International Journal of Heat and Mass Transfer 126 (2018) 1059–1070

Contents lists available at ScienceDirect

International Journal of Heat and Mass Transfer

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

An immersed boundary-discrete unified gas kinetic scheme for simulating natural convection involving curved surfaces

Chao Li [a], Lian-Ping Wang [b,c,*]

[a] Marine Engineering College, Dalian Maritime University, Dalian 116026, Liaoning, China
[b] Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA
[c] Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen 518055, Guangdong, China

ARTICLE INFO

Article history:
Received 7 January 2018
Received in revised form 31 March 2018
Accepted 28 April 2018
Available online 29 May 2018

Keywords:
Natural convection
IB-DUGKS
Strang-splitting
Curved boundary

ABSTRACT

In this work, an immersed boundary-discrete unified gas kinetic scheme (IB-DUGKS) is proposed and presented for the simulation of natural convection with a curved body surface. In this method, two distribution functions are employed for velocity and temperature field, respectively, and they are coupled under the Boussinesq approximation. The IB-DUGKS provides an effective way for the DUGKS to treat a curved boundary. The Strang-splitting method is used to handle the IB force, and its accuracy is first validated by comparing with another implementation method for the base case of natural convection in a square cavity. The widely used direct-forcing immersed boundary method is adopted due to its simplicity, with an iteration procedure to ensure the accuracy of no-slip condition on the immersed boundary. Natural convection between an outer square and an inner circular cylinder is then simulated under different geometric configurations, including different aspect ratios and locations of the cylinder relative to the cavity. The numerical results are in excellent agreement with the results from the literature, confirming the accuracy and robustness of the proposed method.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Natural convection between a body and an enclosure has received a great deal of attention in the past decades, as it is relevant to many industrial applications such as heat exchangers, cooling of electronic equipment, and thermal storage systems [1]. In this paper, the specific geometric configuration of interest is a cold outer square enclosure and a hot inner circular cylinder. The natural convection problem of this geometric configuration has also been investigated by other people in recent years, such as Moukalled and Acharya [2] and Shu and Zhu [3], and has often served as a benchmark case to verify new numerical methods [4–6]. In the present study, an immersed boundary-discrete unified gas kinetic scheme (IB-DUGKS) is developed to investigate such a natural convection problem involving a curved surface.

IB-DUGKS is a kinetic method solving a model Boltzmann equation. Unlike the traditional CFD methods which are based on solving the Navier–Stokes equations, the kinetic methods are based on the kinetic theory. The kinetic methods provide a connection between the macroscopic hydrodynamics and the microscopic physics, and are sometime referred to as mesoscopic methods. Among the different kinds of kinetic methods, the gas kinetic scheme (GKS) [7] and lattice Boltzmann method (LBM) [8] are widely used and have been developed rapidly in recent years. Based on GKS, a unified GKS (UGKS) for all Knudsen number flows was developed by Xu and Huang [9]. And recently, the DUGKS was developed by Guo et al. which combines the advantages of UGKS and LBM [10,11]. It is a finite volume method and derived directly from the Boltzmann equation. Compared with LBM, the DUGKS is more flexible in application, such as fully decoupled time and space steps, and also a non-uniform mesh can be easily employed.

Since DUGKS is relatively new, only a few studies have emerged to explore the potential applications of DUGKS. Wang et al. [12] proposed a coupled DUGKS for Boussinesq flows and the Rayleigh-Bénard convection and natural convection in a square cavity were investigated. Wu et al. [13] proposed a general method to allow the DUGKS to handle an external force term by adding the force term into the Boltzmann equation and DUGKS procedure. Zhu et al. [14] successfully extended the DUGKS to unstructured meshes. Guo and Xu [15] extended the DUGKS to simulate the whole multiscale heat transfer process based on the phonon Boltzmann transport equation. Bo et al. [16] investigated 3D Taylor–Green vortex flow and turbulent channel flow using DUGKS. Zhu et al. [17] developed an open source OpenFOAM solver for the

[*] Corresponding author at: Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA.

E-mail addresses: lichao2013@dlmu.edu.cn (C. Li), lwang@udel.edu (L.-P. Wang).

https://doi.org/10.1016/j.ijheatmasstransfer.2018.04.166
0017-9310/© 2018 Elsevier Ltd. All rights reserved.

Boltzmann model equation with DUGKS. Wang et al. [18] conducted a systematic numerical study of three-dimensional natural convection in a differentially heated cubical cavity with Rayleigh number up to 10^{10}. Recently, the IB-DUGKS has been developed for isothermal flows with curved boundary by Tao et al. [19]. So far, DUGKS has not been applied to heat transfer problems with a curved boundary.

To incorporate curved boundaries, we shall consider the immersed boundary method (IBM) which was first proposed by Peskin in the early 1970s [20]. Due to its simplicity in implementation and flexibility in application, it has drawn particular attention in the recent decades [21–23]. The main idea of IBM is to use two set of grids for the simulation, with a fixed Eulerian grid covering the whole domain for the fluid, the Lagrangian points representing the immersed boundary. The interaction between the fluid and the immersed boundary is handled through the IB force. IBM was first used to simulate elastic material boundary, and the IB force on the boundary points can be determined by the deformation under Hooke's law [20,21]. When the force is distributed to the fluid through a smooth delta function, the effect of the real boundary can be approximated by the IB force of the immersed boundary. The original method to calculate the IB force can be called the penalty or feedback forcing method. Another popular way to decide the IB force is the direct forcing method proposed by Mohd-Yusof [22]. It is simpler to implement. But the original direct forcing method cannot ensure the no-slip condition on the boundary due to the delta function interpolation errors causing the streamlines to penetrate through the immersed boundary. To avoid this problem, the multi-direct forcing method and the implicit direct forcing method were developed. The multi-direct forcing method was first used by Luo et al. [24], and the details will be described in Section 2.2. The implicit direct forcing method was first proposed by Wu et al. [25]. In this implicit method, one does not calculate the IB force. Instead, the velocity corrections at all boundary points are considered as unknowns which are computed in such a way that the non-slip boundary condition at the boundary points is enforced. The drawback of the implicit direct forcing method is that one need to solve a matrix system, but the no-slip boundary condition can be satisfied precisely. Besides these two main methods (feedback forcing and direct forcing methods), there are many other ways to implement the IBM. One of them is the interpolation-based scheme proposed by Kim et al. [26,27], which is based on a finite volume approach on a staggered mesh together with a fractional-step method. The momentum forcing and the mass source/sink are applied on the body surface or inside the body to satisfy the no-slip boundary condition on the immersed boundary and the continuity for the cell containing the immersed boundary, respectively. The heat source/sink is introduced on the body surface or inside the body to satisfy the isothermal or *iso*-heat-flux condition on the immersed boundary. A second-order linear or bilinear interpolation scheme is used to satisfy the no-slip velocity on the immersed boundary, which is numerically stable regardless of the relative position between the grid and the immersed boundary. Kim et al. had validated their method with isothermal flow and heat transfer problems [26,27], which also showed the capability of their method. One can find other versions of the IBM from the literature or from Refs. [28–30].

Within the conventional CFD which solves the Navier-Stokes equations directly, IBM is well established for isothermal problems. A few non-isothermal studies using the IBM are noted here. Kim et al. [4] investigated natural convection between a cold outer square and a hot inner circular cylinder with the interpolation-based IBM. Jiong et al. [6] investigated natural convection in a square enclosure with feedback forcing IBM. Wang et al. [31] investigated natural and forced convection problems with the direct forcing IBM. These and other studies [32–35] reveal that IBM is a competent method for solving a thermal flow within conventional CFD.

The aim of present work is to combine DUGKS (a mesoscopic flow solver) and IBM (a curved boundary treatment) in order to formulate a mesoscopic simulation tool for natural convection problems with complex geometries. The rest of this paper is organized as follows. In Section 2, a brief introduction of DUGKS and IBM, as well as how to couple the two methods are described. In Section 3, the accuracy of the present method is validated by comparing the simulation results for several benchmark problems with the data from the literature. Finally, a brief summary and conclusions are presented in Section 4.

2. Simulation method

In this section, the DUGKS algorithm is described first. Then the direct-forcing immersed boundary method is introduced. At last two different ways for the DUGKS to incorporate an external force term are given.

2.1. Discrete unified gas kinetic scheme

DUGKS was first proposed by Guo et al. [10], one can also find the details about this method from the previous studies [12,13]. Here a brief introduction of the method is given.

2.1.1. DUGKS for velocity field

DUGKS begins with the Boltzmann equation with the BGK collision model [10]

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla f = \Omega \equiv \frac{f^{eq} - f}{\tau_v}, \tag{1}$$

where f is the distribution function for the velocity field, $f = f(x, \xi, t)$ with space $\boldsymbol{x}$, time t and velocity ξ. Ω is the collision term, τ_v is the relaxation time and related to the viscosity coefficient. f^{eq} is the Maxwellian equilibrium state and has the following form:

$$f^{eq} = \frac{\rho}{(2\pi RT_1)^{D/2}} \exp\left(-\frac{(\xi - \mathbf{u})^2}{2RT_1}\right), \tag{2}$$

where ρ is density of the fluid, R is the gas constant, T_1 is a constant temperature, $\mathbf{u}$ is the macroscopic velocity of the fluid, D is the spatial dimension. Here $RT_1 = c_s^2$, c_s is the artificial sound speed. The hydrodynamic variables can be obtained as:

$$\rho = \int f d\xi, \quad \rho\mathbf{u} = \int \xi f d\xi. \tag{3}$$

The DUGKS is a finite volume method, and the flow domain can be divided into a set of control volumes V_j which are centered at $\boldsymbol{x_j}$. Integrating Eq. (1) on V_j from time t_n to t_{n+1}, and using the midpoint rule for the integration of the convection term and trapezoidal rule for the collision term, one can obtain

$$f_j^{n+1} - f_j^n + \frac{\Delta t}{|V_j|} F^{n+1/2} = \frac{\Delta t}{2}(\Omega_j^{n+1} + \Omega_j^n), \tag{4}$$

where Δt is the time step, and

$$F^{n+1/2} = \int_{\partial V_j} (\xi \cdot \mathbf{n}) f(x, t_{n+1/2}) dS \tag{5}$$

is the microflux across the interface, $\boldsymbol{n}$ is the unit vector normal to the cell interface. But Eq. (4) used to update the distribution function f is implicit, so two new distribution functions are defined

$$\tilde{f}_j = f_j - \frac{\Delta t}{2}\Omega_j, \quad \tilde{f}_j^+ = f_j + \frac{\Delta t}{2}\Omega_j. \tag{6}$$

So one can rewritten Eq. (4) using an explicit form as

$$\tilde{f}_j^{n+1} = \tilde{f}_j^{+,n} - \frac{\Delta t}{|V_j|} F^{n+1/2}. \tag{7}$$

In order to update $\tilde{f}$, we need to evaluate the flux $F^{n+1/2}$. Integrating Eq. (1) along the characteristic line within a half time step h (=$\Delta t/2$) yields

$$\begin{aligned} & f(\boldsymbol{x}_b, \xi, t_n + h) - f(\boldsymbol{x}_b - \xi h, \xi, t_n) \\ & = \frac{h}{2}[\Omega(\boldsymbol{x}_b, \xi, t_n + h) + \Omega(\boldsymbol{x}_b - \xi h, \xi, t_n)], \end{aligned} \tag{8}$$

where x_b is the end point located at the cell interface, and the trapezoidal rule is used to treat the collision term. Similarly, two new distribution functions are defined

$$\bar{f} = f - \frac{h}{2}\Omega, \quad \bar{f}^+ = f + \frac{h}{2}\Omega. \tag{9}$$

So Eq. (8) can be rewritten as

$$\bar{f}(x_b, \xi, t_n + h) = \bar{f}^+(x_b - \xi h, \xi, t_n). \tag{10}$$

If we apply Taylor expansion to the right hand side of Eq. (10), and then Eq. (10) becomes

$$\bar{f}(x_b, \xi, t_n + h) = \bar{f}^+(x_b, \xi, t_n) - h\xi \cdot \sigma_b, \tag{11}$$

where $\sigma_b = \nabla \bar{f}^+(x_b, \xi, t_n)$. The density and velocity at the cell interface can be obtained as

$$\rho = \int \bar{f} d\xi, \quad \rho\mathbf{u} = \int \xi \bar{f} d\xi, \tag{12}$$

from which the equilibrium state function f^{eq} at the cell interface can be calculated. And then the original distribution function at the cell interface, which are used to calculate the microflux in Eq. (5), can be calculated by

$$f(x_b, \xi, t_n + h) = \frac{2\tau_v}{2\tau_v + h}\bar{f}(x_b, \xi, t_n + h) + \frac{h}{2\tau_v + h} f^{eq}(x_b, \xi, t_n + h). \tag{13}$$

In the implementation of DUGKS, another two relations are used

$$\bar{f}^+ = \frac{2\tau_v - h}{2\tau_v + \Delta t}\tilde{f} + \frac{3h}{2\tau_v + \Delta t} f^{eq}, \tag{14}$$

$$\tilde{f}^+ = \frac{4}{3}\bar{f}^+ - \frac{1}{3}\tilde{f}. \tag{15}$$

Now all the equations to update the distribution function $\tilde{f}$ have been obtained. And the macroscopic density and velocity of the fluid can be obtained as

$$\rho = \int \tilde{f} d\xi, \quad \rho\mathbf{u} = \int \xi \tilde{f} d\xi. \tag{16}$$

For a low March number flow, the Maxwellian equilibrium distribution function can be approximated by its Taylor expansion

$$f^{eq} = W_i \rho \left[1 + \frac{\xi \cdot \mathbf{u}}{RT_1} + \frac{(\xi \cdot \mathbf{u})^2}{2(RT_1)^2} - \frac{|\mathbf{u}|^2}{2RT_1}\right], \tag{17}$$

where W_i is the weight coefficients, W_0 = 4/9, $W_{1,2,3,4}$ = 1/9, $W_{5,6,7,8}$ = 1/36. For 2D flow problems, the nine velocity model are used

$$\xi_i = \left\{ \begin{array}{c} (0,0), \ i = 0 \\ (\cos[(i-1)\pi/2], \sin[(i-1)\pi/2])c, \ i = 1-4 \\ (\cos[(2i-9)\pi/4], \sin[(2i-9)\pi/4])\sqrt{2}c, \ i = 5-8 \end{array} \right. , \tag{18}$$

where $c = \sqrt{3RT_1}$. The fluid pressure and kinetic viscosity also can be obtained

$$p = \rho RT_1, \quad \nu = \tau_\nu RT_1. \tag{19}$$

2.1.2. DUGKS for temperature field

The procedure for constructing the DUGKS model in temperature field is similar with the velocity [12]. First we have the Boltzmann equation with the BGK collision model

$$\frac{\partial g}{\partial t} + \xi \cdot \nabla g = \Psi \equiv \frac{g^{eq} - g}{\tau_c}, \tag{20}$$

where g is the distribution function for the temperature field, $g = g(x, \xi, t)$ with space $\boldsymbol{x}$, time t and velocity ξ. Ψ is the collision term, τ_c is the relaxation time and related to the thermal diffusivity coefficient. g^{eq} is the Maxwellian equilibrium state and has the following form:

$$g^{eq} = \frac{T}{(2\pi RT_2)^{D/2}} \exp\left(-\frac{(\xi - \mathbf{u})^2}{2RT_2}\right), \tag{21}$$

where T is temperature of the fluid, T_2 is a constant temperature related to the artificial sound speed, just like T_1 in velocity field. The temperature can be obtained as:

$$T = \int g d\xi. \tag{22}$$

Similarly with Eq. (4), from Eq. (20) one can obtain the following equation

$$g_j^{n+1} - g_j^n + \frac{\Delta t}{|V_j|} F_T^{n+1/2} = \frac{\Delta t}{2}(\Psi_j^{n+1} + \Psi_j^n), \tag{23}$$

By introducing two new distribution functions

$$\tilde{g}_j = g_j - \frac{\Delta t}{2}\Psi_j, \quad \tilde{g}_j^+ = g_j + \frac{\Delta t}{2}\Psi_j, \tag{24}$$

Eq. (23) can be rewritten as

$$\tilde{g}_j^{n+1} = \tilde{g}_j^{+,n} - \frac{\Delta t}{|V_j|} F_T^{n+1/2}. \tag{25}$$

In order to update $\tilde{g}$, one need to evaluate the flux $F_T^{n+1/2}$. Integrating Eq. (20) along the characteristic line within a half time step h yields

$$\begin{aligned} & g(\boldsymbol{x}_b, \xi, t_n + h) - g(\boldsymbol{x}_b - \xi h, \xi, t_n) \\ & = \frac{h}{2}[\Psi(\boldsymbol{x}_b, \xi, t_n + h) + \Psi(\boldsymbol{x}_b - \xi h, \xi, t_n)]. \end{aligned} \tag{26}$$

Two new distribution functions are defined

$$\bar{g} = g - \frac{h}{2}\Psi, \quad \bar{g}^+ = g + \frac{h}{2}\Psi, \tag{27}$$

so Eq. (26) can be rewritten as

$$\bar{g}(x_b, \xi, t_n + h) = \bar{g}^+(x_b - \xi h, \xi, t_n) = \bar{g}^+(x_b, \xi, t_n) - h\xi \cdot \bar{\sigma}_b, \tag{28}$$

where $\sigma_b = \nabla \bar{g}^+(x_b, \xi, t_n)$ and the Taylor expansion has been applied. The temperature at the cell interface can be obtained as

$$T = \int \bar{g} d\xi, \tag{29}$$

from which the equilibrium state function g^{eq} at the cell interface can be calculated. And then the original distribution function at the cell interface can be calculated by

$$g(x_b, \xi, t_n + h) = \frac{2\tau_c}{2\tau_c + h}\bar{g}(x_b, \xi, t_n + h) + \frac{h}{2\tau_c + h} g^{eq}(x_b, \xi, t_n + h). \tag{30}$$

In the implementation of DUGKS, another two relations are used

$$\bar{g}^{+} = \frac{2\tau_c - h}{2\tau_c + \Delta t}\tilde{g} + \frac{3h}{2\tau_c + \Delta t}g^{eq}, \tag{31}$$

$$\tilde{g}^{+} = \frac{4}{3}\bar{g}^{+} - \frac{1}{3}\tilde{g}. \tag{32}$$

Now all the equations to update the distribution function $\tilde{g}$ have been obtained. And the temperature of the fluid can be obtained as

$$T = \int \tilde{g} d\xi. \tag{33}$$

The nine velocity model is also used in temperature field, and the equilibrium distribution function can be written as

$$g^{eq} = W_i T\left[1 + \frac{\xi \cdot \mathbf{u}}{RT_2} + \frac{(\xi \cdot \mathbf{u})^2}{2(RT_2)^2} - \frac{|\mathbf{u}|^2}{2RT_2}\right]. \tag{34}$$

The thermal diffusivity and Prandtl number can be obtained as

$$\kappa = \tau_c RT_2, \quad \Pr = \frac{\nu}{\kappa} = \frac{\tau_\nu RT_1}{\tau_c RT_2}. \tag{35}$$

2.1.3. The Boussinesq approximation for natural convection

THE Boussinesq approximation is often used when studying the natural convection problems. Under this assumption, the fluid density ρ is considered as a linear function of the temperature T [36]:

$$\rho = \rho_o - \rho_o\beta(T - T_o), \tag{36}$$

where ρ_o and T_o are the average density and temperature of the fluid, β is the coefficient of thermal expansion. The gravity can be written as:

$$G = \rho g_o = \rho_o g_o - \rho_o g_o \beta(T - T_o), \tag{37}$$

where g_o is the acceleration of gravity. And the Boussinesq equations can be obtained [36,12]:

$$\nabla \cdot \mathbf{u} = 0, \tag{38}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} - \mathbf{a}, \tag{39}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = k \nabla^2 T, \tag{40}$$

where **a** is the acceleration of external force (buoyancy), and here

$$\mathbf{a} = g_o \beta(T - T_o)\mathbf{j}, \tag{41}$$

j is the unit vector in vertical direction. The coupling of velocity and temperature in DUGKS is established by adding the force term in Eq. (1), which can be rewritten as [12]:

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla_x f + \mathbf{a} \cdot \nabla_\xi f = \Omega \equiv \frac{f^{eq} - f}{\tau_\nu}. \tag{42}$$

More details about the DUGKS with an external force will be given in Section 2.3.

2.1.4. Boundary condition treatment

In the previous studies [10–13], the well-known bounce back rule has been used to treat the boundary in DUGKS. The boundary wall is located at the cell interface, so instead of treating the original distribution function f and g, one should implement the boundary treatment with $\bar{f}$ and $\bar{g}$. For the fixed wall in the present study, the distribution function $\bar{f}$ pointing towards the flow field can be obtained as

$$\bar{f}(x_w, \xi_i, t + h) = \bar{f}(x_w, -\xi_i, t + h). \tag{43}$$

And for the temperature field, two types of boundary should be considered [12]. For the constant temperature boundary, the distribution function $\bar{g}$ pointing towards the flow field can be obtained as

$$\bar{g}(x_w, \xi_i, t + h) = -\bar{g}(x_w, -\xi_i, t + h) + 2 \cdot W_i \cdot T_w, \tag{44}$$

where T_w is the wall temperature and W_i is the weight coefficients (same with Eq. (17)). And for the adiabatic boundary, it can also be realized by the bounce back rule, which is

$$\bar{g}(x_w, \xi_i, t + h) = \bar{g}(x_w, -\xi_i, t + h), \tag{45}$$

and ξ_i always pointing towards the flow field.

2.2. Multi-direct forcing immersed boundary method

The key point in IBM is how to calculate the boundary force on the Lagrangian points. According to this question, there are two different ways widely used to implement IBM, which are the penalty method and the direct forcing method. To avoid using the user defined spring constant in the calculation, the direct forcing IBM is chosen in the present study.

To implement the direct-forcing IBM, we first need to calculate the immediate velocity $\mathbf{u}^*$ and temperature T^* without the IB force and heat source from Eq. (16) and Eq. (33). And the boundary velocity and temperature on the Lagrangian points can be interpolated with the delta function

$$\mathbf{U}_b^* = \sum_{i,j} \mathbf{u}^* \cdot D(\mathbf{X_b} - \mathbf{X_{i,j}}) \cdot \Delta x^2, \tag{46}$$

$$T_b^* = \sum_{i,j} T^* \cdot D(\mathbf{X_b} - \mathbf{X_{i,j}}) \cdot \Delta x^2, \tag{47}$$

where the $\mathbf{U_b^*}$ and $\mathrm{T_b^*}$ are the immediate velocity and temperature on the immersed boundary without IB forces, Δx is the grid size. $\boldsymbol{X_{i,j}}$ and $\boldsymbol{X_b}$ are the Eulerian and Lagrangian points, respectively. The delta function used here is

$$D(\mathbf{X}_b - \mathbf{X}_{ij}) = \frac{1}{\Delta x^2}\delta(x_b - x_{ij})\delta(y_b - y_{ij}),$$

$$\delta(r) = \begin{cases} \frac{1}{4} \cdot \left(1 + \cos\left(\frac{\pi|r|}{2\Delta x}\right)\right) & |r| \leqslant 2 \\ 0 & |r| > 2 \end{cases}. \tag{48}$$

As the desired velocity $\mathbf{U_b}$ and temperature $\mathrm{T_b}$ on the boundary are designed, the IBM momentum force $\boldsymbol{F_b}$ and temperature force Q_b are designed as [37,38]

$$\mathbf{F}_b = 2\rho_b \frac{\mathbf{U_b} - \mathbf{U_b^*}}{\Delta t}, \tag{49}$$

$$Q_b = 2\rho_b \frac{T_b - T_b^*}{\Delta t}, \tag{50}$$

where ρ_b is the density of Lagrangian points and it is also interpolated from the Eulerian points using delta function. Now one needs to distribute the forces on the Lagrangian points back to the Eulerian points using the same delta function

$$f_{ij} = \sum_b \mathbf{F}_b \cdot D(\mathbf{X_b} - \mathbf{X_{i,j}}) \cdot \Delta s, \tag{51}$$

$$q_{ij} = \sum_b Q_b \cdot D(\mathbf{X_b} - \mathbf{X_{i,j}}) \cdot \Delta s, \tag{52}$$

where $\boldsymbol{f_{ij}}$ and q_{ij} are the force terms on the Eulerian points. Δs is the arc length between two Lagrangian points. Finally one can update the velocity and temperature on the Eulerian points [37,38]

$$\mathbf{u} = \mathbf{u} + \frac{\Delta t}{2\rho} \bullet f_{ij}, \tag{53}$$

$$T = T + \frac{\Delta t}{2\rho} \bullet q_{ij}. \tag{54}$$

Eqs. (46)–(54) are the procedure of the conventional direct-forcing IBM. But the conventional method cannot ensure the no-slip condition on the boundary due to delta function interpolation errors causing the streamlines to penetrate through the immersed boundary. To avoid this problem, the multi-direct forcing method was first used by Luo et al. [24]. It is easy to implement and one only needs to iterate the forcing procedure, i.e. Eqs. (46)–(54), several times until the error is small enough. In general, 10 times of iteration is enough to ensure the no-lisp boundary condition [24,37].

2.3. DUGKS with external force

The external force has not been considered in the original DUGKS algorithm described in Section 2.1.1. But in the present study, one has to consider the buoyancy force and IB force in the computation. Two different methods can be used to deal with the external force in DUGKS.

2.3.1. External force involved in the DUGKS

In the previous study, Peng et al. [12] and Wu et al. [13] deal with the external force by adding the force term in Eq. (1), just like Eq. (42), which can also be rewritten as

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla_x f = \overline{\Omega} \equiv \frac{f^{eq} - f}{\tau_v} + F, \tag{55}$$

and $\overline{\Omega}$ is the new collision term, the force term F is

$$F = -\mathbf{a} \cdot \nabla_\xi f \approx \mathbf{a} \cdot \frac{(\xi - \mathbf{u})}{RT_1} f^{eq}, \tag{56}$$

where **a** is the acceleration of external force. And Eq. (3) should also be rewritten as

$$\rho \mathbf{u} = \int \xi f d\xi + \frac{\Delta t}{2} \rho \mathbf{a}, \tag{57}$$

and the same modifications should also be done with Eq. (12) and (16). Importantly, Eq. (13) and (14) should be rewritten as [12,13]

$$f(x_b, \xi, t_n + h) = \frac{2\tau_v}{2\tau_v + h} \bar{f}(x_b, \xi, t_n + h) + \frac{h}{2\tau_v + h} (f^{eq}(x_b, \xi, t_n + h) + \tau_v \cdot F), \tag{58}$$

$$\bar{f}^+ = \frac{2\tau_v - h}{2\tau_v + \Delta t} \tilde{f} + \frac{3h}{2\tau_v + \Delta t} (f^{eq} + \tau_v \cdot F). \tag{59}$$

One problem should be noted here is that the interface distribution function $f(x_b, \xi, t_n + h)$ in Eq. (58) also has the force term F. But within the finite-volume framework, the IB forces which are obtained from Eqs. (51) and (52), are only distributed on the cell center, and one cannot obtain the IB force at the interface directly [39]. So the above method is not suitable if there is an immersed boundary.

2.3.2. Strang-splitting method

Strang-splitting method is another way to deal with the external force [40]. When using this method, one does not need to modify the original DUGKS procedure, and a half of the force is added before and after the DUGKS procedure. The Strang-splitting method can be described as [41]

$$\text{pre-forcing}: \quad \frac{\partial f}{\partial t} = \frac{1}{2} F, \tag{60}$$

$$\text{DUGKS}: \quad \frac{\partial f}{\partial t} + \xi \cdot \nabla f = \Omega \equiv \frac{f^{eq} - f}{\tau_v}, \tag{61}$$

$$\text{post-forcing}: \quad \frac{\partial f}{\partial t} = \frac{1}{2} F. \tag{62}$$

In the present study, as we tracked $\tilde{f}$ and $\tilde{g}$ in DUGKS, the pre-forcing step can be implemented as

$$\tilde{f}_i^* = \tilde{f}_i + \frac{\Delta t}{2} \cdot F_i, \tag{63a}$$

$$\tilde{g}_i^* = \tilde{g}_i + \frac{\Delta t}{2} \cdot Q_i, \tag{64a}$$

where [12,13,5,38]

$$F_i = \left(a + \frac{f_{ij}}{\rho}\right) \cdot \frac{(\xi - \mathbf{u})}{RT_1} f^{eq}, \tag{63b}$$

$$Q_i = W_i \cdot q_{ij}. \tag{64b}$$

The velocity and temperature also need to be updated as

$$\mathbf{u}^* = \mathbf{u} + \frac{\Delta t}{2} \cdot \left(a + \frac{f_{ij}}{\rho}\right), \quad T^* = T + \frac{\Delta t}{2} \cdot \frac{q_{ij}}{\rho}, \tag{65}$$

where **a** is the acceleration associated with the buoyancy force, f_{ij} and q_{ij} are the IBM momentum and energy forces. The post-forcing step is similar. But one should note that, in IB-DUGKS, the update of velocity and temperature in post-forcing step has been done in the IBM procedure. In order to prove the accuracy of the Strang-splitting method, a comparison between the two different methods discussed in Sections 2.3.1 and 2.3.2 will be made in next section.

3. Results and discussion

In order to validate the new method, a few natural convection problems are simulated in this section. Some parameters used in all the following simulations will be defined first. The Rayleigh number, which is viewed as a key parameter in natural convection problems, is defined as

$$Ra = \frac{g_o \beta \Delta T L^3}{\nu \kappa}, \tag{66}$$

where g_o is the acceleration of gravity, β is the coefficient of thermal expansion, ΔT is the temperature difference between the hot and cold boundary, L is the characteristic length, ν is the kinematic viscosity, κ is the thermal diffusivity coefficient. The Prandtl number Pr has been defined in Eq. (35), and $Pr = 0.71$ in the present study (for the air) if not stated. For simplicity, we define $RT_1 = RT_2 = 1/3$, just like in the D2Q9 LBM (other values can also be chosen in DUGKS). So the sound speed c_s can be determined ($RT = c_s^2$). The characteristic velocity U_0 can be defined as

$$U_0 = \sqrt{g_o \beta \Delta T L}, \tag{67}$$

and the Mach number is defined as $Ma = U/c_s$. In order to satisfy the nearly incompressible flow, Ma = 0.1 is used in the present study and the characteristic velocity U_0 can then be obtained. The time step Δt (=$\eta \Delta x_{min}/C_{max}$) is determined by the CFL number η, minimum grid spacing $\Delta x_{min,}$ and maximal discrete velocity C_{max}. We set CFL = 0.9 in the present study, which can give a larger time step. In the practical simulation, as we have defined the Mach number, sound speed and Prandtl number, then one can calculate the viscosity and diffusivity first according to the Ra number, and the relaxation times are calculated at last according to Eqs. (19) and (35).

3.1. Natural convection in a square cavity

The first case we simulate is the natural convection in a square cavity. We use the two different ways to deal with the external force, as discussed in Sections 2.3.1 and 2.3.2. Here for this base flow problem, IBM is not needed so both DUGKS forcing methods can be used. The purpose is to make a comparison between the two methods and validate the accuracy of the Strang-splitting method which will be used in the IB-DUGKS simulations. The geometric configuration is illustrated in Fig. 1. The length of the square cavity is L, which is also the characteristic length of the cavity natural convection problems. The left vertical wall is maintained a constant high temperature $T_h = 1$, and the right vertical wall is maintained a constant low temperature $T_c = 0$. The horizontal top and bottom walls are adiabatic. The direction of the gravity is vertical and downward. The boundary conditions for velocity and temperature in Section 2.1.4 are implemented. We set a quiescent and isothermal field as the initial state (the initial temperature $T_0 = (T_h + T_c)/2$). In the simulation, a uniform mesh of 100 × 100 is used. As we will use two different methods to deal with the external force, we name the method in Section 2.3.1 as Case 1, and the Strang-splitting method as Case 2.

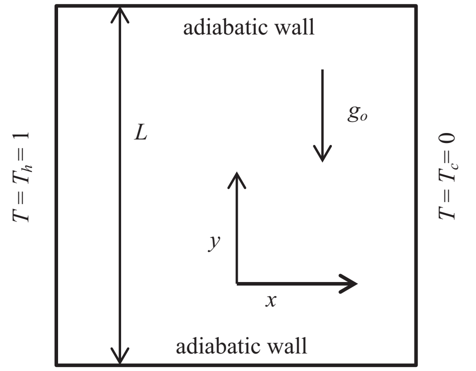


Fig. 1. Illustration of the flow domain for the natural convection in a square cavity.

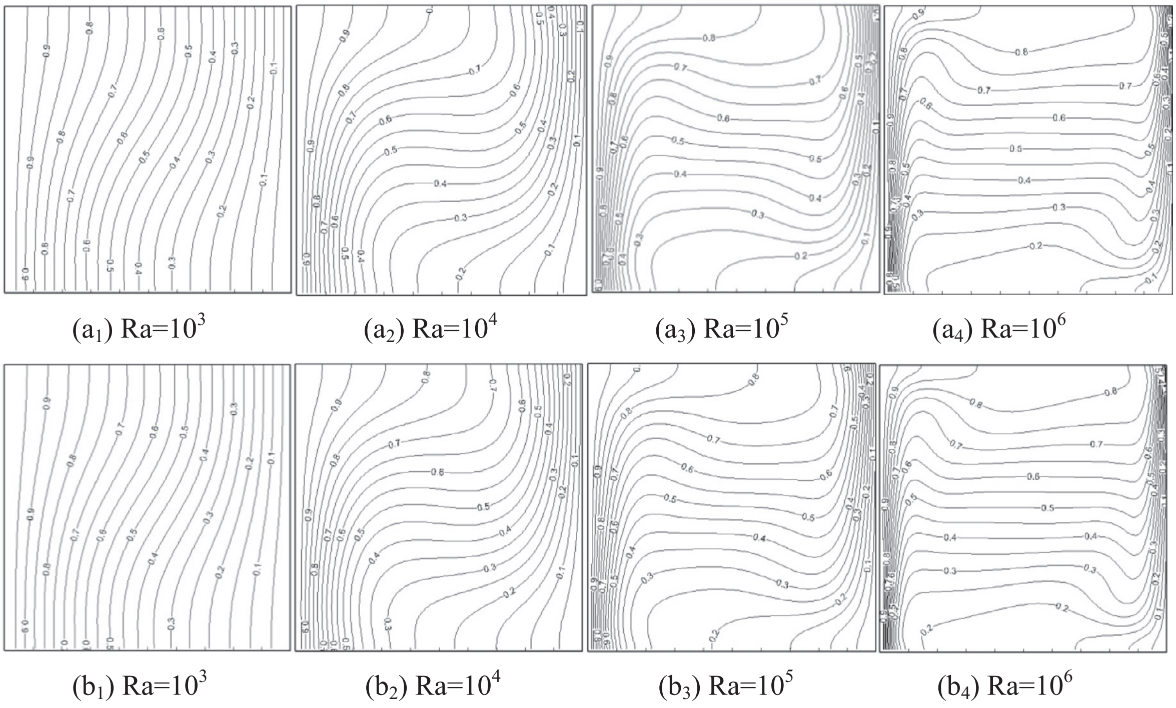
The isothermals of Case 1 and Case 2 for Ra = 10^3, 10^4, 10^5 and 10^6 are shown in Fig. 2. From the figures we can find that there is no difference between the two cases, and they all agree well with the results reported in the literature [12,36]. Our main purpose is to compare the two methods quantitatively with the literature. The data of interest here include the maximum horizontal velocity at the mid-width, u_{max}, and its location, the vertical velocity at the mid-height, v_{max}, and its location, the maximum Nusselt number on the cold wall, Nu_{max}, and its location, and the average Nusselt number in the whole domain. The velocities in our results are all normalized with the reference velocity $u_o = \kappa/L$. The Nusselt number on the cold wall can be obtained directly from the temperature gradient on the cold wall, and the average Nusselt number can be calculated by [42]

$$\overline{Nu} = \frac{1}{N_x N_y \Delta T}\sum_{i,j=1}^{N_x,N_y}\left(u_x T - \frac{\partial T}{\partial x}\right), \tag{68}$$

where N_x, and N_y are the grids number in x and y direction, u_x is the horizontal velocity and also normalized with the reference velocity $u_o = \kappa/L$. The results from Case 1 and Case 2, as well as the results from the literature are shown in Table 1. We conclude that the results from the two cases and from the literature agree well. This also confirms the accuracy of the Strang-splitting method. Then we will conduct IB-DUGKS with the Strang-splitting in the following simulations involving IBM.

3.2. Natural convection in a square cavity with a concentric circular cylinder

To validate the accuracy of the IB-DUGKS, the natural convection between a cold square and a concentric hot circular cylinder is conducted. The hot circular cylinder is the immersed body in the simulation. The geometric configuration is illustrated in Fig. 3. The length of the outer square is L, and the radius of the


(a$_1$) Ra=10^3 (a$_2$) Ra=10^4 (a$_3$) Ra=10^5 (a$_4$) Ra=10^6

(b$_1$) Ra=10^3 (b$_2$) Ra=10^4 (b$_3$) Ra=10^5 (b$_4$) Ra=10^6

Fig. 2. Comparison of the isothermals of natural convection (a for Case 1; b for Case 2).

Table 1
Numerical results of the natural convection in a square cavity.

Ra		u_{max}	*y*	v_{max}	*x*	Nu_{max}	*y*	Nu_{ave}
10^3	Case 1	3.643	0.82	3.699	0.18	1.508	0.91	1.1181
	Case 2	3.643	0.82	3.698	0.18	1.508	0.91	1.1181
	Davis [43]	3.649	0.813	3.697	0.178	1.505	0.908	1.118
10^4	Case 1	16.169	0.83	19.609	0.12	3.529	0.86	2.2424
	Case 2	16.168	0.83	19.608	0.12	3.530	0.86	2.2425
	Davis [43]	16.178	0.823	19.617	0.119	3.328	0.857	2.243
10^5	Case 1	34.788	0.86	68.162	0.07	7.633	0.92	4.4968
	Case 2	34.761	0.86	68.149	0.07	7.640	0.92	4.4977
	Davis [43]	34.73	0.855	68.59	0.066	7.717	0.919	4.519
10^6	Case 1	63.279	0.86	213.621	0.04	16.527	0.96	8.6638
	Case 2	63.121	0.86	213.482	0.04	16.561	0.96	8.6678
	Davis [43]	64.63	0.850	219.36	0.0379	17.925	0.962	8.8

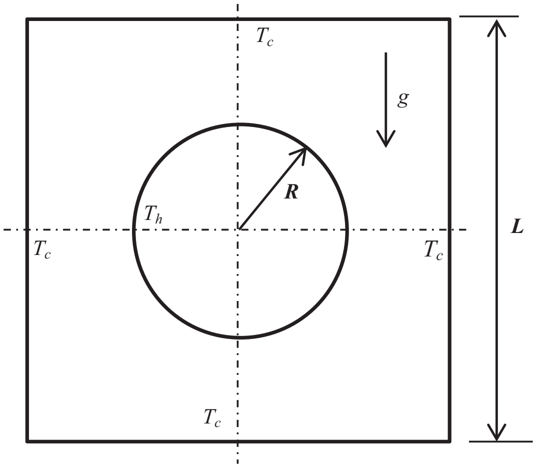
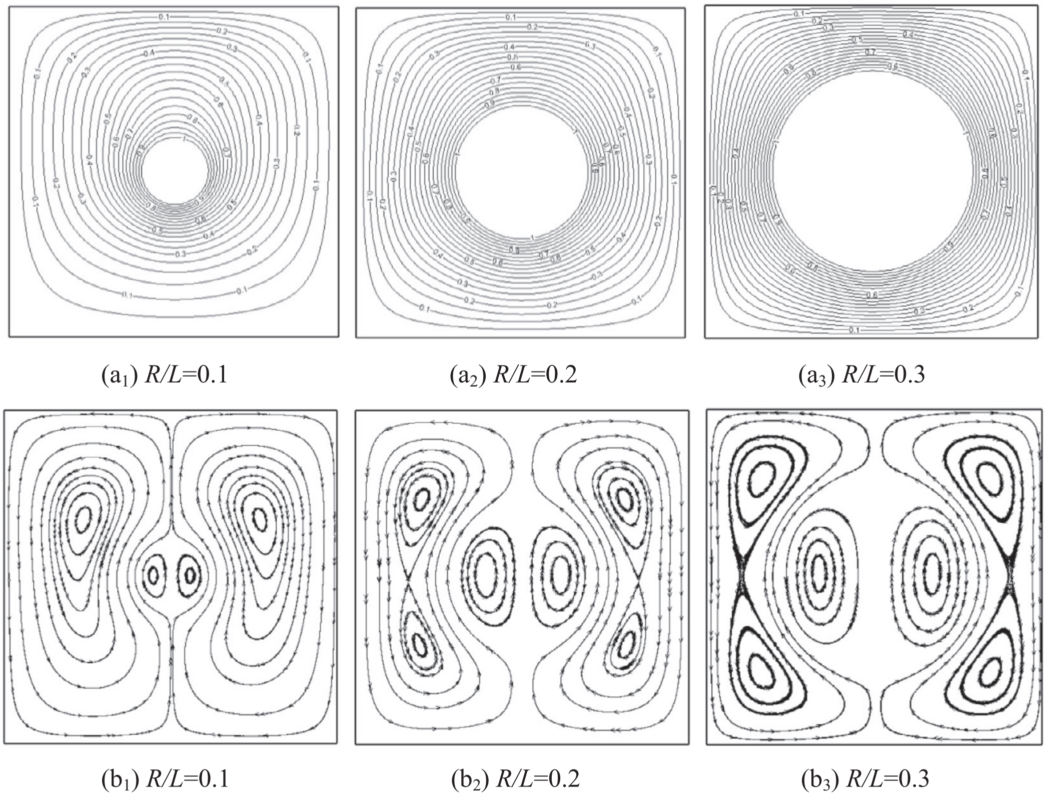


Fig. 3. Illustration of natural convection in a square cavity with a concentric circular cylinder.

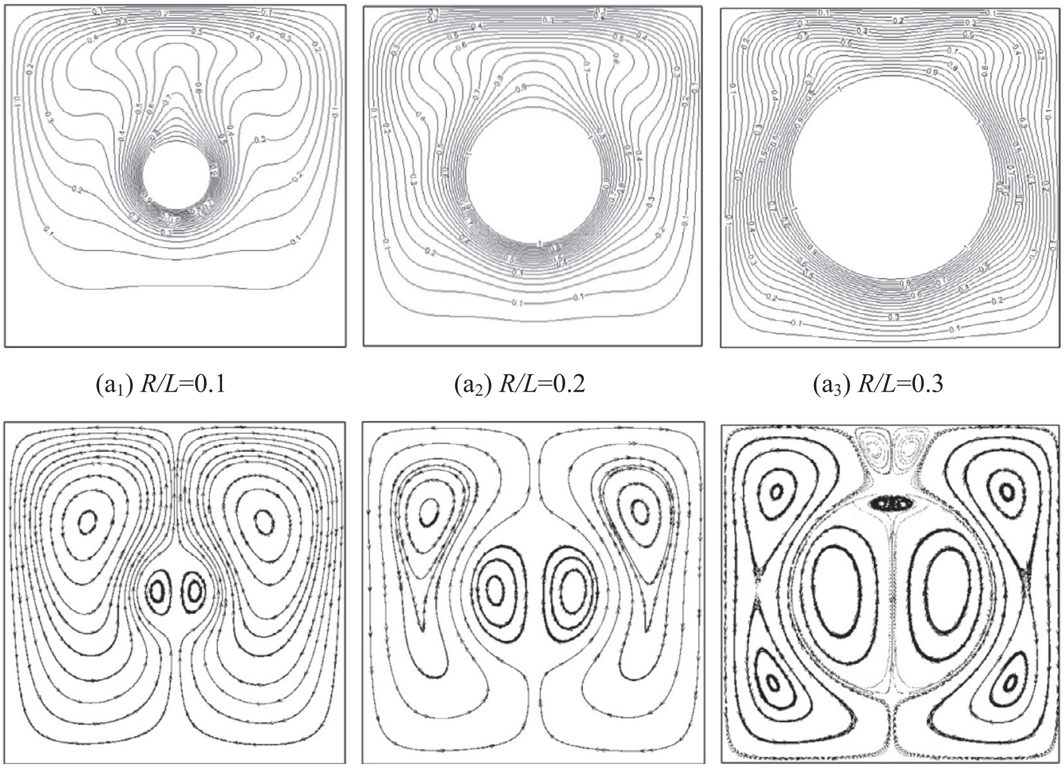
inner cylinder is *R*. The inner cylinder is maintained a constant high temperature $T_h = 1$, and the outer square is maintained a constant low temperature $T_c = 0$. The ratio between *R* and *L* is varied, and three different configurations, $R/L = 0.1$, 0.2 and 0.3, are considered. In the simulations, a uniform mesh of 200 × 200 is adopted [6]. The Lagrangian points are uniformly distributed on the immersed boundary. In order to ensure a higher accuracy on the immersed boundary, the arc length Δs between two Lagrangian points should be small enough, which usually is smaller than the grid spacing Δx [23,24,37,38]. In the present study, 180, 360 and 540 Lagrangian points are used separately for the three different aspect ratios ($\Delta s = 0.7\Delta x$). The multi-direct forcing IBM is adopted. We set a quiescent and isothermal field as the initial state. The bounce back boundary condition for velocity and constant temperature boundary condition for temperature are implemented at the outer square.

The streamlines and isothermals of different configurations for $\mathrm{Ra} = 10^4$, 10^5 and 10^6 are presented in Figs. 4–6, and they show


(a$_1$) *R/L*=0.1 (a$_2$) *R/L*=0.2 (a$_3$) *R/L*=0.3

(b$_1$) *R/L*=0.1 (b$_2$) *R/L*=0.2 (b$_3$) *R/L*=0.3

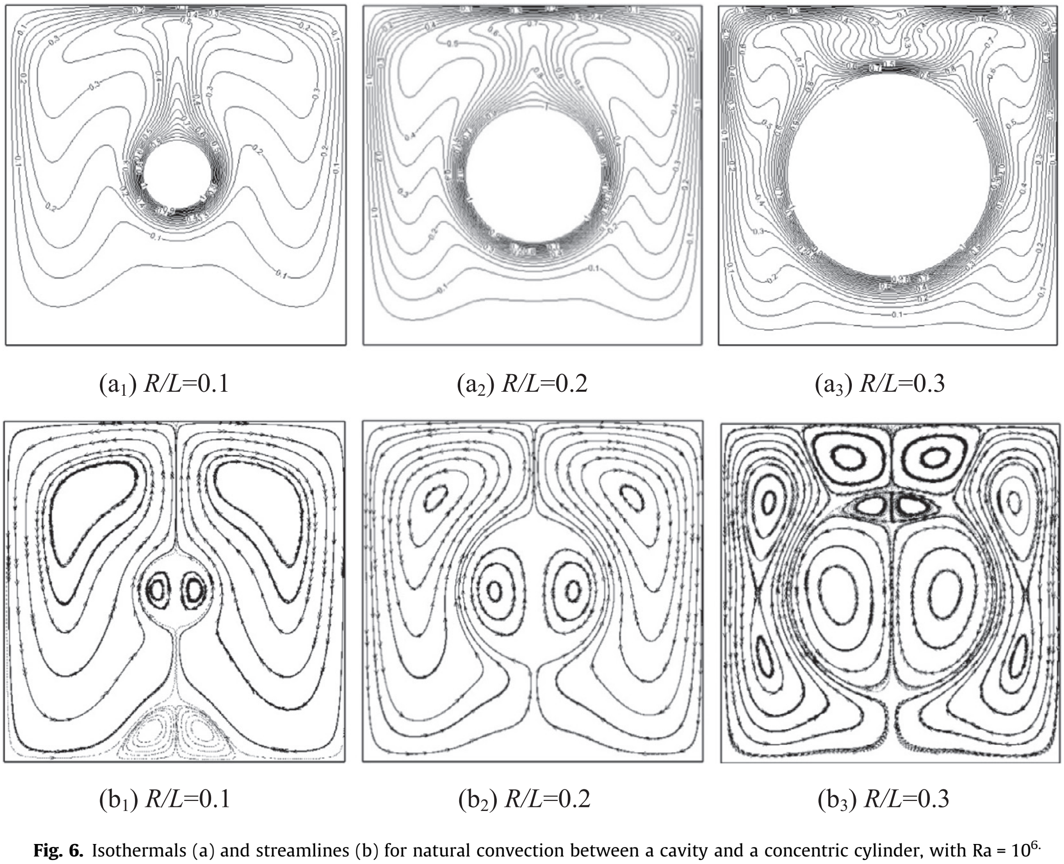
Fig. 4. Isothermals (a) and streamlines (b) for natural convection between a cavity and a concentric cylinder, with Ra = $10^{4.}$



(a$_1$) *R/L*=0.1 (a$_2$) *R/L*=0.2 (a$_3$) *R/L*=0.3

(b$_1$) *R/L*=0.1 (b$_2$) *R/L*=0.2 (b$_3$) *R/L*=0.3

Fig. 5. Isothermals (a) and streamlines (b) for natural convection between a cavity and a concentric cylinder, with Ra = $10^{5.}$



(a$_1$) *R/L*=0.1 (a$_2$) *R/L*=0.2 (a$_3$) *R/L*=0.3

(b$_1$) *R/L*=0.1 (b$_2$) *R/L*=0.2 (b$_3$) *R/L*=0.3

Fig. 6. Isothermals (a) and streamlines (b) for natural convection between a cavity and a concentric cylinder, with Ra = $10^{6.}$

good agreement with the results in the literature [3,38,44]. From the figures one can observe that they are all symmetrical about the mid-width vertical line. Due to the buoyancy, the air moves upward around the hot cylinder, and moves downward along the vertical cold wall. The Rayleigh number has significant influence on the heat transfer rate. When the Rayleigh number is small, heat transfer between the inner cylinder and the outer square is mainly by conduction. When the Rayleigh number increases, convection gradually dominates.

In order to quantitatively validate the results, the average Nusselt numbers are calculated. When using IBM, the average Nusselt number can be obtained with the following equation [44]

$$\overline{Nu} = \frac{S}{\kappa L_b \Delta T}\sum_i Q_b^i \Delta s_i \quad (i = 1 \sim \mathrm{m}), \tag{69}$$

where S is taken as half of the circumferential length of the inner circular cylinder, κ is the thermal diffusivity coefficient, L_b is the circumferential length of the inner circular cylinder, ΔT is the temperature difference between the hot and cold boundaries, Q_b is the heat flux on the Lagrangian points, Δs is the immersed boundary segment length, $\boldsymbol{m}$ is the number of the Lagrangian points on the immersed boundary.

Table 2
Comparison of the average Nusselt Numbers.

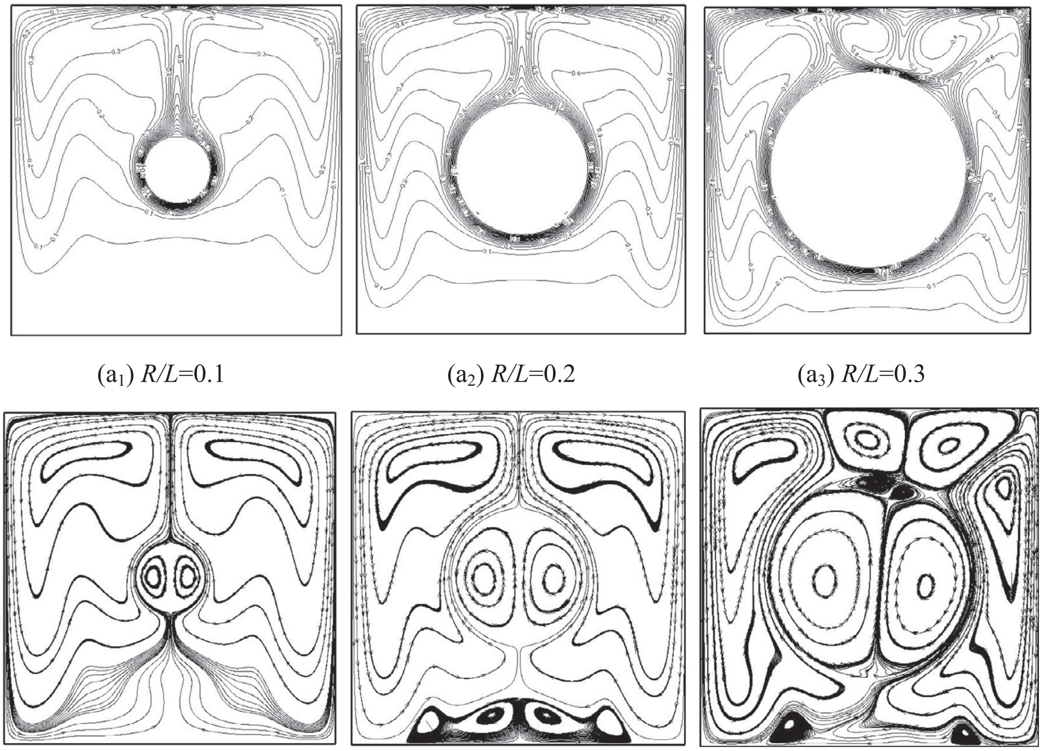
Ra	*R/L*	Present	Shu and Zhu [3]	Moukalled and Acharya [2]
10^4	0.1	2.09	2.08	2.071
	0.2	3.27	3.24	3.331
	0.3	5.46	5.40	5.826
10^5	0.1	3.81	3.79	3.825
	0.2	4.93	4.86	5.08
	0.3	6.28	6.21	6.212
10^6	0.1	6.15	6.11	6.107
	0.2	8.96	8.90	9.374
	0.3	12.08	12.00	11.62
10^7	0.1	10.33	–	–
	0.2	16.06	–	–
	0.3	22.19	–	–

The computed average Nusselt numbers and the results from the literatures are presented in Table 2. The results showed that the present results agree well with the literature data, which validated the accuracy of the present method for simulation of natural convection with a curved surface.

According to Wang et al. [18], the original DUGKS works well for large Ra numbers. So the large Ra numbers were also tested in this study. It was found that the proposed method still works well when the Ra number is larger than 10^{10} in the same domain (200×200), which proved the numerical stability of the new method. In consideration of the accuracy of the simulation, only the results of $\mathrm{Ra} = 10^7$ are presented in Fig. 7 and Table 2. One could investigate larger Ra numbers using higher grid resolutions to ensure the accuracy. From the results one can observe that the convection is more drastic with a larger Ra number and when $R/L = 0.3$ it is not symmetric anymore due to additional physical flow instability in the narrow space between the cold and hot surfaces.

In order to evaluate the accuracy of the proposed method, a set of simulations with different mesh resolutions are conducted. We use $R/L = 0.2$, $\mathrm{Ra} = 10^4$, and the meshes are 100×100, 200×200, 400×400 and 800×800. The CFL numbers were adjusted to keep the time step constant, which are 0.1, 0.2, 0.4 and 0.8, respectively. The L_2 errors in velocity and temperature fields are measured in Table 3, where the L_2 error is defined by [10]

$$E(\phi) = \frac{\sqrt{\sum_{x,y}|\phi(x,y,t) - \phi_e(x,y,t)|^2}}{\sqrt{\sum_{x,y}|\phi_e(x,y,t)|^2}}, \tag{70}$$


(a$_1$) *R/L*=0.1 (a$_2$) *R/L*=0.2 (a$_3$) *R/L*=0.3

(b$_1$) *R/L*=0.1 (b$_2$) *R/L*=0.2 (b$_3$) *R/L*=0.3

Fig. 7. Isothermals (a) and streamlines (b) for natural convection between a cavity and a concentric cylinder, with Ra = $10^{7.}$

Table 3
Error and convergence order in velocity and temperature.

N	100	200	400
E($\boldsymbol{u}$)	5.98×10^{-2}	2.63×10^{-2}	8.89×10^{-3}
Order	–	1.18	1.57
E(T)	1.78×10^{-2}	8.80×10^{-3}	3.58×10^{-3}
Order	–	1.02	1.30

where $\varphi = \boldsymbol{u}$ or T, and φ_e is the benchmark value. As there is no analytical value in this natural convection problem, the results with the mesh of 800 × 800 are chosen to be the benchmark solution. An overall first-order accuracy is confirmed. Since the DUGKS scheme is of second order accuracy in space [10,12], and the order of accuracy of the proposed method is reduced because of the IBM. It is consistent with the results from other references which used the IB-LBM [45–47].

3.3. Natural convection in a square cavity with an eccentric circular cylinder

To further examine the applicability of the present method, the natural convection in a square cavity with an eccentric circular cylinder is conducted. The geometric configuration is illustrated in Fig. 8. The length of the outer square is L, and the radius of the inner cylinder R is $0.2L$. The cylinder is off-center in vertical direction and the distance is $0.1L$. The inner cylinder is maintained a constant high temperature $T_h = 1$, and the two vertical walls are maintained a constant low temperature $T_c = 0$. The horizontal top and bottom walls are adiabatic. The boundary conditions are same with the former cases. We also set a quiescent and isothermal field as the initial state. In the simulations, a uniform mesh of 200 × 200 is adopted [32,48]. The multi-direct forcing IBM is used and 360 Lagrangian points uniformly distributed on the immersed boundary. The Rayleigh number and Prandtl are 10^6 and 10, respectively.

The isothermal and streamline are presented in Fig. 9 and they agree well with the plots in literature [42]. Due to the large Rayleigh number, the convection effect dominates in this case. The air moves upward around the hot cylinder, and moves downward along the vertical cold wall, just like the results in the former section. But the temperature profile is different because of the adiabatic horizontal walls.

In order to quantitatively compare the results with the literature, the local Nusselt numbers on the cold vertical walls are obtained. A comparison with the results from the literatures has been made in Fig. 10 and it also shows a good agreement [32,48].

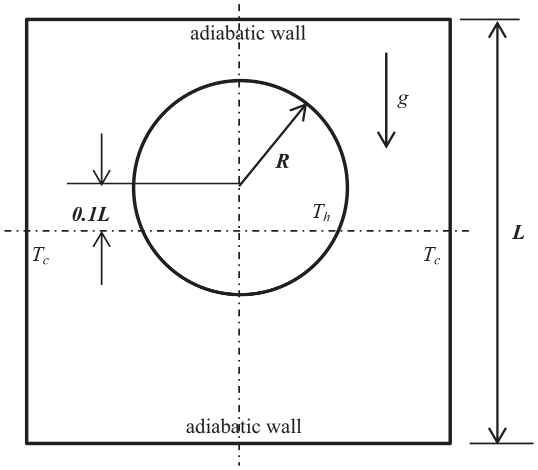


Fig. 8. Illustration of natural convection in a square cavity with an eccentric circular cylinder.

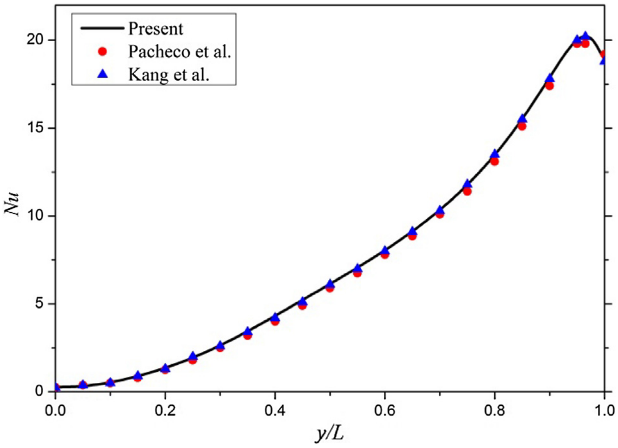


Fig. 10. Comparison of the local Nusselt number along the cold vertical wall of the cavity.

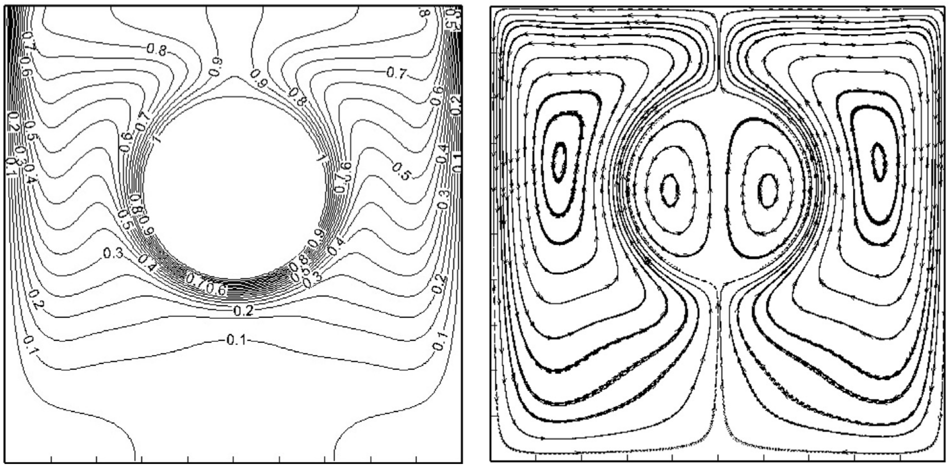


Fig. 9. Isothermals (left) and streamlines (right) for natural convection in a square cavity with an eccentric circular cylinder Ra = $10^{6.}$

4. Summary and conclusion

In this study, an immersed boundary-discrete unified gas kinetic scheme (IB-DUGKS) for simulating natural convection flows with a curved boundary is proposed, developed, and validated. The main motivation was to extend the applicability of DUGKS to thermal flow problems with curved boundaries. In order to treat a curved boundary, the immersed boundary method (IBM) is employed. Specifically, the direct-forcing IBM is used in this study due to its simplicity, with an iteration procedure to ensure the accuracy of the no-slip boundary condition. The Strang-splitting method is used to incorporate the IB force within DUGKS.

As a first step, we simulated the base case of the natural convection in a square cavity to confirm the accuracy of the Strang-splitting method. In this simple case without a curved immersed boundary, two alternative implementation methods to include the external (buoyancy) force, described in Sections 2.3.1 and 2.3.2, were compared. The isothermals are consistent between the two methods. The results were also compared with the data from the literature, and an excellent agreement was found. These comparisons verify the accuracy of the Strang-splitting method.

Next, several benchmark cases were simulated to validate the accuracy of the IB-DUGKS. The first is the natural convection heat transfer between a cold square and a concentric hot circular cylinder. Three different configurations, $R/L = 0.1$, 0.2 and 0.3, were considered. The streamlines and isothermals of different configurations at $\mathrm{Ra} = 10^4$, 10^5 and 10^6 are consistent with the literature results. The calculated average Nusselt numbers are in good agreement with the data from the literature. The results validate the accuracy of the IB-DUGKS as a direct numerical simulation tool for natural convection with curved boundaries. The numerical stability and order of accuracy of the proposed method were also studied.

We also simulated the natural convection heat transfer in a square cavity with an eccentric circular cylinder. The isothermals, streamlines and local Nusselt numbers were compared with the results from the literature. Not only the isothermals and streamlines, but also the Nusselt numbers agree well with the literature. This case further supports the accuracy of the IB-DUGKS and also extends its applicability.

From the above results, it is clear that the proposed IB-DUGKS is capable of simulating natural convection problems with curved boundaries. The combination of IBM and DUGKS provides an effective way for the DUGKS to deal with curved boundary and significantly extends the applicability of the DUGKS to heat transfer problems of complex geometry. Other natural convection geometries and forced convection problems can also be tested. As a new and powerful numerical method, the application domains of DUGKS will continue to expand in the future.

Conflict of interest

We declare that we have no conflict of interest.

Acknowledgements

This work has been supported by the Dalian Maritime University and the U.S. National Science Foundation (NSF) under grants CNS1513031, CBET-1235974, and AGS-1139743.

References

[1] T.H. Kuehn, R.J. Goldstein, An experimental and theoretical study of natural convection in the annulus between horizontal concentric cylinders, J. Fluid Mech. 74 (1976) 695–719.

[2] F. Moukalled, S. Acharya, Natural convection in the annulus between concentric horizontal circular and square cylinders, J. Thermophys. Heat Transfer 10 (1996) 524–531.

[3] C. Shu, Y.D. Zhu, Efficient computation of natural convection in a concentric annulus between an outer square cylinder and an inner circular cylinder, Int. J. Numer. Meth. Fluids 38 (2002) 429–445.

[4] B.S. Kim, D.S. Lee, M.Y. Ha, H.S. Yoon, A numerical study of natural convection in a square enclosure with a circular cylinder at different vertical locations, Int. J. Heat Mass Transfer 51 (2008) 1888–1906.

[5] L. Li, R. Mei, J.F. Klausner, Heat transfer evaluation on curved boundaries in thermal lattice boltzmann equation method, J. Heat Transfer 136 (2014) 012403.

[6] H.K. Jeong, H.S. Yoon, M.Y. Ha, M. Tsutahara, An immersed boundary-thermal lattice Boltzmann method using an equilibrium internal energy density approach for the simulation of flows with heat transfer, J. Comput. Phys. 229 (2010) 2526–2543.

[7] K. Xu, A gas-kinetic BGK scheme for the Navier-Stokes equations and its connection with artificial dissipation and Godunov method, J. Comput. Phys. 171 (2001) 289–335.

[8] S. Succi, The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond, Oxford University Press, 2001.

[9] K. Xu, J. Huang, A unified gas-kinetic scheme for continuum and rarefied flows, J. Comput. Phys. 229 (2010) 7747–7764.

[10] Z. Guo, K. Xu, R. Wang, Discrete unified gas kinetic scheme for all Knudsen number flows: low-speed isothermal case, Phys. Rev. E 88 (2013) 033305.

[11] Z. Guo, R. Wang, K. Xu, Discrete unified gas kinetic scheme for all Knudsen number flows. II. Thermal compressible case, Phys. Rev. E 91 (2015) 033313.

[12] P. Wang, S. Tao, Z. Guo, A coupled discrete unified gas-kinetic scheme for Boussinesq flows, Comput. Fluids 120 (2015) 70–81.

[13] C. Wu, B. Shi, Z. Chai, P. Wang, Discrete unified gas kinetic scheme with a force term for incompressible fluid flows, Comput. Math. Appl. 71 (2016) 2608–2629.

[14] L. Zhu, Z. Guo, K. Xu, Discrete unified gas kinetic scheme on unstructured meshes, Comput. Fluids 127 (2016) 211–225.

[15] Z. Guo, K. Xu, Discrete unified gas kinetic scheme for multiscale heat transfer based on the phonon Boltzmann transport equation, Int. J. Heat Mass Transfer 102 (2016) 944–958.

[16] Y. Bo, P. Wang, Z. Guo, L.-P. Wang, DUGKS simulations of three-dimensional Taylor-Green vortex flow and turbulent channel flow, Comput. Fluids 155 (2017) 9–21.

[17] L. Zhu, S. Chen, Z. Guo, dugksFoam: an open source OpenFOAM solver for the Boltzmann model equation, Comput. Phys. Commun. 213 (2017) 155–164.

[18] P. Wang, Y. Zhang, Z. Guo, Numerical study of three-dimensional natural convection in a cubical cavity at high Rayleigh numbers, Int. J. Heat Mass Transfer 113 (2017) 217–228.

[19] S. Tao, B. Chen, X. Yang, S. Huang, Second-order accurate immersed boundary-discrete unified gas kinetic scheme for fluid-particle flows, Comput. Fluids 165 (2018) 54–63.

[20] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.

[21] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.

[22] J. Mohd-Yusof, Combined Immersed Boundaries B-spline Methods for Simulations of Flows in Complex Geometries, *CTR Annual Research Briefs*, NASA Ames, Stanford University, 1997, pp. 317–327.

[23] Z.-G. Feng, E.E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, J. Comput. Phys. 202 (2005) 20–51.

[24] K. Luo, Z. Wang, J. Fan, K. Cen, Full-scale solutions to particle-laden flows: multidirect forcing and immersed boundary method, Phys. Rev. E 76 (2007) 066709.

[25] J. Wu, C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys. 228 (2009) 1963–1979.

[26] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (2001) 132–150.

[27] J. Kim, H. Choi, An immersed-boundary finite-volume method for simulation of heat transfer in complex geometries, KSME Int. J. 18 (2004) 1026–1035.

[28] X.D. Niu, C. Shu, Y.T. Chew, Y. Peng, A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows, Phys. Lett. A 354 (2006) 173–182.

[29] C. Shu, N. Liu, Y.T. Chew, A novel immersed boundary velocity correction–lattice Boltzmann method and its application to simulate flow past a circular cylinder, J. Comput. Phys. 226 (2007) 1607–1622.

[30] F. Ilinca, J.-F. Hétu, A finite element immersed boundary method for fluid flow around rigid objects, Int. J. Numer. Meth. Fluids 65 (2011) 856–875.

[31] Z. Wang, J. Fan, K. Luo, K. Cen, Immersed boundary method for the simulation of flows with heat transfer, Int. J. Heat Mass Transfer 52 (2009) 4510–4518.

[32] S.K. Kang, Y.A. Hassan, A direct-forcing immersed boundary method for the thermal lattice Boltzmann method, Comput. Fluids 49 (2011) 36–45.

[33] C.-C. Liao, C.-A. Lin, Simulations of natural and forced convection flows with moving embedded object using immersed boundary method, Comput. Methods Appl. Mech. Eng. 213–216 (2012) 58–70.

[34] D. Pan, A general boundary condition treatment in immersed boundary methods for incompressible Navier-Stokes equations with heat transfer, Numer. Heat Transfer, Part B: Fundam. 61 (2012) 279–297.

[35] A. Eshghinejadfard, D. Thévenin, Numerical simulation of heat transfer in particulate flows using a thermal immersed boundary lattice Boltzmann method, Int. J. Heat Fluid Flow 60 (2016) 31–46.

[36] Z. Guo, B. Shi, C. Zheng, A coupled lattice BGK model for the Boussinesq equations, Int. J. Numer. Meth. Fluids 39 (2002) 325–342.

[37] S.K. Kang, Y.A. Hassan, A comparative study of direct-forcing immersed boundary-lattice Boltzmann methods for stationary complex boundaries, Int. J. Numer. Meth. Fluids 66 (2011) 1132–1158.

[38] T. Seta, Implicit temperature-correction-based immersed-boundary thermal lattice Boltzmann method for the simulation of natural convection, Phys. Rev. E 87 (2013) 063304.

[39] R. Yuan, C. Zhong, H. Zhang, An immersed-boundary method based on the gas kinetic BGK scheme for incompressible viscous flow, J. Comput. Phys. 296 (2015) 184–208.

[40] P.J. Dellar, An interpretation and derivation of the lattice Boltzmann method using Strang splitting, Comput. Math. Appl. 65 (2013) 129–141.

[41] H. Zhang, S. Tao, Z. Guo, An immersed boundary method based on the discrete unified gas kinetic scheme, J. Eng. Thermophys. 37 (2016) 539–544.

[42] J. Wang, D. Wang, P. Lallemand, L.-S. Luo, Lattice Boltzmann simulations of thermal convective flows in two dimensions, Comput. Math. Appl. 65 (2013) 262–286.

[43] G.D.V. Davis, Natural convection of air in a square cavity: a bench mark numerical solution, Int. J. Numer. Meth. Fluids 3 (1983) 249–264.

[44] W.W. Ren, C. Shu, J. Wu, W.M. Yang, Boundary condition-enforced immersed boundary method for thermal flow problems with Dirichlet temperature condition and its applications, Comput. Fluids 57 (2012) 40–51.

[45] Y. Hu, D. Li, S. Shu, X. Niu, An efficient immersed boundary-lattice Boltzmann method for the simulation of thermal flow problems, Comm. Comput. Phys. 20 (2016) 1210–1257.

[46] F.-B. Tian, H. Luo, L. Zhu, J.C. Liao, X.-Y. Lu, An efficient immersed boundary-lattice Boltzmann method for the hydrodynamic interaction of elastic filaments, J. Comput. Phys. 230 (2011) 7266–7283.

[47] C. Zhang, Y. Cheng, L. Zhu, J. Wu, Accuracy improvement of the immersed boundary-lattice Boltzmann coupling scheme by iterative force correction, Comput. Fluids 124 (2016) 246–260.

[48] J.R. Pacheco, T. Rodic, R.E. Peck, A. Pacheco-Vega, Numerical simulations of heat transfer and fluid flow problems using an immersed-boundary finite-volume method on nonstaggered grids, Numer. Heat Transfer, Part B: Fundam. 47 (2005) 1–24.