IMMERSED BOUNDARY METHODS IN THE LATTICE BOLTZMANN EQUATION FOR FLOW SIMULATION

A Dissertation

by

SHIN KYU KANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2010

Major Subject: Nuclear Engineering

Immersed Boundary Methods in the Lattice Boltzmann Equation for Flow Simulation

IMMERSED BOUNDARY METHODS IN THE LATTICE BOLTZMANN EQUATION FOR FLOW SIMULATION

A Dissertation

by

SHIN KYU KANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Yassin A. Hassan
Committee Members,	Kalyan Annamalai
	Hamn-Ching Chen
	William H. Marlow
Head of Department,	Raymond J. Juzaitis

December 2010

Major Subject: Nuclear Engineering

ABSTRACT

Immersed Boundary Methods in the Lattice Boltzmann Equation for Flow Simulation.

(December 2010)

Shin Kyu Kang, B.S.; M.S., Seoul National University, South Korea

Chair of Advisory Committee: Dr. Yassin A. Hassan

In this dissertation, we explore direct-forcing immersed boundary methods (IBM) under the framework of the lattice Boltzmann method (LBM), which is called the direct-forcing immersed boundary-lattice Boltzmann method (IB-LBM).

First, we derive the direct-forcing formula based on the split-forcing lattice Boltzmann equation, which recovers the Navier-Stokes equation with second-order accuracy and enables us to develop a simple and accurate formula due to its kinetic nature. Then, we assess the various interface schemes under the derived direct-forcing formula. We consider not only diffuse interface schemes but also a sharp interface scheme. All tested schemes show a second-order overall accuracy. In the simulation of stationary complex boundary flows, we can observe that the sharper the interface scheme is, the more accurate the results are.

The interface schemes are also applied to moving boundary problems. The sharp interface scheme shows better accuracy than the diffuse interface schemes but generates spurious oscillation in the boundary forcing terms due to the discontinuous change of nodes for the interpolation. In contrast, the diffuse interface schemes show smooth

change in the boundary forcing terms but less accurate results because of discrete delta functions. Hence, the diffuse interface scheme with a corrected radius can be adopted to obtain both accurate and smooth results.

Finally, a direct-forcing immersed boundary method (IBM) for the thermal lattice Boltzmann method (TLBM) is proposed to simulate non-isothermal flows. The direct-forcing IBM formulas for thermal equations are derived based on two TLBM models: a double-population model with a simplified thermal lattice Boltzmann equation (Model 1) and a hybrid model with an advection-diffusion equation of temperature (Model 2). The proposed methods are validated through natural convection problems with stationary and moving boundaries. In terms of accuracy, the results obtained from the IBMs based on both models are comparable and show a good agreement with those from other numerical methods. In contrast, the IBM based on Model 2 is more numerically efficient than the IBM based on Model 1.

Overall, this study serves to establish the feasibility of the direct-forcing IB-LBM as a viable tool for computing various complex and/or moving boundary flow problems.

DEDICATION

To my wife Hye Jin and our children Seokjoo and Yeun.

ACKNOWLEDGEMENTS

I am heartily thankful to Dr. Yassin A. Hassan, my advisor and the chair of my degree advisory committee, for his guidance during my research and study at Texas A&M University. His perpetual energy and enthusiasm in research have motivated me.

I would also like to thank my committee members, Dr. Kalyan Annamalai, Dr. Hamn-Ching Chen, and Dr. William H. Marlow, for taking the time to work with me to fulfill all the requirements of the doctoral program.

Finally, I would like to thank my family. Thanks to my parents for their encouragement and endless love and to my brother for his care and help. And most of all, special thanks to my wife Hye Jin and our children Seokjoo and Yeun for their sacrifice, understanding, patience, encouragement, and love. I am forever grateful.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF FIGURES

Page

Page

Page

Page

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

In computational fluid dynamics (CFD), a primary issue is the development of accurate, efficient treatments of complex and/or moving boundaries. Many researchers have developed various numerical methods to resolve this issue. One example of such methods is the immersed boundary method (IBM).

A. Background

1. Immersed boundary method (IBM)

The IBM can be defined as a non-body-conformal grid method which adds a force density (or acceleration) term either explicitly or implicitly to the flow governing equation to satisfy the no-slip condition on the boundary. The adoption of the structured non-body-conformal grid (usually the Cartesian grid) relieves the burden of meshing and reduces the amount of memory and CPU time used compared with unstructured body-conformal grids, and the accurate evaluation of the force density term maintains a high accuracy.

In general, there are two ways to evaluate the boundary force density in the IBM– feedback-forcing method and direct-forcing method. In the feedback-forcing method [1-6], the boundary force density is computed through the feedback process

---

This dissertation follows the style of *International Journal for Numerical Methods in Fluids*.

forcing method [7-27], the boundary force density is directly determined by using a flow equation, such as the Navier-Stokes equation (NSE) or the lattice Boltzmann equation (LBE).

The IBMs require interface schemes additionally because the boundary, in general, does not match the computational nodes. The interface scheme can be subdivided into the diffuse and sharp interface schemes. In the diffuse interface scheme, forcing points, on which the boundary force is evaluated, are located on the boundary, while in the sharp interface scheme, forcing points are placed on computational nodes closest to the boundary. In the diffuse interface scheme, the boundary force effect needs to be distributed into neighboring computational nodes because forcing points are not on the computational nodes. In general, discrete delta functions are used for the force distributions, thus making the boundary diffuse. That is why we call it the diffuse interface scheme. On the other hand, in the sharp interface scheme, the velocity on the forcing node is determined by interpolation so that the corresponding boundary point may satisfy the no-slip condition. Because the type of interface scheme we adopt directly influences the accuracy of the IBM, the selection of the interface scheme is another crucial issue together with the force evaluation method in the IBM.

a. Feedback-forcing IBM

The *feedback-forcing IBM* was pioneered by Peskin [1-2] for the simulation of blood flow in an elastic heart valve. The boundary force was computed by Hooke's law, where the force is a function of the deformation of the surface boundary with the spring

constant. Lai and Peskin [3] applied this forcing method to a rigid boundary problem, such as flow past a circular cylinder, by taking the spring constant to be a large value; that is, they made the spring stiff. Goldstein et al. [4] and Saiki and Birigen [5] developed feedback-forcing methods with two free parameters (called virtual boundary methods) in conjunction with the spectral and finite difference methods, respectively.

In this model [4], the force density term (or acceleration term) is determined by straight feedback of velocity information, i.e., time integration of the velocity difference between calculated velocity and desired velocity (production) and the velocity difference itself (damping) as follows:

$$\mathbf{F} = \alpha \int_0^t (\mathbf{u} - \mathbf{U}^d) d\tau + \beta (\mathbf{u} - \mathbf{U}^d) \tag{1}$$

where $\mathbf{U}^d$ is the desired velocity and α and β are two free parameters to be tuned depending on the flow conditions. It is noted that Peskin's method [3] can also be regarded as the virtual boundary method with one free parameter. In terms of practical applications, the feedback-forcing method introduces one or two free parameters that need to be tuned according to the flow conditions and this, especially for unsteady flows, causes a time step limitation that reduces efficiency. In addition, regarding the use of discrete delta functions, the boundary forces are spread across the boundary, which diffuses over the grid, thus decreasing the accuracy of the solution.

b. Direct-forcing IBM

Direct-forcing IBM with sharp interface schemes

The *direct-forcing IBM* was originally derived by Mohd-Yusof [7] in a spectral context with a *sharp interface scheme*. Fadlun et al. [8] applied the direct-forcing method to various flow problems in the frame of the finite-difference method. The forcing point was located on the interior node (fluid node) closest to the boundary. The boundary force on the forcing point was evaluated by linear interpolation from the boundary and fluid velocities in an arbitrary direction. Kim et al. [9] derived the direct-forcing IBM in the finite-volume method. They introduced the mass source/sink, as well as momentum forcing to satisfy, not only the no-slip boundary condition on the immersed boundary, but also the continuity for the cell containing the immersed boundary. The forcing node was located on the exterior node (solid node) closest to the boundary. To remove the arbitrariness in the interpolation direction, as in [8], they proposed a consistent second-order interpolation scheme based on a bilinear interpolation, which is reduced to a one-dimensional linear interpolation when there are no available points except another forcing point near the boundary.

In the direct-forcing method, the force density (or acceleration) term is naturally determined in the calculation process. In other words, Navier-Stokes equation can be expressed as:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t}=\mathbf{RHS}^{n}+\mathbf{F}^{n} \tag{2}$$

where Δt is a time step, n and n+1 are current and next time steps, and RHS^n includes

convective, viscous, and pressure terms. If the desired velocity at the next time step, $\mathbf{U}^d$, is given, then Equation (2) becomes

$$\frac{\mathbf{U}^d - \mathbf{u}^n}{\Delta t} = \mathbf{RHS}^n + \mathbf{F}^n . \tag{3}$$

Equation (3) can be rewritten in terms of the next-time-step velocity without being forced, $\mathbf{u}^{noF}$ (which is originally calculated regardless of existence of the forcing term), as

$$\mathbf{U}^d = \mathbf{u}^{noF} + \mathbf{F}^n \Delta t \tag{4}$$

with

$$\mathbf{u}^{noF} = \mathbf{u}^n + \mathbf{RHS}^n \Delta t \cdot \tag{5}$$

From Equations (3) and (5), the force density (or acceleration) term can be directly expressed as:

$$\mathbf{F}^n = \frac{\mathbf{U}^d - \mathbf{u}^{noF}}{\Delta t} . \tag{6}$$

It should be pointed out that in the direct-forcing method with a sharp interface scheme, the boundary force density term can be added either implicitly or explicitly, mainly depending on the adopted time-advancement scheme. In the explicit time-advancement scheme, as in [7-8], the boundary force density term is not involved explicitly in the actual calculation. Instead, the velocity at the forcing node is directly replaced by the desired velocity. In contrast, in the semi-implicit time advancement scheme, as in [9-10], the force density term is evaluated at the predictor step because the change of force after the predictor step is sufficiently small not to deteriorate the entire

order of accuracy; it is then incorporated into the remaining flow equation calculations explicitly.

Regarding the location of the forcing points in the direct-forcing method with a sharp interface scheme, either interior (in fluid) nodes as in [8] or exterior (in solid) nodes as in [7, 9] can be selected. Actually, many of the following studies adopted one of these methods. Balaras [10], Gilmanov et al. [11], Choi et al. [12], Ikeno and Kajishima [13] used interior forcing nodes. Majumdar et al. [14], Iaccarino and Verzicco [15], Tseng and Ferziger [16], Ghias et al. [17], and Shen et al. [18] used exterior forcing nodes. All these authors proposed their own systematic interpolation algorithms and successfully applied them to various problems, even turbulent flow problems.

Compared with the feedback-forcing method, in the direct-forcing method with a sharp interface scheme, the forcing does not affect the stability or require force smoothing, and no free parameters to be adjusted are used. However, for moving boundary problems, the direct-forcing IBM with a sharp interface scheme may cause spurious oscillations due to discontinuous changes of interpolation points and freshly cleared cells, which stand for the nodes in the solid region at the previous time step and in the fluid region at the new time step [22, 28].

Direct-forcing IBM with diffuse interface schemes

In the context of the direct-forcing method, a *diffuse interface scheme* was first adopted by Silva et al. [22]. They used a second-order Lagrange polynomial approximation to calculate the pressure and velocity derivatives, which are needed to evaluate the force on

the boundary points. The boundary force is distributed onto the forcing points through the discrete delta function, which was used in the feedback-forcing method. Uhlmann [23] applied the diffuse direct-forcing method to 3D particulate flows. In the IBM, the discrete delta function was used for velocity interpolation on Lagrangian forcing points, as well as for boundary force distribution. In other words, instead of interpolating each term of the Navier-Stokes equation on Lagrangian forcing points to evaluate the boundary force, as in [22], Uhlmann used the unforced velocities on the neighboring nodes in the interpolation process, thereby obtaining the force more easily than in [22]. He also showed that the diffuse direct-forcing method provides smooth solutions for moving particle problems compared with sharp forcing methods, where spurious oscillation occurs.

In the direct-forcing method with a diffuse interface scheme, the velocity field used to evaluate the boundary force is reconstructed by the boundary force again. Thus, the forcing-point velocity interpolated from reconstructed velocities may not satisfy the no-slip condition exactly. To ensure this, several direct-forcing methods with an *implicit diffuse scheme* were proposed. We call previous diffuse schemes, which are not implicit, *explicit diffuse schemes*. Su et al. [24] and Le et al. [25] proposed implicit forcing methods to solve implicit banded force matrix equations. To avoid the complicated calculation of the matrix equations, Luo et al. [26] and Wang et al.[19] proposed multi-direct-forcing, which iterates the procedure of forcing and spreading until a given criterion is satisfied, and applied it to fixed and moving particle problems.

In summary, the direct-forcing IBM consists of the direct-forcing formula and the interface scheme. The direct-forcing formula can change depending on the numerical schemes, especially time advancement schemes. The interface scheme can be classified into the sharp interface scheme and the explicit and implicit diffuse schemes. Figure 1 depicts the structure of direct-forcing immersed boundary method.

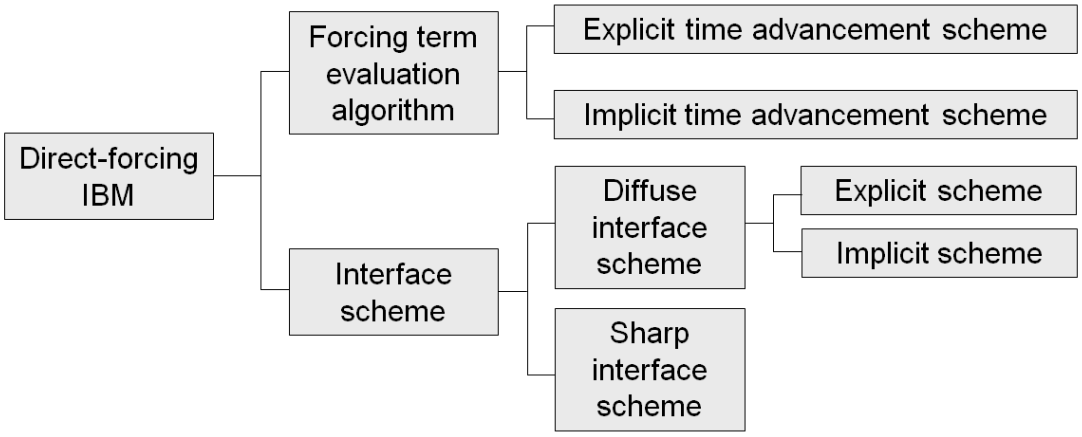

Figure 1. Structure and classification of the direct-forcing IBM.

2. Immersed boundary method for energy equation

In recent years, the concept of forcing in momentum equations under the IBM has been also extended to the energy equation to satisfy the thermal boundary conditions [29-32]. Kim and Choi [29] applied their exterior sharp direct-forcing scheme [9] for the momentum forcing in momentum equation to the energy equation for energy forcing (or heat source/sink). They simulated forced or mixed convection around hot circular cylinders and obtained comparable results with other experiments and calculations.

Pacheco et al. [30] used another exterior sharp direct-forcing scheme to successfully simulate 2D natural convection problems with stationary boundaries. Gilmanov and Acharya [33] used their interior sharp direct-forcing scheme [11] for energy forcing to simulate not only flow past a hot rigid sphere, but also deformable hot spherical particle sedimentation by coupling the material point method (MPM). Zhang et al. [32] used their diffuse direct-forcing scheme [34] to simulate the convection with flows over stationary and oscillating cylinders, respectively. Feng and Michaelides [31, 35] used the explicit diffuse direct (momentum and energy) forcing scheme to successfully simulate various cases of particle sedimentation. In this study, we call the forcing terms in the momentum equations for no-slip boundary conditions "momentum-forcing" and the energy source term in the energy equation for thermal boundary conditions "energy-forcing" [30]. The principles of forcing term evaluation in both momentum and energy equations are basically the same; the only difference is that the no-slip boundary condition is only considered in the momentum-forcing IBM, while more various boundary conditions, such as Dirichlet and Neumann types, can be involved in the energy-forcing IBM.

3. Immersed boundary-lattice Boltzmann method

Due to its simplicity and efficiency, the lattice Boltzmann method (LBM) has been broadly used to simulate complex flows as an alternative to the Navier-Stokes methods [36-37]. The lattice Boltzmann equation (LBE) is a kinetic equation of particle distribution functions (PDF) discretized on the Cartesian grid. At each node, taking

moments of PDFs provides hydrodynamic variables, such as density and velocity. The common feature of using the Cartesian grid motivates the coupling of the LBM and the IBM, which is called the immersed boundary-lattice Boltzmann method (IB-LBM). By replacing the Navier-Stokes equation with the lattice Boltzmann equation for flow field calculations, almost the same discussion as in the IBM based on Navier-Stokes equation can be applied to the IB-LBM.

Feng and Michaelides [6] first proposed the IB-LBM. Their IB-LBM is basically the same as the feedback-forcing IBM used by Lai and Peskin [3], with the exception that it solves the LBE instead of the NSE for fluid flows. Then, they proposed a direct-forcing IB-LBM with an explicit diffuse interface scheme [20] to solve 3D particulate flow problems. However, in their direct-forcing IB-LBMs, they used the NSE for the evaluation of boundary forces. Niu et al. [38] proposed an IB-LBM with an explicit diffuse interface scheme, which is called the momentum-exchange-based IB-LBM. In this method, instead of solving Navier-Stokes equations for the boundary force evaluation, they used the bounce-back rule, which is used in the LBM for wall boundary conditions but is not second-order when applied on nodes [39], as in this IB-LBM. Dupuis et al. [21] proposed a direct-forcing IB-LBM without solving the NSE for the evaluation of boundary force density. Their method can be said to be a pure direct-forcing IB-LBM since they used the LBE to evaluate the boundary force density as well as to solve the fluid flow. They tested an explicit diffuse and an interior sharp schemes through the simulation of flow past an impulsively started cylinder. However, the direct-forcing IB-LBMs [20-21] neglected the kinetic nature of the lattice Boltzmann method

and used the lumped-forcing LBE, which does not recover the NSE with second-order accuracy. These issues will be discussed in greater detail in Chapters II and III.

In contrast, some authors [40-41] proposed direct-forcing IB-LBMs based on the split-forcing LBE proposed by Guo et al. [42]. The adoption of the split-forcing LBE gives not only recovery of the NSE with second-order accuracy but also more accurate interface results. Kang and Hassan [40] used an explicit diffuse interface scheme to simulate stationary and moving particle flows, while Wu and Shu [41] used an implicit diffuse interface scheme solving banded matrix equations, as in Su et al. [24] and Le et al. [25], for complex boundary flows.

B. Motivation

In terms of the accurate, efficient treatment of complex and moving boundaries, coupling the direct-forcing concept and various interface schemes developed in the NSE method with the LBM with simplicity and efficiency is very promising. As noticed in the direct-forcing IBMs based on the NSE, different interface schemes have been used depending on the purpose. For stationary complex boundary problems, the sharp interface schemes were mainly used to achieve the greater accuracy in results. Their applications were extended even to turbulent flow problems [13-16]. For moving boundary problems, although the diffuse interface schemes have diffuse (less accurate) solutions, they were usually used because they relieve or remove the spurious oscillations [23]. Thus, for the applications of the direct-forcing IB-LBM based on the split-forcing LBE to various problems, we need to assess systematically the various

interface schemes, including the sharp interface scheme as well as the diffuse interface scheme. In addition, to cover a broader range of thermal hydraulic problems including the non-thermal flows, the extension of the IB-LBM to energy equations is needed.

C. Objectives

The objective of this dissertation is to develop and assess the direct-forcing immersed boundary-lattice Boltzmann method (IB-LBM), based on the split-forcing lattice Boltzmann equation models, applicable to 2D/3D isothermal flow problems with stationary or moving boundaries and also to extend the isothermal IB-LBMs to thermal IB-LBMs for covering the non-isothermal flows. The focus areas are as follows:

- Application of the IB-LBMs to complex, stationary boundary problems;
- Application of the IB-LBMs to moving boundary problems covering solid-fluid two-phase flows;
- Extension of the IB-LBM to the energy equation for the thermal flow simulation.

To better understand these topics, we start with discussing the lattice Boltzmann equation models, which are adopted as governing equations for the simulation of fluid flows and heat transfers in this study.

CHAPTER II

LATTICE BOLTZMANN METHOD

Recently, the lattice Boltzmann method (LBM) has been successfully applied to simulate fluid flows and transport phenomena [36]. In contrast to conventional CFD methods using the Navier-Stokes equation (NSE) based on macroscopic continuum theory, the LBM is based on mesoscopic kinetic equations, in which the collective behavior of particles is adopted to simulate the continuum mechanics of the system [43]. In this chapter, we discuss basic theories of the lattice Boltzmann method needed for the application of the immersed boundary method (IBM), which cover the lattice Boltzmann equation (LBE) and thermal lattice Boltzmann equation (TLBE). The details of the IB-LBM and its applications are presented in the following chapters.

A. Lattice Boltzmann Equation

1. Single-relaxation-time lattice Boltzmann equation

The lattice Boltzmann equation with a single-relaxation-time (SRT-LBE) [44-45] is usually adopted in the LBM. Historically, the SRT-LBE evolved empirically from the lattice gas automata (LGA) [46-47], which is a discrete particle kinetics utilizing a discrete lattice and discrete time, through overcoming some serious drawbacks of the LGA, such as large statistical noise, limited range of physical parameters, non-Galilean invariance, and implementation difficulty in three dimensions. However, it was shown later that the SRT-LBE can be directly derived from the continuous Boltzmann equation

through discretization in both time and phase space [48], and the equilibriums and collision matrices can be constructed independently to better suit numerical applications [49].

a. Boltzmann BGK equation

Let **x** be Cartesian coordinates of a physical space and $\boldsymbol{\xi}$ the molecular velocity. Then the continuous Boltzmann equation of the particle distribution function $f(\mathbf{x},\boldsymbol{\xi},t)$ can be written as

$$\frac{\partial f}{\partial t}+\boldsymbol{\xi}\cdot\nabla f = J(f) \tag{7}$$

where $J(f)$ is the collision operator and models the rate of change of the particle distribution function f due to molecular collisions. This collision operator has a complex integral form, so Equation (7) is a nonlinear integro-differential equation.

The simple single-relaxation-time (SRT) model used in kinetic theory is also called the BGK model, named after Bhatnager, Gross, and Krook, who introduced the model in a paper published in 1954 [50]. In this model, the collision term $J(f)$ in the Boltzmann equation is simplified as

$$J(f) = -\frac{1}{\lambda}(f - f^{(0)}). \tag{8}$$

Here, the Maxwellian distribution function $f^{(0)}$ is expressed in terms of the local mean velocity and temperature and λ is the mean relaxation time, which may depend on temperature but not on molecular velocity. Therefore, the Boltzmann BGK equation can be expressed as

$$\frac{\partial f}{\partial t}+\boldsymbol{\xi}\cdot\nabla f=-\frac{1}{\lambda}(f-f^{(0)}). \tag{9}$$

b. Single-relaxation-time lattice Boltzmann equation

The SRT-LBE can be directly derived from the Boltzmann BGK equation (9). Integrating Equation (9) along characteristics and then performing Taylor-series expansion to the first-order in time, we can obtain

$$f(\mathbf{x}+\boldsymbol{\xi}\Delta t,\boldsymbol{\xi},t+\Delta t)-f(\mathbf{x},\boldsymbol{\xi},t)=-\frac{1}{\tau}[f(\mathbf{x},\boldsymbol{\xi},t)-f_M(\mathbf{x},\boldsymbol{\xi},t)] \tag{10}$$

where $\tau=\lambda/\Delta t$ is the dimensionless mean relaxation time and Δt is the discretized time step. Discretizing Equation (10) in the velocity space, we can obtain the following simple SRT-LBE for incompressible flow:

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t,t+\Delta t)-f_\alpha(\mathbf{x},t)=-\frac{1}{\tau}[f_\alpha(\mathbf{x},t)-f_\alpha^{(eq)}(\mathbf{x},t)] \tag{11}$$

where $f_\alpha(\mathbf{x},t)\equiv f(\mathbf{x},\mathbf{e}_\alpha,t)$ is the discretized particle distribution function (PDF), $f_\alpha^{(eq)}(\mathbf{x},t)$ is the discretized equilibrium PDF, and $\{\mathbf{e}_\alpha\}$ is the discrete velocity set. The equilibrium distribution function and the discrete velocity set change depending on the lattice model selected. The equilibrium PDF, $f_\alpha^{(eq)}$, is obtained by using the Taylor series expansion of the Maxwell-Boltzmann distribution function with velocity **u** up to second order, and it can be expressed as

$$f_\alpha^{(eq)}=w_\alpha\rho\left[1+\frac{3}{c^2}(\mathbf{e}_\alpha\cdot\mathbf{u})+\frac{9}{2c^4}(\mathbf{e}_\alpha\cdot\mathbf{u})^2-\frac{3}{2c^2}\mathbf{u}^2\right] \tag{12}$$

where the lattice speed c=Δx/Δt, and Δx and Δt are the lattice size and the time step size,

respectively. The weighting coefficient, w_α, depends on the discrete velocity set, $\{\mathbf{e}_\alpha\}$. Although there are various discrete velocity sets, we use the 9-velocity model on a 2D square lattice, denoted as the D2Q9 model, and the 19-velocity model on a 3D cubic lattice, denoted as the D3Q19 model, which have been shown to have better performance than other models [51]. In D2Q9 and D3Q19 models, discrete velocity vectors are defined by

$$\mathbf{e}_\alpha = \begin{cases} (0,0), & \alpha = 0 \\ c(\pm 1,0), c(0,\pm 1), & \alpha = 1,2,3,4 \\ c(\pm 1,\pm 1), & \alpha = 5,6,7,8 \end{cases} \tag{13a}$$

and

$$\mathbf{e}_\alpha = \begin{cases} (0,0,0), & \alpha = 0 \\ c(\pm 1,0,0), c(0,\pm 1,0), c(0,0,\pm 1), & \alpha = 1,2,\cdots,6 \\ c(\pm 1,\pm 1,0), c(\pm 1,0,\pm 1), c(0,\pm 1,\pm 1), & \alpha = 7,8,\cdots,18 \end{cases}, \tag{13b}$$

respectively, and the corresponding weighting coefficients w_α are

$$w_\alpha = \begin{cases} 4/9, & \alpha = 0 \\ 1/9, & \alpha = 1,2,3,4 \\ 1/36, & \alpha = 5,6,7,8 \end{cases} \tag{14a}$$

and

$$w_\alpha = \begin{cases} 1/3, & \alpha = 0 \\ 1/18, & \alpha = 1,2,\cdots,6 \\ 1/36, & \alpha = 7,8,\cdots,18 \end{cases}, \tag{14b}$$

respectively. Figure 2 illustrates discrete velocity spaces (lattices) in the D2Q9 and D3Q19 models.

Applying the Chapman-Enskog multi-scale analysis [42], we can show that SRT-LBE (11) recovers the NSE. Here, density and velocity are defined by the 0^{th} and 1^{st} moments of PDFs, respectively:

$$\rho = \sum_{\alpha} f_{\alpha} = \sum_{\alpha} f_{\alpha}^{(eq)}, \tag{15}$$

$$\rho \mathbf{u} = \sum_{\alpha} \mathbf{e}_{\alpha} f_{\alpha} = \sum_{\alpha} \mathbf{e}_{\alpha} f_{\alpha}^{(eq)}. \tag{16}$$

The pressure, p, is determined by the following equation of state (EOS):

$$p = c_s^2 \rho \tag{17}$$

and the kinematic viscosity, ν, is determined by

$$\nu = \left(\tau - 1/2\right) c_s^2 \Delta t. \tag{18}$$

where c_s is the speed of sound and is related to the lattice speed, c by $c_s = c/\sqrt{3}$.

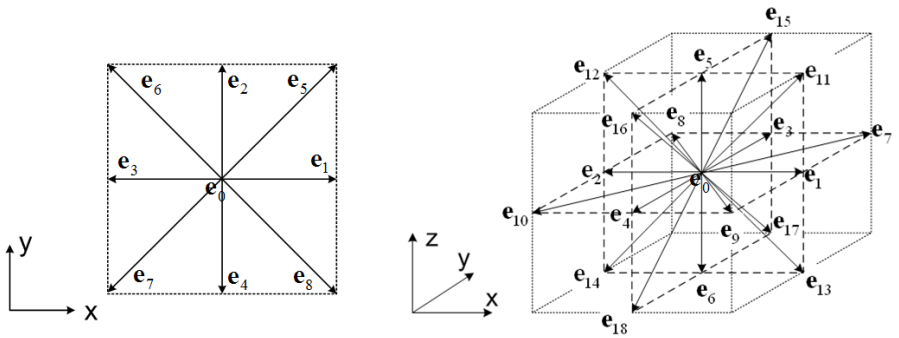

Figure 2. Discrete velocity spaces (lattices) in (a) the D2Q9 and (b) the D3Q19 models.

c. LBM algorithm

The SRT-LBE (12) can be simply implemented in numerical calculation as the following two steps.

Collision step: $f'_\alpha(\mathbf{x},t) = f_\alpha(\mathbf{x},t) - \frac{1}{\tau}[f_\alpha(\mathbf{x},t) - f_\alpha^{(eq)}(\mathbf{x},t)]$ (19a)

Streaming step: $f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f'_\alpha(\mathbf{x},t)$ (19b)

Here, f'_α is the post-collision PDF. It should be noted that in the streaming step, no arithmetic calculation is involved and only the data shifting occurs. Figure 3 depicts the LBM calculation algorithm.

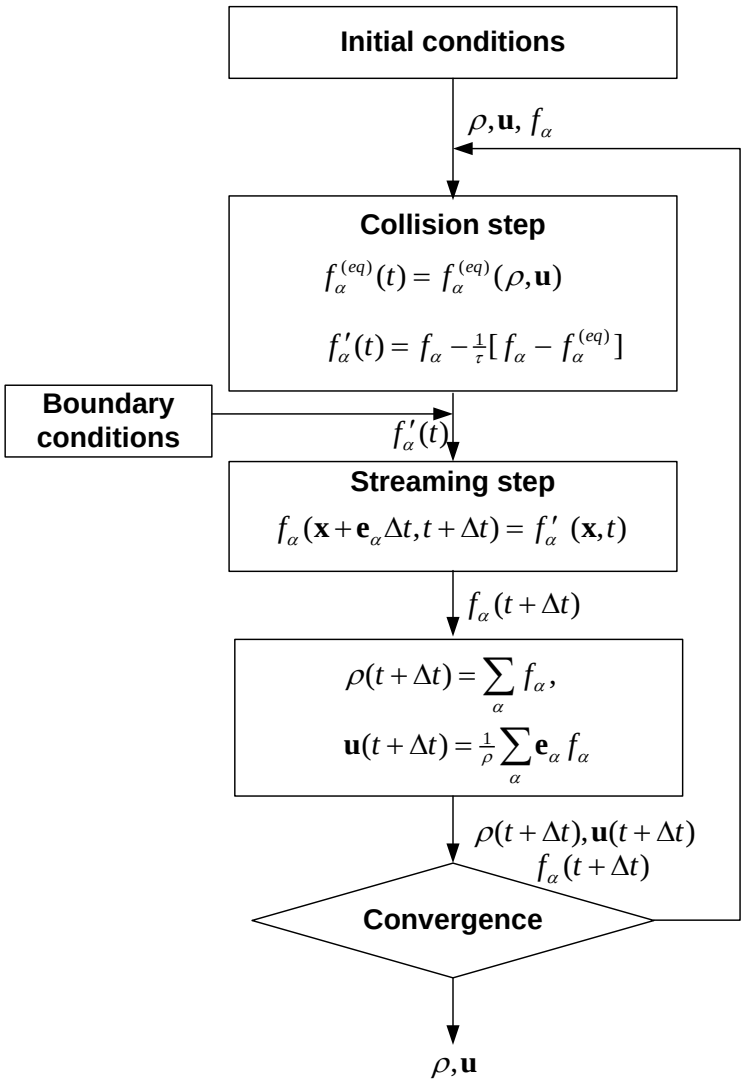

Figure 3. LBM calculation algorithm.

d. Characteristics of the LBM

The LBM based on the kinetic equation (Boltzmann equation) can be regarded as a mesoscopic approach because instead of considering each individual molecule at the microscopic level in molecular dynamics, it considers the fluid particles at mesocopic scale, which are made up of a group of molecules.

In microscopic and mesoscopic approaches, the macroscopic variables, such as density and velocity, can be obtained by taking moments of particle variables, such as particle distribution functions, as Equations (15) and (16) and the LBE can recover the Navier-Stokes equation (NSE) by multi-scale analysis. This is in contrast to the conventional macroscopic approach based on continuum fluid mechanics, in which macroscopic variables are directly obtained by solving differential or integral forms of the Navier-Stokes equation of the macroscopic variables. Figure 4 illustrates the relations between the lattice Boltzmann equation (LBE), the Boltzmann BGK equation, and the Navier-Stokes equation (NSE).

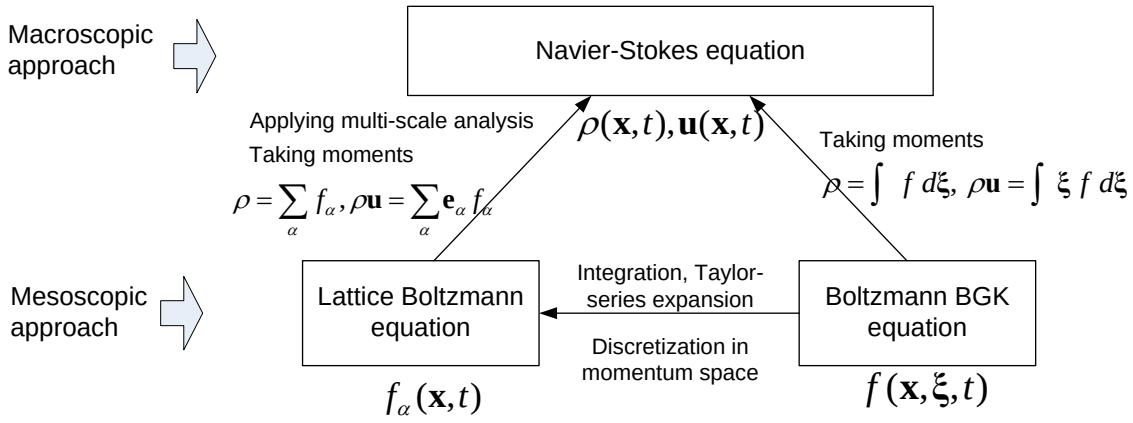

Figure 4. Relations between the LBE, the Boltzmann BGK equation, and the NSE.

The major characteristics of the LBE distinguished from the conventional CFD methods based on the NSE originate from the facts that (i) the LBE is based on the kinetic equation with a simple collision rule and (ii) the discretization of configuration space is defined by discretization of momentum (velocity) space. As a result, the LBE has the following advantages over the conventional CFD methods [37, 43]:

- The NSE method must deal with nonlinear convective terms, whereas in the LBE model the streaming process corresponding to the convective term is linear in velocity space and is handled by simple advection along the constant streamline, i.e., uniform data shifting. It should be noted that although the collision step in the LBM includes non-linear terms, it is local, thus not requiring complex calculations involving neighboring information.
- Since the incompressible Navier-Stokes equations can be obtained in the nearly incompressible limit in the LBM, pressure is explicitly obtained through an equation of state, and data communication is always local in the LBM. In contrast, for incompressible flows, the NSE method usually employs expensive iterative procedures to solve the elliptic Poisson equation, which involves global communication of data. Hence, the LBM with low communication/computation ratio is efficient for parallel computing based on the Message Passing Interface (MPI).
- The LBM seeks the minimum set of velocities in phase spaces. Therefore, simple relations exist between particle density distribution functions and macroscopic

variables and mass and momentum conservations are guaranteed to machine round-off in computer implementation.

- The strain rate tensor can be evaluated locally. Like pressure, the strain rate tensor is directly obtained from the moment of the non-equilibrium PDFs without additional communications with neighbor nodes. This aspect in the LBM is beneficial to eddy-viscosity-type subgrid scale models in large eddy simulation (LES) and non-Newtonian fluid modeling requiring the evaluation of the strain rate tensor.
- The LBE consists of simple arithmetic calculations, so it is easy to program.
- Due to the kinetic nature of the Boltzmann equation, the physics associated with the molecular level interaction can be incorporated more easily in the LBE model.
- The LBE is ideally suited for handling multi-phase flow with phase transition and multi-species mixtures where diffusivity is important. NSE solvers can be computationally too expensive for these flows.

These characteristics configure the LBE as a special finite-difference scheme for fluid dynamics and enable the LBE to show impressive growth as an alternative numerical technique for complex fluid flow problems.

2. Multiple-relaxation-time lattice Boltzmann equation

Although the SRT-LBE is the most popular in the LBM calculation due to its simplicity [44-45], the SRT-LBE has limitations due to its numerical instability [52] and inaccuracy in boundary conditions [53]. Most of these limitations in the SRT-LBE can

be resolved by using the multiple-relaxation-time lattice Boltzmann equation (MRT-LBE) [37, 52, 54-55]. The MRT-LBE has marked advantages over the SRT-LBE. For a given resolution, the MRT-LBE is significantly more stable numerically and more accurate for problems with anisotropy, with an insignificant additional computational overhead; this allows access to a greater range of problems, particularly at higher Reynolds numbers [56].

In the MRT model, different moments of the distribution function relax at different rates, while in SRT model, all moments relax at the same rate [57]. Specifically, the MRT-LBE adopts an equivalent representation of particle distribution functions in terms of their moments, including various hydrodynamic fields such as density, mass flux, and stress tensor. Therefore, the relaxation process due to collision can more naturally be described in terms of a space spanned by such moments, which can, in general, relax at different rates [52, 54]. By carefully separating the time scales of various hydrodynamic and kinetic modes through a linear stability analysis, the numerical stability of the MRT-LBE can be significantly improved when compared with the SRT-LBE [52, 58].

a. D3Q19 MRT-LBE

Since we use the MRT-LBE only for 3D flow simulations in the present study, we focus on the D3Q19 model hereafter. To discuss the MRT-LBE, we start with rewriting the SRT-LBE in a vector form:

$$\mathbf{f}(\mathbf{x}+\mathbf{e}_\alpha \Delta t, t) - \mathbf{f}(\mathbf{x}, t) = -\mathbf{S}[\mathbf{f}(\mathbf{x}, t) - \mathbf{f}^{(eq)}(\mathbf{x}, t)] \tag{20}$$

with a particle distribution vector $\mathbf{f} = (f_0, f_1, \cdots, f_{18})^T$, an equilibrium particle distribution vector $\mathbf{f}^{(eq)} = (f_0^{(eq)}, f_1^{(eq)}, \cdots, f_{18}^{(eq)})^T$, and a collision matrix $\mathbf{S} = (1/\tau)\mathbf{I}$. In the MRT-LBE, we consider the moment vector $\hat{\mathbf{f}} = (\hat{f}_0, \hat{f}_1, \cdots, \hat{f}_{18})^T$ with the relation of $\hat{\mathbf{f}} = \mathbf{M}\mathbf{f}$ where $\mathbf{M}$ is the transformation matrix from the particle velocity space to the moment space. Here, the elements of $\mathbf{M}$ are obtained in a suitable orthogonal basis as combinations of monomials of the Cartesian components of the particle velocity $\mathbf{e}_\alpha$ through the standard Gram-Schmidt procedure [54]. The resulting transformation matrix $\mathbf{M}$ in D3Q19 model is given as

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-30 & -11 & -11 & -11 & -11 & -11 & -11 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
12 & -4 & -4 & -4 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & -4 & 4 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & -4 & 4 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 4 & -4 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
0 & 2 & 2 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\
0 & -4 & -4 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\
0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -2 & -2 & 2 & 2 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1
\end{bmatrix}.$$

The resulting 19 moments in the moment vector are arranged as follows:

$$\hat{\mathbf{f}} = (\rho, e, e^2, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{xz}, m_x, m_y, m_z)^T \qquad (21)$$

Here, ρ is the density fluctuation, e and e^2 are related to energy and its square respectively, j_x, j_y, j_z are components of the momentum, q_x, q_y, q_z are related to components of the heat flux, p_{xx}, p_{ww} and p_{xy}, p_{yz}, p_{zx} are related to the components of the symmetric and traceless strain-rate tensors, π_{xx}, π_{ww} are the fourth-order moments, and m_x, m_y, m_z are the third-order moments [54].

The MRT-LBE computes the collision in moment space, while the streaming step is performed in the usual particle velocity space [54]. The MRT-LBE can be expressed as

$$\mathbf{f}(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t)-\mathbf{f}(\mathbf{x},t)=-\mathbf{M}^{-1}\hat{\mathbf{S}}[\hat{\mathbf{f}}-\hat{\mathbf{f}}^{(eq)}]. \tag{22}$$

Here, the equilibrium moments $\hat{f}_\alpha^{(eq)}$ are the function of the conserved moments, i.e., $\hat{f}_0^{(eq)}=\rho$, $\hat{f}_3^{(eq)}=j_x$, $\hat{f}_5^{(eq)}=j_y$, $\hat{f}_7^{(eq)}=j_z$ and are given as [58]

$$\begin{aligned}
&\hat{f}_1^{(eq)}=-11\rho+19\frac{\mathbf{j}\cdot\mathbf{j}}{\rho},\ \hat{f}_2^{(eq)}=3\rho-\frac{11}{2}\frac{\mathbf{j}\cdot\mathbf{j}}{\rho},\ \hat{f}_4^{(eq)}=-\frac{2}{3}j_x,\ \hat{f}_6^{(eq)}=-\frac{2}{3}j_y,\\
&\hat{f}_8^{(eq)}=-\frac{2}{3}j_z,\ \hat{f}_9^{(eq)}=\frac{1}{\rho}(3j_x^2-\mathbf{j}\cdot\mathbf{j}),\ \hat{f}_{10}^{(eq)}=-\frac{1}{2}\hat{f}_9^{(eq)},\ \hat{f}_{11}^{(eq)}=\frac{1}{\rho}(j_y^2-j_z^2),\\
&\hat{f}_{12}^{(eq)}=-\frac{1}{2}\hat{f}_{11}^{(eq)},\ \hat{f}_{13}^{(eq)}=\frac{1}{\rho}j_xj_y,\ \hat{f}_{14}^{(eq)}=\frac{1}{\rho}j_yj_z,\ \hat{f}_{15}^{(eq)}=\frac{1}{\rho}j_xj_z,\ \hat{f}_{16,17,18}^{(eq)}=0.
\end{aligned} \tag{23}$$

Also, $\hat{\mathbf{S}}=diag(s_0,s_1,\cdots,s_{18})$ is the diagonal collision matrix in moment space, and its elements $s_0,s_1,\cdots,s_{18}$ are relaxation times for the respective moments. The transport properties of the fluid flow, such as bulk and kinematic viscosities, can be related to the appropriate relaxation times through either Chapman-Enskog analysis of the MRT-LBE or the von Neumann stability analysis of its linearization version [59]. The kinematic viscosity ν and the bulk viscosity ζ of the model are

$$\nu = \frac{1}{3}\left(\frac{1}{s_\alpha} - \frac{1}{2}\right)\Delta t, \ \alpha = 9,11,13,14, \text{and } 15 \tag{24}$$

$$\zeta = \frac{2}{9}\left(\frac{1}{s_1} - \frac{1}{2}\right)\Delta t. \tag{25}$$

Based on linear stability analysis [52], the relaxation parameters are determined as follows [54]:

$$s_1 = 1.19, \ s_2 = s_{10} = s_{12} = 1.4, \ s_4 = s_6 = s_8 = 1.2, \ s_{16} = s_{17} = s_{18} = 1.98. \tag{26}$$

b. MRT-LBE calculation algorithm

Numerical implementation of the MRT-LBE is also very simple as in the SRT-LBE. The difference from the SRT-LBE is that in collision step the MRT-LBE involves additional matrix multiplications for the calculation of the relaxation in moment space. The streaming step is the same as that of the SRT-LBM. In other words,

Collision step:

$$f'_\alpha(\mathbf{x},t) = \begin{cases} f_\alpha(\mathbf{x},t) - [f_\alpha(\mathbf{x},t) - f_\alpha^{(eq)}(\mathbf{x},t)]/\tau & \text{for the SRT-LBM} \\ f_\alpha(\mathbf{x},t) - \{\mathbf{M}^{-1}\hat{\mathbf{S}}[\hat{\mathbf{f}}(\mathbf{x},t) - \hat{\mathbf{f}}^{(eq)}(\mathbf{x},t)]\}_\alpha & \text{for the MRT-LBM} \end{cases} \tag{27a}$$

Streaming step:

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f'_\alpha(\mathbf{x},t). \tag{27b}$$

B. Thermal Lattice Boltzmann Equation

To deal with thermal flows in the lattice Boltzmann equation framework, several thermal lattice Boltzmann models (TLBM) have been developed. The TLBMs can be classified

into three categories: the single-population model, the double-population model, and the hybrid model. In the single-population model (or multi-speed model), only the particle distribution function was used, but additional discrete velocities were introduced to obtain the energy equation, and the equilibrium distribution functions usually include higher order velocity terms [60-61]. However, the multi-speed type single-population models were found to suffer from severe numerical instability, and the range of temperature variation was limited [62].

On the other hand, in the double-population models, distribution functions for temperature (or internal energy) were introduced in addition to the original density-distribution function, so that the athermal LBE of density distribution functions for momentum and the thermal LBE of temperature (or internal energy) distribution functions for energy were separately solved. As a result, this kind of model could effectively overcome two limitations of the multi-speed models, namely, severe numerical instability and narrow range of temperature variation [63]. The correct double-population model, including terms of the viscous heat dissipation and the compression work done by pressure, was derived by He et al. [63]. In this thermal model, the density distribution function and the energy distribution function satisfy the following equations, respectively:

$$\bar{f}_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t)-\bar{f}_\alpha(\mathbf{x},t)=-\frac{\Delta t}{\tau_f+0.5\Delta t}\left[\bar{f}_\alpha(x,t)-f_\alpha^{(eq)}(x,t)\right]+\frac{\tau_f F_\alpha \Delta t}{\tau_f+0.5\Delta t}, \quad (28a)$$

$$\bar{g}_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t)-\bar{g}_\alpha(\mathbf{x},t)=-\frac{\Delta t}{\tau_g+0.5\Delta t}\left[\bar{g}_\alpha(x,t)-g_\alpha^{(eq)}(x,t)\right]-\frac{\tau_g f_\alpha(\mathbf{x},t) q_\alpha \Delta t}{\tau_g+0.5\Delta t}, \quad (28b)$$

where two new variables, $\bar{f}$ and $\bar{g}$, are used to keep the consistency of the viscosity as well as to keep the scheme explicit and defined as

$$\bar{f}_\alpha = f_\alpha + \frac{\Delta t}{2\tau_f}(f_\alpha - f_\alpha^{(eq)}) - \frac{\Delta t}{2}F_\alpha, \tag{29a}$$

$$\bar{g}_\alpha = g_\alpha + \frac{\Delta t}{2\tau_g}(g_\alpha - g_\alpha^{(eq)}) + \frac{\Delta t}{2}f_\alpha q_\alpha, \tag{29b}$$

respectively. The other variables are defined as

$$F_\alpha = \frac{\mathbf{G}\cdot(\mathbf{e}_\alpha - \mathbf{u})}{RT} f_\alpha^{eq} = \frac{\mathbf{G}\cdot(\mathbf{e}_\alpha - \mathbf{u})}{c_s^2} f_\alpha^{eq}, \tag{30}$$

$$q_\alpha = (\mathbf{e}_\alpha - \mathbf{u})\cdot\left[\frac{1}{\rho}(-\nabla p + \nabla\cdot\mathbf{\Pi}) + (\mathbf{e}_\alpha - \mathbf{u})\cdot\nabla\mathbf{u}\right] = (\mathbf{e}_\alpha - \mathbf{u})\cdot\left[\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{e}_\alpha\cdot\nabla)\mathbf{u}\right], \tag{31}$$

$$\mathbf{\Pi} = \rho\nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T). \tag{32}$$

Here, **G** is the external force acting on the unit mass, τ_f and τ_g are relaxation times for isothermal and thermal LBEs, respectively, and $g^{(eq)}$ is the equilibrium energy distribution function. In this double population model, the macroscopic variables are determined by

$$\rho = \sum_\alpha \bar{f}_\alpha, \tag{33}$$

$$\rho\mathbf{u} = \sum_\alpha \mathbf{e}_\alpha \bar{f}_\alpha + \frac{\rho\mathbf{G}}{2}\Delta t, \tag{34}$$

$$\rho e = \sum_\alpha \bar{g}_\alpha + \frac{\Delta t}{2}\sum_\alpha f_\alpha q_\alpha \tag{35}$$

where *e* is the internal energy, and viscosity and diffusivity are determined by

$$\nu = \tau_f RT_0 = \frac{\tau_f c^2}{3}, \tag{36}$$

$$\alpha = 2\tau_g RT_0 = \frac{2}{3}\tau_g c^2 . \tag{37}$$

However, this method is too complicated to use because it contains a complex gradient term $f_\alpha q_\alpha$ in the thermal LBE. Hence, several double-population models with a simplified thermal LBE have been proposed [64-66] and applied to various heat transfer problems successfully [67-71]. The simplified thermal LBEs corresponded to the energy equation without terms of the viscous dissipation and compression work done by pressure, i.e., advection-diffusion equation of temperature. Among them, the simplified TLBE proposed by Peng et al. [64] is adopted in the present study. This simplified thermal model is based on the assumption that in real incompressible applications, the compressible work done by the pressure and the viscous heat dissipation are negligible. In the model, the complicated gradient term is discarded in the simplified thermal model because the term in the original thermal energy distribution model is mainly used to recover the compressible work and the viscous heat dissipation. After this simplification, there is no viscous term in the evolution for the new density distribution function, so there is no need to introduce the new variables such as Equations (29a) and (29b) to keep the viscosity same for both governing equations. As a result, the complexity in the original thermal energy distribution model can be overcome [64].

The simplified TLBE can be expressed as [64]:

$$g_\alpha(x + \mathbf{e}_\alpha \Delta t, t + \Delta t) = g_\alpha(\mathbf{x}, t) - \frac{1}{\tau_g}[g_\alpha(\mathbf{x}, t) - g_\alpha^{(eq)}(\mathbf{x}, t)] \tag{38}$$

where g_α is the energy distribution function, and $g_\alpha^{(eq)}$ is the equilibrium energy distribution function, which is determined in D2Q9 model by:

$$g_\alpha^{(eq)} = \begin{cases} w_\alpha \rho e\left(-1.5\dfrac{u^2}{c^2}\right), & \alpha = 0 \\ w_\alpha \rho e\left(1.5 + 1.5\dfrac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c^2} + 4.5\dfrac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{c^4} - 1.5\dfrac{u^2}{c^2}\right), & \alpha = 1,2,3,4 \\ w_\alpha \rho e\left(3 + 6\dfrac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c^2} + 4.5\dfrac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{c^4} - 1.5\dfrac{u^2}{c^2}\right), & \alpha = 5,6,7,8 \end{cases} \tag{39}$$

where $e=RT$ with the gas constant R and $c^2=3RT_0$ with the mean temperature T_0. The energy distribution functions satisfy the following condition:

$$\sum_\alpha g_\alpha = \rho e . \tag{40}$$

Applying the Chapman-Enskog multi-scale analysis, we can show that Equation (38) recovers the following energy equation:

$$\frac{\partial}{\partial t}(\rho e) + \nabla \cdot (\rho e \mathbf{u}) = \chi \nabla^2 (\rho e) \tag{41}$$

where the thermal diffusivity χ is expressed as

$$\chi = \frac{2}{3}\left(\tau_g - \frac{1}{2}\right) c^2 \Delta t . \tag{42}$$

It should be noted that in Equations (38) and (41), the compressible work and the viscous heat dissipation terms are neglected.

Even double-population models with simplified TLBEs are still numerically inefficient because they utilize a full set of distribution functions to calculate the temperature, although a reduced set of distribution functions [65, 72] can slightly

improve the numerical inefficiency [55]. Thus, the hybrid TLBM [55, 73-74] was proposed in which the mass and momentum conservation are solved by the usual athermal LBE, while the advection-diffusion equation satisfied by the temperature is solved separately by a finite difference technique. The hybrid methods could effectively overcome both the instability of single-population models and the numerical inefficiency of double-population models.

CHAPTER III

DIRECT-FORCING IMMERSED BOUNDARY-LATTICE BOLTZMANN METHOD*

A. Introduction

In this chapter, we derive the direct-forcing formula based on a split-forcing lattice Boltzmann equation (LBE) and assess several interface schemes for stationary complex boundary flows under the derived direct-forcing formula.

We investigate not only the common diffuse interface schemes but also a sharp interface scheme. For the diffuse interface schemes, we consider the explicit diffuse interface scheme, as in [40], and the implicit diffuse scheme, as in [41] The differences of this study from these previous schemes are: (i) to clearly see the effect of boundary diffuseness on accuracy, we adopt both 2-point and 4-point discrete delta functions, which give second-order approximations, whereas previous studies [40-41] only used the cosine-type 4-point discrete delta function, which gives first-order approximation [75-76]; (ii) for the implicit diffuse interface scheme, we adopt the simple multi-direct-forcing scheme used under the IBM based on the NSE in [19, 26] to avoid solving the complicated banded matrix equations as in [41]. For a sharp interface scheme, we consider the exterior sharp interface scheme used in the direct-forcing IBM based on the

---

* Reproduced in part with permission from "A comparative study of direct-forcing immersed boundary-lattice Boltzmann methods for stationary complex boundaries" Shin K. Kang, Yassin A. Hassan, 2010, *Int. J. Numer. Meth. Fluids*. DOI: 10.1002/fld.2304,

NSE by Kim et al. [9], where forcing nodes are located on the exterior (solid) nodes. This is the first trial of application of a sharp interface scheme to a direct-forcing IB-LBM based on the split-forcing LBE.

It should be noted that in this chapter, we focus mainly on the stationary boundary problems as a first step. Assessment and applications of the present IB-LBM for moving boundary problems will be discussed in Chapter IV.

The organization of this chapter is as follows: In Section B, we explain not only explicit-type split-forcing SRT-LBE [42] but also explicit-type split-forcing MRT-LBE [58], which is important for the broad applications of the IB-LBM, especially for high-Reynolds-number flows, because the LBE with MRT can attain better stability than the LBE with SRT. Then, in Section C.1, we derive the direct-forcing IB-LBM based on the split-forcing LBEs. We also mention the previous direct-forcing IB-LBMs' implicit assumption that makes the boundary intrinsically diffuse. This is an important motivation to adopt the split-forcing LBE in the direct-forcing IBM; however, previous works [40-41] did not mention it. In section C.2, we account for the explicit and implicit diffuse and sharp interface schemes to be assessed in this study. In Section D.1, the accuracy of the direct-forcing IB-LBMs with each interface scheme is investigated through the Taylor-Green vortex, which has the analytical solutions. Then, in Section D.2, each scheme is applied to steady and unsteady flows over stationary circular cylinder; those flows have been tested by various numerical methods for the evaluation of accuracy. We also consider the laminar flow past a sphere in order to validate the present IBM included in D3Q19 MRT-LBE in Section D.3. In Section D.4, this IB-LBM

is applied to flow in a pebble channel, which is a preliminary numerical test to extend real flow in Pebble Bed Reactor (PBR). Finally, the summary and conclusions of this chapter are presented in Section E.

B. Lattice Boltzmann Equation with Forcing Term

To adopt the immersed boundary method in the LBM, we need the LBE with a forcing term. Therefore, in this section, we first explore the SRT- and MRT-LBEs with a forcing term, which keep the accuracy second-order.

The LBE adopted in most of previous IB-LBMs [6, 20-21] is a lumped-forcing LBE, in which the forcing term is simply added to the LBE (Equation (11)) without any changes. This lumped-forcing LBE enables us to directly derive the simple direct-forcing [21]. However, it was shown that this lumped-forcing LBE cannot recover the Navier-Stokes equation with a second-order accuracy for unsteady and non-uniform force required in the IBM [42]. Besides, as will be discussed in Section C.1, the derived direct-forcing formula has intrinsic diffuse properties. On the other hand, the split-forcing LBE, in which momentum needed for the equilibrium distribution function is first increased by the half force and then an explicit forcing term is added to the LBE, overcomes the deficiencies of the lumped-forcing LBE.

In the following subsections, we discuss the difference between the lumped-forcing and the split-forcing in detail and show that the split-forcing is more accurate in terms of the IBM. Then, we derive the direct-forcing formula based on split-forcing LBE and explain various interface schemes.

1. Single-relaxation-time lattice Boltzmann equation with forcing term

The lumped-forcing SRT-LBE can be expressed as the following explicit form [77-78]:

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f_\alpha(\mathbf{x},t) - \frac{1}{\tau}[f_\alpha(\mathbf{x},t) - f_\alpha^{(eq)}(\mathbf{x},t)] + F_\alpha(\mathbf{x},t)\Delta t \tag{43}$$

where the discrete force distribution function can be defined as

$$F_\alpha(\mathbf{x},t) = \frac{w_\alpha}{c_s^2}\mathbf{e}_\alpha \cdot \mathbf{F}(\mathbf{x},t) \tag{44a}$$

or

$$F_\alpha(\mathbf{x},t) = w_\alpha\left[3\frac{\mathbf{e}_\alpha - \mathbf{u}(\mathbf{x},t)}{c^2} + 9\frac{\mathbf{e}_\alpha\cdot\mathbf{u}(\mathbf{x},t)}{c^4}\mathbf{e}_\alpha\right]\cdot\mathbf{F}(\mathbf{x},t) \tag{44b}$$

both of which satisfy the following relations of the 0^{th} and the 1^{st} moments:

$$\sum_\alpha F_\alpha(\mathbf{x},t) = 0\,, \tag{45}$$

$$\sum_\alpha \mathbf{e}_\alpha F_\alpha(\mathbf{x},t) = \mathbf{F}(\mathbf{x},t)\,. \tag{46}$$

It should be noted that Equations (44a) and (44b) satisfy the following 2^{nd} moment relations [78]:

$$\sum_\alpha \mathbf{e}_\alpha\mathbf{e}_\alpha F_\alpha(\mathbf{x},t) = 0\,, \tag{47a}$$

$$\sum_\alpha \mathbf{e}_\alpha\mathbf{e}_\alpha F_\alpha(\mathbf{x},t) = \mathbf{u}(\mathbf{x},t)\mathbf{F}(\mathbf{x},t) + \mathbf{F}(\mathbf{x},t)\mathbf{u}(\mathbf{x},t) \tag{47b}$$

respectively. Other relations for the LBE without a forcing term are still valid in this lumped-forcing SRT-LBE.

However, if we perform the Chapman-Enskog multi-scale expansion, Equation (43) with Equation (44a) have an extra force divergence term in the continuity equation

and extra terms of time derivative of force and divergence of velocity-force density tensor, i.e., RHS of Equation (47b), in the momentum equation. The adoption of Equation (44b) instead of Equation (44a) removes the extra term of divergence of velocity-force density tensor; however, it still contains other remaining extra terms, which can be removed for steady and uniform force such as gravity.

To remove the extra terms even for unsteady, non-uniform force, Guo et al. [42] proposed the split-forcing LBE, which enables the LBE to recover the NSE (continuity and momentum equations) with second-order accuracy. It has the same form as Equation (43). However, Guo et al. inserted the external force effect to the momentum by redefining the velocity (momentum) as

$$\rho\mathbf{u} = \sum_{\alpha} \mathbf{e}_{\alpha} f_{\alpha} + \frac{\Delta t}{2}\mathbf{F} \tag{48}$$

instead of Equation (16). Correspondingly, they changed the discrete force distribution function from Equation (44b) to

$$F_{\alpha}(\mathbf{x},t) = \left(1-\frac{1}{2\tau}\right) w_{\alpha}\left[3\frac{\mathbf{e}_{\alpha}-\mathbf{u}(\mathbf{x},t)}{c^2} + 9\frac{\mathbf{e}_{\alpha}\cdot\mathbf{u}(\mathbf{x},t)}{c^4}\mathbf{e}_{\alpha}\right]\cdot\mathbf{F}(\mathbf{x},t) \tag{49}$$

which satisfies

$$\sum_{\alpha} F_{\alpha}(\mathbf{x},t) = 0\,, \tag{50}$$

$$\sum_{\alpha} \mathbf{e}_{\alpha} F_{\alpha}(\mathbf{x},t) = \left(1-\frac{1}{2\tau}\right)\mathbf{F}(\mathbf{x},t)\,. \tag{51}$$

For consistency with the MRT-LBE, which will be discussed in the next section, this explicit-type split-forcing SRT-LBE can be rewritten as

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f_\alpha(\mathbf{x},t) - \frac{1}{\tau}[f_\alpha(\mathbf{x},t) - f_\alpha^{(eq)}(\mathbf{x},t)] + \left(1-\frac{1}{2\tau}\right)F_\alpha(\mathbf{x},t)\Delta t \tag{52}$$

with Equation (44b) instead of Equation (43) with Equation (49).

Alternatively, Cheng and Li [79] proposed a split-forcing LBE with a semi-implicit form:

$$\begin{aligned} f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f_\alpha(\mathbf{x},t) &- \frac{1}{\tau}[f_\alpha(\mathbf{x},t) - f_\alpha^{(eq)}(\mathbf{x},t)] \\ &+ \frac{\Delta t}{2}[F_\alpha(\mathbf{x},t) + F_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t)] \end{aligned} \tag{53}$$

with Equation (44b).

Two split-forcing LBEs proposed by Guo et al. [42] and Cheng and Li [79] are equivalent as shown in Appendix A. The difference is that Guo et al.'s equation represents the implicitness of force density by redefining the velocity, as in Equation (48), and correspondingly changing Equation (44b) to Equation (49). It should be also pointed out that the split-forcing LBE with multiple-relaxation-time (MRT) proposed by Premnath et al. [56, 58], which recovers the NSE with a second-order accuracy, has the similar explicit form as the split-forcing LBE with SRT by Guo et al. [38]. Thus, we can extend the same direct-forcing concept to the split-forcing LBE with MRT. Details are given in the next section.

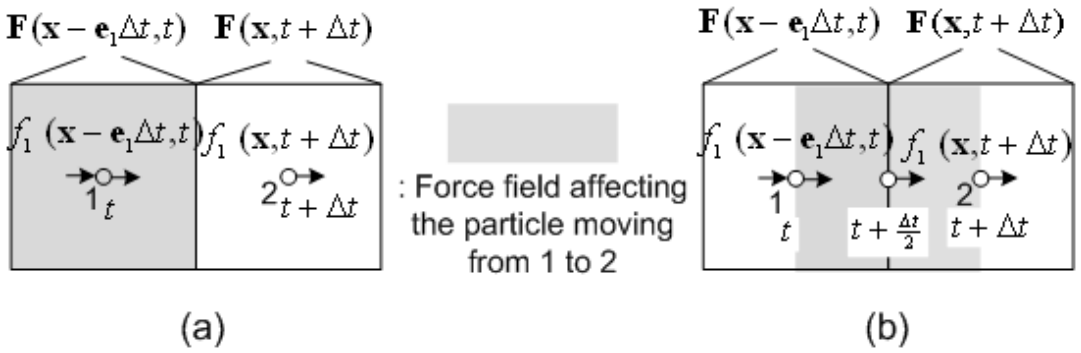

Figure 5. Comparison of force fields affecting the particle density distribution function in (a) the lumped-forcing LBE and (b) the split-forcing LBE.

The differences between the lumped- and split-forcing LBEs can be easily explained in terms of particle kinetics. As shown in Figure 5, a particle moves from point 1 to 2 during one time step under different force fields, $\mathbf{F}(\mathbf{x}_1,t)$ and $\mathbf{F}(\mathbf{x}_2,t+\Delta t)$. The momentum change of the particle is equal to the impulse (force multiplied by time). In the lumped-forcing LBE, only $\mathbf{F}(\mathbf{x}_1,t)$ is exerted during one time step. In contrast, in the split-forcing LBE, $\mathbf{F}(\mathbf{x}_1,t)$ and $\mathbf{F}(\mathbf{x}_2,t+\Delta t)$ are exerted during the first and the second half-time step, respectively. The important difference in terms of direct-forcing is that in the lumped-forcing LBE the momentum on point 2 at t+Δt is affected by only the force on point 1 at t, whereas in the split-forcing LBE the momentum is affected by the force on point 2 at t+Δt as well as the force on point 1 at t.

In actual numerical calculation, the LBE with a forcing term can be solved in the following four steps:

- First-forcing step:

$$\rho(\mathbf{x},t)\mathbf{u}(\mathbf{x},t)=\sum_{\alpha}\mathbf{e}_{\alpha}f_{\alpha}(\mathbf{x},t)+\frac{\Delta t}{2}\mathbf{F}(\mathbf{x},t), \tag{54a}$$

$$f'_{\alpha}(\mathbf{x},t)=f_{\alpha}(\mathbf{x},t)+\frac{\Delta t}{2}F_{\alpha}(\mathbf{x},t) \tag{54b}$$

- Collision step:

$$f'_{\alpha}(\mathbf{x},t)=f_{\alpha}(\mathbf{x},t)-\frac{1}{\tau}[f_{\alpha}(\mathbf{x},t)-f_{\alpha}^{(eq)}(\mathbf{x},t)] \tag{55a}$$

$$f''_{\alpha}(\mathbf{x},t)=f'_{\alpha}(\mathbf{x},t)-\frac{1}{\tau}[f'_{\alpha}(\mathbf{x},t)-f'^{(eq)}_{\alpha}(\mathbf{x},t)] \tag{55b}$$

- Second-forcing step:

$$f''_{\alpha}(\mathbf{x},t)=f'_{\alpha}(\mathbf{x},t)+\Delta t\,F_{\alpha}(\mathbf{x},t) \tag{56a}$$

$$f'''_{\alpha}(\mathbf{x},t)=f''_{\alpha}(\mathbf{x},t)+\frac{\Delta t}{2}F_{\alpha}(\mathbf{x},t) \tag{56b}$$

- Streaming step:

$$f_{\alpha}(\mathbf{x}+\mathbf{e}_{\alpha}\Delta t,t+\Delta t)=f''_{\alpha}(\mathbf{x},t) \tag{57a}$$

$$f_{\alpha}(\mathbf{x}+\mathbf{e}_{\alpha}\Delta t,t+\Delta t)=f'''_{\alpha}(\mathbf{x},t) \tag{57b}$$

where Equations (54a), (55a), (56a), and (57a) and Equations (54b), (55b), (56b), and (57b) represent the calculation algorithms for Guo et al.'s and Cheng and Li's LBEs, respectively. In Guo et al.'s LBE, f'_{α} and f''_{α} are called post-collision and post-forcing particle distribution functions, respectively, while in Cheng and Li's LBE, f'_{α}, f''_{α}, and f'''_{α} are called post-first-forcing, post-collision, and post-second-forcing PDFs, respectively. We adopt Guo et al.'s LBE in the present study because calculating

only the macroscopic variables in the first-forcing step here is more efficient than calculating all PDFs in Cheng and Li's LBE, and the force term in Guo et al.'s LBE can be explicitly applied in the IB-LBM as will be discussed in Section C.1. It should be also noted that two results are the same.

2. Multiple-relaxation-time lattice Boltzmann equation with forcing term

The MRT-LBE with a forcing term [58] also computes the forcing term, which represents the effect of external forces as a second-order accurate time-discretization, in moment space.

The explicit-type split-forcing MRT-LBE can be expressed as [56]:

$$\mathbf{f}(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = \mathbf{f}(\mathbf{x},t) - \mathbf{M}^{-1}\{\mathbf{S}[\hat{\mathbf{f}}-\hat{\mathbf{f}}^{(eq)}] + (\mathbf{I}-\tfrac{1}{2}\mathbf{S})\hat{\mathbf{f}}_F\} \tag{58}$$

where $\hat{\mathbf{f}}_F = \mathbf{M}\mathbf{f}_F$ with $\mathbf{f}_F = \{F_\alpha\}_{\alpha=0,\cdots,18} = (F_0, F_1, \cdots, F_{18})^T$ in D3Q19 model. The forcing term F_α can be written as [80-81]:

$$F_\alpha = \frac{(\mathbf{e}_\alpha - \mathbf{u})\cdot\mathbf{F}}{\rho c_s^2} f_\alpha^{(eq)}. \tag{59}$$

By neglecting terms with higher order than $O(\mathrm{Ma}^2)$, Equation (59) can be simplified as

$$F_\alpha(\mathbf{x},t) = w_\alpha\left[3\frac{\mathbf{e}_\alpha - \mathbf{u}(\mathbf{x},t)}{c^2} + 9\frac{\mathbf{e}_\alpha\cdot\mathbf{u}(\mathbf{x},t)}{c^4}\mathbf{e}_\alpha\right]\cdot\mathbf{F}(\mathbf{x},t) \tag{60}$$

which is the same as Equation (44b). Also, the macroscopic velocity is redefined as in the explicit-type split-forcing SRT-LBE by Equation (48).

It should be noted that the explicit-type split-forcing SRT-LBE (52) and MRT-LBE (58) have the same form with the exception that the collision calculation part in Equation (58) is expressed in the moment space.

In the MRT-LBE without the forcing term, the value of the relaxation times for the conserved moments ($s_0, s_3, s_5,$ and s_7) are insignificant because their corresponding equilibrium distribution is set to the value of the respective moments itself. However, with a forcing term, they need to be nonzero [82-83]. In this study, we use $s_0 = s_3 = s_5 = s_7 = 1.0$ as in [58]. Other relaxation times are the same as in Equation (26) in the MRT-LBE without a forcing term.

C. Direct-Forcing Immersed Boundary-Lattice Boltzmann Method

In this section, we derive the direct-forcing formula based on the explicit-type split-forcing LBE and apply the various interface schemes to the derived direct-forcing formula.

1. Direct-forcing formula

Here, we consider the direct-forcing formula to evaluate the boundary force density in the IB-LBM. In the previous direct-forcing IB-LBMs, most of authors [20-21, 84] used the following direct-forcing formula for the evaluation of the boundary force, as in the direct-forcing IBM based on the NSE method:

$$\mathbf{F}(\mathbf{x},t) = \rho \frac{\mathbf{U}^d - \mathbf{u}^{noF}(\mathbf{x}, t+\Delta t)}{\Delta t} \tag{61}$$

where $\mathbf{U}^{d}$ and $\mathbf{u}^{noF}$ are the desired velocity and the unforced velocity at a forcing point $\mathbf{x}$ and time t+Δt, respectively. Here, the unforced velocity can be evaluated from the information at ($\mathbf{x}$,t) by using the NSE without a forcing term, as in [20, 84], or the LBE without a forcing term, as in [21]. The direct-forcing formula (61) is valid in the IB-LBM if the LBE is

$$f_\alpha(\mathbf{x}, t+\Delta t) = f_\alpha(\mathbf{x}, t) - \frac{1}{\tau}[f_\alpha(\mathbf{x}, t) - f_\alpha^{(eq)}(\mathbf{x}, t)] + F_\alpha(\mathbf{x}, t)\Delta t \tag{62}$$

instead of Equation (43). However, in the LBM, as shown in Equation (43), regardless of adopting lumped- and split-forcing LBEs, the velocity on a given node at the next time step is determined by the PDFs not on the given node at the current time step but on neighboring nodes at the current time step, which will be streamed to the given node at the next time step. This is due to the kinetic nature of the LBE. Actually, on the basis of the lumped-forcing LBE, the following direct-forcing formula is satisfied:

$$\sum_\alpha \mathbf{e}_\alpha F_\alpha(\mathbf{x} - \mathbf{e}_\alpha \Delta t, t) = \rho \frac{\mathbf{U}^d - \mathbf{u}^{noF}(\mathbf{x}, t+\Delta t)}{\Delta t} \tag{63}$$

whose derivation is given in Appendix B. Thus, if we adopt Equation (64) for the boundary force calculation in the LBM, the force field is intrinsically diffuse because it assumes implicitly that

$$\sum_\alpha \mathbf{e}_\alpha F_\alpha(\mathbf{x} - \mathbf{e}_\alpha \Delta t, t) = \mathbf{F}(\mathbf{x}, t), \tag{64}$$

which means that the force field is $\mathbf{F}$($\mathbf{x}$,t) even on the neighboring nodes, i.e., diffused by one node; thus producing less accurate (more diffuse) results.

On the other hand, based on the split-forcing LBE, we can derive the following direct-forcing formula:

$$\mathbf{F}(\mathbf{x}, t+\Delta t) = 2\rho(\mathbf{x}, t+\Delta t)\frac{\mathbf{U}^d - \mathbf{u}^{noF}(\mathbf{x}, t+\Delta t)}{\Delta t}, \tag{65}$$

whose derivation is presented in Appendix C. It is evident that the same formula is valid for the split-forcing MRT-LBE. The direct-forcing formula based on the split-forcing LBE, Equation (65), does not involve the diffuse force field as in Equation (61). The preliminary calculation of flow past a circular cylinder with the explicit diffuse interface schemes with 2- and 4-point discrete delta functions showed that Equation (61), coupled with lumped-forcing LBE as in [21], produces more diffuse results than the results from Equation (65) coupled with split-forcing LBE. For example, at Re=40, the former had drag coefficients of 1.595 and 1.617 for 2- and 4-point discrete delta functions, respectively, while the latter had 1.576 and 1.597. We could find that the drag coefficient from the direct-forcing formula (61) with a 2-point discrete delta function is comparable to that from the direct-forcing formula (65) with a 4-point discrete delta function. In this simulation, the effect of lumped-forcing LBE on the results can be neglected because extra terms related to time derivative of force and divergence of velocity-force density tensor in momentum equation are almost zero due to steady flow and stationary boundary conditions, respectively; the extra term related to divergence of force density in continuity equation is zero due to the assumption of Equation (64). Thus, the difference of drag coefficients can be mainly due to the difference between two direct-forcing formulas.

2. Interface schemes

In this section, we consider interface schemes under the direct-forcing formula based on the split-forcing LBE, Equation (65). The interface region affected by the boundary force can be set either diffusely or sharply. In this paper, both the diffuse and sharp interface schemes are dealt with. For the diffuse interface scheme, both explicit and implicit methods are considered.

a. Diffuse interface scheme

In the diffuse interface scheme, the boundary is represented by a set of the Lagrangian forcing points on the boundary, and the flow field is represented by the Eulerian computational nodes covering both inside and outside the boundary. Hence, interpolation from neighboring nodes to boundary points for the boundary force evaluation and distribution of the force to the neighboring nodes are needed.

Figure 6 shows the calculation procedure of the direct-forcing IB-LBM with an explicit diffuse interface scheme. After the streaming step, with streamed PDFs, unforced velocities in Eulerian nodes ($\mathbf{u}_{ijk}^{noF}$) are calculated in Step (a). Then, in Step (b), the unforced velocity on the boundary point ($\mathbf{u}_{b}^{noF}$) is calculated by interpolating from neighboring unforced velocities. The boundary force on boundary point b ($\mathbf{F}_b$) is evaluated using the interpolated velocity ($\mathbf{u}_{b}^{noF}$) and a desired velocity ($\mathbf{U}_b$) given by the no-slip condition in Step (c). Its distribution to neighboring nodes is implemented in Step (d). Then, in Step (e), the velocities of neighboring Eulerian nodes are updated (forced). Here, D is the discrete delta function, h is the mesh spacing and is equal to

lattice size; that is, $h=\Delta x$, and Δs_b is the arc length of the boundary segment in 2D and the area of the boundary surface in 3D. In this study, to check the effect of the sharpness of the boundary on the solution, two types of discrete delta functions D are chosen:

$$D(\mathbf{x}_{ij}-\mathbf{x}_b)=\frac{1}{h^2}d_h\left(\frac{x_i-x_b}{h}\right)d_h\left(\frac{y_j-y_b}{h}\right) \text{ in 2D,} \tag{66a}$$

$$D(\mathbf{x}_{ijk}-\mathbf{x}_b)=\frac{1}{h^3}d_h\left(\frac{x_i-x_b}{h}\right)d_h\left(\frac{y_j-y_b}{h}\right)d_h\left(\frac{z_k-z_b}{h}\right) \text{in 3D} \tag{66b}$$

with

$$d_h(r)=\begin{cases}1-|r|, & |r|\le 1\\ 0, & |r|>1\end{cases} \tag{67}$$

and

$$d_h(r)=\begin{cases}\frac{1}{8}\left(3-2|r|+\sqrt{1+4|r|-4r^2}\right), & 0\le|r|<1\\ \frac{1}{8}\left(5-2|r|-\sqrt{-7+12|r|-4r^2}\right), & 1\le|r|<2\\ 0, & |r|\ge 2\end{cases}. \tag{68}$$

Here, Equation (67) is the 2-point discrete delta function, which corresponds to the bilinear interpolation in 2D and tri-linear interpolation in 3D, and Equation (68) is the 4-point discrete delta function introduced by Peskin [75]. It should be pointed out that the 2-point and 4-point discrete delta functions adopted in this study satisfy the 0^{th} and 1^{st} discrete moment conditions, while cosine-type 4-point discrete delta function used in [41, 84] only satisfies the 0^{th} discrete moment condition. Thus, the former gives a second-order approximation and the latter gives a first-order approximation [75-76].

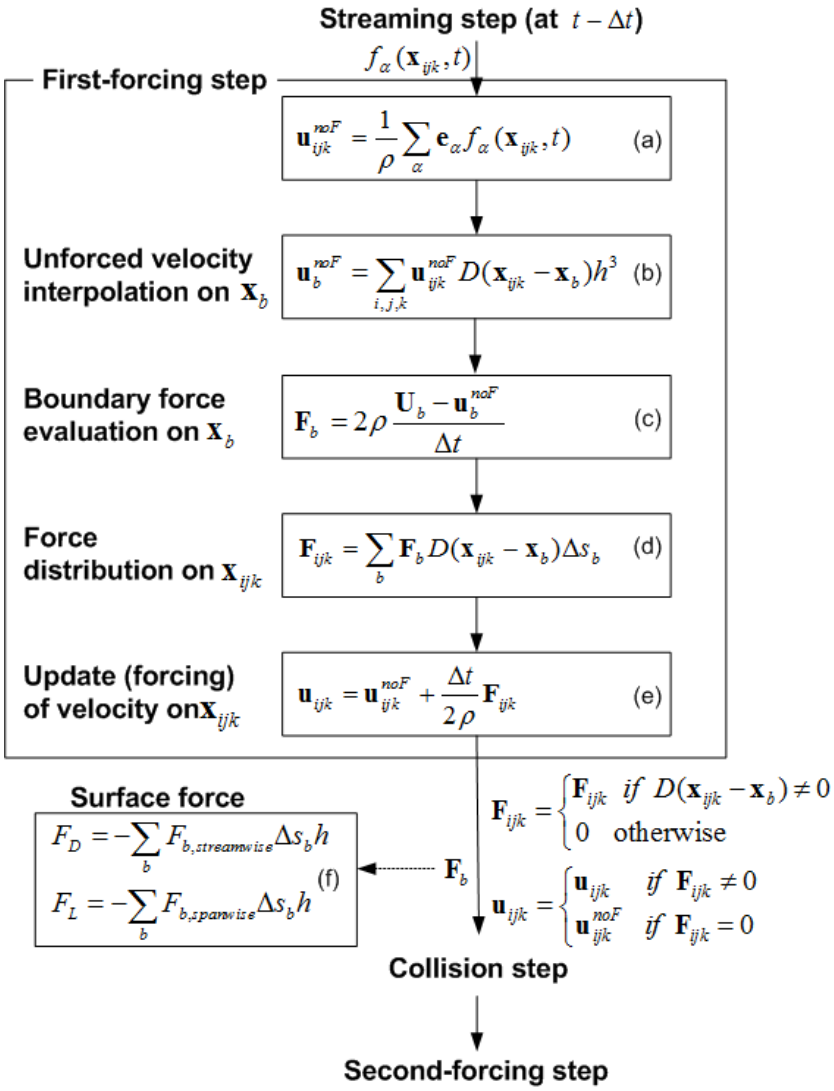

Figure 6. Calculation algorithm of the explicit diffuse direct-forcing IB-LBM.

In the IBM, the surface force for the stationary body can be easily evaluated using [3]:

$$\mathbf{F}_s=-\sum_b \mathbf{F}(\mathbf{x}_b)\Delta s_b=-\sum_{i,j}\mathbf{F}(\mathbf{x}_{ij})h^2 \quad \text{in 2D,} \tag{69a}$$

$$\mathbf{F}_s = -\sum_b \mathbf{F}(\mathbf{x}_b)\Delta s_b = -\sum_{i,j,k} \mathbf{F}(\mathbf{x}_{ijk})h^3 \text{ in 3D} \quad (69b)$$

We can adopt the first- or the second-summation formulas in the diffuse interface scheme. The surface force evaluation for moving boundary problems will be discussed in the next chapter.

In this explicit diffuse interface scheme, the boundary force density is explicitly obtained. However, as mentioned in Chapter I, the explicit forcing may not ensure the no-slip condition when the velocity on $\mathbf{x}_b$ is interpolated again from the updated (forced) velocities on neighboring Eulerian nodes; this is because the forces used for updating (forcing) the velocities were from velocities before update (forcing). Thus, in this study, we also consider the implicit forcing scheme.

For the implicit diffuse interface scheme, we adopt the multi-direct-forcing method proposed by Luo et al. [26] and Wang et al. [19] in order to avoid the calculation of complicated banded matrix equations as in [25, 41]. The calculation algorithm is depicted in Figure 7. Steps (a) to (e) are the same as in the explicit diffuse forcing scheme. The differences occur after this. We iterate the forcing procedures in Steps (c) to (f) until the difference between the boundary velocity interpolated from the updated velocities and the desired velocity becomes very small; that is, the no-slip condition on the boundary points is better satisfied. In the multi-direct-forcing method, the number of forcing (NF) adjusts the implicitness of the method. We can also notice that if there is no iteration, that is, NF=1, then this corresponds to an explicit forcing scheme.

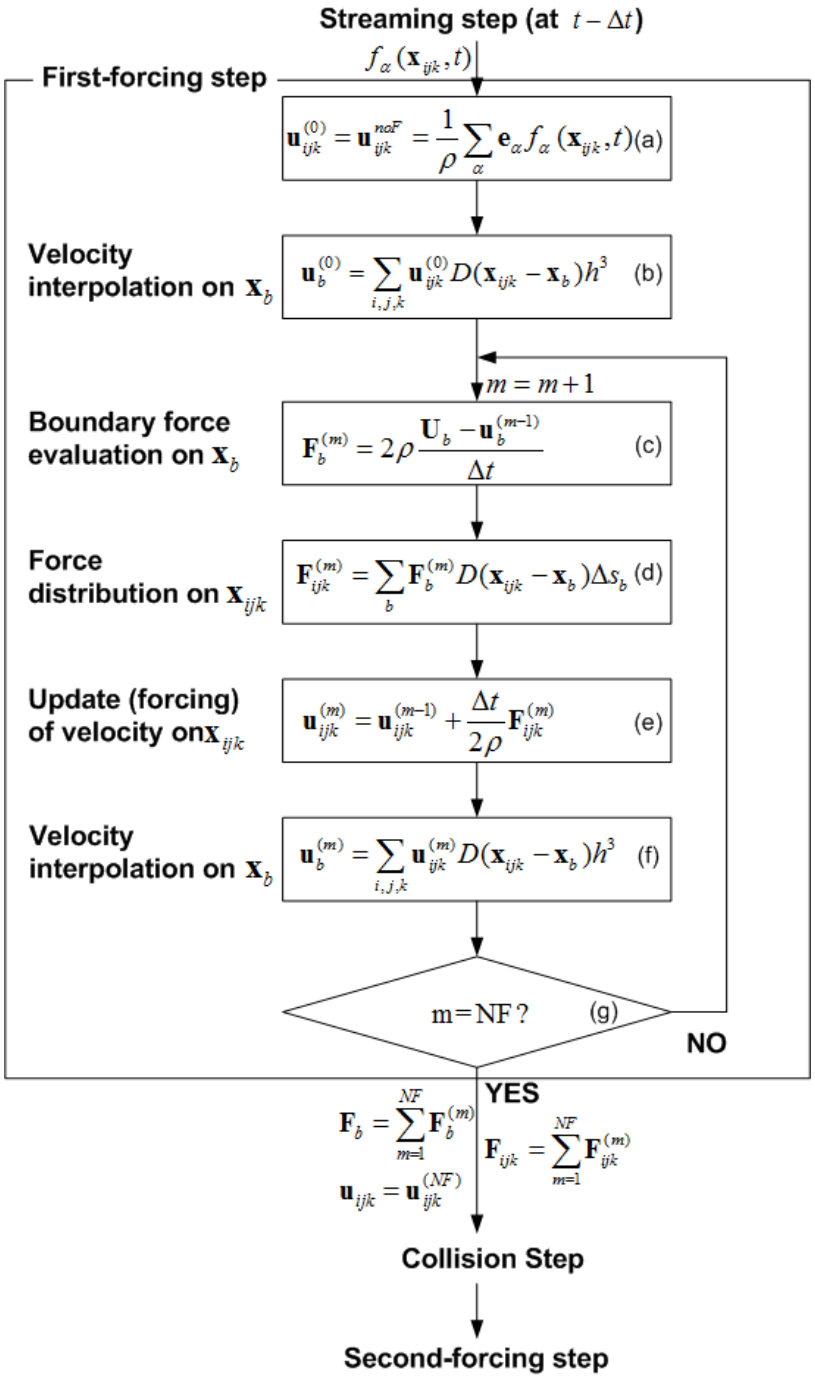

Figure 7. Calculation algorithm of the multi-direct-forcing IB-LBM (implicit diffuse direct-forcing IB-LBM).

b. Sharp interface scheme

In this section, the sharp interface scheme is explained. Figure 8 presents the calculation procedure of the direct-forcing IB-LBM with a sharp interface scheme. Because the forcing nodes are located on the computational nodes, not on the boundary, the interpolation procedure is required to obtain the desired velocity on the node so that its corresponding boundary points can satisfy the no-slip boundary condition and the force distribution step is not required. In addition, surface forces can be directly calculated by the second-summation formula in Equation (69a) and (69b) as in [3, 9].

The accuracy of the sharp interface scheme depends on interpolation I in Step (b). In this paper, we consider the case of exterior forcing nodes, that is, the exterior sharp interface scheme. As mentioned in Chapter I, the exterior forcing point is located on the computational node outside the interested fluid domain and closest to the boundary.

To evaluate the boundary force density on the forcing node, we adopt the simple, systematic interpolation procedure proposed by Kim et al. [9]. It uses second-order linear and bilinear interpolations from neighboring fluid node velocities and the boundary point velocity where the no-slip boundary condition is satisfied in order to evaluate the desired velocity and boundary forcing on the forcing point. Figure 9 illustrates the interpolation procedure in 2D geometry to evaluate the desired velocity on the forcing point (f-point), which makes the no-slip boundary satisfied at the adjacent boundary point (b-point). The interpolation has two typical cases in 2D, as shown in Figure 9(a). For Case 1, where three unforced fluid nodes are available, the following

second-order bilinear interpolation is used (Figure 9(b)):

$$\mathbf{u}_f = \frac{1}{\Delta_x \Delta_y}\left\{\mathbf{U}_b - [\Delta_x(1-\Delta_y)\mathbf{u}_2 + (1-\Delta_x)(1-\Delta_y)\mathbf{u}_3 + (1-\Delta_x)\Delta_y\mathbf{u}_4]\right\} \tag{70}$$

where the location of $\mathbf{U}_b$ (b-point) is chosen to be a point on the boundary where a straight line passes through the forcing point (f-point) perpendicularly, intersecting the boundary. However, if only two unforced fluid nodes are available, as in Case 2, the bilinear interpolation involves another forcing node; thus, requiring the iteration. To avoid this, for Case 2, the following second-order linear interpolation is adopted (Figure 9(c)):

$$\mathbf{u}_f = \begin{cases} \dfrac{1}{\Delta}\mathbf{U}_b - \dfrac{1-\Delta}{\Delta}\mathbf{u}_1, & \text{if } \Delta \geq 1/2 \\ 2\mathbf{U}_b - 2\Delta\mathbf{u}_1 - (1-2\Delta)\mathbf{u}_2, & \text{if } \Delta < 1/2 \end{cases}. \tag{71}$$

The reason we separate the linear interpolation into two cases is that if Δ is small, the denominator in the upper interpolation becomes small, thus causing the instability [9].

For 3D problems, we can begin with the tri-linear interpolation if the forcing node has seven neighboring fluid nodes to be used for the interpolation. Otherwise, the interpolation degenerates into bilinear or linear interpolations, as explained in the 2D case.

Notably, the present direct-forcing IB-LBM with a sharp interface scheme is similar to Kim et al.'s [9] not only in that both of them adopt the same velocity interpolation scheme for the evaluation of the boundary force density on the forcing nodes but also in that both of them explicitly add the force density term to the governing equations. The difference is that Kim et al. [9] used the predictor step for the evaluation

of the body force density, while we used the discrete lattice effect due to the kinetic nature of the split-forcing LBE.

It should also be noted that this interpolation procedure for exterior forcing nodes is one example of various interpolation methods. As mentioned in Chapter I, the other systematic interpolation procedures in [14-17] can also be adopted. Also, for interior forcing points, refer to [10-13].

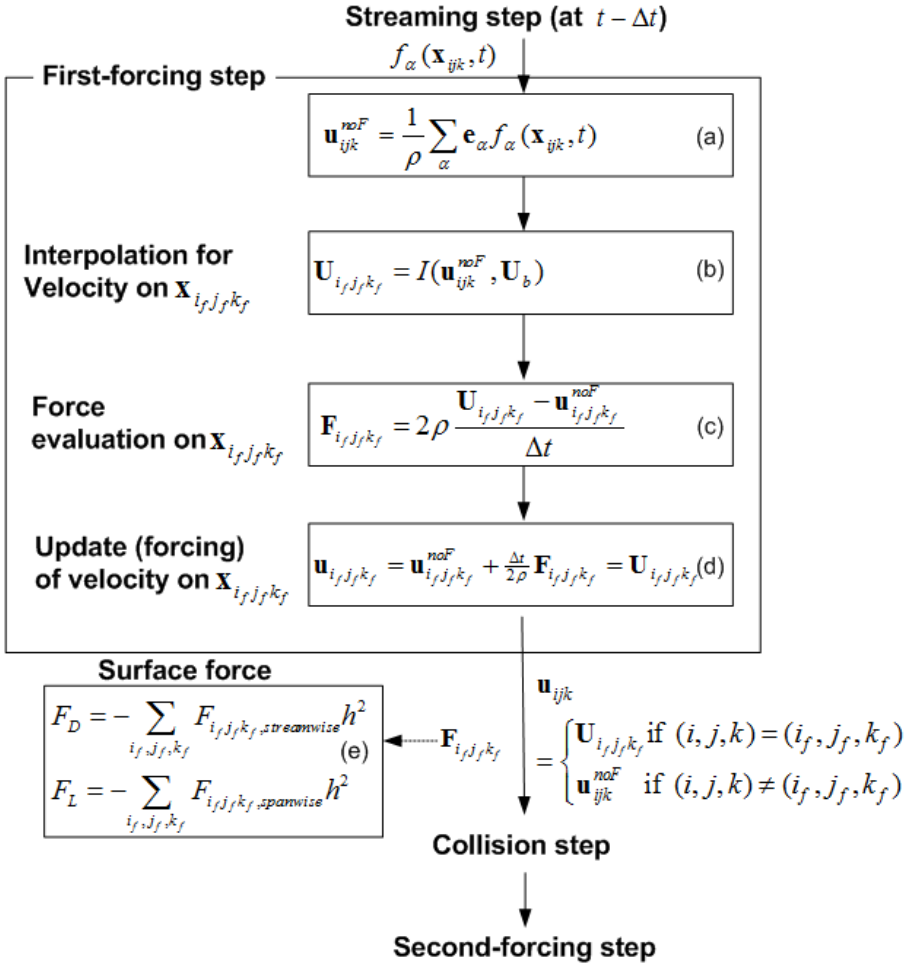

Figure 8. Algorithm of the sharp direct-forcing IB-LBM.

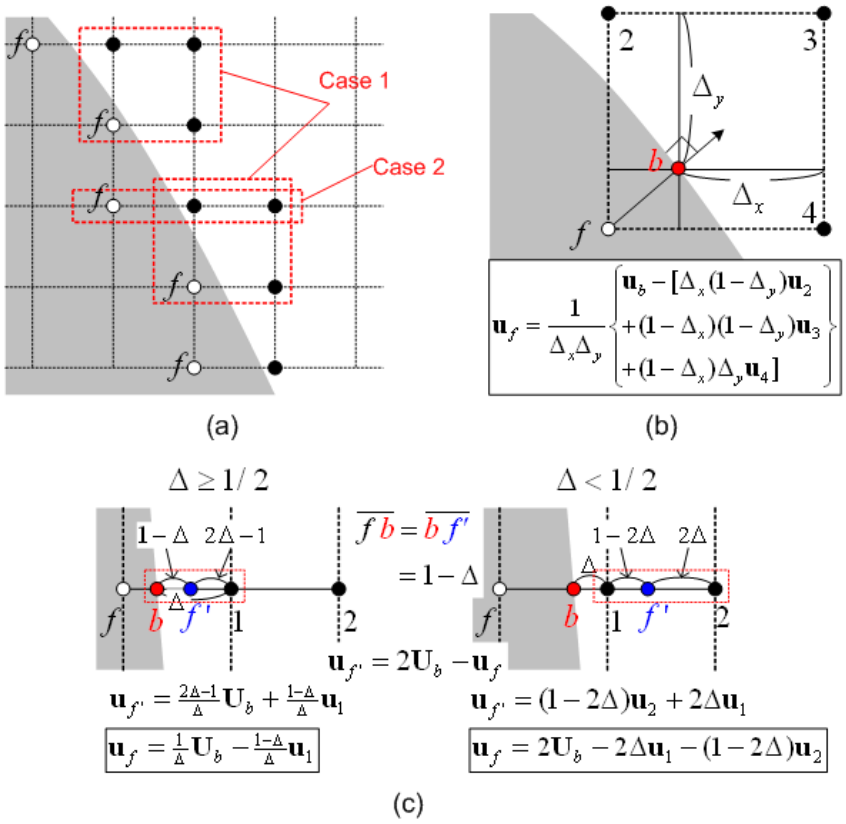

Figure 9. Schematic diagram of bilinear and linear interpolations for evaluating the desired velocity on the exterior forcing point f in 2D problem: (a) typical two cases, (b) bilinear interpolation for Case 1, and (c) linear interpolation for Case 2.

D. Simulation Results

1. Taylor-Green decaying vortex

To evaluate the accuracy of the proposed IB-LBM, the simulation of unsteady flow is carried out in this section. The test problem is the 2D Taylor-Green vortex flow in a square box, which has the following analytical solutions:

$$u_x = -u_0 \cos(\pi x / L)\sin(\pi y / L)\exp(-2\nu(\pi / L)^2 t)\,, \tag{72a}$$

$$u_y = u_0 \sin(\pi x / L)\cos(\pi y / L)\exp(-2\nu(\pi / L)^2 t), \tag{72b}$$

$$p = p_0 - \frac{u_0^2}{4}\left[\cos(2\pi x / L) + \cos(2\pi y / L)\right]\exp(-4(\pi / L)^2 \nu t). \tag{72c}$$

In this simulation, a circle with the radius of 0.5L is embedded at the center of the square domain [-L,L]×[-L,L]. Initial conditions are imposed by Equations (72a) through (72c) with t=0. Time-dependent boundary conditions at the square and the embedded circle are given by Equations (72a) and (72b) using the non-equilibrium extrapolation scheme [85] and the immersed boundary methods explained in the previous section, respectively. In the exterior sharp interface scheme, forcing points are located on the Eulerian nodes outside the embedded circle because the interested flow field is inside the circle. For the diffuse interface schemes, forcing points are distributed on the circle boundary with a spacing of $\Delta s=h/1.5$, which is small enough to ensure the spacing-independent solution [25]. Reynolds number is taken as $Re=u_0L/\nu=10$, and the dimensionless relaxation time is set to be $\tau=0.65$, as in [34]. The simulation is carried out using four sets of grids, L=10, 20, 40, and 80Δx. At time $t=L/u_0=1$, the overall error of velocities inside the embedded circle is evaluated by using the following L_2-norm error:

$$L_2\text{-error} \equiv \sqrt{(1/n)\sum (u_x^c - u_x^a)^2 + (u_y^c - u_y^a)^2} \tag{73}$$

where the summation is over the nodes inside the circle, and thus, *N* is number of the nodes and superscripts, *c* and *a* mean computational and analytical values, respectively. In addition, to evaluate the no-slip error on the boundary points for the explicit and implicit diffuse direct-forcing schemes, the following boundary-error is adopted:

$$\text{boundary-error} \equiv \sqrt{\frac{\sum (u_x^c - u_x^a)^2 + (u_y^c - u_y^a)^2}{\sum (u_x^a)^2 + (u_y^a)^2}} \tag{74}$$

where the summation is over boundary points on the circle.

Figure 10 shows the velocity magnitude and vector plots at t=1 when using the explicit diffuse interface scheme. Figure 11 presents the overall errors versus the number of grids across the circle in the log scale. Regarding the order of accuracy, the present IB-LBM is almost second-order. Regarding accuracy itself, the sharp interface scheme is more accurate than the explicit diffuse interface schemes and its accuracy is close to that without the embedded circle. In the explicit diffuse interface scheme, the 2-point discrete delta function produces more accurate results than the 4-point discrete delta function does. The results indicate that the narrower (sharper) is the distribution of the forced point, the better is the accuracy.

Figures 12 and 13 present the effect of the implicit forcing on the accuracy in cases of using the 2-point discrete delta function and the 4-point discrete delta function, respectively. As can be seen in Figures 12(a) and 13(a) in both cases, as the number of forcing increases, the boundary-error decreases clearly, although in the case of the 4-point discrete delta function, the decrease in the boundary-error is reduced as the grid size decreases. However, the local and overall accuracies are not improved, as shown in Figures 12(b) and (c) and 13(b) and (c). Besides, in the diffuse interface scheme with the 2-point discrete delta function, both local (L_∞) and overall (L_2) errors rather increase with the increasing number of forcing, as shown in Figures 12(b) and (c). The results of implicit forcing effects indicate that implicit forcing enhances the accuracy on boundary

points but does not improve the accuracies of the flow field due to its intrinsic nature of discrete delta functions. Instead, a more fundamental enhancement of accuracy in the diffuse direct-forcing scheme can be attained by selecting the sharper discrete delta functions, as discussed in the explanation of Figure 11.

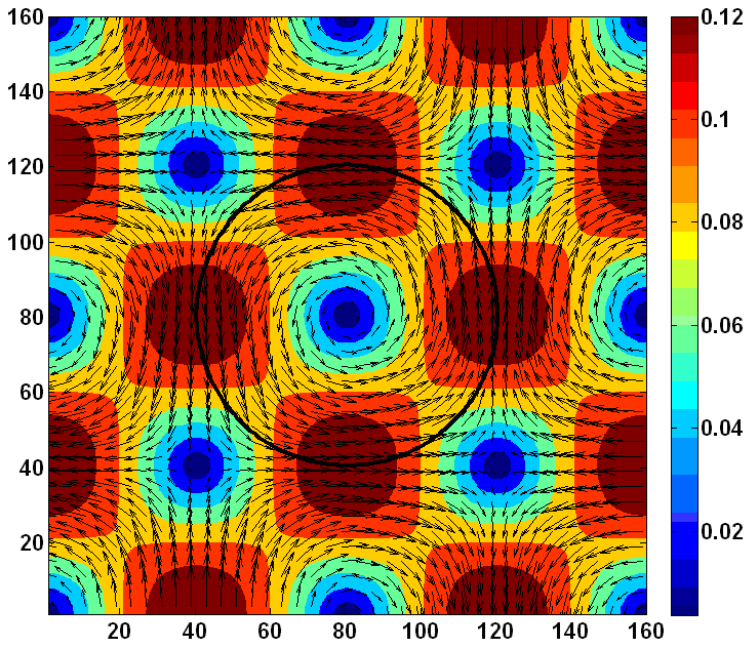

Figure 10. Velocity magnitude and vector plots of the Taylor-Green vortex at t=1 resulting from the explicit diffuse direct-forcing scheme with L=D=80Δx. The solid line indicates the embedded circle.

It should be noted that Wu and Shu [41] also performed the same simulation with a different implicit diffuse interface scheme and showed the deterioration of the order of overall accuracy (1.9), while the present result shows 1.98. As Wu and Shu stated, this can be attributed to the use of a cosine-type 4-point discrete delta function, which gives only a first-order approximation [75-76].

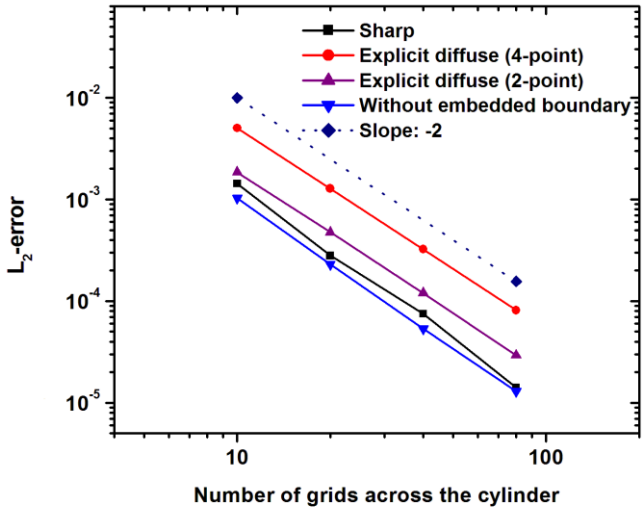

Figure 11. Overall accuracy of the present IB-LBMs for the Taylor-Green vortex.

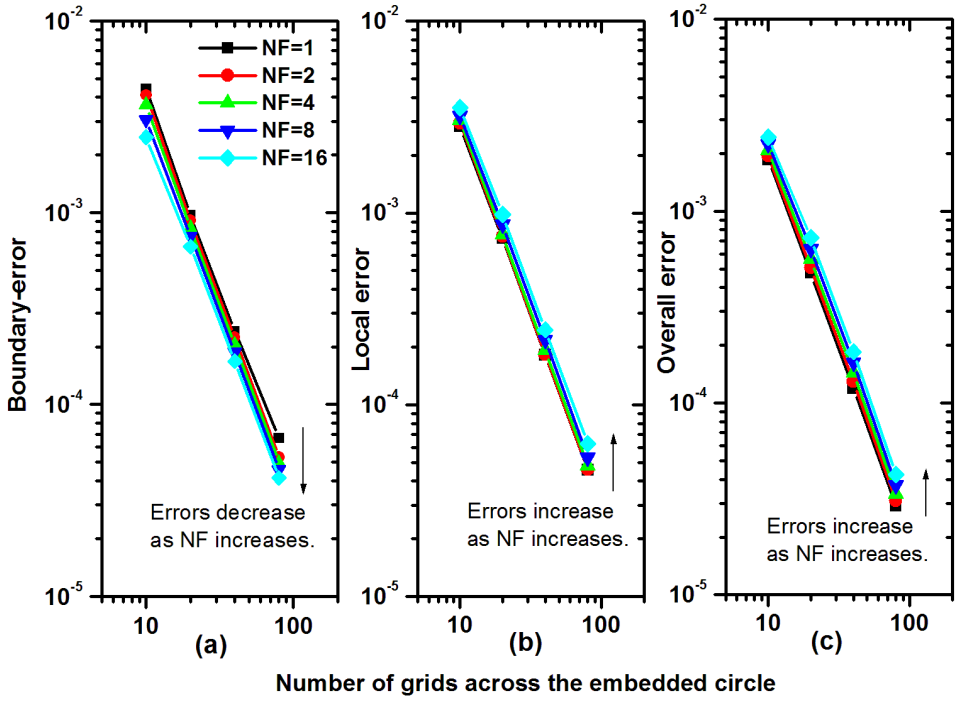

Figure 12. The effect of the number of forcing on (a) the boundary-error, (b) the local L_∞-error, and (c) the overall L_2-error in diffuse direct-forcing schemes with the 2-point discrete delta function.

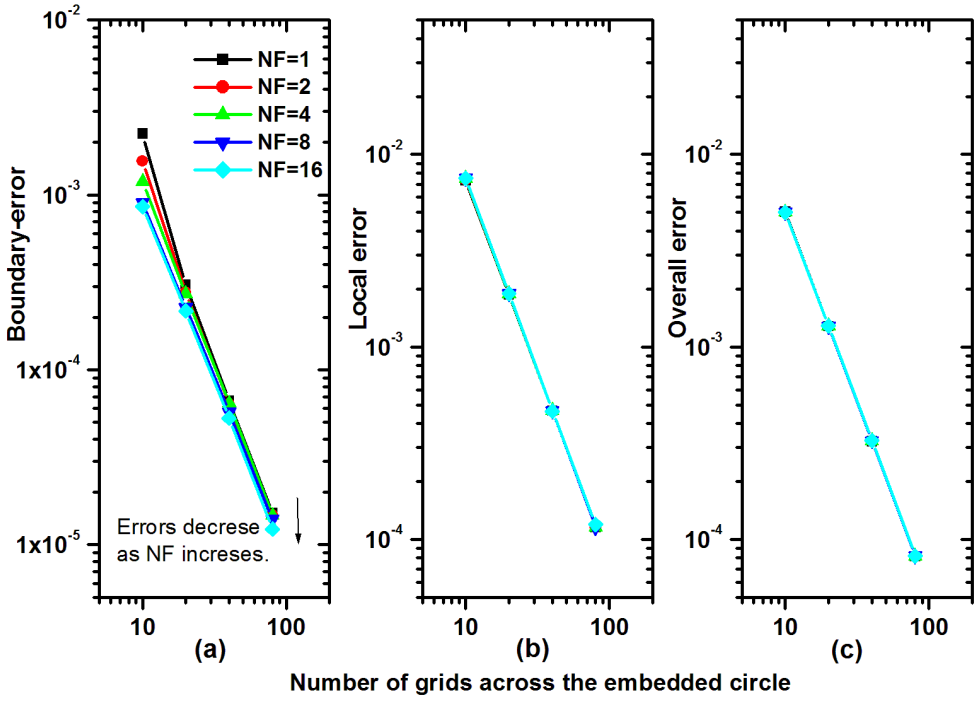

Figure 13. The effect of the number of forcing on (a) the boundary-error, (b) the local L_∞-error, and (c) the overall L_2-error in diffuse direct-forcing schemes with the 4-point discrete delta function.

2. Flow past a circular cylinder

Now, we assess the interface schemes for flow past a circular cylinder. This is one of the representative benchmark problems for checking the accuracy of a numerical method in complex geometries and thus there are many comparable results from various numerical methods available. In this problem, the flow pattern changes according to the Reynolds number, which is defined as $Re=u_\infty D/\nu$, where u_∞ is the freestream velocity and D is the diameter of the cylinder. At a low Reynolds number ($Re<46$), the flow is steady, and a pair of counter-rotating vortices is generated symmetrically about the centerline of the

wake. In this study, Re=20 and 40 are investigated for steady flows in this range. For Re>46, unsteadiness arises spontaneously, and vortex shedding occurs. As Reynolds number increases further, the transition to 3D is induced in the near wake. Thus, for unsteady flows, Reynolds numbers ranging up to 150, specifically Re=100 and 150, are investigated in this study because 2D flow simulations do not accurately reflect the transition range (Re=150~300) [86-87].

In the steady flow simulation, the computational domain is taken as 40D×40D with 801×801 grid points for the uniform grid, and a circular cylinder is located at the center of the domain. To check the effect of grid size on the solution efficiently, a refined grid near the cylinder is also used. For the refined grid, the grid refinement technique for the LBE developed by Rohde et al. [88] is adopted. Figure 14 shows the grid distribution with the refined grid around a cylinder. Here, one large lattice corresponds to 20 actual coarse lattices, and one small lattice corresponds to 20 actual refined lattices. In the refined grid, 40 grid points are used across the cylinder. For inlet and far-field boundaries, the Dirichlet boundary condition is used and for the outlet boundary, the homogeneous Neumann boundary condition is used. For the diffuse interface schemes, forcing points are uniformly distributed on the cylinder surface with a spacing of $\Delta s = h/1.5$.

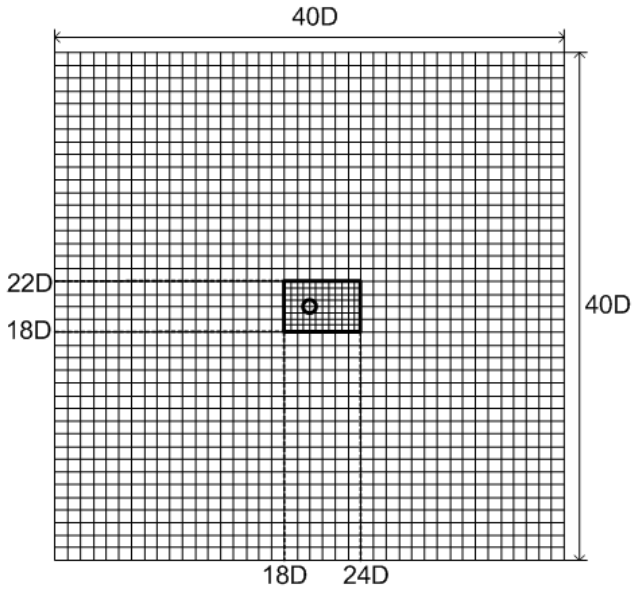

Figure 14. Grid refinement for steady flow past a circular cylinder.

The drag coefficient (C_D) and the recirculation length (L_w) are computed in the steady flows. The drag coefficient is defined as

$$C_D = \frac{F_D}{u_\infty^2 D/2} \tag{75}$$

where F_D is the drag force and can be easily obtained by using Equation (69a). Since the boundary point velocity is zero, the boundary-error to evaluate the no-slip error on the boundary point for the explicit and implicit diffuse interface schemes is redefined as

$$\text{boundary-error} \equiv \sqrt{(1/N_b)\sum (u_x^c - u_x^a)^2 + (u_y^c - u_y^a)^2} \tag{76}$$

where the summation is over the boundary points on the cylinder surface, and thus, N_b is the number of the nodes.

For unsteady flows with Re=100 and 150, the refined grid is used, and the downstream part is extended by 10D; thus, the domain size changes from 40D×40D to 50D×40D. For unsteady flows, lift coefficients (C_L) and Strouhal numbers (St) as well as drag coefficients (C_D) are compared with other experimental and numerical data. The lift coefficient is defined as

$$C_L = \frac{F_L}{u_\infty^2 D/2} \tag{77}$$

where F_L is the lift force and is easily obtained by Equation (69a). The Strouhal number is defined as

$$St = f_q D/u_\infty \tag{78}$$

where f_q is the vortex shedding frequency and can be obtained from the time evolution of lift coefficients.

Figure 15 shows the streamlines near the circular cylinder resulting from the exterior sharp interface scheme in the refined grid. The symmetric vortices are clearly observed in the wake region.

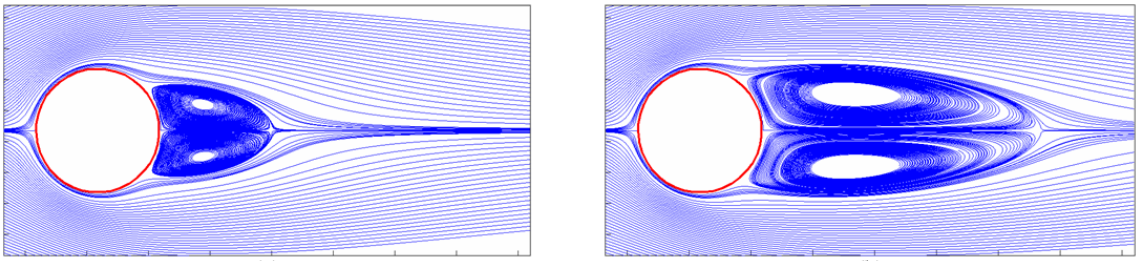

Figure 15. Streamlines at (a) Re=20 and (b) Re=40 resulting from the exterior sharp interface scheme in the refined grid (D=40Δx).

Table 1. The effect of the number of forcing on drag coefficients and recirculation lengths at Re=40 in the implicit multi-direct-forcing scheme with 2-point and 4-point discrete delta functions.

The number of forcing (NF)		1 (Explicit forcing)	2	5	10	20
With 4-point discrete delta function	C_D	1.597	1.589	1.585	1.584	1.584
	L_w	2.525	2.501	2.487	2.485	2.486
	boundary-error	1.6105×10^{-3}	4.5004×10^{-4}	8.5795×10^{-5}	4.4316×10^{-5}	4.3167×10^{-5}
With 2-point discrete delta function	C_D	1.576	1.573	1.574	1.575	1.577
	L_w	2.435	2.431	2.435	2.438	2.441
	boundary-error	7.4550×10^{-4}	4.5476×10^{-4}	3.2953×10^{-4}	2.5711×10^{-4}	1.8260×10^{-4}

Table 1 shows the effect of the number of forcing on drag coefficients, recirculation lengths, and boundary-errors at Re=40 in the coarse grid ($D=20\Delta x$). As the number of forcing increases, the boundary error decreases clearly. For the 4-point discrete delta function, the boundary-error is steeply reduced until 5 forcing times, and little change is observed at 10 and 20 forcing times. For the 2-point discrete delta function, the boundary-error is steeply reduced at 2 forcing times, and then it decreases slowly. In addition, by comparing boundary-errors between 2- and 4-point discrete delta functions for each forcing time, it is observed that the effect of implicit forcing iterations on the decrease in boundary-error is larger in the 4-point discrete delta function than in the 2-point discrete delta function. The reduction of the boundary-error leads to a better satisfaction of the no-slip condition inside the flow fields reconstructed by the discrete delta function interpolation. This can also be confirmed in the comparison of streamlines in Figure 16, where streamlines in the explicit diffuse interface schemes penetrate the

boundary of the cylinder, while those in the implicit diffuse interface schemes do not, which was also observed in [41]. For the 4-point discrete delta function, the drag coefficient and the recirculation length decrease similarly with the change of boundary-error as the number of forcing increases, although their reduction magnitudes are small, below 1% and 1.5 %, respectively. However, for the 2-point discrete delta function, inconsistent results are observed.

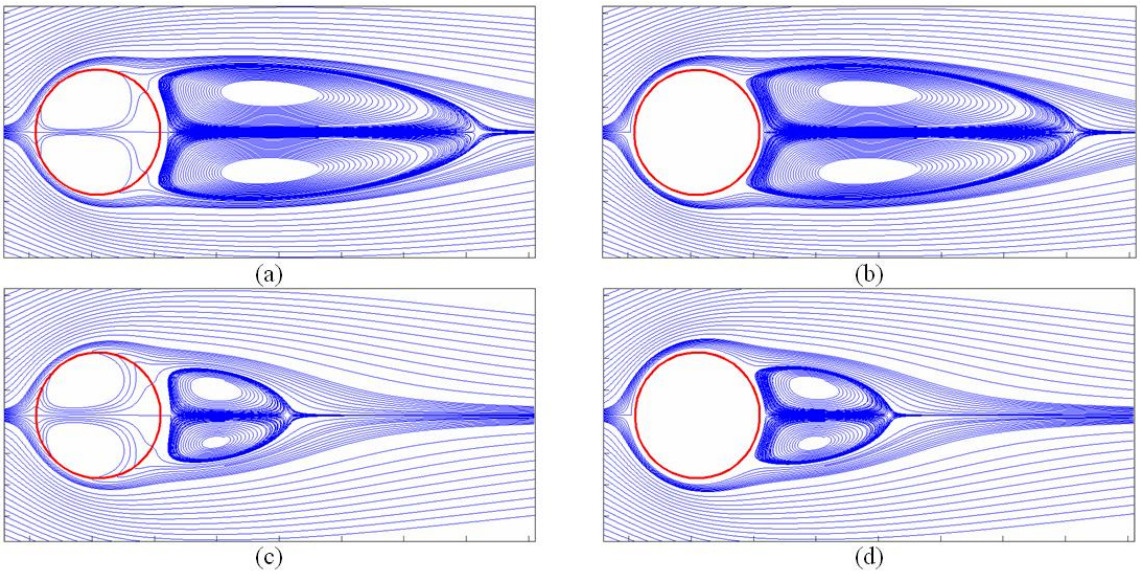

Figure 16. Streamlines when using (a) explicit diffuse forcing and (b) implicit diffuse forcing at Re=40 and (c) explicit diffuse forcing and (d) implicit diffuse forcing at Re=20. For all results, 4-point discrete delta functions were used in the coarse grid, and for implicit forcing methods, the number of forcing was 20.

Table 2 shows the effect of grid refinement on the drag coefficient and the recirculation length for each direct-forcing scheme. For the diffuse interface schemes,

the drag coefficient and the recirculation length decrease by 1% to 2% and 3% to 7%, respectively, as the grid size around the cylinder is halved. It is also observed that at Re=20, even for 4-point discrete delta function, the recirculation length is increased very slightly by the implicit forcing, while the drag coefficient still decreases. This result is consistent with the Taylor-Green vortex in that the implicit forcing increases the accuracy on the boundary points but does not ensure an increase in accuracy of the flow field, even for 4-point discrete delta functions. In contrast to the coarse grid, in the refined grid, the drag coefficient and the recirculation length are decreased by the implicit forcing for the 2-point discrete delta function at Re=40. Also, in this steady flow, the reduction of the drag coefficient obtained by adopting the sharper discrete delta function is larger than that obtained by adopting the implicit interface scheme instead of the explicit interface scheme.

Table 2. The effect of the grid refinement on the drag coefficient and the recirculation length at Re=20 and 40.

	C_D / L_w	Explicit diffuse (4-point delta)	Implicit diffuse (NF=20) (4-point delta)	Explicit diffuse (2 point-delta)	Implicit diffuse (NF=20) (2 point-delta)
Re=20	Uniform (D=20Δx)	2.125 / 1.021	2.119 / 1.033	2.098 / 0.981	2.102 / 0.999
	Refined (D=40Δx)	2.090 / 0.951	2.084 / 0.962	2.076 / 0.935	2.075 / 0.951
Re=40	Uniform (D=20Δx)	1.597 / 2.525	1.584 / 2.486	1.576 / 2.435	1.577 / 2.441
	Refined (D=40Δx)	1.572 / 2.398	1.560 / 2.360	1.560 / 2.352	1.555 / 2.340

From Tables 1 and 2, we find that at steady flows of Re=20 and 40, the implicit scheme enhances the accuracy on the boundary in the diffuse flow field reconstructed by

the discrete delta function as compared with the explicit scheme. However, it does not fundamentally improve the accuracy of the surface force (from the results of drag coefficients) and the neighboring flow field (from the results of recirculation lengths).

Table 3 shows the comparison of drag coefficients and recirculation lengths at Re=20 and 40 with other numerical results [9, 26, 38, 41, 89-92]. The explicit and implicit diffuse interface schemes have larger drag coefficients and recirculation lengths than the sharp interface scheme. This can be attributed to their diffuse nature. The results are consistent with those of previous diffuse forcing methods based on the NSE [26, 91] and the LBE [38, 41]. For the sharp interface scheme, the recirculation length lies within the range of data obtained from other body-fitted grid and sharp methods based on the NSE [89-90, 92], and the drag coefficient is slightly larger but within 1.3% of [92].

Table 3. Comparison of drag coefficients and recirculation lengths at Re=20 and 40. Present results were computed in the refined grid. The number of forcing in the implicit multi-direct-forcing methods is 20.

	Year	Characteristics	Re=20		Re=40	
			C_D	L_w	C_D	L_w
Niu et al. [38]	2006	Explicit diffuse forcing, LBE	2.144	0.95	1.589	2.26
Le et al. [25]	2008	Implicit diffuse direct-forcing, NSE	2.07	0.98	1.58	2.49
Wang et al. [19]	2009	Implicit diffuse direct-forcing, NSE	2.25	0.98	1.66	2.35
Wu and Shu [41]	2009	Implicit diffuse direct-forcing, LBE	2.091	0.93	1.565	2.31
Fornberg [89]	1980	Body-fitted grid, NSE	2.0	0.91	1.5	2.24
Park et al. [90]	1998	Body-fitted grid, NSE	2.01	-	1.51	-
Ye et al. [92]	1999	Cut cell method (sharp method), NSE	2.03	0.92	1.52	2.27
Kim et al. [9]	2001	Exterior sharp direct-forcing, NSE	-	-	1.51	-
Present	2009	Explicit diffuse direct-forcing (4-point), LBE	2.090	0.95	1.572	2.40
Present	2009	Explicit diffuse direct-forcing (2-point), LBE	2.076	0.94	1.560	2.35
Present	2009	Implicit diffuse direct-forcing (4-point), LBE	2.084	0.96	1.560	2.36
Present	2009	Implicit diffuse direct-forcing (2-point), LBE	2.075	0.95	1.555	2.34
Present	2009	Exterior sharp direct forcing, LBE	2.057	0.91	1.538	2.25

Figure 17 presents streamline and vorticity contours at Re=100 and 150 at a time instant resulting from the sharp interface scheme in the refined grid. It is clearly observed that each flow has vortex shedding in the wake. Figure 18 shows the time evolution of drag and lift coefficients. The periodicity of the vortex shedding is clearly revealed for each scheme, although its value is different depending on the interface scheme adopted.

Table 4 presents average drag coefficients, lift coefficients, and Strouhal numbers at Re=100 for the present method and other experiments and numerical methods [3, 9, 12, 25-26, 41, 84, 90, 92-94]. Regarding the effect that the choice of discrete delta functions has in the present diffuse interface schemes, 2-point delta functions reduce the drag coefficient by 1% from that of 4-point delta functions for the explicit diffuse interface schemes, while the drag coefficients change little for the implicit diffuse interface schemes. In addition, the effect of the implicit forcing begins to be clearly revealed in this flow. The drag coefficients in the implicit diffuse interface schemes decrease by 2.2% and 1.2% for the 4- and 2-point discrete delta functions, respectively, becoming closer to those of the sharp scheme than those in the explicit diffuse interface schemes. This is in contrast to the results of the steady flows given in Table 2, where the reduction of drag coefficients was small, below 0.5%, in the refined grid. However, even the implicit diffuse interface schemes show higher average drag coefficients than the exterior sharp interface scheme does. This trend is also observed in other diffuse and sharp numerical methods given in this table. However, the present explicit and implicit diffuse interface direct-forcing schemes show slightly lower drag

coefficients (but closer to those of sharp schemes) (2% to 5%) than other explicit diffuse forcing methods [3, 84] and implicit diffuse forcing methods do [25-26, 41]. Results from the present exterior sharp direct-forcing IB-LBM show a good agreement with the body-fitted methods [90, 93], other sharp direct-forcing methods [9, 12], and experiments [93-94] for the Strouhal number.

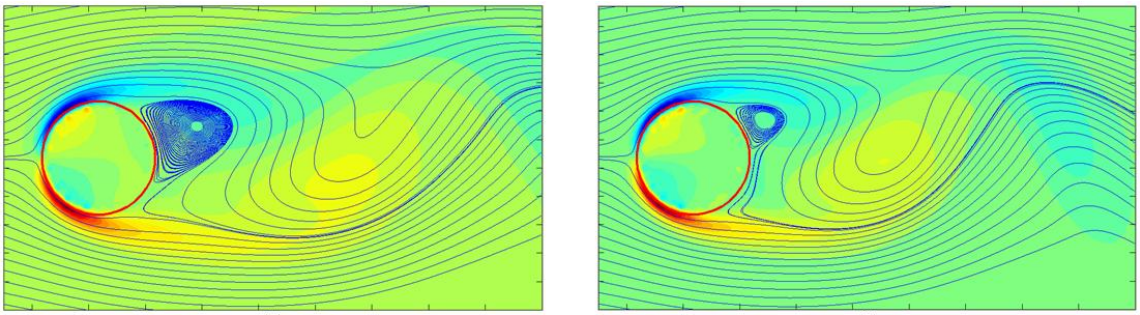

Figure 17. The instantaneous streamline and vorticity contours at (a) Re=100 and (b) 150 resulting from the sharp direct-forcing method in the refined grid (D=40Δx).

Table 4. Comparison of drag and lift coefficients and Strouhal number at Re=100. Present results were computed in the refined grid. The number of forcing in the implicit multi-direct-forcing methods is 20.

	Year	Characteristics	Avg. C_D	C_L	St
Roshko [93]	1953	Experiment	-	-	0.164
Williamson [94]	1989	Experiment	-	-	0.166
Lai and Peskin [3]	2000	Explicit diffuse Feedback-forcing, NSE	1.447	±0.330	0.165
Sui et al. [84]	2007	Explicit diffuse direct-forcing, LBE	1.438	±0.344	0.166
Su et al. [24]	2007	Implicit diffuse direct-forcing, NSE	1.40	±0.33	0.166
Le et al. [25]	2008	Implicit diffuse direct-forcing, NSE	1.39	±0.346	0.16
Wu and Shu [41]	2009	Implicit diffuse direct-forcing, LBE	1.364	±0.344	0.163
Kim et al. [9]	2000	Exterior sharp direct-forcing, NSE	1.33	±0.32	0.165
Choi et al. [12]	2007	Interior sharp direct-forcing, NSE	1.34	±0.315	0.164
Park et al. [90]	1998	Body-fitted method, NSE	1.33	±0.33	0.165
Liu et al. [93]	1998	Body-fitted method, NSE	1.35	±0.339	0.164
Present	2009	Explicit diffuse direct-forcing (4-point) , LBE	1.399	±0.343	0.162
Present	2009	Explicit diffuse direct-forcing (2-point), LBE	1.385	±0.345	0.163
Present	2009	Implicit diffuse direct-forcing (4-point), LBE	1.368	±0.346	0.162
Present	2009	Implicit diffuse direct-forcing (2-point), LBE	1.368	±0.346	0.163
Present	2009	Exterior sharp direct-forcing, LBE	1.336	±0.329	0.165

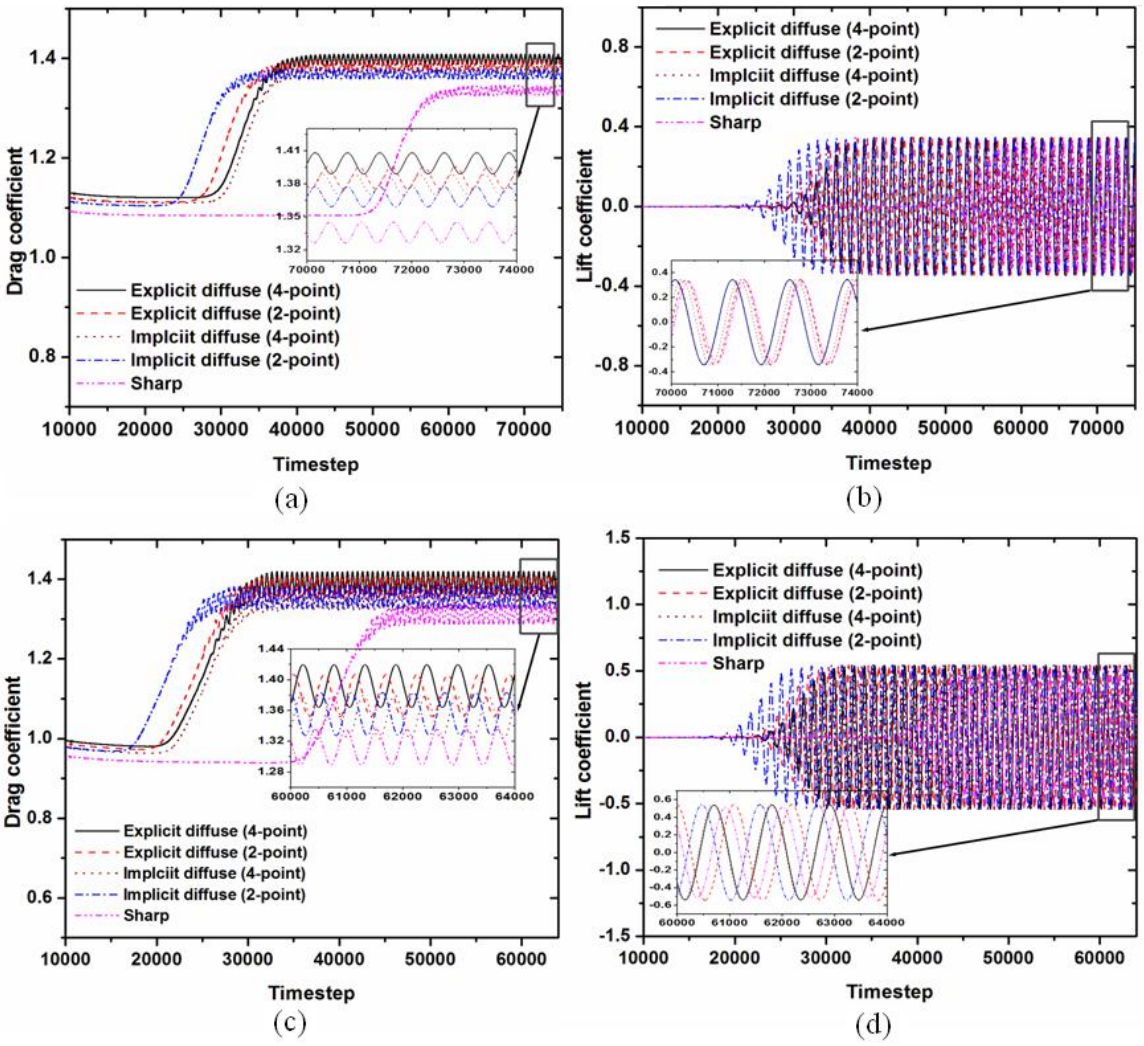

Figure 18. Time evolution of (a) drag and (b) lift coefficients at Re=100 and (c) drag and (d) lift coefficients at Re=150.

As can be seen in Table 5, the implicit forcing becomes more effective at Re=150 than at Re=100. The drag coefficients obtained from the implicit forcing schemes decrease by 2.9% and 1.7% for the 4- and 2-point discrete delta functions, respectively from those obtained from explicit forcing schemes. The present IB-LBMs also follow the trend that diffuse methods [3, 24] have higher average drag coefficients than sharp

methods do [90, 93]. The present explicit and implicit diffuse direct-forcing schemes also show slightly lower drag coefficients and Strouhal numbers (2% to 4%) than other explicit diffuse forcing schemes [3] and implicit diffuse forcing schemes do [24], respectively. The sharp direct-forcing method shows comparable results (within 2%) with other body-fitted grid methods [90, 93] and experiments [93-94].

Table 5. Comparison of drag and lift coefficients and Strouhal numbers at Re=150. Present results were computed in the refined grid. The number of forcing in the implicit multi-direct-forcing methods is.

	Year	Characteristics	Avg. C_D	C_L	St
Roshko [93]	1953	Experiment	-	-	0.182
Williamson [94]	1989	Experiment	-	-	0.183
Lai and Peskin [3]	2000	Explicit diffuse feedback-forcing, NSE	1.44	-	0.184
Su et al. [24]	2007	Implicit diffuse direct-forcing, NSE	1.39	-	0.187
Park et al. [90]	1998	Body-fitted grid, NSE	1.32 (Re=140) 1.32 (Re=160)	±0.482 (Re=140) ±0.550 (Re=160)	- -
Liu et al. [93]	1998	Body-fitted grid, NSE	1.334	±0.530	0.182
Present	2009	Explicit diffuse direct-forcing (4-point), LBE	1.392	±0.540	0.182
Present	2009	Explicit diffuse direct-forcing (2-point), LBE	1.379	±0.544	0.182
Present	2009	Implicit diffuse direct-forcing (4-point), LBE	1.351	±0.542	0.181
Present	2009	Implicit diffuse direct-forcing (2-point), LBE	1.355	±0.543	0.181
Present	2009	Exterior sharp direct forcing, LBE	1.312	±0.513	0.184

3. Flow past a sphere

To test the applicability of the immersed boundary method in the D3Q19 MRT-LBE, we consider flow past a sphere. Many studies on this problem have been implemented through experiments and numerical simulations [8-9, 12, 95-101]. Flow past a sphere is steady and axisymmetric at very low Reynolds numbers (steady axisymmetric regime where Re<200). With increasing Reynolds number, the flow loses the axisymmetry first (steady planar-symmetric regime, where $210 < \mathrm{Re} < 270$) and then the steadiness

(unsteady planar-symmetric regime, where $280 < \mathrm{Re} < 375$). It should be pointed out that the Reynolds numbers indicating the range of each regime slightly differ for different researchers.

We simulate various laminar flows past a sphere with Re=100, 150, 200, 250, and 300 belonging to the regimes mentioned above. A sphere of diameter D is located in the center of the computational domain of 10D×10D×20D. To reduce the number of nodes used in the calculation, we also consider the local grid refinement method [88]. As presented in Figure 19, two-level grid refinements are used in this simulation. In the figure, the region outside the red block consists of cubic cells with grid sizes of Δx=0.2D for Re=100, Δx=0.1D for Re=150 and 200, and Δx=0.0625 D for Re=250 and 300. The red block (3D×3D×6.5D), excluding the blue block, (level 1) has the halved grid size of the original size, and the blue block (2D×2D×5D) (level 2) has the halved grid size of the size in the level 1 block.

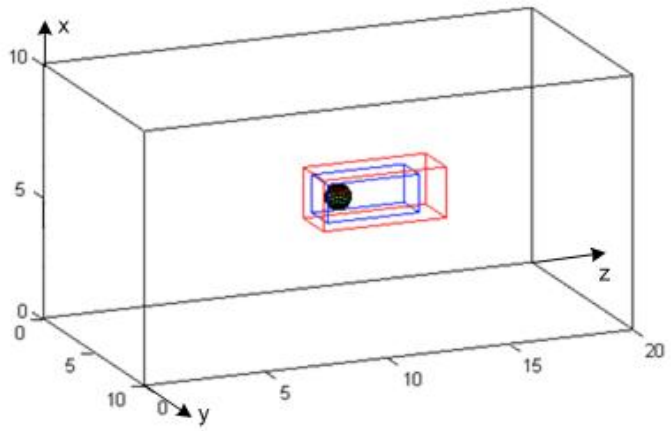

Figure 19. Two-level local grid refinement adopted in the simulation (Unit: D).

From the calculations, we obtained the steady results for Re=100, 150, 200, and 250 and the unsteady result for Re=300. Figure 20 presents steady streamlines past a sphere at Re=100, 150, 200, and 250 on yz-plane. The streamlines at Re=100, 150, and 200 show axisymmetric vortices behind a sphere with a larger size at a higher Reynolds number; however the axisymmetry of vortices is broken in the streamline at Re=250, although the flow remains steady.

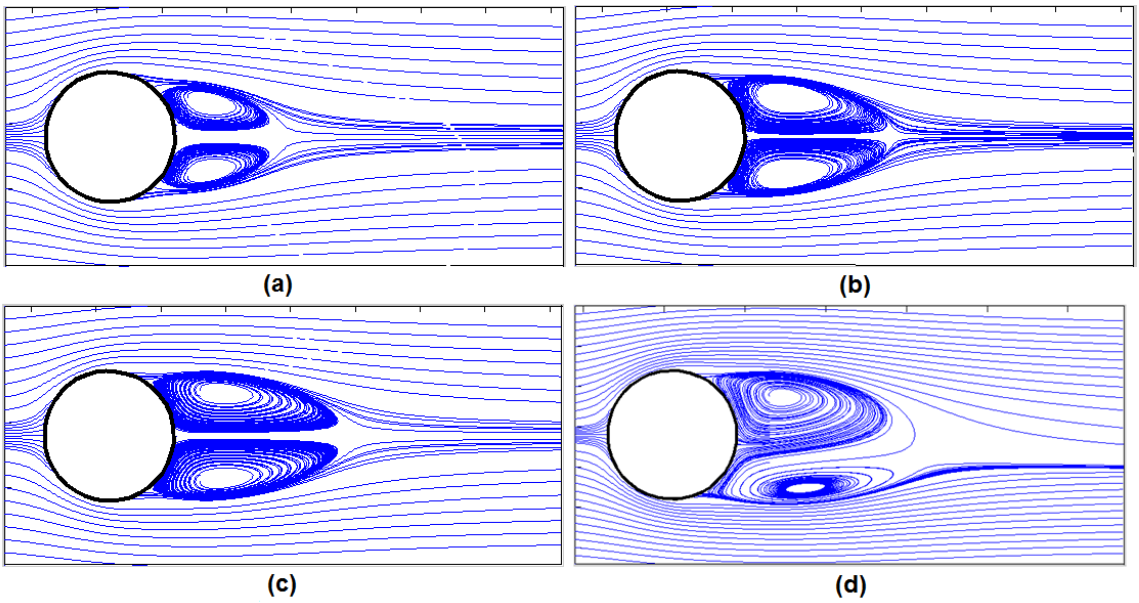

Figure 20. Streamlines past a sphere on the yz-plane at (a) Re=100, (b) Re=150, (c) Re=200, and (d) Re=250.

Figure 21 shows the vortical structure past a sphere at Re=300 at an instant time. Here, the vortical surfaces are obtained using the method of Jeong and Hussain [102]. It is observed that hairpin-like vortices are periodically shed in a fixed orientation, which is a characteristic of the unsteady planar-symmetric regime.

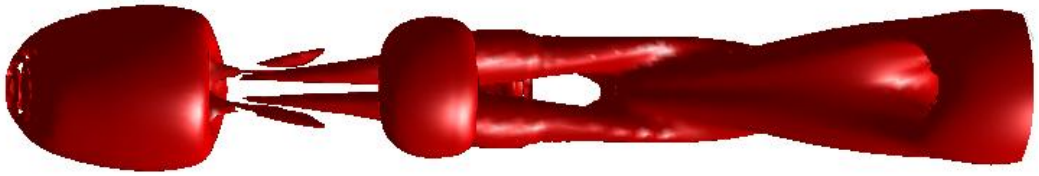

Figure 21. Vortical structure past a sphere at Re=300.

Table 6 presents drag coefficients, lift coefficients (for Re=250 and 300 only), and Strouhal numbers (for Re=300 only) from the present and other numerical calculations. They show a good agreement.

Table 6. Comparison of drag and lift coefficients and Strouhal numbers with other numerical experimental results for flow past a sphere.

Re	100	150	200	250		300		
	C_D	C_D	C_D	C_D	C_L	C_D	C_L	St
Apte et al. [95]	1.10	0.90	-	-	-	0.686	-	-
Mittal [96]	1.09	-	-	-	-	-	-	-
Mittal et al. [97]	1.08	0.88	-	-	-	0.68	-	-
Clift et al. [99]	1.09	0.89	-	-	-	0.684	-	-
Johnson and Patel [100]	1.09	0.9	-	0.70	0.062	0.656	0.069	0.137
Marella et al. [101]	1.06	-	-	-	-	0.621	-	-
Kim et al. [9]	1.087	-	0.815	0.701	0.059	0.657	0.067	0.134
Fadlun et al. [8]	1.0794	-	-	-	-	-	-	-
Constantinescu and Squires [98]	-	-	0.7683	0.70	0.062	0.655	0.065	0.136
Choi et al. [12]	1.09	-	-	0.70	0.052	0.658	-	-
Present	1.0802	0.889	0.7695	0.7048	0.0582	0.6577	0.0654	0.1336

In this section, we validated the IB-LBM based on the split-forcing D3Q19 MRT-LBE with the sharp interface scheme. Thus, it is expected that with the suitable turbulence models the present IB-LBM can be extended to various 3D applications including high-Reynolds-number flows.

4. Flow in the pebble channel

The packed (or pebble) bed has been broadly used in chemical catalytic reactors due to its high potential for the enhancement of heat and mass transfer [103-106]. In the nuclear engineering field, the pebble bed reactor (PBR) has been considered as a type of very high temperature reactor (VHTR) [107]. In the PBR, the helium gas is used as coolant and the flow has a relatively high Reynolds number under high pressure and temperature conditions. From the viewpoint of nuclear safety, the heat transfer in the reactor core, especially the maximum fuel temperature and its location should be predicted. Therefore, the local flow field analysis in the reactor is required.

One of the issues of CFD calculation in the PBR geometry is mesh generation near the contacting point between pebbles because meshes with high resolution and quality are required in the region. To circumvent this problem, many researchers assumed narrow gaps between adjacent pebbles [108-111]. However, Lee et al. [111] showed that the assumption of a narrow gap could distort the flow and heat transfer phenomena. They also showed that point or area contacts have similar flow and heat transfer. Here, it needs to be pointed out that the concept of the area contact is physically

reasonable because the real pebbles are touching with some area, not a point, due to their weights.

In this section, we apply the direct-forcing IB-LBM based on the split-forcing MRT-LBE with the sharp interface scheme to the flow in a pebble channel where pebble-pebble and pebble-wall contacts exist as shown in Figure 22. Calis et al. [108] measured pressure drop for this case. This is a preliminary study for the simulation of real flow through a pebble bed in PBR, where the flow is turbulent and pebbles are randomly distributed.

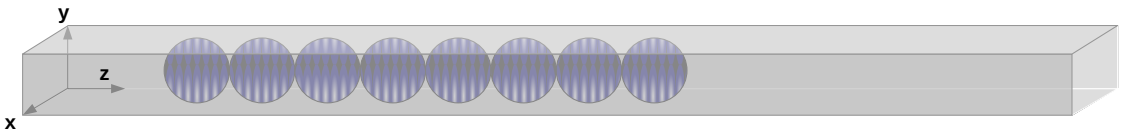

Figure 22. Geometry of the pebble channel.

In the application of the current method to this problem, we also face the problem of treatment of the region near the contacting point. However, this is now not a mesh generation problem but an interpolation problem in the IB-LBM.

When we applied the sharp interface scheme to the external flow past a sphere in Section D.3, the forcing node inside the solid had at least a neighboring node outside the boundary in a certain direction for the interpolation because all computational nodes just outside the solid boundary are fluid nodes. The selection of tri-linear, bilinear, and linear interpolations depended on the number of available neighboring nodes outside the solid boundary. However, in this pebble-channel problem, some of such nodes outside the

solid boundary may be no longer fluid nodes because other solid objects such as wall or pebbles are toughing at some region. Therefore, the level of interpolation of forcing nodes may degenerate. For example, the forcing node with a tri-linear interpolation available without contacts may become only possible for the bi-linear or linear interpolations with contacts. Especially, the forcing node with only a linear-interpolation available can be no longer a forcing node around the solid contacts. In this case, no forcing is imposed in the region around the contacting point, thus automatically having area contacts.

With this methodology, we simulate the laminar flows in a pebble channel with Reynolds numbers of 100, 200, and 300. We consider the case that aligned eight contacting pebbles with diameters of 12.7 mm located in the square channel with a channel-to-pebble diameter ratio of 1, i.e., with a simple cubic (SC) structure, as shown in Figure 22. The porosity (ε) in this geometry is 0.4764. The channel length is set to 200 mm. Uniform velocity and pressure boundary conditions are imposed at inlet and outlet, respectively. The halfway bounce-back scheme [39] is adopted for the channel wall, and the immersed boundary method is used for the pebble boundary. The grid size is set to $\Delta x=D/80$.

We first compare the friction factors. The friction factor, i.e., pressure drop, is also very important parameter in the PBR design. In the PBR design calculation, the friction factor based on superficial velocity (u_0) and pebble diameter D_p, which is defined by

$$f_D \equiv \frac{\Delta P}{L}\frac{D_p \varepsilon^3}{\rho u_0^2 (1-\varepsilon)}, \tag{79}$$

is usually adopted. Many correlations are developed for this friction coefficient. Among them, the following correlations are prevalently used [112-113]:

$$f_D^{Ergun} = 150\,\mathrm{Re}_\varepsilon^{-1} + 1.75 \tag{80a}$$

$$f_D^{KTA} = 160\,\mathrm{Re}_\varepsilon^{-1} + 3\,\mathrm{Re}_\varepsilon^{-0.1} \tag{80b}$$

where Re_ε is the modified Reynolds number defined by

$$\mathrm{Re}_\varepsilon \equiv \frac{u_0 D_p}{\nu(1-\varepsilon)}. \tag{81}$$

In contrast, the friction factor provided in the Calis et al.'s experiment is defined by

$$f_{D,Calis} = \Delta P\left(\frac{D_h}{L}\right)\frac{1}{\rho u^2/2} \tag{82}$$

where D_h is a hydraulic diameter considering not only pebble surfaces but also channel walls, and u is not a superficial velocity (u_0) but a mean velocity in the pebble channel with the relation of $u = u_0/\varepsilon$. The following relations between two correlations and their parameters hold:

$$D_h = k\frac{\varepsilon}{1-\varepsilon}D_p, \tag{83a}$$

$$\mathrm{Re}_h = \frac{u\ D_h}{\nu} = \frac{(u_0/\varepsilon)kD_p\varepsilon/(1-\varepsilon)}{\nu} = k\frac{u_0 D_p}{\nu\ (1-\varepsilon)} = k\,\mathrm{Re}_\varepsilon, \tag{83b}$$

$$\begin{aligned} f_{D,Calis} &= \Delta P\left(\frac{D_h}{L}\right)\frac{1}{\rho u^2/2} = \Delta P\frac{kD_p\varepsilon/(1-\varepsilon)}{L}\frac{1}{\rho(u_0/\varepsilon)^2/2} = \frac{\Delta P}{L}\frac{D_p}{\rho u_0^2}\frac{\varepsilon^3}{1-\varepsilon}2k \\ &= 2k\ f_D \end{aligned} \tag{83c}$$

where k changes depending on whether the channel walls are considered in the hydraulic diameter calculation, as in the following:

$$k = \begin{cases} \dfrac{2}{3} & \text{with considering channel walls} \\ \dfrac{2}{3}\left(\dfrac{\pi}{\pi+4}\right) & \text{without considering channel walls} \end{cases} . \tag{83d}$$

As a result, the above correlations (80a) and (80b) can be written for the friction factor adopted in Calis et al.'s experiment as

$$f_{D,Calis}^{Ergun} = \begin{cases} 133\,\mathrm{Re}_{h1}^{-1} + 2.33 \\ 25.8\,\mathrm{Re}_{h2}^{-1} + 1.37 \end{cases} \tag{84a}$$

$$f_{D,Calis}^{KTA} = \begin{cases} 142\,\mathrm{Re}_{h1}^{-1} + 3.84\,\mathrm{Re}_{h1}^{-0.1} \\ 27.5\,\mathrm{Re}_{h2}^{-1} + 1.56\,\mathrm{Re}_{h1}^{-0.1} \end{cases} \tag{84b}$$

where h_1 and h_2 indicate the hydraulic diameter without and with considering the channel walls, respectively.

Figure 23 presents the comparison of friction factors between the present method, experiment [108], and correlations (84a) and (84b). The results from the present method and the experiment show a good agreement. However, the results from the correlations (with and without considering wall channel in the calculation of hydraulic diameter) show big differences. This may be because the correlations were derived from the experiment with large channel-to-particle diameter ratio, and the particles were randomly distributed.

Figure 24 presents velocity vector field in the yz-plane with x=D/4 at Re=200. Symmetric vortices behind each pebble are clearly captured. Similar symmetric vortices are also observed at Re=100 and 300.

Figures 25 and 26 present streamwise and lateral velocities, respectively, in the xy-planes at the center of the 4-th pebble (z_4), the right-end (z_4+r), and the middle of two points (z_4+0.5r) where r is the radius of the pebble. In Figure 25, we can observe that the steamwise velocity contours at Re=300 have distorted shape compared with those at Re=100. Here, the center circles in Figure 25(e) and (f) indicate the area contact regions generated by using the sharp interface scheme without modification. In Figure 26, we consider only the quarter section of the entire area to see the details. It is observed that more vortices are generated at higher Reynolds numbers. The existence of these vortices also explains the distorted shapes of the streamwise velocity contours at Re=300 in Figure 25.

Figure 27 shows the comparison of the lateral velocity vector field at Re=100 between the current calculation with area contact and the narrow gap approach, where pebbles with diameter of 0.98D were considered under the same other conditions. Two lateral velocity fields are similar, but the vortex center obtained from the narrow gap approach is slightly closer to the center region. The penetration of vortices can be attributed to no blockage at the center region in the narrow gap approach. It should be noted that, as discussed above [111], in the turbulent case, the narrow gap approach could produce quite different results from the contacting approach unlike these laminar flow results.

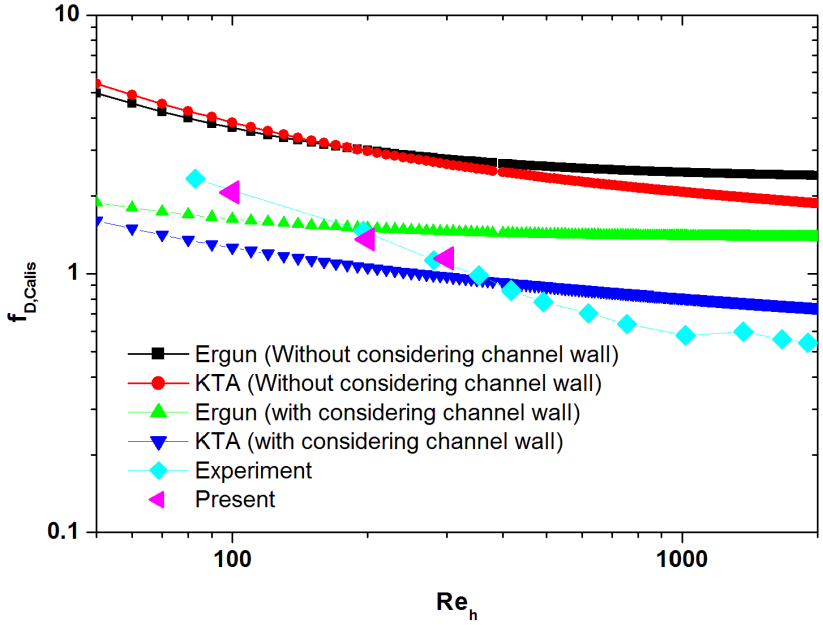

Figure 23. Comparison of friction factors in the pebble channel.

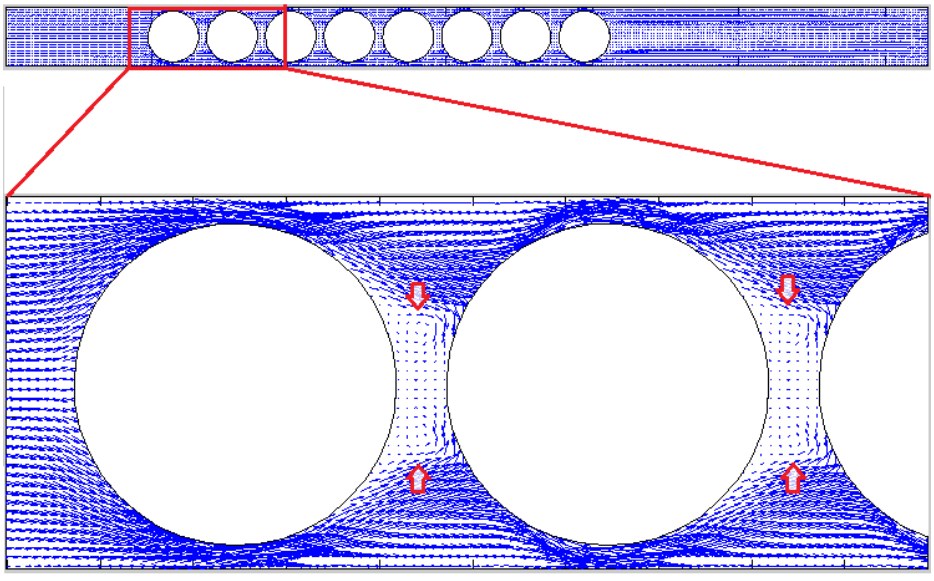

Figure 24. Velocity vector field on the yz-plane with x=D/4 at Re=200.

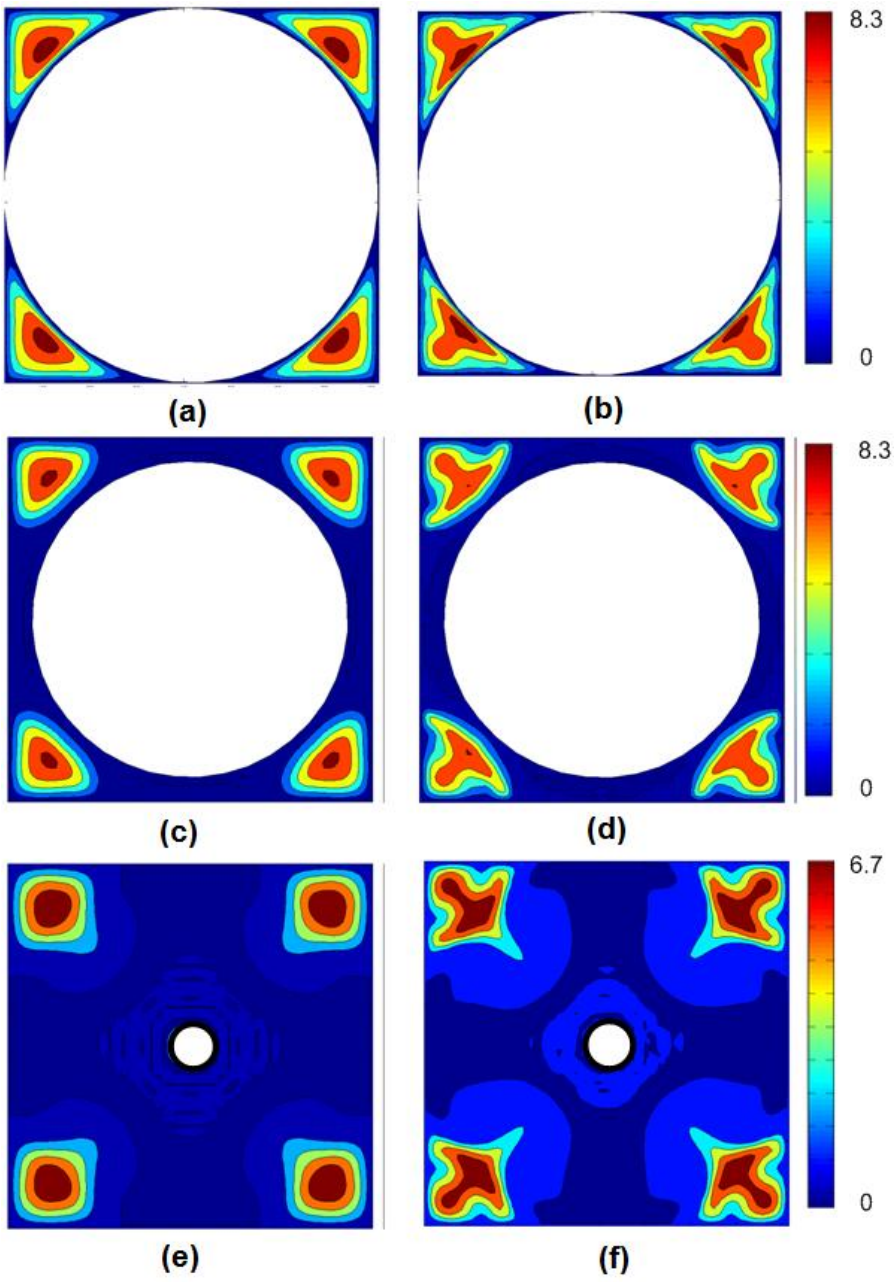

Figure 25. Streamwise velocity contours in the xy-plane: (a) Re=100 and (b) Re=300 $z=z_4$, (c) Re=100 and (d) Re=300 at $z=z_4+0.5r$, and (e) Re=100 and Re=300 at $z=z_4+r$. The velocity normalized by inlet velocity was used.

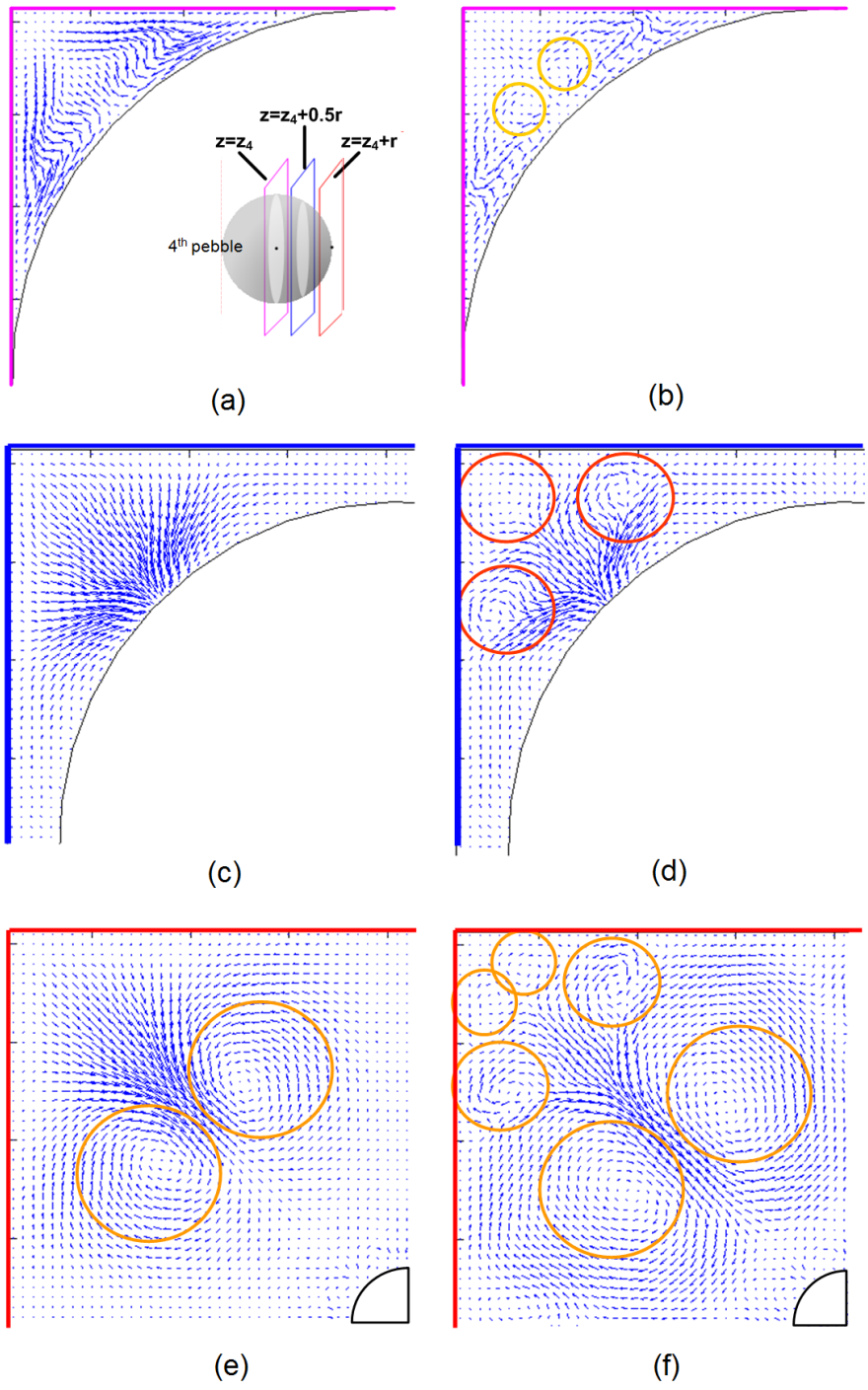

Figure 26. Lateral velocity vector fields in the quarter section of the xy-plane: (a) Re=100 and (b) Re=300 $z=z_4$, (c) Re=100 and (d) Re=300 at $z=z_4+0.5r$, and (e) Re=100 and Re=300 at $z=z_4+r$. Circles indicate the vortices.

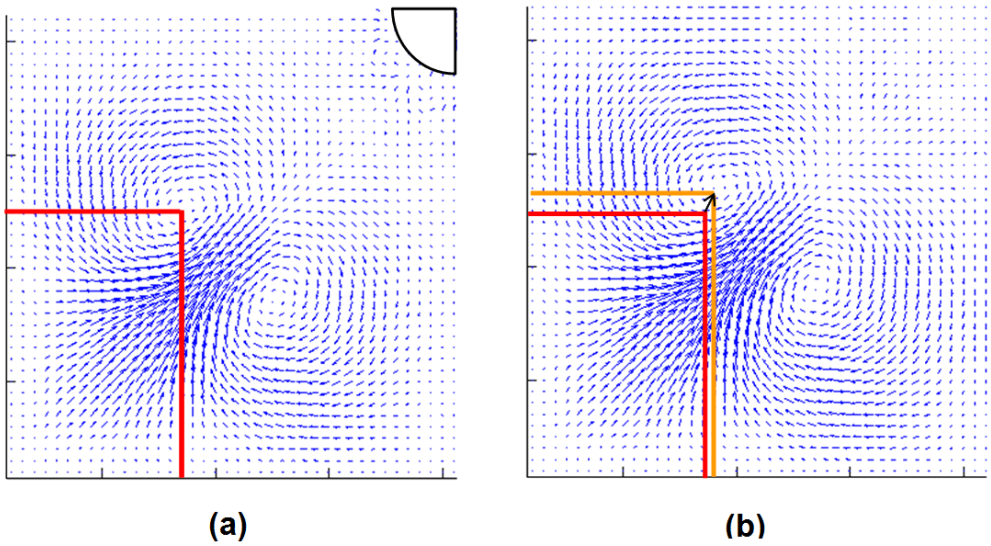

Figure 27. Comparison of lateral velocity vector fields between (a) area contacting due to the immersed boundary method and (b) the narrow gap approach.

E. Conclusions

In this chapter, the direct-forcing formula based on the explicit-type split-forcing LBE [42, 114] was derived, and under the formula, various interface schemes were assessed through flow problems with stationary complex boundaries.

By simulating the Taylor-Green decaying vortex, the direct-forcing IB-LBM with diffuse and sharp interface schemes turned out to have a second-order overall accuracy. In the problem, by comparing the diffuse interface schemes with 2-point and 4-point discrete delta functions and the exterior sharp interface scheme, we found that better accuracy is attained as a narrower interface of forced points is adopted. In the simulation of flows past a circular cylinder with various Reynolds numbers, the present IB-LBM with each interface scheme showed comparable results with other experiments

and numerical methods. For the implicit forcing effect, the boundary accuracy was consistently enhanced as the number of forcing increased in both problems, whereas the flow field accuracy did not show enhancement of the accuracy; this could be explained by the fact that the scheme still has the diffuse nature due to the adoption of discrete delta functions. Thus, the IB-LBM with a sharp interface scheme is recommended for the complex boundary problem in order to obtain more accurate results.

The direct-forcing IB-LBM based on the D3Q19 MRT-LBE with the sharp interface scheme was validated through the flow past a sphere and then successfully applied to the laminar flow in a pebble channel, which is a preliminary study for real flows in the PBR.

CHAPTER IV

IMMERSED BOUNDARY-LATTICE BOLTZMANN METHOD FOR MOVING BOUNDARY PROBLEMS

A. Introduction

The suitable treatment of moving boundary problems is one of challenging issues in the computational fluid dynamics (CFD) field. Various numerical methods have been developed for these problems. The methods for describing complex moving solid boundaries can be classified into three major groups depending on the "grid structure" used [115]: body-conformal grid approach, overset grid approach (or chimera method), and non-body-conformal grid approach.

In the body-conformal grid approach, the no-slip boundary condition is easily satisfied. However, as a solid boundary moves, the mesh should be deformed and adapted to the fluid-solid boundary. The Arbitrary Lagrangian Eulerian (ALE) method [116-117] is an example of this concept. The ALE method is based on a moving unstructured mesh. This technique has a relatively good accuracy if care is taken during the mesh adaptation. However, frequent re-meshing requires a considerable computational cost.

In the overset grid approach, we use a combination of moving and stationary grids, where the surroundings are described on a stationary grid and a moving grid is attached to the object. In this approach, boundary conditions on the object can be set easily; however, a major disadvantage is associated with numerical problems in the

information transport between the grids, i.e., the interpolation procedure implies reduced computational efficiency and, potentially, accuracy of the numerical scheme as compared with a single (stationary) grid approach. In addition, this approach would be unsuited when considering particles or deformable objects [115].

In the non-body-conformal grid approach, including the immersed boundary method, we usually use the fixed Cartesian grid. Instead of adapting the mesh to the boundary, the momentum source term is adopted to describe the boundary effect. This approach avoids the inefficiency of frequent re-meshing in body-fitted grid approach and broader applications to moving boundary problems compared with the overset grid approach. Besides, the forces acting on the object will be directly available; therefore, studies of fluid-structure or fluid-particle interactions are more straightforward.

To properly describe the moving solid boundary problems, two-way interactions from solid to fluid and from fluid to solid should be considered. The effect of the solid to fluid is realized by the no-slip boundary condition, which is attained by the boundary forcing term in the IBM, and the effect of the fluid to solid is represented by the surface force exerted on the solid by the fluid. The surface force usually plays a role as a source term of movement of solid objects. For example, the surface force exerted on the particle is a source term of the governing equation of particle motion (Newton's equation of motion) in particle-fluid two-phase flows, and the surface force on the cylinder is a source term of the equation governing flow-induced-vibration (FIV) of elastically mounted cylinders (e.g., the mass-spring-damper equation).

In this chapter, we apply the present IB-LBM to various moving boundary problems. In many cases, we compare both the sharp and diffuse interface schemes simultaneously to understand the advantages and disadvantages of each scheme and find suitable combinations of the direct-forcing formula and the interface scheme for moving boundary problem simulations. In the validation and application processes, we are required to evaluate the surface force and solve additional equations of solid motions. Therefore, we begin with a discussion of the calculation models related to surface force evaluation, Newton's equation of motions, particle-particle and particle-wall collision models in Section B. Then, in Section C, we simulate various moving boundary problems. We first consider a problem of flow induced by inline oscillation of a circular cylinder (Section C.1) since both experimental and body-conformal grid method results are available for this problem [118]. Then, to check the applicability of the present IB-LBM to moving boundary problems, we consider the following problems in this chapter:

- Sedimentation of 2D (single, double, and multiple) particles (Sections C.2, C.3, C.4, and C.5);
- Particle behaviours in the channel with holes (Section C.6);
- Sedimentation of spherical particles (Sections C.7);
- Flow-induced vibration of the cylinder (Section C.8).

B. Calculation Models

To realize the two-way coupling of solid-fluid interactions, an accurate surface force evaluation, as well as an accurate application of the no-slip condition, is very important

because the surface force determines the displacement of moving solid objects (structures or particles). In the context of the IBM, the surface force exerted on the object by the fluid is directly evaluated by the summation of the boundary force with added mass force. This is shown in Section B.1. Also, to simulate the particle motion in the particle-fluid flows, Newton's equations of (translational and angular) motions are required. The surface force exerted on the particle surface by the fluid is coupled with the Newton's equation of motions as a source term. This is discussed in Section B.2. In addition, since particles not only interact with the fluid but also collide with the wall and other particles in usual multiple particle-fluid flows, we need particle-particle and particle-wall collision models. We adopt simple repulsive models based on the distance between particles or between particle and wall in this study. The details are given in Section B.3.

1. Surface force evaluation in the immersed boundary method

In the IBM, the surface force exerted on the solid can be easily evaluated. As depicted in Figure 28, we can consider two control surfaces (S_s and S_f), which can vary with time, and the resulting control volumes surrounded by the two control surfaces (V, V_s, and V_f) in the fluid field. For the control surface S_s, the force from the fluid outside the surface (V_f) to the surface (S_s) can be expressed as:

$$\mathbf{F}_{f\to s} = -\int_{S_s} [\rho \mathbf{u}(\mathbf{u}-\mathbf{u}_s)] + \boldsymbol{\sigma}] \cdot \mathbf{n}_s dS \tag{85}$$

where $\mathbf{u}_s$ is a boundary velocity of the control surface S_s, $\boldsymbol{\sigma}$ is the viscous stress tensor,

and $\mathbf{n}_s$ is its outward surface vector. If the S_s is an impermeable surface, i.e., no flow passes through the surface, then $\mathbf{u} = \mathbf{u}_s$, so that Equation (85) becomes

$$\mathbf{F}_{f\to s} = -\int_{S_s} \boldsymbol{\sigma}\cdot\mathbf{n}_s dS \,. \tag{86}$$

For the control volume V_f surrounded by the control surfaces S_s and S_f, from the Cauchy's stress principle, the linear momentum balance can be written as:

$$\frac{\partial}{\partial t}\int_{V_f} \rho\mathbf{u}\, dV = -\int_{S_f} [\rho\mathbf{u}(\mathbf{u}-\mathbf{u}_f)+\boldsymbol{\sigma}]\cdot\mathbf{n}_f\, dS - \int_{S_s} [\rho\mathbf{u}(\mathbf{u}-\mathbf{u}_s)+\boldsymbol{\sigma}]\cdot(-\mathbf{n}_s)\, dS \tag{87}$$

where $\mathbf{n}_f$ is the outward surface vector of S_f. Using Equation (86), we can rewrite Equation (87) as:

$$\frac{\partial}{\partial t}\int_{V_f} \rho\mathbf{u}\, dV = -\mathbf{F}_{f\to s} - \int_{S_f} [\rho\mathbf{u}(\mathbf{u}-\mathbf{u}_f)+\boldsymbol{\sigma}]\cdot\mathbf{n}_f\, dS \,. \tag{88}$$

For the control volume V covering both V_f and V_s, when boundary forces exist, the linear momentum balance can be expressed as:

$$\frac{\partial}{\partial t}\int_{V_f\cup V_s} \rho\mathbf{u}\, dV = -\int_{S_f} [\rho\mathbf{u}(\mathbf{u}-\mathbf{u}_f)+\boldsymbol{\sigma}]\cdot\mathbf{n}_f\, dS + \int_{V_f\cup V_s} \mathbf{F}\, dV \tag{89}$$

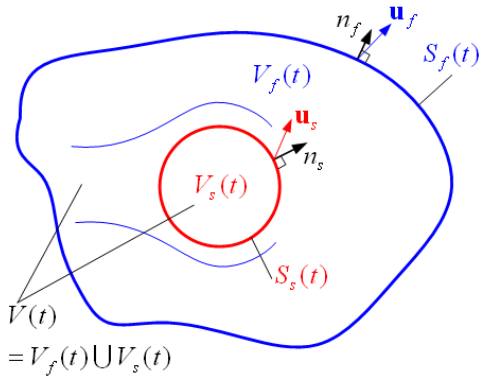

Figure 28. Two time-varying control surfaces (S_s and S_f) and corresponding control volumes (V, V_s, and V_f) in fluid domain.

Subtracting Equation (88) from Equation (89), we can obtain the following surface force formula:

$$\mathbf{F}_{f\to s} = \frac{\partial}{\partial t}\int_{V_s} \rho \mathbf{u}\, dV - \int_{V_f \cup V_s} \mathbf{F} dV \tag{90}$$

where the first term in the left-hand side indicates the added mass effect. If the volume V_s is a rigid solid body with a center-of-mass velocity of $\mathbf{U}_c$, the first term in Equation (90) can be replaced by $\rho V_s \partial \mathbf{U}_c / \partial t$ [23, 119-120], and thus Equation (90) becomes

$$\mathbf{F}_{f\to s} = \rho V_s \frac{\partial \mathbf{U}_c}{\partial t} - \int_{V_f \cup V_s} \mathbf{F} dV\,, \tag{91}$$

which will be used in this chapter for the surface force evaluation of the solid body with acceleration. For reference, if the rigid solid body is fixed or moving with a constant velocity, Equation (91) simply becomes

$$\mathbf{F}_{f\to s} = -\int_{V_f \cup V_s} \mathbf{F}\, dV\,. \tag{92}$$

From Equations (90), (91), and (92), we can find that if the immersed boundary force is exact, the surface force on the solid body can be directly calculated by integrating (or summing in a discrete sense) the boundary force terms regardless of the positions (inside or outside the boundary). This is one of the advantages of using the IBM. Specifically, for the sharp interface schemes, since dV directly matches with cubic cell volume (area in 2D), the term can be evaluated using [3, 9]:

$$\int_{V_f \cup V_s} \mathbf{F}\, dV \approx \sum_{i,j} \mathbf{F}_{i,j} \Delta x^2 \quad \text{in 2D} \tag{93a}$$

$$\int_{V_f \cup V_s} \mathbf{F}\, dV \approx \sum_{i,j,k} \mathbf{F}_{i,j,k} \Delta x^3 \quad \text{in 3D} \tag{93b}$$

In contrast, for diffuse interface schemes, the term can be calculated by [20]

$$\int_{V_f \cup V_s} \mathbf{F}\, dV \approx \sum_b \mathbf{F}_b \Delta S_b\, \Delta x \tag{94}$$

where ΔS_b is the area (arc length in 2D) of the surface boundary at a forcing point b.

2. Newton's equation of motion for moving particle

For the simulation of a moving particle, we have to consider the motion equations of the particle. The Newton's equation of translational particle motion is

$$M_s \frac{d\mathbf{U}_c}{dt} = -\int_S \boldsymbol{\sigma} \cdot d\mathbf{S} + (\rho_s - \rho_f) V_s \mathbf{g} \tag{95}$$

where $\mathbf{U}_c$ is the center-of-mass velocity of the particle; M, S, V, and ρ are mass, surface, volume, and density, respectively; and subscripts f and s indicate the fluid and the solid, respectively. The first term in the right-hand side of Equation (95) indicates the force from fluid to solid, which consists of stationary surface force and added mass force due to acceleration. From Equations (91), (92), (93), and (94), the stationary surface force can be expressed in terms of the boundary forcing based on Equation (65). Hence,

$$-\int_S \boldsymbol{\sigma} \cdot d\mathbf{S} = -\int_V \mathbf{F}_b dV + \frac{\partial}{\partial t}\int_V \rho_f \mathbf{u} dV = -\int_V \mathbf{F}_b dV + M_f \frac{d\mathbf{U}_s}{dt}. \tag{96}$$

On the other hand, Newton's equation of angular particle motion is

$$I_s \frac{d\boldsymbol{\Omega}_c}{dt} = -\int_S (\mathbf{X}_w - \mathbf{X}_c) \times \boldsymbol{\sigma} \cdot d\mathbf{S} \tag{97}$$

where $\boldsymbol{\Omega}_c$ is the angular velocity of the particle, I_s is the moment of inertia, and $\mathbf{X}_w$ and $\mathbf{X}_c$ are position vectors of a wall surface and the center. Equation (97) can be rewritten in

terms of boundary forcing as

$$I_s \frac{d\mathbf{\Omega}_c}{dt} = -\int_V (\mathbf{x}_b - \mathbf{X}_c) \times \mathbf{F}_b dV + \frac{\partial}{\partial t} \int_V (\mathbf{X}_w - \mathbf{X}_s) \times \rho_f \mathbf{u}\, dV$$
$$= -\int_V (\mathbf{x}_b - \mathbf{X}_c) \times \mathbf{F}_b dV + I_f \frac{\partial \mathbf{\Omega}_c}{\partial t} \tag{98}$$

where $I_f = M_f R_s^2$ in 2D and $I_f = 2M_f R_s^2 / 5$ in 3D. As a result, the discretized Newton's equations of motion corresponding to Equations (96) and (98) can be written as

$$\mathbf{U}_c^{n+1} = \mathbf{U}_c^n + (1/M_s)[-\sum_b \mathbf{F}_b^n \Delta V_b + (M_s - M_f)\mathbf{g}]\Delta t + (M_f / M_s)(\mathbf{U}_c^n - \mathbf{U}_c^{n-1}) \tag{99}$$

and

$$\mathbf{\Omega}_c^{n+1} = \mathbf{\Omega}_c^n + (1/I_s)[-\sum_b (\mathbf{x}_b - \mathbf{X}_c) \times \mathbf{F}_b^n \Delta V_b]\Delta t + (I_f / I_s)(\mathbf{\Omega}_c^n - \mathbf{\Omega}_c^{n-1}), \tag{100}$$

respectively. Here, translational and angular acceleration terms are discretized based on current (n) and previous time steps (n-1), as in [120]. The center position at n+1 time step can be expressed as:

$$\mathbf{X}_c^{n+1} = \mathbf{X}_c^n + 0.5(\mathbf{U}_c^{n+1} + \mathbf{U}_c^n)\Delta t. \tag{101}$$

Thus, the wall velocity on the forcing point $\mathbf{X}_w$ at the next time step can be evaluated as

$$\mathbf{U}_w^{n+1} = \mathbf{U}_s^{n+1} + \mathbf{\Omega}_s^{n+1} \times (\mathbf{X}_w - \mathbf{X}_c), \tag{102}$$

and using this velocity and the direct-forcing formula, we can obtain the next time step boundary force $\mathbf{F}_b^{n+1}$ at forcing nodes.

3. Particle-particle and particle-wall collision models

In the simulation of particles in the fluid, collision models are required to prevent particles from penetrating into other particles or wall. Following the repulsive force

model for particle-particle collisions in [20], the repulsive force on the *i*-th particle from the *j*-th particle is

$$\mathbf{F}_{ij}^{p}=\begin{cases} 0, & \| \mathbf{x}_i-\mathbf{x}_j \| > R_i+R_j+\varsigma \\ \dfrac{c_{ij}}{\varepsilon_B}\left(\dfrac{\| \mathbf{x}_i-\mathbf{x}_j \|-R_i-R_j-\varsigma}{\varsigma}\right)^2 \dfrac{\mathbf{x}_i-\mathbf{x}_j}{\| \mathbf{x}_i-\mathbf{x}_j \|}, & R_i+R_j < \| \mathbf{x}_i-\mathbf{x}_j \| \le R_i+R_j+\varsigma \\ \left[\dfrac{c_{ij}}{\varepsilon_B}\left(\dfrac{\| \mathbf{x}_i-\mathbf{x}_j \|-R_i-R_j-\varsigma}{\varsigma}\right)^2 + \dfrac{c_{ij}}{E_B}\dfrac{R_i+R_j-\| \mathbf{x}_i-\mathbf{x}_j \|}{\varsigma}\right] \dfrac{\mathbf{x}_i-\mathbf{x}_j}{\| \mathbf{x}_i-\mathbf{x}_j \|}, & \| \mathbf{x}_i-\mathbf{x}_j \| \le R_i+R_j \end{cases} \quad (103)$$

where **x** is the particle center position vector; the parameter c_{ij} is the force scale which is chosen to be the buoyancy force on the body; ε_B is the stiffness parameter for particle-particle collisions; E_B is also the stiffness parameter but has the higher value than ε_B; R is the radius of particle; and ζ is the range of the repulsive force. Here, subscripts i and j, except those in c_{ij}, indicate i-th and j-th particles.

For particle-wall collisions, the repulsive force on the i-th particle by the wall collision is similarly given as [20]:

$$\mathbf{F}_{i}^{w}=\begin{cases} 0, & \| \mathbf{x}_i-\mathbf{x}_{i,j} \| > 2R_i+\varsigma \\ \dfrac{c_{ij}}{\varepsilon_W}\left(\dfrac{\| \mathbf{x}_i-\mathbf{x}_{i,j} \|-2R_i-\varsigma}{\varsigma}\right)^2 \dfrac{\mathbf{x}_i-\mathbf{x}_{i,j}}{\| \mathbf{x}_i-\mathbf{x}_{i,j} \|}, & 2R_i < \| \mathbf{x}_i-\mathbf{x}_{i,j} \| \le 2R_i+\varsigma \\ \left[\dfrac{c_{ij}}{\varepsilon_W}\left(\dfrac{\| \mathbf{x}_i-\mathbf{x}_{i,j} \|-2R_i-\varsigma}{\varsigma}\right)^2 + \dfrac{c_{ij}}{E_W}\left(\dfrac{2R_i-\| \mathbf{x}_i-\mathbf{x}_{i,j} \|}{\varsigma}\right)\right] \dfrac{\mathbf{x}_i-\mathbf{x}_{i,j}}{\| \mathbf{x}_i-\mathbf{x}_{i,j} \|}, & \| \mathbf{x}_i-\mathbf{x}_{i,j} \| \le 2R_i \end{cases} \quad (104)$$

where $\mathbf{x}_{i,j}$ is the position of a fictitious particle $P_{i,j}$, which is located symmetrically on the

other side of the wall, and ε_W and E_W are stiffness parameters for wall-particle collisions. Then, the total collision repulsive force exerted on the *i*-th particle by other particles and the wall, $\mathbf{F}_i^c$, can be expressed as:

$$\mathbf{F}_i^c = \sum_{j=1, j\neq i} \mathbf{F}_{ij}^p + \mathbf{F}_i^w \,. \tag{105}$$

This force is additionally included as a source term in the right-hand side of Newton's equation of motion (Equation (99)) of the *i*-th particle.

C. Simulation Results

1. Fluid induced by an inline-oscillating 2D circular cylinder

In this section, we investigate the flow induced by an inline-oscillating cylinder in the fluid initially at rest as depicted in Figure 29. The inline-oscillation of the cylinder is governed by the following harmonic oscillation:

$$X_c = -A\sin(2\pi f t) \tag{106}$$

where X_c is the position of the cylinder center, and A and f are the amplitude and the frequency of the oscillation, respectively. This flow is characterized by Reynolds (Re) and Keulegan-Carpenter (KC) numbers, which are defined as:

$$\mathrm{Re} = \frac{U_{c,\max} D}{\nu} \tag{107}$$

and

$$\mathrm{KC} = \frac{U_{c,\max}}{f D} \tag{108}$$

respectively. Here, $U_{c,\max}$ is the maximum velocity of the cylinder during oscillation, D

is the cylinder diameter, and ν is kinematic viscosity. In this study, the computation is implemented at Re=100 and KC=5, at which the experimental (LDV) and numerical (body-fitted method) data by Dütsch et al. [118] are available. Hence, we can quantitatively compare the sharp and diffuse interface schemes with those.

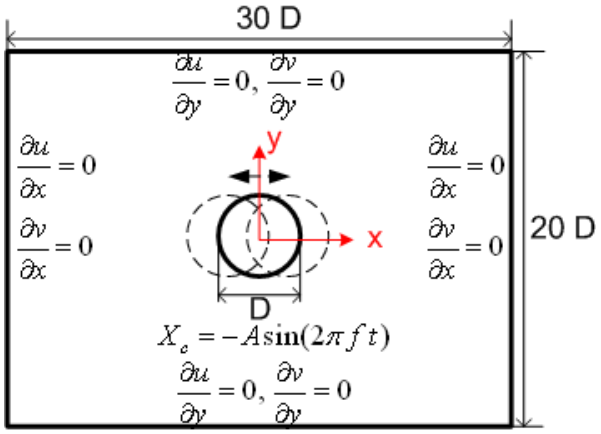

Figure 29. Geometry, computational domain, coordinates, and boundary conditions of the inline-oscillating cylinder problem.

The computational domain size is 30D×20D, and Neumann boundary conditions are imposed on four outer sides of the domain, as shown in Figure 29. For the cylinder surface boundary treatments, the IBMs with the sharp and implicit diffuse interface schemes are used. In the implicit diffuse interface scheme, forcing points are uniformly distributed on the cylinder boundary with the spacing of $\Delta s_b=\Delta x$. We first performed the sensitivity study on time step and grid sizes. The implicit diffuse interface schemes were used for the sensitivity study. As a target variable, we considered the streamwise force coefficient, which is defined as:

$$C_x = \frac{F_x}{\frac{1}{2}\rho U_{c,\max}^2 D} \tag{109}$$

where F_x is the streamwise surface force and is directly obtained from the immersed boundary method using Equation (91) with Equations (93a) or (94) depending on the interface scheme employed.

First, for the sensitivity of time step size, in the domain of 30D×20D with D=40Δx, we considered Δt=T/2000, T/3000, and T/4000. Figure 30 shows the resulting streamwise force coefficients. At Δt=T/2000, it shows a slight discrepancy, whereas at Δt=T/3000 and T/4000, similar results are displayed. Thus, we take Δt=T/3000 in the later calculations.

For the sensitivity of the grid size, we considered Δx= D/20, D/30, D/40, D/50, and D/60. Figure 31 presents the resulting streamwise force coefficients. It is observed that almost converged results are reached from D=40Δx.

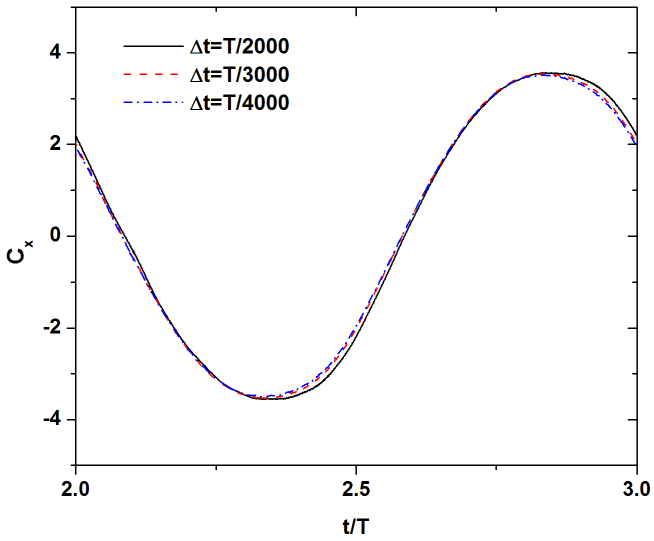

Figure 30. The effect of time step size on the streamwise force coefficient.

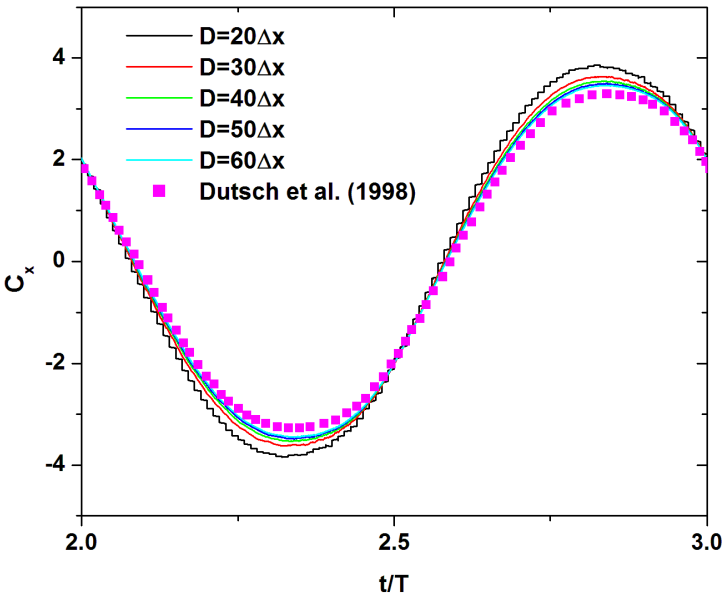

Figure 31. The effect of grid size on the streamwise force coefficient.

Figure 32 presents the vorticity fields at four phase angles of 0^o, 96^o, 192^o, and 288^o. Both the sharp and implicit diffuse interface schemes show qualitatively similar vorticity fields observed by body-fitted grid methods in Dütsch et al. [118]. However, the pressure field (not given here) from the sharp interface scheme has some wiggles, which may be due to the spurious oscillation. This oscillation is later shown during the variation of streamwise force coefficients.

Next, to quantitatively assess the two schemes, we compare the velocity data at locations x = -0.6, 0, 0.6, and 1.2D at the phase angle 330^o. Figures 33 and 34 present the horizontal and the vertical velocities at the phase angle 330^o obtained from the sharp interface scheme and the implicit diffuse interface scheme, respectively. Both figures

include experimental and numerical data of Dütsch et al. [118]. We can observe that both schemes display a good agreement with Dütsch et al.'s results.

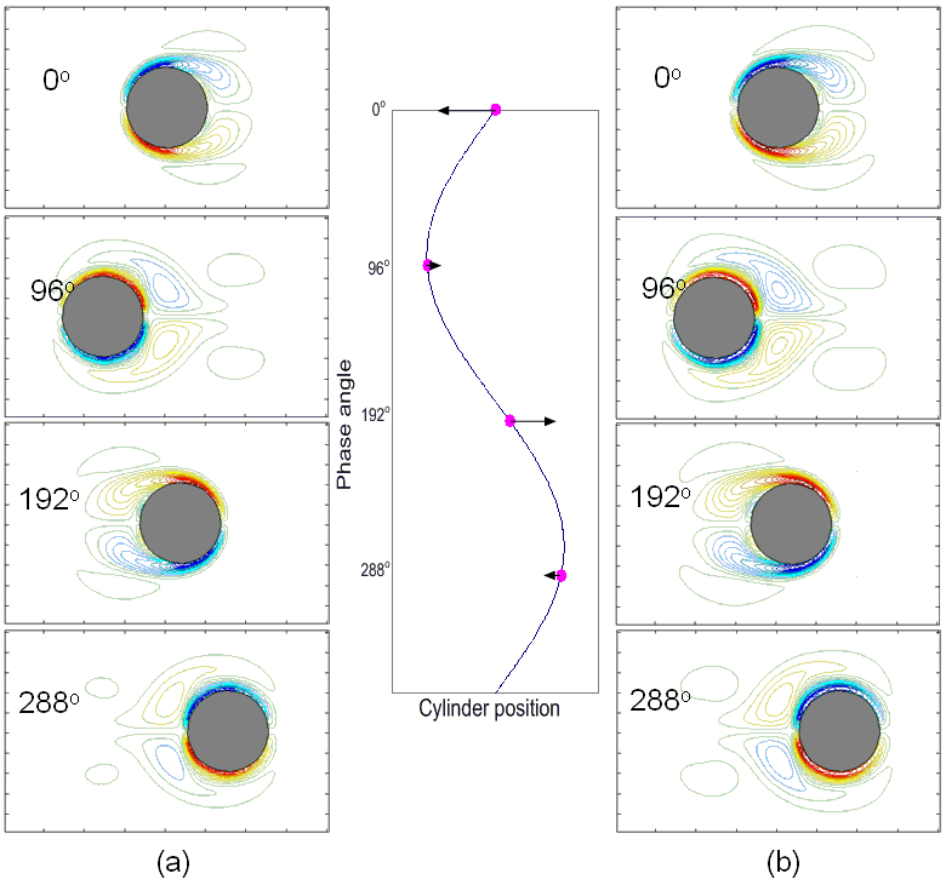

Figure 32. Vorticity fields obtained from (a) the sharp interface scheme and (b) the implicit diffuse interface scheme.

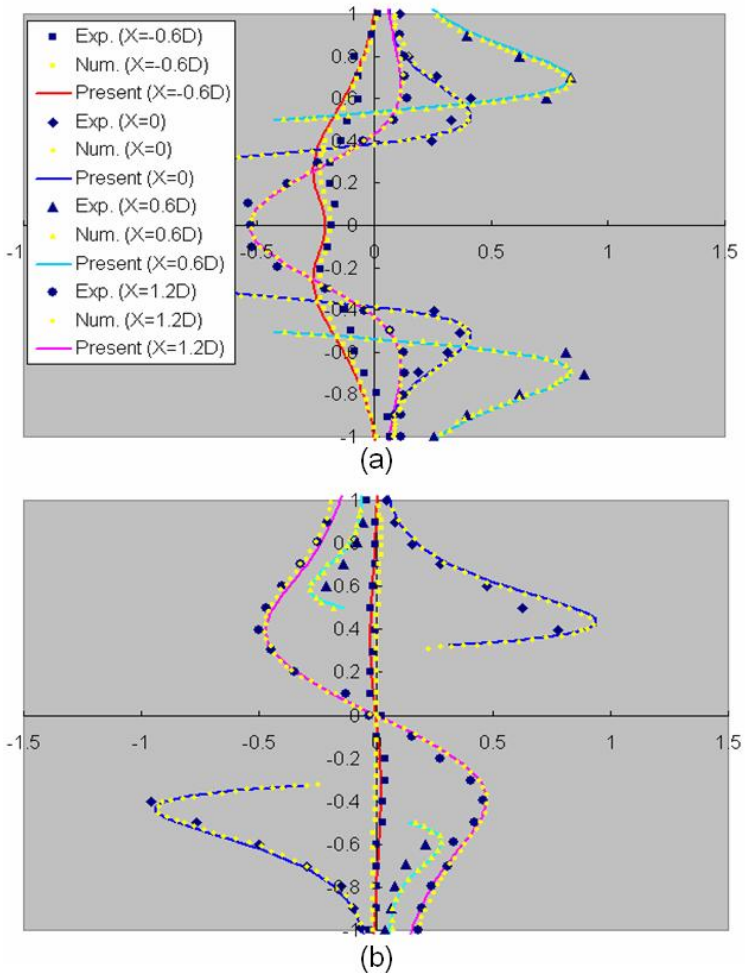

Figure 33. Comparison of (a) horizontal and (b) vertical velocities at the phase angle of 330^{o} at x = -0.6, 0, 0.6, and 1.2D between the sharp interface scheme and Dütsch et al. (1998).

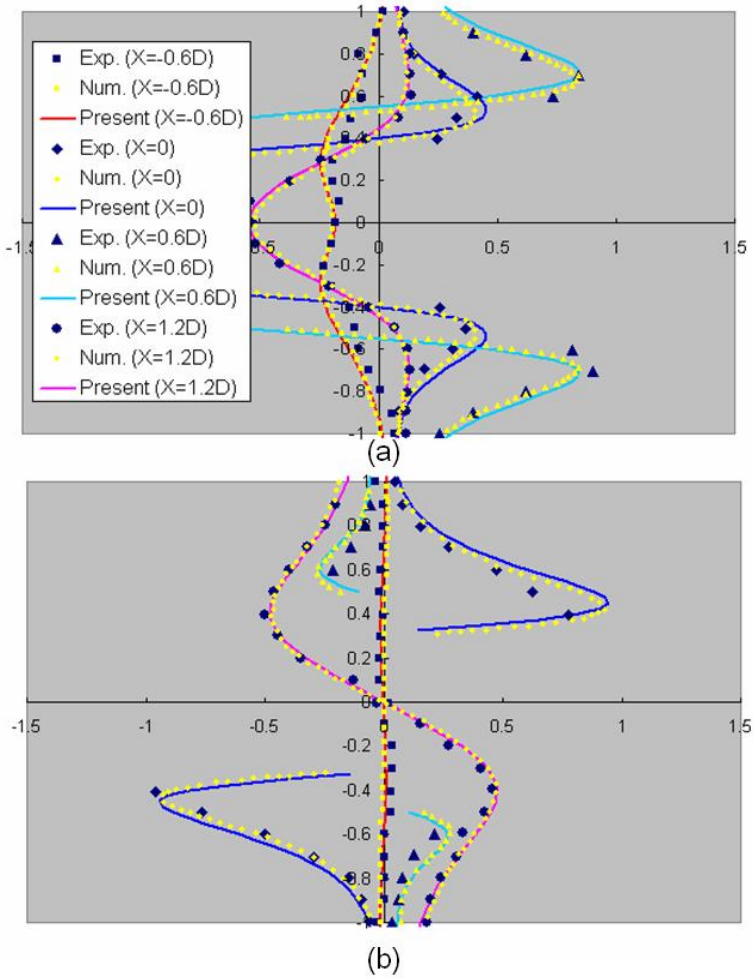

Figure 34. Comparison of (a) horizontal and (b) vertical velocities at the phase angle of 330^{o} at x = -0.6, 0, 0.6, and 1.2D between the implicit diffuse interface scheme and Dütsch et al. (1998).

Figure 35 presents the streamwise force coefficient variation obtained from the exterior sharp interface scheme with $\Delta x=D/40$. The gray line shows spurious oscillations due to discontinuous change of nodes used in the interpolation. However, if we adopt the filtering (for example, low-pass FFT filtering) or smoothing, as Miller and Peskin [121] and Shen et al. [122] recommended, it shows a good agreement with Dütsch et al.'s data.

It should be pointed out that we also tested the sharp interface schemes, which had been developed for reducing such spurious oscillations by Yang and Balaras [28] and Liao et al. [119], but only minor improvements were obtained under the IB-LBM based on split-forcing LBE. This may be due to the difference between the Navier-Stokes equations and the lattice Boltzmann equations.

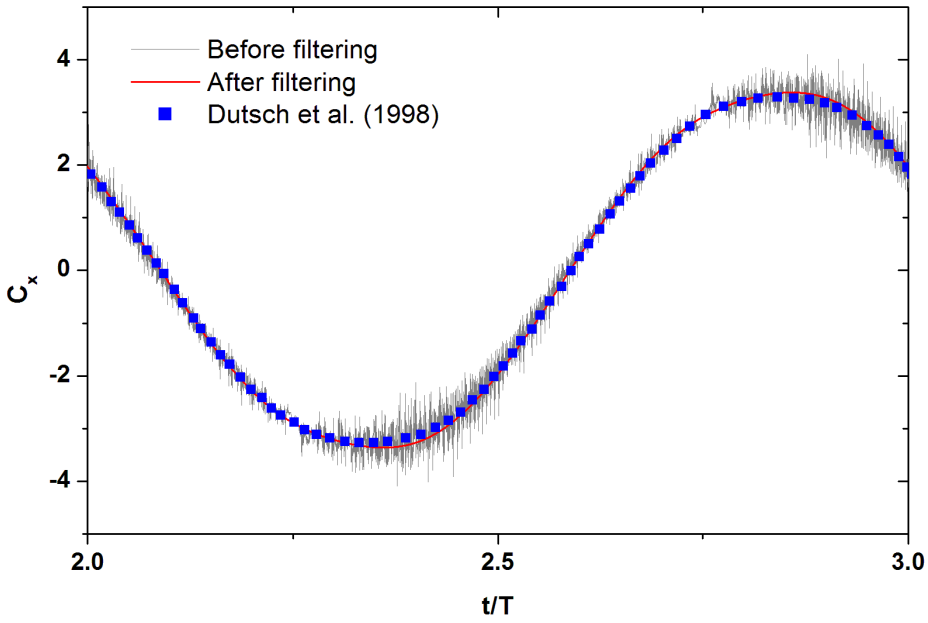

Figure 35. Streamwise force coefficients obtained from the sharp interface scheme before and after low-pass FFT filtering.

When the grid sensitivity study with Δx=D/20, D/30, D/40, D/50, and D/60 was performed for the exterior scheme with filtering, despite not being given here, the results almost converged from Δx=D/30. In contrast, as presented in Figure 31, when adopting the diffuse interface schemes, the streamwise force coefficients show a bigger

discrepancy with the measure data. Particularly, the largest discrepancies occurred near phase angles of 90^o and 270^o (where the magnitude of acceleration is the maximum) and the least discrepancies occurred at angle phases of 0 and 180^o (where the magnitude of acceleration is zero). In other words, the larger the magnitude of acceleration the cylinder has, the larger the discrepancy in the surface force coefficient is. It should also be noted that to the best of our knowledge, no direct-forcing IBM with diffuse interface schemes except [91] was documented for this problem. In [91], the multi-direct-forcing IBM with the diffuse interface scheme was used but the governing equations were based on vorticity equations, not pressure equations. In their calculation results of the force coefficient for the comparison, they also provided the results from the direct-forcing IBM with the diffuse interface schemes based on the pressure equation, which showed large discrepancies near phase angles of 90^o and 270^o as in this study. On the other hand, from the results of flow past a stationary circular cylinder in Chapter III, we can observe that the diffuse interface schemes can have maximum 5% larger drag coefficients than body-fitted methods can at $\Delta x=D/40$. However, the largest discrepancy for the oscillating cylinder in this study is about 8% under $\Delta x=D/40$.

To mitigate the over-prediction, we can adopt the concept of the effective radius as in [123], which corrects over-estimation of the drag coefficient due to the effect of diffuse boundary. From the stationary results in Chapter III, we can approximate the effective radius (r_{eff}) as

$$r_{eff} \approx r_s + 0.5\Delta x. \tag{110}$$

Thus, we retract the surface on which the forcing points are distributed by an amount of

0.5Δx from the geometric surface. The resulting surface coefficient variation at Δx=D/40 is given in Figure 36, which shows a good agreement with Dütsch et al.'s data. Using the reduced radius can be interpreted as locating the forcing node at the center of the cell with a surface boundary instead of putting the forcing node on the surface boundary. It should also be noted that the radius for the evaluation of V_s in the added mass term is the geometric radius (r_s), not the corrected radius.

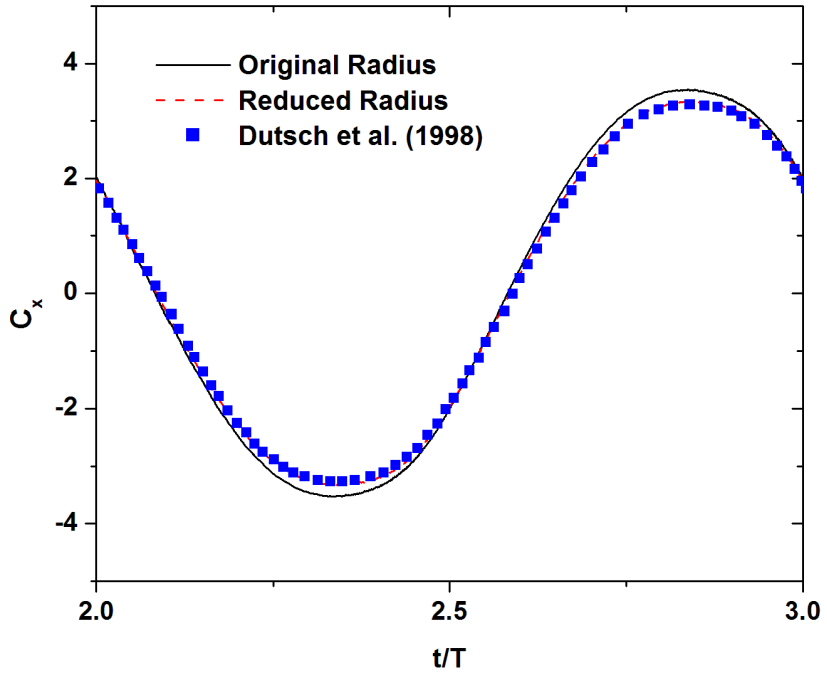

Figure 36. The streamwise force coefficients with and without considering the diffuse boundary effect.

In summary, velocity results (vorticity, horizontal and vertical velocities) from both schemes showed overall a good agreement with experimental and body-fitted calculation data of Dütsch et al. [118]. However, the sharp interface scheme showed

spurious oscillations in the streamwise surface force coefficient, although after filtering or smoothing, it showed a good agreement with the experiment. In contrast, the diffuse interface scheme produced smooth variations in the surface force coefficient due to the use of discrete delta function, which involves the information of interior nodes inside the solid boundary, thus smoothing the discontinuous change of nodes. However, such a property of the discrete delta function reduced the accuracy. Specifically, the reduction of accuracy was larger in the oscillating cylinder in this study than in the fixed cylinder. This discrepancy could be mitigated by considering the effective radius.

2. Sedimentation of single 2D circular particle

As a starting point of the application of the present IB-LBM to particle-fluid two-phase flows, we perform the numerical simulation of the motion of a 2D circular cylinder particle falling in the channel. The channel is [0,2]×[0,6] in cm and the circular particle with diameter D_p=0.25 cm is initially located at (1,4). The fluid and the particle are initially at rest. The density (ρ_f) and dynamic viscosity (μ) of the fluid are 1 g/cm^3 and 0.1g/cm-s, respectively, and the density of the particle (ρ_p) is 1.25 g/cm^3. Calculations are implemented for two different lattice sizes of 0.01 and 0.005 cm.

For the validation of the present IB-LBM, we adopt the sharp interface scheme and the explicit diffuse interface scheme with 4-point discrete delta function.

Figure 37 shows velocities, pressures, and vorticities of the fluid affected by a moving particle at a time instant. Here, two vortices are clearly captured in the wake region of the falling particle with low pressure.

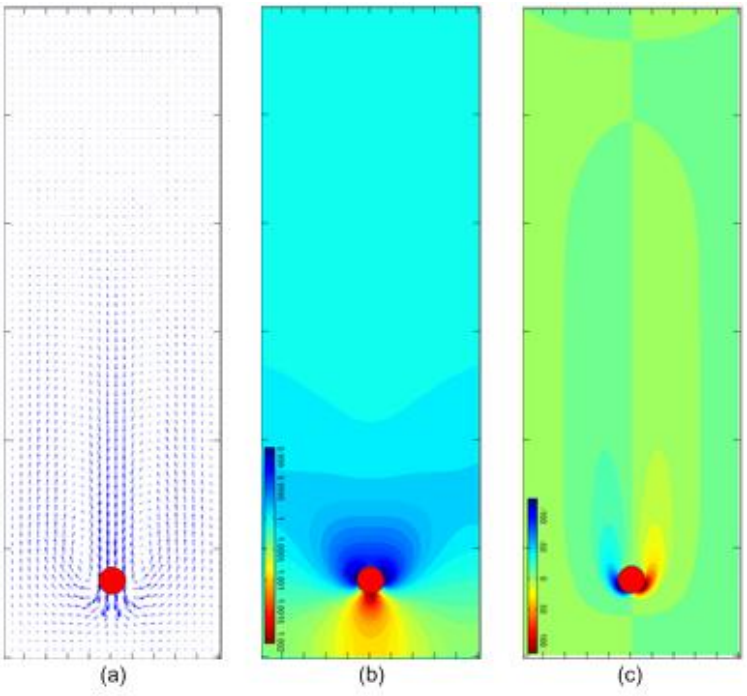

Figure 37. (a) Velocity vector field and (b) pressure and (c) vorticity contours at 0.7 s obtained from the standard diffuse scheme.

To quantitatively validate our simulations, the maximum particle Reynolds number is considered. The maximum particle Reynolds number is defined as

$$\mathrm{Re}_{\max} = \max_{t}\left[\frac{\rho_p D_p \sqrt{u_p^2(t)+v_p^2(t)}}{\mu}\right] \tag{111}$$

where u_p and v_p are x- and y-velocities of the falling particle. Table 7 presents the maximum Reynolds numbers obtained from the present calculations and the Lagrange multiplier fictitious domain method by Glowinski et al. [124]. The results from the sharp interface scheme and the diffuse interface scheme with the corrected radius show an excellent agreement with those from the Lagrange multiplier fictitious domain method. However, the results from the standard diffuse interface scheme with the actual radius

and added mass force show under-prediction. Without considering the added mass force, the results are more under-predicted.

Table 7. Comparison of the maximum Reynolds numbers in single particle sedimentation.

Sharp interface scheme	Diffuse interface scheme			Glowinski et al. [124]
	Standard	Without added mass force	With corrected radius	
17.44 (1/100 cm) 17.46 (1/200 cm)	17.06 (1/100 cm) 17.28 (1/200 cm)	17.02 (1/100 cm) 17.20 (1/200 cm)	17.44 (1/100 cm) 17.47 (1/200 cm)	17.44 (1/192 cm) 17.51 (1/256 cm)

To investigate this in greater detail, we compare time histories of particle vertical position and velocity in Figures 38 and 39, respectively. The particle starts to move downward because the gravitational force is greater than the buoyancy force. As the particle falls down, its velocity increases, and accordingly, the drag force also does. The drag force becomes close to the difference between gravitational and buoyant forces, and accordingly, its velocity becomes constant. In Figure 38, the vertical particle velocities from the present calculation are compared with those from the Lagrange multiplier fictitious domain method [124]. We can find that consideration of the added mass force is important because without it, the vertical velocity and position show quite large discrepancies with other cases. The effect of the corrected radius is relatively small compared with that of the added mass effect. However, as will be shown in Section C.7, the effect of the corrected radius is more dominant in the sedimentation of a 3D spherical particle because more forcing nodes are involved in the evaluation of the surface force.

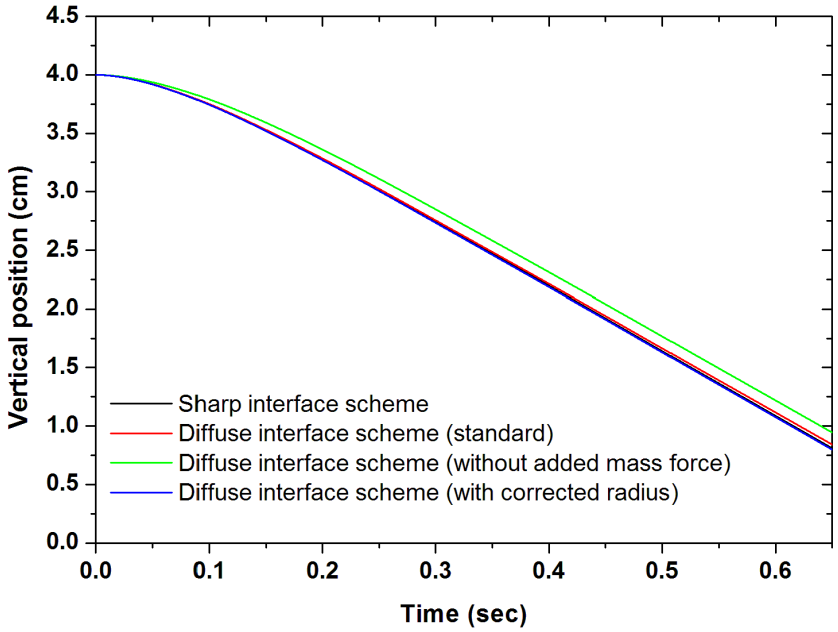

Figure 38. Vertical position of a circular particle falling in the channel with time.

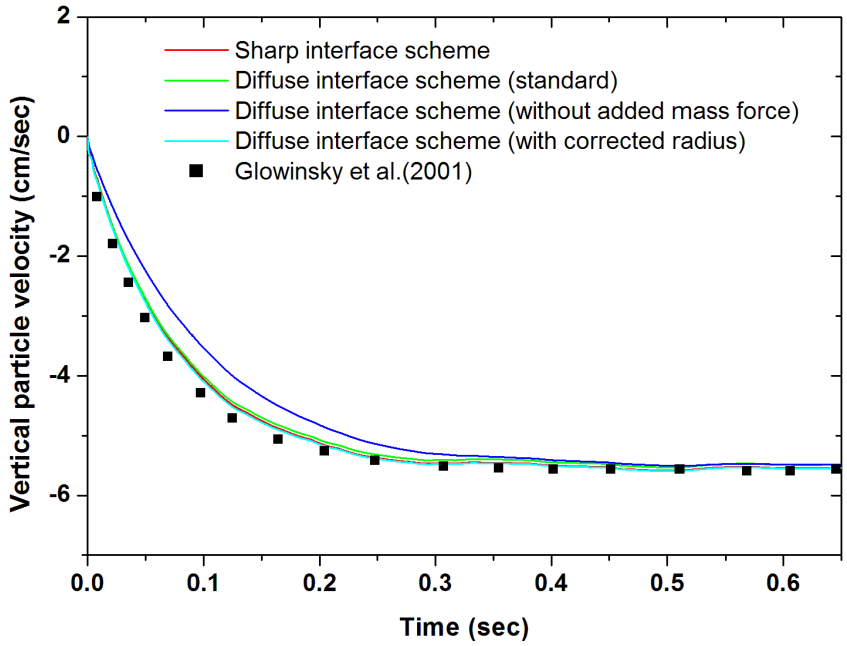

Figure 39. Vertical particle velocity variations.

The sharp interface scheme and the diffuse interface scheme with the corrected radius show almost the same vertical variation of position and velocity. However, as presented in Figure 40, the diffuse and sharp interface schemes have different surface force variation although the average behavior is similar. This may be because, as discussed in Section C.1, the sharp interface scheme has the spurious oscillation due to the discontinuous change of neighboring nodes for the force interpolation, while the diffuse interface scheme has the smooth variation. Hence, we can expect that as particle Reynolds numbers become higher, this oscillation can increase, thus resulting in an unreasonable solution.

To support this prediction, we simulated another case with a higher particle Reynolds number by changing the density ratio from 1.25 to 1.5 and the viscosity from 0.1g/cm-s to 0.01 g/cm-s, with other conditions being the same. Figure 41 presents the

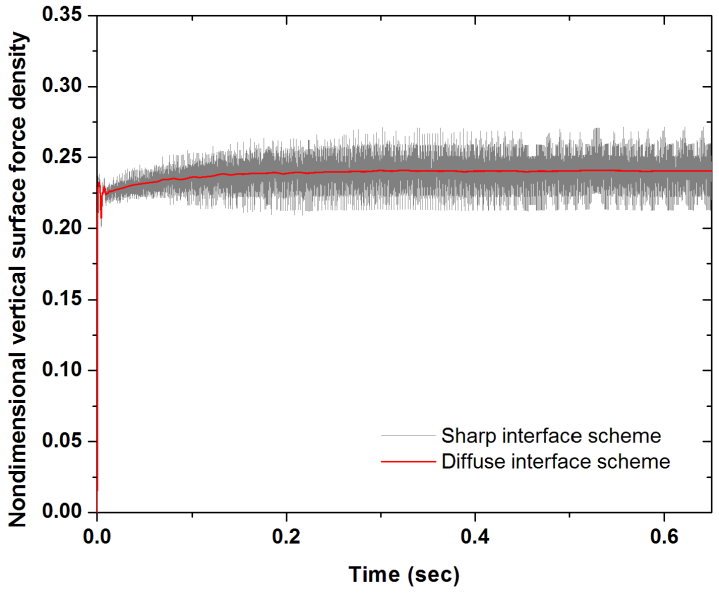

Figure 40. Variation of vertical component of non-dimensional surface force density.

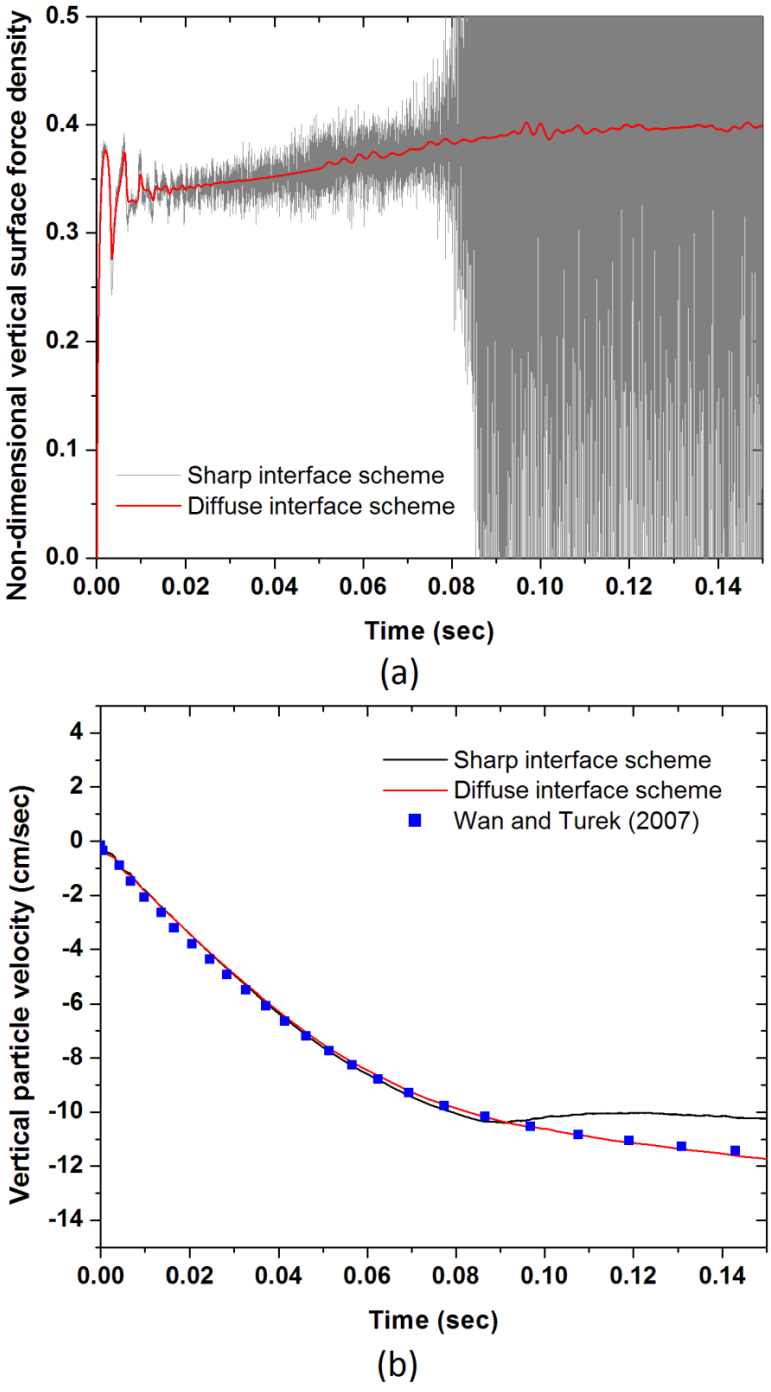

Figure 41. Variation of (a) vertical component of non-dimensional surface force and (b) vertical particle velocities under the higher Reynolds condition.

variations of the non-dimensional vertical surface force density and vertical particle velocity with $\Delta x=1/300$cm. The surface force oscillation in the sharp interface scheme is steeply amplified after 0.85 sec, thus causing vertical velocity variation different from those of the diffuse interface scheme and the fictitious boundary and moving mesh method of Wan and Turek [125]. Thus, the use of the sharp interface scheme in a moving particle simulation should be limited in relatively low Reynolds number.

3. Sedimentation of double 2D circular particles

To test the capability to simulate solid-solid interactions as well as solid-fluid interactions, two circular cylinder particles falling in the channel are considered. We consider two cases with relatively low and high particle Reynolds numbers. Physical conditions and numerical parameters adopted in the calculation of two cases are summarized in Tables 8 and 9, respectively.

Table 8. Physical conditions of sedimentation of two circular cylinder particles.

Case	Channel size (cm)	Particle			Fluid	
		Diameter (cm)	Density (g/cm^3)	Initial positions of particles (cm)	Density (g/cm^3)	Viscosity (g/cm-s)
1	8 × 2	0.2	1.01	(0.999, 7.2), (1, 6.8)	1.0	0.01
2	6 × 2	0.25	1.50	IP1: (5, 1.001), (4.5, 0.999) IP2: (5, 1), (4.5, 1)	1.0	0.01

Table 9. Numerical conditions of sedimentation of two circular cylinder particles.

Case	Conversion factors		Lattice Boltzmann parameters		Solid collision parameters	
	Lattice size, Δx (m)	Time step, Δt (s)	Domain size	Relaxation time, τ	Force range, ζ	Stiffness parameter, ε_p
1	1×10^{-4}	5×10^{-4}	800×200	0.65	2	0.1
2	5×10^{-5}	2.5×10^{-5}	1200×400	0.53	2	0.1

We adopt the explicit diffuse interface scheme with the 4-point discrete delta function and the sharp interface scheme for this simulation.

As shown in Figure 42, this simulation captures the well-known drafting-kissing-tumbling (DKT) phenomenon. Due to the wake of low pressure, which the leading particle creates, the trailing particle falls faster than the leading one (drafting) (t=1s). The increased speed of the trailing particle makes the gap between two disks narrower, resulting in contact with each other (kissing) (t=1.5s). After kissing, the two particles fall together as an elongated body. This state is unstable, and as a result, the elongated body rotates and becomes perpendicular to the flow direction so as to be more stable (tumbling) (t=2.5s). The two particles are finally separated because they are not a body (t=3.5s).

Calculated vertical velocities of two particles are compared with those from the feedback-forcing IB-LBM by Feng and Michaelides [6] and the implicit diffuse direct-forcing IBM (based on the NSE) by Wang et al. [19], as shown in Figure 43, and show an overall good agreement. Different vertical velocity variations after kissing between the present diffuse interface scheme and other numerical schemes may be due to the difference of particle collision models, and the early start of tumbling in the sharp interface scheme can be attributed to the instability from the oscillating surface force.

However, it should be noted that in these calculations, the added mass force was not considered. If the added mass force is included in the equation of particle motion, the vertical velocity variation changes as shown in Figure 44. Here, the diffuse interface scheme was used.

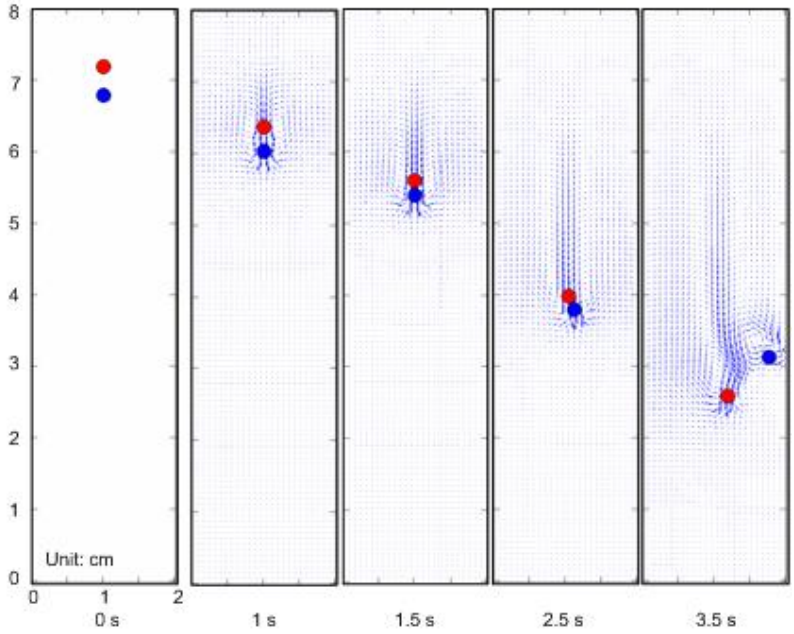

Figure 42. Velocity vector fields and particle positions.

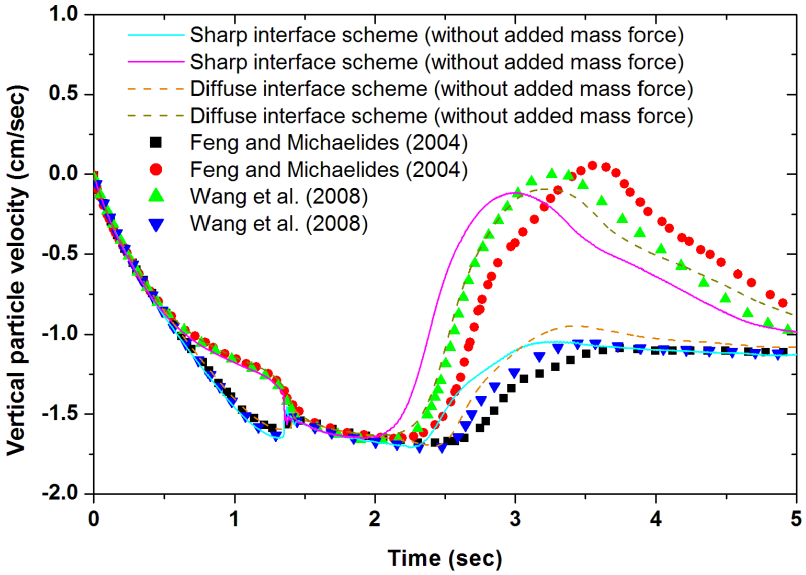

Figure 43. Comparison of vertical particle velocities in Case 1.

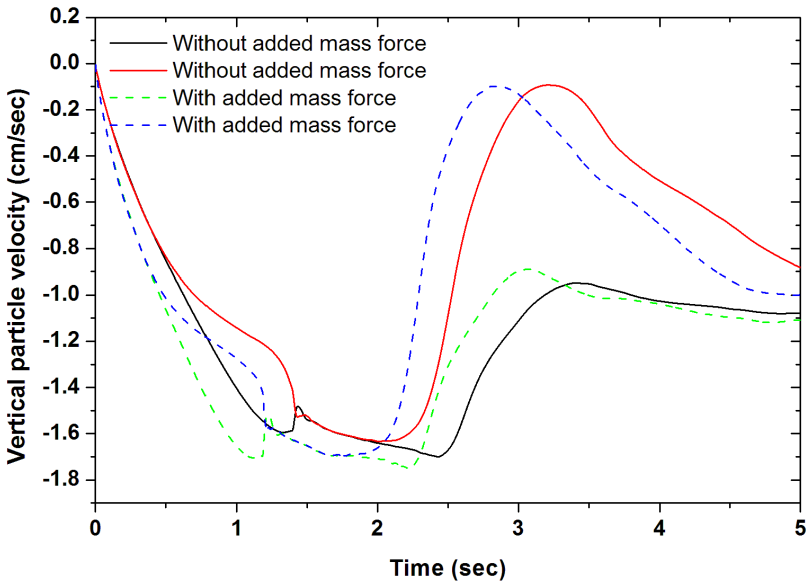

Figure 44. Variation of vertical velocities with and without considering the added mass force.

In Case 2, two different initial particle positions – vertically off-centered initial particle positions (IP1) and vertically aligned initial particle positions (IP2) – are considered as shown in Table 8.

Figure 45 shows the vertical position and velocity variations in Case 2 with vertically off-centered initial positions (IP1) calculated from the diffuse interface scheme. In Case 2 with vertically aligned initial particle positions (IP2), only drafting and kissing are observed. Before tumbling, particles collide with the bottom of the wall. This is because at the kissing state, the elongated body composed of two particles with the aligned initial positions (IP2) is more stable than that composed of two particles with

slightly off-centered initial positions (IP1), thus keeping the kissing state longer. This also means that the diffuse scheme maintains the symmetry well.

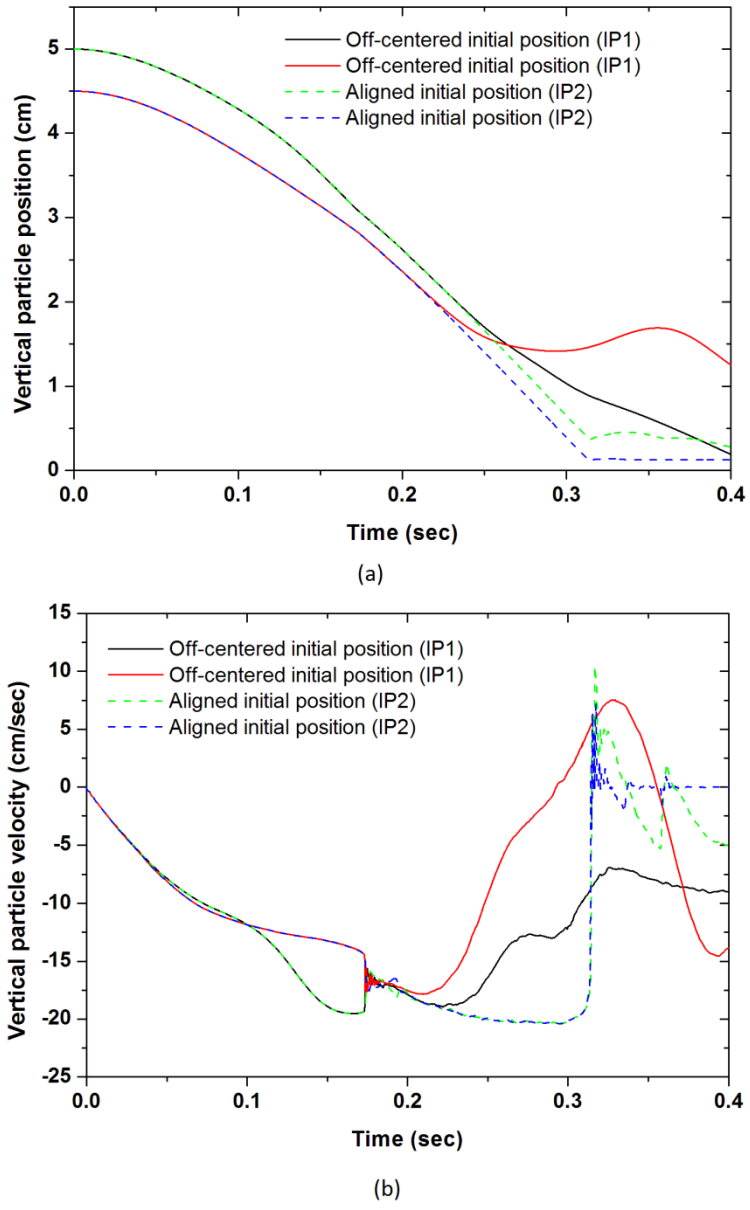

Figure 45. Variations of (a) vertical positions and (b) vertical velocities of particles when using the diffuse interface scheme under different initial particle positions.

Figure 46 presents vertical velocities of two particles under the present diffuse interface scheme and the explicit diffuse direct-forcing scheme based on the Navier-Stokes equation by Uhlmann [23]. Compared with Case 1, the kissing state is short (0.175~0.2 sec). The results from the current diffuse interface scheme show a good agreement with those from Uhlmann's direct-forcing IBM. Different vertical velocity variations after kissing may also be due to the difference of particle collision models.

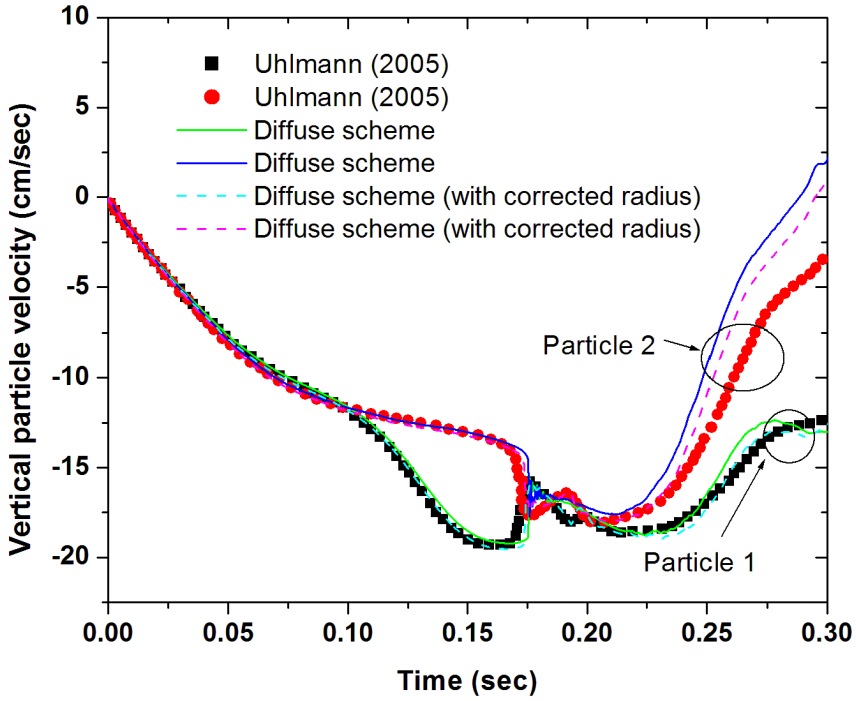

Figure 46. Comparison of vertical velocities of particles in Case 2.

On the other hand, the results from the sharp interface scheme show different behaviors from those above, as demonstrated in Figure 47. The reason could be found from the fact that the physical conditions in Case 2 are the same as those in Figure 41 in

Section C.2. In other words, the spurious force oscillation due to the discontinuous node change misleads the velocities. Besides, it is observed that around 0.3 sec, the solution diverges.

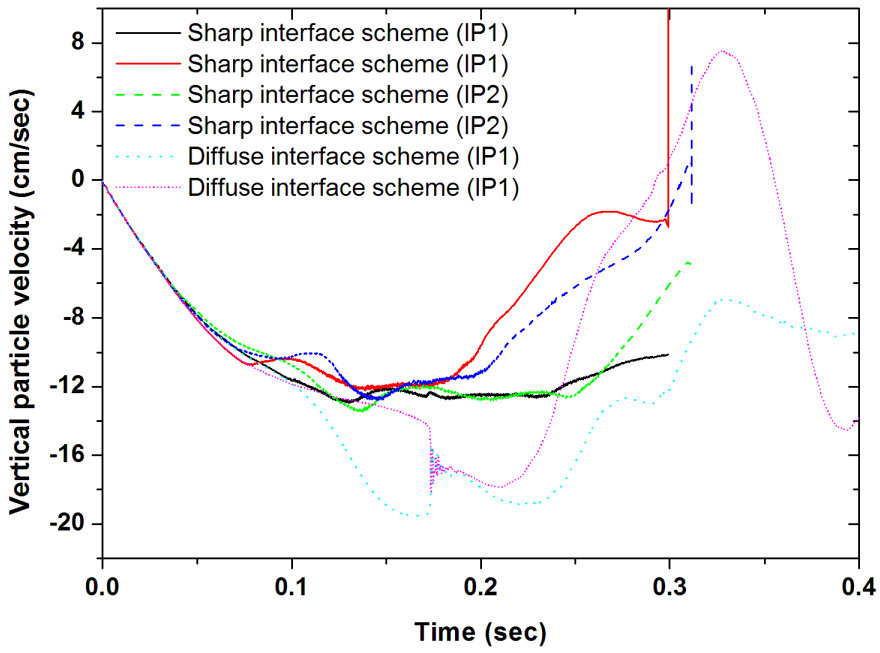

Figure 47. Comparison of vertical particle velocities between different initial positions.

In addition, as presented in Figure 48, without considering the added mass force the results show very different behaviors from those shown with considering the added mass force, similar to Uhlmann's results. This reaffirms the importance of considering the added mass effect. Here, the diffuse interface scheme with off-centered initial positions (IP1) was used.

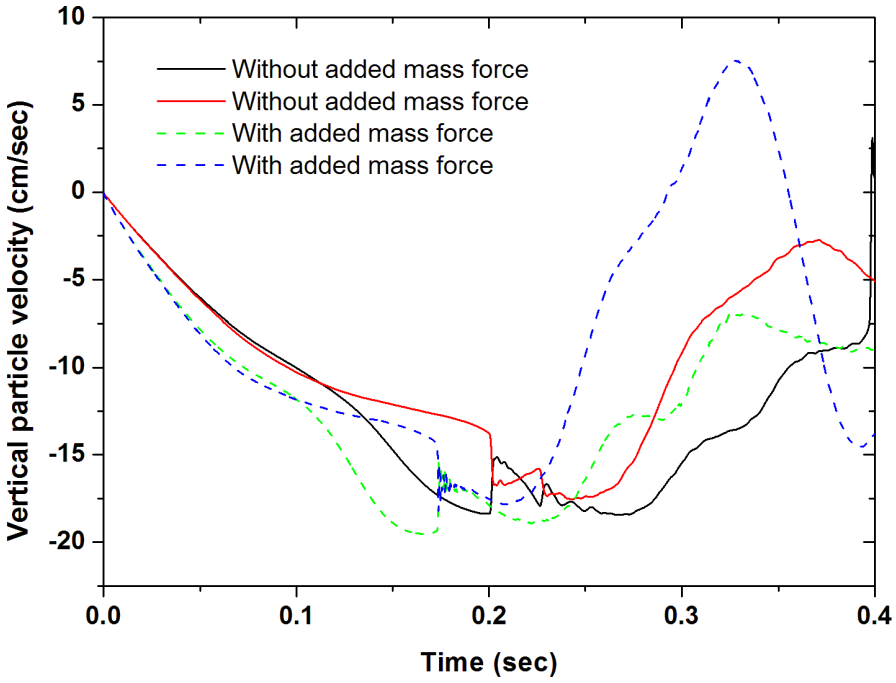

Figure 48. Variation of vertical particle velocities with and without considering the added mass force in Case 2.

4. Sedimentation of six 2D particles in channel

In this section, we consider various cases of sedimentation of six particles in the channel. Six particles of the diameter D=0.2 cm with the density of 1.01g/cm^3 are initially located at three different initial positions, given in Table 10, in the channel of 2 cm×8 cm. The density and viscosity of the fluid are 1.0 g/cm^3 and 0.001 cm^2/s. Figure 49 presents the geometry and parameters for the six-particle sedimentation in Case 1.The difference from Case 1 in Section C.3 is the channel is open. We adopt the explicit diffuse interface scheme with 4-point discrete delta functions in this simulation.

Table 10. Initial positions of six particles.

Case	Array	Initial positions (cm)
1	3 by 2	(-0.2, 7.2), (+0.2,7.2); (-0.2, 6.8), (+0.2, 6.8); (-0.2, 6.4), (+0.2, 6.4)
2-1	2 by 3	(-0.4, 7.2) (0, 7.2), (+0.4,7.2); (-0.4, 6.8), (0, 6.8), (+0.4, 6.8)
2-2	2 by 3	(-0.399, 7.2) (-0.001, 7.2), (+0.399,7.2); (-0.4, 6.8), (0, 6.8), (+0.4, 6.8)

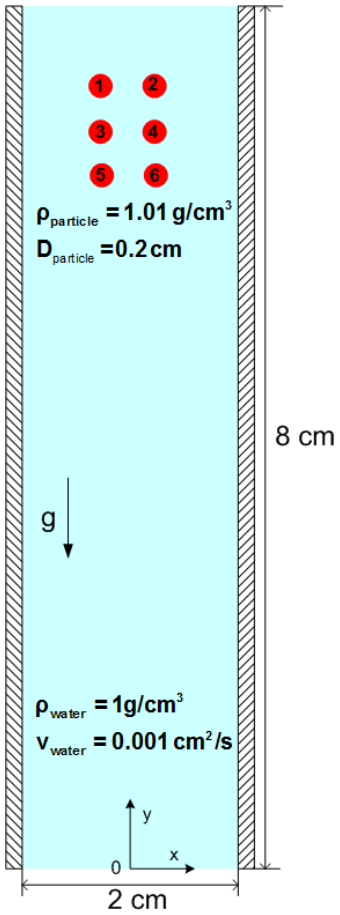

Figure 49. Geometry and physical conditions in Case 1 of six-particle sedimentation.

Figure 50 shows the velocity vector fields at different time instants. Particles 3 and 4 are first drafted by the wake of low pressure behind particles 5 and 6 (at t=1 sec) and kiss (at t=1.5 sec). Then, tumbling (clockwise for particles 3 and 5 and counterclockwise for particles 4 and 6) occurs, and thus particles 5 and 6 are located above Particles 3 and 4 (at t=2.0 sec). While particles 5 and 6 are away from Particles 3 and 4 due to rotational force, particles 1 and 2 kiss particles 3 and 4, respectively. Then, particles 1 and 3 rotate in clockwise and particles 2 and 4 rotate in counterclockwise. During the tumbling, particles 1 and 2 experience the collision, so that the repulsive force is exerted (at t=3 sec). As a result of the tumbling, vertical position changes of the particles occur again (t= 4sec). While particles 3 and 4 are under upward rotational force, particles 5 and 6 approach particles 1 and 2 (at t=5 sec).

Figure 51 presents the velocity vector fields in Cases 2-1 and 2-2. Both cases show similar behaviors before kissing. However, the kissing state remains longer in Case 2-1 than in Case 2-2 because the aligned initial positions in Case 2-1 keep the elongated body consisting of two kissing particles more stable. In addition, it is observed that the elongated bodies near the wall start tumbling earlier than that in the middle in each case due to the wall effect. As a result, after 4 sec, both cases have entirely different particle behaviors.

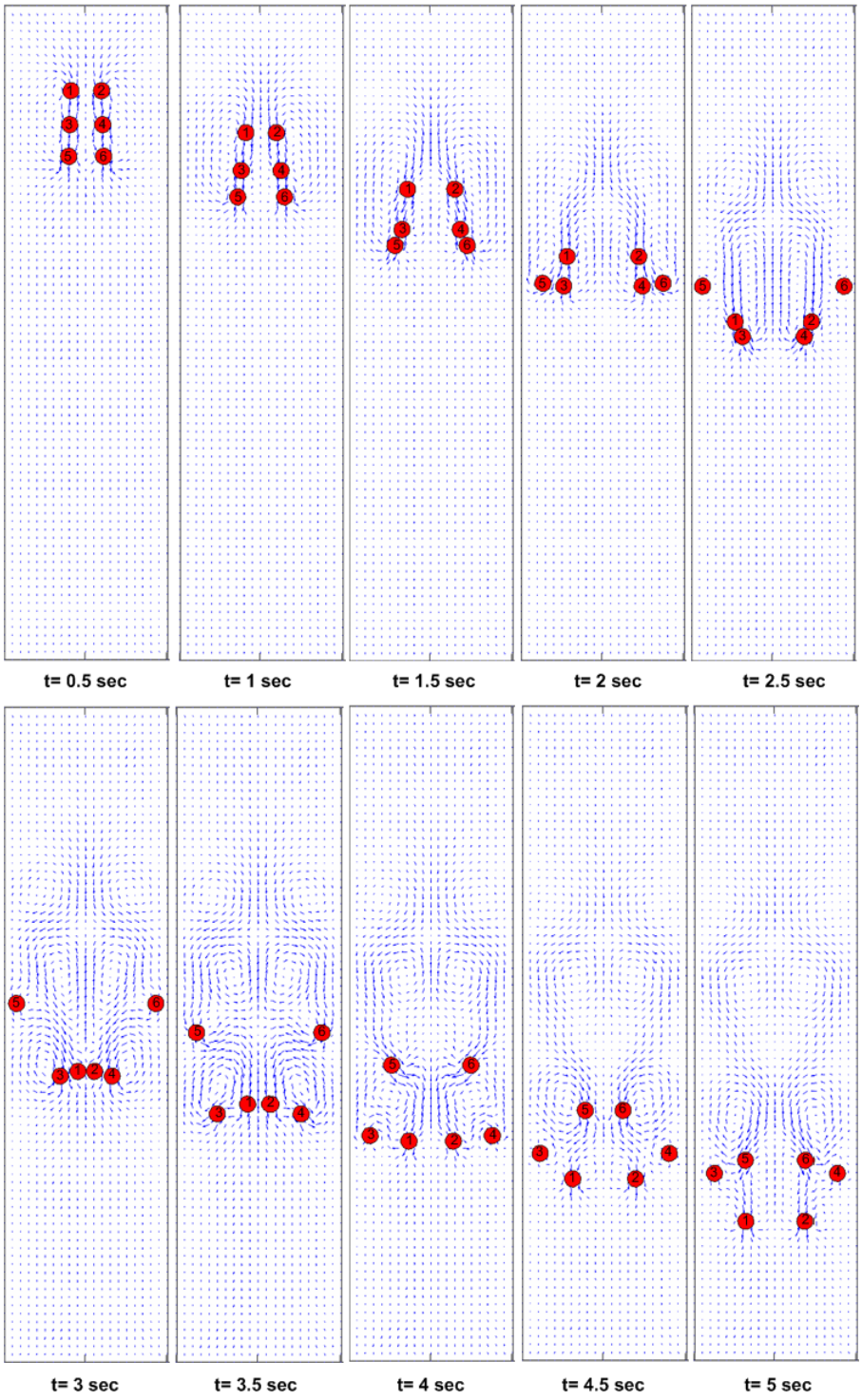

Figure 50. Velocity vector plots of sedimentation of six particles at different time instants (Case 1).

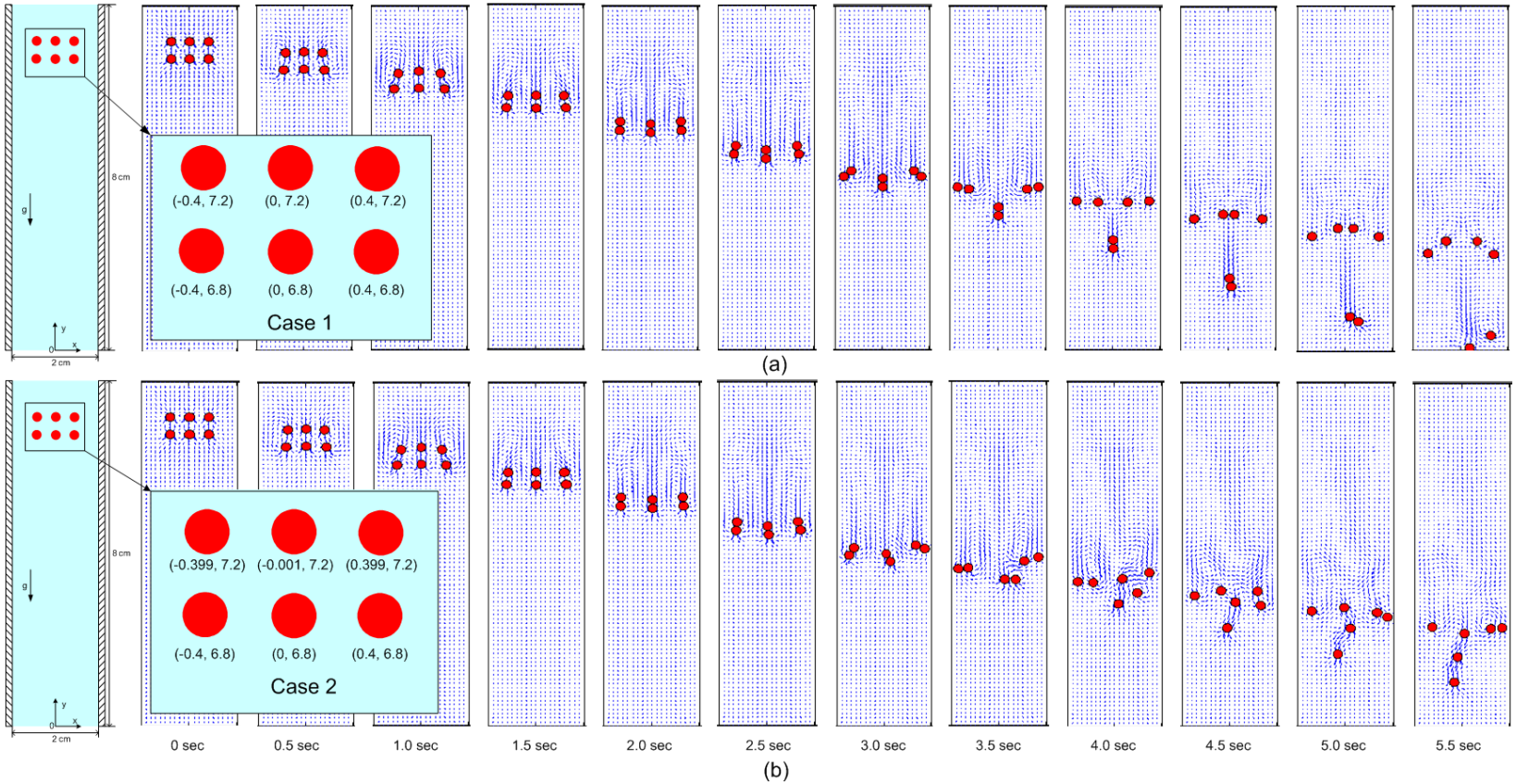

Figure 51. Transient behaviors of six particles in two different initial positions (Cases 2-1 and 2-2).

5. Sedimentation of a large number of 2D particles

Now, the sedimentation of 504 circular particles is simulated to test the capability of the present method to deal with a large number of particles. Here, we use the explicit diffuse interfaces scheme with 4-point discrete delta functions. The 504 circular particles of the diameter D_p=0.0625 cm are in a closed square cavity of [0,2]×[0,2] in cm. Density (ρ_f) and dynamic viscosity (μ) of the fluid are 1 g/cm^3 and 0.01g/cm-s, respectively, and the density of each disk (ρ_d) is 1.01 g/cm^3. Initially, 18 lines of particles with each line having 28 particles are located as shown in Figure 51(a). The lattice size is 0.00390625 cm.

Snapshots for the evolution of the fluid vector field with the 504 circular particles are presented in Figure 52. These results show the Rayleigh-Taylor instability. Two eddies near the sidewalls are created, as depicted in Figures 52(b) and 52(c), and each eddy penetrates into the inner part along the sidewall and bifurcates into upward and downward eddies, as shown in Figures 52(d) and 52(e). Then, two downward eddies from each sidewall grow and become dominant, beginning to pull particles, as seen in Figures 52(f) and 52(g). Finally, the particles start to settle at the bottom of the cavity, and the large eddies disappear, as depicted in Figures 52(h) and 52(i). These behaviors are comparable to those reported by Feng and Michaelides [6].

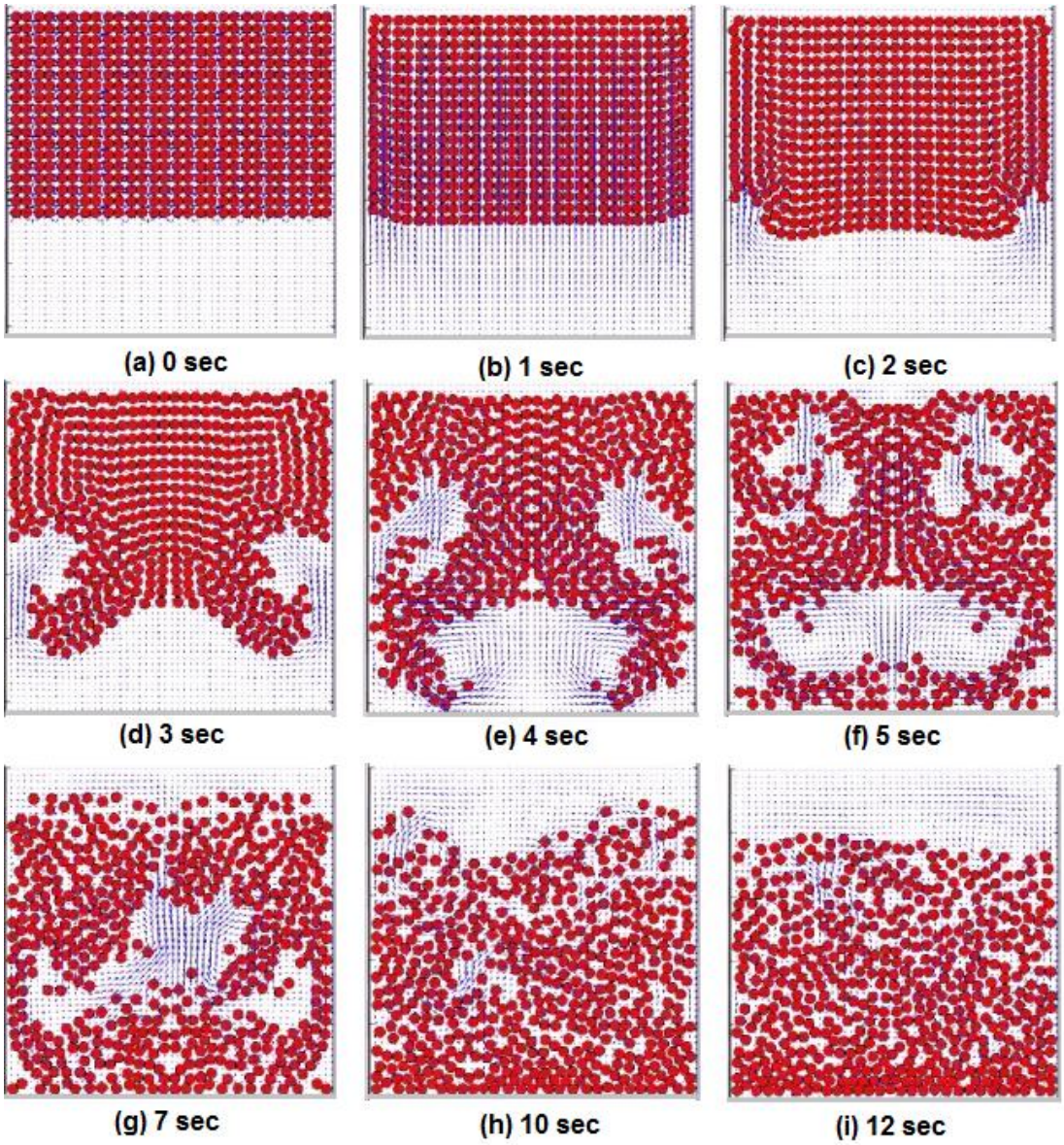

Figure 52. Sedimentation of 504 particles in the closed square cavity.

6. Particle behaviors in the 2D channel with leakage

In this section, we simulate particle behaviors in a 2D pipe with holes. This case is expected to be applied to ball sealer performance optimization in petroleum engineering and any pipes with leakage smaller than the particle size in general engineering fields.

In petroleum engineering, ball sealers are injected into well treating fluids for the purpose of contacting and sealing those perforations that are accepting the fluid flow [126], thereby diverting reservoir treatments to other portions of the target zone. Important ball sealer behaviors are seating and unseating. Seating efficiency is primarily influenced by the velocity of balls down the pipe and the fluid velocity through the perforations. To divert the sealer to the perforation, the inertial force of the ball must be overcome by the drag force created by the fluid velocity through the perforation [126].

To simulate the seating behavior of ball sealers, we adopt the explicit diffuse scheme with 4-point discrete delta functions and particle-particle and particle-wall collision models given in Section B.3. For stability, we assume that once the particle is seated on the hole, it does not move until the net force of the particle in the opposite direction to wall is greater than zero.

As a preliminary case, we consider a single particle in the fluid channel with a hole. Physical conditions and numerical parameters in the present calculation are summarized in Table 11. In this calculation, we investigate the effects of (i) initial positions of particles, (ii) particle-fluid density ratios, and (iii) hole outlet pressures on the particle behavior. Figure 53(a) shows different trajectories of particles with different initial horizontal positions (x=-1.1D, 0, and +1.1D). In Figure 53(b), the particle with

particle-fluid density ratio of 1.00005 has a flatter trajectory than that with particle-density ratio of 0.99995 because of stronger downward inertial force. In Figure 53(c), it is observed that as the pressure difference becomes large, the trajectory approaches the wall side with a hole.

Table 11. Physical and numerical conditions for the preliminary calculation.

Particle size, D	Channel size	Hole size	Hole position	Initial vertical positions, y	LBM parameters	
					Relaxation time, τ*	Gravitational constant, g*
20Δx	5D×25D	0.4 D	(2.5D, 7.5D)	19D	0.6	0.05

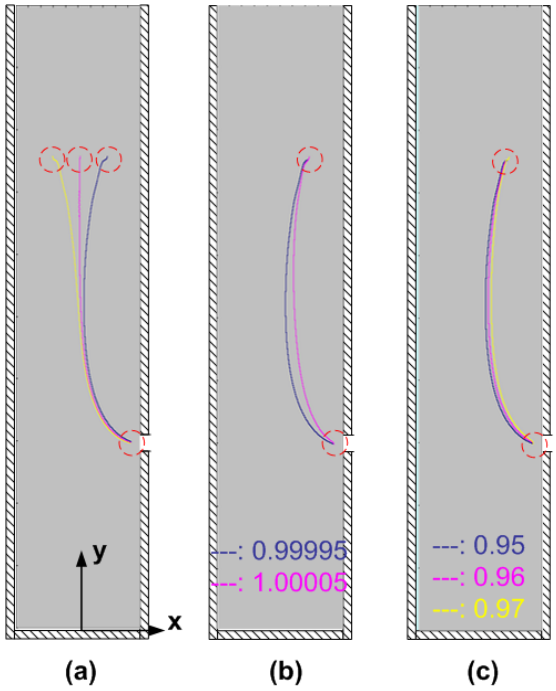

Figure 53. Effects of (a) initial position (left, center, right), (b) particle-fluid density ratio (ρ_s/ρ_f=0.99995 and 1.00005), and (c) hole outlet pressure: 0.95, 096, and 0.97 P_0 where P_0 is the inlet pressure.

Next, we consider two cases with three particles in the channel with six holes. Geometry and boundary conditions are presented in Figure 54. The fluid flows out through side holes by pressure difference, and flow is being supplied from the upper part. The bottom of the channel is closed and six holes are located at the side of the channel. The flow enters at mass flow rate of 1.27 kg/s. The density and viscosity of the fluid are 1 g/cm^3 and 0.01 cm^2/sec, respectively. Three circular buoyant particles with the density of 0.99995 of fluid density are released. A particle with diameter D=2.2 cm is larger than the hole with width D_h=0.9 cm.

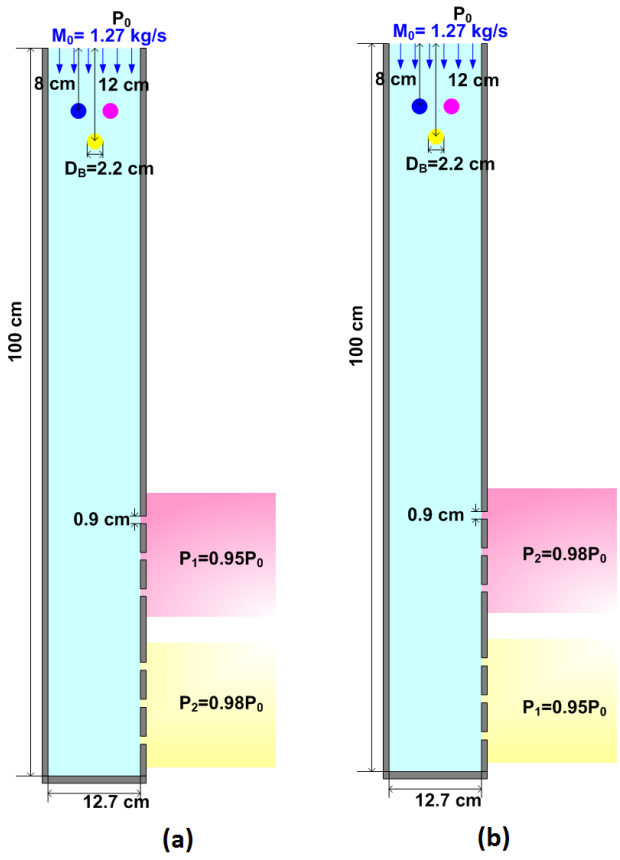

Figure 54. Geometry and initial and boundary conditions in (a) Case 1 and (b) Case 2.

To see the sensitivity of pressure differences, pressures outside the channel are subdivided into two parts. In Case 1, the pressure difference in the upper part (0.05 P_0) is higher than that in lower part (0.02 P_0), whereas the pressure difference in the lower part (0.02 P_0) is higher than that in the upper part (0.05 P_0) in Case 2.

Figure 55 presents the velocity vector field at certain time instants and the trajectory of three particles for each case. From velocity vector fields, we can see that the flow fluctuates due to solid particle movement. Also, from the particle trajectories, it is observed that the final seating positions of particles are different between the two cases. This is because the pressure difference changes affected the timing of collision of two particles (yellow and pink ones). In this simulation, the present IB-LBM reasonably captured the complex interactions of fluid-particle, particle-particle, and particle-wall.

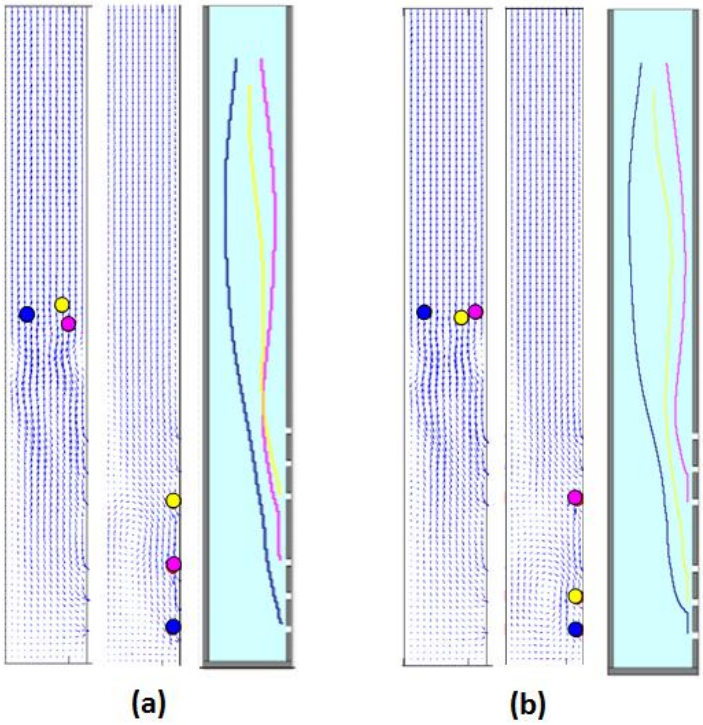

Figure 55. Velocity vector fields and particle trajectories in (a) Case 1 and (b) Case 2.

7. Sedimentation of single 3D particle

In this section, a spherical particle falling in the three-dimensional channel is simulated. In 2D particle problems, the results from the present IB-LBMs were only compared with those from other numerical schemes. Here, the results from our IB-LBM are compared with experimental data measured using PIV by ten Cate et al. [127].

As shown in Figure 56, the channel size is $100\times100\times160$ mm^3, and a spherical particle of 15 mm in diameter and 1120 kg/m^3 in density is released at a height of 120 mm from the bottom. The fluids have densities ranging from 960 to 970 kg/m^3 and kinematic viscosities ranging from 0.058 to 0.373 Ns/m^2. The lattice size is set to $\Delta x=1$ mm. The simulation is performed for four different Reynolds numbers. Physical and numerical conditions for this simulation are summarized in Table 12.

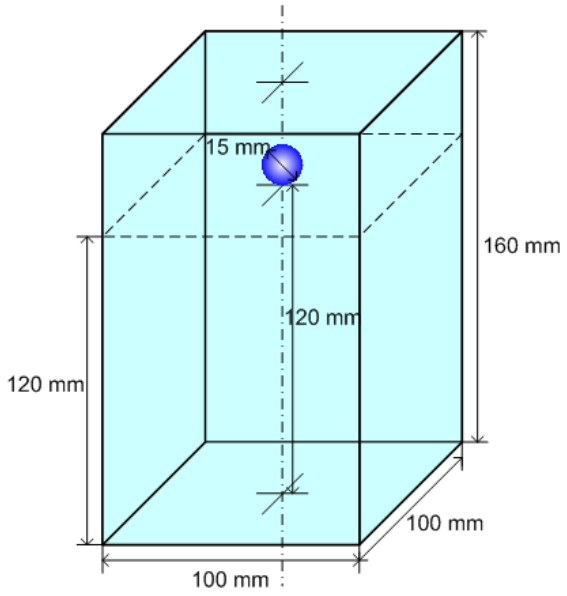

Figure 56. Schematic diagram of a single spherical particle settling in the channel.

Table 12. Physical (experimental) and numerical conditions for single spherical particle sedimentation.

Condition	Re	Fluid density, ρ_f (kg/m^3)	Fluid viscosity, μ_f (Ns/m^2)	Relaxation time, τ	Δt (s)
1	1.5	970	0.373	0.9	3.46738×10^{-4}
2	4.1	965	0.212	0.9	6.06918×10^{-4}
3	11.6	962	0.113	0.8	8.51327×10^{-4}
4	32.2	960	0.058	0.65	8.27586×10^{-4}

Both the diffuse and sharp interface schemes are tested under the D3Q19 SRT-LBE. In the diffuse scheme, we use the 4-point discrete delta function. One practical issue when using the diffuse scheme for sphere geometry is the distribution of forcing points on the sphere surface because it is theoretically impossible to evenly distribute points on the sphere. Therefore, some approximate methods [20, 23, 120] were proposed. Among them, we adopt Feng and Michaelides' method [120] in this calculation. In addition, they assigned forcing points at the volume-based midway of the spherical shell, not on the actual surface. For a particle with radius r, the forcing points resided on a sphere surface with radius $r_b = \sqrt[3]{(r^3 + (r-\Delta x)^3)/2}$, which is similar to the corrected (reduced) radius discussed in the simulation of flow induced by an inline-oscillating cylinder. We consider both the actual and corrected radius in the calculation.

Figure 57 shows the pressure contours and velocity vector fields on the vertical plane passing through the center of the particle in two time instants at Re=4.1. Vortices due to the low pressure in the wake region of the sphere are observed.

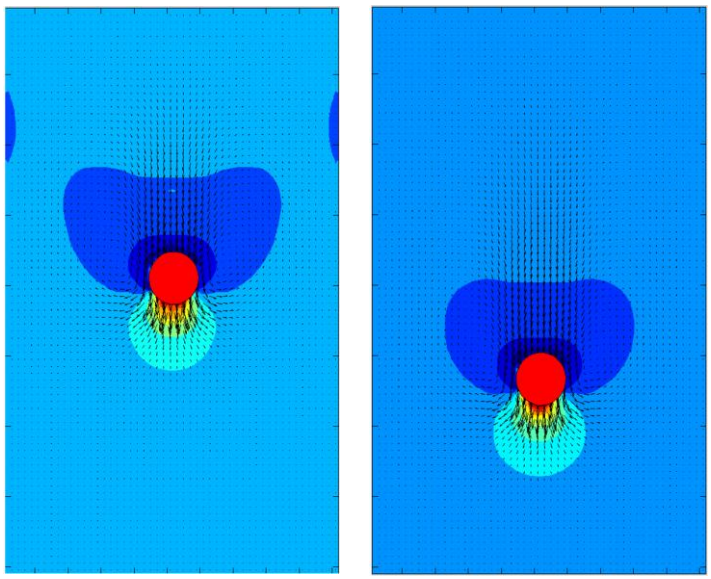

Figure 57. Pressure contours and velocity vector fields on the vertical plane passing through the center of a spherical particle at some time instants in Case 2.

Figure 58 shows the vertical particle velocity variations. Here, we used the corrected radius in the diffuse scheme calculation. Both results from the sharp and diffuse schemes show a good agreement and are comparable to experimental data. If the actual radius is adopted instead of the corrected radius in the diffuse scheme calculation, as shown in Figure 59, vertical particle velocities are under-predicted in magnitude, and the discrepancies become larger at higher Reynolds numbers. This result may justify the adoption of the corrected radius again, as in the simulation of flow induced by an inline-oscillating cylinder.

In this simulation, we validated that the present IB-LBMs with the sharp interface scheme and the diffuse interface scheme with the corrected radius are applicable to 3D moving boundary problems.

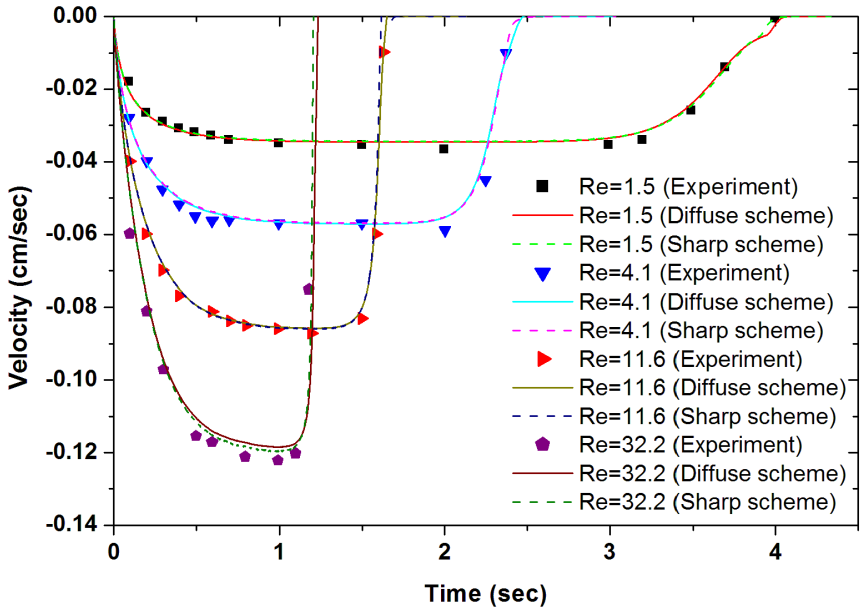

Figure 58. Vertical particle velocity in a spherical particle sedimentation.

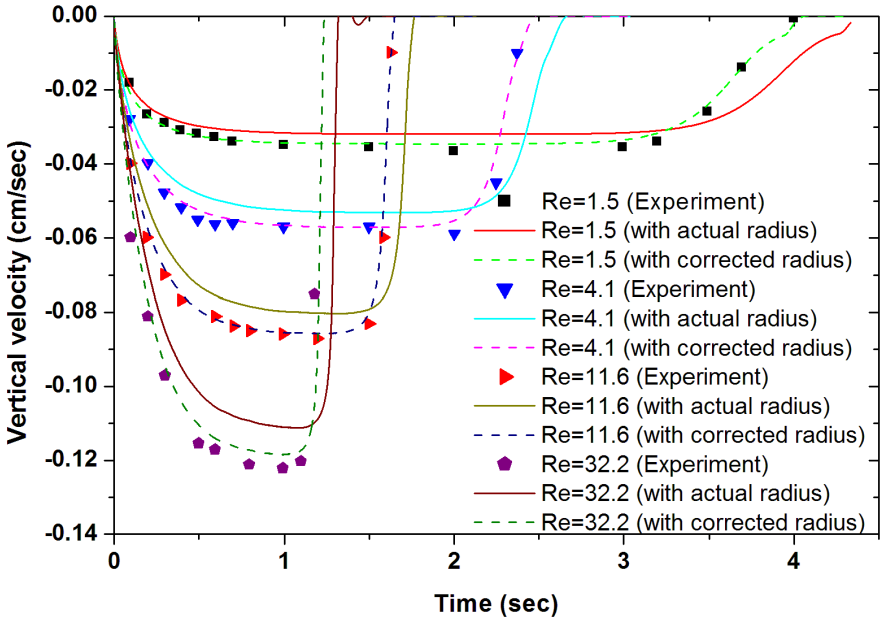

Figure 59. Comparison of vertical particle velocities between the diffuse schemes with actual and corrected radii.

8. Flow-induced vibration of the cylinder

The fluid-structure interaction (FSI) is an important safety issue of nuclear reactor components [128-129]. For example, when continuous repetitive flow is rapidly exerted on immersed long cylinders such as steam generator tube bundles, the resulting flow-induced vibration (FIV) may cause fretting wear on the contact surface with supporting plates.

In this section, we consider an elastically mounted cylinder subjected to a uniform inlet flow. In this problem, oscillating submerged solid objects are treated as a second order mass-spring-damper system. The equation governing the motion of an elastically mounted cylinder that is allowed to move in y-direction can be written as

$$m\ddot{y} + c\dot{y} + ky = F_{f\to s,y} \tag{112}$$

where y is the displacement from the equilibrium position, m is the cylinder mass, c is the damping coefficient, k is the spring constant, and $F_{f\to s,y}$ is the transverse force acting on the cylinder by the fluid. Figure 60 presents the schematic of the mass-spring-damper system adopted here.

The governing equations can be easily coupled with the immersed boundary method. From Equation (90), we can write

$$F_{f\to s,y} = \frac{\partial}{\partial t}\int_{V_s} \rho u_y \, dV - \int_{V_f \cup V_s} F_y dV = m_f \ddot{y} - \int_{V_f \cup V_s} F_y dV \tag{113}$$

where m_f is the fluid mass inside the cylinder. Then Equation (112) becomes

$$(m - m_f)\ddot{y} + c\dot{y} + ky = -\int_{V_f \cup V_s} F_y dV \tag{114}$$

where the term in the right-hand side is directly given by the IBM calculation. By

solving the second-order ordinary differential equation (114), we can obtain the velocity and the position of the cylinder.

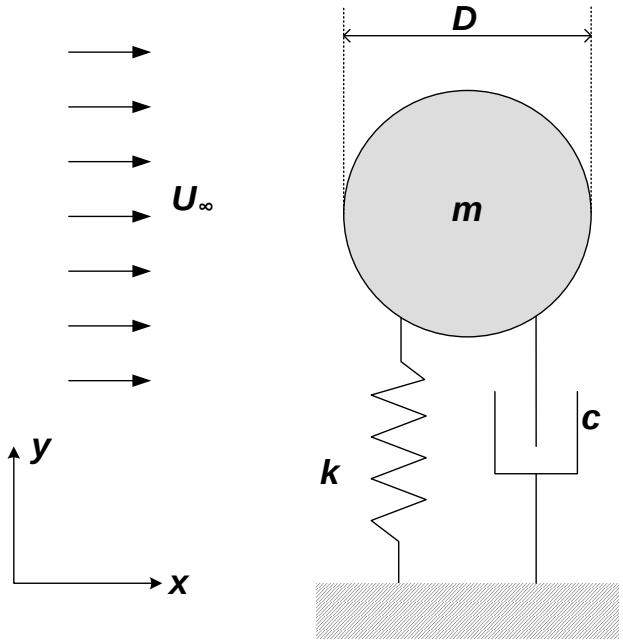

Figure 60. Schematic of an elastically mounted cylinder in y-direction.

For the validation, the flow and structural parameters are chosen according to Ahn and Kallinderis [130], in which the unstructured, finite-element ALE scheme was used, and Borazjani et al. [131] , in which the curvilinear immersed boundary method was used, as follows:

$$\mathrm{Re}=\frac{U_\infty D}{\nu}=150,\ M_{red}=\frac{m}{\rho D^2}=4,\ U_{red}=\frac{U_\infty}{f_n D}=3,4,5,6,7,8 \tag{115}$$

where f_n is the natural frequency of the cylinder structure and is determined by $f_n=(1/2\pi)\sqrt{k/m}$.

The computational domain is set to 40D×20D with D=50Δx and the cylinder is initially located in (10D, 10D). The uniform velocity is imposed on inlet boundary, and Neumann boundary conditions are given on other boundaries. The explicit diffuse scheme with 4-point discrete delta function is adopted for this simulation.

Figure 61 presents the variation of the maximum displacement of the cylinder with changing the reduced velocity (U_{red}). In this figure, we can observe the "lock-in" phenomenon in $U_{red} \in [4,7]$. The present results show a good agreement with those from other numerical calculations [130-131]. Figures 62 and 63 show the wake patterns at some instants and the amplitude variation as a function of time and under U_{red}=5.

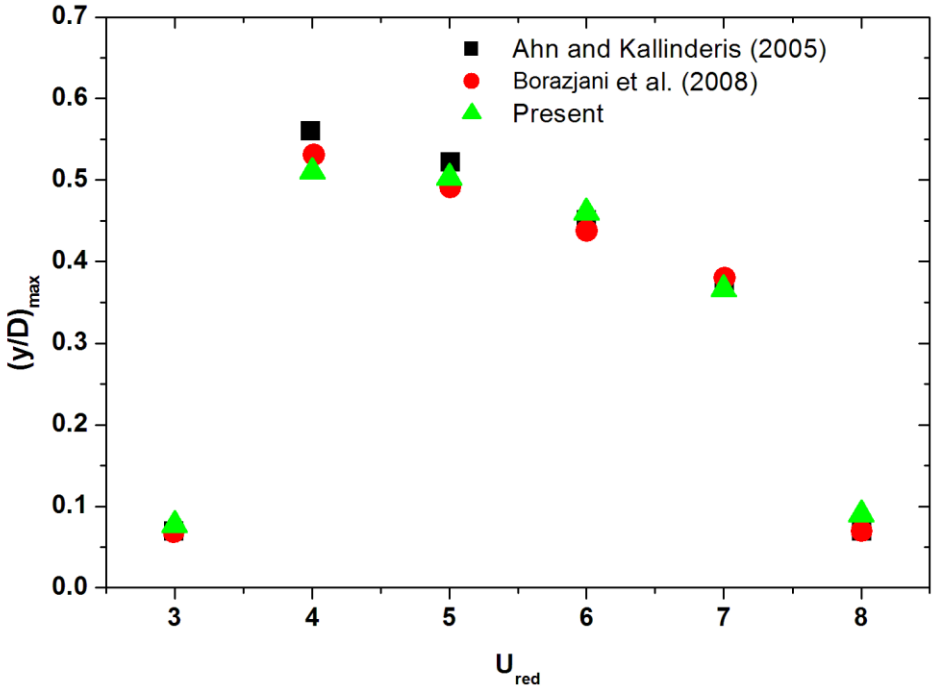

Figure 61. The variation of the maximum displacement of the cylinder as a function of reduced velocity (U_{red}).

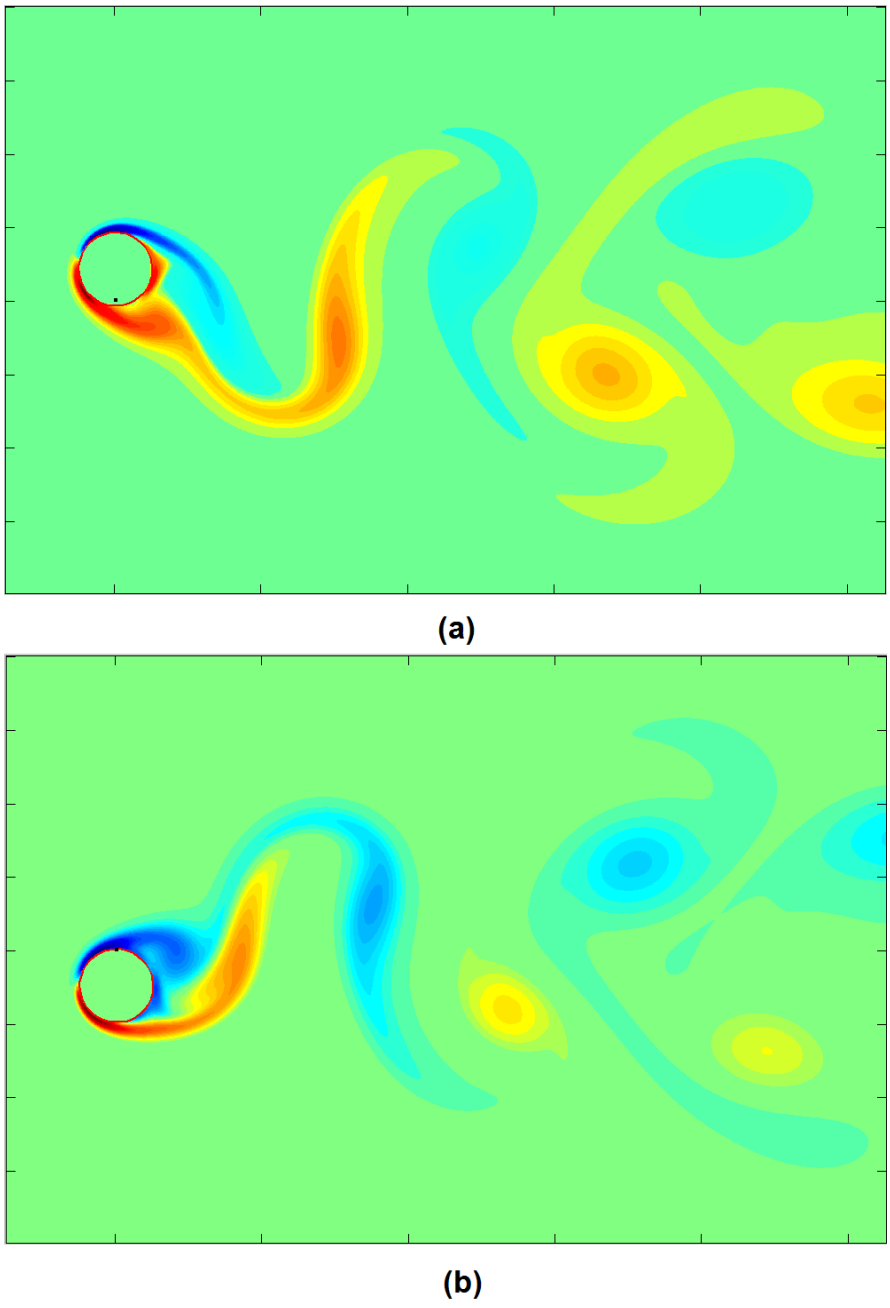

Figure 62. Vorticity contours in the vicinity of the cylinder at (a) t*=200 and (b) t*=240 under Re=150, U_{red}=5, and M_{red}=2.

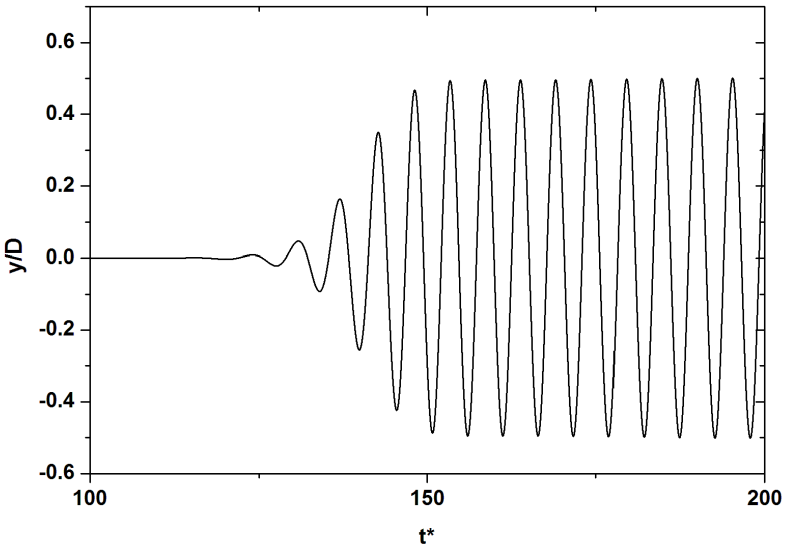

Figure 63. Time variation of the amplitude of the oscillating cylinder under Re=150, U_{red}=5 and M_{red}=2.

D. Conclusions

In this chapter, the direct-forcing IB-LBMs based on the split-forcing LBE with the sharp and/or diffuse interface schemes were applied to various moving boundary problems. The inferences from these applications are as follows:

- The present IB-LBM can be easily coupled with equations governing solid motions, such as particle motion or an elastically mounted cylinder's vibration.
- Both the sharp and diffuse interface schemes can be applied to moving boundary problems.
- However, the sharp interface scheme generates spurious oscillation in the boundary force. This can be due to the discontinuous change of nodes adopted in the interpolation at each time step. Therefore, filtering or smoothing may be

required. Also, when it is applied to moving particle problems with relatively high Reynolds numbers, strange particle behaviors or divergence can occur due to the spurious oscillation. Thus, the sharp interface scheme is recommended to use for mildly moving boundary problems.

- In contrast, the diffuse interface schemes produce smooth variation of the boundary forces. This may be attributed to the use of the discrete delta functions. However, the discrete delta function reduces the accuracy. It is found that in the simulation of moving cylinders or spheres, the adoption of the reduced radius, i.e., assigning forcing points not on the boundary surface (line in 2D) but in the center of the boundary volume (area in 2D), can mitigate the over-prediction of the surface force.
- For moving boundary problems with acceleration, not only the boundary force directly obtained from the immersed boundary method, but also the added mass force should be considered in the surface force evaluation.
- The present IB-LBM with diffuse schemes can be successfully applied to problems with a large number of particles.

CHAPTER V

THERMAL IMMERSED BOUNDARY-LATTICE BOLTZMANN METHODS

A. Introduction

Although numerous IB-LBMs have been applied to isothermal flows, the coupling between the IBM and thermal lattice Boltzmann equation (TLBE) for non-isothermal flows was not documented until the recent work of Jeong et al. [132]. They called the method the immersed boundary-thermal lattice Boltzmann method (IB-TLBM). However, they adopted the feedback-forcing method to evaluate the momentum and energy forcing, thus causing the stability problem and arbitrariness in selecting the parameter. In addition, their method is based on the double-population model with a complex TLBE [63].

The objective of this chapter is to extend the direct-forcing IBM for the isothermal LBM to the thermal LBM. Our strategy is as follows. We consider two thermal LBM models: (i) the double-population model with a simplified thermal LBE and (ii) the hybrid model with an advection-diffusion equation of temperature. We introduce the energy source term for each thermal equation and then derive the direct-forcing IBM formula for both equations, as done in the isothermal IBM. To obtain more accurate boundary effects, we adopt the sharp interface scheme based on bilinear and linear interpolations instead of the diffuse interface scheme based on discrete delta functions. The proposed methods are tested through convective heat transfer problems with not only stationary but also moving boundaries.

The remaining part of this chapter is organized as follows. The direct-forcing method is derived from a simplified internal energy LBE with an energy source term under the double-population TLBM model in Section B.1 and then from an advection-diffusion equation of temperature under the framework of the hybrid TLBM model in Section B.2. To validate the LBM code for these TLBM models, the natural convection in a square cavity is first considered in Section C.1. Then, the present IBM is applied to a fixed-boundary heat transfer problem (the natural convection in a square cavity with an eccentrically located cylinder) in Section C.2, and, subsequently, to a moving-boundary heat transfer problem (a cold particle sedimentation in a hot infinite channel) in Section C.3. In Section D, the summary and conclusions are addressed.

B. Thermal Immersed Boundary-Lattice Boltzmann Methods

The direct-forcing IB-LBM based on the SRT-LBE, derived in Chapter III, is directly adopted for the momentum equation. Instead, for the simulation of the non-isothermal flows with a significant buoyancy force effect, we adopt the Boussinesq approximation, so that the body force term in the SRT-LBE becomes

$$\mathbf{F} = -\rho\beta(T - T_0)\mathbf{g} \tag{116}$$

where T_0 is the average temperature and β is the thermal expansion coefficient at T_0.

To extend the direct-forcing formula to energy equations, we consider a simplified thermal LBE (Equation (38)) under the double population model and a finite difference energy equation under the hybrid model.

1. Immersed boundary method for the thermal lattice Boltzmann equation

In the IBM, we use the energy source term to model the thermal boundary effect. Thus, we need to additionally consider the energy source term in the simplified TLBE. The TLBE with an energy source term can be expressed as [68]:

$$g_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = g_\alpha(\mathbf{x},t) - \frac{1}{\tau_\alpha}\left[g_\alpha(\mathbf{x},t) - g_\alpha^{(eq)}(\mathbf{x},t)\right] + \Delta t\left(1-\frac{1}{2\tau_g}\right)Q_\alpha \tag{117}$$

where the discrete energy source function is

$$Q_\alpha = w_\alpha Q, \tag{118}$$

and Q is the energy source density term. Here, the macroscopic internal energy can be calculated by

$$\rho e = \sum_\alpha g_\alpha + \Delta t Q/2. \tag{119}$$

Applying the Chapman-Enskog multi-scale analysis, we can show that Equation (117) recovers the following energy equation:

$$\frac{\partial}{\partial t}(\rho e) + \nabla\cdot(\rho e\mathbf{u}) = \chi\nabla^2(\rho e) + Q \tag{120}$$

It should be noted that in Equations (117) and (120), the compressible work and the viscous heat dissipation terms are neglected as in Equations (38) and (41).

In the same manner as with the IB-LBM, we can apply the IBM to the non-isothermal lattice Boltzmann equation. After the streaming step, the temperature under no external energy source, T^{noE}, at ($\mathbf{x}$,t+Δt) can be expressed as:

$$\rho(\mathbf{x},t+\Delta t)\frac{c^2}{3T_0}T^{noE}(\mathbf{x},t+\Delta t) = \sum_\alpha g_\alpha(\mathbf{x},t+\Delta t). \tag{121}$$

If the desired temperature, T^{d}, which satisfies the thermal boundary condition on the boundary, is given, then from Equation (119), we can obtain

$$\rho(\mathbf{x},t+\Delta t)\frac{c^2}{T_0}T^d=\sum_{\alpha}g_{\alpha}(\mathbf{x},t+\Delta t)+\frac{\Delta t}{2}Q_b(\mathbf{x},t+\Delta t). \tag{122}$$

Subtracting Equation (121) from Equation (122), we can obtain the following direct-forcing formula for the boundary energy-forcing term:

$$Q_b(\mathbf{x},t+\Delta t)=2\rho(\mathbf{x},t+\Delta t)\frac{c^2}{3T_0}\frac{T^d-T^{noE}(\mathbf{x},t+\Delta t)}{\Delta t}. \tag{123}$$

To evaluate T^{d} on the energy-forcing node in complex boundary problems, where the boundary does not match computational nodes, we can directly use the same interface scheme as in the isothermal IB-LBM in case the Dirichlet-type boundary condition is imposed. Although we only consider the Dirichlet-type boundary condition in this study, if the Neumann-type boundary condition is imposed, we can adopt the procedure of transferring it into the Dirichlet-type boundary condition [29-30] and then apply direct-energy-forcing formula (123).

2. Immersed boundary method for the finite difference energy equation

In the hybrid lattice-Boltzmann finite-difference method (LB FDM), the non-isothermal LBE is solved for velocities and pressures and the finite difference advection-diffusion equation for temperature. For the boundary conditions of the momentum equation – i.e., for no-slip boundary conditions – we adopt the IB-LBM, while for the thermal boundary conditions, we utilize the IBM for the advection-diffusion equation instead of the simplified thermal LBE in the previous section. The energy equation neglecting the

viscous dissipation and the compression work done by the pressure can be expressed as:

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \alpha \nabla^2 T + q \tag{124}$$

with

$$q = \frac{Q}{\rho c_p} \tag{125}$$

where c_p is the specific heat. We explicitly discretize Equation (124) using the first-order forward difference scheme in time and the second-order central difference scheme in space as in [133]. The resulting discretized equation (Model 2) can be expressed as:

$$T_{i,j}^{n+1} = T_{i,j}^{n} + (\mathrm{RHS}^n + q_{i,j}^n)\Delta t \tag{126}$$

with

$$\begin{aligned} \mathrm{RHS}^{\mathrm{n}} = & -4\alpha T_{i,j}^n + (\alpha + 0.5u_{i+1,j}^n)T_{i+1,j}^n \\ & + (\alpha + 0.5u_{i-1,j}^n)T_{i-1,j}^n + (\alpha + 0.5v_{i,j+1}^n)T_{i,j+1}^n + (\alpha + 0.5v_{i,j-1}^n)T_{i,j-1}^n \end{aligned}. \tag{127}$$

Here, Δx=Δy=1 as in the standard LBM, and T is the normalized temperature by the temperature difference. A von Neumann stability analysis of this discretized equation provides the following stability constraint of Model 2 [134]:

$$\frac{u^2 + v^2}{2} \le \alpha \le 0.25. \tag{128}$$

The direct-forcing IBM can be easily applied to this equation. If there were no energy forcing, the temperature at the next time step would become

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \mathrm{RHS}^n \Delta t. \tag{129}$$

Using $T_{i,j}^{noF}$ to discern the temperature from that under external force, we can rewrite

Equation (126) as

$$T_{i,j}^{n+1} = T_{i,j}^{noF} + q_{i,j}^{n}\Delta t \tag{130}$$

From Equations (129) and (130), the energy-forcing term can be evaluated as:

$$q_{i,j}^{n} = \frac{T_{i,j}^{d} - T_{i,j}^{noE}}{\Delta t} \tag{131}$$

It should be pointed out that in the actual calculation, there is no need to explicitly evaluate the energy-forcing term because the explicit time-advancement scheme is adopted [28]. For the interface scheme, we adopt the same method as in the IB-LBM.

C. Simulation Results

1. Natural convection in the square cavity

At first, to test the present LBM code, we considered the natural convection in a square cavity, which has been used as a typical benchmark problem for the validation of the code capability to simulate the natural convection. The square cavity has hot and cold isothermal boundary conditions at left and right vertical sides, respectively, and adiabatic boundary conditions at top and bottom horizontal sides, as shown in Figure 64.

This natural convection is characterized by two non-dimensional numbers: the Rayleigh number (Ra) and the Prandtl number (Pr), which are defined by

$$\mathrm{Ra} \equiv \frac{g\beta\Delta T L^3}{\nu\alpha}, \quad \mathrm{Pr} \equiv \frac{\nu}{\alpha} \tag{132}$$

where g is the gravitational constant, L is the height or the width of the square cavity, and ΔT is the temperature difference between hot and cold walls, i.e., $\Delta T = T_{hot} - T_{cold}$.

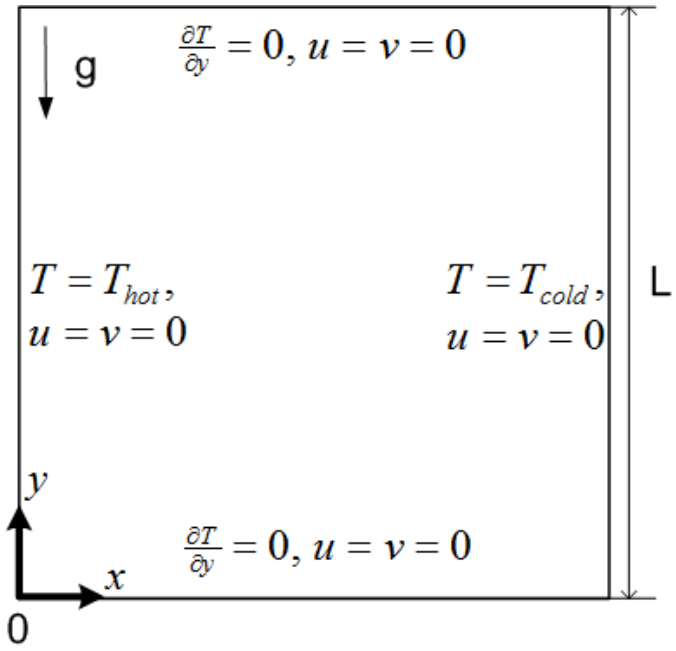

Figure 64. Geometry and boundary conditions for the simulation of the natural convection in a square cavity.

Therefore, viscosity and the corresponding non-dimensional relaxation time, and thermal diffusivity and the corresponding non-dimensional thermal relaxation time, can be written as

$$\nu = \sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}} U_c L; \quad \tau = 3\sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}} U_c L + \frac{1}{2}, \tag{133a}$$

$$\alpha = \frac{U_c L}{\sqrt{\mathrm{Ra}\,\mathrm{Pr}}}; \quad \tau_g = \frac{3U_c L}{2\sqrt{\mathrm{Ra}\,\mathrm{Pr}}} + \frac{1}{2} \tag{133b}$$

where U_c is the characteristic velocity, which is defined by

$$U_c = \sqrt{\beta g \Delta T L}\,. \tag{134}$$

The characteristic velocity should be selected to be small so that the compressibility error remains small.

For the athermal and thermal boundary conditions, the following non-equilibrium bounce-back scheme [63] was adopted

$$f_{\alpha}^{neq} = f_{\bar{\alpha}}^{neq} \tag{135a}$$

$$g_{\alpha}^{neq} = -g_{\bar{\alpha}}^{neq} \tag{135b}$$

where $\mathbf{e}_{\bar{\alpha}} = -\mathbf{e}_{\alpha}$. Here, to apply this scheme to the adiabatic boundary, the temperature on the wall, $T(y_0)$, should be predetermined. For this, the following second-order accurate discretization was adopted:

$$\left.\frac{\partial T}{\partial y}\right|_{y_0} = \frac{1}{2\Delta y}[4T(y_0+\Delta y) - T(y_0+2\Delta y) - 3T(y_0)] + O(\Delta y^2) = 0. \tag{136}$$

Simulations were implemented for Ra=10^3, 10^4, 10^5, and 10^6 with the present method, and in all simulations, the Prandtl number (Pr) was set to be 0.71. The following convergence criteria [64] were used:

$$\max(|\sqrt{(u^{n+1})^2+(v^{n+1})^2} - \sqrt{(u^n)^2+(v^n)^2}\,|) < 10^{-9}, \tag{137a}$$

$$\max(|T^{n+1} - T^n|) < 10^{-9} \tag{137b}$$

where n and n+1 indicate the old and the new time steps, respectively.

Quantitative comparisons of the maximum horizontal velocity ($u_{\max}$) and its vertical position ($y_{\max}$) on the vertical centerline (x=L/2), the maximum vertical velocity ($v_{\max}$) and its horizontal position ($x_{\max}$) on the horizontal centerline (y=L/2), and the average Nusselt number were performed. The average Nusselt number is defined by

$$\mathrm{Nu}_{avg} = \frac{L}{\alpha \Delta T}\frac{1}{L^2}\int_0^L \int_0^L q_x(x,y)\,dxdy \tag{138}$$

with the local heat flux in horizontal direction

$$q_x(x,y) = uT(x,y) - \alpha \frac{\partial T(x,y)}{\partial x} \tag{139}$$

where u is x-direction velocity.

Table 13 displays the effect of the characteristic velocity on the results from both thermal LBM models under Ra=10^4 and the grid size of L/150. It is observed that as the characteristic velocity decreases, the calculation results approach the reference data [135]. Under the condition of $U_c \leq 0.1$, the calculated Nusselt numbers do not show much improvement. Thus, in the remaining calculations, we selected the characteristic velocity as U_c=0.05.

Figure 65 presents errors of the average Nusselt numbers when changing grid sizes from L/50 to L/200 under Ra=10^4 and U_c=0.05. The error is defined by

$$Error = \left| Nu_{avg} - Nu_{avg}^{L/400} \right| \tag{140}$$

where $Nu_{avg}^{L/400}$ indicates the average Nusselt number under the grid size of L/400, which is adopted as an exact solution because the analytical solution does not exist. It was observed that the two models have the second-order accuracy in space. Based on such grid sensitivity studies, the grid sizes were selected as L/100, L/150, L/200, and L/250 for Ra=10^3, 10^4, 10^5, and 10^6, respectively.

Table 13. The effect of the characteristic velocity.

Model	U_c	Nu_{avg}	u_{max}	v_{max}
Model 1	0.01	2.243	16.164	19.610
	0.05	2.243	16.161	19.604
	0.1	2.242	16.157	19.620
	0.2	2.242	16.114	19.565
	0.5	2.215	15.842	19.355
Model 2	0.01	2.243	16.168	19.613
	0.05	2.243	16.165	19.614
	0.1	2.243	16.152	19.596
	0.12	2.242	16.152	19.624
	0.14	2.241	16.147	19.628
Reference	-	2.243	16.178	19.617

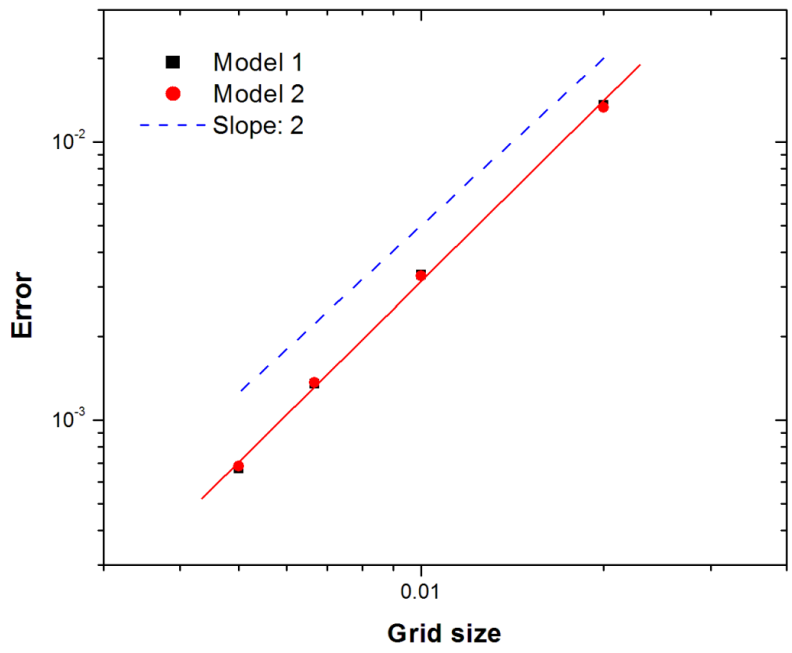

Figure 65. Accuracy of the two thermal LBM models.

Table 14 presents the results for Ra=10^3, 10^4, 10^5, and 10^6. In terms of accuracy, all results showed a good agreement between the two models and the reference calculation [135] within 1%. However, in terms of efficiency, the two models showed some differences. Although the simplified thermal LBE model (Model 1) greatly reduced the CPU time, compared with He et al.'s original double-population thermal LBE model [63], it still spends about 50% more CPU time than the hybrid thermal LBE model (Model 2), as shown in Table 14. This can be attributed to the use of a larger number of energy distribution functions at collision and streaming steps, and the complicated boundary condition calculation in Model 1.

Table 14. Comparison of velocities and Nusselt numbers.

		Ra=10^3	Ra=10^4	Ra=10^5	Ra=10^6
u_{max}	Model 1	3.645	16.161	34.679	64.553
	Model 2	3.646	16.165	34.680	64.596
	Reference	3.649	16.178	34.73	64.63
y_{max}	Model 1	0.810	0.820	0.855	0.848
	Model 2	0.810	0.820	0.855	0.848
	Reference	0.813	0.823	0.855	0.850
v_{max}	Model 1	3.694	19.604	68.527	219.670
	Model 2	3.695	19.614	68.545	219.593
	Reference	3.697	19.617	68.59	219.36
x_{max}	Model 1	0.180	0.120	0.065	0.036
	Model 2	0.180	0.120	0.065	0.036
	Reference	0.178	0.119	0.066	0.0379
Nu_{avg}	Model 1	1.118	2.243	4.514	8.798
	Model 2	1.118	2.243	4.514	8.794
	Reference	1.118	2.243	4.519	8.800
CPU time (sec)	Model 1	396	2539	16182	47741
	Model 2	263	1733	11096	27151

It should be also pointed out that according to Equation (128), Model 2 has the stability constraint of

$$\frac{u^2+v^2}{2} \le \frac{U_c L}{\sqrt{\mathrm{Ra}\,\mathrm{Pr}}} \le 0.25. \tag{141}$$

For low Rayleigh number conditions, for example, under Ra=10^3 and grid size of L/100 and Ra=10^4 and grid size of L/150, Equation (145) provides $U_c \le 0.0666$ and $U_c \le 0.140435$, respectively. Therefore, when we selected $U_c > 0.0666$ under the former condition and $U_c > 0.140435$ under the latter condition, we could not obtain the converged solution, and thus this range was not included in Table 13.

2. Natural convection in the square cavity with an eccentric cylinder inside

To check the applicability of the present IBM to complex, fixed boundary, we considered the natural convection of the fluid in a square cavity with an eccentric circular cylinder. In a square cavity with a width of L, a circular cylinder with a diameter of D=0.4L is eccentrically located in the cavity by 0.1 L upward from the center, as shown in Figure 66. Hot and cold temperatures were imposed on the circular cylinder boundary and vertical sidewalls of the cavity boundary, respectively, and adiabatic conditions are imposed on the horizontal top and bottom walls. This problem has been simulated by various numerical methods under Ra=10^6 and Pr=10 [30-31, 136-137].

In the present simulation, the direct-forcing IBMs based on Model 1 and Model 2 were adopted for the boundary conditions on the circular cylinder wall. For the square

cavity boundary conditions, the non-equilibrium bounce-back scheme was used as in the previous section. The characteristic velocity of 0.2 was adopted.

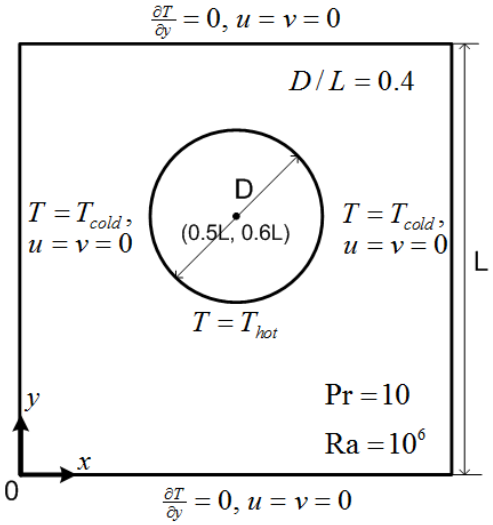

Figure 66. Geometry and boundary conditions of the natural convection in a square cavity with an eccentric cylinder.

Figures 67 and 68 present the isotherms and the streamlines from the IBM based on Model 1 and Model 2 under the grid size of L/200. Due to the buoyancy force, the heated flows around the hot cylinder move upward and the flows cooled by the cold walls move downward along the cold side walls, thus forming the two symmetric free circulations. The isotherms and streamlines show a very good agreement with those in [30]. To quantitatively validate the present method, we compared the local Nusselt numbers along one of the cold sidewalls with different grid sizes of L/100 and L/200 with those in other numerical methods [136-137] as shown in Figure 69. It was observed that the results from the IBM based on both models – even under the grid size of L/100 –

showed a good agreement with those under high resolution although there were some discrepancies near the top wall, which are removed under the grid size of L/200. In terms of numerical efficiency, the IBM based on Model 1 spent 50% more time than the IBM based on Model 2.

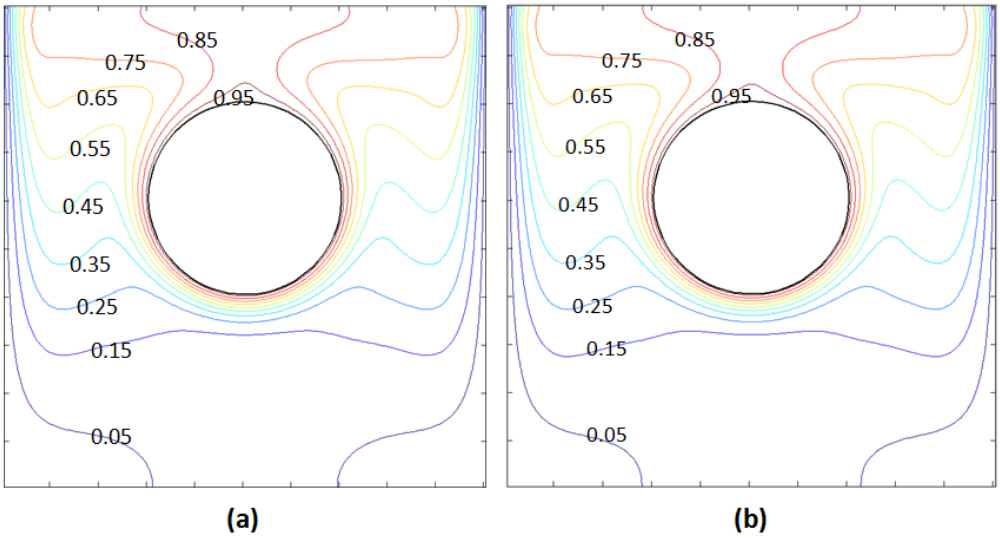

Figure 67. Isotherms obtained from the IBM based on (a) Model 1 and (b) Model 2.

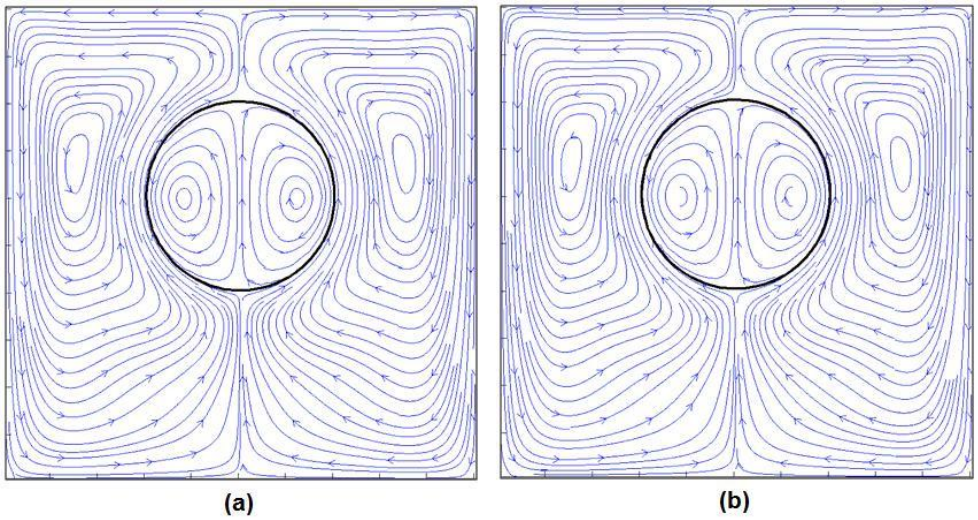

Figure 68. Streamlines obtained from the IBM based on (a) Model 1 and (b) Model 2.

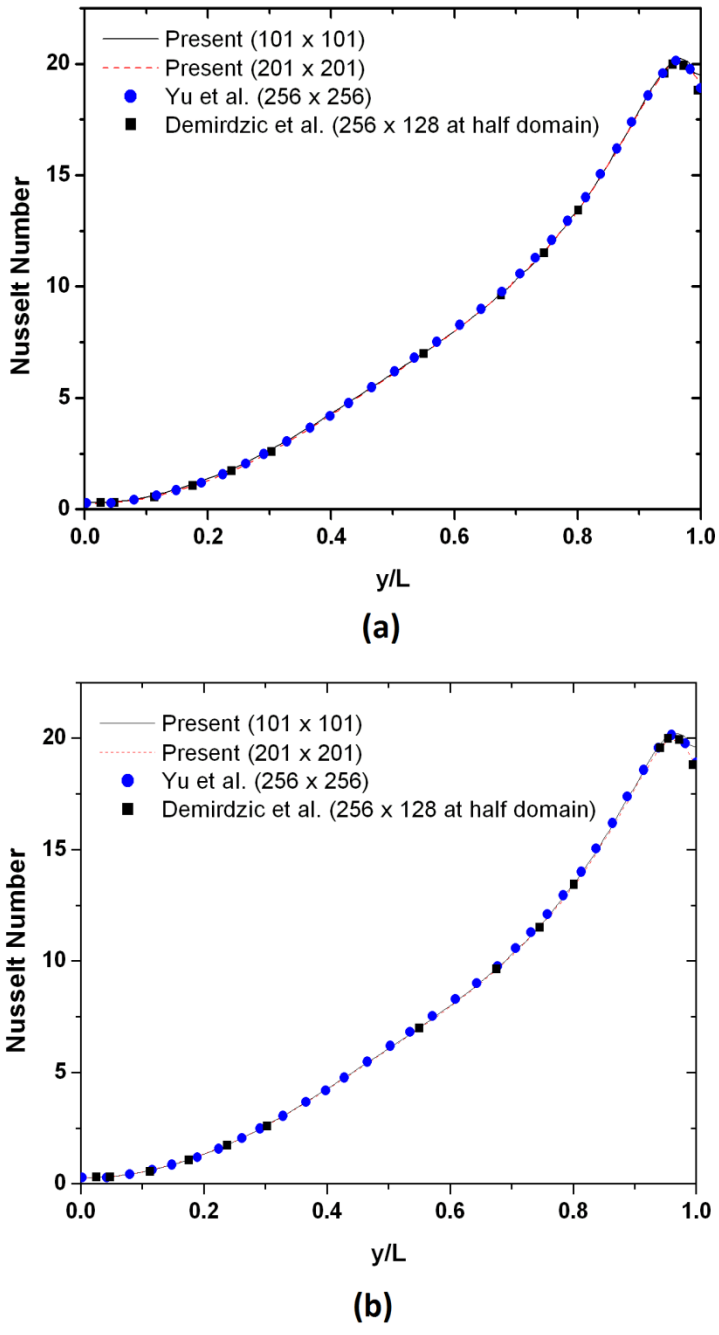

Figure 69. Local Nusselt number variation along the cold wall for L=200Δx from the IBM based on (a) Model 1 and (b) Model 2.

3. Cold particle sedimentation

One of great advantages of the IBM is that it can be easily applied to the moving-boundary problem without re-meshing. To validate the applicability of the present IBM to the heat transfer problem involving moving, complex boundaries, we considered a cold-particle sedimentation in a hot, infinite channel. This problem is challenging because it involves complicated mechanisms between the forced and natural thermal convections and a strong wall confinement effect [137]. First, Gan et al. [138] used the Arbitrary Lagrangian-Eulerian (ALE) method to simulate this problem where the particle had been initially released at the centerline. They suggested the five flow regimes according to different Grashof numbers based on the particle equilibrium positions and the wake structures. Subsequently, Yu et al. [137] used the fictitious domain method to simulate the problem. However, they initially located the particle off the centerline by one particle radius, because the trajectories in some regimes where the particle migrates away from the centerline are not deterministic, since the migrations depend on the random numerical disturbances. They also found the different regimes from Gan et al.'s at high Grashof numbers over Gr>4000. Especially at Gr=4500, it was observed that flows become turbulent-like, and the particle oscillates violently but still regularly. They attributed the difference to the use of fine meshes only for the region in the vicinity of the particle boundary, resulting in a lack of description of the far field in the ALE method. This modified Grashof regime at high Grashof numbers was confirmed by the IBM based on the NSE [31]. In the present simulation, we followed the conditions in [137] so that the direct comparison is possible. As shown in Figure 70, a circular

cylinder with a diameter of D was initially located off the centerline by D/2. The Prandtl number was 0.7, and the density ratio of the solid particle to the fluid (ρ_r) was 1.00232. The cylinder had the constant cold temperature, and the sidewalls had the constant hot temperature. The fluid is initially at rest with the hot temperature.

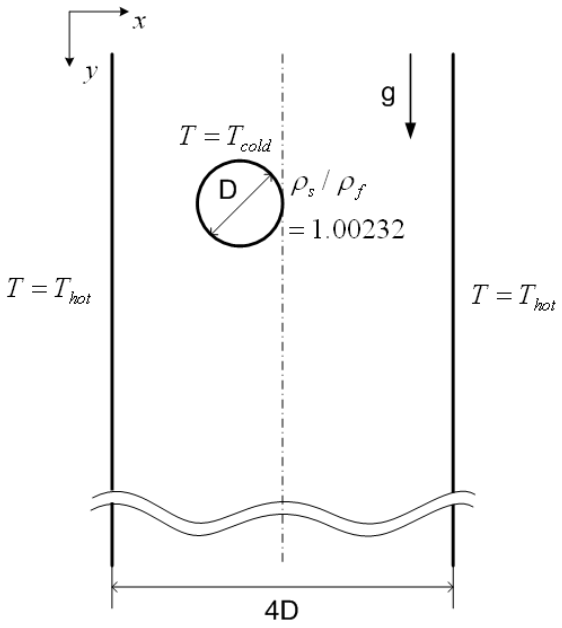

Figure 70. Geometry and boundary conditions in a cold particle sedimentation problem.

The characteristic velocity as adopted in previous sections is no longer valid in this problem because the flow involves forced convection due to particle movement. The relaxation time was selected as 0.65, as in [70]. To determine the gravitation constant in the LBM frame, we used the reference Reynolds number of 40.5, as in [31, 137]. Here, the reference Reynolds number is defined by [137]:

$$\mathrm{Re}_{ref} = \frac{U_{ref} D}{\nu} \tag{142}$$

where the reference velocity U_{ref} is defined by

$$U_{ref} = \sqrt{\pi (D/2)(\rho_r - 1) g}\ . \tag{143}$$

To simulate the infinite channel, the computational domain of 4D×160D was adopted. The present IBMs based on Model 1 and Model 2 were applied to the cold wall particle boundary. The motion of the particle was calculated using Equations (99) and (100).

For isothermal case (i.e. Gr=0), the terminal Reynolds number ($\mathrm{Re}_T = u_T D/\nu$) with the terminal velocity (u_T) obtained from the IB-LBM based on both models was 21.2, which is the same as that in [137].

Figure 71 presents the time evolution of the particle horizontal positions under different Grashof numbers and Figure 72 isotherms and vorticity contours at time t*= 129.6, where the time t* is defined as $t^* = D/U_{\mathrm{ref}}$. The results follow the modified flow regimes well. At Gr=100 (Regime A in [138]), the particle settles steadily along the centerline, and the wake vortices are steady and symmetric (Figures 71 and 72(a)). At Gr=564 (Regime B in [138]), vortex shedding occurs from the particle and the particle oscillates regularly about the centerline (Figures 71 and 72(b)). At Gr=1000 and 2000 (Regime C in [138]), two types of migrations are observed: one with oscillation as a natural extension of Regime B (Figures 71 and 72(c)), and the other without oscillation (Figures 71 and 72(d)). At Gr=2500 (Regime D in [138]), the centerline regains a stable equilibrium position and vortex shedding is absent (Figures 71 and 72(e)). At Gr=4500

(Regime F in [137]), the flows become turbulent-like, and the particle oscillates violently but still regularly (Figures 71 and 72(f)).

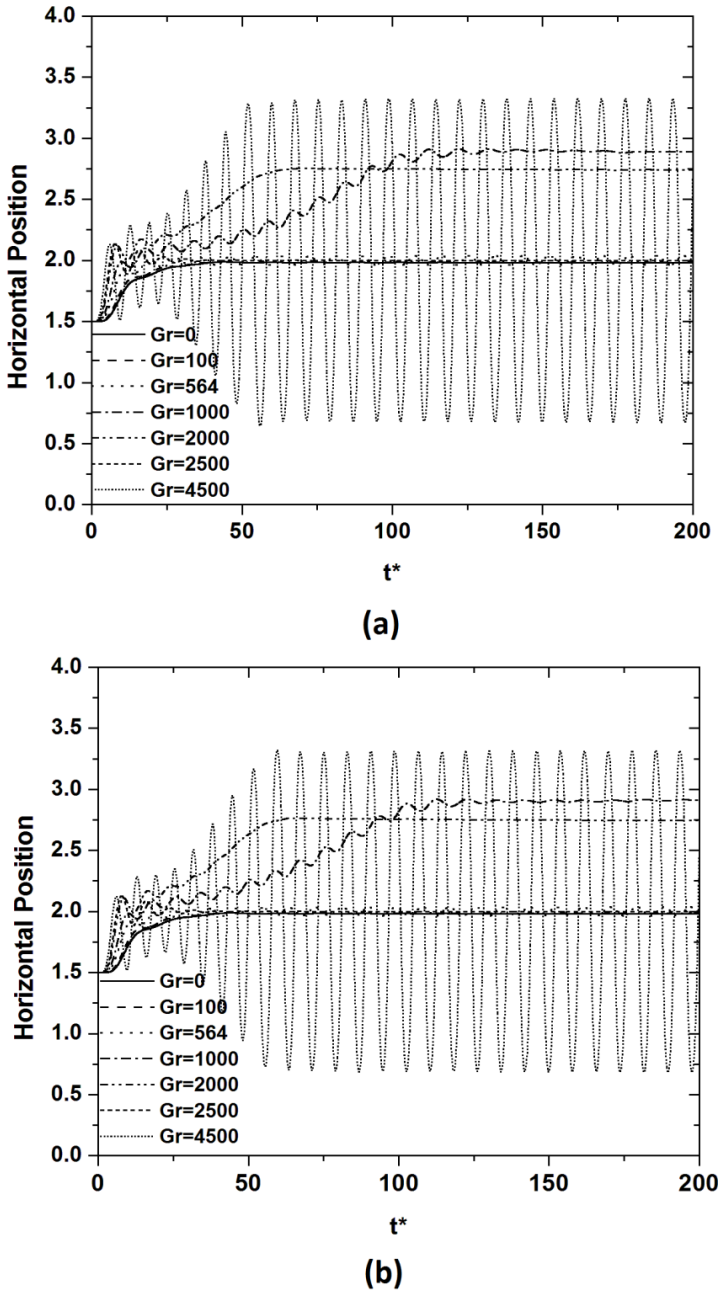

Figure 71. Horizontal position evolutions obtained from the IBM based on (a) Model 1 and (b) Model 2.

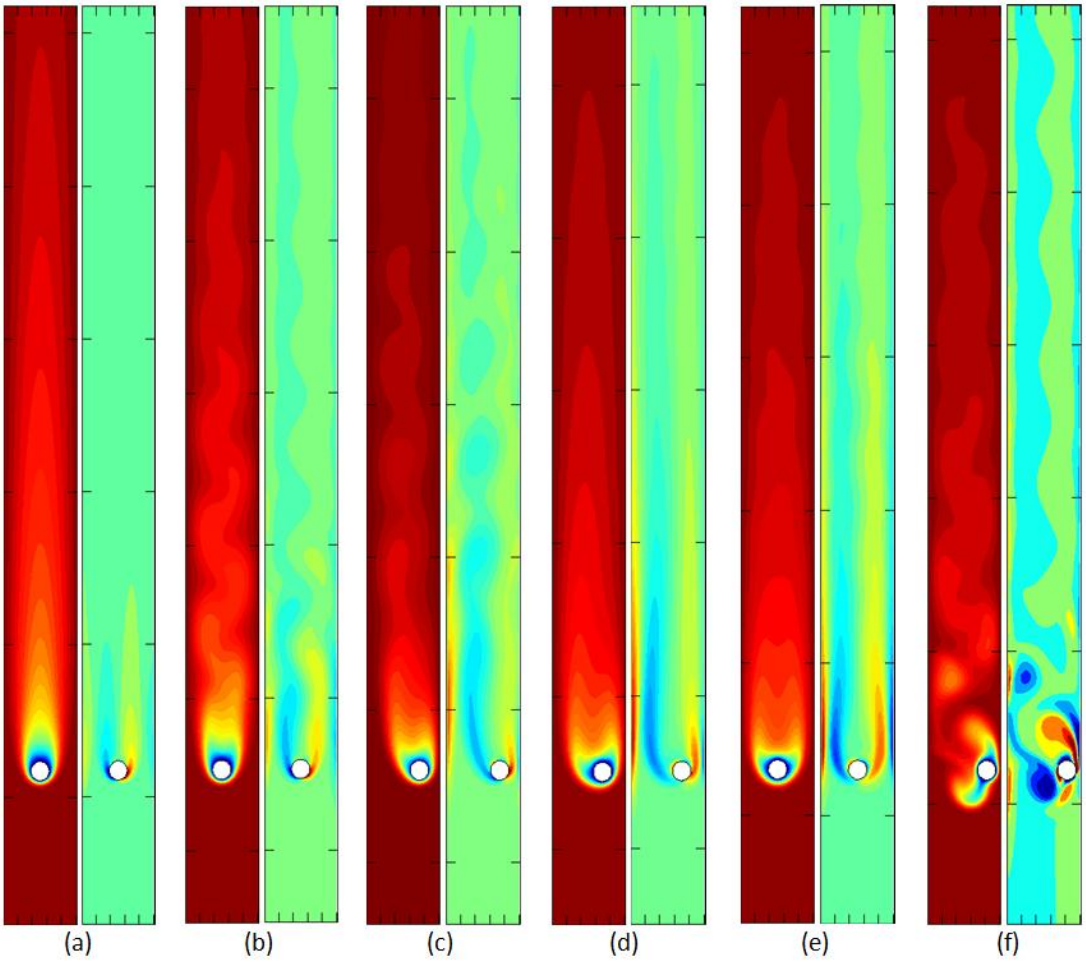

Figure 72. Isotherms (left) and vorticity contours (right) at time t*=129.6 under (a) Gr=100, (b) Gr=564, (c) Gr=1000, (d) Gr=2000, (e) Gr=2500, and (f) Gr=4500.

For the quantitative comparison, we compared the particle horizontal positions between the present results and the previous results. Table 15 presents a comparison of equilibrium positions at Gr=1000 and 2000, and the amplitude at Gr=4500 between the present scheme, the fictitious domain method of [137] and the IBM based on Naiver-Stokes equation of [31]. Results from both models show a good agreement with others.

Table 15. Comparison of equilibrium positions at Gr=1000 and 2000 and amplitudes at Gr=4500.

Gr	Present (Model 1)	Present (Model 2)	Yu et al. [137]	Feng and Michaelides [31]
1000	2.89	2.91	2.89	2.90
2000	2.74	2.74	2.74	2.73
4500	1.32	1.32	1.33	1.35

From this simulation, it is confirmed that the present direct-forcing IBM based on the two thermal LBM models can produce accurate results even for the heat transfer problem with moving boundaries. Again, the IBM based on Model 2 is numerically 50% more efficient than that based on Model 1.

D. Conclusions

In this chapter, we proposed the direct-forcing IBMs coupled with two thermal LBM models: the double-population model with simplified internal energy distribution functions (Model 1) and the hybrid model with a finite difference advection-diffusion equation (Model 2). We showed that the IBMs based on both models had a good accuracy for heat transfer problems with not only stationary but also moving complex boundaries where the Boussinesq approximation is valid, and that the viscous heat dissipation and the compression work done by the pressure was negligible. However, the IBM based on Model 2 was faster than that based on Model 1 by 50%. Therefore, the

IBM based on the hybrid thermal LBM model is recommended for the actual calculation in terms of efficiency as well as accuracy.

In this study, we only considered the SRT lattice Boltzmann equation for velocity and pressure. However, the multiple-relaxation-time (MRT) lattice Boltzmann equation could be adopted for improved stability. In addition, the MRT model could also consider temperature-dependent transport coefficients without explicitly using the Boussinesq approximation [55].

We predict that the direct-forcing formula can be coupled with other simplified TLBE models [65-66]. In addition, it is expected that we can easily extend the present method to 3D problems by adopting the D3Q19 thermal LBE and the interface scheme based on tri-linear, bi-linear, and linear interpolations.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In this dissertation, we presented results from applications of the immersed boundary-lattice Boltzmann method (IB-LBM) to isothermal and non-isothermal flows with stationary and moving boundaries. For this, the direct-forcing formulas based on the split-forcing LBE and TLBE were derived and various interface schemes were coupled with the formulas.

To evaluate and benchmark the effectiveness and applicability of the IB-LBM, we considered the following problems:

- 2D Taylor-Green decaying vortex;
- Flow past a 2D circular cylinder;
- Flow past a sphere;
- Flow in the pebble channel;
- Flow induced by an inline-oscillating cylinder;
- Fluid-induced vibration (FIV) of the cylinder;
- Sedimentation of 2D and 3D particles;
- Natural convection in 2D geometry;
- Particle sedimentation with heat transfer.

We implemented detailed comparisons with other numerical calculations and experimental data if available. Our inferences are as follows:

- The derived IB-LBMs with interface schemes have the second-order accuracy. However, the sharper the adopted interface schemes, the more accurate the results will be. Thus, for complex stationary boundary problems requiring high accuracy, the IB-LBM with the sharp interface scheme is desirable.
- Both the sharp and diffuse interface schemes can be applied to moving boundary problems. However, the sharp interface scheme induces spurious oscillation in the boundary force due to the discontinuous change of nodes adopted in the interpolation, so that filtering or smoothing may be required. Also, when it is applied to moving particle problems with relatively high Reynolds numbers, the spurious oscillation may cause strange particle behaviors or divergence. Thus, the sharp interface scheme is recommended to use for mildly moving boundary problems.
- On the other hand, the diffuse interface schemes produce a smooth variation of the boundary forces because of the discrete delta functions. However, the discrete delta function makes the interface diffuse, thus producing less accurate results. It was found that in the simulation of moving cylinders or spheres, the adoption of the reduced radius, i.e., assigning forcing points not on the boundary surface (line) but the center of the boundary volume (area), can relieve the over-prediction of the surface force.
- The natural convection phenomena with stationary and moving boundaries were well reproduced by the derived direct-forcing thermal immersed boundary-lattice Boltzmann methods based on the thermal LBE and the finite-difference energy

equation. However, in terms of efficiency, the thermal IB-LBM with finite difference energy equation is recommended.

For the future work, we plan to parallelize the present code using the Message Passing Interface (MPI). As mentioned in Chapter II, one of the great advantage of the LBM is that it is very suitable for the MPI parallel computation because the collision step is local and the streaming step is almost local, i.e., just related to the neighboring node information. Hence, information communication between decomposed domains occurs at only the interface at each time step. It should be noted that we implemented the OpenMP parallel computations for some 3D simulations such as flow past a 3D sphere and flow in the pebble channel. However, the parallelization in the OpenMP is mostly performed for loop calculations and based on the shared memory concept, thus limiting the number of CPUs available. Hence, the MPI programming, where a parallel program is running on a distributed memory system, is desirable.

In the MPI code, we will apply the present IB-LBM to turbulent flow simulation using the large eddy simulation (LES) turbulence model [56-58, 139-142] for actual nuclear engineering applications. These applications could include, for example, flow in a rod bundle with mixing vanes [143-145] and flow in a pebble bed [109-111] as stationary complex boundary problems, flow-induced vibration in steam generator tube bundles [146] and pebble defueling process in liquid-salt-cooled pebble bed reactors[147] as moving boundary problems, and T-junction mixing [148-150] as a

thermal complex boundary problem. In this dissertation, we have conducted preliminary research in some of these areas.

Other possible applications are two-phase bubbly flow by coupling the surface tension force, which can be obtained from other methods such as the Moving Particle Semi-implicit (MPS) method [151], with the present IB-LBM, non-Newtonian fluid flow induced by a rotating cylinder [152] by applying the current IBM to non-Newtonian LBE, and other fluid mechanics problems involving complex or moving boundaries.

REFERENCES

1. Peskin CS. Flow patterns around heart valves - numerical method. *Journal of Computational Physics* 1972; **10**:252-271.
2. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**:220-252.
3. Lai MC, Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics* 2000; **160**:705-719.
4. Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force-field. *Journal of Computational Physics* 1993; **105**:354-366.
5. Saiki EM, Biringen S. Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. *Journal of Computational Physics* 1996; **123**:450-465.
6. Feng ZG, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. *Journal of Computational Physics* 2004; **195**:602-628.
7. Mohd-Yusof J. Combined immersed boundaries/B-spline methods for simulations of flows in complex geometries. CTR Annual Research Briefs, NASA Ames/Stanford University, Stanford, CA, 1997.
8. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics* 2000; **161**:35-60.

9. Kim J, Kim D, Choi H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics* 2001; **171**:132-150.

10. Balaras E. Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. *Computers & Fluids* 2004; **33**:375-404.

11. Gilmanov A, Sotiropoulos F, Balaras E. A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids. *Journal of Computational Physics* 2003; **191**:660-669.

12. Choi JI, Oberoi RC, Edwards JR, Rosati JA. An immersed boundary method for complex incompressible flows. *Journal of Computational Physics* 2007; **224**:757-784.

13. Ikeno T, Kajishima T. Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations. *Journal of Computational Physics* 2007; **226**:1485-1508.

14. Majumdar S, Iaccarino G, Durbin P. RANS solvers with adaptive structured boundary non-conformal grids. CTR Annual Research Briefs, NASA Ames/Stanford University, Stanford, CA, 2001.

15. Iaccarino G, Verzicco R. Immersed boundary technique for turbulent flow simulations. *Applied Mechanics Reviews* 2003; **56**:331-347.

16. Tseng YH, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593-623.

17. Ghias R, Mittal R, Dong H. A sharp interface immersed boundary method for compressible viscous flows. *Journal of Computational Physics* 2007; **225**:528-553.

18. Shen TWH, Ting HF, Lin RK. An immersed boundary method for the incompressible Navier-Stokes equations in complex geometry. *International Journal for Numerical Methods in Fluids* 2008; **56**:877-898.

19. Wang ZL, Fan JR, Luo K. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. *International Journal of Multiphase Flow* 2008; **34**:283-302.

20. Feng ZG, Michaelides EE. Proteus: A direct forcing method in the simulations of particulate flows. *Journal of Computational Physics* 2005; **202**:20-51.

21. Dupuis A, Chatelain P, Koumoutsakos P. An immersed boundary-lattice-Boltzmann method for the simulation of the flow past an impulsively started cylinder. *Journal of Computational Physics* 2008; **227**:4486-4498.

22. Lima E Silva ALF, Silveira-Neto A, Damasceno JJR. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *Journal of Computational Physics* 2003; **189**:351-370.

23. Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics* 2005; **209**:448-476.

24. Su SW, Lai MC, Lin CA. An immersed boundary technique for simulating complex flows with rigid boundary. *Computers & Fluids* 2007; **36**:313-324.

25. Le DV, Khoo BC, Lim KM. An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:2119-2130.

26. Luo K, Wang Z, Fan JR, Cen KF. Full-scale solutions to particle-laden flows: Multidirect forcing and immersed boundary method. *Physical Review E* 2007; **76**:066709.

27. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239-261.

28. Yang JM, Balaras E. An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics* 2006; **215**:12-40.

29. Kim J, Choi HC. An immersed-boundary finite-volume method for simulation of heat transfer in complex geometries. *KSME International Journal* 2004; **18**:1026-1035.

30. Pacheco JR, Pacheco-Vega A, Rodic T, Peck RE. Numerical simulations of heat transfer and fluid flow problems using an immersed-boundary finite-volume method on nonstaggered grids. *Numerical Heat Transfer Part B-Fundamentals* 2005; **48**:1-24.

31. Feng ZG, Michaelides EE. Heat transfer in particulate flows with direct numerical simulation (DNS). *International Journal of Heat and Mass Transfer* 2009; **52**:777-786.

32. Zhang N, Zheng ZC, Eckels S. Study of heat-transfer on the surface of a circular cylinder in flow using an immersed-boundary method. *International Journal of Heat and Fluid Flow* 2008; **29**:1558-1566.

33. Gilmanov A, Acharya S. A computational strategy for simulating heat transfer and flow past deformable objects. *International Journal of Heat and Mass Transfer* 2008; **51**:4415-4426.

34. Zhang N, Zheng ZC. An improved direct-forcing immersed-boundary method for finite difference applications. *Journal of Computational Physics* 2007; **221**:250-268.

35. Feng ZG, Michaelides EE. Inclusion of heat transfer computations for particle laden flows. *Physics of Fluids* 2008; **20**:040604.

36. Chen S, Doolen GD. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics* 1998; **30**:329-364.

37. Yu DZ, Mei RW, Luo LS, Shyy W. Viscous flow computations with the method of lattice Boltzmann equation. *Progress in Aerospace Sciences* 2003; **39**:329-367.

38. Niu XD, Shu C, Chew YT, Peng Y. A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows. *Physics Letters A* 2006; **354**:173-182.

39. Ziegler DP. Boundary conditions for lattice Boltzmann simulations. *Journal of Statistical Physics* 1993; **71**:1171-1177.

40. Kang SK, Hassan YA. An immersed boundary-lattice Boltzmann method for large particle sedimentation. *International Topical Meeting on Nuclear Reactor Thermal Hydraulics, Operations and Safety (NUTHOS7)*, Seoul, Korea, 2008.

41. Wu J, Shu C. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *Journal of Computational Physics* 2009; **228**:1963-1979.

42. Guo ZL, Zheng CG, Shi BC. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E* 2002; **65**:046308.

43. Yuan P, Schaefer L. A thermal lattice Boltzmann two-phase flow model and its application to heat transfer problems - Part 1. Theoretical foundation. *Journal of Fluids Engineering-Transactions of the ASME* 2006; **128**:142-150.

44. Chen SY, Wang Z, Shan XW, Doolen GD. Lattice Boltzmann computational fluid dynamics in three dimensions. *Journal of Statistical Physics* 1992; **68**:379-400.

45. Qian YH, d'Humieres D, Lallemand P. Lattice BGK models for Navier-Stokes equation. *Europhysics Letters* 1992; **17**:479-484.

46. Frisch U, Hasslacher B, Pomeau Y. Lattice-gas automata for the Navier-Stokes equation. *Physical Review Letters* 1986; **56**:1505-1508.

47. Wolfram S. Cellular automaton fluids. 1. Basic theory. *Journal of Statistical Physics* 1986; **45**:471-526.

48. He XY, Luo LS. A priori derivation of the lattice Boltzmann equation. *Physical Review E* 1997; **55**:R6333-R6336.

49. Higuera FJ, Succi S, Benzi R. Lattice gas dynamics with enhanced collisions. *Europhysics Letters* 1989; **9**:345-349.

50. Bhatnagar PL, Gross EP, Krook M. A model for collision processes in gases. 1. Small amplitude processes in charged and neutral one-component systems. *Physical Review* 1954; **94**:511-525.

51. Kandhai D, Koponen A, Hoekstra A, Kataja M, Timonen J, Sloot PMA. Implementation aspects of 3D lattice-BGK: Boundaries, accuracy, and a new fast relaxation method. *Journal of Computational Physics* 1999; **150**:482-501.

52. Lallemand P, Luo LS. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Physical Review E* 2000; **61**:6546-6562.

53. Ginzburg I, d'Humieres D. Multireflection boundary conditions for lattice Boltzmann models. *Physical Review E* 2003; **68**:066614.

54. d'Humieres D, Ginzburg I, Krafczyk M, Lallemand P, Luo LS. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society of London Series A-Mathematical Physical and Engineering Sciences* 2002; **360**:437-451.

55. Lallemand P, Luo LS. Theory of the lattice Boltzmann method: Acoustic and thermal properties in two and three dimensions. *Physical Review E* 2003; **68**:036706.

56. Premnath KN, Abraham J. Three-dimensional multi-relaxation time (MRT) lattice-Boltzmann models for multiphase flow. *Journal of Computational Physics* 2007; **224**:539-559.

57. Yu H, Luo LS, Girimaji SS. LES of turbulent square jet flow using an MRT lattice Boltzmann model. *Computers & Fluids* 2006; **35**:957-965.

58. Premnath KN, Pattison MJ, Banerjee S. Generalized lattice Boltzmann equation with forcing term for computation of wall-bounded turbulent flows. *Physical Review E* 2009; **79**:026703.

59. Lammers P, Beronov KN, Volkert R, Brenner G, Durst F. Lattice BGK direct numerical simulation of fully developed turbulence in incompressible plane channel flow. *Computers & Fluids* 2006; **35**:1137-1153.

60. McNamara G, Alder B. Analysis of the lattice Boltzmann treatment of hydrodynamics. *Physica A* 1993; **194**:218-228.

61. Alexander FJ, Chen S, Sterling JD. Lattice Boltzmann thermohydrodynamics. *Physical Review E* 1993; **47**:R2249-R2252.

62. Mcnamara GR, Garcia AL, Alder BJ. Stabilization of thermal lattice Boltzmann models. *Journal of Statistical Physics* 1995; **81**:395-408.

63. He X, Chen S, Doolen GD. A novel thermal model for the lattice Boltzmann method in incompressible limit. *Journal of Computational Physics* 1998; **146**:282-300.

64. Peng Y, Shu C, Chew YT. Simplified thermal lattice Boltzmann model for incompressible thermal flows. *Physical Review E* 2003; **68**:026701.

65. Guo ZL, Shi BC, Zheng CG. A coupled lattice BGK model for the Boussinesq equations. *International Journal for Numerical Methods in Fluids* 2002; **39**:325-342.

66. Li Q, He YL, Wang Y, Tang GH. An improved thermal lattice Boltzmann model for flows without viscous heat dissipation and compression work. *International Journal of Modern Physics C* 2008; **19**:125-150.

67. Yuan P, Schaefer L. A thermal lattice Boltzmann two-phase flow model and its application to heat transfer problems - Part 2. Integration and validation. *Journal of Fluids Engineering-Transactions of the ASME* 2006; **128**:151-156.

68. Wang JK, Wang M, Li ZX. A lattice Boltzmann algorithm for fluid-solid conjugate heat transfer. *International Journal of Thermal Sciences* 2007; **46**:228-234.

69. Zu YQ, Yan YY, Shi WP, Ren LQ. Numerical method of lattice Boltzmann simulation for flow past a rotating circular cylinder with heat transfer. *International Journal of Numerical Methods for Heat & Fluid Flow* 2008; **18**:766-782.

70. Mandujano F, Rechtman R. Thermal levitation. *Journal of Fluid Mechanics* 2008; **606**:105-114.

71. Barrios G, Rechtman R, Rojas J, Tovar R. The lattice Boltzmann equation for natural convection in a two-dimensional cavity with a partially heated wall. *Journal of Fluid Mechanics* 2005; **522**:91-100.

72. Xuan YM, Yu K, Li Q. Investigation on flow and heat transfer of nanofluids by the thermal lattice Boltzmann model. *Progress in Computational Fluid Dynamics* 2005; **5**:13-19.

73. Filippova O, Hanel D. Lattice-BGK model for low Mach number combustion. *International Journal of Modern Physics C* 1998; **9**:1439-1445.

74. Filippova O, Hanel D. A novel lattice BGK approach for low Mach number combustion. *Journal of Computational Physics* 2000; **158**:139-160.

75. Peskin CS. Immersed boundary method. *Acta Numerica* 2002; **11**:479-517.

76. Yang XL, Zhang X, Li ZL, He GW. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics* 2009; **228**:7821-7836.

77. He XY, Zou QS, Luo LS, Dembo M. Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model. *Journal of Statistical Physics* 1997; **87**:115-136.

78. Luo LS. Theory of the lattice Boltzmann method: Lattice Boltzmann models for nonideal gases. *Physical Review E* 2000; **62**:4982-4996.

79. Cheng YG, Li JP. Introducing unsteady non-uniform source terms into the lattice Boltzmann model. *International Journal for Numerical Methods in Fluids* 2008; **56**:629-641.

80. Premnath KN, Abraham J. Lattice Boltzmann simulations of drop-drop interactions in two-phase flows. *International Journal of Modern Physics C* 2005; **16**:25-44.

81. He XY, Shan XW, Doolen GD. Discrete Boltzmann equation model for nonideal gases. *Physical Review E* 1998; **57**:R13-R16.

82. Premnath KN, Abraham J. Lattice Boltzmann model for axisymmetric multiphase flows. *Physical Review E* 2005; **71**:056706.

83. McCracken ME, Abraham J. Multiple-relaxation-time lattice-Boltzmann model for multiphase flow. *Physical Review E* 2005; **71**:036701.

84. Sui Y, Chew YT, Roy P, Low HT. A hybrid immersed-boundary and multi-block lattice Boltzmann method for simulating fluid and moving-boundaries interactions. *International Journal for Numerical Methods in Fluids* 2007; **53**:1727-1754.

85. Guo ZL, Zheng CG, Shi BC. Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method. *Chinese Physics* 2002; **11**:366-374.

86. Mansy H, Yang PM, Williams DR. Quantitative measurements of 3-dimensional structures in the wake of a circular cylinder. *Journal of Fluid Mechanics* 1994; **270**:277-296.

87. Barkley D, Henderson RD. Three-dimensional Floquet stability analysis of the wake of a circular cylinder. *Journal of Fluid Mechanics* 1996; **322**:215-241.

88. Rohde M, Kandhai D, Derksen JJ, van den Akker HEA. A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes. *International Journal for Numerical Methods in Fluids* 2006; **51**:439-468.

89. Fornberg B. A numerical study of steady viscous-flow past a circular cylinder. *Journal of Fluid Mechanics* 1980; **98**:819-855.

90. Park J, Kwon K, Choi H. Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160. *KSME International Journal* 1998; **12**:1200-1205.

91. Wang ZL, Fan JR, Cen KF. Immersed boundary method for the simulation of 2D viscous flow based on vorticity-velocity formulations. *Journal of Computational Physics* 2009; **228**:1504-1520.

92. Ye T, Mittal R, Udaykumar HS, Shyy W. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics* 1999; **156**:209-240.

93. Liu C, Zheng X, Sung CH. Preconditioned multigrid methods for unsteady incompressible flows. *Journal of Computational Physics* 1998; **139**:35-57.

94. Williamson CHK. Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics* 1989; **206**:579-627.

95. Apte SV, Martin M, Patankar NA. A numerical method for fully resolved simulation (FRS) of rigid particle-flow interactions in complex flows. *Journal of Computational Physics* 2009; **228**:2712-2738.

96. Mittal R. A Fourier-Chebyshev spectral collocation method for simulating flow past spheres and spheroids. *International Journal for Numerical Methods in Fluids* 1999; **30**:921-937.

97. Mittal R, Dong H, Bozkurttas M, Najjar FM, Vargas A, von Loebbecke A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics* 2008; **227**:4825-4852.

98. Constantinescu GS, Squires KD. LES and DES investigations of turbulent flow over a sphere at Re=10,000. *Flow Turbulence and Combustion* 2003; **70**:267-298.

99. Clift R, Grace J, Weber ME. *Bubbles, drops, and particles*. New York: Academic Press; 1978.

100. Johnson TA, Patel VC. Flow past a sphere up to a Reynolds number of 300. *Journal of Fluid Mechanics* 1999; **378**:19-70.

101. Marella S, Krishnan S, Liu H, Udaykumar HS. Sharp interface Cartesian grid method I: An easily implemented technique for 3D moving boundary computations. *Journal of Computational Physics* 2005; **210**:1-31.

102. Jeong J, Hussain F. On the identification of a vortex. *Journal of Fluid Mechanics* 1995; **285**:69-94.

103. Martin H. Low Peclet number particle-to-fluid heat and mass transfer in packed beds. *Chemical Engineering Science* 1978; **33**:913-919.

104. Nemec D, Levec J. Flow through packed bed reactors: 1. Single-phase flow. *Chemical Engineering Science* 2005; **60**:6947-6957.

105. Yu JA, Zhang MC, Fan WD, Zhou YG, Zhao GF. Study on performance of the ball packed-bed regenerator: Experiments and simulation. *Applied Thermal Engineering* 2002; **22**:641-651.

106. Choi YS, Kim SJ, Kim D. A semi-empirical correlation for pressure drop in packed beds of spherical particles. *Transport in Porous Media* 2008; **75**:133-149.

107. Driscoll MJ, Hejzlar P. Reactor physics challenges in GEN-IV reactor design. *Nuclear Engineering and Technology* 2005; **37**:1-10.

108. Calis HPA, Nijenhuis J, Paikert BC, Dautzenberg FM, van den Bleek CM. CFD modelling and experimental validation of pressure drop and flow profile in a novel structured catalytic reactor packing. *Chemical Engineering Science* 2001; **56**:1713-1720.

109. Lee JJ, Yoon SJ, Park GC, Lee WJ. Turbulence-induced heat transfer in PBMR core using LES and RANS. *Journal of Nuclear Science and Technology* 2007; **44**:985-996.

110. Kim MH, Lim HS, Lee WJ. Computational fluid dynamics assessment of the local hot core temperature in a pebble-bed type very high temperature reactor. *Journal of Engineering for Gas Turbines and Power-Transactions of the ASME* 2009; **131**:012905.

111. Lee JJ, Park GC, Kim KY, Lee WJ. Numerical treatment of pebble contact in the flow and heat transfer analysis of a pebble bed reactor core. *Nuclear Engineering and Design* 2007; **237**:2183-2196.

112. Ergun S. Fluid flow through packed columns. *Chemical Engineering Progress* 1952; **48**:89-94.

113. German Nuclear Safety Standards Commission (KTA). Reactor core design of high-temperature gas-cooled reactor. Part 3: Loss of pressure through friction in pebble bed cores. *KTA 3102.3.*, 1981.

114. Premnath KN, Pattison MJ, Banerjee S. Steady state convergence acceleration of the generalized lattice Boltzmann equation with forcing term through preconditioning. *Journal of Computational Physics* 2009; **228**:746-769.

115. Revstedt J. A virtual boundary method with improved computational efficiency using a multi-grid method. *International Journal for Numerical Methods in Fluids* 2004; **45**:775-795.

116. Hu HH, Patankar NA, Zhu MY. Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian-Eulerian technique. *Journal of Computational Physics* 2001; **169**:427-462.

117. Farhat C, Pierson K, Degand C. Multidisciplinary simulation of the maneuvering of an aircraft. *Engineering with Computers* 2001; **17**:16-27.

118. Dutsch H, Durst F, Becker S, Lienhart H. Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan-Carpenter numbers. *Journal of Fluid Mechanics* 1998; **360**:249-271.

119. Liao CC, Chang YW, Lin CA, McDonough JM. Simulating flows with moving rigid boundary using immersed-boundary method. *Computers & Fluids* 2010; **39**:152-167.

120. Feng ZG, Michaelides EE. Robust treatment of no-slip boundary condition and velocity updating for the lattice-Boltzmann simulation of particulate flows. *Computers & Fluids* 2008; **38**:370-381.

121. Miller LA, Peskin CS. When vortices stick: An aerodynamic transition in tiny insect flight. *Journal of Experimental Biology* 2004; **207**:3073-3088.

122. Shen LW, Chan ES, Lin PZ. Calculation of hydrodynamic forces acting on a submerged moving object using immersed boundary method. *Computers & Fluids* 2009; **38**:691-702.

123. Hofler K, Schwarzer S. Navier-Stokes simulation with constraint forces: Finite-difference method for particle-laden flows and complex geometries. *Physical Review E* 2000; **61**:7146-7160.

124. Glowinski R, Pan TW, Hesla TI, Joseph DD, Periaux J. A distributed Lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: Application to particulate flow. *Computer Methods in Applied Mechanics and Engineering* 2000; **184**:241-267.

125. Wan DC, Turek S. Fictitious boundary and moving mesh methods for the numerical simulation of rigid particulate flows. *Journal of Computational Physics* 2007; **222**:28-56.

126. Brown RW, Neill GH, Loper RG. Factors influencing optimum ball sealer performance. *Transactions of the Society of Petroleum Engineers of AIME* 1963; **228**:450-454.

127. ten Cate A, Nieuwstad CH, Derksen JJ, Van den Akker HEA. Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity. *Physics of Fluids* 2002; **14**:4012-4025.

128. Lee TR, Chang YS, Choi JB, Kim YJ. Numerical simulation of cylinder oscillation by using a direct forcing technique. *Nuclear Engineering and Design* 2010; **240**:1941-1948.

129. Pettigrew MJ, Carlucci LN, Taylor CE, Fisher NJ. Flow-induced vibration and related technologies in nuclear-components. *Nuclear Engineering and Design* 1991; **131**:81-100.

130. Ahn HT, Kallinderis Y. Strongly coupled flow/structure interactions with a geometrically conservative ALE scheme on general hybrid meshes. *Journal of Computational Physics* 2006; **219**:671-696.

131. Borazjani I, Ge L, Sotiropoulos F. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies. *Journal of Computational Physics* 2008; **227**:7587-7620.

132. Jeong HK, Yoon HS, Ha MY, Tsutahara M. An immersed boundary-thermal lattice Boltzmann method using an equilibrium internal energy density approach for the simulation of flows with heat transfer. *Journal of Computational Physics* 2010; **229**:2526-2543.

133. Moussaoui MA, Jami M, Mezrhab A, Naji H. MRT-lattice Boltzmann simulation of forced convection in a plane channel with an inclined square cylinder. *International Journal of Thermal Sciences* 2010; **49**:131-142.

134. Hindmarsh AC, Gresho PM, Griffiths DF. The stability of explicit Euler time-integration for certain finite-difference approximations of the multi-dimensional advection diffusion equation. *International Journal for Numerical Methods in Fluids* 1984; **4**:853-897.

135. Davis GD. Natural-convection of air in a square cavity - A bench-mark numerical-solution. *International Journal for Numerical Methods in Fluids* 1983; **3**:249-264.

136. Demirdzic I, Lilek Z, Peric M. Fluid-flow and heat-transfer test problems for nonorthogonal grids - Bench-mark solutions. *International Journal for Numerical Methods in Fluids* 1992; **15**:329-354.

137. Yu ZS, Shao XM, Wachs A. A fictitious domain method for particulate flows with heat transfer. *Journal of Computational Physics* 2006; **217**:424-452.

138. Gan H, Chang JZ, Feng JJ, Hu HH. Direct numerical simulation of the sedimentation of solid particles with thermal convection. *Journal of Fluid Mechanics* 2003; **481**:385-411.

139. Yu HD, Girimaji SS. Near-field turbulent simulations of rectangular jets using lattice Boltzmann method. *Physics of Fluids* 2005; **17**:125106.

140. Yu HD, Girimaji SS, Luo LS. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *Journal of Computational Physics* 2005; **209**:599-616.

141. Pattison MJ, Premnath KN, Banerjee S. Computation of turbulent flow and secondary motions in a square duct using a forced generalized lattice Boltzmann equation. *Physical Review E* 2009; **79**:026704.

142. Premnath KN, Pattison MJ, Banerjee S. Dynamic subgrid scale modeling of turbulent flows using lattice-Boltzmann method. *Physica A-Statistical Mechanics and Its Applications* 2009; **388**:2640-2658.

143. Shin BS, Chang SH. CHF experiment and CFD analysis in a 2× 3 rod bundle with mixing vane. *Nuclear Engineering and Design* 2009; **239**:899-912.

144. Lee CM, Choi YD. Comparison of thermo-hydraulic performances of large scale vortex flow (LSVF) and small scale vortex flow (SSVF) mixing vanes in 17×17 nuclear rod bundle. *Nuclear Engineering and Design* 2007; **237**:2322-2331.

145. Ikeno T, Kajishima T, Murata T. The effect of mixing-vane arrangements in a subchannel turbulent flow. *Journal of Nuclear Science and Technology* 2006; **43**:1194-1205.

146. Pettigrew MJ, Taylor CE, Fisher NJ, Yetisir M, Smith BAW. Flow-induced vibration: Recent findings and open questions. *Nuclear Engineering and Design* 1998; **185**:249-276.

147. Ingersoll DT, Forsberg CW, MacDonald PE. Trade studies on the liquid-salt-cooled very high-temperature reactor: Fiscal year 2006 progress report. *ORNL/TM-2006/140*, Oak Ridge National Laboratory, Tennessee, 2007.

148. Kuczaj AK, Komen EMJ, Loginov MS. Large-eddy simulation study of turbulent mixing in a T-junction. *Nuclear Engineering and Design* 2010; **240**:2116-2122.

149. Kuhn S, Braillard O, Niceno B, Prasser HM. Computational study of conjugate heat transfer in T-junctions. *Nuclear Engineering and Design* 2010; **240**:1548-1557.

150. Zhu WY, Lu T, Jiang PX, Guo ZJ, Wang KS. Large eddy simulation of hot and cold fluids mixing in a T-junction for predicting thermal fluctuations. *Applied Mathematics and Mechanics-English Edition* 2009; **30**:1379-1392.

151. Koshizuka S, Oka Y. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering* 1996; **123**:421-434.

152. Nasr-El-Din HA, Al-Mohammad AM, Al-Aamri AD, Al-Fuwaires O. Reaction of gelled acids with calcite. *SPE Production & Operations* 2008; **23**:353-361.

APPENDIX A

DERIVATION OF EXPLICIT-TYPE FROM IMPLICIT-TYPE SPLIT-FORCING SRT-LBEs

Here, we derive the SRT-LBE with an explicit discrete forcing term (Equation (43) with Equations (48) and (49)) [42] from the SRT-LBE with an implicit discrete forcing term SRT-LBE (Equation (53) with Equations (16) and (44b)) [79].

From Equation (53), the implicit-type split-forcing LBE is

$$\begin{aligned} &f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) - f_\alpha(\mathbf{x},t) \\ &= -\frac{1}{\tau}[f_\alpha(\mathbf{x},t) - f_\alpha^{(eq)}(\rho,\mathbf{u})] + \frac{1}{2}\Delta t[F_\alpha(x,t) + F_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t)] \end{aligned}. \tag{A.1}$$

This equation is time-implicit due to the last discrete forcing term. The Equation (A.1) can be changed into time-explicit form by introducing the following substitution:

$$\bar{f}_\alpha(\mathbf{x},t) = f_\alpha(\mathbf{x},t) - \frac{1}{2}\Delta t F_\alpha(\mathbf{x},t). \tag{A.2}$$

Then, Equation (A.1) changes to:

$$\begin{aligned} &\bar{f}_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) - \bar{f}_\alpha(\mathbf{x},t) \\ &= -\frac{1}{\tau}[\bar{f}_\alpha(\mathbf{x},t) - f_\alpha^{(eq)}(\rho(\mathbf{x},t),\mathbf{u}(x,t))] + \left(1-\frac{1}{2\tau}\right)\Delta t F_\alpha(\mathbf{x},t) \end{aligned}. \tag{A.3}$$

It should be noted that in the equilibrium density function of Equation (A.3), the velocity is defined as:

$$\rho\mathbf{u}(\mathbf{x},t) = \sum_\alpha \mathbf{e}_\alpha \bar{f}_\alpha(\mathbf{x},t) + \frac{1}{2}\Delta t\,\mathbf{F}(\mathbf{x},t) \tag{A.4}$$

because taking 1st hydrodynamic moments of Equation (A.2), i.e., summing Equation

(A.2)$\cdot \mathbf{e}_\alpha$ over α, we can get

$$\sum_\alpha \mathbf{e}_\alpha \bar{f}_\alpha(\mathbf{x},t) = \sum_\alpha \mathbf{e}_\alpha f_\alpha(\mathbf{x},t) - \frac{1}{2}\Delta t\,\mathbf{F}(x,t) = \rho\mathbf{u}(\mathbf{x},t) - \frac{1}{2}\Delta t\,\mathbf{F}(x,t)\,. \tag{A.5}$$

Also, defining

$$\bar{F}_\alpha(\mathbf{x},t) = \left(1-\frac{1}{2\tau}\right)F_\alpha(\mathbf{x},t) = \left(1-\frac{1}{2\tau}\right)w_\alpha\left[3\frac{\mathbf{e}_\alpha-\mathbf{u}(\mathbf{x},t)}{c^2}+9\frac{\mathbf{e}_\alpha\cdot\mathbf{u}(\mathbf{x},t)}{c^4}\mathbf{e}_\alpha\right]\cdot\mathbf{F}(\mathbf{x},t)\,, \tag{A.6}$$

Equation (A.3) becomes

$$\bar{f}_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) - \bar{f}_\alpha(\mathbf{x},t) = -\frac{1}{\tau}[\bar{f}_\alpha(\mathbf{x},t) - f_\alpha^{(eq)}(\rho(\mathbf{x},t),\mathbf{u}(x,t))] + \Delta t\bar{F}_\alpha(\mathbf{x},t) \tag{A.7}$$

Dropping the overbars in Equations (A.4), (A.6), and (A.7), we can obtain Guo et al.'s LBE:

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f_\alpha(\mathbf{x},t) - \frac{1}{\tau}[f_\alpha(\mathbf{x},t) - f_\alpha^{(eq)}(\mathbf{x},t)] + F_\alpha(\mathbf{x},t)\Delta t \tag{43}$$

with

$$\rho\mathbf{u}(\mathbf{x},t) = \sum_\alpha \mathbf{e}_\alpha f_\alpha(\mathbf{x},t) + \frac{1}{2}\Delta t\,\mathbf{F}(\mathbf{x},t) \tag{48}$$

$$F_\alpha(\mathbf{x},t) = \left(1-\frac{1}{2\tau}\right)w_\alpha\left[3\frac{\mathbf{e}_\alpha-\mathbf{u}(\mathbf{x},t)}{c^2}+9\frac{\mathbf{e}_\alpha\cdot\mathbf{u}(\mathbf{x},t)}{c^4}\mathbf{e}_\alpha\right]\cdot\mathbf{F}(\mathbf{x},t)\,. \tag{49}$$

Thus, it is confirmed that explicit and implicit split-forcing SRT-LBEs are equivalent. It can also be shown in the same manner that the explicit and implicit split-forcing MRT-LBEs are equivalent.

APPENDIX B

DERIVATION OF THE DIRECT-FORCING FORMULA FROM THE LUMPED-FORCING SRT-LBE

From Equation (43), on the node **x** at time t+Δt, forced particle distribution functions can be written as

$$\begin{aligned} & f_\alpha(\mathbf{x}, t+\Delta t) \\ &= f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) - \frac{1}{\tau}[f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) - f_\alpha^{(eq)}(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t)] + \Delta t\, F_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) \end{aligned} . \tag{B.1}$$

The particle distribution functions under no external force can be written as

$$f_\alpha^{noF}(\mathbf{x}, t+\Delta t) = f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) - \frac{1}{\tau}[f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) - f_\alpha^{(eq)}(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t)] . \tag{B.2}$$

Subtracting Equation (B.2) from Equation (B.1), we obtain

$$f_\alpha(\mathbf{x}, t+\Delta t) - f_\alpha^{noF}(\mathbf{x}, t+\Delta t) = \Delta t\, F_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) . \tag{B.3}$$

Summing Equation (B.3)$\cdot\mathbf{e}_\alpha$ and using Equation (16), we can obtain

$$\rho(\mathbf{x}, t+\Delta t)\mathbf{u}(\mathbf{x}, t+\Delta t) - \rho(\mathbf{x}, t+\Delta t)\mathbf{u}^{noF}(\mathbf{x}, t+\Delta t) = \Delta t\sum_\alpha \mathbf{e}_\alpha F_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) . \tag{B.4}$$

where we used the definition of

$$\rho(\mathbf{x}, t+\Delta t)\mathbf{u}^{noF}(\mathbf{x}, t+\Delta t) \equiv \sum_\alpha \mathbf{e}_\alpha f_\alpha^{noF}(\mathbf{x}, t+\Delta t) . \tag{B.5}$$

We can rewrite Equation (B.4) as

$$\sum_\alpha \mathbf{e}_\alpha F_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) = \rho(\mathbf{x}, t+\Delta t)\frac{\mathbf{u}(\mathbf{x}, t+\Delta t) - \mathbf{u}^{noF}(\mathbf{x}, t+\Delta t)}{\Delta t} \tag{B.6}$$

If, as in the direct-forcing IBM based on the NSE, the desired velocity which satisfies

the no-slip condition on the boundary is given, that is, $\mathbf{U}^d = \mathbf{u}(\mathbf{x}, t+\Delta t)$, then,

$$\sum_{\alpha} \mathbf{e}_{\alpha} F_{\alpha}(\mathbf{x} - \mathbf{e}_{\alpha}\Delta t, t) = \rho \frac{\mathbf{U}^d - \mathbf{u}^{noF}(\mathbf{x}, t+\Delta t)}{\Delta t}. \tag{63}$$

APPENDIX C

DERIVATION OF THE DIRECT-FORCING FORMULA

FROM THE SPLIT-FORCING SRT-LBE

From implicit split-forcing SRT-LBE (53), the particle distribution function at (**x**,t+Δt) is determined by

$$\begin{aligned} f_\alpha(\mathbf{x}, t+\Delta t) = & f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) - \frac{1}{\tau}[f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) - f_\alpha^{(eq)}(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t)] \\ & + \frac{\Delta t}{2}[F_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) + F_\alpha(\mathbf{x}, t+\Delta t)] \end{aligned} \tag{C.1}$$

The unforced particle distribution function under no external force at (**x**,t+Δt) can be written as

$$\begin{aligned} f_\alpha^{noF}(\mathbf{x}, t+\Delta t) = & f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) - \frac{1}{\tau}[f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) - f_\alpha^{(eq)}(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t)] \\ & + \frac{\Delta t}{2}F_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t) \end{aligned} \tag{C.2}$$

Subtracting Equation (C.2) from Equation (C.1), we obtain

$$f_\alpha(\mathbf{x}, t+\Delta t) - f_\alpha^{noF}(\mathbf{x}, t+\Delta t) = \frac{\Delta t}{2}F_\alpha(\mathbf{x}, t+\Delta t). \tag{C.3}$$

Summing Equation (C.3)$\cdot\mathbf{e}_\alpha$ and using Equation (46), we can obtain

$$\begin{aligned} \rho(\mathbf{x}, t+\Delta t)\mathbf{u}(\mathbf{x}, t+\Delta t) - \rho(\mathbf{x}, t+\Delta t)\mathbf{u}^{noF}(\mathbf{x}, t+\Delta t) &= \frac{\Delta t}{2}\sum_\alpha \mathbf{e}_\alpha F_\alpha(\mathbf{x}, t+\Delta t) \\ &= \frac{\Delta t}{2}\mathbf{F}(\mathbf{x}, t+\Delta t) \end{aligned}. \tag{C.4}$$

where we used the definition of

$$\rho(\mathbf{x},t+\Delta t)\mathbf{u}^{noF}(\mathbf{x},t+\Delta t)\equiv\sum_{\alpha}\mathbf{e}_{\alpha}f_{\alpha}^{noF}(\mathbf{x},t+\Delta t)\,. \tag{C.5}$$

We can rewrite Equation (C.4) as

$$\mathbf{F}(\mathbf{x},t+\Delta t)=2\rho(\mathbf{x},t+\Delta t)\frac{\mathbf{u}(\mathbf{x},t+\Delta t)-\mathbf{u}^{noF}(\mathbf{x},t+\Delta t)}{\Delta t} \tag{C.6}$$

If the desired velocity which satisfies the no-slip condition on the boundary is given, that is, $\mathbf{U}^{d}=\mathbf{u}(\mathbf{x},t+\Delta t)$, then,

$$\mathbf{F}(\mathbf{x},t+\Delta t)=2\rho(\mathbf{x},t+\Delta t)\frac{\mathbf{U}^{d}-\mathbf{u}^{noF}(\mathbf{x},t+\Delta t)}{\Delta t} \tag{65}$$

We can start from the explicit split-forcing SRT-LBE (43). The unforced particle distribution function under no external force at (**x**,t+Δt) can be written as

$$\rho(\mathbf{x},t+\Delta t)\mathbf{u}^{noF}(\mathbf{x},t+\Delta t)=\sum_{\alpha}\mathbf{e}_{\alpha}f_{\alpha}(\mathbf{x},t+\Delta t)\,. \tag{C.8}$$

If the desired velocity at the next time step, $\mathbf{U}^{d}$, which satisfies the no-slip condition on the boundary, is given, then from Equation (48)

$$\rho(\mathbf{x},t+\Delta t)\mathbf{U}^{d}=\sum_{\alpha}\mathbf{e}_{\alpha}f_{\alpha}(\mathbf{x},t+\Delta t)+\frac{\Delta t}{2}\mathbf{F}(\mathbf{x},t+\Delta t) \tag{C.9}$$

Subtracting Equation (C.8) from Equation (C.9), we obtain

$$\mathbf{F}(\mathbf{x},t+\Delta t)=2\rho(\mathbf{x},t+\Delta t)\frac{\mathbf{U}^{d}-\mathbf{u}^{noF}(\mathbf{x},t+\Delta t)}{\Delta t}\,. \tag{65}$$

VITA

Shin Kyu Kang received his B.S. and M.S. in nuclear engineering from Seoul National University in 1999 and 2004, respectively. He began his Ph.D. study in nuclear engineering at Texas A&M University in the fall of 2005. His research interests include thermal hydraulics in nuclear engineering and computational fluid dynamics (CFD). His permanent mailing address is in the care of Dr. Yassin A. Hassan at Texas A&M University, Department of Nuclear Engineering, 3133 TAMU, College Station, TX 77843-3133. His e-mail address is paengki1@tamu.edu.