertical line marks the boundary between the coarse domain and the fine domain.

Table 4

Computer CPU and memory occupied by three grid structures with cavity flow.

	UCG	UCG-L	UFG
CPU time (ms)	5.866	12.836	22.388
Memory (kB)	2260	2890	7212

Table 5

Parameters setting of three grid structures in lattice Boltzmann space.

Parameter	UCG	UCG-L	UFG
$N_x \times N_y$	201×101	201×101	402×202
D	25.25	25.25	50.5
(X_{c0}, Y_{c0})	(30, 54)	(30, 54)	(60, 108)
u_0	0.005	0.005	0.0025
U_b	0.1	0.1	0.05

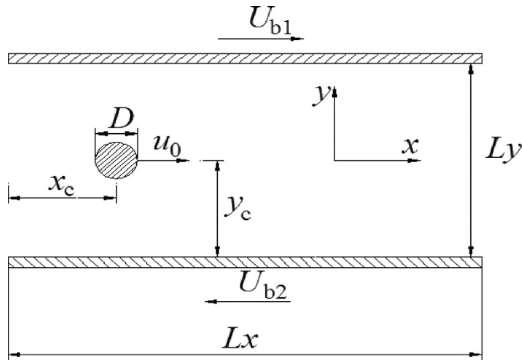


Fig. 14. Sketch of 2D Couette flow containing a cylinder.

adequate grid resolution. These two figures demonstrate a great benefit of local grid refinement: without the refinement, the results from UCG show unphysical oscillations. On the other hand, the UCG-L and UFG results are almost identical.

In Table 4, we compare the CPU time per running step used and memory required for the three cases. While the UFG run required a longer CPU time (by a factor of 4) and larger memory (by a factor 3.5) when compared to the UCG run, these factors for the UCG-L reduced to 2.19 in CPU and 1.29 in memory.

5.2. An asymmetrically placed cylinder in a 2D Couette flow

Next, we consider the same flow studied in [30], namely, an asymmetrically placed cylinder in a 2D Couette flow (Fig. 14). The flow can be simulated in two frames of reference to study the accuracy of a moving particle simulation. In the first (or the fixed cylinder case) case, the cylindrical particle is fixed relative to the lattice grid and the upper and lower channel boundaries move in opposite direction with the same constant velocity (U_b). In the second case (the moving cylinder case), the cylinder moves at a ve-

locity u_0 , with the top wall and bottom wall moving at velocity $U_{b1} = U_b + u_0$ and $U_{b2} = -U_b + u_0$, respectively. Initially, the flow starts from the rest in the two cases. Physically, the two cases are identical. Numerically, the second case is much more difficult due to the need to treat the curved moving fluid-cylinder surface. We implemented local grid refinement in both cases. For the moving cylinder case, the fine domain shifts by one lattice grid every time the center of the cylinder is moved by one lattice grid.

The geometric parameters for this problem include the channel width L_y and length L_x , the diameter of the cylinder D , and the cylinder center at the initial time (X_{c0}, Y_{c0}) . Periodic boundary condition is used in the x direction, and the no-slip condition is assumed at the top and bottom channel walls as well as on the cylinder surface. Again, we consider three grid configurations: uniform coarse grid (UCG), uniform coarse grid with local grid refinement around the cylinder (UCG-L), and uniform fine grid (UFG). The kinematic viscosity is $\nu = 1/9$, which yields $S_\nu = 1.2$ in the coarse domain. Other relaxation parameters in the coarse domain are set to $S_\varepsilon = 1.4$ and $S_\varepsilon = S_q = 1.5$ (i.e., the bulk viscosity is $1/18$). The other parameters in the moving cylinder simulations are set in Table 5. Note that the parameters for the fixed cylinder case are the same, except $u_0 = 0$. In the fine domain with $n = 2$, the resulting relaxation parameters are $S_\nu^f = 6/7$, $S_\varepsilon^f = 1.2$, $S_q^f = 1.5$, $S_\varepsilon^f = 1.4$. The number of iterations for all the following case is $T = 5000$.

5.2.1. The fixed cylinder case

In this case, $u_0 = 0$. The size of the fine domain is a square of size equal to 36. The region covers $12 < x < 48$ and $36 < y < 72$. The no-slip condition on the cylinder surface was handled by a quadratic interpolation scheme [19,30,31].

Figs. 15, 16 and 17 show the drag force F_x , lift force F_y , and torque M as functions of time acting on the particle, respectively.

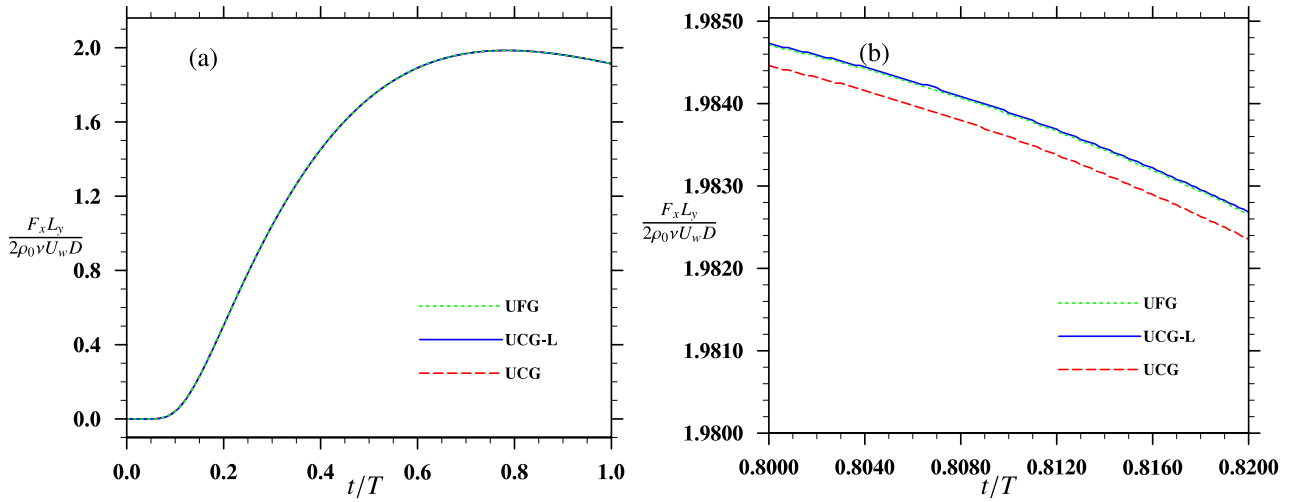


Fig. 15. The normalized hydrodynamic force F_x acting on the fixed cylinder, (a): the whole time interval; (b): zoom-in plot.

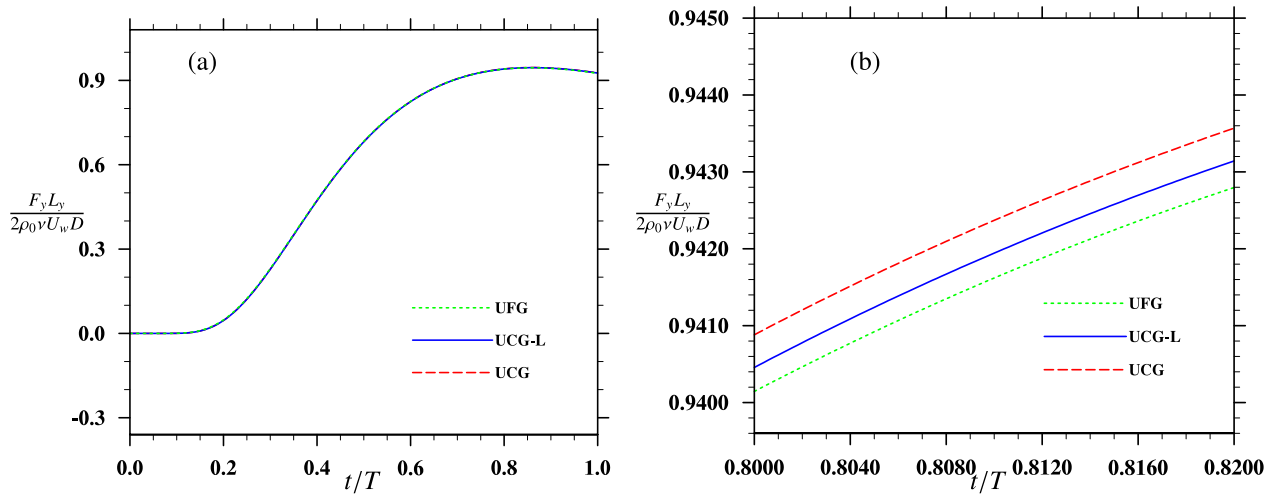


Fig. 16. The normalized hydrodynamic force F_y acting on the fixed cylinder, (a): the whole time interval; (b): zoom-in plot.

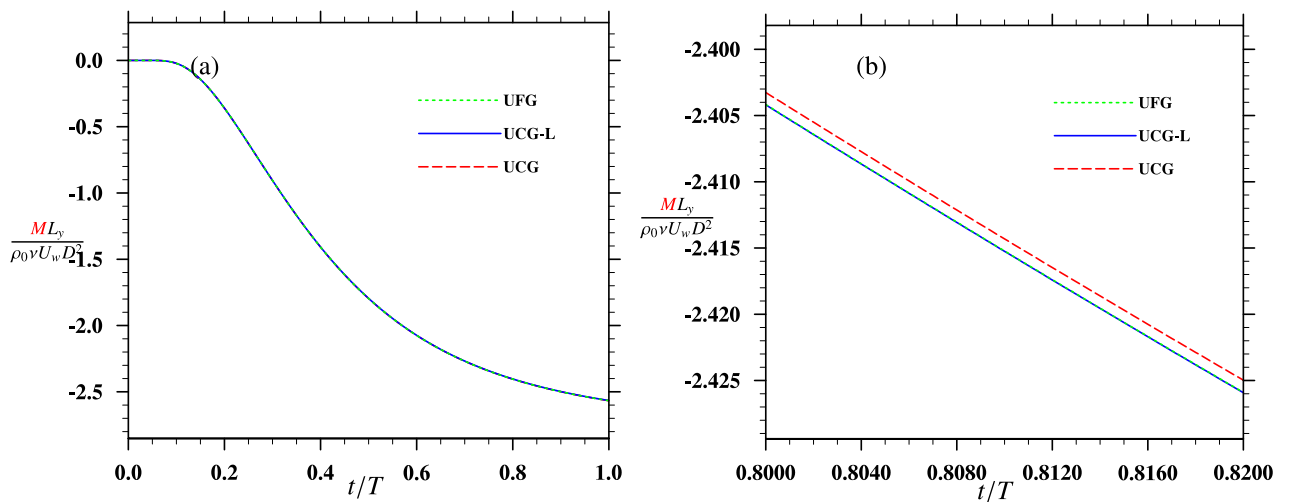


Fig. 17. The normalized hydrodynamic torque acting on the fixed cylinder, (a): the whole time interval; (b): zoom-in plot.

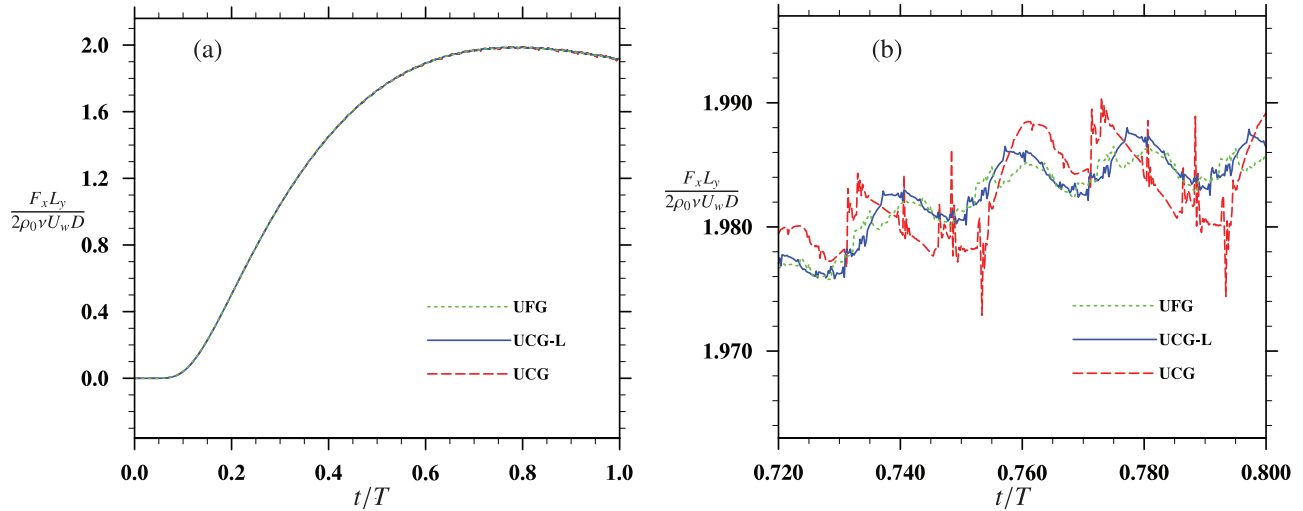


Fig. 18. The normalized hydrodynamic force F_x acting on the moving cylinder, (a): the whole time interval; (b): zoom-in plot.

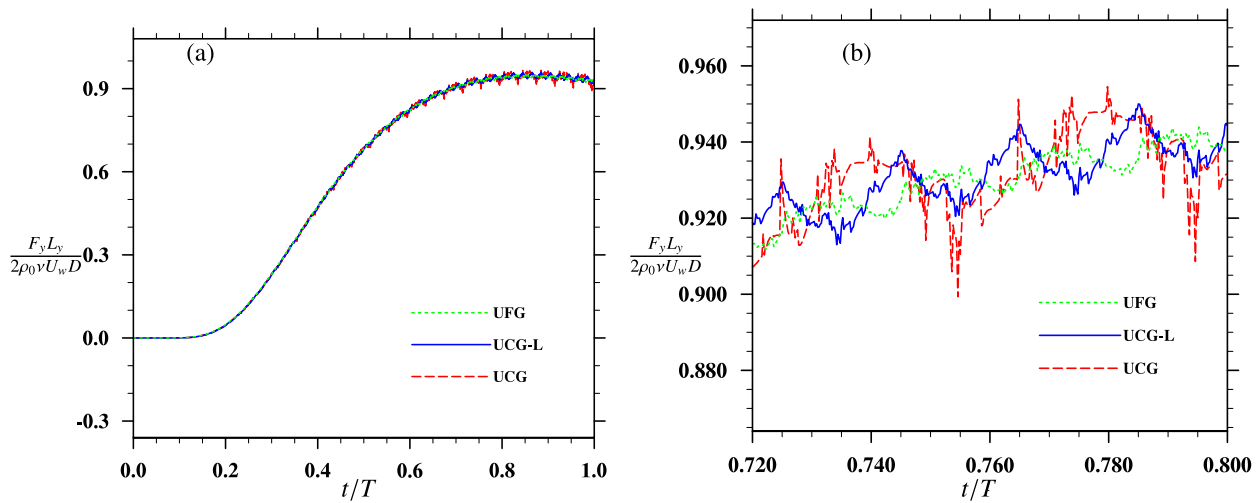


Fig. 19. The normalized hydrodynamic force F_y acting on the moving cylinder, (a): the whole time interval; (b): zoom-in plot.

The force and torque are computed by the Galilean invariant momentum exchange method [19,32].

Overall, the results from the three grid configurations are in excellent agreement. The zoom-in plots for $4000 < t < 4100$ show a very minor difference, typical 0.05% relative difference or less. This is clearly negligible. Therefore, each of these fixed cylinder results can be used as a benchmark to examine results for the moving cylinder case.

5.2.2. Moving particle flow simulation

When the cylinder is moving at u_0 , the fine domain is also a square but with width equal to 32, initially covering $14 < x < 46$ and $38 < y < 70$. It is more or less placed with the cylinder near the center. Every time the cylinder moves by one lattice unit, the fine domain is shifted in the same direction. When the cylinder moves relative to the grid, a solid node may become a fluid node and the distribution functions at such new fluid nodes need to be filled. The refilling scheme is based on a newly developed velocity-constrained extrapolation scheme [31]. In Figs. 18–20, we show the drag force F_x , lift force F_y , and torque as functions of time acting on the particle, respectively. Note that due to the improved scheme, the level of force fluctuations in Figs. 18 and 19 is significantly less

than the level of force fluctuations shown in Fig. 5 of [30], that was computed with UCG case.

The zoom-in view shows that the UCG run has larger magnitude of force fluctuations when compared to that of the UFG run. The results from the UCG-L run are more similar to the UFG run than to the UCG run, showing the benefit of local grid refinement. In order to compare the level of force fluctuations quantitatively, we use the data from the fixed cylinder case as the benchmark and compute the L^2 norm of the difference as [30],

$$\Delta = \sqrt{\frac{\sum |F_1 - F_0|^2}{\sum |F_0|^2}} \quad (44)$$

Where F_1 and F_0 are the force values of the later part simulated with moving particle and fixed particle, respectively. The results are listed in Table 6. The local grid refinement reduces the level of unphysical force fluctuations by roughly a factor of 2.

In Fig. 21, we show the normal stress components τ_{xx} , τ_{yy} and shear stress τ_{xy} at the end of the simulation $T = 5000$. The two figures (a) and (b) represent the stress distributions before the left solid wall and after the right solid boundary of the particle, respectively. The vertical line marks the fine-coarse boundary. Of importance is that all stress profiles show a consistency at fine-coarse

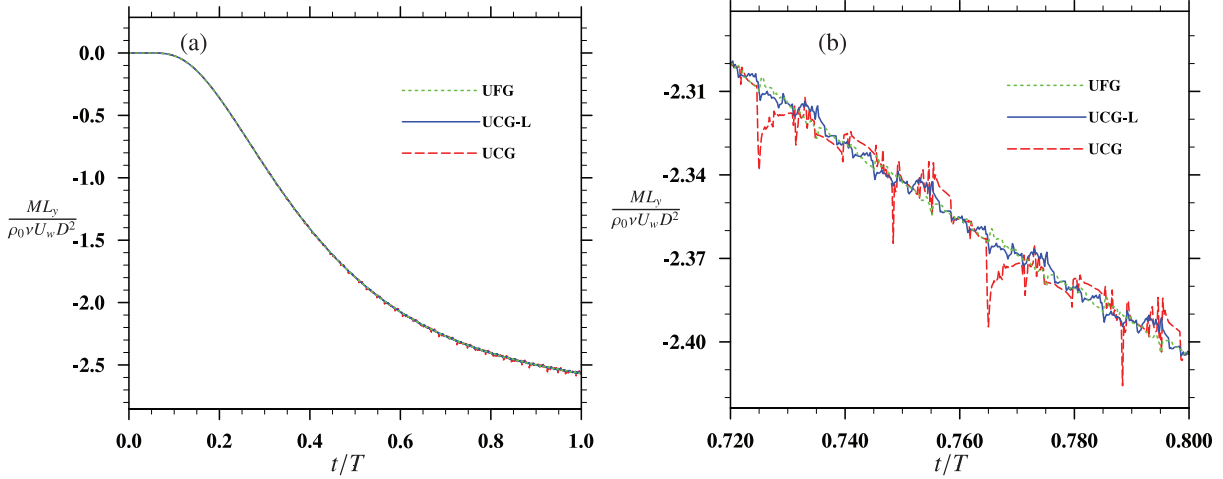


Fig. 20. The normalized hydrodynamic torque acting on the moving cylinder, left: the whole time interval; right: zoom-in plot.

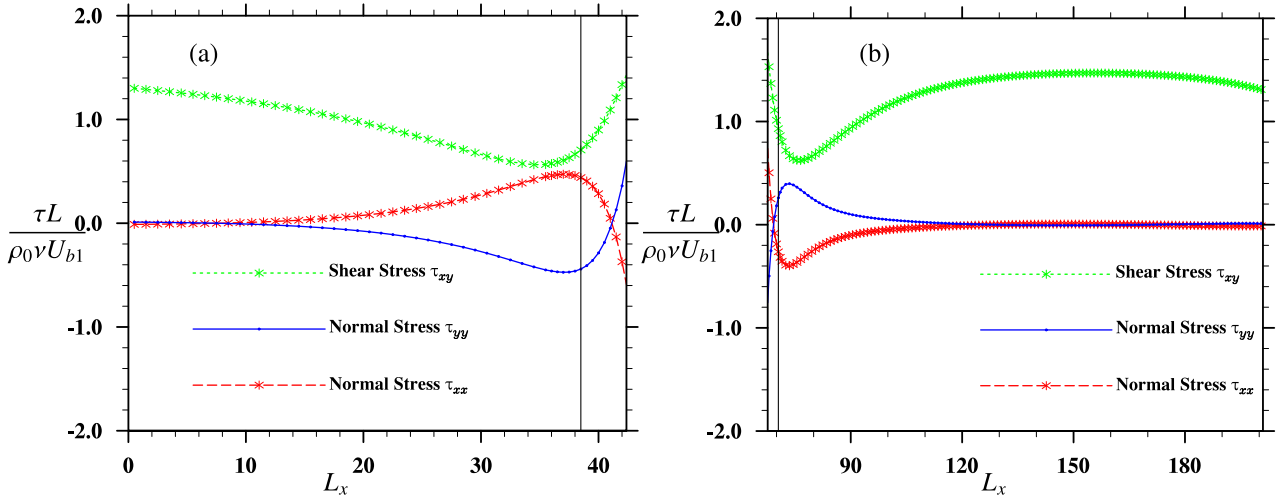


Fig. 21. Normalized stress distribution along $y = 51$. The thin vertical lines mark the fine-coarse domain boundary. (a) stresses distribution before the left solid wall. (b) stresses distribution after the right solid wall.

Table 6
 L^2 error norm for three grid configurations: the moving cylinder case.

Force	UCG	UCG-L	UFG
$\Delta_x(t)$	0.00151	0.000771	0.000589
$\Delta_y(t)$	0.00942	0.005648	0.002972

Table 7
Computer CPU and memory occupied by three grid structures.

	UCG	UCG-L	UFG
CPU time (ms)	19.274	24.432	64.827
Memory (kB)	7400	9416	25480

grid interface, namely, both the value and slope at the fine-coarse boundary are continuous. This again validates our implementation of the local grid refinement.

The CPU time per running step and computer memory requirements are compared in Table 7. We find that the computing resources needed for UCG-L run are very similar to UCG, while these for UFG are much larger.

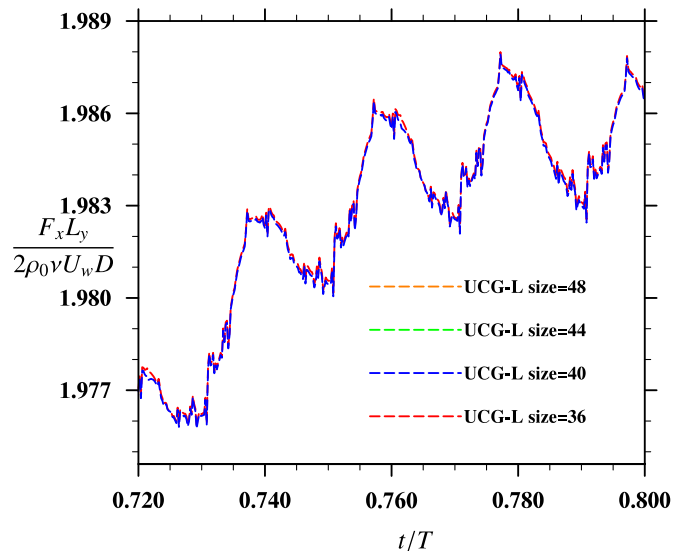


Fig. 22. Comparison of normalized F_x computed with four different sizes of the fine domain.

Table 8

The relative error in the shear stress for the Poiseuille flow simulated by UCG-L with different channel widths.

Ly	100	200	300	400
e	2.5706E-3	6.6931E-4	2.8548E-4	1.6212E-4

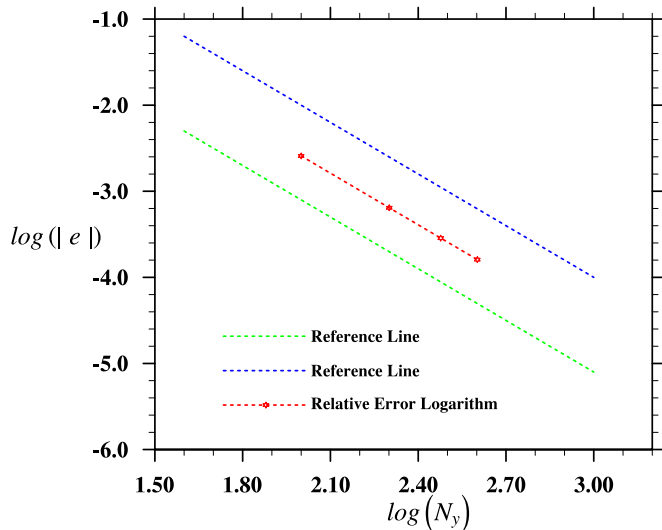


Fig. 23. The relative error in the shear stress for the Poiseuille flow simulated by UCG-L, as a function of channel grid resolution. The dash lines show a slope of -2 .

Fig. 22 shows the effect of fine-domain size in the computed F_x with four sizes of 36, 40, 44, 48. Here, the level of force fluctuations are independent of the fine-domain size. The minimum fine-domain size in this case is

$$S_{min} = D + 6, \quad (45)$$

which is 31.

5.2.3. Accuracy

Here, we use the 2D Poiseuille flow driven by a constant pressure gradient to examine if the numerical accuracy of the LBM scheme is affected by the local grid refinement. As shown in Fig. 14, the refinement domain size is fixed at 48×48 , the channel length is also fixed as $L_x = 200$. The maximum velocity in the mid-plane along the flow direction is fixed at $u_f = 0.1$. Four different channel widths ($Ly = 100, 200, 300, 400$) are considered. The pressure gradient is set to $8\nu u_f / (Ly)^2$, which is 8.9×10^{-6} , 2.22×10^{-6} , 9.87×10^{-7} and 5.57×10^{-7} , respectively. We calculated the relative error of the shear stress, defined as $e = (\tau - \tau_{th}) / \tau_{th}$, where τ is computed by the third equation of Eq. (18), the theoretical shear stress τ_{th} is,

$$\tau_{th} = \frac{4\nu u_f}{Ly} \left[2 \left(\frac{y}{Ly} \right) - 1 \right]. \quad (46)$$

The relative error data are shown in both Table 8 and Fig. 23. Clearly, the relative errors in Fig. 23 confirm that the local grid refinement treatment maintains the second-order accuracy of the lattice Boltzmann method.

5.2.4. Simplified treatment for the ghost moments

From the Chapman-Enskog analysis, energy square ε and energy fluxes q_x and q_y are ghost moments. They do not affect the hydrodynamic variables including the stress components. We use Eqs. (23)–(25) to relate the non-equilibrium parts of these moments. If these relations are ignored and we simply make each the

same in the coarse domain and fine domain, namely, by setting the diagonal conversion matrix defined by Eq. (32) as follows

$$T^f = \text{diag} \left[1, \frac{nS_e^f}{S_e^c}, 1, 1, 1, 1, 1, \frac{nS_v^f}{S_v^c}, \frac{nS_v^f}{S_v^c} \right]. \quad (47)$$

In order to reproduce the simulation results reported in Peng et al. [19], we also did an additional simulation with the second term in T^f set to one (denoted as the Peng et al. treatment). The results with these two simple alternative treatments are shown in Fig. 24 and are compared with results based on the formulation presented in Section 3. The results in Fig. 24 demonstrate that the relationships for the non-equilibrium parts of these ghost moments between the coarse and fine grids do not affect the resulting hydrodynamic forces and their fluctuations. This is anticipated as the non-equilibrium parts of these moments do not enter the Navier-Stokes equations. Here, the treatment in Eq. (47) for the ghost moments is the same as Peng et al. [19]. When compared to our treatment and the alternative one shown in Eq. (47), Peng et al. results show a slight difference of 0.03%. We also compared results for pressure and velocity divergence distributions along the channel centerline, it is found that the pressure distributions computed by three different treatments agree well with each other (not plotted here), but the velocity divergence from Peng's treatment appears to have larger oscillations near the domain interfaces than the results based on the treatments of Eqs. (32) and (47).

Therefore, although the energy moment is not relevant to the incompressible Navier-Stokes equations, it is better to treat this moment consistently, as suggested in this paper, in the lattice Boltzmann approach since the energy moment can have an impact on the local bulk viscosity. Fig. 25 shows that an inconsistent treatment of the energy moment can cause larger oscillations in velocity divergence.

6. Summary and conclusions

This paper was motivated by the desire to improve LBM simulation of the interaction of a moving particle with the carrier viscous flow. The use of uniform grid in LBM is not the ideal choice for resolving the viscous boundary layer near the surface of a moving solid particle. As one option, local grid refinement near the surface of the solid particle can be used to improve the simulation results.

We first re-examined the necessary relationships, within the MRT LBM approach, between the relaxation parameters and the distribution functions on the coarse and fine grids, in order to meet the physical requirements of the fluid hydrodynamics (continuity in pressure, velocity, and stress). We also proposed additional relations to relate the non-equilibrium components of the non-conservative moments, based on the Chapman-Enskog multi-scaling analysis. The details of grid arrangement, specifically the information transfer on interface (or buffer) layers have been presented. As known previously [6,9], the conversion between the two domains can be performed either before or after the collision sub-step, but it is stressed that the conversion relations are different for the two implementations. Both forms of the conversion relations have been developed in detail within the MRT LBM model. Although not shown explicitly in the paper, we found that the physical results from these two alternatives of information transfer between the two domains are identical provided that the hydrodynamic variables including the stress components are computed after the streaming sub-step. Our approach is general in that multiple levels of grid refinement could be implemented. The boundary between the coarse domain and fine domain can be arbitrary and move with the solid particle. We should note here that adaptive grids have already been developed for LBM simulations of various flows [14,23].

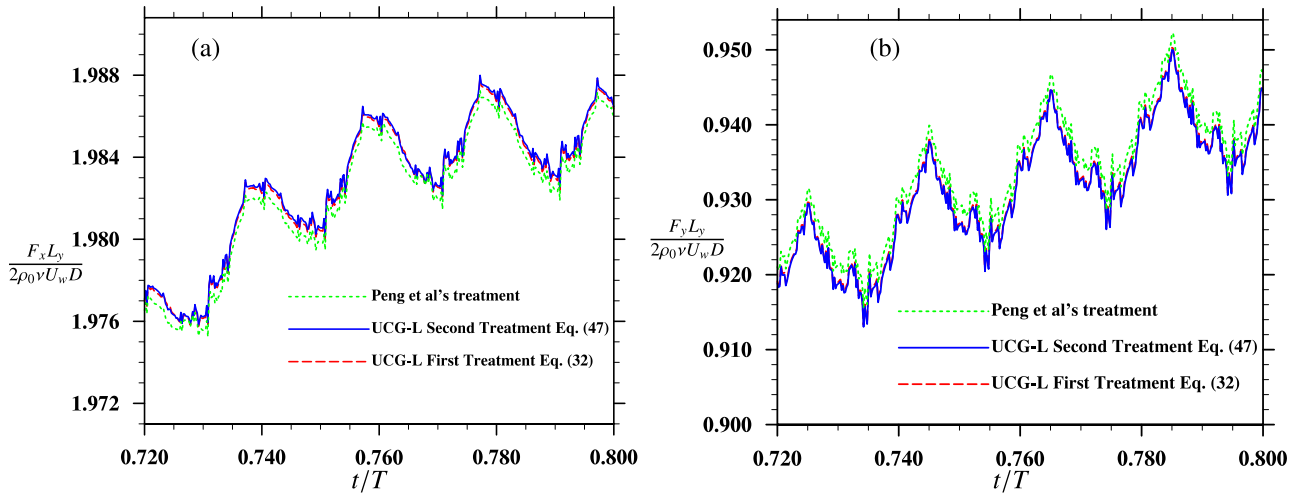


Fig. 24. The computed normalized hydrodynamic forces using different treatments of the conversion matrix, left: F_x ; right: F_y .

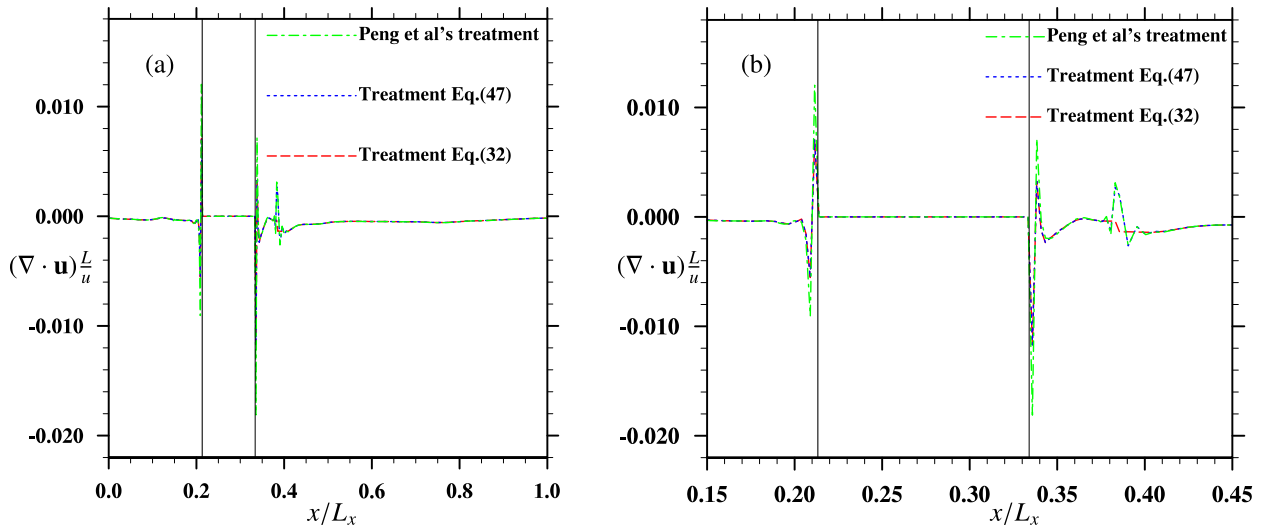


Fig. 25. The computed normalized velocity divergence using different treatments of the conversion matrix, left: whole view; right: zoom-in. The thin vertical lines denote the solid boundary of the particle.

Our approach was then applied to two numerical test cases to demonstrate that the local grid refinement can significantly improve the physical results with a high computational efficiency. Simulations from three grid configurations were compared: a uniformly coarse grid, a uniformly coarse grid with local refinement, and a uniformly fine grid. In addition to velocity profiles, stress profiles were carefully examined in these tests, which were rarely performed in previous studies. For the lid-driven cavity flow, the local refinement essentially yields a local flow field that is comparable to those based on the uniformly fine grid, but with much less computational cost. In the Couette flow with a moving cylinder, the local refinement suppresses the level of force fluctuations. Results from the moving particle test case show that even grid refinement in a small region surrounding the solid particle can significantly improve the simulation results, implying a great potential for the local grid refinement strategy in the lattice Boltzmann method for moving particle problems. Numerical tests showed that the local grid refinement treatment has very little influence on the numerical stability. We also confirm that the coarse-fine grid relationships between the non-equilibrium moments of energy square and energy fluxes do not affect the simulation results, as previ-

ously noted in Geller et al.[20]. Therefore, there is some flexibility at the domain interface which may be used to further optimize the numerical stability. We are in the process of applying our approach to a freely moving particle suspended in a turbulent flow. Local grid refinement could help resolve turbulent flow near the surface of solid particles while increasing accuracy and numerical stability [31]. There are, however, other difficulties in implementing the present algorithm due to multiple solid-particle interactions and related scalable implementation.

Acknowledgments

This work is supported by the [National Natural Science Foundation of China \(51176102\)](#). The work is also granted by the U.S. [National Science Foundation \(NSF\)](#) under grants [CBET-1235974](#) and [AGS-1139743](#) and by [Air Force Office of Scientific Research](#) under grant [FA9550-13-1-0213](#), LPW also acknowledges support from the Ministry of Education of P.R. China and Huazhong University of Science and Technology through Chang Jiang Scholar Visiting Professorship.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.compfluid.2016.06.009](https://doi.org/10.1016/j.compfluid.2016.06.009).

References

- [1] Chen S, Doolen G. Lattice Boltzmann method for fluid flows. *Annu Rev Fluid Mech* 1998;30:329–64.
- [2] Yu D, Mei R, Luo L-S, Shyy W. Viscous flow computations with the method of lattice Boltzmann equation. *Progr Aerospace Sci* 2003;39:329–67.
- [3] Raabe D. Overview of the lattice Boltzmann method for nano- and microscale fluid dynamics in materials science and engineering. *Model Simul Mater Sci Eng* 2004;12:R13–46.
- [4] Succi S. *The lattice Boltzmann equation for fluid dynamics and beyond*. Oxford: Clarendon Press; 2001.
- [5] Bhatnagar PL, Gross EP, Krook M. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component system. *Phys Rev* 1954;94:511–25.
- [6] Lallemand P, Luo L-S. Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, galilean invariance, and stability. *Phys Rev E* 2000;61:6546.
- [7] Humieres D, Ginzburg I, Krafczyk M, Lallemand P, Luo L-S. Multiple relaxation-time lattice Boltzmann models in three dimensions. *Philos Trans R Soc London A* 2002;360:437–51.
- [8] Mei R, Luo L-S, Lallemand P, d'Humieres D. Consistent initial conditions for lattice Boltzmann simulation. *Comput Fluids* 2006;35:855–62.
- [9] Filippova O, Hanel D. Grid refinement for lattice-BGK models. *J Comput Phys* 1998;147:219–28.
- [10] Filippova O, Hanel D. Acceleration of lattice-BGK schemes with grid refinement. *J Comput Phys* 2000;165:407–27.
- [11] Yu D, Mei R, Shyy W. A multi-block lattice Boltzmann method for viscous fluid flows. *Int J Numer Methods Fluids* 2002;39:99–120.
- [12] Yu D, Girmaji SS. Multi-block lattice Boltzmann method: extension to 3d and validation in turbulence. *Physica A* 2006;362:118–24.
- [13] Farhat H, Lee JS. Fundamentals of migrating multi-block lattice Boltzmann model for immiscible mixtures in 2D geometries. *Int J Multiphase Flow* 2010;36:769–79.
- [14] Crouse B, Rank E, Krafczyk M, Tolke J. A LB-based approach for adaptive flow simulations. *Int J Mod Phys B* 2002;17:109–12.
- [15] Eitel-Amor G, Meinke M, Schroder W. A lattice-Boltzmann method with hierarchically refined meshes. *Comput Fluids* 2013;75:127–39.
- [16] Lagrava D, Malaspinas O, Latt J, Chopard B. Advances in multi-domain lattice Boltzmann grid refinement. *J Comput Phys* 2012;231:4808–82.
- [17] Dietzel M, Sommerfeld M. Numerical calculation of flow resistance for agglomerates with different morphology by the lattice-Boltzmann method. *Powder Technol* 2013;250:122–37.
- [18] Premnath KN, Pattison MJ, Banerjee S. An investigation of the lattice Boltzmann method for large Eddy simulation of complex turbulent separated flow. *J Fluids Eng* 2013;135:051401–12.
- [19] Peng Y, Shu C, Chew YT, Niu XD, Lu XY. Application of multi-block approach in the immersed boundary lattice Boltzmann method for viscous fluid flows. *J Comput Phys* 2006;218:460–78.
- [20] Geller S, Tölke J, Krafczyk M. Lattice Boltzmann method on quadtree type grids for fluid structure interaction. In: *Lecture notes in computational science and engineering*, Volume 53. p. 270–93.
- [21] Geller S, Krafczyk M, Tölke J, Turek S, Hron J. Benchmark computations based on lattice Boltzmann, finite element and finite volume methods for laminar flows. *Comput Fluids* 2006b;35:888–97.
- [22] Tölke J, Freudiger S, Krafczyk M. An adaptive scheme using hierarchical grids for lattice Boltzmann multi-phase flow simulations. *Comput Fluids* 2006;35:820–30.
- [23] Stiebler M, Krafczyk M, Freudiger S, Geier M. Lattice Boltzmann large eddy simulation of subcritical flows around a sphere on non-uniform grids. *Comput Math Appl* 2011;61:3475–84.
- [24] Geller S, Uphoff S, Krafczyk M. Turbulent jet computations based on MRT and cascaded lattice Boltzmann models. *Comput Math Appl* 2013;65:1956–66.
- [25] Fakhari A, Lee T. Numerics of the lattice Boltzmann method on nonuniform grids: standard LBM and finite-difference LBM. *Comput Fluids* 2015;107:205–13.
- [26] Arora N, Gupta A, Shyy W. A shifting discontinuous-grid-block lattice Boltzmann method for moving boundary simulations. *Comput Fluids* 2016;125:59–70.
- [27] Chopard B, Droz M. *Cellular automata modeling of physical systems*. Cambridge University Press; 1998.
- [28] Shan X, Yuan XF, Chen H. Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation. *J Fluid Mech* 2006;550:413–41.
- [29] Ghia U, Ghia KN, Shin CT. High resolution for incompressible flow using the Navier-Stokes equations and a multi-grid method. *J Comput Phys* 1982;48:387–411.
- [30] Lallemand P, Luo LS. Lattice Boltzmann method for moving boundaries. *J Comput Phys* 2003;184:406–421.
- [31] Peng C., Teng Y., Hwang B., Guo Z., Wang L.-P. Implementation issues and benchmarking of moving particle simulations in a viscous flow. submitted to the ICMMES2014 special issue2016;.
- [32] Wen B, Zhang C, Yu Y, et al. Galilean invariant fluid-solid interfacial dynamics in lattice Boltzmann simulations. *J Comp Phys* 2014;266:161–70.