

IMMERSED BOUNDARY (IB) -LATTICE BOLTZMANN METHOD (LBM) COUPLED  
WITH ADAPTIVE MESH REFINEMENT (AMR) TECHNIQUES  
FOR SIMULATION OF INCOMPRESSIBLE VISCOUS FLOW

by

Xixiong Guo

Bachelor of Engineering, Northwestern Polytechnical University, Xi'an, China, 2011

A Thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Mechanical and Industrial Engineering

Toronto, Ontario, Canada, 2014

©Xixiong Guo, 2014

## Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Immersed Boundary (IB) -Lattice Boltzmann Method (LBM)  
Coupled with Adaptive Mesh Refinement (AMR) Techniques  
for Simulation of Incompressible Viscous Flow

Master of Applied Science, 2014

Xixiong Guo  
Department of Mechanical and Industrial Engineering  
Ryerson University

## Abstract

This study is aimed at developing a novel computational framework that seamlessly incorporates the feedback forcing model and adaptive mesh refinement (AMR) techniques in the immersed-boundary (IB) lattice Boltzmann method (LBM) approach, so that challenging problems, including the interactions between flowing fluids and moving objects, can be numerically investigated. Owing to the feedback forcing based IB model, the advantages, such as simple mechanics principle, explicit interpolations, and inherent satisfaction of no-slip boundary condition for solid surfaces, are fully exhibited. Additionally, the "bubble" function is employed in the local mesh refinement process, so that the solution of second- order accuracy at newly generated nodes can be obtained only by the spatial interpolation but no temporal interpolation. Focusing on both steady and unsteady flows around a single cylinder and bi-cylinders, a number of test cases performed in this study have demonstrated the usefulness and effectiveness of the present AMR IB-LBM approach.

## Acknowledgements

I am very grateful to Dr. Jun Cao for giving me the opportunity to work on this exciting project and for his constant encouragement and valuable guidance during this study.

This work is financially supported by Dr. Cao's Discovery Grant awarded by the National Science and Engineering Research Council (NSERC) of Canada.

# Table of Contents

Author’s declaration .....	ii
Abstract.....	iii
Acknowledgements .....	iv
List of Tables.....	viii
List of Figures .....	ix
Nomenclature.....	xii
Greek Symbols.....	xv
Subscripts/Superscripts.....	xvi
Chapter 1.Introduction.....	1
1.1.    Background .....	1
1.1.1.    On lattice-Boltzmann method (LBM) .....	2
1.1.2.    On immersed boundary (IB) method.....	5
1.1.3.    On local mesh refinement method.....	8
1.2.    Thesis objective and significance.....	10
1.3.    Thesis organization .....	13

Chapter 2. Two-Dimensional Lattice-Boltzmann Model .....	14
2.1. Two-Dimensional Nine-Velocity (D2Q9) Model .....	14
2.2. Single-Relaxation-Time Lattice-Boltzmann Equation .....	15
2.3. Multiple-Relaxation-Time LBE .....	17
2.4. Boundary Condition .....	20
2.4.1. Straight boundary .....	20
2.4.2. Curved boundary .....	23
Chapter 3. Mesh Refinement Methods .....	29
3.1. Multi-level local mesh refinement method .....	29
3.2. Bubble function for local mesh refinement used in lattice-Boltzmann method .....	33
3.3. Adaptive mesh refinement (AMR) technique .....	39
Chapter 4. Numerical Experiments .....	42
4.1. Introduction .....	42
4.2. Flow over a single cylinder at $Re=40$ .....	46
4.2.1. Steady flow over a stationary cylinder .....	46
4.2.2. Steady flow over a moving cylinder .....	51
4.3. Flow over a single cylinder at $Re=100$ .....	53
4.3.1. Unsteady flow over a stationary cylinder .....	53
4.3.2. Unsteady flow over a moving cylinder .....	56

4.4.	Flow over two stationary vertically aligned and gapped cylinders.....	58
4.4.1.	Re=40, $g^* = 1$ .....	59
4.4.2.	Re=70, $g^* = 0.5$ .....	61
4.4.3.	Re=100, $g^* = 0.2, 0.7, 1.5, 3$ .....	63
4.5.	Flow over two moving cylinders .....	69
4.6.	Local mesh refinement sensitivity .....	73
4.6.1.	Stationary cylinder (Re=40) with different horizontal refinement zones .....	73
4.6.2.	Stationary cylinder (Re=40) with adaptive mesh refinement.....	78
4.6.3.	Moving cylinder (Re=100) with different vertical refinement zones .....	81
4.6.4.	Moving cylinder (Re=100) with adaptive mesh refinement.....	83
4.6.5.	Computation cost and accuracy comparison between different meshing techniques ....	85
Chapter 5. Concluding and Outlooking Remarks.....		89
5.1.	Conclusions.....	89
5.2.	Recommendations.....	90
References.....		91

## List of Tables

Table 4.1 Comparison of drag coefficient and length of recirculation zone at $Re=40$ .....	50
Table 4.2 Comparisons of time-averaged drag coefficient and Strouhal number at $Re=100$ .....	56
Table 4.3 Six cases using different zones of finest grid resolution for $Re=40$ .....	75
Table 4.4 Comparison of drag coefficient and recirculation length for $Re=40$ .....	75
Table 4.5 Comparison of drag coefficients and length of recirculation zone at $Re=40$ .....	80
Table 4.6 Comparison of drag coefficient for $Re=100$ .....	83
Table 4.7 Comparison of drag coefficients and Strouhal number for $Re=100$ .....	85
Table 4.8 Statistics for $Re=40$ .....	87
Table 4.9 Statistics for $Re=100$ .....	88

## List of Figures

Figure 2.1 D2Q9 model on a square lattice .....	15
Figure 2.2 Bounce-back scheme.....	21
Figure 2.3 Non-equilibrium extrapolation scheme.....	23
Figure 2.4 Schematic of immersed boundary method.....	24
Figure 3.1 Structure of two-level mesh of different lattice spacings.....	30
Figure 3.2 Schematic of an overlay region.....	34
Figure 3.3 Numbering for neighboring nodes .....	36
Figure 3.4 Numbering example for (a) neighboring cells and (b) and neighboring nodes .....	40
Figure 3.5 Schematic for bubble function interpolation used in AMR .....	41
Figure 4.1 Computational domain, setup of zones for multi-level local mesh refinements, and cylinder location .....	45
Figure 4.2 Close-up view of the location of the cylinder and the mesh refinement transition ...	45
Figure 4.3 Streamline pattern around cylinder at $Re=40$ .....	47
Figure 4.4 Pressure around cylinder at $Re=40$ .....	47
Figure 4.5 Velocity magnitude around cylinder (left) and its close-up near the stagnation point (right) at $Re=40$ .....	48
Figure 4.6 Vorticity around cylinder at $Re=40$ .....	48
Figure 4.7 Pressure distribution on surface of cylinder at $Re=40$ .....	50

Figure 4.8 Streamline pattern around a moving cylinder at $Re=40$ .....	52
Figure 4.9 Pressure distribution on surface of a moving cylinder at $Re=40$ .....	52
Figure 4.10 Streamline pattern around cylinder at $Re=100$ .....	54
Figure 4.11 Velocity magnitude around cylinder at $Re=100$ .....	54
Figure 4.12 Vorticity around cylinder at $Re=100$ .....	55
Figure 4.13 Evolution of drag and lift coefficients at $Re = 100$ .....	55
Figure 4.14 Two instantaneous distributions of vorticity around a moving cylinder at $Re=100$	57
Figure 4.15 Evolution of drag and lift coefficients at $Re = 100$ in the case of moving cylinder	57
Figure 4.16 Schematic of two vertically aligned cylinders .....	59
Figure 4.17 Instantaneous streamline and vorticity contours at $Re=40, g^* = 1$ .....	60
Figure 4.18 Evolution of drag and lift coefficients for two cylinders at $Re=40, g^* = 1$ .....	61
Figure 4.19 Instantaneous streamline and vorticity contours at $Re=70, g^* = 0.5$ .....	62
Figure 4.20 Evolution of drag and lift coefficients for two cylinders at $Re=70, g^* = 0.5$ .....	63
Figure 4.21 Instantaneous streamline and vorticity contours at $Re=100, g^* = 0.2$ .....	65
Figure 4.22 Evolution of drag and lift coefficients for two cylinders at $Re=100, g^* = 0.2$ .....	66
Figure 4.23 Instantaneous streamline and vorticity contours at $Re=100, g^* = 0.7$ .....	66
Figure 4.24 Evolution of drag and lift coefficients for two cylinders at $Re=100, g^* = 0.7$ .....	67
Figure 4.25 Instantaneous streamline and vorticity contours at $Re=100, g^* = 1.5$ .....	67
Figure 4.26 Evolution of drag and lift coefficients for two cylinders at $Re=100, g^* = 1.5$ .....	68
Figure 4.27 Instantaneous streamline and vorticity contours at $Re=100, g^* = 3$ .....	68

Figure 4.28 Evolution of drag and lift coefficients for two cylinders at $Re=100, g^* = 3$ .....	69
Figure 4.29 Computational domain for two cylinders moving with towards each other.....	71
Figure 4.30 Vorticity contour when two cylinders are the closest.....	71
Figure 4.31 Vorticity contour when two cylinders are distanced at 60 .....	71
Figure 4.32 Evolution of drag coefficient for the upper cylinder .....	72
Figure 4.33 Evolution of lift coefficient for the upper cylinder .....	72
Figure 4.34 Streamline patterns around cylinder at $Re=40$ in six cases.....	76
Figure 4.35 Pressure contours at $Re=40$ in six cases.....	77
Figure 4.36 AMR mesh at $Re=40$ .....	79
Figure 4.37 AMR results of streamlines and pressure distribution .....	79
Figure 4.38 Comparison of pressure coefficient .....	80
Figure 4.39 Zone of finest grid resolution in the four cases of a moving cylinder for $Re=100$ ..	81
Figure 4.40 Vorticity contours at $Re=100$ in four cases .....	82
Figure 4.41 Final adaptive mesh for $Re=100$ .....	84
Figure 4.42 Vorticity contours on final adaptive mesh for $Re = 100$ .....	85

## Nomenclature

$a$	Free coefficient used in the “bubble” function
$b$	Free coefficient used in the “bubble” function
$C_d$	Drag coefficient
$C_l$	Lift coefficient
$C_p$	Pressure coefficient
$c$	Lattice speed
$D$	Diameter of cylinder
$\vec{e}$	Particle velocities
$e$	Energy
$e_1$	Particle velocity component in the $x$ direction used in the “bubble” function
$e_2$	Particle velocity component in the $y$ direction used in the “bubble” function
$\vec{F}$	Boundary force density
$F_D$	Drag component of the aerodynamic force
$F_L$	Lift force of the aerodynamic force
$\vec{f}$	Force density acting on the fluid
$f$	Particle distribution function
$f_q$	Vortex shedding frequency
$f$	Space of distribution functions used in MRT

$g$	Gap width between two cylinders
$j$	Mass flux used in MRT
$k$	Spring coefficient
$m$	Moment space of distribution functions used in MRT
$\mathbf{M}$	Transformation matrix used in MRT
$n$	Boolean variables
$p$	Pressure
$P$	Stress tensor used in MRT
$q$	Energy flux
Re	Reynolds number
$\mathbf{S}$	Relaxation matrix
$s$	Lagrangian parametric coordinates
$s$	Relaxation rate
St	Strouhal number
$t$	Instantaneous time
$U_\infty$	Free stream velocity
$\vec{u}$	Fluid velocity
$\vec{V}$	Macroscopic velocity used in conventional IBM
$w$	Weighting factor

$\vec{X}$	Position vector expressed using the Lagrangian coordinate system
$x$	Eulerian $x$ -coordinate
$y$	Eulerian $y$ -coordinate

## Greek Symbols

$\alpha$	Negative free constant used in feedback forcing method
$\beta$	Negative free constant used in feedback forcing method
$\nu$	Kinematic viscosity
$\Gamma$	Boundary curve
$\Omega$	Flow field
$\tau$	Relaxation time
$\delta$	Dirac delta function
$\rho$	Density
$\varepsilon$	Criterion in AMR method
$\zeta$	Bulk viscosity
$\vec{\xi}$	Particle velocity
$\Delta\vec{\xi}$	Displacement used in Hook's Law
$\Delta x$	Grid space in the $x$ -direction
$\Delta y$	Grid space in the $y$ -direction
$\delta x$	Lattice size
$\delta t$	Time step
$\pi$	Moment

## Subscripts/Superscripts

$()^+$	Post-collision state
$()_a$	Number of directions of velocities at each node
$()^{neq}$	Non-equilibrium state
$()_{ij}$	Point with coordinate $(i, j)$
$()_i^j$	$j^{\text{th}}$ time step for a Lagrangian point $i$
$()_{boundary}$	Boundary point
$()_{fluid}$	Fluid point
$()_{\infty}$	Far field
$()_c$	Coarse mesh
$()_f$	Fine mesh
$()_s$	Sound speed
$()_x$	$x$ -direction
$()_{.xx}$	Diagonal component used for MRT parameters
$()_{.xy}$	Off-diagonal component used for MRT parameters
$()_y$	$y$ -direction
$()^{eq}$	Equilibrium state
$()_{object}$	Object

# Chapter 1. Introduction

## 1.1. Background

As an important branch of the physics subject, fluid dynamics studies the fluid flow behaviors that take place in many aspects of the real world. Although tremendous efforts on the theoretical, computational, and experimental fronts of this subject have been made for a long history towards more and better understanding of fluid dynamics, great challenges still remain in this research area. In the study of computational fluid dynamics (CFD), despite fast development of digital computer technologies, numerical solutions with high accuracy are difficult to achieve for many engineering problems, which particularly requires the conventional numerical methods to be enhanced by novel computational strategies.

From the viewpoint of fluid motion description, existing CFD methods can be divided into three categories: macroscopic method, microscopic method, and mesoscopic method. The conventional methods, such as finite-difference method (FDM), finite-volume method (FVM), finite-element method (FEM), and spectral method, belong to the macroscopic category as they all assume the fluid as a continuum. The motion of the fluid satisfies the momentum and energy conservations, and is governed by Euler equations or Navier-Stokes equations. When using a macroscopic method to deal with fluid dynamics problems, the governing equations are discretized into algebraic equations, which are to be numerically solved to obtain discrete solutions on the

computational domain.

Different from the macroscopic methods, the microscopic and mesoscopic methods no longer assume the fluid as continuum. In the microscopic method, fluid is considered as consisting of a large number of discrete molecules, and their thermal property and movement in the system manifest as the irregular heat motion of molecules. Thus, the microscopic method relies on the simulation of movement for every molecule, resulting in the macroscopic rules based on statistic average method. Due to numerous molecules in fluids, computation and data storage appear inevitably huge, which becomes unaffordable by the present computational devices despite existing tremendous improvement in information technology.

Fluid particles in the mesoscopic method are viewed larger than the molecules, but much smaller than the macroscopic scale. Based on the fact that the characteristics of the flowing fluid cannot be affected by a single particle, an evolution mechanism is proposed in the mesoscopic approach that fully satisfies the physical laws, and the quantities describing the fluid flow can be calculated under the guidance of the pre-set mechanism and, thus, inherently conform to the physical laws. Obviously, the mesoscopic method inherits the nuggets of both macroscopic and microscopic methods, such as using less assumptions and neglecting the movement details of each single particle.

### **1.1.1. On lattice-Boltzmann method (LBM)**

As examples, the lattice gas automata (LGA) [1], direct simulation Monte Carlo (DSMC)

method [2-3], and lattice-Boltzmann method (LBM) are routinely employed mesoscopic methods. As aforementioned, unlike conventional numerical schemes through discretization of macroscopic continuum equations, the mesoscopic approach views a group of fluid molecules as a fictitious "fluid particle" and restrains such fluid particles into the framework of simplified mesoscopic kinetic models, which constitutes an evolution mechanism for each fluid particle in a fluid flow field. Consequently, the averaged properties, which are the macroscopic quantities used for describing the fluid flow, can still inherently satisfy the desired macroscopic equations.

LBM was originated from LGA. The LGA approach provides a simplified micro-world in which space, time, and particle velocities are all discrete. A set of Boolean variables  $n_i(\vec{x}, t) (i = 1, \dots, b)$  describing the particle evolution is defined, where  $b$  is the number of possible velocity directions of the particle at each node. However, the LGA model contains some shortcomings such as non-Galilean invariance, inherent statistical noise, and exponential complexity.

McNamara and Zanetti [4] proposed the earliest model of LBM with the explicit intention of circumventing the statistical noise problem arising in LGA. They merely replaced the Boolean occupation numbers  $n_i$  with the corresponding lump-averaged local particle distribution function, and replaced the LGA evolution equation with the lattice-Boltzmann equation. In their model, the form of collision operator was still considered to be exponential complexity. Then, Higuera and Jimenez derived a simplified model [5] to linearize the collision operator by introducing the equilibrium distribution function and collision matrix. The merit of this model includes the

prevention of the statistical noise and the reduction of the complexity of the collision operator; however, other drawbacks in LGA remains uncorrected since Fermi-Dirac equilibrium distribution function was still employed in this simplified model.

In order to further improve LBM, a remarkable innovation was made by Chen et al [6] and Qian et al [7] that replaced the Fermi-Dirac distribution function by the Maxwell-Boltzmann distribution function, making the computation significantly efficient. This model is now usually referred to as the Single Relaxation Time (SRT) or Lattice Bhatnagar-Gross-Krook (LBGK) model. Another key point of SRT is the replacement of the collision matrix in LBM by a parameter named relaxation time, enabling more transport coefficients to play roles in the model. Although SRT has completely overcome the drawbacks incurred by LGA, other issues associated with SRT remain to be resolved. For instance, this model may lead to numerical instability when the relaxation time is close to 0.5. To overcome this shortcoming, d'Humeriers revised SRT to an enhanced version that allows for a number of adjustable parameters and, thus, is often referred to as Multiple Relaxation Time (MRT) model [8].

In practice, when using LBM to study the dynamics of fictitious fluid particles, the density distribution function is sought. The macroscopic variables, such as the pressure, velocity, and momentum, are calculated through their distribution functions. Compared to those conventional numerical methods, LBM is relatively new, but noticeably advantageous in terms of simplicity for implementation, parallelizability for algorithmic development, and robustness for applications. With these advantages, LBM has been successfully applied in a variety of complex fluid systems, such as

hydrodynamics [9], flow through porous media [10], multiphase fluid flow [11][12], simulation of fluid turbulence [13], particle-fluid suspension flows [14], nano- and micro- scale fluid flow [15], heat transfer problems [16-17], etc. LBM simulations have covered the isothermal incompressible flow, inviscid compressible flow, as well as viscous compressible flow with some representative examples found in [18]. In the LBM family, distinguished from SRT, the MRT model exhibits a remarkably improved numerical stability while maintaining the simplicity and computational efficiency of SRT. Recent MRT applications [19] have demonstrated this approach is particularly pertinent in simulating flows at elevated Reynolds numbers or with solution multiplicity.

Nowadays, LBM is still under further development since more enhanced capabilities of this mesoscopic approach are expected for the purpose of simulation of more challenging fluid flow scenarios.

### **1.1.2. On immersed boundary (IB) method**

Relevant to a large number of industrial and daily life applications, such as blood flow through heart valves, insect hovering, and fish locomotion, the interaction between fluid and moving objects has become an attractive research subject. In computer simulations of these interaction problems, numerical approaches are classified into two major categories: boundary conforming methods and non-boundary conforming methods.

In the boundary conforming methods, boundary points have to coincide with the grid nodes, so the physical boundary condition can be directly implemented. The Arbitrary Lagrangian Eulerian

(ALE) approach [20, 21] is the most popular boundary conforming method, which has been applied in connection with the finite difference, finite volume, and finite element schemes. Owing to the retained conformity of regenerated mesh to the solid boundary, ALE can usually yield high-order accuracy. However, in the case of moving boundary problems, especially when complex geometry possibly along with its irregular deformation are present, grid reconstruction procedure has to be unceasingly utilized, which is technically challenging and computationally time-consuming. When applying a non-boundary conforming method, the effect of the solid boundary to the fluid is taken into consideration by proper treatment to the variables at the Cartesian grid nodes that are located in the vicinity of the moving boundary, so that the governing equations can still be routinely solved on the fixed Cartesian mesh without the need of mesh regeneration. Compared to the boundary conforming methods, the non-boundary conforming methods remove the tedious mesh regeneration step.

The non-boundary conforming methods can be further classified into Cartesian grid methods and immersed boundary methods. In the Cartesian grid methods [22, 23], a grid cell around the body is considered broken by the cutting of the immersed boundary, causing great inconvenience in numerical treatment if the involved portion of the immersed boundary is an irregular curve.

Unlike the Cartesian grid method, the immersed boundary (IB) method, which was proposed by Peskin [24, 25], introduces an additional "restoring" force in the vicinity of the immersed object in order to account for the effect of the solid boundary, and then this additional force will be distributed to the Cartesian mesh as a special body force. Clearly, the IB approach allows for a fixed

mesh no matter how the immersed body moves or deforms and, thus, can be viewed as the simplest non-boundary conforming method.

The IB method was originally developed for modeling the interaction between incompressible viscous fluid and moving boundaries immersed in the fluid, and then it was also extended to investigate problems arising from solid mechanics problems [26]. Along the IB development path, the formulation of restoring force has always been revised towards its better conformity to the physical nature of original scenario. As forerunner of IB approach, Peskin [24] applied a penalty method to ensure the restoring force to obey Hook's law. More recently, Fadlun et al. [27] proposed a direct forcing method in which the restoring force at points of an immersed boundary was derived directly from the momentum equations.

In 2002, the IB approach was incorporated in LBM the first time by Feng et al [28, 29], and this conjunctive numerical strategy was abbreviated to IB-LBM. Within the past decade, many IB-LBM applications have been made by the CFD community for investigations of fluid flow problems associated with complex geometries and various moving boundaries. For instance, Niu et al [30] proposed a modified IB-LBM to compute boundary force with the aid of momentum exchange mechanism. Although this method appears explicit and offers a straightforward way for computation, flow penetration through moving boundary may arise in the simulation result, which distinctly distorts the physical flow behavior. Further investigation revealed that the restoring force pre-calculated from the momentum exchange mechanism is unable to secure the accuracy of velocity distribution near the boundary. Then, Wu et al. [31] presented an implicit scheme named

velocity-correction method, in which the restoring force is no longer pre-calculated and keeps updating to constantly ensure the satisfaction of non-slip condition. Owing to this improvement, numerical flow penetration through boundary can be eliminated [32]. More recently, Dash et al. [33] derived a flexible forcing scheme for IB-LBM that also features satisfying the non-slip condition for moving solid boundary. However, the velocity-correction and flexible forcing methods are both implicit schemes, which inevitably consume tremendous computational resources and usually lead to a painful simulation process for fluid flow problems with a high degree of sophistication, such as complex and deforming geometries.

### **1.1.3. On local mesh refinement method**

Conventionally, structured Cartesian meshes with uniform square cells have been routinely employed in standard two dimensional LBM applications [18]. Obviously, uniform mesh is mostly an inappropriate option since homogeneous flow or uniformly transited flow are nowadays of very little interest to numerical investigation. On the contrary, intricate flow problems associated with various complexities, such as irregular geometry, multiple phases, evolves with rapidly varying fluid properties in some sensitive zones of a computational domain, demanding non-uniform meshes for more effective and economical computations. This poses challenges to LBM as the traditional version of LBM was formulated on uniform Cartesian mesh only. If persistently employing the traditional LBM in the simulation of complex flows, a higher grid resolution in critical regions is mandatory to ensure the accuracy of LBM solution, and such an elevated node density has to be

maintained uniformly over the entire computational domain despite the pointlessness of finer grid for the zones of little sensitivity; as a result, the computation process using a uniform grid with a higher resolution, which can also be viewed as an aimlessly refined mesh, must be undesirably time-consuming.

To dislodge the reliance of LBM on uniform Cartesian mesh, conjunction techniques were sought to make LBM executable on a non-uniform mesh to LBM. A variety of interpolation schemes have been derived so that, as of today, unstructured mesh [34], multi-block grid [35], along with multigrid [36] and meshless [37] techniques are all incorporated in the LBM framework. On the other hand, local mesh refinement techniques have proven efficient in reducing the computational expense without sacrificing the numerical solution accuracy. In particular, more attractive is the adaptive mesh refinement (AMR) strategy that dynamically optimizes the grid resolution levels over the entire computational domain based on the current numerical solution. The AMR concept was first proposed by Berger et al. [38] for structured grids in 1984 and, then, the AMR techniques grew up fast owing to its great benefits [9]. However, the incorporation of mesh refinement techniques in LBM appeared in publication only about ten years ago and, at its beginning stage, the local mesh refinement in LBM was performed in a non-adaptive fashion [39], [40]. Very recently, a few examples have been found with AMR in LBM. Crouse et al. [41] employed a quadtree-based data structure for AMR to be coupled with LBM. P. Neumann et al. [42] demonstrated an AMR-LBM version based on octree data structure. Other similar examples include the AMR-LBM studies reported by Tolke et al. [11] and Z. Yu [12]. All these cited pieces of

AMR-LBM work used the tree-type data structure to manage a hierarchy of grids at different resolution levels. Another type of AMR-LBM algorithm is found in [43] that coupled AMR with finite-difference lattice Boltzmann method (FDLBM) in order not to rely on the tree-type data structure in the code.

The key difficulty for using non-uniform meshes in LBM is that the LBM solution interpolation in response to the grid non-uniformity resulting from local mesh refinement and un-refinement has to be manipulated with great care to ensure the physical conservation laws. Particularly, the accuracy of interpolation may become worse in the AMR case, since AMR can result in very steep and irregular transition between different levels of grid resolution on an adaptively generated mesh.

## **1.2. Thesis objective and significance**

As aforementioned, the conjunctive IB-LBM approach has been proposed by the LBM researchers, but most existing IB-LBM schemes, such as the velocity-correction method [31, 32], do not have an explicit formulation to straightforwardly account for the effect of the restoring force arising from the IB approach. The weakness of implicit scheme based IB-LBM manifests mainly in excessive computational expenses. In addition, when employing an implicit scheme to eventually compute the velocity affected by the moving boundary, a matrix inversion process is required, and this matrix inversion may turn out to be a singular problem in case strongly irregularly-shaped moving boundary is present.

Fortunately, a novel IB model, which is based upon the feedback forcing and embedded in a finite difference flavored Navier–Stokes solver, was presented by Huang et al. [44] for studying the interaction between the flowing fluid and moving objects immersed in the fluid. This model exhibits an explicit scheme for obtaining the restoring force term directly from the feedback law, requesting no matrix inversion and, thus, tremendously simplifying the entire solution procedure. Additionally, this feedback law based IB scheme essentially precludes, owing to the feedback law, the fluid flow from penetrating the boundary of a moving solid object immersed in the fluid, leading to strict satisfaction of the non-slip boundary condition. Compared other IB-LBM models, this feedback law based IB-LBM framework looks distinctive for numerical investigation of fluid-solid interaction, featuring simple principle, easy implementation, and full parallelizability. Also, in order to render computation more efficient, it looks very promising for embedding local mesh refinement techniques in this new IB-LBM framework, but as of today this incorporation is nowhere to find yet.

In terms of incorporation of mesh refinement algorithm in LBM, relevant progress was reviewed in the previous section. The most challenging issue in this regard is the complexity of solution interpolation at new nodes generated by local mesh refinement when a LBM mesh manifests strong non-uniformity in grid resolution. Unlike the conventional numerical methods such as the finite element, finite difference, and finite volume approaches, LBM is not a macroscopic method, and it requires, for each new grid node, the interpolation scheme to bear higher accuracy so that the LBM solution, which eventually appears in an averaged form, can keep the accuracy to a satisfactory degree. Also, the physical conservation laws should be constantly taken into

consideration whenever the LBM solution interpolation takes place. To ensure the LBM solution accuracy in the course of local mesh refinements, many existing interpolation formulations appear in the implicit manner. For example, the multi-block version of LBM [35] uses both the time and spatial interpolations in the interface between the coarse and fine grid zones, and the interpolations have to employ the solutions corresponding to more than one time step, which turns out to be an implicit interpolation method. The implicit interpolation requests extra data storage and consumes an undesirably great amount of computation time, which adversely affects the LBM solution efficiency.

A new interpolation strategy of delight was proposed by Geier [45] for LBM mesh refinement, which relies on a "bubble" function that eliminates the need of time interpolation and, thus, results in an explicit scheme. Moreover, the "bubble" formulation guarantees second-order numerical precision. Hence, the "bubble" function based interpolation technique renders the coupling of local mesh refinements and LBM free of worry about worsened numerical solution accuracy or significantly increased computational cost. The "bubble" function coupled with local mesh refinement method has been successfully applied in LBM and, very hopefully, it can also greatly benefit the adaptive mesh refinement (AMR) version of LBM as well as the AMR-LBM with immersed boundary (IB) approach incorporated.

The goal of this thesis is to derive a new numerical framework that incorporates both the immersed boundary (IB) model and the "bubble" function based local mesh refinement algorithm into the lattice Boltzmann method (LBM). This new LBM model featuring incorporation of more

innovative numerical strategies, i.e., both IB and AMR, should be able to globally reduce the simulation cost in terms of data storage and computation time while maintaining the solution accuracy on a satisfactory level. The AMR-IB-LBM model presented in this study may render the LBM approach more efficient in simulating complex fluid flow problems arising from engineering applications.

### **1.3. Thesis organization**

After this introduction chapter which describes the background, motivation, objective, and significance of this study, the rest of this thesis is organized as follows: Chapter 2 explains in detail the LBM and IB approaches, and the necessary modifications required for the present study; then, Chapter 3 will describe the multi-level local mesh refinement techniques in both imposed and AMR versions; a number of benchmark cases will be tested in Chapter 4, and the validity of the present AMR-IB-LBM model can be demonstrated through detailed comparisons against available relevant numerical results previously obtained by others; finally, concluding remarks about the AMR-IB-LBM model are made in Chapter 5 with possible extensions outlined there as well.

## Chapter 2. Two-Dimensional Lattice-Boltzmann Model

### 2.1. Two-Dimensional Nine-Velocity (D2Q9) Model

The Lattice-Boltzmann method (LBM) assumes fluid to be divided into a large amount of discrete particles which is viewed as a mesoscopic method. These particles are settled in fixed nodes and are allowed to move from one node to another based on the given discrete velocity. So the fundamental task of LBM is to determine the movement track or discrete velocity of all fluid particles. In the present study, the discrete velocities in the two-dimension nine-velocity (D2Q9) model [46] are defined as:

$$\vec{e}_a = \begin{cases} \vec{0} & a = 0 \\ c \left( \cos \left[ (a-1)\pi/2 \right], \sin \left[ (a-1)\pi/2 \right] \right)^T & a = 1, 2, 3, 4 \\ \sqrt{2}c \left( \cos \left[ (a-9/2)\pi/2 \right], \sin \left[ (a-9/2)\pi/2 \right] \right)^T & a = 5, 6, 7, 8 \end{cases} \quad (2.1)$$

where  $\vec{e}_a$  and  $c$  stand for the velocity of a fluid particle and the lattice speed respectively. The lattice speed  $c = \delta x / \delta t$  takes 1, where  $\delta x$  is the lattice size ( $\delta x = \delta y$  in the case of two-dimensional square lattice) and  $\delta t$  is the time step. As illustrated in Figure 2.1, the D2Q9 model means the square lattice occupies a unit area, the fluid particle at the central node owns eight nearest neighboring nodes and accordingly forms eight paths. The interpretation of (2.1) is that, along a path, a particle can move from the central node to only one of its nearest neighboring nodes within one time step.

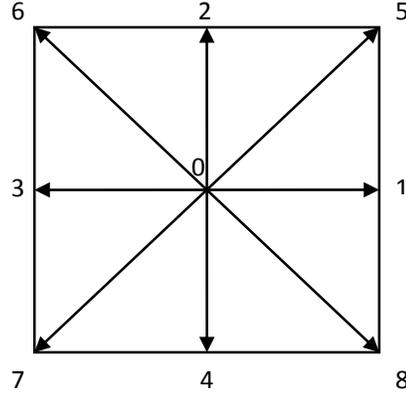


Figure 2.1 D2Q9 model on a square lattice

## 2.2. Single-Relaxation-Time Lattice-Boltzmann Equation

The Boltzmann equation with the single relaxation time (SRT) approximation reads:

$$\frac{\partial f}{\partial t} + \vec{\xi} \cdot \vec{\nabla} f = -\frac{1}{\tau} (f - f^{(eq)}) \quad (2.2)$$

where  $\vec{\xi}$  is the particle velocity,  $f$  is the particle distribution function,  $\tau$  is the single relaxation time, and  $f^{(eq)}$  is the equilibrium distribution function. The lattice-Boltzmann equation (LBE) can be derived by temporal and spatial discretizations of Eqn.(2.2), leading to the following LBM model [9]:

$$f_a(\vec{x} + \vec{e}_a \Delta t, t + \Delta t) = f_a(\vec{x}, t) - \frac{1}{\tau} (f_a(\vec{x}, t) - f_a^{eq}(\vec{x}, t)) \quad (2.3)$$

which is the single-relaxation-time (SRT) LBE, and also referred to as LBGK model as it involves the Bhatnagar-Gross-Krook relaxation term. In Eqn.(2.3),  $f_a(\vec{x}, t)$  is the distribution function at  $(\vec{x}, t)$ ,  $f_a^{eq}(\vec{x}, t)$  is the corresponding equilibrium distribution function, which can be obtained by:

$$f_a^{eq}(\vec{x}, t) = \rho w_a \left[ 1 + \frac{\vec{e}_a \cdot \vec{u}}{c_s^2} + \frac{(\vec{e}_a \cdot \vec{u})^2 - c_s^2 |\vec{u}|^2}{2c_s^4} \right] \quad (2.4)$$

where  $w_a$  is the weighting factor given by

$$w_a = \begin{cases} 4/9, & a = 0 \\ 1/9, & a = 1, 3, 5, 7 \\ 1/36, & a = 2, 4, 6, 8 \end{cases} \quad (2.5)$$

and  $c_s = \frac{c}{\sqrt{3}} = \frac{1}{\sqrt{3}}$  represents the sound speed in this model,  $\vec{e}_a$  is defined by Eqn.(2.1). The macroscopic density,  $\rho$ , velocity,  $\vec{u}$ , and pressure,  $p$ , are defined in terms of distribution function as follows:

$$\rho = \sum_a f_a \quad (2.6)$$

$$\rho \vec{u} = \sum_a \vec{e}_a f_a \quad (2.7)$$

$$p = \rho c_s^2 \quad (2.8)$$

and the relaxation time,  $\tau$ , is related to the kinematic viscosity,  $\nu$ , as follows:

$$\nu = (\tau - 1/2) c_s^2 \delta t \quad (2.9)$$

which implies the relaxation time  $\tau$  has to be greater than 0.5 to guarantee the positivity of viscosity. In practice, for a given Reynolds number, the characteristic velocity for the LBE system, and an appropriately chosen lattice size, the kinematic viscosity,  $\nu$ , can be calculated, and then the relaxation time,  $\tau$ , can be determined.

According to the LBGK model, the particle distribution functions should be updated by splitting each time increment step into two steps, one is called collision process and the other is for streaming. The collision step occurs when the particles are moving toward the same node, and the interaction between the particles will result in the change of the particle distribution function and the direction of velocity according to the scattering rule. Then the particle can be considered to stream

to the nearest neighbor node based on its discrete velocities shown in Eqn.(2.1).

The two LGBK steps are computed as follows:

( I ) collision step:

$$f_a^+(\vec{x}, t) = f_a(\vec{x}, t) - \frac{1}{\tau} (f_a(\vec{x}, t) - f_a^{eq}(\vec{x}, t)) \quad (2.10)$$

( II ) streaming step:

$$f_a(\vec{x}, t + \Delta t) = f_a^+(\vec{x} - \vec{e}_a \Delta t, t) \quad (2.11)$$

where the " + " symbol on  $f_a$  denotes the distribution function in its post-collision state.

### 2.3. Multiple-Relaxation-Time LBE

As implied in Eqn.(2.9), the closer the relaxation time is to 0.5, the less the numerical instability; once the relaxation time reaches 0.5 or when it is less than 0.5, the LBGK model may break down. To overcome this bottleneck situation in LBGK, a Multiple-Relaxation-Time (MRT) model was proposed in [47], leading to the following revised form of the governing equation:

$$\mathbf{f}(\vec{x} + \vec{e}_a \Delta t, t + \Delta t) = \mathbf{f}(\vec{x}, t) - \mathbf{M}^{-1} \mathbf{S} [\mathbf{m}(\vec{x}, t) - \mathbf{m}^{eq}(\vec{x}, t)] \quad (2.12)$$

where the moment space  $\mathbf{m}$  is arranged in the following order:  $m_0 = \rho$  is the fluid density,  $m_1 = e$  is the energy,  $m_2 = \varepsilon$  is related to the energy square,  $m_{3,5} = j_{x,y}$  are two components of the momentum  $\vec{j} = (j_x, j_y)^T = \rho(u_x, u_y)^T$ ,  $m_{4,6} = q_{x,y}$  are related to energy flux,  $m_{7,8} = p_{xx,yy}$  are components of the symmetric and traceless strain rate tensor. Totally, the moment space  $\mathbf{m}$  is related to the distribution function by the following  $9 \times 9$  transformation matrix  $\mathbf{M}$ :

$$\mathbf{m} = \begin{pmatrix} \rho \\ e \\ \varepsilon \\ j_x \\ q_x \\ j_y \\ q_y \\ p_{xx} \\ p_{xy} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \end{pmatrix} = \mathbf{M}\mathbf{f} \quad (2.13)$$

Note the row vectors in the above  $9 \times 9$  equation are mutually orthogonal, i.e.,  $\mathbf{M}\mathbf{M}^T$  is a diagonal matrix; thus,  $(\mathbf{M}\mathbf{M}^T)^{-1}$  can be easily obtained. It is convenient to compute the matrix using:

$$\mathbf{M}^{-1} = \mathbf{M}^T (\mathbf{M}\mathbf{M}^T)^{-1} \quad (2.14)$$

and  $\mathbf{m}^{eq}$  groups the corresponding equilibrium values as follows:

$$\mathbf{m}^{eq} = \rho \begin{pmatrix} 1 \\ -2 + 3u^2 \\ 1 - 3u^2 \\ u_x \\ -u_x \\ u_y \\ -u_y \\ u_x^2 - u_y^2 \\ u_x u_y \end{pmatrix} \quad (2.15)$$

The relaxation matrix  $\mathbf{S}$  is a diagonal matrix:

$$\mathbf{S} = \text{diag}(s_0, s_1, \dots, s_8) = \text{diag}(0, s_e, s_\varepsilon, 0, s_q, 0, s_q, s_v, s_v) \quad (2.16)$$

where  $s_0 = s_3 = s_5$  can be set at 0 to reduce the computational cost because of the mass and

momentum conservation during the collision process. The remaining relaxation parameters are set as follows:

$$e^+ = e - s_e [e - e^{(eq)}], \quad (2.17)$$

$$\varepsilon^+ = \varepsilon - s_\varepsilon [\varepsilon - \varepsilon^{(eq)}], \quad (2.18)$$

$$q_x^+ = q_x - s_q [q_x - q_x^{(eq)}], \quad (2.19)$$

$$q_y^+ = q_y - s_q [q_y - q_y^{(eq)}], \quad (2.20)$$

$$p_{xx}^+ = p_{xx} - s_v [p_{xx} - p_{xx}^{(eq)}], \quad (2.21)$$

$$p_{xy}^+ = p_{xy} - s_v [p_{xy} - p_{xy}^{(eq)}], \quad (2.22)$$

where the variables with superscript+ are post-collision values. The relaxation rates  $s_v$  and  $s_e$  are related to the kinematic viscosity  $\nu$  and bulk viscosity  $\zeta$  :

$$\nu = c_s^2 \left( \frac{1}{s_v} - \frac{1}{2} \right) \delta_t \quad (2.23)$$

$$\zeta = c_s^2 \left( \frac{1}{s_e} - \frac{1}{2} \right) \delta_t \quad (2.24)$$

The relaxation rates  $s_e, s_q, s_\varepsilon$  can be adjusted within the interval of  $(0, 2)$ . Note when  $s_e = s_q = s_\varepsilon = 1/\tau$ , MRT LBE will restore to LBGK model.

Similar to the LBGK model, the MRT evolution process includes two steps, namely collision and stream. The collision step is executed in the momentum space, and the stream step is executed in the velocity space, which can be respectively described by:

$$\mathbf{m}^+(\vec{x}, t) = \mathbf{m}(\vec{x}, t) - \mathbf{M}^{-1} \mathbf{S} [\mathbf{m}(\vec{x}, t) - \mathbf{m}^{eq}(\vec{x}, t)] \quad (2.25)$$

and

$$f(\vec{x} + \vec{e}_a \Delta t, t + \Delta t) = f^+(\vec{x}, t) = \mathbf{M}^{-1} \mathbf{m}^+(\vec{x}, t) \quad (2.26)$$

where  $f^+(\vec{x}, t)$  and  $\mathbf{m}^+(\vec{x}, t)$  respectively denote the density distribution function and the moment space after the collision step.

## 2.4. Boundary Condition

The standard LBM solution procedure, which has been described in the precedent section, is no longer executable for nodes located at boundaries. On the other hand, the values of the distribution function at those boundary nodes are involved in the streaming step of the LBM computation. Therefore, the distribution function at nodes on the boundary must be determined in a fashion that is consistent with the LBM model. Technically, the boundary condition treatments have to be made differently for straight and curved boundaries in the LBM framework to ensure the numerical accuracy, stability, and efficiency.

### 2.4.1. Straight boundary

There are two mechanisms for setting up the boundary condition for a straight boundary, namely bounce-back scheme and non-equilibrium extrapolation scheme.

#### 2.4.1.1. Bounce-back scheme

For the stationary no-slip boundary [48], bounce-back scheme is the most common method. As shown in Figure 2.2, node  $(i,1)$  is located on a straight boundary represented by a solid line,

and the bounce-back mechanism used in LBM signifies the distribution function  $f_8$  streams from node  $(i-1, 2)$  to  $(i, 1)$  and returns in the same path without going through the collision step, resulting in:

$$f_6(i, 1) = f_8(i-1, 2) \quad (2.27)$$

Similarly, other distribution functions at the node  $(i, 1)$  can be obtained as follows:

$$f_2(i, 1) = f_4(i-1, 2) \quad (2.28)$$

$$f_5(i, 1) = f_7(i-1, 2) \quad (2.29)$$

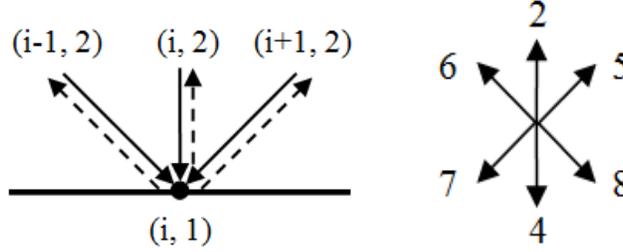


Figure 2.2 Bounce-back scheme

This bounce-back boundary scheme has been widely used owing to its simple principle and easy implementation. However, this scheme is unable to render highly accurate solution due to its first-order accuracy.

#### 2.4.1.2. Non-equilibrium extrapolation scheme

For the purpose of accuracy improvement, Guo et al [49] presented a non-equilibrium extrapolation scheme, which features second-order accuracy in both time and space and enhanced numerical stability.

The basic idea of the non-equilibrium extrapolation scheme is to decompose the distribution

function at the boundary node into its equilibrium and non-equilibrium parts, and numerically treat these two parts in different ways before re-combining them to obtain the values of distribution function at there. That is, at boundary nodes, the equilibrium part is approximated using Eqn.(2.4) and the boundary condition if points at the boundary are involved in this approximation, and the non-equilibrium part is obtained using the first-order extrapolation of the non-equilibrium part of the distribution function at the neighboring fluid node. As depicted in Figure 2.3, a particle  $\vec{x}_b$  is located at a boundary node, and the sides above and below the boundary are fluid and solid, respectively. The unknown distribution functions are  $f_2, f_5, f_6$ , and all the macroscopic density and velocity at along the streaming nodes  $\vec{x}_f = \vec{x}_b + \vec{e}_a \Delta t (a = 2, 5, 6)$  are known. To determine the unknown distribution function on the boundary point, the distribution function of particle  $\vec{x}_b$  is divided to two parts:

$$f_a(\vec{x}_b, t) = f_a^{eq}(\vec{x}_b, t) + f_a^{neq}(\vec{x}_b, t), a = 2, 5, 6 \quad (2.30)$$

where  $f_a^{eq}(\vec{x}_b, t)$  and  $f_a^{neq}(\vec{x}_b, t)$  represent the equilibrium and non-equilibrium parts of the function, respectively. For the equilibrium part, as the macroscopic density at the node represented by  $\vec{x}_b$  is unknown, the missing information would be replaced by the value at node  $\vec{x}_f$ . Therefore, the solid boundary density is replaced by the fluid density at  $\vec{x}_f$  near the boundary node  $\vec{x}_b$  in the calculation of equilibrium distribution function part.

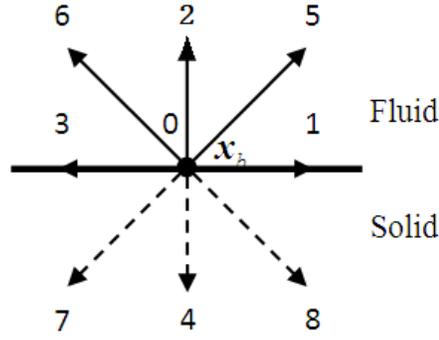


Figure 2.3 Non-equilibrium extrapolation scheme

For the non-equilibrium part, the function is approximately replaced by the non-equilibrium function at node  $\vec{x}_f$  :

$$f_a^{neq}(\vec{x}_b, t) = f_a^{neq}(\vec{x}_f, t) = f_a(\vec{x}_f, t) - f_a^{eq}(\vec{x}_f, t), a = 2, 5, 6 \quad (2.31)$$

From the above, the distribution function at the boundary node can be obtained,

$$f_a(\vec{x}_b, t) = f_a^{eq}(\vec{x}_b, t) + [f_a(\vec{x}_f, t) - f_a^{eq}(\vec{x}_f, t)], a = 2, 5, 6 \quad (2.32)$$

If the collision step is taken into consideration, the distribution function after the collision can be expressed as:

$$f_a^+(\vec{x}_b, t) = f_a^{eq}(\vec{x}_b, t) + (1 - \frac{1}{\tau}) [f_a(\vec{x}_f, t) - f_a^{eq}(\vec{x}_f, t)], a = 2, 5, 6 \quad (2.33)$$

### 2.4.2. Curved boundary

The boundary of an object immersed in a flowing fluid is not necessarily a straight line. Since the immersed objects are allowed to move as well, the immersed boundary (IB) method is adopted

in this study. In the IB approach, the flow field is represented by a set of Eulerian points, which are in fact the fixed Cartesian mesh points in the LBM frame, and the boundary of curved object, which is immersed in the fluid field, is represented by a set of Lagrangian points. These two sets of points in the IB-LBM framework are illustrated in Figure 2.4. The fundamental idea of IB is that the effect of the fluid to the object boundary is modeled by a restoring force that should be additionally placed in the governing equations.

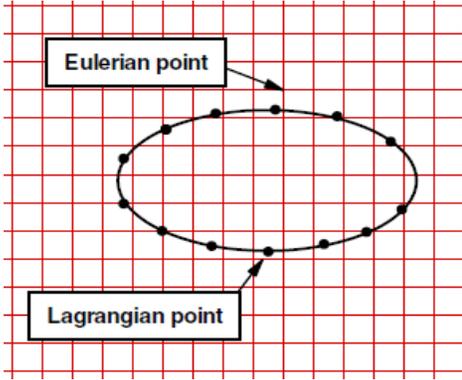


Figure 2.4 Schematic of immersed boundary method

#### 2.4.2.1. Conventional immersed boundary approach

The governing equations of the conventional IBM for an incompressible viscous fluid flow can be written as [28]

$$\rho \left( \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) + \nabla p = \mu \Delta \vec{u} + \vec{f} \quad (2.34)$$

$$\nabla \cdot \vec{u} = 0 \quad (2.35)$$

$$\vec{f}(\vec{x}, t) = \int_{\Gamma} \vec{F}(s, t) \delta(\vec{x} - \vec{X}(s, t)) ds \quad (2.36)$$

$$\frac{\partial \vec{X}(s,t)}{\partial t} = \vec{u}(\vec{X}(s,t),t) = \int_{\Omega} \vec{u}(\vec{x},t) \delta(\vec{x} - \vec{X}(s,t)) d\vec{x} \quad (2.37)$$

where  $\vec{x}$ ,  $\vec{u}$  and  $\vec{f}$  are the Eulerian coordinates, fluid velocity, and force density acting on the fluid,  $s$  represents the Lagrangian parametric coordinates,  $\rho$  and  $p$  stand for the fluid density and fluid pressure, respectively,  $\vec{X}$  and  $\vec{F}$  are the Lagrangian coordinates and boundary force density, and  $\delta(\vec{x} - \vec{X}(s,t))$  is the Dirac delta function. Eqn.(2.34) and Eqn.(2.35) are the Navier-Stokes Equations with external force. Eqn.(2.36) and Eqn.(2.37) show the interaction between the immersed object and the fluid flow by distributing the boundary force, which acts at the Lagrangian points, to the Eulerian points, and interpolating the velocity, which takes place at the Eulerian points, to the Lagrangian points as well. From the above equations, the key step is to calculate the boundary force density,  $\vec{F}$ , which is also referred to as "*restoring force*".

Conventionally, the restoring force  $\vec{F}$  is obtained by using Hooke's law as follows:

$$\vec{F}(s,t) = -k\Delta\vec{\xi} = -k(\vec{V}_{fluid}\Delta t - \vec{V}_{boundary}\Delta t) \quad (2.38)$$

where  $\vec{V}_{fluid}$  is the fluid velocity at the boundary point,  $\vec{V}_{boundary}$  is the boundary velocity of the immersed object,  $k$  is the spring coefficient, and  $\Delta\vec{\xi}$  the displacement between the object point and the coordinate reference point. This method has proven unnecessarily satisfying the non-slip boundary condition. Indeed, it is only approximately satisfied. This drawback can cause the flow penetration from the fluid field to the solid body, calling for more improvement in the IB methodology [30-33].

### 2.4.2.2. Feedback forcing immersed boundary based Lattice-Boltzmann method

To be invoked in the LBM framework, Eqn.(2.34) and Eqn.(2.35) can be replaced by LBE with an additional forcing term proposed by Guo et al [50]:

$$f_a(\vec{x} + \vec{e}_a \Delta t, t + \Delta t) - f_a(\vec{x}, t) = -\frac{1}{\tau} (f_a(\vec{x}, t) - f_a^{eq}(\vec{x}, t)) + F_a \Delta t \quad (2.39)$$

$$F_a = \left(1 - \frac{1}{2\tau}\right) w_a \left( \frac{\vec{e}_a - \vec{u}}{c_s^2} + \frac{\vec{e}_a \cdot \vec{u}}{c_s^4} \vec{e}_a \right) \cdot \vec{f} \quad (2.40)$$

$$\rho \vec{u} = \sum_a \vec{e}_a f_a + \frac{1}{2} \vec{f} \Delta t \quad (2.41)$$

where  $\vec{f}$  is the density of the force acting on the fluid due to solid boundary, and  $\vec{F}$  is the transformed version of  $\vec{f}$  that can be used in the LBE framework.

For an incompressible viscous flow field  $\Omega$ , the immersed object boundary is denoted by  $\Gamma$ ,  $\vec{x}$  and  $\vec{X}_B^l$ , ( $l = 1, 2, \dots, m$ ) stand for Eulerian coordinates and Lagrangian coordinates, respectively.

Based on Eqn.(2.37), the velocity at Lagrangian points can be discretized as

$$\vec{U}_B^l(\vec{X}_B^l, t) = \sum_{i,j} \vec{u}(\vec{x}, t) D_{ij}(\vec{x}_{ij} - \vec{X}_B^l) \Delta x \Delta y \quad (2.42)$$

$\delta(\vec{x} - \vec{X}_B^l(s, t))$  is approximated by a continuous kernel distribution:

$$\delta(\vec{x} - \vec{X}_B^l(s, t)) = D_{ij}(\vec{x}_{ij} - \vec{X}_B^l) = \delta(x_{ij} - X_B^l) \delta(y_{ij} - Y_B^l) \quad (2.43)$$

where  $\delta(r)$  is a continuous function for all real  $r$  with respect to the mesh width. Numerical value for  $\delta(r)$  is proposed by Peskin [24] as

$$\delta(r) = \begin{cases} \frac{1}{4} \left( 1 + \cos\left(\frac{\pi|r|}{2}\right) \right), & |r| \leq 2 \\ 0, & |r| > 2 \end{cases} \quad (2.44)$$

Similarly, Eqn.(2.45) is derived by Eqn.(2.36), the distribution of the Lagrangian force to the nearby

Eulerian points can be expressed as

$$\vec{f}(\vec{x}_{ij}, t) = \sum_l \vec{F}_B(\vec{X}_B^l, t) D_{ij}(\vec{x}_{ij} - \vec{X}_B^l) \Delta s_l \quad (2.45)$$

where  $\Delta s_l$  is the arc length of the boundary element. The interaction force per unit volume between the fluid and the immersed object can be calculated by the feedback law [44]

$$\vec{F}_i^n = \alpha \sum_{j=1}^n \left( (\vec{U}_B)_i^j - \vec{U}_i^j \right) \Delta t + \beta \left( (\vec{U}_B)_i^n - \vec{U}_i^n \right), i = 1, 2, \dots, m \quad (2.46)$$

where  $\vec{U}_i^j$  is the velocity of the immersed object expressed as  $\vec{U}_i^j = (\vec{X}_i^j - \vec{X}_i^{j-1}) / \Delta t$ ,  $j$  denotes the  $j$ th time step and  $i$  represents the Lagrangian point surround the object boundary (apparently, for a stationary object,  $\vec{U}_i^j = \vec{0}$ ),  $\vec{U}_B$  is the velocity of the boundary point, which has been calculated using the IB model,  $\alpha$  and  $\beta$  are negative free constants having dimensions of  $ML^{-3}T^{-2}$  and  $ML^{-3}T^{-1}$ , respectively. According to [44],  $\alpha$  and  $\beta$  are stable for moderate values and are both insensitive to their exact values, and the stability interval is [-100, -1] for both.

The solution process of this feedback IB based LBM framework can be summarized as follows:

- 1) Set initial values and set the force density  $\vec{f} = \vec{0}$  as the initial condition;
- 2) Use Eqn.(2.39) to get the distribution function at time  $t = t_n$  (initially  $\vec{f} = \vec{0}$ ), then compute the macroscopic variables, such as density, pressure and velocity using Eqn.(2.6)-Eqn.(2.8);
- 3) Obtain the force density using Eqn.(2.42) and Eqn.(2.46);
- 4) Interpolate the Lagrangian force term into Eulerian coordinate which is exerted on fluid by

the immersed boundary;

5) Compute the equilibrium distribution function using Eqn.(2.4);

6) Repeat Step 2) to Step 5) until convergence is reached.

## Chapter 3. Mesh Refinement Methods

Structured Cartesian meshes with uniform square cells have been routinely employed in standard two dimensional LBM applications, which may cause low efficiency and time-consuming in the simulation of complex flows. This occurs, for example, when there is a mandatory for a high resolution in critical regions. When employed traditional LBM, an elevated node density has to be maintained uniformly over the entire computational domain including the zones of little sensitivity. To overcome the drawback of traditional LBM, mesh refinement is introduced to incorporate into LBM.

### 3.1. Multi-level local mesh refinement method

To illustrate the main idea of this local refinement method, a two-level mesh system consisting of a coarse level and a fine level mesh [35] is depicted in Figure 3.1. The ratio of the lattice space between the coarse-fine mesh is defined as:

$$m = \frac{x_c}{x_f} \quad (3.1)$$

where the subscript c and f stand for the coarse mesh and the fine mesh respectively.

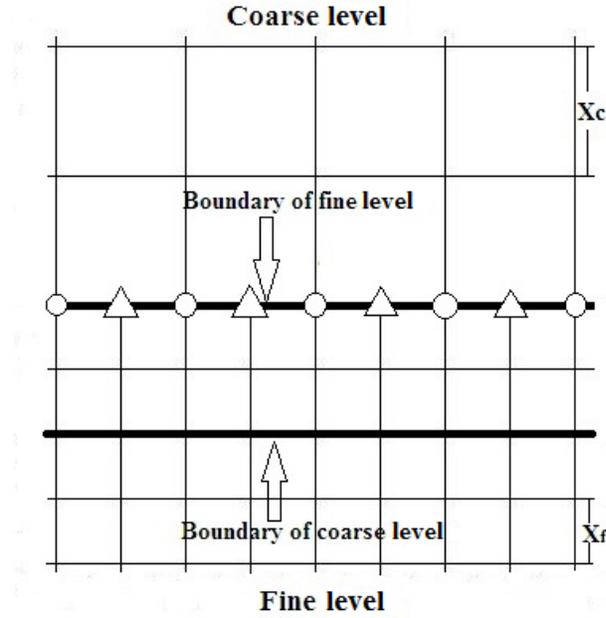


Figure 3.1 Structure of two-level mesh of different lattice spacings

The viscosity of the fluid in a given lattice size  $\delta_x$  is defined as:

$$\nu = (2\tau - 1)\delta_x c / 6 \quad (3.2)$$

In order to keep a consistent viscosity in the entire flow field, the relation between relaxation time,  $\tau_c$  on the coarse mesh and  $\tau_f$  on the fine mesh, which can be derived from (3.2), should satisfy the following equation:

$$\tau_f = \frac{1}{2} + m \left( \tau_c - \frac{1}{2} \right) \quad (3.3)$$

Then density and velocity must be continuous across the interface between coarse-fine mesh, so based on Eqn.(2.4),

$$f_{a,c}^{eq} = f_{a,f}^{eq} \quad (3.4)$$

should be guaranteed.

To maintain continuity in deviatoric stresses,

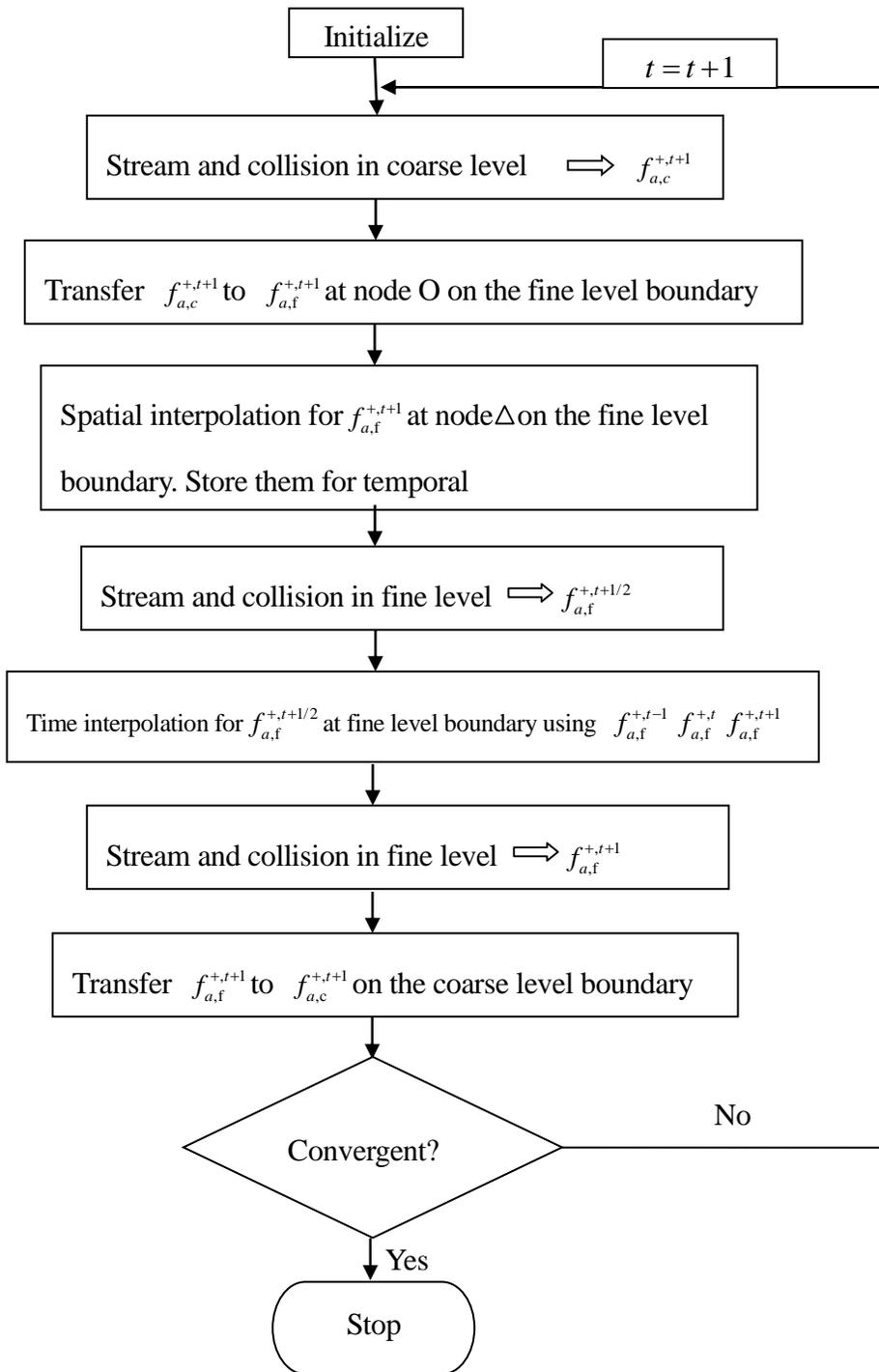
$$\left(1 - \frac{1}{2\tau_c}\right) f_{a,c}^{neq} = \left(1 - \frac{1}{2\tau_f}\right) f_{a,f}^{neq} \quad (3.5)$$

Combining Eqns.(2.10), (3.4) and (3.5), the transform of distribution function between two-mesh can be derived:

$$f_{a,f} = f_{a,c}^{eq} + \frac{\tau_f - 1}{m(\tau_c - 1)} (f_{a,c} - f_{a,c}^{eq}) \quad (3.6)$$

$$f_{a,c} = f_{a,f}^{eq} + m \frac{\tau_c - 1}{\tau_f - 1} (f_{a,f} - f_{a,f}^{eq}) \quad (3.7)$$

The procedure for the multi-level LBM can be summarized as follows:



### 3.2. Bubble function for local mesh refinement used in lattice-Boltzmann method

Bubble function interpolation method is proposed as a second-order accuracy approach for the mesh refinement [45]. Since the bubble function is obtained by the second-order variables  $k_{xx}$ ,  $k_{yy}$ ,  $k_{xy}$  from moment space, MRT model should be employed when incorporating this method into LBM.

The main idea for this interpolation method is allowing for a small overlapping region. As shown in Figure 3.2, two more coarse mesh units exist in coarse-fine overlapping region. With this strategy, temporal interpolation can be eliminated. In the fine-to-coarse overlapping unit, the moments for the single coarse node in the middle of the four fine nodes is updated with interpolated momentum values from the four fine nodes, which are taken from the bubble function. Also, in the coarse-to-fine overlapping unit, the moments for the fine nodes are obtained from the coarse neighboring nodes using bubble function by bilinear interpolation. One time step on the coarse mesh corresponding to two time steps on the fine mesh. The schematic of the overlay region is depicted in Figure 3.2.

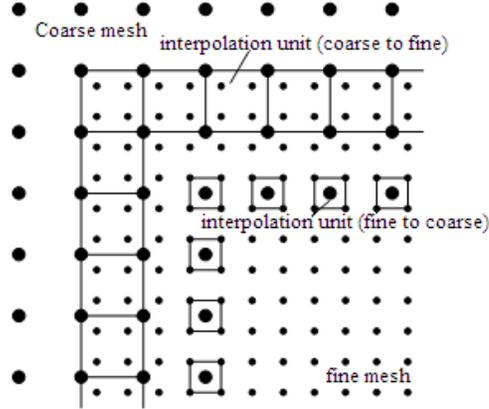


Figure 3.2 Schematic of an overlay region

In moment space, density  $\rho$  and momentum  $(\pi_x, \pi_y)$  can be calculated by Eqns.(2.6)-(2.7),

and other six moments  $k_{xy}, k_{xx} - k_{yy}, k_{xx} + k_{yy}, k_{xxy}, k_{xyy}, k_{xxyy}$  are defined as:

$$k_{x^m y^n} = \sum_a (e1_a - u_x)^m (e2_a - u_y)^n f_a, (m, n = 0, 1, 2) \quad (3.8)$$

where  $e1_a$  and  $e2_a$  are the velocity components for the respective particles. Four non-dimensional nodes consist of one square grid, and the four nodes are numbered as (0, 0), (1, 0), (0, 1), (1, 1), respectively. The aim is to find a momentum interpolation function based on the four nodes, which can not only satisfy the second-order accuracy, but also require no information of other nodes far away. Therefore, bubble function is put forward. Two of the moments in the pre-collision step are of great importance:  $k_{xx}$  and  $k_{xx} - k_{yy}$ . They are related to the shear stress and partial derivative of momentum.

$$k_{xy} = -\frac{c_s^2}{w_{xy}} \left( \frac{\partial \pi_y}{\partial x} + \frac{\partial \pi_x}{\partial y} \right) \quad (3.9)$$

$$k_{xx} - k_{yy} = -\frac{2c_s^2}{w_{xx}} \left( \frac{\partial \pi_x}{\partial x} - \frac{\partial \pi_y}{\partial y} \right) \quad (3.10)$$

where  $w_{xy} = w_{xx} = 1/\tau$ . The two coefficients are the relaxation rate of second-order momentum used to adjust shear viscosity. And four second-order partial derivative at the center point can be easily derived based on Eqns.(3.9) and (3.10):

$$\frac{\partial k_{xy}}{\partial x} = -\frac{c_s^2}{w_{xy}} \left( \frac{\partial^2 \pi_y}{\partial x^2} + \frac{\partial^2 \pi_x}{\partial x \partial y} \right) \quad (3.11)$$

$$\frac{\partial k_{xy}}{\partial y} = -\frac{c_s^2}{w_{xy}} \left( \frac{\partial^2 \pi_y}{\partial x \partial y} + \frac{\partial^2 \pi_x}{\partial y^2} \right) \quad (3.12)$$

$$\frac{\partial (k_{xx} - k_{yy})}{\partial x} = -\frac{2c_s^2}{w_{xx}} \left( \frac{\partial^2 \pi_x}{\partial x^2} - \frac{\partial^2 \pi_y}{\partial x \partial y} \right) \quad (3.13)$$

$$\frac{\partial (k_{xx} - k_{yy})}{\partial y} = -\frac{2c_s^2}{w_{xx}} \left( \frac{\partial^2 \pi_x}{\partial x \partial y} - \frac{\partial^2 \pi_y}{\partial y^2} \right) \quad (3.14)$$

Then second-order accuracy finite difference of  $k_{xx}$  and  $k_{xx} - k_{yy}$  can be obtained by central difference scheme between the diagonally neighboring nodes (Figure 3.3):

$$\frac{\partial k_{xy}}{\partial x} + \frac{\partial k_{xy}}{\partial y} \approx \frac{k_{xy}^{(1,1)} - k_{xy}^{(0,0)}}{l} \quad (3.15)$$

$$\frac{\partial k_{xy}}{\partial x} - \frac{\partial k_{xy}}{\partial y} \approx \frac{k_{xy}^{(1,0)} - k_{xy}^{(0,1)}}{l} \quad (3.16)$$

$$\frac{\partial (k_{xx} - k_{yy})}{\partial x} + \frac{\partial (k_{xx} - k_{yy})}{\partial y} \approx \frac{(k_{xx} - k_{yy})^{(1,1)} - (k_{xx} - k_{yy})^{(0,0)}}{l} \quad (3.17)$$

$$\frac{\partial (k_{xx} - k_{yy})}{\partial x} - \frac{\partial (k_{xx} - k_{yy})}{\partial y} \approx \frac{(k_{xx} - k_{yy})^{(1,0)} - (k_{xx} - k_{yy})^{(0,1)}}{l} \quad (3.18)$$

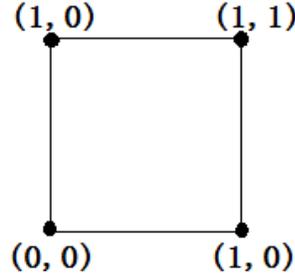


Figure 3.3 Numbering for neighboring nodes

where  $l = 1$  denotes the distance between two neighboring nodes, and the subscript stands for the coordinate of the node mentioned above. Together with the above equations, quadratic bubble function can be solved as follows:

$$\pi_x(x, y) = a_0 + a_x x + a_{xx} x^2 + a_{xy} xy + a_y y + a_{yy} y^2 \quad (3.19)$$

$$\pi_y(x, y) = b_0 + b_x x + b_{xx} x^2 + b_{xy} xy + b_y y + b_{yy} y^2 \quad (3.20)$$

The coefficients are:

$$a_0 = \pi_x^{(0,0)} \quad (3.21)$$

$$a_x = -0.5 \cdot (m_1 + \pi_y^{(0,0)} - \pi_y^{(0,1)} - \pi_y^{(1,0)} + \pi_y^{(1,1)} + 2\pi_x^{(0,0)} - 2\pi_x^{(1,0)}) \quad (3.22)$$

$$a_y = -0.5 \cdot (m_4 - \pi_y^{(0,0)} + \pi_y^{(0,1)} + \pi_y^{(1,0)} - \pi_y^{(1,1)} + 2\pi_x^{(0,0)} - 2\pi_x^{(0,1)}) \quad (3.23)$$

$$a_{xx} = 0.5 \cdot (m_1 + \pi_y^{(0,0)} - \pi_y^{(0,1)} - \pi_y^{(1,0)} + \pi_y^{(1,1)}) \quad (3.24)$$

$$a_{xy} = \pi_x^{(0,0)} - \pi_x^{(0,1)} - \pi_x^{(1,0)} + \pi_x^{(1,1)} \quad (3.25)$$

$$a_{yy} = 0.5 \cdot (m_4 - \pi_y^{(0,0)} + \pi_y^{(0,1)} + \pi_y^{(1,0)} - \pi_y^{(1,1)}) \quad (3.26)$$

$$b_0 = \pi_y^{(0,0)} \quad (3.27)$$

$$b_x = -0.5 \cdot (m_3 + 2\pi_y^{(0,0)} - 2\pi_y^{(1,0)} - \pi_x^{(0,0)} + \pi_x^{(0,1)} + \pi_x^{(1,0)} - \pi_x^{(1,1)}) \quad (3.28)$$

$$b_y = -0.5 \cdot \left( -m_2 + 2\pi_y^{(0,0)} - 2\pi_y^{(0,1)} + \pi_x^{(0,0)} - \pi_x^{(0,1)} - \pi_x^{(1,0)} + \pi_x^{(1,1)} \right) \quad (3.29)$$

$$b_{xx} = -0.5 \cdot \left( -m_3 + \pi_x^{(0,0)} - \pi_x^{(0,1)} - \pi_x^{(1,0)} + \pi_x^{(1,1)} \right) \quad (3.30)$$

$$b_{xy} = \pi_y^{(0,0)} - \pi_y^{(0,1)} - \pi_y^{(1,0)} + \pi_y^{(1,1)} \quad (3.31)$$

$$b_{yy} = -0.5 \cdot \left( m_2 - \pi_x^{(0,0)} + \pi_x^{(0,1)} + \pi_x^{(1,0)} - \pi_x^{(1,1)} \right) \quad (3.32)$$

In order to express more clearly in each coefficient next, a few auxiliary variables are defined:

$$m_1 = -\frac{w_{xx}}{4c_s^2} \left( (k_{xx} - k_{yy})^{(1,0)} + (k_{xx} - k_{yy})^{(1,1)} - (k_{xx} - k_{yy})^{(0,1)} - (k_{xx} - k_{yy})^{(0,0)} \right) \quad (3.33)$$

$$m_2 = -\frac{w_{xx}}{4c_s^2} \left( (k_{xx} - k_{yy})^{(0,1)} + (k_{xx} - k_{yy})^{(1,1)} - (k_{xx} - k_{yy})^{(1,0)} - (k_{xx} - k_{yy})^{(0,0)} \right) \quad (3.34)$$

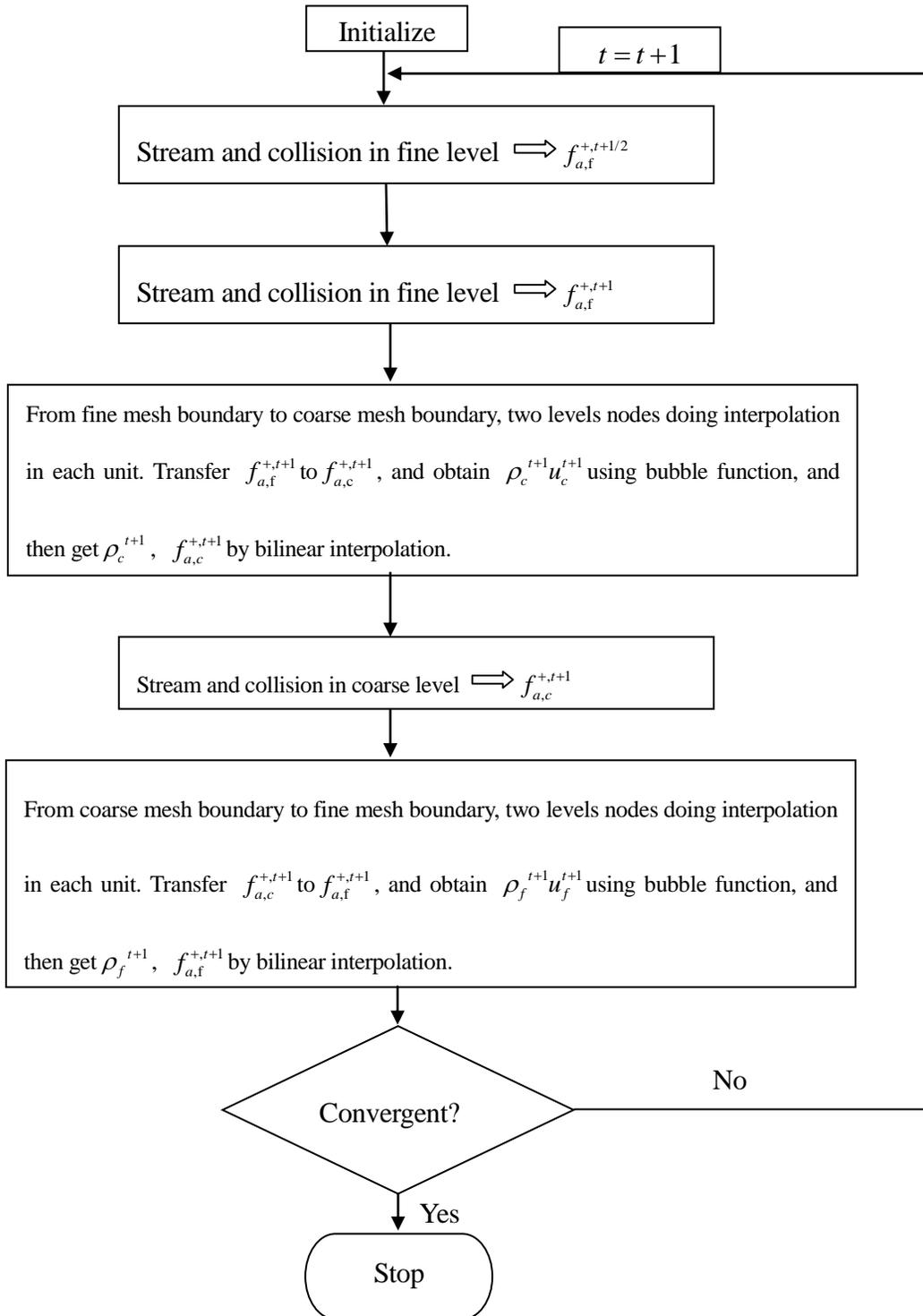
$$m_3 = -\frac{w_{xy}}{2c_s^2} \left( k_{xy}^{(1,0)} + k_{xy}^{(1,1)} - k_{xy}^{(0,1)} - k_{xy}^{(0,0)} \right) \quad (3.35)$$

$$m_4 = -\frac{w_{xy}}{2c_s^2} \left( k_{xy}^{(0,1)} + k_{xy}^{(1,1)} - k_{xy}^{(1,0)} - k_{xy}^{(0,0)} \right) \quad (3.36)$$

Then density  $\rho$  and distribution function  $f$  can be obtained by bilinear interpolation. Velocity

$u_x = \frac{\pi_x}{\rho}$  and  $u_y = \frac{\pi_y}{\rho}$ . The procedure for the bubble function interpolation method can be depicted

as follows:



### 3.3. Adaptive mesh refinement (AMR) technique

Different from the imposed local mesh refinement strategy described in Section 3.1, the adaptive mesh refinement (AMR) algorithm is more complex since it involves an unstructured data storage process. A number of lists of interrelated data corresponding to different levels of the adaptive mesh are constructed in the in-house CFD code.

The dynamic refinement and coarsening of grid cells is controlled by monitoring parameters which estimate the error for a quantity of interest. For LBM, several error estimators were discussed in the literature. Among these methods, Crouse et al. [41] proposed to use the divergence of the velocity. Here depending on different situations, the gradient of velocity or vorticity is used as the criterion to guide the AMR process. Detailed information about the criterion will be shown in Chapter 4.

When incorporating AMR into IB-LBM, the bubble function is also employed for the solution interpolation purpose. Considering the data structure in AMR in-house code, each mesh cell is defined as the structure variable, which consists of 4 pointers in 4 nodes, 8 neighboring elements (Figure 3.4 (a)) and the pointer corresponding to both the next level and last level of the cell. For the node structure variable, the relative coordinates, macroscopic variables, the pointer of 9 neighboring nodes (Figure 3.4 (b)) should all be included.

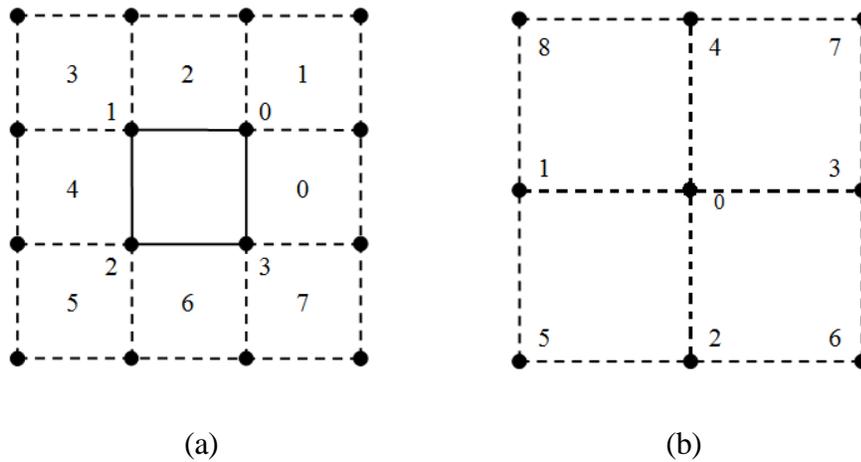


Figure 3.4 Numbering example for (a) neighboring cells and (b) neighboring nodes

After initializing the flow field, a uniform Cartesian mesh, named level 0 underlying mesh is generated, with defined node and cell structure variables. Then each cell should be determined to be refined or not using the criterion mentioned above. For these cells that need to be refined, a new mesh cell with half meshsize will be inserted into the origin one, which defined as level 1 (Figure 3.5 (a)). Then an index relation between different levels is generated. With the new formed cell and its corresponding neighbors, a new level of mesh, which possesses a higher resolution can be generated (Figure 3.5 (b)). Using the same strategy, mesh numbering from level 0 to level 4 is formed at the initialize step ahead of calculate step.

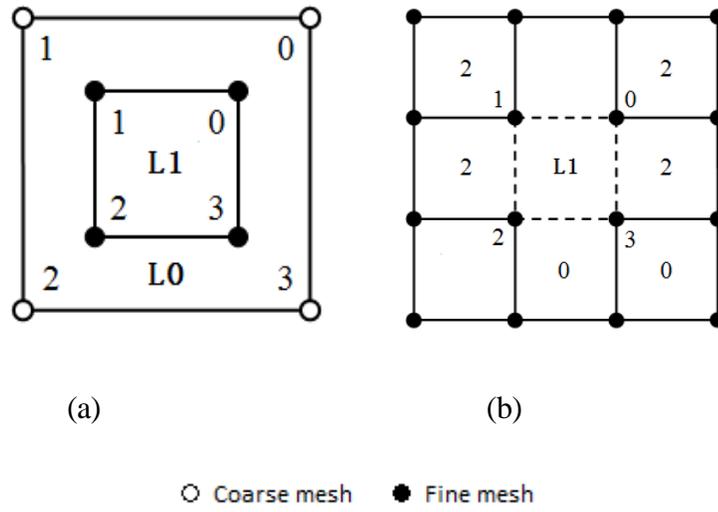


Figure 3.5 Schematic for bubble function interpolation used in AMR

## Chapter 4. Numerical Experiments

### 4.1. Introduction

To verify the validity and accuracy of the present IB-LBM approach, a series of incompressible viscous flows past a cylinder or bi-cylinders are chosen as numerical experiments. The cylinder(s) may remain at rest, or keep in motion to make scenarios of moving objects in the flowing fluid. Such selected problems have been studied extensively and are considered as the benchmark cases when justifying new approaches. A large number of results including numerical, theoretical, and experimental studies are available in literature, and some will serve as references for comparison purpose in this chapter, as seen in subsequent sections.

Incompressible viscous flows around a cylinder are usually differentiated using Reynolds number, which is defined as:

$$\text{Re} = \frac{U_{\infty} D}{\nu} \quad (4.1)$$

where  $U_{\infty}$  is the free stream velocity in the case of a stationary cylinder, or the relative velocity of the free stream to the moving cylinder,  $D$  is the diameter of the cylinder, and  $\nu$  is the kinematic viscosity of the fluid. For this type of flow, the drag force,  $F_D$ , arises from two sources: the shear stress and the pressure distribution along the body [51]. Since the cross section of a cylinder remains in circular shape, which is symmetric about center, the angle of attack for the incoming flow is

conveniently always set at zero, and the drag force can be calculated using:

$$F_D = -\int_{\Omega} f_x dv \quad (4.2)$$

where  $f_x$  stands for the  $x$ -component of the force density. Then, the drag coefficient is then defined as:

$$C_d = \frac{2F_D}{\rho U_{\infty}^2 D} \quad (4.3)$$

and the lift coefficient can be defined as:

$$C_l = \frac{2F_L}{\rho U_{\infty}^2 D} \quad (4.4)$$

where  $F_L$  is the lift force obtained by:

$$F_L = -\int_{\Omega} f_y dv \quad (4.5)$$

in which  $f_y$  represents the  $y$ -component of the force density. The pressure coefficient,  $C_p$ , is calculated by:

$$C_p = \frac{2(p_B - p_{\infty})}{\rho U_{\infty}^2} \quad (4.6)$$

where  $p_B$  and  $p_{\infty}$  are the pressure on the boundary of cylinder and far-field pressure, respectively.

Similar to Eqn.(2.42) that has been described in Section 2.4.2.2,  $p_B$  can be obtained by the following interpolation in the flow field:

$$p_B = \sum_{i,j} p(\vec{x}, t) D_{ij} (\vec{x}_{ij} - \vec{X}_B^l) \Delta x \Delta y \quad (4.7)$$

In all present test cases, the fluid density is set at  $\rho = 1.0$  and the free stream velocity is  $U_{\infty} = 0.1$ , unless otherwise specified. The diameter of the cylinder is 5. The computational domain

occupies a  $512 \times 256$  rectangular domain, with its horizontal central line located on the  $x$ -axis and starting at the origin of the Cartesian coordinate system.

For each case, as a result of pre-implemented local mesh refinements, four levels of mesh in different grid resolutions are used in the simulations, and the cell unit in each level is  $\Delta x = \Delta y = 1, 0.5, 0.25, 0.125$ , respectively. The zone covered by each level of mesh is specified by using the minimum and maximum  $x$ - and  $y$ - coordinates, denoted by  $x_{nL}, x_{nR}, y_{nL}, y_{nU}$ , ( $n = 1, 2, 3, 4$ ), as shown in Figure 4.1. Unless otherwise notified, four levels of mesh occupy four nested zones respectively defined by  $[0, 512] \times [-128, 128]$ ,  $[39, 295] \times [-64, 64]$ ,  $[59, 187] \times [-32, 32]$  and  $[63, 127] \times [-16, 16]$ , with coordinates rounded to their integer parts if necessary.

In order to accurately capture the flow characteristics around the cylinder, the center of the cylinder is always placed at  $(80, 0)$ , which is at the middle of the width of the entire computational zone with the cylinder fully located in the finest mesh zone, as seen in Figure 4.1 and Figure 4.2. The cylinder surface is described by 60 evenly distributed Lagrangian points. The LBM computation is restrained with the given free stream at uniform velocity at the left border, while at other borders of the rectangular domain, the equilibrium distribution function is employed as the boundary condition.

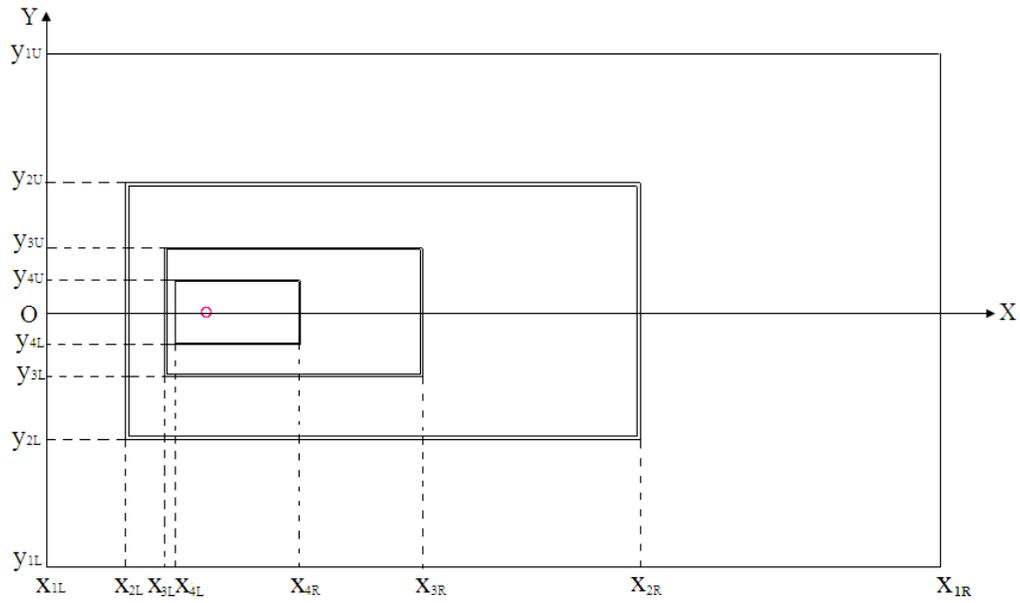


Figure 4.1 Computational domain, setup of zones for multi-level local mesh refinements, and cylinder location

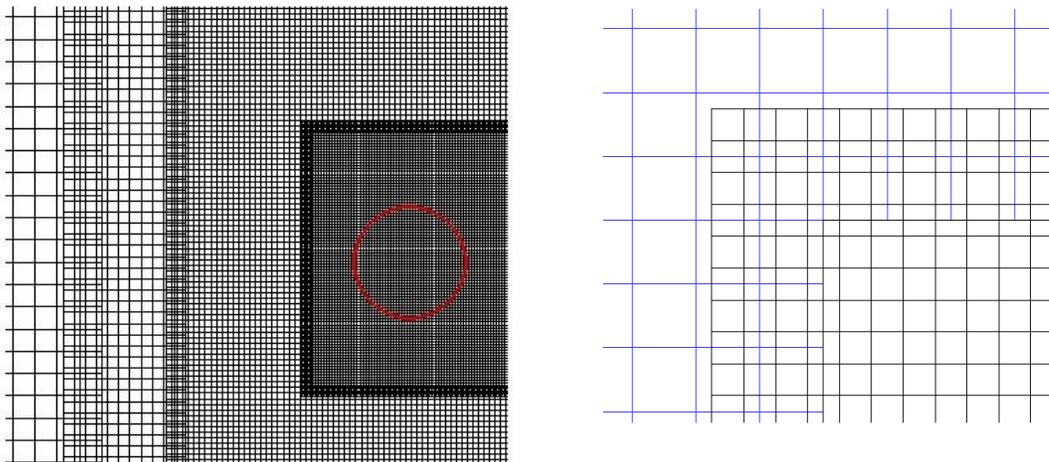


Figure 4.2 Close-up view of the location of the cylinder and the mesh refinement transition

## 4.2. Flow over a single cylinder at $Re=40$

### 4.2.1. Steady flow over a stationary cylinder

The incompressible viscous flow over a stationary cylinder at  $Re=40$  is a classic benchmark case. A priori, this flow features symmetry about the central horizontal line passing through the center of cylinder, which is confirmed again by the present IB-LBM simulations results as shown in Figure 4.3-Figure 4.6 for the distributions of streamline, pressure, horizontal and vertical velocity components, as well as vorticity, respectively. In particular, Figure 4.3 also clearly demonstrates a pair of symmetric recirculating eddies behind the cylinder, while there is no indication of flow penetration across the boundary of the immersed object, confirming the success of incorporation of the IB approach in the LBM framework. Another important remark can be made on the satisfaction of the no-slip boundary condition on the surface of the cylinder in the present IB-LBM model, since Figure 4.5 (b) illustrates the velocity strictly remains zero at the solid boundary.

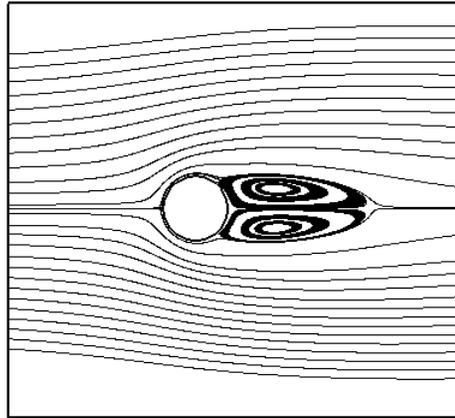


Figure 4.3 Streamline pattern around cylinder at  $Re=40$

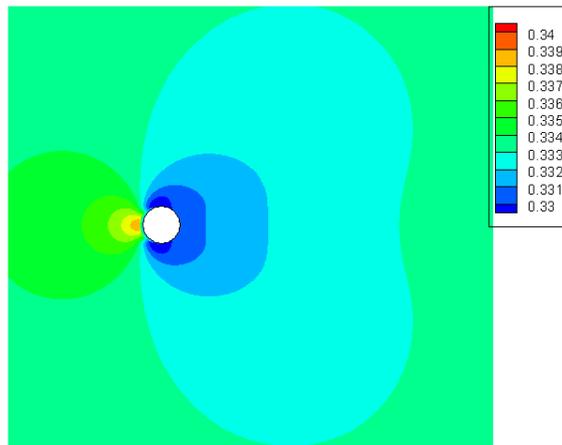


Figure 4.4 Pressure around cylinder at  $Re=40$

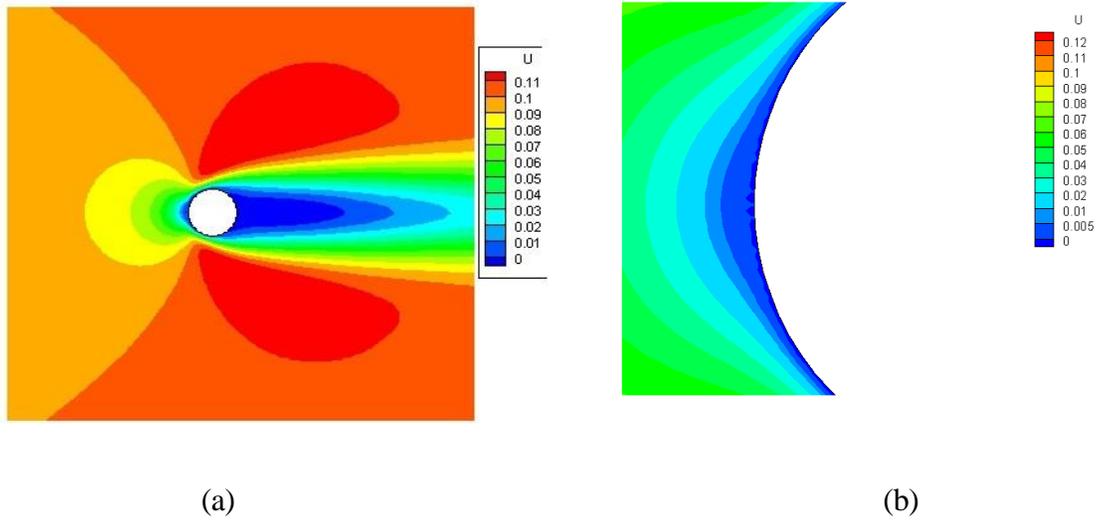


Figure 4.5 Velocity magnitude around cylinder (left) and its close-up near the stagnation point (right) at

$Re=40$

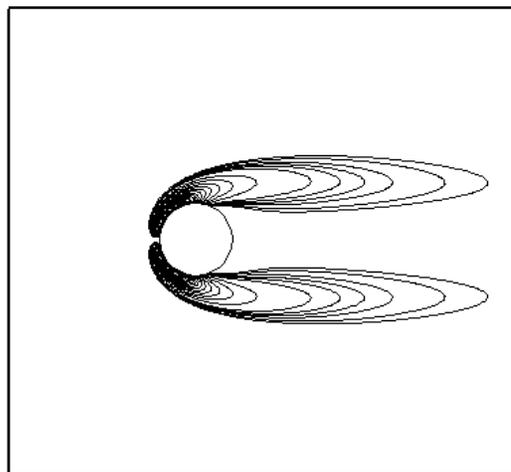
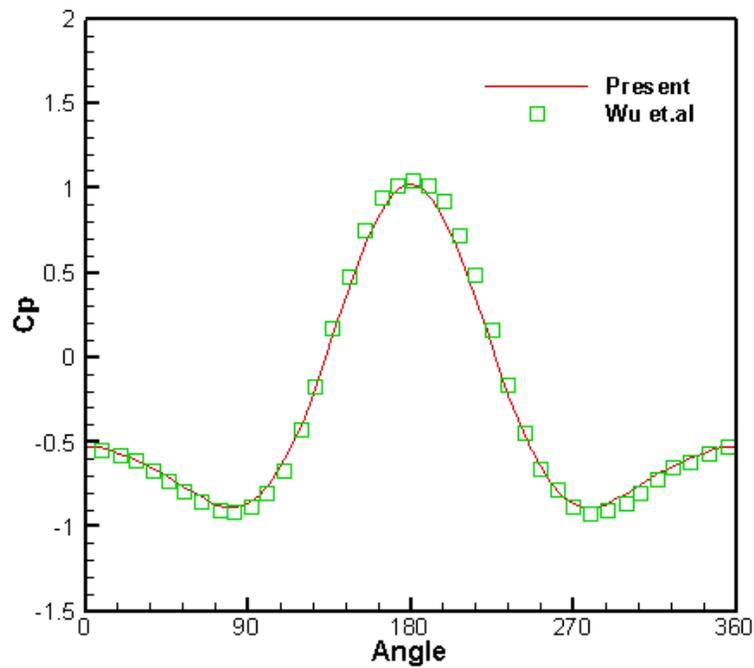


Figure 4.6 Vorticity around cylinder at  $Re=40$

A quantitative comparison can also be made using the pressure coefficient on the surface of cylinder,  $C_p$ , the drag coefficient,  $C_d$ , and the length of recirculation zone,  $L$  (scaled by the diameter of cylinder). Figure 4.7 depicts both the present pressure coefficient distribution result on the entire surface of the cylinder and a reference result [31], which obviously agree with each other. The curve is strictly symmetric about the position corresponding to the  $180^\circ$  angle, at which the stagnation point is formed and the maximum  $C_p$  takes place. At the angles of  $90^\circ$  and  $270^\circ$ , which are respectively the highest and lowest points of the cylinder, the negative pressure reaches its extreme value as the maximum velocity arises locally. Echoing the symmetry exhibited in Figure 4.3 - Figure 4.6, the  $C_p$  distribution remains symmetric about the  $180^\circ$  angle, which represents the horizontal diameter of the cylinder. Table 4.1 groups the drag coefficient and recirculation zone length data obtained using the IB-LBM model and those reported by other literature, indicating the present study can produce reliable numerical results since they satisfactorily match with other reported data.

Figure 4.7 Pressure distribution on surface of cylinder at  $Re=40$ Table 4.1 Comparison of drag coefficient and length of recirculation zone at  $Re=40$ 

References	Drag coefficient $C_d$	Length of recirculation zone $L$
Fornberg[52]	1.498	2.25
Shukla et al.[53]	1.550	2.34
Niu et al.[30]	1.589	2.26
Wu et al.[31]	1.554	2.30
Present	1.493	2.36

### 4.2.2. Steady flow over a moving cylinder

The precedent test case can be modified such that the free stream velocity keeps  $U_\infty = 0.095$  while the cylinder itself also moves towards the incoming constantly at  $U_{object} = -0.005$ , starting from the position from (100, 0). This modification generates a horizontal translating velocity for the cylinder, which creates a scenario of moving boundary and provides a new test case for verifying the effectiveness of the present IB-LBM model in dealing with a flow associated with an immersed moving obstacle. However, compared to the first test case with a stationary cylinder, the modification made here is not expected to alter the flow pattern, since the incoming flow velocity relative to the moving cylinder evidently remains 0.1. Hence, in this modified case, Reynolds number keeps  $Re=40$ .

Unsurprisingly, in this moving boundary case, despite the translation of the cylinder, the flow behavior around the cylinder looks identical and steady if observed from the moving cylinder itself. A representative plot, Figure 4.8, is made using the stream function values obtained using the present model, and exhibits no difference from the streamline distribution in the stationary cylinder case as illustrated in Figure 4.3. Further examination can be performed using the pressure coefficient along the entire surface of the cylinder, leading to a quantitative comparison against other available relevant data as depicted in Figure 4.9. A high degree of agreement is found again, which reveals the capability of the IB-LBM approach for simulation of flow problems with obstacle moving in the fluid and, moreover, confirms the Galilean invariance of the present method that holds independent

of whether the immersed obstacle is in motion or at rest.

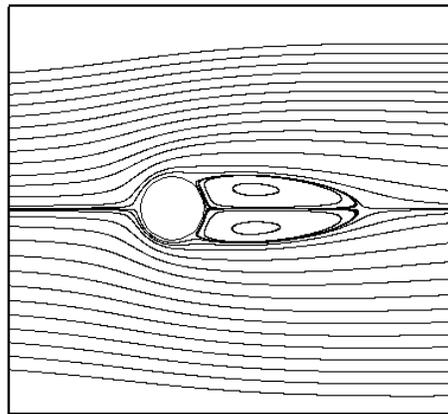


Figure 4.8 Streamline pattern around a moving cylinder at  $Re=40$

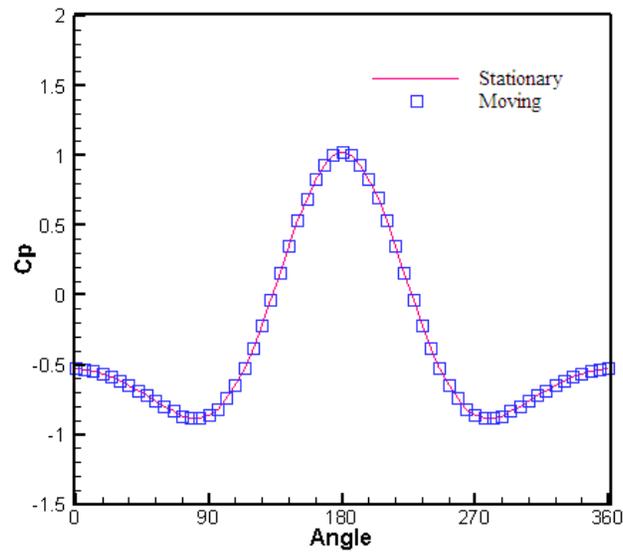


Figure 4.9 Pressure distribution on surface of a moving cylinder at  $Re=40$

### 4.3. Flow over a single cylinder at $Re=100$

#### 4.3.1. Unsteady flow over a stationary cylinder

For simulating the unsteady flow at an elevated Reynolds number  $Re=100$ , the computational domain and the location of the cylinder in the flow field is the same as the first case above. Figure 4.10-Figure 4.12 illustrate the instantaneous streamlines, velocity and vorticity contours in the vicinity of the cylinder, respectively. Karman vortex street has been obviously revealed in Figure 4.12. As seen from the vorticity contours in Figure 4.12, the vortex is shedding at a constant frequency. Here, the Strouhal number is employed to examine the dimensionless frequency when the vortices are shed from the body:

$$St = \frac{f_q D}{U_\infty} \quad (4.8)$$

with  $f_q$  representing the vortex shedding frequency. This frequency can be acquired by a look into the time evolution of the drag and lift coefficients depicted in Figure 4.13. The oscillation of the two coefficients looks periodic, which demonstrates the flow field varies periodically when the vortices are shed from the cylinder. Table 4.2 compares the time-averaged drag coefficient,  $\bar{C}_d$ , and Strouhal number,  $St$ , of this unsteady flow case obtained using the present model to those obtained elsewhere, which shows that the present results are within the range of values reported by the references.

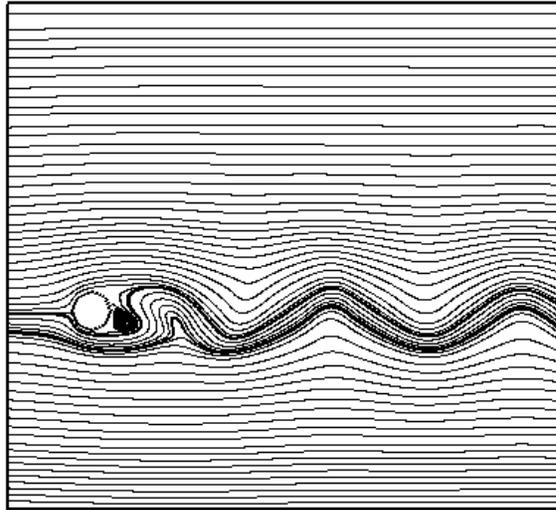


Figure 4.10 Streamline pattern around cylinder at  $Re=100$

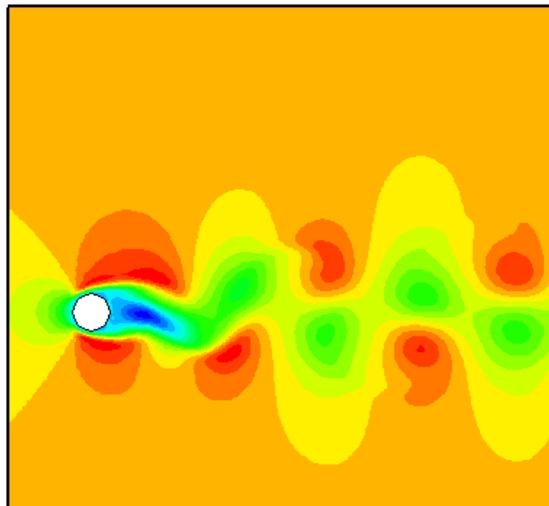


Figure 4.11 Velocity magnitude around cylinder at  $Re=100$

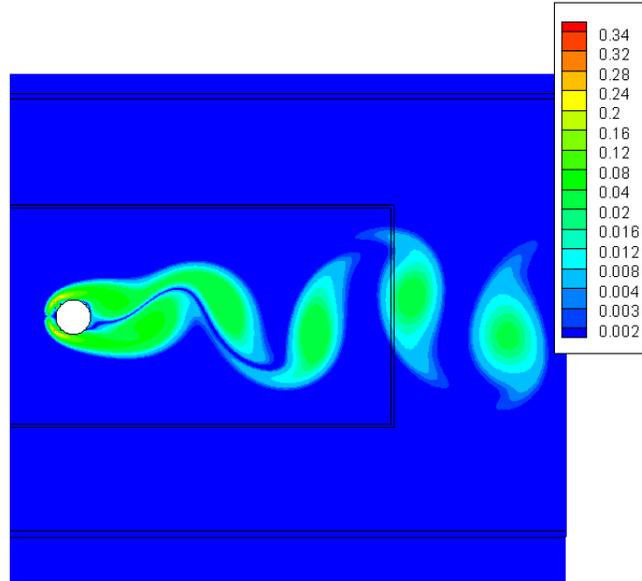


Figure 4.12 Vorticity around cylinder at  $Re=100$

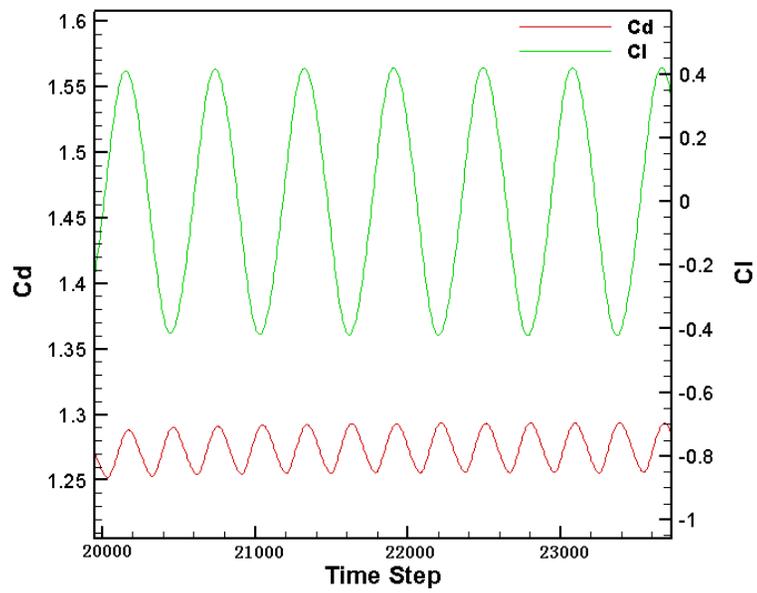


Figure 4.13 Evolution of drag and lift coefficients at  $Re = 100$

Table 4.2 Comparisons of time-averaged drag coefficient and Strouhal number at Re=100

References	Drag Coefficient $\bar{C}_d$	Strouhal number $St$
Wu et al.[31]	1.364	0.163
Saiki et al.[54]	1.26	0.171
Sui et al.[32]	1.438	0.166
Russell et al.[22]	1.43	0.175
Present	1.242	0.167

### 4.3.2. Unsteady flow over a moving cylinder

Similar to the strategy used in Section 4.2 for the selection of test cases at Re=40, in the case of Re=100 here, the free incoming stream moves at  $U_\infty = 0.095$  while the cylinder itself is allowed to be horizontally migrating from (100, 0) with a constant velocity  $U_{object} = -0.005$  in the opposite direction to the free stream, and all other conditions remain unchanged. The effect of the modification made now is that the obstacle is in motion in lieu of at rest, while the Reynolds number Re=100 is maintained as the velocity of the incoming flow is still 0.1 with respect to the cylinder. Hence, the local flow pattern observed at a watching station adhered to the cylinder is expected to be identical to what was obtained in Section 4.3.1. Figure 4.14 contains two different instantaneous view of the vorticity contours, illustrating both the motion of the cylinder itself and the evolution of the Karman vortex street; moreover, as anticipated, the right part of Figure 4.14 looks identical to

Figure 4.12 where the cylinder was set stationary. Figure 4.15 shows the time evolution of drag and lift coefficients, indicating the vortex shedding frequency remains constant value. A careful comparison between Figure 4.15 and Figure 4.13 for the  $Re=100$  cases of moving and stationary cylinders, respectively, reveals that the two cases share the same vortex shedding frequency.

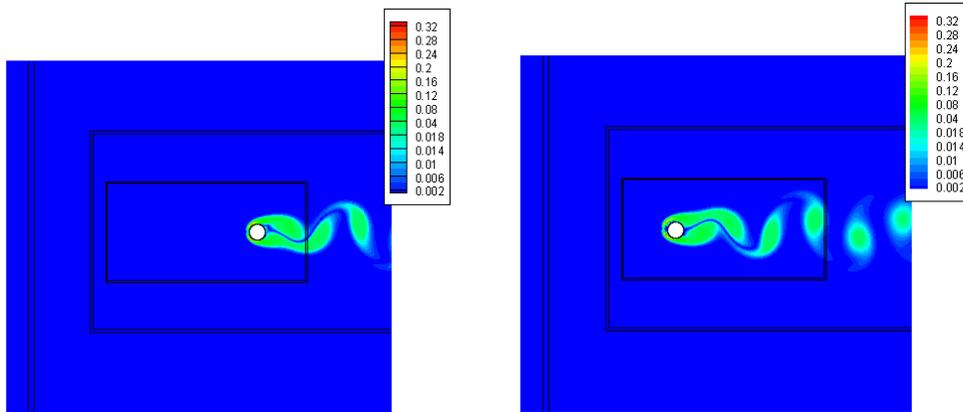


Figure 4.14 Two instantaneous distributions of vorticity around a moving cylinder at  $Re=100$

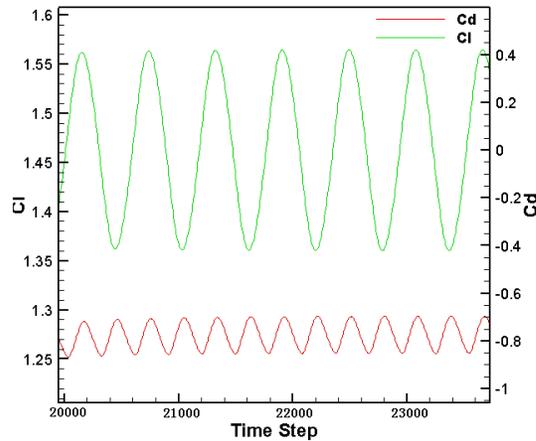


Figure 4.15 Evolution of drag and lift coefficients at  $Re = 100$  in the case of moving cylinder

#### 4.4. Flow over two stationary vertically aligned and gapped cylinders

The IB-LBM model coupled with the multi-level mesh refinement algorithm has been successfully applied in simulating both steady and unsteady flows over a single cylinder in both stationary and moving cases in Sections 4.2 and 4.3. Now, the effectiveness of the present approach is tested using more challenging cases with flow scenarios enriched by two obstructing cylinders, which are stationary and vertically aligned and gapped in this section. Hence, not only are such flows characterized by Reynolds number as previously defined by Eqn.(4.9), but also by the vertical gap depth,  $g$ , (see Figure 4.16 ), as this parameter plays an important role in the interaction between the two cylinders. In practice, the non-dimensional gap space with respect to the cylinder diameter,  $D$ , i.e.,  $g^* = g/D$ , is employed to differentiate the cases in this type of flows. The influence of  $Re$  and  $g^*$  on such flows has been investigated by Kang [55]. It was found from previous references that, when  $g^* \geq 5$ , there is almost no mutual interference between the two vertically aligned cylinders, and the two vortices arise behind the two cylinders separately with no merging indication. Therefore, this study mainly looks into the more sensitive range of  $0.2 \leq g^* \leq 3$  on anticipation of distinctive variation of flow patterns with altered gap spaces.

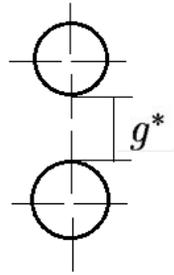


Figure 4.16 Schematic of two vertically aligned cylinders

#### 4.4.1. $Re=40$ , $g^* = 1$

Recall that the test case provided in Section 4.2.1 has shown the flow is strictly symmetric when a single stationary cylinder is immersed in a fluid that flows with  $Re=40$ , and the resulting flow pattern features two fully large symmetric eddies that forms a recirculation zone behind the single cylinder, as seen in Figure 4.3. Extended from that case, the obstacles now become two vertically aligned cylinders with a gap of  $g^* = 1$ . The details to be explored for this new flow phenomena are whether the local recirculation zone behind each of the two cylinders remains symmetric and, furthermore, whether the two local recirculation zones also appear symmetric with each other.

Using the IB-LBM model, Figure 4.17 unveils the flow behavior in this bi-cylinder case. It can be observed that the fluid is now allowed to flow across the gap area between the two obstructing cylinders and, thus, serves to push a bit away the two local recirculation zones, which are respectively generated behind the two cylinders. This pushing action turns each local recirculation zone to a deformed shape, which is now no longer symmetric. However, from the

global view of the flow pattern, the two local recirculation zones remain symmetric. There are two conjunctive reasons leading to this globally symmetric flow. One reason is that the Reynolds number in this case reaches 40 only, which is not elevated at all. The other factor is that the gap is relatively large ( $g^* = 1$ ) as the gap itself is comparable to the characteristic length of this flow, which is the diameter of the cylinder, yielding enough gap flow space so that the upper and lower parts of the flow have little occasion of mutual intervention. In consistency with the global symmetry exhibited in Figure 4.17, Figure 4.18 reveals that the drag and lift coefficients of the two cylinders remain nearly unchanged with time. All the present IB-LBM results of the flow around two vertically aligned cylinders at  $g^* = 1$  and  $Re=40$  confirms the so called "*steady weak flow pattern*", as discovered in [55].

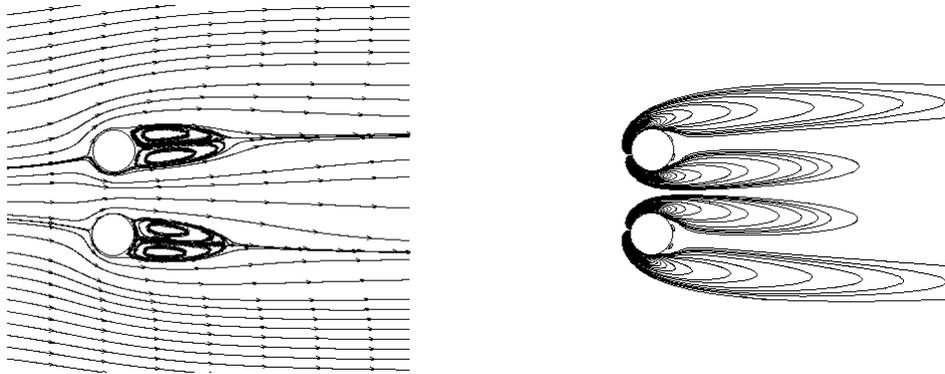


Figure 4.17 Instantaneous streamline and vorticity contours at  $Re=40$ ,  $g^* = 1$

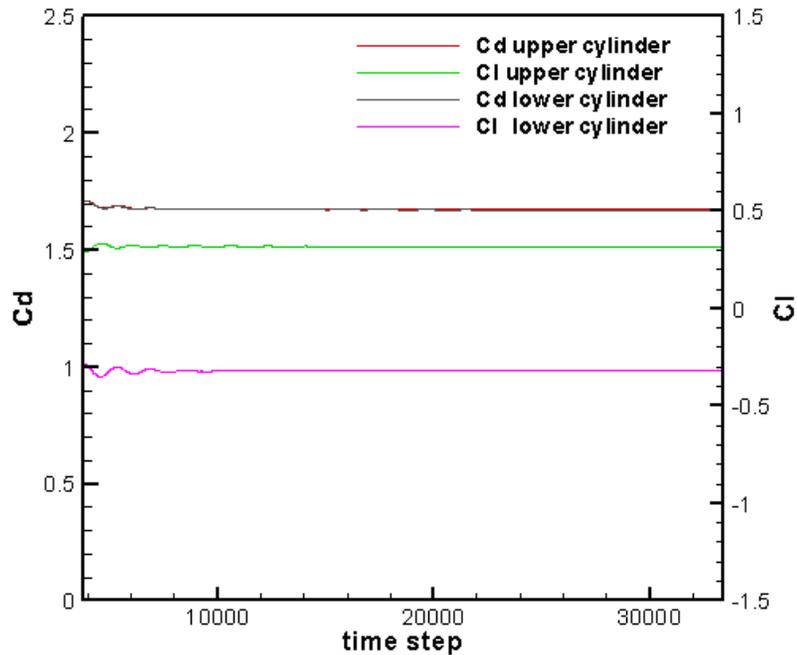


Figure 4.18 Evolution of drag and lift coefficients for two cylinders at  $Re=40$ ,  $g^* = 1$

#### 4.4.2. $Re=70$ , $g^* = 0.5$

Based on the analysis performed for the flow over two vertically aligned and gaped cylinders with  $Re=40$  and  $g^* = 1$  in the precedent section, here the Reynolds number is increased to  $Re=70$  while the gap is decreased to  $g^* = 0.5$ , so that an unsteady and asymmetric flow pattern can be anticipated. Effectively, due to the two altered parameters, the streamlines and vorticity contours obtained by the present numerical method no longer look symmetric, as demonstrated in Figure 4.19. On the other hand, Figure 4.20 shows the time evolution of the drag coefficients for the two cylinders are no longer flat, indicating the flow now becomes unsteady; moreover, unlike the case of

$Re=40$  and  $g^* = 1$  where the drags on the upper and lower cylinders are the same with each other (see Figure 4.18), now the drag coefficient of the lower cylinder is remarkably higher than that of the upper one (see Figure 4.20) as iteration time elapses. This suggests that the flow pattern is deflected to the side of higher drag coefficient cylinder, i.e., the lower cylinder, such that the eddy behind the lower cylinder is larger than the other eddy, as seen in Figure 4.19. The flow deflection phenomenon observed in the present study echoes the so called "*deflected wake pattern*", as described in [55].

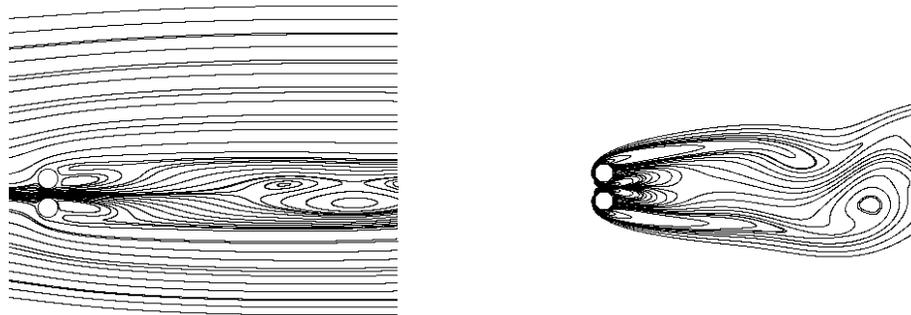


Figure 4.19 Instantaneous streamline and vorticity contours at  $Re=70$ ,  $g^* = 0.5$

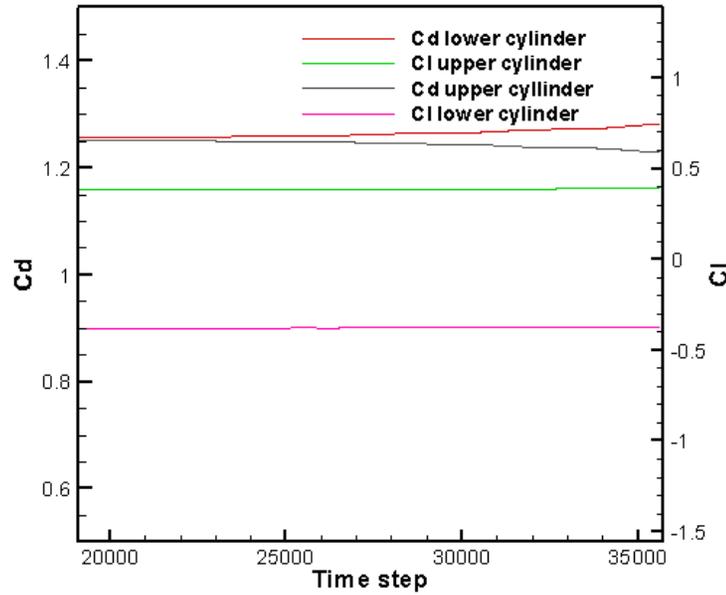


Figure 4.20 Evolution of drag and lift coefficients for two cylinders at  $Re=70$ ,  $g^* = 0.5$

#### 4.4.3. $Re=100$ , $g^* = 0.2, 0.7, 1.5, 3$

For the Reynolds number of  $Re=100$ , the flow is influenced heavily by the gap depth of the two cylinders. A series of test cases with  $g^* = 0.2, 0.7, 1.5$ , and  $3$  have been investigated in [55], and the resulting flow patterns are referred to as "*single bluff-body pattern*", "*flip-flopping pattern*", "*in-phase-synchronized pattern*" and "*anti-phase-synchronized wake pattern*", respectively. In this IB-LBM study, these four representative cases are re-simulated to examine the capability of the present method.

When both cylinders are placed very close with each other, e.g.,  $g^* = 0.2$ , the flow within the gap becomes too weak to affect the flow pattern in the whole domain. The flow behaves as if a

single bluff-body with a characteristic length of twice as long as the diameter of a single cylinder. A street of vortices shed behind of the side-by-side cylinders, and the flow pattern becomes periodic as depicted in Figure 4.21 and Figure 4.22. Thus, the structure of this flow is named "*single bluff-body wake pattern*".

After further increasing the gap spacing to  $g^* = 0.7$ , the flow is no longer periodic but appears unsteady (see Figure 4.23). Both drag and lift coefficients undergo irregular changes as observed in Figure 4.24. Previous investigation [55] has shown that the flow between the two cylinders deflects to the side of higher-frequency cylinder, and this cylinder sustains a higher drag coefficient than the other. Moreover, both cylinders take irregular turns in experiencing the higher frequency, which indicates that the two wakes randomly flip-flop behind the cylinders. Thus, the flow structure is called "*flip-flopping wake pattern*".

In the case of  $g^* = 1.5$ , the two wakes finally become synchronized (see Figure 4.25 and Figure 4.26). The lift coefficients for both cylinders are in phase, however the drag coefficients are out of phase. Therefore, the flow structure is named "*in-phase-synchronized wake pattern*". The vortices shedding at both cylinders are in nearly the same phase with vortices tending to merge on a pair-by-pair basis.

At  $g^* = 3$ , the flow becomes periodic once more, and the two single-cylinder wakes also become synchronized but in a different way compared with  $g^* = 1.5$  case. According to the drag and force coefficients depicted in Figure 4.28, the drag coefficients for the two cylinders are almost the same but the lift coefficients appear in anti-phase. Thus, the flow structure is called

"*anti-phase-synchronized wake pattern*". Besides, the streamline shown in Figure 4.27 demonstrates that the upper and lower cylinders possess a good symmetry in the case of  $g^* = 3$ .

Using the IB-LBM approach, four different flow patterns are all precisely observed, and the details of the present results, including the streamline, vorticity contour, drag and lift coefficients, agree in a satisfactory manner with those reported by [55]. This justifies once more the validity and reliability of the IB-LBM model developed in this study.

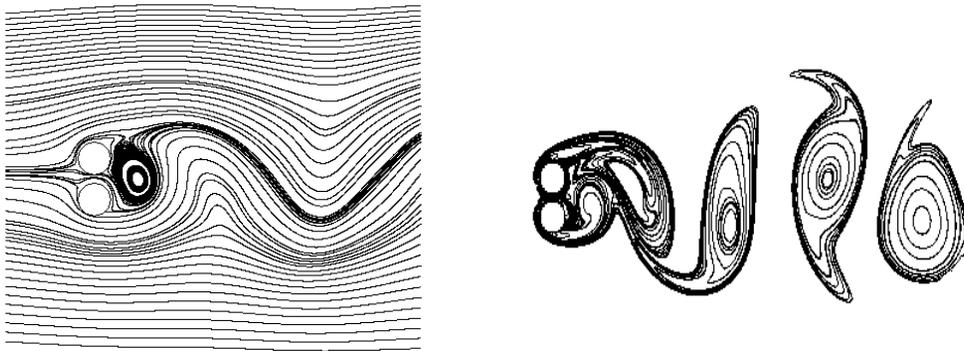


Figure 4.21 Instantaneous streamline and vorticity contours at  $Re=100$ ,  $g^* = 0.2$

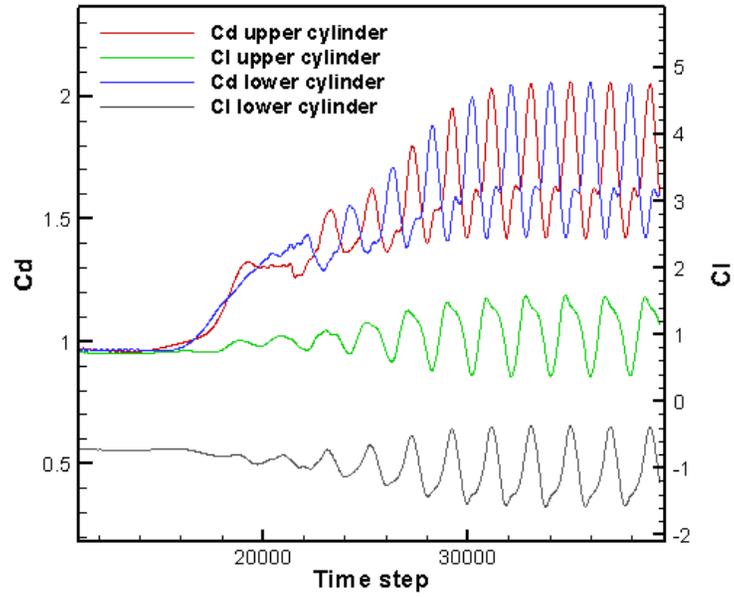


Figure 4.22 Evolution of drag and lift coefficients for two cylinders at  $Re=100$ ,  $g^* = 0.2$

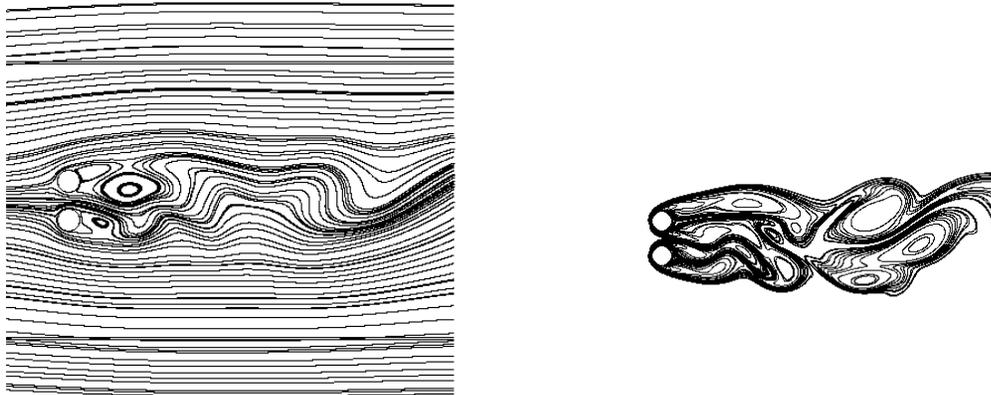


Figure 4.23 Instantaneous streamline and vorticity contours at  $Re=100$ ,  $g^* = 0.7$

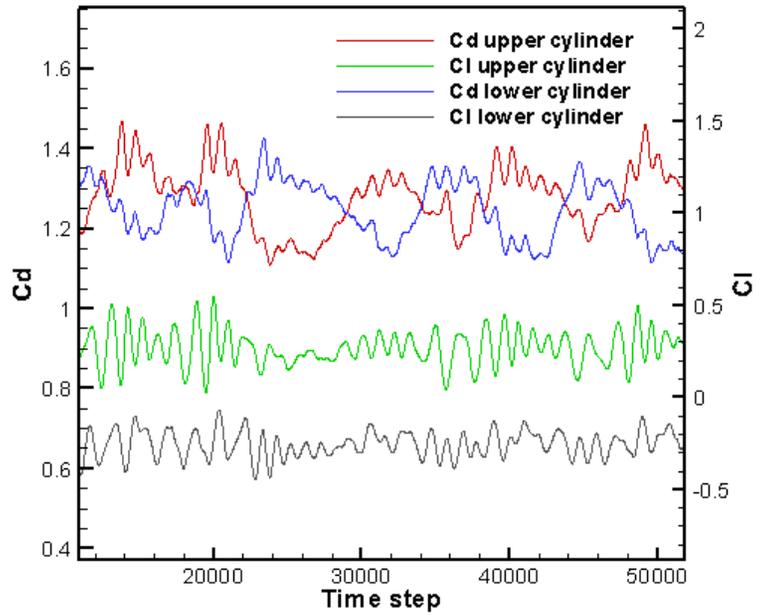


Figure 4.24 Evolution of drag and lift coefficients for two cylinders at  $Re=100$ ,  $g^* = 0.7$

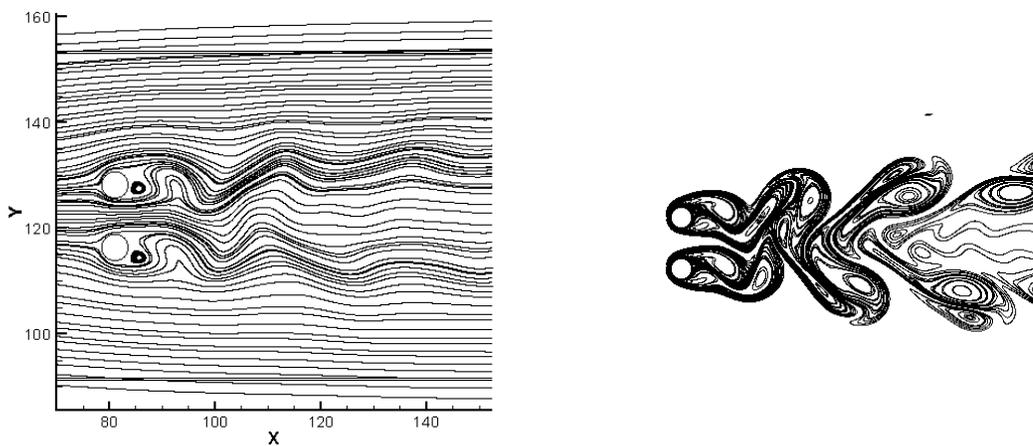


Figure 4.25 Instantaneous streamline and vorticity contours at  $Re=100$ ,  $g^* = 1.5$

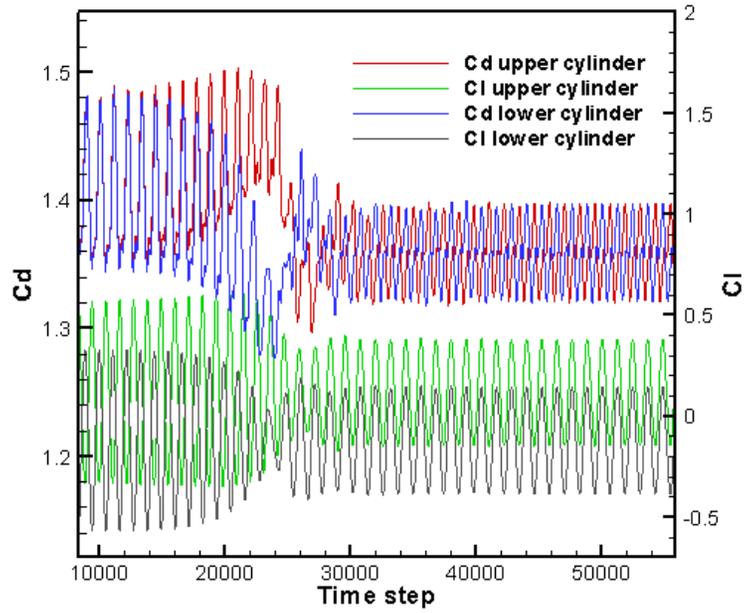


Figure 4.26 Evolution of drag and lift coefficients for two cylinders at  $Re=100$ ,  $g^* = 1.5$

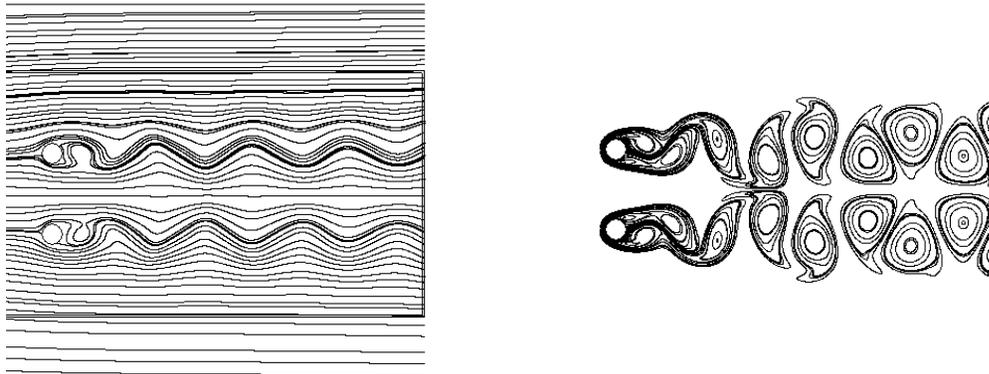


Figure 4.27 Instantaneous streamline and vorticity contours at  $Re=100$ ,  $g^* = 3$

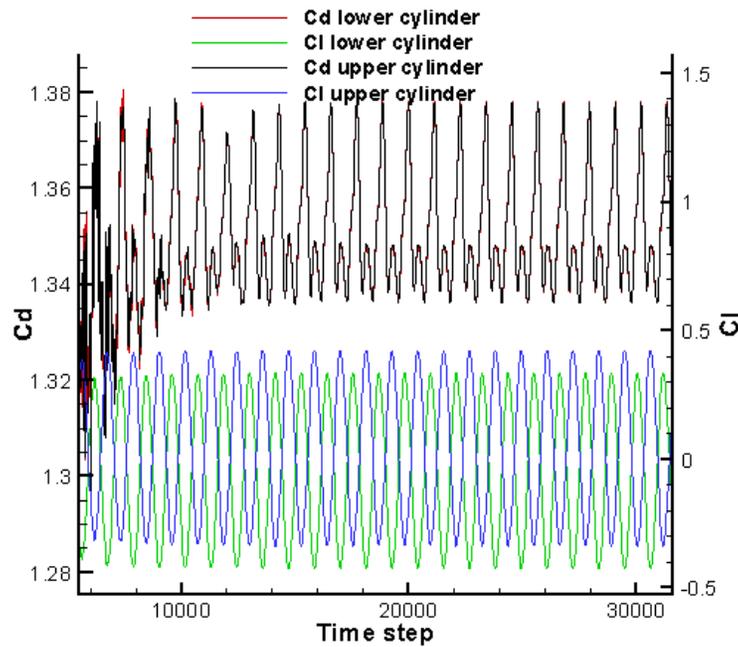


Figure 4.28 Evolution of drag and lift coefficients for two cylinders at  $Re=100$ ,  $g^* = 3$

#### 4.5. Flow over two moving cylinders

Unlike the test cases in the precedent section where two cylinders keep stationary, this section examines the flow associated with two moving cylinders that are immersed in the flowing fluid, so as to demonstrate the effectiveness of the present IB-LBM model for flow simulation in the case of multiple moving objects immersed in the fluid.

As sketched in Figure 4.29, the two cylinders with an identical diameter of 5 are away from each other at a horizontal distance of 60 at the starting time point, and the vertical distance between the centers of the two is fixed at 7.5, which is three times of the radius of the two cylinders. Both

cylinders are allowed only for horizontal motion with the same speed,  $U=0.025$ , but opposite direction. The two cylinders are both using the highest grid resolution, which covers a sufficiently long zone to accommodate the motion of the two cylinders.

Figure 4.30 shows the vorticity contour when the two cylinders are the closest to each other, and Figure 4.31 plots the vorticity contour when the horizon distance between the two cylinders turns to 60 again after they move apart. The evolution of drag and lift coefficients of the upper cylinder is shown in Figure 4.32 and Figure 4.33, respectively. Figure 4.32 shows that the drag force tends to slightly increase when the two cylinders are moving closer to each other, then steeply decreases as they pass in close proximity, but steeply bounces back when the continuous motion renders them apart from each other, and at the end reaches its original level. The restoration trend also arises in Figure 4.33 for the lift force. The variation of aerodynamic forces with the motion in this case is compared to the results obtained in [39] with the same observation duration employed for both numerical experiments, and satisfactory agreement is achieved in this quantitative comparison.

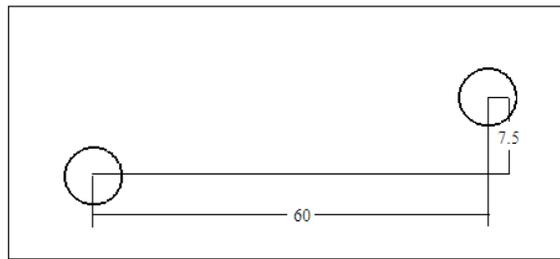


Figure 4.29 Computational domain for two cylinders moving with towards each other

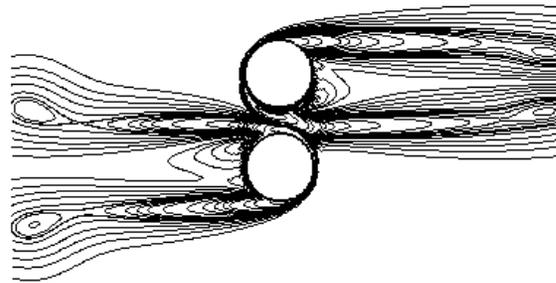


Figure 4.30 Vorticity contour when two cylinders are the closest

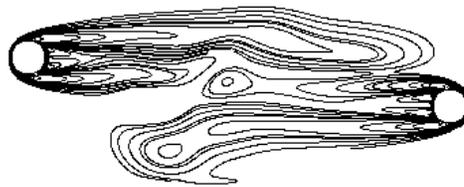


Figure 4.31 Vorticity contour when two cylinders are distanced at 60

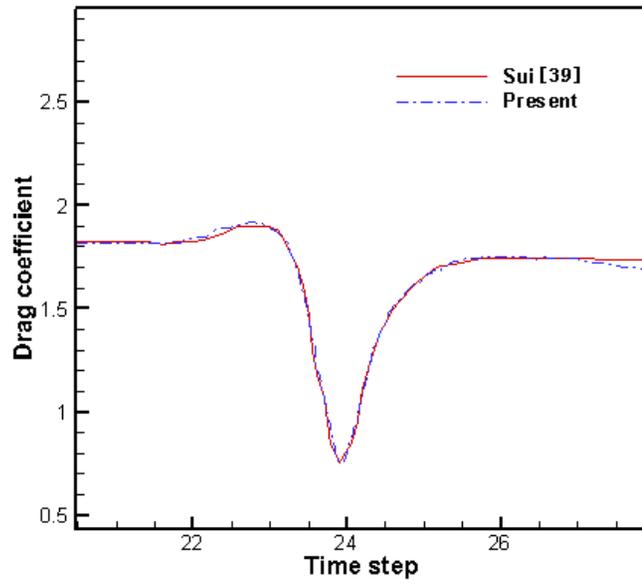


Figure 4.32 Evolution of drag coefficient for the upper cylinder

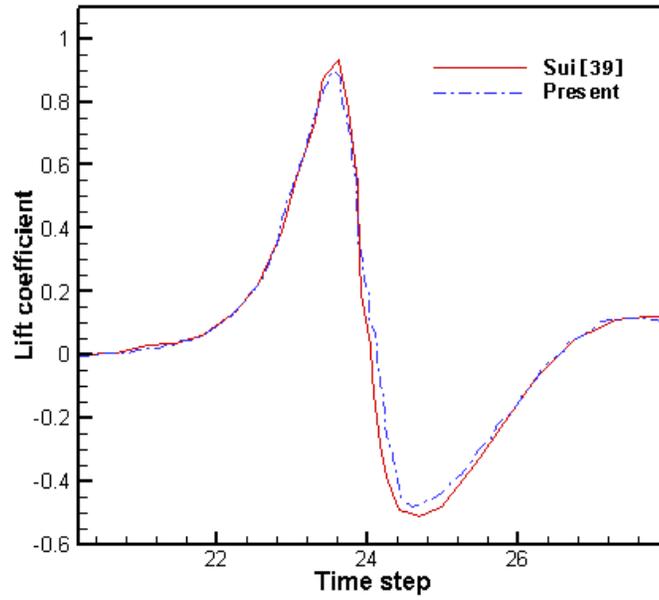


Figure 4.33 Evolution of lift coefficient for the upper cylinder

#### 4.6. Local mesh refinement sensitivity

When dealing with flow around immersed object problems using local mesh refinement algorithms, it is critical to control the position of the immersed solid object relative to the refined zone. If the immersed object is too close to the boundary of refined region, the computation accuracy may be affected by the interpolation error arising from the interface strip between the refined zone and its neighborhood where the mesh is not refined. On the other hand, if the boundary of refined region is too far away from the object, the computing efficiency will be definitely reduced due to the unnecessarily large mesh size. To assess the sensitivity of local mesh refinement to the numerical solution accuracy, a series of tests with differently chosen refinement zones will be performed corresponding to the two cases of a stationary cylinder immersed in a fluid flowing at  $Re=40$  and  $100$ , respectively, as previously investigated in Sections 4.2.1 and 4.3.1 using the IB-LBM model

##### 4.6.1. Stationary cylinder ( $Re=40$ ) with different horizontal refinement zones

Using different lengths for the finest mesh zone located behind a cylinder immersed in a flowing fluid at  $Re=40$ , a series of tests is performed to examine the sensitivity of local mesh refinement to the IB-LBM solution accuracy. As shown in Table 4.3, the finest grid resolution zone are deployed with varying length but the width of the zone is locked at a reasonable value, since in this case the length of the recirculation zone is of numerical investigation, and has been detected to be approximately to  $x = 110$  according to Section 4.2.1 of this study.

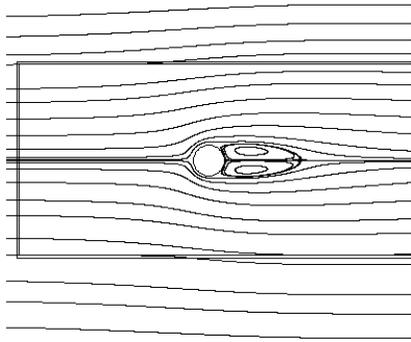
The streamlines around the cylinder for all six differently refined meshes are grouped in Figure 4.34, behind each of which the rectangle frame represents the highest resolution zone. Although the streamlines in Figure 4.34 are not apparently distinguished from each other, a more careful examination on the drag coefficient and recirculation zone length can identify the sensitivity of local mesh refinement in this case, as shown in Table 4.4. When the boundary of the finest grid region is close to the immersed object, for example, in case 4, 5, 6, the pair of symmetric recirculation bubbles behind the stationary cylinder penetrate the interface of the finest and second finest grid zones, as seen in the (d), (e), and (f) parts of Figure 4.34; thus, different grid resolutions employed on the sites of identical physical sensitivity, which results in some numerical errors. Therefore, the drag coefficients and recirculation lengths detected in Case 4, 5, 6 more noticeably deviate from the baseline Case 3. On the other hand, when the refinement zone enlarges to be beyond the recirculation zone, the drag coefficient and recirculation length remain almost unchanged, but the computation has to take more time as a result of increased mesh size. This series of tests reveals that a pertinently refined mesh is a key to the optimized balance between the solution accuracy and cost.

Table 4.3 Six cases using different zones of finest grid resolution for  $Re=40$ 

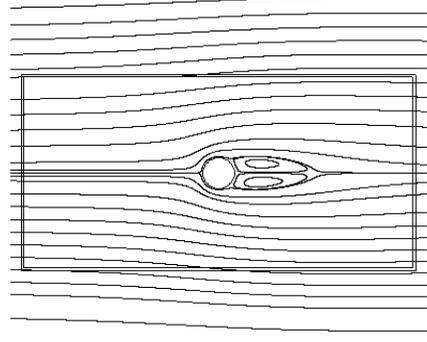
Case	Zone with finest grid resolution	Refinement category
Case1	$[63, 158] \times [-16, 16]$	excessive refinement
Case2	$[63, 127] \times [-16, 16]$	excessive refinement
Case3	$[63, 109.5] \times [-16, 16]$	proper refinement
Case4	$[63, 107] \times [-16, 16]$	insufficient refinement
Case5	$[63, 104.5] \times [-16, 16]$	insufficient refinement
Case6	$[63, 102] \times [-16, 16]$	insufficient refinement

Table 4.4 Comparison of drag coefficient and recirculation length for  $Re=40$ 

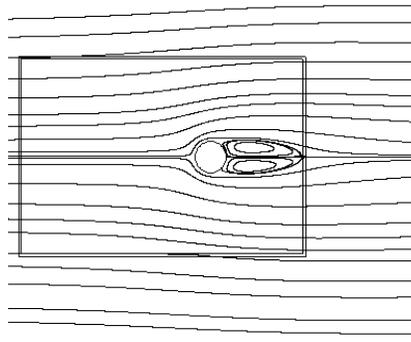
Case	Drag coefficient $C_d$	Length of detected recirculation zone $L$
Case1	1.46732	2.398
Case2	1.46738	2.398
Case3	1.46804	2.397
Case4	1.46815	2.394
Case5	1.46887	2.390
Case6	1.47128	2.382



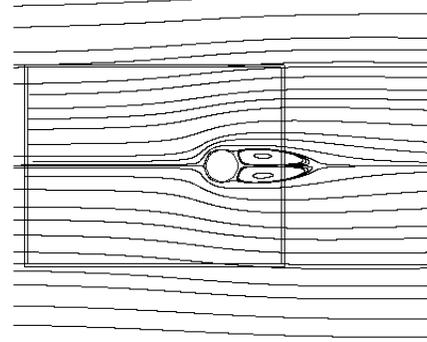
(a) Case 1



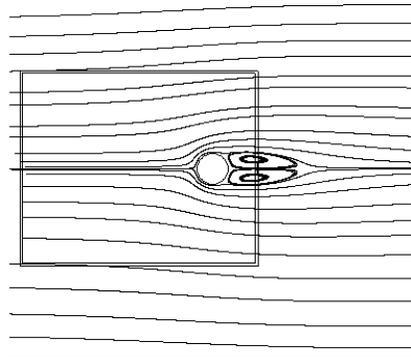
(b) Case 2



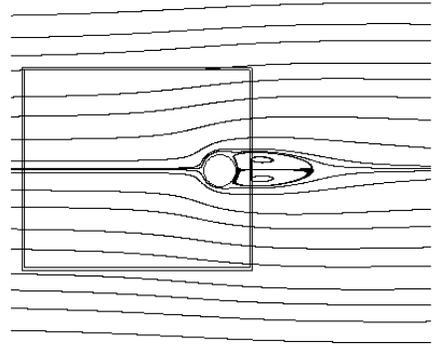
(c) Case 3



(d) Case 4

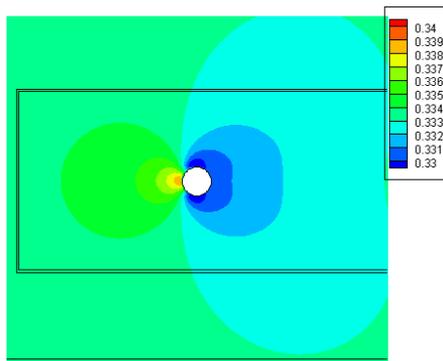


(e) Case 5

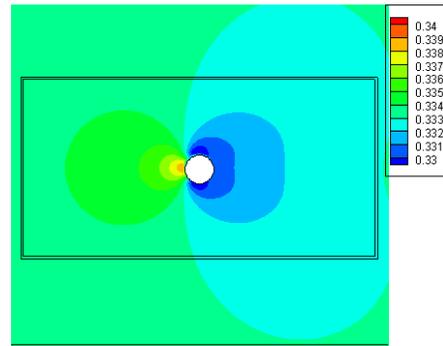


(f) Case 6

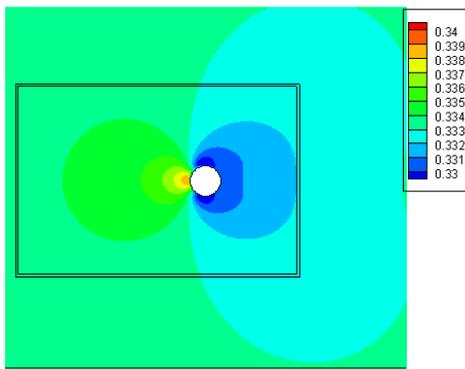
Figure 4.34 Streamline patterns around cylinder at  $Re=40$  in six cases



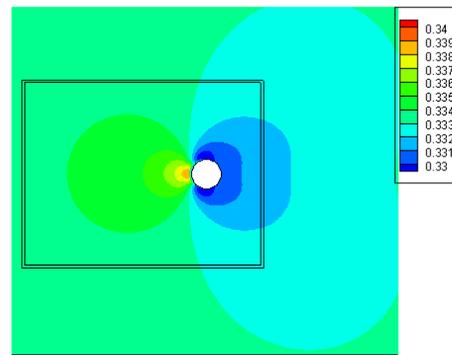
(a) Case 1



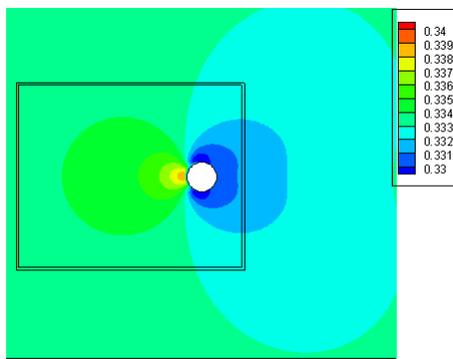
(b) Case 2



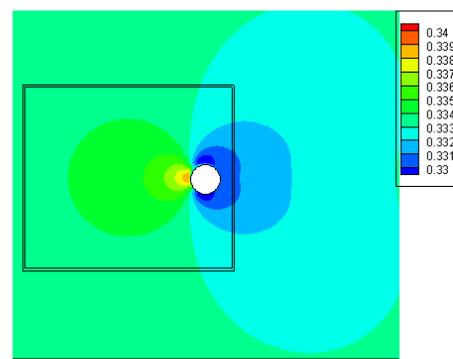
(c) Case 3



(d) Case 4



(e) Case 5



(f) Case 6

Figure 4.35 Pressure contours at  $Re=40$  in six cases

#### 4.6.2. Stationary cylinder (Re=40) with adaptive mesh refinement

In this section, AMR technique is coupled with the IB-LBM model to simulate once more the case of Re=40. According to the results of the same case in the previous sections, the variation of the velocity in the re-circulation region is found the strongest, the criterion,  $\varepsilon$ , used to guide the mesh adaptation for this case is defined using the gradients of the velocity:

$$\varepsilon = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2} \quad (4.10)$$

where  $u$  and  $v$  are the velocity components in the  $x$ - and  $y$ -directions, respectively. Local mesh refinement takes place wherever  $\varepsilon > \varepsilon_{average}$ .

Figure 4.36 shows the final adaptively refined mesh, in which the local refinement is no longer restrained in rectangular zones. The finest grid resolution appear in the proximity of the cylinder as well as in the recirculation zone. The resulting pressure distribution (in colored background) and streamline pattern (in black lines) are both shown in Figure 4.37. Both figures generally feature symmetric distribution, confirming the nature of symmetry in this low Reynolds number flow. The drag coefficient and length of recirculation zone are grouped in Table 4.5 with comparison between the results of AMR, non-AMR, and other available references. Overall, the drag coefficient and recirculation zone length obtained by the AMR-IB-LBM approach are both closer to the references than the results produced by the IB-LBM model with the aid of imposed local mesh refinements. At the end, Figure 4.38 groups the pressure coefficient results obtained by the IB-LBM model coupled with AMR and imposed local mesh refinement technique, respectively,

and their agreement looks remarkable.

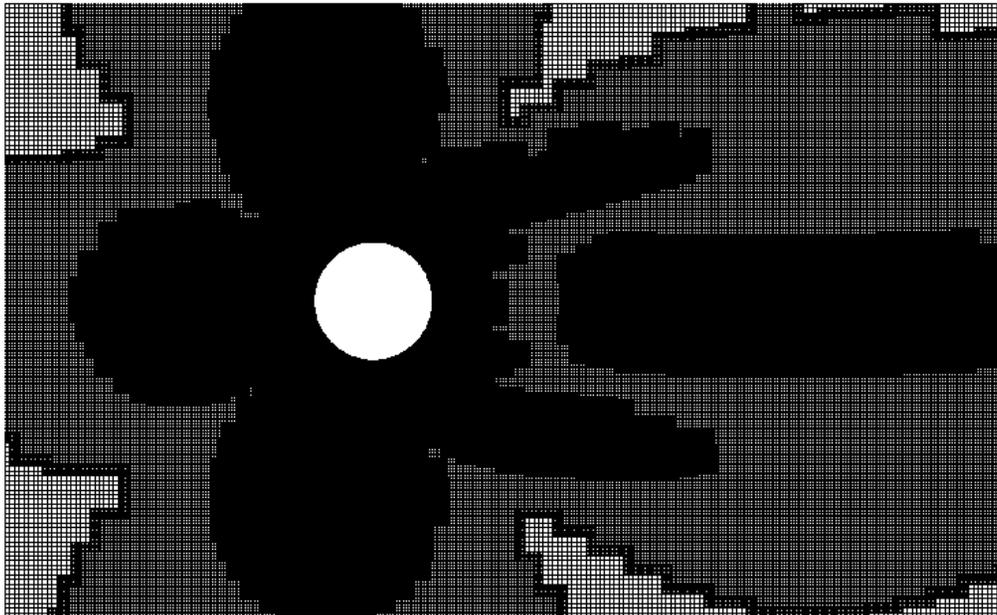


Figure 4.36 AMR mesh at  $Re=40$

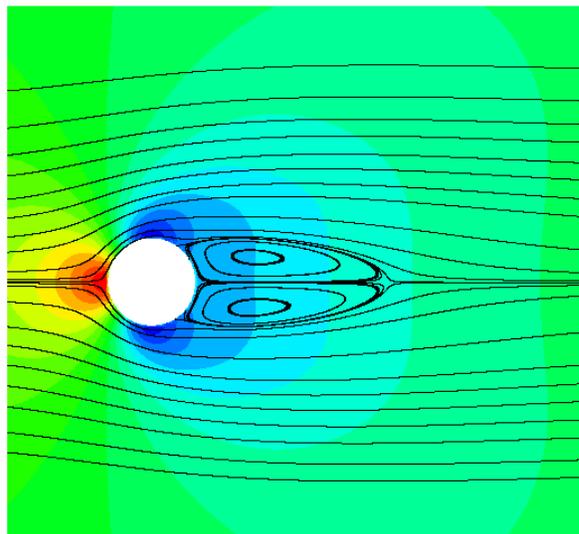


Figure 4.37 AMR results of streamlines and pressure distribution

Table 4.5 Comparison of drag coefficients and length of recirculation zone at  $Re=40$ 

References	Drag coefficient $C_d$	Length of detected recirculation zone $L$
Fornberg[52]	1.498	2.25
Shukla et al.[53]	1.550	2.34
Niu et al.[30]	1.589	2.26
Wu et al.[31]	1.554	2.30
Imposed mesh refinement (Sec.4.2.1)	1.493	2.36
AMR	1.556	2.30

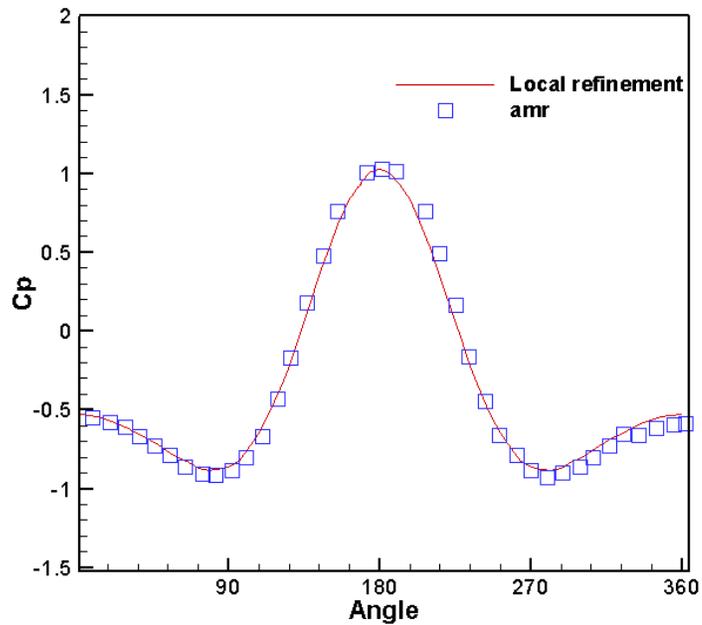


Figure 4.38 Comparison of pressure coefficient

### 4.6.3. Moving cylinder (Re=100) with different vertical refinement zones

After a detailed study of the case of Re=40 using different horizontal refinement zones in Section 4.6.1, the sensitivity of local mesh refinement to the IB-LBM solution accuracy is examined here using different vertical refinement zones for the case of Re=100 with a moving cylinder. As in Section 4.3.2, the free stream flows at  $U_\infty = 0.095$ , while the cylinder moves toward the free stream at  $U_{object} = -0.005$  starting from (100, 0). Figure 4.39 illustrates the four refined meshes be tested in this section, which employ the finest grid resolution for a fixed-length zone, [63, 127], with different widths, 32, 24, 16, and 8, denoted by Cases 1, 2, 3, and 4, respectively. Note that the first case here is the same as the test performed in Section 4.3.2 in both mesh refinement setup and motion scenario; hence, Case 1 can be considered as the baseline case.

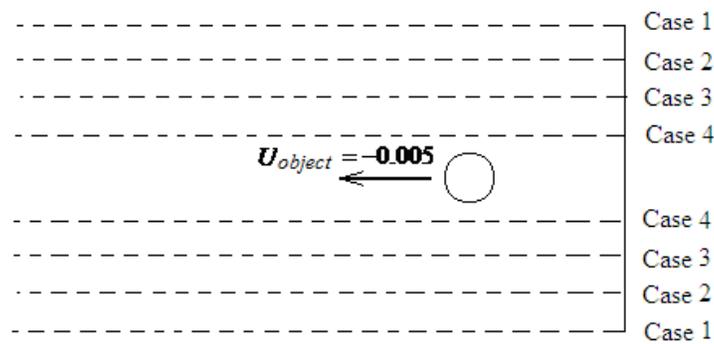


Figure 4.39 Zone of finest grid resolution in the four cases of a moving cylinder for Re=100

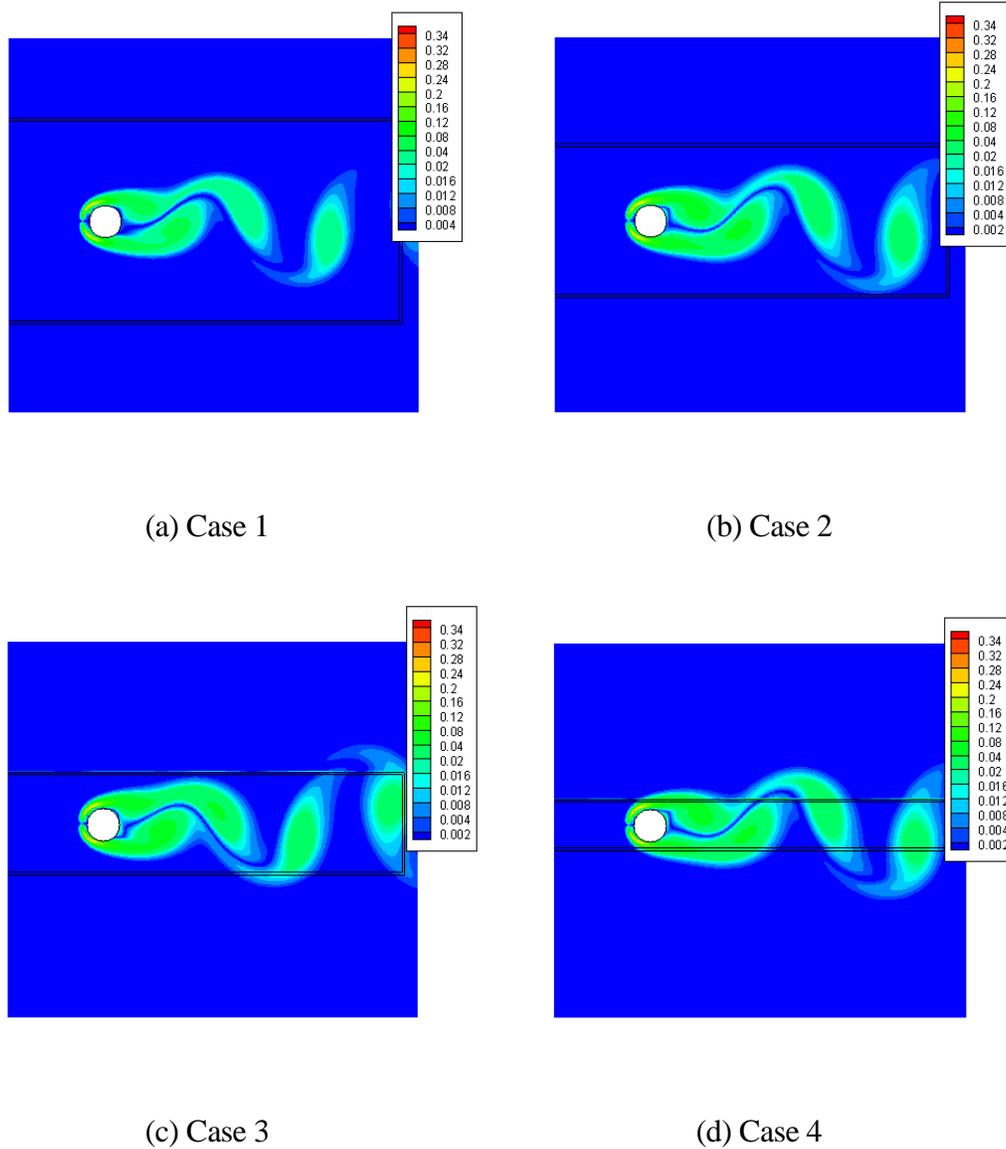
Figure 4.40 Vorticity contours at  $Re=100$  in four cases

Figure 4.40(a) to Figure 4.40(d) show the vorticity contours in the four cases corresponding to an identical time point, and reveal some slight difference between each other. As a more quantitative comparison, Table 4.6 groups, for the four sets of drag coefficient results obtained using the present IB-LBM model. The drag coefficient decreases slightly when the width of the finest level grid

changes from 32 to 16; however, when the width is gradually close to the cylinder, for example, in Case 4, the change of the drag coefficient becomes more obvious. Hence, the width of the finest refinement region plays an important role in ensuring the numerical solution accuracy in this case with Karman vortex street. It is thus strongly recommended that the finest grid resolution should apply to a region that can securely cover the entire obstacle with sufficient surrounding marginal area.

Table 4.6 Comparison of drag coefficient for Re=100

Case	Range of drag coefficient $C_d$
Case1	1.26288-1.28787
Case2	1.26393-1.28853
Case3	1.26591-1.28946
Case4	1.25288-1.27131

#### 4.6.4. Moving cylinder (Re=100) with adaptive mesh refinement

As extension of the precedent section, the moving cylinder case for Re=100 is now tested again using the strategy of coupling the IB-LBM model with AMR techniques, so that its effectiveness for unsteady flow simulation can be assessed. Since the vortex shedding take place in this case, the magnitude of the vorticity is employed as the AMR criterion:

$$\varepsilon = \sqrt{\left(\frac{\partial \omega}{\partial x}\right)^2 + \left(\frac{\partial \omega}{\partial y}\right)^2} \quad (4.11)$$

where  $\omega$  is the nodewise vorticity. If  $\varepsilon > \varepsilon_{average}$  at a node, its neighboring cells will be locally

refined.

Figure 4.41 and Figure 4.42 shows the final adaptive mesh and the corresponding vorticity contours, respectively. As predicted, for the area behind the moving cylinder, the closer to the obstacle, the larger vorticity, and the denser mesh. A detailed comparison of the drag coefficient and Strouhal number results are grouped in Table 4.7. In general, the AMR results look closer to those obtained elsewhere than the imposed mesh refinement method can do. This indicates the AMR in conjunction with the IB-LBM model is capable of achieving better accuracy even when simulating unsteady flows.

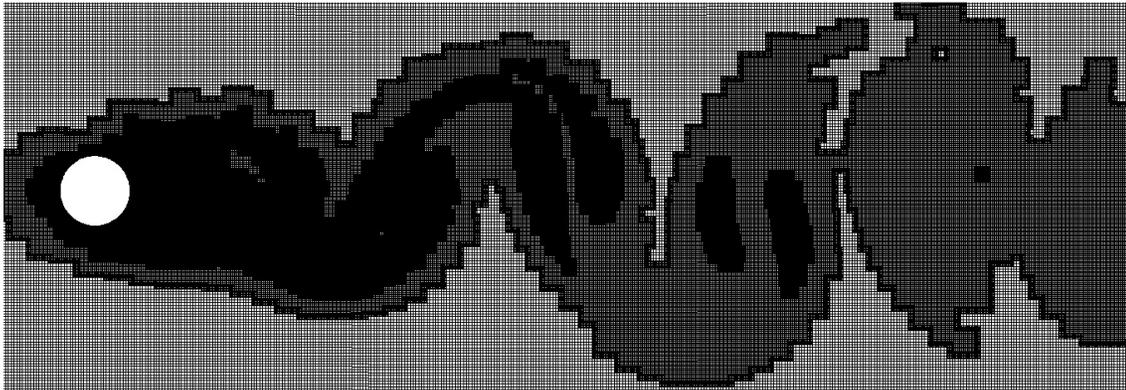
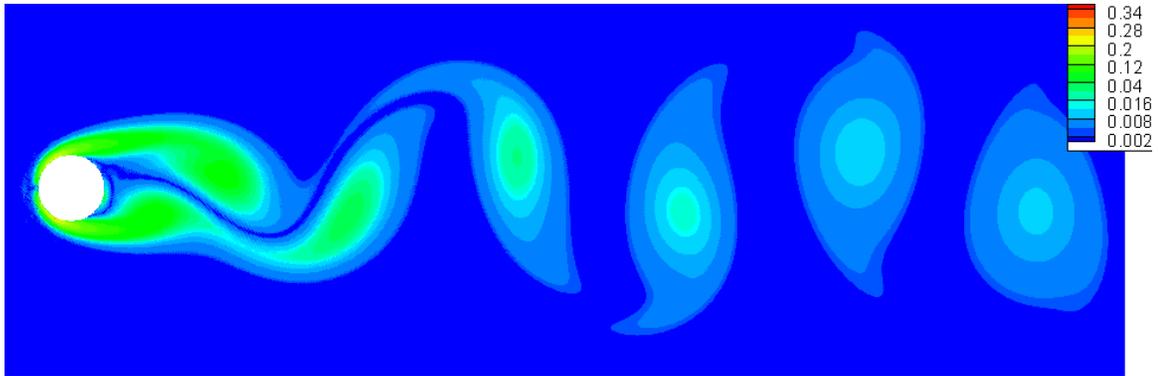


Figure 4.41 Final adaptive mesh for  $Re=100$

Figure 4.42 Vorticity contours on final adaptive mesh for  $Re = 100$ Table 4.7 Comparison of drag coefficients and Strouhal number for  $Re=100$ 

References	Drag Coefficient $C_d$	Strouhal number $St$
Wu et al.[31]	1.364	0.163
Saiki et al.[54]	1.26	0.171
Sui et al.[32]	1.438	0.166
Russell et al.[22]	1.43	0.175
Imposed mesh refinement (Sec. 4.3.1)	1.242	0.167
AMR	1.310	0.172

#### 4.6.5. Computation cost and accuracy comparison between different meshing techniques

To further demonstrate the advantage of mesh refinement methods, detailed statistics are grouped into Table 4.8 and Table 4.9 for the cylinder flow cases at  $Re=40$  and  $100$ , respectively.

Note that this study used Intel Core i5-3337U CPU for all test cases and, in both tables, the drag coefficient is selected as index of accuracy while the numbers of nodes, cells, computation time units are counted as representation of computation expenses.

Unfolding from an identical coarse mesh with  $\Delta x = \Delta y = 1$ , three different meshing techniques for both cases are compared, namely, imposed multi-level local mesh refinement, adaptive mesh refinement (AMR), and uniformly refined mesh with  $\Delta x = \Delta y = 1/8$  everywhere. In a relative sense, the IB-LBM result on the uniformly refined mesh may be assumed with the best accuracy, as it has the finest grid covering the entire computational domain, preventing numerical interpolation error from taking place due to the transition of grid resolution. However, uniform refinement can also be considered as aimless refinement, which leads to exponential increase in mesh size and, in turn, fairly significant increase in computational cost.

For example, in the steady flow case of  $Re=40$  (see Table 4.8), the uniformly refined mesh requires 14554 time units for the computation, which is nearly 3 times more time than the computation time consumed by using the imposed multi-level local mesh refinement method. In this sense, the imposed local mesh refinement option is advantageous over the uniform refinement option. However, if comparing the imposed local mesh refinement method to AMR, the latter looks much more preferable, since AMR can reduce the mesh size by nearly half if compared to the mesh size using imposed refinement and, accordingly, the AMR computation time is only half of the time spent with the imposed mesh refinement method. In terms of solution accuracy, the two different mesh refinement methods leads to the two resulting drag coefficient at two different sides of the

baseline value on the uniformly refined mesh. A more careful comparison reveals that the AMR result is closed to the baseline value than the imposed mesh refinement can do. Overall, AMR looks more recommendable in the case of steady flow simulation.

Table 4.8 Statistics for Re=40

	<b>Uniform course mesh</b>	<b>Imposed refinement</b>	<b>AMR</b>	<b>Uniform refinement</b>
<b>Drag coefficient</b>	1.710	1.493	1.556	1.538
<b>Number of nodes</b>	131072	524288	314594	8388608
<b>Number of cells</b>	130305	521220	308320	8339520
<b>Number of computation time units</b>	261	4767	2340	14554

Likewise, the comparison shown in Table 4.9 for the unsteady flow of Re=100, which employs the same initial mesh as in the Re=40 case, delivers practically the same conclusion as Table 4.8 have done. The better outcome achieved in this unsteady flow case is that the multi-level locally refined mesh needs only one-sixteenth of the nodes occupied by the uniformly refined mesh, and takes about one third of the computation time required by the uniformly refined mesh. As well, AMR even outperforms the multi-level local mesh refinement in its imposed version in terms of mesh size and computation time reductions, remaining in the advantageous position in the case of unsteady flow simulation.

Table 4.9 Statistics for Re=100

	<b>Uniform coarse mesh</b>	<b>Imposed refinement</b>	<b>AMR</b>	<b>Uniform refinement</b>
<b>Drag coefficient</b>	1.455	1.275	1.310	1.280
<b>Number of nodes</b>	131072	524288	470077	8388608
<b>Number of cells</b>	130505	521220	462694	8339520
<b>Number of computation time units</b>	261	4804	3712	14681

## Chapter 5. Concluding and Outlooking Remarks

### 5.1. Conclusions

This study has developed a novel numerical framework that incorporates both the feedback forcing based immersed boundary (IB) model and the "bubble" function based mesh refinement algorithm into the lattice Boltzmann method (LBM).

To the author's knowledge, no IB-LBM model has been found, prior to this study, that can successfully get the feedback forcing approach and the "bubble" function in a conjunctive use.

The present hybrid approach is capable of simulating an extended class of incompressible fluid flow problems, including the challenging ones associated with moving obstacles immersed in a flowing fluid. To justify the present AMR-IB-LBM approach, a series of test cases are numerically investigated, including flows over a stationary or moving cylinder at  $Re=40$  and  $100$ , two stationary gapped cylinders, and two cylinders moving towards each other. Numerical results obtained in this study are discussed in detail and compared with other available data reported in literature.

The satisfactory agreement between the present results and those obtained elsewhere, as well as the demonstrated improvement of numerical solution accuracy and reduction of computation time owing to the AMR techniques embedded in the present model makes this AMR-IB-LBM approach look promising for its further applications to practical problems of more engineering interests.

## 5.2. Recommendations

The following future developments are expected to greatly enhance the capabilities of the present AMR-IB-LBM framework:

(a) The current criterion is of physical type, and requests some adjustment of definition when test case changes from one to another, e.g., from the  $Re=40$  case in Section 4.6.2 to the  $Re=100$  case in Section 4.6.4, even if the governing mathematical model itself remains unchanged. Therefore, unlike the physical criteria that turn out to be on ad hoc basis for each new application, a new type of criterion is expected that should provide more generality for AMR implementation.

(b) The present hybrid model has been applied to two-dimensional flow simulations in this study. Since engineering problems sometimes are difficult to be reduced from the real three-dimensional nature to a two-dimensional stylization, enhancing the current AMR-IB-LBM approach with three dimensionality will certainly render the model more anticipated functionalities.

(c) Application of LBM to simulations of complex flow problems associated with sophisticated geometries inevitably leads to a large-size computation. This calls for high performance computation technology so as to get the computation of real-world flow problems significantly accelerated. Therefore, of high priority is some extensive work to enable the present hybrid model to run with the aid of MPI or GPU technology.

## References

1. U. Frisch, B. Hasslacher, Y. Pomeau. "Lattice-gas automata for the Navier-Stokes equations", *Physical Review Letters*, 56: 1505-1508, 1986.
2. A. L. Garcia, J. B. Bell, W. Y. Crutchfield, B. J. Alder. "Adaptive mesh and algorithm refinement using direct simulation Monte Carlo", *Journal of computational Physics*, 154(1), 134-155, 1999.
3. M. Wang, Z. Li. "Simulations for gas flows in microgeometries using the direct simulation Monte Carlo method", *International Journal of Heat and Fluid Flow*, 25(6): 975-985, 2004.
4. G. R. McNamara, G. Zanetti. "Use of the Boltzmann equation to simulate lattice automata", *Physical Review Letters*, 61: 2332-2335, 1988.
5. F. Higuera, J. Jimenez. "Boltzmann approach to lattice gas simulations", *Europhysics Letters*, 9: 663-668, 1989.
6. S. Chen, H. Chen, D. Martinez, W. Matthaeus. "Lattice Boltzmann model for simulation of magneto-hydrodynamics", *Physical Review Letters*, 67: 3776-3779, 1991.
7. Y. H. Qian, D. d'Humieres, P. Lallemand. "Lattice BGK models for Navier-Stokes equation", *Euro physics Letters*, 17: 479-484, 1992.
8. D. d'Humières. "Generalized lattice Boltzmann equations". *Rarefied gas dynamics- Theory and simulations*, 159: 450-458, 1994.
9. J. Wackers, G. Deng, A. Leroyer, P. Queutey, M. Visonneau. "Adaptive grid refinement for hydrodynamic flows", *Computers & Fluids*, 55, 85-100, 2012.
10. Z. Guo, T.S. Zhao. "Lattice Boltzmann model for incompressible flows through porous media", *Physical Review E*, 66, 036304, 2002.
11. J. Tolke, S. Freudiger, M. Krafczyk. "An adaptive scheme using hierarchical grids for lattice Boltzmann multi-phase flow simulations". *Computers & Fluids*, 35: 820-830, 2006.
12. Z. Yu, L. S. Fan. "An interaction potential based lattice Boltzmann method with adaptive mesh refinement (AMR) for two-phase flow simulation", *Journal of Computational Physics*, 228: 6456-6478, 2009.
13. H. D. Chen, S. Kandasamy, S. Orszag. "Extended Boltzmann kinetic equation for turbulent flows", *Science*, 301: 633-636, 2003.
14. A. J. C. Ladd, R. Verberg. "Lattice-Boltzmann simulations of particle-fluid suspensions", *Journal of Statistical Physics*, 104(5-6): 1191-1251, 2001.
15. J. Zhang. "Lattice Boltzmann method for microfluidics: models and applications", *Microfluidics and Nanofluidics*, 10(1): 1-28, 2011.
16. Z. Guo, T. S. Zhao. "A Lattice Boltzmann model for convection heat transfer in porous media", *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, 47(2): 157-177, 2005.

17. F. Ehsan, M. Farhadi, K. Sedighi. "Lattice Boltzmann simulation of natural convection heat transfer in eccentric annulus." *International journal of thermal sciences*, 49(12): 2353-2362, 2010.
18. S. Chen, G. D. Doolen. "Lattice Boltzmann method for fluid flows", *Annual Review of Fluid Mechanics*, 30:329–364, 1998.
19. X. X. Guo, C. W. Zhong, C. S. Zhuo, J. Cao. "Multiple-relaxation-time lattice Boltzmann method for study of two-lid-driven cavity flow solution multiplicity", *Theoretical and Computational Fluid Dynamics*, 28(2), 215-231, 2014.
20. C. W. Hirt, A. A. Amsde, J. L. Cook. "An arbitrary Lagrangian-Eulerian computing method for all flow speeds". *Journal of Computational Physics*, 14:227–253, 1974.
21. R. W. Anderson, N. S. Elliott, R. B. Pember. "An arbitrary Lagrangian-Eulerian method with adaptive mesh refinement for the solution of the Euler equations", *Journal of Computational Physics*, 199:598–617, 2004.
22. D. Russell, Z. J. Wang. "A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow", *Journal of Computational Physics*, 191:177–205, 2003.
23. J. J. Quirk. "An alternative to unstructured grids for computing gas dynamic flows around arbitrary complex two-dimensional bodies", *Computers and Fluids*, 23:125–142, 1994.
24. C. S. Peskin. "Numerical analysis of blood flow in the heart", *Journal of Computational Physics*, 25:220–252, 1977.
25. C. S. Peskin. "The immersed boundary method", *Acta Numerica*, 11:479–517, 2002.
26. D. Goldstein, R. Handler, L. Sirovich. "Modeling a no-slip flow boundary with an external force field". *Journal of Computational Physics*, 105:354–366, 1993.
27. E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof. "Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations", *Journal of Computational Physics*, 161:35–60, 2000.
28. Z. G. Feng, E. E. Michaelides. "The immersed boundary–lattice Boltzmann method for solving fluid–particles interaction problems", *Journal of Computational Physics*, 195:602–628, 2004.
29. Z. G. Feng, E. E. Michaelides. "Proteus: a direct forcing method in the simulations of particulate flows". *Journal of Computational Physics*, 202:20–51, 2005
30. X. D. Niu, C. Shu, Y. T. Chew, Y. Peng. "A momentum exchanged-based immersed boundary-lattice Boltzmann method for simulation incompressible viscous flows", *Physics Letters A*, 354: 173-182, 2006.
31. J. Wu, C. Shu. "Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications", *Journal of Computational Physics*, 228: 1963-1979, 2009.
32. J. Wu, C. Shu, Y. H. Zhang. "Simulation of incompressible viscous flows around moving objects by a variant of immersed boundary-lattice Boltzmann method", *International Journal for Numerical Methods in Fluids*, 62:327-354, 2010.

33. S.M. Dash, T.S. Lee, T.T. Tim, H. Huang. “A flexible forcing three dimension IB–LBM scheme for flow past stationary and moving spheres”, *Computers & Fluids*, 95: 159-170, 2014.
34. G. W. Peng, H. W. Xi, C. Duncan, S. H. Chou. “Finite volume scheme for the lattice Boltzmann method on unstructured meshes”, *Physical Review E*, 59(4): 4675-4682, 1999.
35. D. Z. Yu, R. W. Mei, W. Shyy. “A multi-block lattice Boltzmann method for viscous fluid flows”. *International journal for numerical methods in fluids*, 39: 99-120, 2002.
36. J. Tolke, M. Krafczyk, E. Rank. “A multigrid solver for the discrete Boltzmann equation”. *Journal of Statistical Physics*, 107: 573-591, 2002.
37. X. M. Sun, F. He, “A high-order meshless Bhatnagar-Gross-Kross scheme based on point collocation”. *Chinese Physical Letter*, 21:233-236, 2004.
38. M. J. Berger and J. Olinger. “Adaptive mesh refinement for hyperbolic partial differential equations”, *Journal of Computational Physics*, 53, 484–512, 1984.
39. Y. Sui, Y. T. Chew, P. Roy, H. T. Low. “A hybrid immersed-boundary and multi-block lattice Boltzmann method for simulating fluid and moving-boundaries interactions”, *International Journal for Numerical Methods in Fluids*, 53: 1727-1754, 2007.
40. J. Wu, C. Shu. “An improved immersed boundary–lattice Boltzmann method for simulating three-dimensional incompressible flows”, *Journal of Computational Physics*, 229, 5022–5042, 2010.
41. B. Crouse, E. Rank, M. Krafczyk, J. Tolke. “A LB-based approach for adaptive flow simulations”. *International Journal of Modern Physics B*, 17: 109-112, 2002.
42. P. Neumann, T. Neckel. “A dynamic mesh refinement technique for Lattice Boltzmann simulations on octree-like grids”, *Computational Mechanics*, 51: 237-253, 2013.
43. A. Fakhari, T. Lee. “Finite-difference lattice Boltzmann method with a block-structured adaptive-mesh-refinement technique”, *Physical Review E*, 89, 033310, 2014.
44. W. X. Huang, H. J. Sung. “Simulation of flexible filaments in a uniform flow by the immersed boundary method”. *Journal of Computational Physics*, 226(2): 2206-2228, 2007.
45. M. Geier, A. Greiner, J. G. Korvink. “Bubble functions for the lattice Boltzmann method and their application to grid refinement”, *The European Physical Journal*, 171:173-179, 2009.
46. S. Hou, Q. Zou, S. Chen, G. Doolen, A.C. Cogley. “Simulation of cavity flow by the lattice Boltzmann method”. *Journal of Computational Physics*, 118: 329–347, 1995.
47. P. Lallemand, L. S. Luo. “Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance and stability”. *Physical Review E*, 61(6): 6546-6562, 2000.
48. S. Chen, D. Martinez, R.W. Mei. “On boundary conditions in lattice Boltzmann methods”. *Physics of Fluids*, 8:2257, 1996.

49. Z. L. Guo, C. G. Zhang, B. C. Shi. “Non-equilibrium extrapolation method for velocity and boundary conditions in the lattice Boltzmann method”. *Chinese Physics*, 11: 0366-0374, 2002.
50. Z. L. Guo, C. G. Zheng, B. C. Shi. “Discrete lattice effects on the forcing term in the lattice Boltzmann method”, *Physical Review E*, 65(4): 046308, 2002.
51. M.C. Lai, C.S. Peskin. “An immersed boundary method with formal second-order accuracy and reduced numerical viscosity”, *Journal of Computational Physics*, 160: 705–719, 2000.
52. Fornberg B. “A numerical study of steady viscous flow past a circular cylinder”, *Journal of Fluid Mechanics*, 98:819–855, 1980.
53. R. K. Shukla, M. Tatineni, X, Zhong. “Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier–Stokes equations”, *Journal of Computational Physics*, 224:1064–1094, 2007.
54. E.M. Saiki, S. Biringen. “Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method”, *Journal of Computational Physics*, 123: 450–465, 1996.
55. S. Kang. “Characteristics of flow over two circular cylinders in a side-by-side arrangement at low Reynolds numbers”, *Physics of Fluids* , 15:2486–2498, 2003.