Computers and Fluids 125 (2016) 59–70

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

A shifting discontinuous-grid-block lattice Boltzmann method for moving boundary simulations

Nipun Arora [a], Amit Gupta [a,*], Wei Shyy [b]

[a] Department of Mechanical Engineering, Indian Institute of Technology, Hauz Khas, New Delhi, India
[b] Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China

ARTICLE INFO

Article history:
Received 20 July 2015
Revised 4 November 2015
Accepted 7 November 2015
Available online 17 November 2015

Keywords:
Lattice Boltzmann
Discontinuous-grid-block
Multi-relaxation time
Moving boundary

ABSTRACT

A translating discontinuous-grid-block model for moving boundaries of finite thickness based on multi-relaxation time version of lattice Boltzmann method has been developed. The implementation of this model to simulate moving boundary flows has been demonstrated for the cases of a cylinder in simple shear flow, a single rigid wing executing 'clap and fling' motion, and the propulsion of a plunging flat plate. A number of interpolation schemes of linear, quadratic and cubic natures are assessed around the discontinuous grid interface. It is shown that the implementation of a body-fitted refined mesh that moves along with the object reduces the spurious oscillations registered in the force and velocity measurements compared to a single coarse grid block. Moreover, use of multiple relaxation times helps overcome stability issues at high Reynolds number, normally encountered in the single-relaxation time model. Significantly, in the former model the same base grid could handle flows with good accuracy for $10 \leq Re \leq 1000$. The proposed technique offers significant advantage in terms of capturing flow around moving solids at lower computational cost and simulation time as compared to the stationary discontinuous-grid-block method.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

In the field of fluid mechanics, dealing with complicated processes like phase change and moving boundary flows has emerged as a big challenge, especially with ever increasing cost of experimental equipment and limited data availability [1]. However, with rapid improvements in computational facilities, design of systems via numerical analyses is being given more preference over conducting physical experiments. Most of these are based on the solution of Navier–Stokes equation and have been successful as well. However, the existing numerical schemes of re-meshing, grid generation and efficient matrix solvers, etc. that has garnered significant attention in the recent past suggest that there is still scope for improvement. Encountering various limitations in Navier–Stokes based mathematical models and experiencing complexities in implementation, researchers started opting for other alternatives. In the past decade, the lattice Boltzmann method (LBM) emerged as one such substitute which enjoys certain advantages over existing techniques. Some of the salient features are (a) the solution process is local in nature and hence there is no requirement of solving simultaneous linear algebraic equations which makes the solution process non-iterative and free of matrix inversions, and (b) it is easy to implement and parallelize.

By convention, LBM utilizes a regular, uniform and stationary Cartesian grid for solving the discretized Boltzmann equation where the particle distribution functions are calculated and whose hydrodynamic moments provide the macroscopic variables (density, velocity and temperature) [2–6]. The most commonly used version is the BGK (Bhatnagar, Gross and Krook) or also known as the single-relaxation time (SRT) model. Despite its advantages, the BGK model poses numerical instability at low values of relaxation time and hence is difficult to use for flows at higher Reynolds number (Re). Contrary to SRT, the multi-relaxation time (MRT) model exhibits better numerical stability even at high Re (i.e., at very low values of relaxation time) and has been quite successful in curtailing the spurious oscillations registered in force measurement [7,8].

Although quite successful in reducing noise and fluctuations observed in the calculation of forces, MRT does not eliminate them completely, especially for moving bodies. This is due to the variability in shape or volume of the object, i.e., the number of nodes or grid points inside the solid do not remain constant as these undergo transitions from non-fluid region to fluid region or vice versa due to the movement of the solid (which contributes to the noise). As demonstrated with the help of Fig. 1 which shows the standard LBM interpretation of a solid body immersed in fluid, this issue can be resolved by improving the resolution of moving body by refining mesh or grid

* Corresponding author. Tel.: +911126591270; fax: +91 11 26582053.
E-mail address: agupta@mech.iitd.ac.in, amitgupta2004@gmail.com (A. Gupta).

http://dx.doi.org/10.1016/j.compfluid.2015.11.003
0045-7930/© 2015 Elsevier Ltd. All rights reserved.

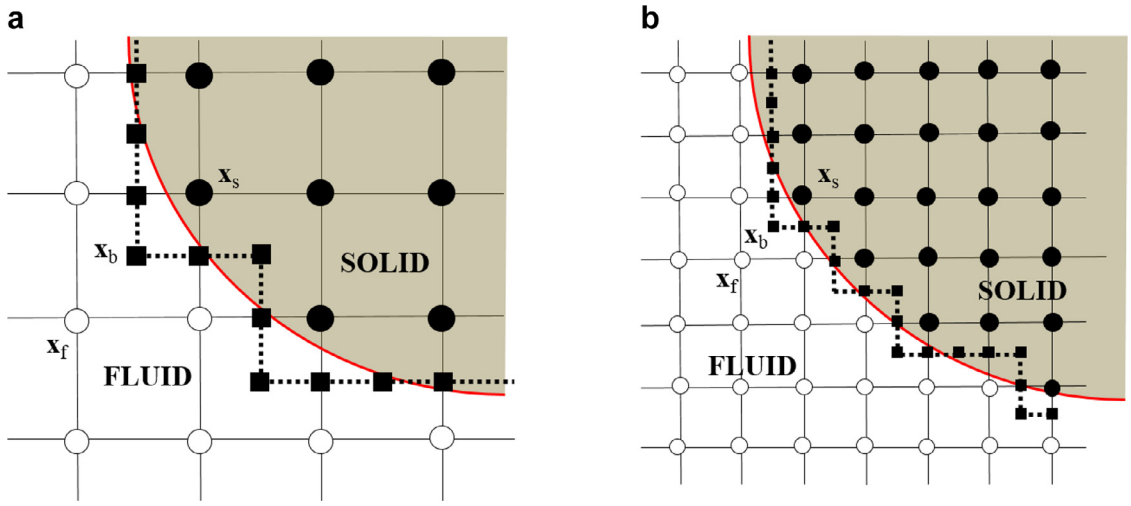

Fig. 1. Layout of the regularly spaced lattices and curved solid boundary (solid red). The hollow and filled circles denote fluid and solid nodes respectively. The solid squares denote the boundary nodes. The dotted line represents the halfway bounceback interpretation of the curved geometry with (a) normal and (b) increased resolution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

size near the surface of the solid to capture the minutest of details. Here, the outline of a moving object is represented by a set of fixed nodes lying exactly halfway between the immediate solid and fluid nodes on either side of its boundary. The true shape of a moving body can also be preserved by the employment of the immersed boundary methods, where the boundary is represented by a set of nodes which are not stationary (as in standard LBM) but are constantly moving with the body [9]. But in either case, as the mesh size decreases, the resolution of an object will improve which can be expected to lead to increased accuracy in results obtained.

Regrettably, refinement of the grid throughout the domain (even in those regions where the flow is not expected to evolve rapidly and drastically) is computationally expensive and inefficient. This problem can be overcome by using a coarser mesh far away from the body thereby creating zones of different mesh sizes, which we term as the discontinuous-grid-block method in LBM. Creation of separate coarse and fine block can greatly reduce the computational time and memory in comparison to the employment of uniform grid throughout. In the past, several groups have demonstrated the principles of the discontinuous-grid-block method applied to flows past stationary solids [10–12]. However, for a moving body, developing a body-fitted mesh which is dynamic and moves along with the object on a 'carpet' of the coarse block is essential to get improved accuracy and at the same time maintaining a reasonable computational efficiency.

Thus, in the present study we extend the earlier work carried out by Yu et al. [10,11] and Peng et al. [12] on the stationary discontinuous-grid-block (SRT and MRT, respectively) by developing an algorithm for situations where the finer block is desired to translate with the moving solid (and in essence, a shifting discontinuous-grid-block LBM). Although the SRT version of the moving block LBM has been developed previously [13], the focus of this work is to highlight and illustrate the details of this translating multiblock method using MRT version of LBM. Additionally, the method as applied to the analysis of flows with $1 \leq \mathrm{Re} \leq 1000$ is also demonstrated and discussed from stability and accuracy considerations.

2. Methodology

2.1. Multi-relaxation time LBM

The detailed description of the SRT as well as MRT uniform grid models, calculation of macroscopic variables from the distributions, the force evaluation on a moving body using halfway bounceback method (as proposed by Ladd [6]) along with proper validations are provided in earlier publications [14,15]. However, since the multi-block method described here is based on the MRT version, it is re-iterated here to avoid any confusion or discontinuity.

In this method, the distribution function f (as defined in the SRT model) in the discrete velocity space **B** is mapped onto the moment space **K** by using a transformation matrix M [8], i.e.,

$$\hat{f} = Mf \qquad and \qquad f = M^{-1}\hat{f} \tag{1}$$

where $\hat{f}$ is a column matrix consisting of moments of the velocity distribution function (each row vector) which represents the following quantities in two-dimensional space:

$$\hat{f} = [\rho, e, \varepsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy}]^T \tag{2}$$

where ρ is the density, e signifies the kinetic energy, ε epitomizes the square of kinetic energy, j_x and j_y are the x and y components of the momentum flux, q_x and q_y are related to the x and y components of energy density, and p_{xx} and p_{xy} refer to the diagonal and off-diagonal terms in viscous stress tensor. Since the moments represent different physical quantities, MRT enjoys an added advantage of independently varying the relaxation time scales. Also, the transformation matrix is orthogonal (i.e. $M.M^T = I$) which ensures that the relaxation time matrix in moment space **K** comes out to be diagonal [16].

The equilibrium values for the non-conserved moments can be obtained from the conserved moments ρ and $\boldsymbol{j}$ (j_x and j_y) [8] as

$$e^{eq} = -2\rho + \frac{3}{\rho}\left(j_x^2 + j_y^2\right) \tag{3}$$

$$\varepsilon^{eq} = \rho - \frac{3}{\rho}\left(j_x^2 + j_y^2\right) \tag{4}$$

$$q_x^{eq} = -j_x, \quad q_y^{eq} = -j_y \tag{5}$$

$$p_{xx}^{eq} = \frac{1}{\rho}\left(j_x^2 - j_y^2\right) \tag{6}$$

$$p_{xy}^{eq} = \frac{1}{\rho} j_x j_y \tag{7}$$

The transformation matrix (orthogonalized by Gram-Schmidt procedure [17]) as employed by Lallemand and Luo [8] for the D2Q9

model is given by

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{8}$$

The collision process is performed in the **K** space which is as follows:

$$\hat{f}_* = \hat{f} - S[\hat{f} - \hat{f}^{eq}] \tag{9}$$

where $\hat{f}_*$ is the post collision operator, and S is the diagonal relaxation time matrix

$$S = diag[s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9] \tag{10}$$

where $s_1 = s_4 = s_6 = 0$ (relaxation time for the conserved quantities).

The moments $\hat{f}$ are transformed back and thus the advection step is carried out in the **B** space on the distribution function f. Thus, Eq. (9) can be written as

$$\tilde{f} = M^{-1}\hat{f} - M^{-1}S[\hat{f} - \hat{f}^{eq}] \tag{11}$$

where $\tilde{f}$ is the post collision distribution function in **B** space which are then streamed to the neighboring nodes in direction of the lattice velocities. It should be noted that the MRT model reduces to the single-relaxation time BGK model if all the diagonal terms are equal to $s = 1/\tau$, where τ is the single-relaxation time.

2.2. Discontinuous-grid-block MRT

In multi-block LBM, the fine and the coarse mesh overlap over a thickness of δx_c (unit lattice length in coarse block) with their respective boundaries lying in each other's zone as is shown in Fig. 2 [10]. The ratio of the coarser to finer grid size is given as

$$m = \frac{\delta x_c}{\delta x_f} \tag{12}$$

In order to keep lattice velocities ($c = \frac{\delta x}{\delta t}$) same across the grid boundary, lattice time steps must follow the same relation as stated above i.e.

$$\frac{\delta t_c}{\delta t_f} = \frac{\delta x_c}{\delta x_f} = m \tag{13}$$

With the MRT, the usual procedure of collision and streaming in LBM is followed in both the blocks with a slight modification at the interface where collision step is omitted. This is because after streaming, distributions at these nodes are unknown. But since the two blocks overlap with each other, i.e., the fine block boundary lies inside the coarse block and vice versa, the required information can be obtained by transferring the data from interior nodes of either block to the coinciding boundary nodes of the other after completion of every time step in the coarse lattice. Thus a relationship needs to be established between the distribution functions of these blocks and is derived below.

Certain differences between the SRT and MRT implementations also need to be highlighted. First, the viscosity in the SRT model of LBM is calculated as

$$\nu = \frac{(2\tau - 1)\delta x\, c}{6} \tag{14}$$

where τ is the single relaxation time and c is the lattice speed which is equal to unity in the non-dimensional form of LBE. Hence, to maintain the consistency of viscosity across the interface, the relationship between the coarse and fine grid relaxation times can be written as

$$\tau_f = \frac{1}{2} + m\left(\tau_c - \frac{1}{2}\right) \tag{15}$$

However, the MRT model is based on different relaxation times for all moments of the distribution function. Hence, Eq. (15) has to be rewritten as

$$\frac{1}{s_i^f} = \frac{1}{2} + m\left(\frac{1}{s_i^c} - \frac{1}{2}\right), \qquad for\, i = 2, 8, 9 \tag{16}$$

Similar to the SRT model, the distribution function $\hat{f}$ in **K** space can be written as

$$\hat{f} = \hat{f}^{eq} + \hat{f}^{neq} \tag{17}$$

where $\hat{f}^{eq}$ and $\hat{f}^{neq}$ are the equilibrium and non-equilibrium parts of the distribution function. In order to maintain the continuity of the velocity and density across the interface, the equilibrium distribution functions should be the same in both blocks,

$$\hat{f}^{eq,\,c} = \hat{f}^{eq,\,f} \tag{18}$$

The stresses depend on the non-equilibrium part of the distribution function and are evaluated as

$$p_{xx} = \left(1 - \frac{s_8}{2}\right)(p_{xx} - p_{xx}^{eq}) = \left(1 - \frac{s_8}{2}\right)p_{xx}^{neq} \tag{19}$$

$$p_{xy} = \left(1 - \frac{s_9}{2}\right)(p_{xy} - p_{xy}^{eq}) = \left(1 - \frac{s_9}{2}\right)p_{xy}^{neq} \tag{20}$$

Hence, in a similar manner to the viscosity, the continuity of stresses in the blocks is satisfied by equating them in the coarse and fine grids, which gives the following relationship

$$p_{xx}^{neq,\,c} = m\,\frac{s_8^f}{s_8^c}\,p_{xx}^{neq,\,f} \tag{21}$$

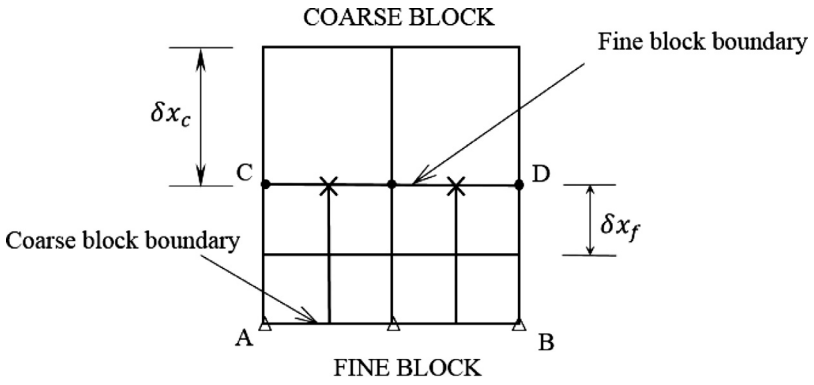

Fig. 2. The interface between coarse and fine block with lattice spacing ratio $m = \delta x_c/\delta x_f = 2$. The coarse block boundary lies inside the fine block whose nodes are represented by Δ. The fine block boundary lies inside the coarse block whose nodes are represented by $\bullet$ and $\times$.

$$p_{xy}^{neq,\,c} = m\,\frac{s_9^f}{s_9^c}\,p_{xy}^{neq,\,f} \tag{22}$$

Thus, the non-equilibrium part of distribution function in the coarse and fine blocks is related by the following expression:

$$\hat{f}^{neq,\,c} = \begin{bmatrix} 0 \\ e-e^{eq} \\ \varepsilon-\varepsilon^{eq} \\ 0 \\ q_x-q_x^{eq} \\ 0 \\ q_y-q_y^{eq} \\ p_{xx}-p_{xx}^{eq} \\ p_{xy}-p_{xy}^{eq} \end{bmatrix}^c = T \begin{bmatrix} 0 \\ e-e^{eq} \\ \varepsilon-\varepsilon^{eq} \\ 0 \\ q_x-q_x^{eq} \\ 0 \\ q_y-q_y^{eq} \\ p_{xx}-p_{xx}^{eq} \\ p_{xy}-p_{xy}^{eq} \end{bmatrix}^f = T\,\hat{f}^{neq,\,f} \tag{23}$$

where $T = diag[1, 1, 1, 1, 1, 1, m\frac{s_8^f}{s_8^c}, m\frac{s_9^f}{s_9^c}]$.

From Eqs. (11), (17), (18) and (23), a relationship between the coarse and fine block distributions at the interface to exchange information can be obtained which is as follows:

$$\begin{aligned} \tilde{f}^c &= M^{-1}\hat{f}^{eq,\,c} + M^{-1}(1-S^c)\hat{f}^{neq,\,c} \\ &= M^{-1}\hat{f}^{eq,\,f} + M^{-1}(1-S^c)T\hat{f}^{neq,\,f} \end{aligned} \tag{24}$$

Eliminating $\hat{f}^{neq,\,f}$, the above equation can be further simplified as

$$\tilde{f}^c = M^{-1}[\hat{f}^{eq,f} + (1-S^c)T(1-S^f)^{-1}(M\tilde{f}^f - \hat{f}^{eq,\,f})] \tag{25}$$

Similarly, the transfer of distributions from coarse to fine block can also be derived,

$$\tilde{f}^f = M^{-1}[\hat{f}^{eq,\,c} + (1-S^f)(1-S^c)^{-1}T^{-1}(M\tilde{f}^c - \hat{f}^{eq,\,c})] \tag{26}$$

The above equations for MRT have also been separately derived by Peng et al. [12].

With reference to Fig. 2, the post collision distributions at nodes Δ can be obtained using Eq. (25) on the coarse block boundary AB. Similarly, the post collision distributions at nodes $\bullet$ can be determined using Eq. (26) on the fine block boundary CD. However, the distributions at nodes $\times$ are not known and can be calculated by spatially interpolating the information that was transferred to nodes $\bullet$.

2.3. Spatial and temporal interpolation

In previous studies [10,12], cubic spline interpolation (based on the idea of a piecewise defined function in the subintervals) has been implemented which is expressed as

$$F_i(x) = a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3, \quad i = 1, 2, \ldots\ldots, n \tag{27}$$

where i is the sub-interval in which the above equation is applicable for $x_{i-1} \le x \le x_i$. Thus, there are n third-degree polynomials in each interval which are used to interpolate the values at intermediate points. a_i, b_i, c_i, d_i are the $4n$ unknown coefficients which are obtained by satisfying the following conditions [10,18]:

1. The nodal values $(y_0, y_1, \ldots\ldots, y_n)$ must be known.
2. The spline should be continuous across the intervals which can interpolate all the nodes i.e. $F_i(x_{i-1}) = y_{i-1}$ and $F_i(x_i) = y_i$ for $i = 1, 2, \ldots\ldots, n$.
3. The first and second derivatives should be continuous at the interior nodes i.e. $F_i'(x_i) = F_{i+1}'(x_i)$ and $F_i''(x_i) = F_{i+1}''(x_i)$ for $i = 1, 2, \ldots\ldots, n-1$.

The above conditions constitute $4n-2$ equations. Thus, a suitable end condition of zero second derivatives i.e. $F_1''(x_0) = F_n''(x_n) = 0$ is applied to determine the remaining two unknowns [10].

As stated, the lattice time step in fine block is $1/m^{th}$ of that in the coarse block and the transfer of distributions are performed at the end of a unit time in the latter. Thus, at the intermediate time step in fine grid and with reference to Fig. 2, the post collision distributions at boundary CD are again unknown which are determined using temporal interpolation based on p- point Lagrangian formula, given by

$$y(t) = \sum_{k=1}^{p} y_k \left(\prod_{\substack{j=1 \\ j\neq k}} \frac{t-t_j}{t_k-t_j} \right) \tag{28}$$

Yu et al. [10] used a 3-point Lagrangian function with $t_k = t_{n-1},\ t_n$ and t_{n+1} for temporally interpolating the unknown distributions at $t_{n+1/m}$.

Further, we also apply and study the use of Lagrangian 2-point (linear interpolation), 3-point (quadratic interpolation) and 4-point (cubic interpolation) schemes (as explained in the Appendix) to spatial interpolations along with cubic spline and report their influence on the final outcome. The selection of a particular combination of interpolations requires a careful analysis of the merits and deficiencies of each method over others. Thus, simulations using moving discontinuous-grid-block performed with these interpolations are compared against the results obtained using its stationary counterpart with same grid refinement ratio. Based upon the comparison, a hierarchical order of these interpolations can be generated which will be useful in the assessment of these methods. In addition, the computation time required for carrying out these simulations was also compared between the stationary and moving discontinuous-grid-block.

2.4. Moving discontinuous-grid-block algorithm

The algorithm for moving discontinuous-grid-block ($m = 2$) is explained through a flow chart shown in Fig. 3. The procedure for a stationary discontinuous-grid-block can be found in literature [10–12]. However, for the case of a moving finer block, some key modifications need to be incorporated. The fine grid will shift by $m\,\delta x_f$ along the X or Y direction as the object in the fine grid moves by δx_c in either direction. Hence, at the end of a unit time step in coarse grid δt_c, the distance propagated by the object in fine grid is examined. If it exceeds δx_c, a layer of fine grid is added (removed) along (against) the direction of translation, as is shown in Fig. 4. In the newly added layer, the unknown distributions $\tilde{f}^f$ on new fine block boundary (line 14 in Fig. 4(b)) can be obtained by the transfer from known coarse block distributions $\tilde{f}^c$ using Eq. (26) and at intermediate grid points using cubic spline or Lagrangian interpolation. For distributions on line 13, various interpolations can be used to determine the unknown distributions which are discussed next. The missing distributions $\tilde{f}^c$ on new coarse block boundary (line 4 in Fig. 4(b)) can be determined using the information at coinciding fine grid nodes with the help of Eq. (25) (the distributions on remaining coarse block boundaries are already known). Note that the fine grid remains stationary as long as the distance traversed by the moving body (from its position at the time instant when fine grid had moved earlier) is less than δx_c.

2.4.1. Interpolation schemes around discontinuous grid interface

Even though a variety of interpolation methods exist in the literature, we limit our attention in the present work to only two kinds of interpolations, namely Lagrangian p point and cubic spline. With reference to Fig. 4, the various combinations possible to estimate the unknown distributions on grid points of line 13 in newly added layer are stated in Table 1. For line 14, the interpolations are executed in a similar fashion as in a stationary discontinuous-grid-block.

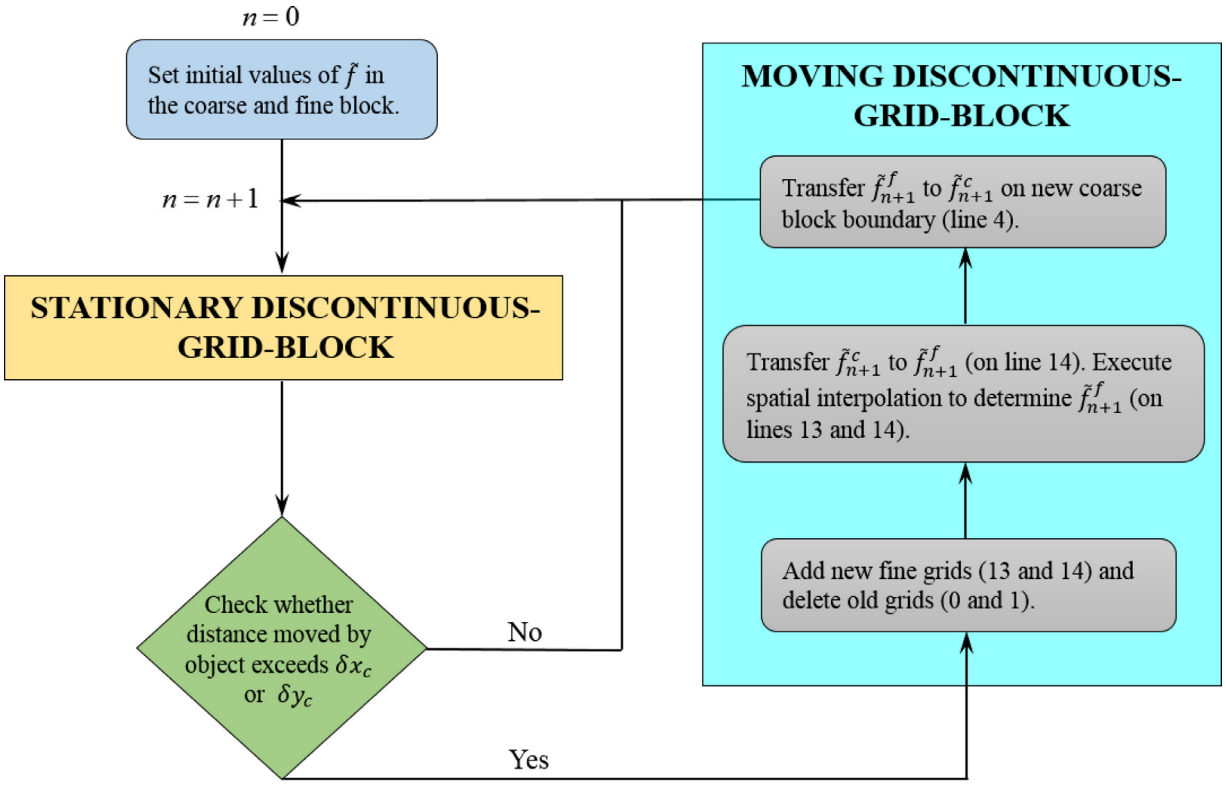

Fig. 3. Flow chart of the computational procedure for moving discontinuous-grid-block as implemented in present study with reference to Fig. 4..

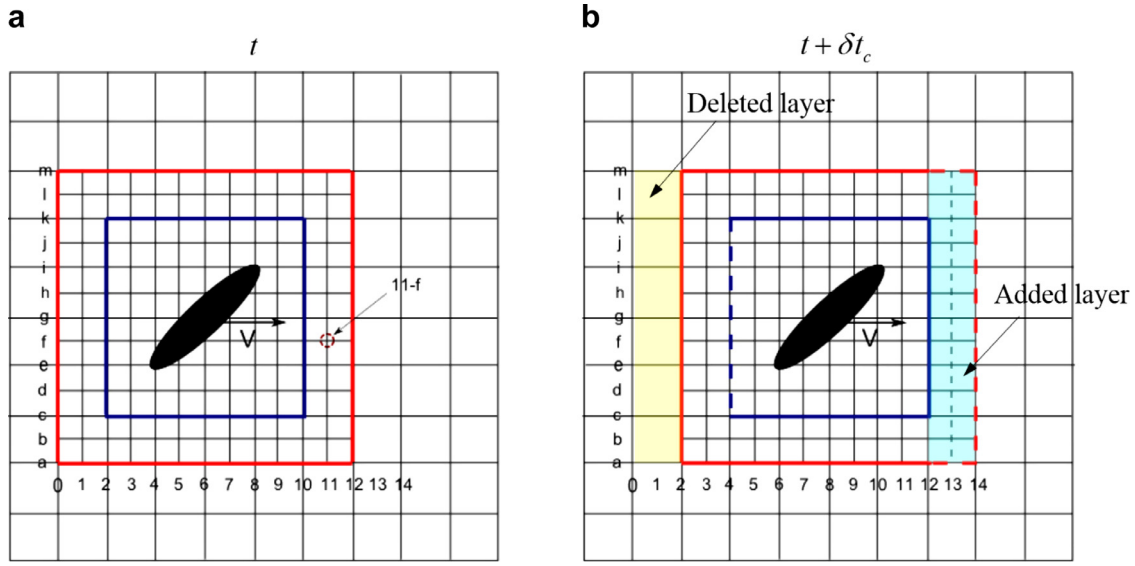

Fig. 4. The fine block is displaced by $2\,\delta x_f$ as the object traverses δx_c in coarse block at time $t + \delta t_c$. Red and blue lines signify the fine block and coarse boundaries, respectively. The dashed lines represent unknown distributions which need to be determined either through transfer from other block or via interpolations. The marked numbers and alphabets denote the columns and rows respectively, which are used to characterize the coordinates of fine grid nodes or coinciding coarse grid nodes as represented by dotted circle in (a). Coordinates are measured with respect to a stationary frame as the fine block moves from time t to $t + \delta t_c$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Similarly, these schemes for determining the unknown distributions on line 13 (Fig. 4(b)) can be applied to linear/quadratic/cubic (Lagrangian 2/3/4 point) spatial interpolations as well. Unlike cubic spline which considers all the grid points, for linear, quadratic and cubic Lagarangian interpolations only 2 (12a and 14a), 3 (10a, 12a and 14a) and 4 (8a, 10a, 12a and 14a) neighboring points are needed, respectively, to interpolate the information at 13a (refer the formulae given in appendix). Until and unless specified, scheme 1 (using coarse grid points) was used for interpolating the unknown distributions on newly added fine layer in all the cases discussed in the following section.

The above set of combinations are described for $m = 2$. Similarly, many more combinations can be framed for $m = 3$, 4 or higher as additional layers are created.

Table 1
Various interpolation schemes in newly added layer for grid points of line 13.

Scheme	Description
Spline with coarse grid points	Fit a cubic spline function horizontally choosing only coarse grid nodes, i.e., (2-a,4-a, 6-a…,14-a) and then interpolate to determine the distributions at 13-a. Apply a similar process in the transverse direction to determine the distributions at all grid points of line 13, i.e., (13-b, 13-c,……,13-m).
Spline with fine grid points	Fit a cubic spline function horizontally choosing all nodes of the fine grid i.e. (2-a, 3-a, 4-a…, 14-a) and then interpolate to determine the value at 13-a. Apply a similar process in the transverse direction to determine the distributions at all grid points of line 13, i.e., (13-b, 13-c,……,13-m).
Spline-spline with coarse grid points	Fit a cubic spline function horizontally choosing only coarse grid nodes i.e. (2-a,4-a, 6-a…,14-a) and then interpolate to determine the value at 13-a. Apply a similar process for all coarse grid points of line 13, i.e., (13-c, 13-e, 13-g,....., 13-m). Now on line 13, fit a vertical spline by selecting the newly computed coarse grid points to compute the distributions at remaining nodes i.e. (13-b, 13-d, 13-f,……,13-l).

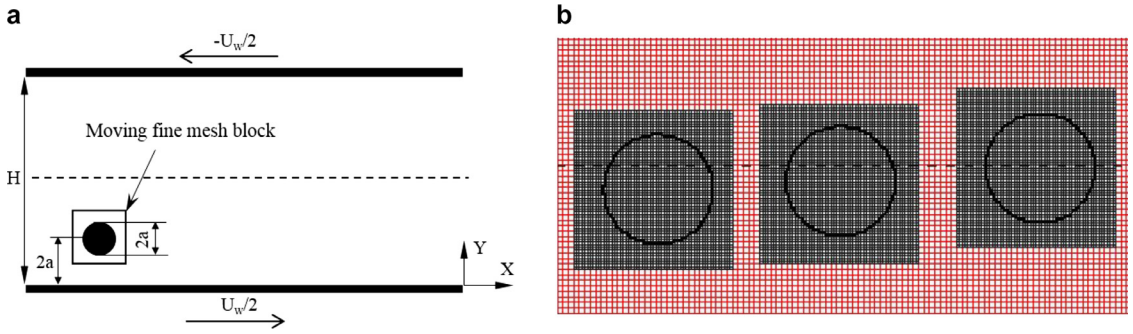

Fig. 5. (a) A schematic diagram of neutrally buoyant cylinder in a simple shear flow between two walls initially placed at $y = 0.25H$. (b) Positions of the cylinder and moving fine mesh block at different time instants during their movement towards the centerline. Red mesh indicates the coarse block. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

3. Results and discussion

To analyze the performance of the dynamic discontinuous-grid-block MRT LBM, numerical simulations were performed for some standard moving boundary problems described below. In all these cases standard halfway bounceback method has been used for imposing no slip boundary condition and the momentum exchange method for evaluating the forces experienced by the solid [6].

3.1. Cylinder in a shear flow

The motion of neutrally buoyant ($\rho_s = \rho_f$) circular cylinder initially placed at $y = 0.25H$ with the upper and lower walls moving in opposite directions at velocity $U_w/2$ for $Re = 0.625$ with confinement ratio $H/a = 8$ was studied. Here, Reynolds number is defined as $Re = Ga^2/\nu$, where 'G' is the shear rate ($G = U_w/H$), 'a' is the radius of cylinder and 'ν' is the kinematic viscosity of the fluid. The schematic diagram of the problem solved is shown in Fig. 5(a). Owing to the non-uniform velocity distribution in the flow, a pressure difference is created above and below the cylinder. This causes it to drift towards the center of the domain beyond which it ceases to travel in the vertical direction. A base grid of size $2000\Delta x_c \times 80\Delta x_c$ lattices was considered with the center of cylinder initially placed at $(200\Delta x_c, 20\Delta x_c)$. For the fine block (with grid refinement ratio $m = 2$) and as shown by the temporally moving grid in Fig. 5(b), $58\Delta x_f \times 58\Delta x_f$ nodes were taken around the cylinder. However with a stationary multi-block, $3240\Delta x_f \times 130\Delta x_f$ nodes had to be considered around the cylinder to ensure its confinement within fine block throughout its travel to the center of the domain. Thus, the resolution of cylinder (to obtain same level of precision in size and force measurement) is same in both cases but the latter encompasses a larger number of lattices. Fig. 6 compares the time evolution of the two components of dimensionless cylinder translational velocity (where U_P and V_P are the dimensional horizontal and vertical velocity components, respectively) between the stationary and dynamic discontinuous-grid-block MRT (cubic spline interpolation). A good match in both horizontal and vertical velocity for the two cases was observed. In addition for the sake of validation, horizontal velocity profile recorded in both cases for the conditions mentioned, was compared with the published results of Feng and Michaelides [9]. For Gt<10, the flow evolvement is very rapid and very high horizontal velocity is recorded. In the immersed boundary formulation, no slip (boundary node velocity matches the fluid velocity) condition requires an interpolation function to map the fluid nodes data to solid nodes and vice versa. As interpolations involved have a certain level of accuracy, some difference in horizontal velocity profile is observed. However, for Gt>10, both results show good agreement whereas the vertical velocity profile shows consistent results throughout. In terms of computational resources,

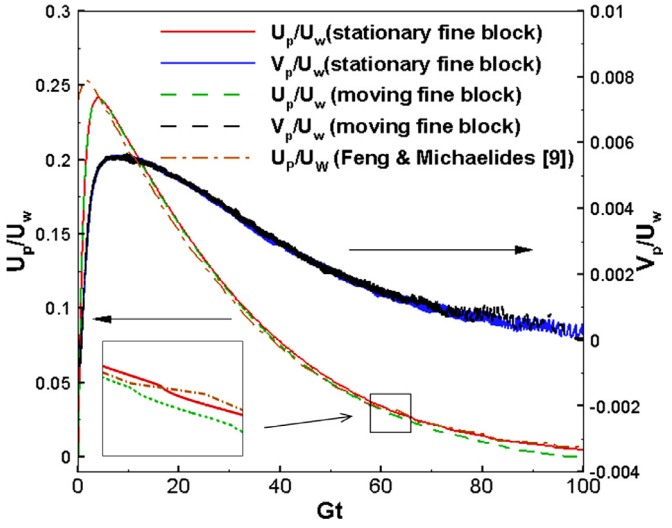

Fig. 6. Comparison of horizontal and vertical dimensionless velocities of the cylinder with dimensionless time for stationary and moving discontinuous-grid-block ($m = 2$) MRT model.

Table 2
A comparison of computation time for stationary and moving discontinuous-grid-block MRT for different cases.

Case study	Stationary fine grid			Moving fine grid		
	No. of fine lattices ($\times 10^6$)	Total no. of lattices ($\times 10^6$)	Computation time (min)	No. of fine lattices ($\times 10^6$)	Total no. of lattices ($\times 10^6$)	Computation time (min)
Cylinder in a shear flow	0.421	0.476	5660	0.0033	0.163	600
Clap and Fling (1 cycle)	0.2	2.71	340	0.04	2.59	300
Rigid plunging flat plate	2.0	11.5	1410	0.08	10.06	640

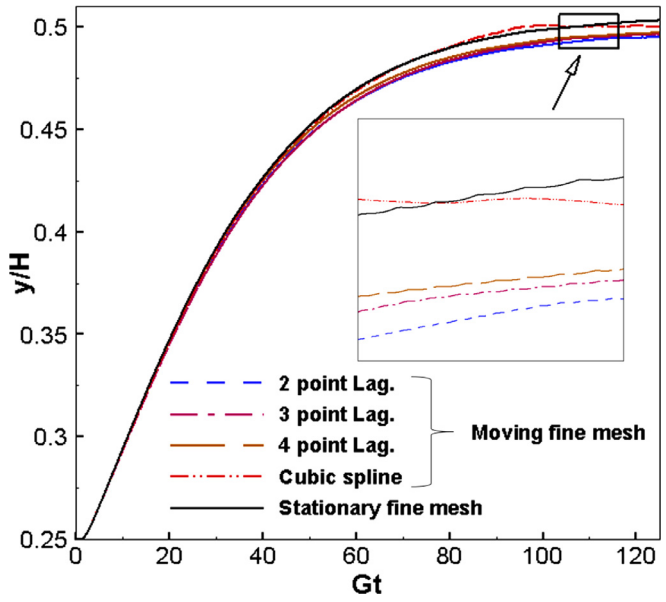

Fig. 7. Comparison of the vertical trajectory as a function of dimensionless time between stationary and moving fine mesh ($m = 2$) MRT with different interpolation schemes. Inset shows the zoomed version of the window.

a drastic reduction in computation time with the implementation of moving discontinuous-grid-block in comparison to its stationary counterpart is shown in Table 2.

To evaluate the accuracy of various interpolation methods adopted in the shifting discontinuous-grid-block algorithm, the vertical trajectory of cylinder was also compared. Fig. 7 shows a comparison of the vertical trajectory as a function of non-dimensional time Gt between the stationary grid-block MRT and shifting discontinuous-grid-block MRT with linear, quadratic, cubic Lagrangian and cubic spline interpolations. The profiles are almost similar until the cylinder starts to approach the centerline of the domain, near which there is a minor deviation in results (to meticulously distinguish the difference, a zoomed in section is shown as an inset). The linear (2-point Lagrangian), quadratic (3-point Lagrangian) and cubic (4-point Lagrangian) profiles are nearly coinciding but marginally underpredict the trajectory profile. However, with the increase in order of Lagrangian interpolation, the profile starts approaching the actual curve. The cubic spline interpolation curve has the best resemblance with the true trajectory.

3.2. Clap and fling

The moving grid algorithm was further tested by conducting numerical simulations on a single wing employing 'clap and fling' motion. The Reynolds number is defined as $\mathrm{Re} = \mathrm{VC}/\nu$, where V is the maximum translational velocity, C is chord of the wing and ν is the kinematic viscosity of the fluid. The wing was assumed to be rigid and elliptical in shape with the thickness as 2.5% of the chord. A computational domain of size $20C \times 20C$ was considered. A pictorial representation of the computational domain is shown in Fig. 8(a). Periodic boundary conditions were imposed on all four boundaries of the domain. Fig. 8(b) shows the positions of the wing at various time instants performing the clap and fling motion for $\xi = 50\%$ (translational rotational overlap). This motion for center of mass of the wing was developed using the kinematics as described in [14].

The chord C was chosen to be $80\Delta x_c$ and a moving discontinuous-grid-block of $200\Delta x_f \times 200\Delta x_f$ nodes ($100\Delta x_c \times 100\Delta x_c$ coarse lattice units) was taken around the wing. In addition to this, simulation was also performed using a stationary discontinuous-grid-block MRT. For the current kinematics, the wing traverses about 5 chord lengths in the horizontal direction (Fig. 8(b)). Therefore, a multi block of size $1000\Delta x_f \times 200\Delta x_f$ lattices in the fine grid was chosen so that its upstroke-downstroke movement was always within the finer mesh domain. The instantaneous lift and drag coefficients for the two-dimensional motion were calculated using

$$C_L = \frac{F_L}{\frac{1}{2}\rho C V^2} \tag{29}$$

$$C_D = \frac{F_D}{\frac{1}{2}\rho C V^2} \tag{30}$$

where C_L and C_D are the lift and drag coefficients per wing, respectively. F_L is the lift force per unit spanwise length, F_D is the drag force per unit spanwise length and ρ is the density of the fluid. These time-dependent forces F_L and F_D acting on the wing were calculated by the momentum exchange method as explained in [14]. Fig. 9(a) and (b) show a comparison of lift and drag coefficients for the stationary and moving discontinuous-grid-block MRT at $\mathrm{Re} = 128$. A good agreement in both lift and drag can be observed, again indicating that the moving grid accurately captures the solid-fluid interaction. The first peak in the lift coefficients (in the first half of cycle or fling/downstroke) corresponds to the lift forces generated during wing rotation whereas the second peak corresponds to the period of translational acceleration. With moving discontinuous-grid-block, the computation time (for one cycle) was recorded to be 11% lower and the memory consumption was 6.2% lesser than its stationary counterpart. Opposite to the case of cylinder in shear flow, however, in this case the reduction in computation time was comparatively lesser since the wing translated back and forth unlike the former where the cylinder propagated indefinitely in one direction. A comparison of the actual computation time expended in stationary and moving discontinuous-grid-block MRT for this simulation on a single core machine is given in Table 2.

In order to further show the applicability of this method at higher Re and its superiority over conventional uniform grid or stationary single-block SRT, simulations were carried out up to $\mathrm{Re} = 1000$ for a single wing undergoing 'clap and fling'. With SRT, an increase in Re resulted in a corresponding increase in fluctuations recorded in lift and drag coefficients. Additionally, at very high Re the simulations were no longer stable. Fig. 9(c) and (d) show the comparison between lift and drag coefficients obtained with single-block SRT and moving discontinuous-grid-block MRT for Re=128. Large fluctuations in force calculation, especially during $t/T = 0.1$–0.4 when the wing is

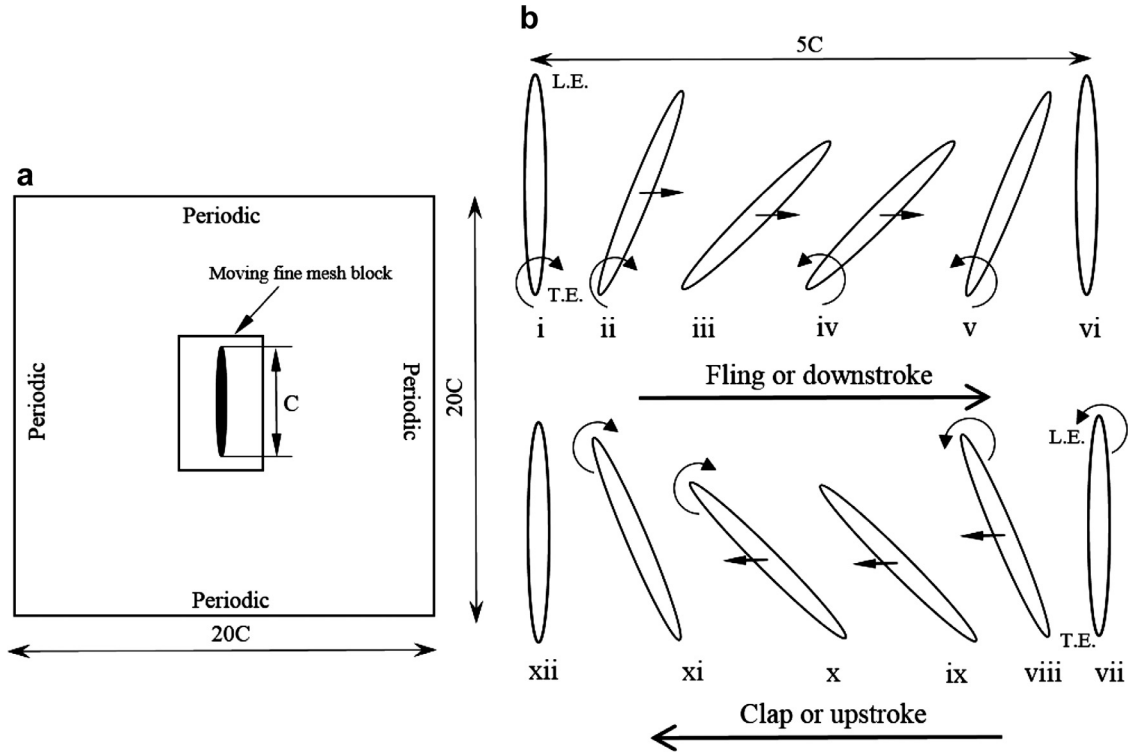

Fig. 8. (a) Layout of the computational domain with imposed boundary conditions. (b) Wing performing clap and fling motion for $\xi = 50\%$.

translating at its peak velocity, were registered with the use of SRT. During these time phases, the shift of nodes from fluid to solid region and vice versa is very extensive which gives rise to these fluctuations registered in the force calculation. A few earlier studies have also reported fluctuations in force calculations pertaining to the moving boundary using single block LBM [8]. However, these fluctuations were abated due to the implementation of multi-block as the resolution of wing improved and it better resembled the true shape as it moved.

Thus, the resolution of the uniform grid in SRT had to be improved in order to obtain a numerically stable solution. The L_2 norm when simulation results of discontinuous-grid-block MRT were compared with higher resolution SRT was found to be less than 5% indicating that the former grid was capable of obtaining accurate solutions. Moreover, the base grid and computation time for moving discontinuous-grid-block MRT remained the same for the range of Re considered. For Re = 10, the resolution of base grid was good enough for SRT to yield quantitatively similar results and took lower time than MRT. However, for Re = 1000 with SRT the resolution had to be increased by 16 times ($4\times X$ and $4\times Y$) to achieve a stable solution and resulted in an increase in the computation time by a factor of ~64. Thus, the moving discontinuous-grid-block MRT caters to the stability and resolution issues pertaining to high Re flows and imprints its authority over the conventional uniform grid SRT.

The stability of MRT over SRT can be attributed to various reasons. First, the moments of distribution function represent different physical quantities in MRT, thereby providing an added advantage of independently varying their relaxation time scales depending on their collision rate [19]. In the BGK model, all these quantities relax to their equilibrium values at the same rate. This characteristic of individual adjustment of relaxation times in MRT attributes to its higher numerical stability than BGK version [17,20] even at low relaxation times. Second, the deviatoric stress tensor constitutes of two components, shear viscosity ν (or kinematic viscosity) and the bulk viscosity ζ (or second viscosity coefficient). For D2Q9 model, these are related to the relaxation times as [12]

$$\nu = \frac{1}{3}\left(\frac{1}{s_8} - \frac{1}{2}\right) \tag{31}$$

$$\zeta = \frac{1}{6}\left(\frac{1}{s_2} - \frac{1}{2}\right) \tag{32}$$

The bulk viscosity remains a constant in the BGK version whereas in MRT it can be independently altered. Dellar [21] reported that a bulk viscosity of 10–100 times higher than the shear viscosity can significantly dissipate the spurious vortices generated in flow field especially at high Reynolds number. In present study, s_2 was taken as 1.2 that ensured $\zeta \geq 10\,\nu$ for all cases even at the highest Reynolds number (minimum ν). This feature of independently enhancing the bulk viscosity in MRT leads to stable results than with BGK version for high Re flows. Lastly, for a fixed length and velocity scale, a high Re is obtained at low values of kinematic viscosity which is calculated from Eq. (14). It is widely known that the SRT formulation tends to be unstable at very low values of relaxation time or when close to threshold value of 0.5 [7,8]. Therefore, for a stable solution at higher Re, a reduction in the grid size (i.e., increase in the node density) is mandatory. To conclude, for SRT to be applicable for analysis of flows at high Re (and the fact that MRT can handle such flows with ease), a higher resolution is required.

3.3. Rigid plunging flat plate

Shyy et al. [22–31] have used numerical simulations frequently to analyze the aerodynamics of rigid and flexible flapping wings. However, substantial interest remains in the further development of accurate and efficient numerical techniques. In this section, the application of the discontinuous grid-block to capture the fluid mechanics that emanates from a self-propelled heaving rigid flat plate in a quiescent fluid has been discussed. Recent experiments have shown that locomotion can be achieved in still air by a rigid wing if it is flapped above a critical Reynolds number (based on the flapping frequency)

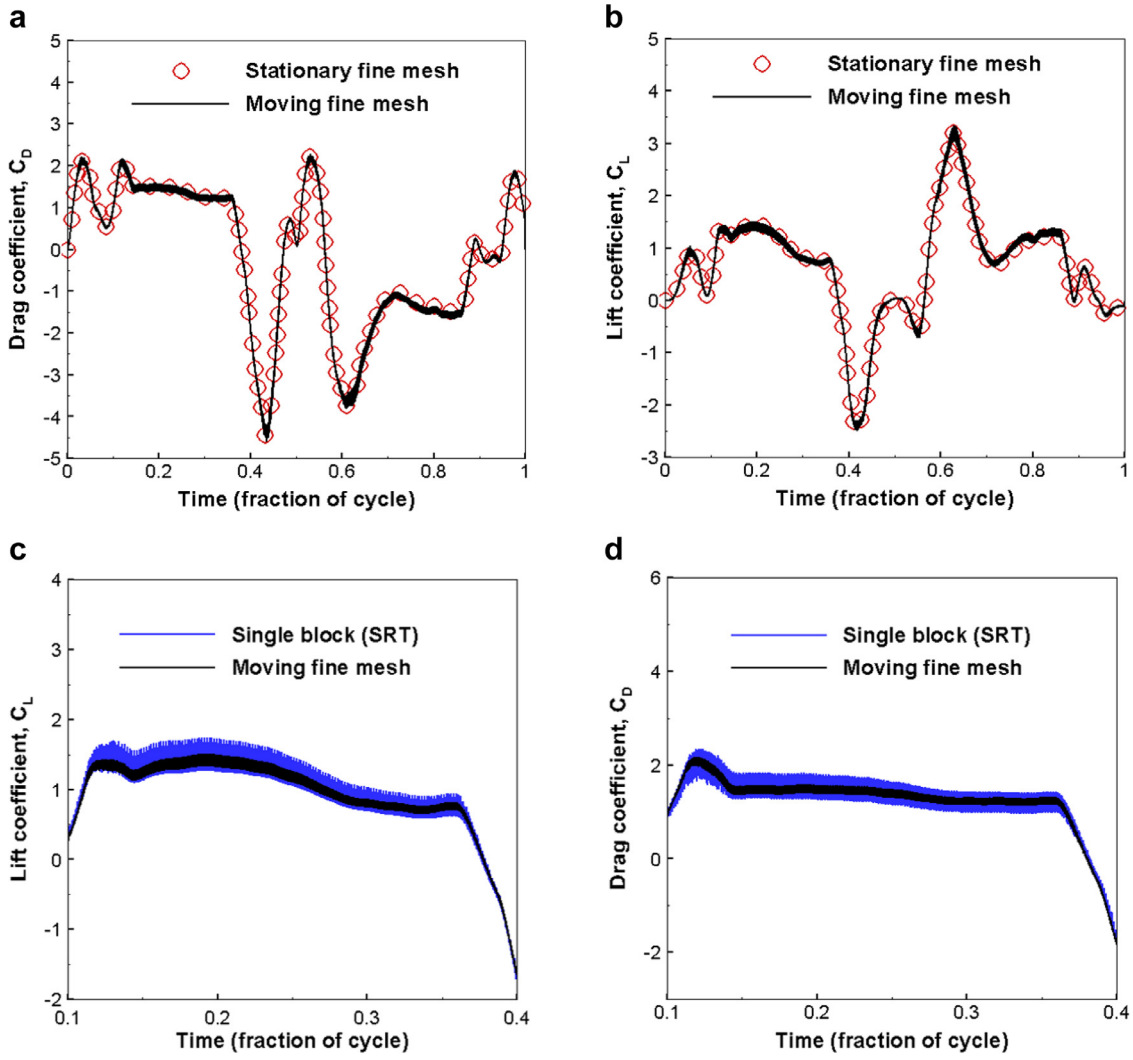

Fig. 9. A comparison of (a) Lift and (b) Drag coefficients obtained with stationary and moving discontinuous-grid-block ($m = 2$) MRT for a single wing clap and fling with $\xi = 50\%$ and Re = 128. Comparison of (c) Lift and (d) Drag coefficients obtained with a uniform grid SRT versus moving discontinuous-grid-block MRT during $t/T = 0.1$–0.4.

[32–34]. From these studies, it is clear that some of the key parameters that play a crucial role in the propulsive performance are the aspect ratio, flapping amplitude and density ratio.

Thus, in this problem a flat plate is considered which is unconstrained, or allowed to translate in the horizontal direction whereas plunging takes place in the vertical direction. The vertical displacement as a function of time is given by

$$y(t) = -A\cos(2\pi\gamma t) \tag{33}$$

where γ is the flapping frequency and A is the plunging amplitude.

The frequency based Reynolds number is defined as

$$\mathrm{Re}_f = \frac{A\gamma C}{\nu} \tag{34}$$

where ν is the kinematic viscosity of the fluid and C is the chord length.

Due to the fluid-structure interaction, a hydrodynamic force will be exerted on the flat plate. The horizontal component of this force will thus be responsible for the forward locomotion along the x-axis. According to Newton's second law, the horizontal position of the plate x_p is updated by the following equation

$$\bar{F}_x = m\frac{d^2x_p}{dt^2} \tag{35}$$

where m is the mass of the plate and $\bar{F}_x$ is calculated using momentum exchange method. The horizontal velocity u can be updated by the following rule

$$u = \frac{dx_p}{dt} \tag{36}$$

The propulsion of a plunging rigid flat plate with a uniform coarse mesh throughout was considered earlier [15]. An initial momentum corresponding to 0.01 times the maximum plunging velocity in the -X direction was imparted to the flat plate for the first half cycle. This small momentum did not affect the final velocity and ensured that on the instigation of instability wing moved towards the inlet (or towards left). A pictorial representation of the domain is shown in Fig. 10. A domain of size 50C × 20C was considered, where C was taken as $100\Delta x_c$. The thickness of wing was taken as 2% of the chord and it plunged with an amplitude A = 0.4C. The mass density ($\rho*$) of the wing was chosen to be 10 times that of the surrounding fluid. The leading edge of wing was positioned 20C downstream of the inlet and 30C upstream of the outlet. The wing plunged in a fine block of size $400\Delta x_f \times 200\Delta x_f$(or $200\Delta x_c \times 100\Delta x_c$). The dimensions of moving fine block were chosen such that the vertical displacement of the plate was confined in it and the block translated with the plate only along the horizontal direction. No slip boundary condition (zero fluid velocity) has been applied on the top and bottom walls of the do-

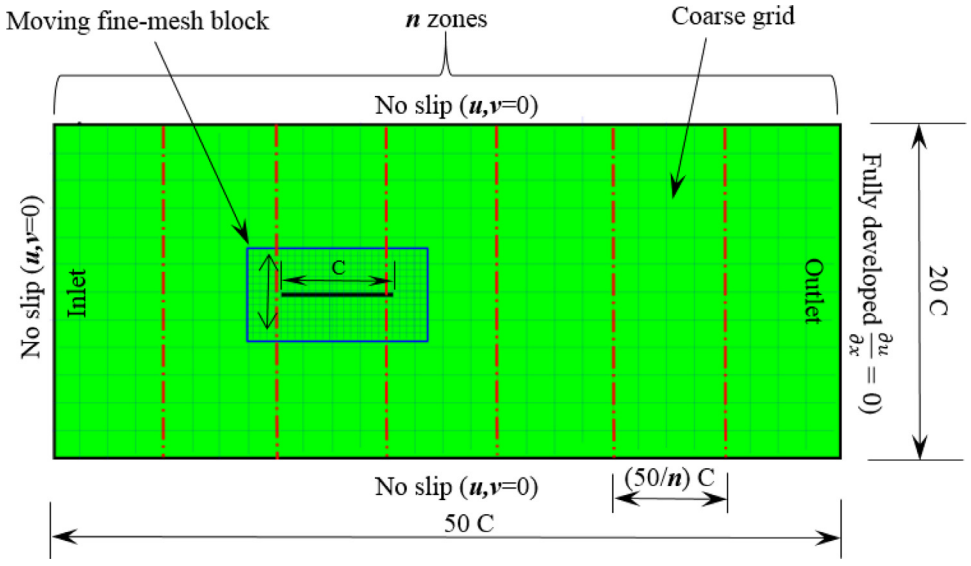

Fig. 10. Layout of the parallelized computational domain with imposed boundary conditions. Dotted lines demarcate the zones handled by n processors.

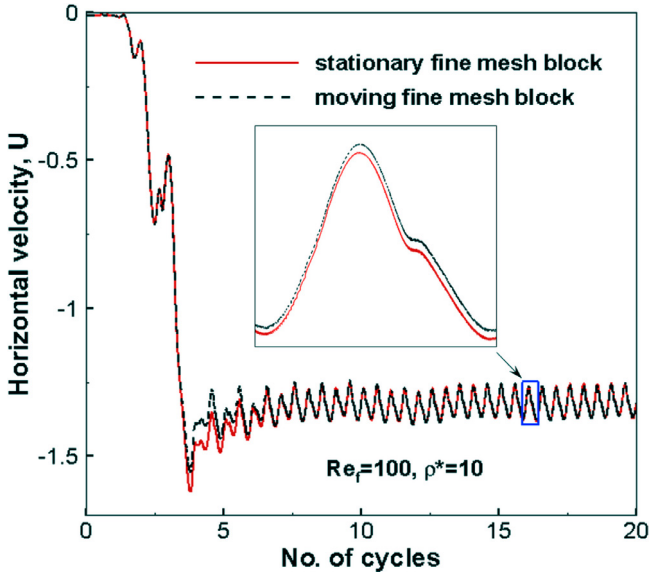

Fig. 11. Comparison of dimensionless horizontal velocity of the plunging airfoil at $Re_f = 100$ and $\rho^* = 10$ as a function of number of cycles with a stationary and moving discontinuous-grid-block MRT ($m = 2$).

main along with the inlet. At the outlet, stress-free or fully developed boundary condition (zero normal derivative of velocity) is imposed [35]. Similar to fine-block, the entire computational domain shifts, i.e., as the wing translates a unit distance in coarse block, a layer of fluid nodes is removed from outlet and added at the inlet [35].

To benefit from the use of LBM, the code developed was parallelized using MPI (Message Passing Interface) in which the domain along the x direction is sliced into $\boldsymbol{n}$ sub-zones, where $\mathbf{n}$ is the number of processors. Thus, each processor is assigned a sub-domain with $(50/\boldsymbol{n})$C x 20C physical size. On an average, an 8-core i7 processor was able to perform simulations for ~20 flapping cycles per day (which varied with flapping frequency or Re_f) for the large domain size considered in this work.

Fig. 11 shows a comparison in the profiles of dimensionless velocity U (non-dimensionalized by maximum plunging velocity $v_{max} = 2\pi\gamma A$) obtained with a stationary and moving discontinuous-grid-block for $Re_f = 100$, respectively. For the stationary multi-block, a fine mesh that spanned the entire horizontal length of domain (50C) was imposed whereas the vertical dimension was same as that in the moving fine-block (i.e. $10000\Delta x_f \times 200\Delta x_f$). As visible, instability is achieved within one plunging cycle and the wing begins to propel forward. Subsequently, the velocity magnitude increases in the -X-direction until it reaches a quasi-steady state at around 8th cycle. A minor difference is observed in velocity profiles from $t/T = 4$–6. This could be due to the reason that the wing which was moving at a very high acceleration and had attained a maximum cruising velocity (prior to $t/T<4$), suddenly experienced a sharp deceleration. Thus, the fluid velocity and pressure changed abruptly at the leading/trailing edge. As the coarse/fine block interface involves interpolations, obtaining near-exact solution depends on the accuracy of the interpolation function and its distance from the solid. In the case of a stationary multi-block, the coarse/fine block interfaces are very far away from the solid and do not exhibit any effect of this behavior. The same is not the case with the moving multi-block method. However, a good agreement in the magnitude of final velocity for these two cases was observed which indicates the efficacy of the moving mesh.

Flow structures (vorticity) around the flat plate at different time instants are plotted in the insets of Fig. 12 at (i) $t/T = 0.5$, (ii) $t/T = 2.5$ and (iii) $t/T = 18.5$ (where t is the actual time and T is the time period of one cycle) for the moving discontinuous-grid-block case. In early stages (i), when the magnitude of horizontal velocity (initial perturbation) is extremely small, the vortical flow structures around the airfoil appear nearly symmetrical. Here the wing is performing the up-stroke and hence vortices thus generated are attached to the bottom surface. However, the movement of the wing due to initially imparted momentum marks the onset of instability. At (ii), the wing is shown interacting with the previously shed vortices which induces asymmetry in the flow (unequal vortices on the right and left edge) resulting in thrust generation and initiation of propulsion. The vortex attached to the right edge is seen interacting with the oppositely directed vortex shed during the previous downstroke from the opposite edge. Consequently, due to the formation of a vortex dipole, there is annihilation of vortices on the right edge. However, there is a large suction force on the left edge due to the attached LEV. This creates an imbalance in the force acting on the wing, and thus imparts the momentum to the wing essential for propulsion in the forward direction. At (iii), the wing has already attained a stable leftward velocity with symmetric wake patterns as two opposite directed vortices are shed per cycle, one during the upstroke and other during the downstroke. Both the half strokes are identical and mirror images of each other.

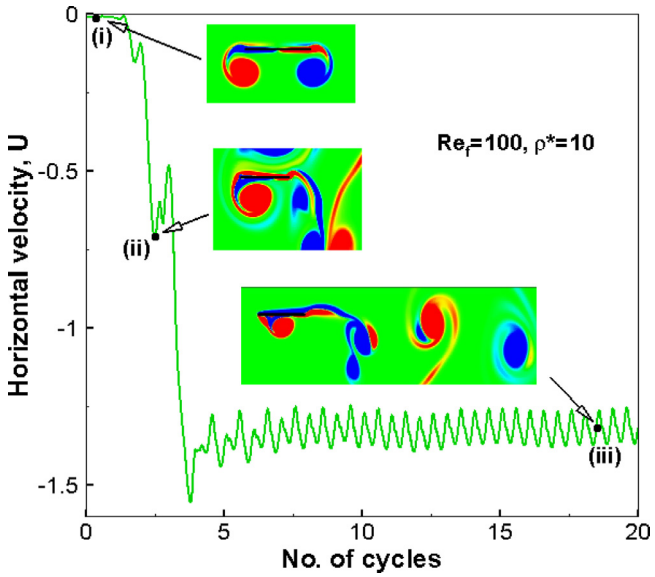

Fig. 12. Vorticity contours of fluid flow at (i) $t/T = 0.5$, (ii) $t/T = 2.5$ and (iii) $t/T = 18.5$ are plotted in the inset (the wing is performing upstroke and positioned at the maximum displacement). Red and blue colors indicate anti-clockwise (out of the plane of pater) and clockwise rotation (into the plane of paper), respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Just like the previous two cases, computation time consumed in the simulation using a stationary and moving discontinuous-grid-block is shown in Table 2. This again ascertains the effectiveness of moving fine-block to reduce the computation time and memory requirements while maintaining the accuracy as that of the stationary multi-block. Thus, this problem which involves large domain size in which the object keeps travelling indefinitely in one direction is ideally suited for the moving discontinuous-grid-block system. The three-dimensional extension of this method will be pursued in near future and is expected to be computationally more economical than the existing two-dimensional model when compared to their stationary counterparts.

4. Conclusions

The present work focused on the development and implementation of a multi-relaxation time lattice Boltzmann method using a shifting discontinuous-grid-block around a moving solid body with finite thickness. The intricacies in building a framework for conducting simulations with a two-dimensional non-stationary discontinuous-grid-block which moves along with the object as it traverses in fluid were presented. The movement of the fine block was controlled by measuring the distance propagated by the moving solid object after every time step in the coarse block. If this distance exceeded a unit lattice in coarse block, the fine block shifted by the same distance in coarse block. This movement resulted in the creation (deletion) of new (old) layers and required transfer of distributions from coarse to fine block and vice-versa. In this regard, various combinations possible to estimate the unknown distributions on the grid points of added layer were devised. Four methods for spatial interpolation of distributions at the intermediate nodes in a discontinuous-grid-block using linear, quadratic, cubic Lagrangian and cubic spline were compared. The results with cubic spline interpolation were found to be the closest to the solution obtained using finer grids throughout the domain.

This method was tested on a cylinder placed off-center in a shear flow, a rigid wing performing 'clap and fling' motion, and a rigid plunging flat plate which propels forward. It was observed that due to the movement of the solid objects, their represented shape varied as the number of nodes or grid points inside the solid underwent transitions from non-fluid region to fluid region or vice versa. This gave rise to fluctuations in the force calculated using the momentum exchange method. An advantage of grid refinement was that it was successful in curtailing the fluctuations recorded so often in a coarse grid simulation. Also, the moving discontinuous-grid-block MRT overcame resolution and stability issues encountered in SRT at higher Re. With discontinuous-grid-block MRT, the same base grid at $\mathrm{Re} = 10$ was sufficient enough to handle flows even at $\mathrm{Re} = 1000$. However, in the latter case, the resolution had to be increased by 16 times from the base domain to give stable and quantitatively comparable results with that of former. Consequently, the computation time went up by almost 64 times which proves the dominance of this MRT-based method over SRT. In addition, the computation time and memory consumed for performing these simulations were compared with that of a stationary discontinuous-grid-block MRT for all three cases. Except for the clap and fling case where wing translated back-and-forth, there was a sharp drop in the computation time with the implementation of moving fine-block. This suggested that this method is highly suitable for flows in which the moving object keeps traversing indefinitely in one direction and where use of a stationary fine block may potentially lead to an increased computation cost, thereby rendering such simulations difficult and inefficient. Future work is directed towards pursuing the three-dimensional extension of this model.

Acknowledgment

The authors acknowledge the financial support received from the Aeronautics Research and Development Board (ARDB) under SIGMA project No. 1705.

Appendix

With reference to Fig. 13, the distributions at intermediate nodes were obtained using the interpolation formulae given in Table 3.

For the corner nodes, 4 pt. schemes might not be applicable (as i+3 or i-3 node doesn't exist). Thus, $\tilde{f}_i$ is interpolated using the information at nodes i-5 or i+5.

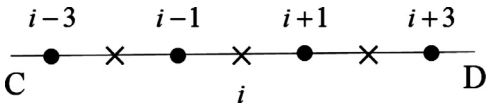

Fig. 13. Fine block boundary CD (with reference to Fig. 2) where the fine nodes are represented by × and underlying coarse nodes by •.

Table 3
Interpolation formulae for Lagrangian p point method.

Scheme	Formula
2 pt.	$\tilde{f}_i = \frac{1}{2}(\tilde{f}_{i-1} + \tilde{f}_{i+1})$
3 pt.	$\tilde{f}_i = (\frac{-1}{2})(\frac{1}{4})\tilde{f}_{i-3} + (\frac{3}{2})(\frac{1}{2})\tilde{f}_{i-1} + (\frac{3}{4})(\frac{1}{2})\tilde{f}_{i+1}$ or $\tilde{f}_i = (\frac{1}{2})(\frac{3}{4})\tilde{f}_{i-1} + (\frac{1}{2})(\frac{3}{2})\tilde{f}_{i+1} + (\frac{1}{4})(\frac{-1}{2})\tilde{f}_{i+3}$
4 pt.	$\tilde{f}_i = (\frac{-1}{2})(\frac{1}{4})(\frac{1}{2})\tilde{f}_{i-3} + (\frac{3}{2})(\frac{1}{2})(\frac{3}{4})\tilde{f}_{i-1} + (\frac{3}{4})(\frac{1}{2})(\frac{3}{2})\tilde{f}_{i+1} + (\frac{1}{2})(\frac{1}{4})(\frac{-1}{2})\tilde{f}_{i+3}$ for corner nodes, $\tilde{f}_i = (\frac{1}{2})(\frac{3}{4})(\frac{5}{6})\tilde{f}_{i-1} + (\frac{1}{2})(\frac{3}{2})(\frac{5}{4})\tilde{f}_{i+1} + (\frac{1}{4})(\frac{-1}{2})(\frac{5}{2})\tilde{f}_{i+3} + (\frac{1}{6})(\frac{-1}{4})(\frac{-3}{2})\tilde{f}_{i+5}$

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compfluid.2015.11.003.

References

[1] Shyy W, Udaykumar HS, Rao MM, Smith RW. Computational fluid dynamics with moving boundaries. New York: Dover Publications; 1996.
[2] Succi S. The lattice boltzmann equation for fluid dynamics and beyond. New York: Oxford University Press; 2001.
[3] Chen S, Doolen GD. Lattice Boltzmann method for fluid flows. Annu Rev Fluid Mech 1998;30:329–64.
[4] Qi D. Lattice-Boltzmann simulations of particles in non-zero-reynolds-number flows. J Fluid Mech 1999;385:41–62.
[5] He X, Luo L-S. Theory of the Lattice Boltzmann method: from the boltzmann equation to the lattice boltzmann equation. Phys Rev E 1997;56(6):6811–17.
[6] Ladd AJC. Numerical simulations of particulate suspensions via a discretized boltzmann equation. Part 1. Theoretical Foundation. J Fluid Mech 1994;271:285–309.
[7] McCracken ME, Abraham J. Multiple-relaxation-time lattice-Boltzmann model for multiphase flow. Phys Rev E 2005;71:036701.
[8] Lallemand P, Luo L-S. Lattice Boltzmann method for moving boundaries. J Comput Phys 2003;184:406–21.
[9] Feng ZG, Michaelides EE. The immersed boundary-lattice boltzmann method for solving fluid-particles interaction problems. J Comput Phys 2004;195:602–28.
[10] 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.
[11] Yu D, Mei R, Luo L, Shyy W. Viscous flow computations with the method of lattice Boltzmann equation. Progr Aerosp Sci 2003;39:329–67.
[12] 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.
[13] Inamuro T. Lattice Boltzmann methods for moving boundary flows. Fluid Dyn Res 2012;44:024001.
[14] Arora N, Gupta A, Sanghi S, Aono H, Shyy W. Lift-drag and flow structures associated with the 'clap and fling' motion. Phys Fluids 2014;26:071906.
[15] Arora N, Jain A, Singh A, Gupta A, Sanghi S, Aono H, Shyy W. Forward propulsion of a rigid plunging airfoil. In: 32nd AIAA Applied Aerodynamics Conference, Atlanta, Georgia, USA; June 2014.
[16] Pan C, Luo L-S, Miller CT. An evaluation of lattice Boltzmann schemes for porous medium flow simulation. Comput Fluids 2006;35:898–909.
[17] Lallemand P, Luo L-S. Theory of the lattice Boltzmann method: acoustic and thermal properties in two and three dimensions. Phys Rev E 2003;68:036706.
[18] Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., Numerical recipes in fortran 77: the art of scientific computing. 2nd Ed. New York: Cambridge University Press; 2003.
[19] Hao L, Cheng P. Lattice Boltzmann simulations of anisotropic permeabilities in carbon paper gas diffusion layers. J Power Sources 2009;186:104–14.
[20] D'Humières D, Ginzburg I, Krafczyk M, Lallemand P, Luo L-S. Multiple–relaxation–time lattice Boltzmann models in three dimensions. Philos Trans R Soc Lond A: Math, Phys Eng Sci 2002;360(1792):437–51.
[21] Dellar PJ. Bulk and shear viscosities in lattice Boltzmann equations. Phys Rev E 2001;64:031203.
[22] Qi D, Liu Y, Shyy W, Aono H. Simulations of dynamics of plunge and pitch of a three-dimensional flexible wing in a low Reynolds number. Phys Fluids 2010;22:091901.
[23] Shyy W, Aono H, Kang C-K, Liu H. An introduction to flapping wing aerodynamics. New York: Cambridge University Press; 2013.
[24] Shyy W, Trizila P, Kang C, Aono H. Can tip vortices enhance lift of a flapping wing. AIAA J 2009;47:289–93.
[25] Trizila P, Kang C-K, Aono H, Visbal M, Shyy W. Low reynolds number aerodynamics of a flapping rigid flat plate. AIAA J 2011;49:806–23.
[26] Kang CK, Aono H, Cesnik CES, Shyy W. Effects of flexibility on the aerodynamic performance of flapping wings. J Fluid Mech 2011;689:32–74.
[27] Kang C-K, Aono H, Baik YS, Bernal LP, Ol M, Shyy W. Fluid dynamics of pitching and plunging flat plate at intermediate reynolds numbers. AIAA J 2013;51:315–29.
[28] Kang C, Shyy W. Scaling law and enhancement of lift generation of an insect-size hovering flexible wing. J R Soc Interface 2013;10:0361.
[29] Vandenheede, Ruben BR, Bernal LP, Morrison CL, Gogulapati A, Friedmann PP, Kang C-K, Shyy W. Experimental and computational study on flapping wings with bio-inspired hover kinematics. AIAA J 2014;52:1047–58.
[30] Kang C, Shyy W. Analytical model for instantaneous lift and shape deformation of an insect-scale flapping wing in hover. J R Soc Interface 2014;11:0933.
[31] Arora N, Gupta A, Aono H, Shyy W. Propulsion of a plunging flexible airfoil using a torsion spring model. In: 33rd AIAA Applied Aerodynamics Conference, Dallas, Texas, USA; June 2015.
[32] Alben S, Shelley M. In: Coherent locomotion as an attracting state for a free flapping body, 102; 2005. p. 11163–6.
[33] Vandenberghe N, Zhang J, Childress S. Symmetry breaking leads to forward flapping flight. J Fluid Mech 2004;506:147–55.
[34] Vandenberghe N, Childress S, Zhang J. On unidirectional flight of a free flapping wing. Phys Fluids 2006;18:014102.
[35] Aidun CK, Lu Y, Ding E-J. Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation. J Fluid Mech 1998;373:287–311.